

SIEMENS

SIMATIC

S7 Automate programmable S7-1200

Manuel système


Avant-propos


Présentation du produit	1
Nouvelles fonctions	2
Logiciel de programmation STEP 7	3
Installation	4
Concepts concernant les automates programmables	5
Configuration des appareils	6
Concepts de programmation	7
Instructions de base	8
Instructions avancées	9
Instructions technologiques	10
Communication	11
Serveur Web	12
Processeur de communication et Modbus TCP	13
Communication TeleService (courrier électronique SMTP)	14
Outils en ligne et de diagnostic	15
Caractéristiques techniques	A
Calcul d'un bilan de consommation	B
Numéro de référence	C
Remplacement de l'appareil et compatibilité des pièces de rechange	D


Mentions légales

Signalétique d'avertissement

Ce manuel donne des consignes que vous devez respecter pour votre propre sécurité et pour éviter des dommages matériels. Les avertissements servant à votre sécurité personnelle sont accompagnés d'un triangle de danger, les avertissements concernant uniquement des dommages matériels sont dépourvus de ce triangle. Les avertissements sont représentés ci-après par ordre décroissant de niveau de risque.

 DANGER
signifie que la non-application des mesures de sécurité appropriées entraîne la mort ou des blessures graves.

 ATTENTION
signifie que la non-application des mesures de sécurité appropriées peut entraîner la mort ou des blessures graves.

 PRUDENCE
signifie que la non-application des mesures de sécurité appropriées peut entraîner des blessures légères.

IMPORTANT
signifie que la non-application des mesures de sécurité appropriées peut entraîner un dommage matériel.


En présence de plusieurs niveaux de risque, c'est toujours l'avertissement correspondant au niveau le plus élevé qui est reproduit. Si un avertissement avec triangle de danger prévient des risques de dommages corporels, le même avertissement peut aussi contenir un avis de mise en garde contre des dommages matériels.

Personnes qualifiées

L'appareil/le système décrit dans cette documentation ne doit être manipulé que par du **personnel qualifié** pour chaque tâche spécifique. La documentation relative à cette tâche doit être observée, en particulier les consignes de sécurité et avertissements. Les personnes qualifiées sont, en raison de leur formation et de leur expérience, en mesure de reconnaître les risques liés au maniement de ce produit / système et de les éviter.

Utilisation des produits Siemens conforme à leur destination

Tenez compte des points suivants:

 ATTENTION
Les produits Siemens ne doivent être utilisés que pour les cas d'application prévus dans le catalogue et dans la documentation technique correspondante. S'ils sont utilisés en liaison avec des produits et composants d'autres marques, ceux-ci doivent être recommandés ou agréés par Siemens. Le fonctionnement correct et sûr des produits suppose un transport, un entreposage, une mise en place, un montage, une mise en service, une utilisation et une maintenance dans les règles de l'art. Il faut respecter les conditions d'environnement admissibles ainsi que les indications dans les documentations afférentes.

Marques de fabrique

Toutes les désignations repérées par ® sont des marques déposées de Siemens AG. Les autres désignations dans ce document peuvent être des marques dont l'utilisation par des tiers à leurs propres fins peut enfreindre les droits de leurs propriétaires respectifs.

Exclusion de responsabilité

Nous avons vérifié la conformité du contenu du présent document avec le matériel et le logiciel qui y sont décrits. Ne pouvant toutefois exclure toute divergence, nous ne pouvons pas nous porter garants de la conformité intégrale. Si l'usage de ce manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition.

Avant-propos

Objet du manuel

La famille S7-1200 est constituée d'automates programmables (API) utilisables dans des applications d'automatisation variées. Sa forme compacte, son faible prix et son important jeu d'instructions en font une solution idéale pour la commande d'applications très variées. Les modèles S7-1200 ainsi que l'outil de programmation STEP 7 (Page 37) se basant sur Windows vous offrent la souplesse nécessaire pour résoudre vos problèmes d'automatisation.

Ce manuel qui fournit des informations sur l'installation et la programmation des automates S7-1200 est conçu pour des ingénieurs, des programmeurs, des installateurs et des électriciens ayant une connaissance générale des automates programmables.

Connaissances fondamentales requises

Pour comprendre ce manuel, vous devez avoir des connaissances générales sur l'automatisation et les automates programmables.

Domaine de validité

Ce manuel décrit les produits suivants :

- STEP 7 V15 Basic et Professional (Page 37)
- CPU S7-1200, version de firmware V4.2.3

Vous trouverez la liste complète des produits S7-1200 décrits dans ce manuel dans les caractéristiques techniques (Page 1429).

Certification, label CE, homologation C-Tick et autres homologations

Reportez-vous aux caractéristiques techniques (Page 1429) pour plus d'informations à ce sujet.

Service et assistance

En complément de notre documentation, Siemens propose son expertise technique sur Internet et via le site Web du service client (<https://support.industry.siemens.com/cs/ww/fr>).

Adressez-vous à votre agence Siemens si certaines de vos questions techniques restent sans réponse, si vous voulez connaître les offres de formation ou si vous désirez commander des produits S7. Comme ce personnel est techniquement formé et a des connaissances très pointues sur vos activités, vos processus et vos industries, ainsi que sur les différents produits Siemens que vous utilisez, il peut apporter les réponses les plus rapides et les plus efficaces possibles à tout problème que vous pourriez rencontrer.

Documentation et information

Le S7-1200 et STEP 7 proposent une grande variété de documents et autres ressources contenant les informations techniques dont vous avez besoin.

- Le manuel système de l'automate programmable S7-1200 fournit des informations spécifiques sur le fonctionnement, la programmation et les caractéristiques de toute la gamme du S7-1200.

Le manuel système est disponible au format électronique (PDF). Vous pouvez télécharger ou consulter ce manuel et les autres manuels électroniques sur le site Web Industry Online Support de Siemens (<https://support.industry.siemens.com/cs/ww/fr>). Le manuel système se trouve également sur le disque de documentation qui accompagne chaque CPU S7-1200.

- Le système d'information en ligne de STEP 7 offre un accès immédiat aux informations conceptuelles et aux instructions spécifiques qui décrivent le fonctionnement et les fonctionnalités du progiciel de programmation et le principe de fonctionnement des CPU SIMATIC.
- Le site Web Industry Online Support de Siemens (<https://support.industry.siemens.com/cs/ww/fr>) permet d'accéder aux versions électroniques (PDF) de la documentation SIMATIC, y compris le manuel système et le système d'information STEP 7. Les documents existants sont accessibles grâce au lien Product Support. Avec cet accès à la documentation en ligne, vous pouvez également extraire des textes de divers documents pour créer votre propre manuel personnalisé. Les mises à jour des manuels système précédemment publiés sont également disponibles sur le site Industry Online Support de Siemens.

Vous pouvez accéder à la documentation en ligne en cliquant sur "mySupport" sur le côté gauche de la page et en sélectionnant "Documentation" dans les menus de navigation. Vous devez ouvrir une session en tant qu'utilisateur enregistré pour pouvoir utiliser les fonctions de documentation mySupport.

- Le site Web Siemens Industry Online Support fournit également des FAQ et autres documents utiles concernant le S7-1200 et STEP 7.
- Vous pouvez également suivre ou rejoindre des discussions sur les produits sur le forum technique Service & Support (<https://support.industry.siemens.com/tf/ww/en/?Language=en&siteid=csius&treeLang=en&groupid=4000002&extranet=standard&viewreg=WW&nodeid0=34612486>). Ces forums vous permettent de dialoguer avec différents experts produits.
 - Forum pour le S7-1200 (<https://support.industry.siemens.com/tf/ww/en/threads/237?title=simatic-s7-1200&skip=0&take=10&orderBy=LastPostDate+desc>)
 - Forum pour STEP 7 Basic (<https://support.industry.siemens.com/tf/ww/en/threads/243?title=step-7-tia-portal&skip=0&take=10&orderBy=LastPostDate+desc>)

Notes relatives à la sécurité

Siemens commercialise des produits et solutions comprenant des fonctions de sécurité industrielle qui contribuent à une exploitation sûre des installations, solutions, machines, équipements et réseaux.

Pour garantir la sécurité des installations, systèmes, machines et réseaux contre les cybermenaces, il est nécessaire d'implémenter (et de préserver) un concept de sécurité industrielle global et moderne. Les produits et solutions de Siemens ne constituent qu'une partie d'un tel concept.

Il incombe au client d'empêcher tout accès non autorisé à ses installations, systèmes, machines et réseaux. Les systèmes, machines et composants doivent uniquement être connectés au réseau d'entreprise ou à Internet si et dans la mesure où c'est nécessaire et si des mesures de protection correspondantes (p. ex. utilisation de pare-feux et segmentation du réseau) ont été prises.

En outre, vous devez tenir compte des recommandations de Siemens concernant les mesures de protection correspondantes. Pour plus d'informations sur la sécurité industrielle, rendez-vous sur (<http://www.siemens.com/industrialsecurity>).

Les produits et solutions Siemens font l'objet de développements continus pour être encore plus sûrs. Siemens vous recommande donc vivement d'effectuer des actualisations dès que les mises à jour correspondantes sont disponibles et de ne toujours utiliser que les versions de produit actuelles. L'utilisation de versions obsolètes ou qui ne sont plus prises en charge peut augmenter le risque de cybermenaces.

Afin d'être informé des mises à jour produit dès qu'elles surviennent, abonnez-vous au flux RSS Siemens Industrial Security sous (<http://www.siemens.com/industrialsecurity>).

Sommaire

	Avant-propos	3
1	Présentation du produit	27
1.1	Introduction à l'automate S7-1200	27
1.2	Possibilités d'extension de la CPU	31
1.3	Tableaux IHM basiques	33
2	Nouvelles fonctions	35
3	Logiciel de programmation STEP 7	37
3.1	Besoins du système	38
3.2	Différentes vues pour faciliter votre travail.....	39
3.3	Simplicité des outils	41
3.3.1	Insertion d'instructions dans votre programme utilisateur	41
3.3.2	Accès aux instructions depuis la barre d'outils "Favoris".....	41
3.3.3	Création d'une équation complexe avec une instruction simple.....	42
3.3.4	Ajout d'entrées ou de sorties à une instruction CONT ou LOG.....	44
3.3.5	Instructions extensibles.....	44
3.3.6	Sélection d'une version d'une instruction.....	45
3.3.7	Modification de l'apparence et de la configuration de STEP 7	45
3.3.8	Glisser-déplacer entre les éditeurs	46
3.3.9	Changement de l'état de fonctionnement de la CPU.....	47
3.3.10	Modification du type d'appel pour un DB	48
3.3.11	Déconnexion temporaire d'appareils d'un réseau	49
3.3.12	Désenfichage virtuel d'appareils de la configuration	50
3.4	Compatibilité amont	51
4	Installation	53
4.1	Conseils pour l'installation d'appareils S7-1200	53
4.2	Bilan de consommation.....	56
4.3	Procédures d'installation et de désinstallation.....	58
4.3.1	Dimensions de montage pour les appareils S7-1200.....	58
4.3.2	Installation et désinstallation de la CPU	62
4.3.3	Installation et désinstallation d'un SB, d'un CB ou d'un BB	64
4.3.4	Installation et désinstallation d'un SM.....	66
4.3.5	Installation et désinstallation d'un CM ou d'un CP.....	68
4.3.6	Démontage et remontage du bornier de connexion S7-1200	69
4.3.7	Installation et désinstallation du câble d'extension	70
4.3.8	Adaptateur TS Adapter (TeleService).....	72
4.3.8.1	Connexion de l'adaptateur TeleService.....	72
4.3.8.2	Installation de la carte SIM	74
4.3.8.3	Installation de l'ensemble adaptateur TS sur un profilé support.....	75
4.3.8.4	Montage de l'adaptateur TS sur un panneau	76

4.4	Conseils de câblage.....	77
5	Concepts concernant les automates programmables	85
5.1	Exécution du programme utilisateur	85
5.1.1	Etats de fonctionnement de la CPU.....	89
5.1.2	Traitement du cycle à l'état MARCHE	93
5.1.3	Blocs d'organisation (OB)	94
5.1.3.1	OB de cycle de programme	94
5.1.3.2	OB de démarrage.....	95
5.1.3.3	OB d'alarme temporisée	95
5.1.3.4	OB d'alarme cyclique	96
5.1.3.5	OB d'alarme de processus.....	97
5.1.3.6	OB d'erreur de temps.....	98
5.1.3.7	OB d'alarme de diagnostic.....	99
5.1.3.8	OB de débrogage/enfichage de modules	101
5.1.3.9	OB de défaillance du châssis ou de la station	102
5.1.3.10	OB d'alarme horaire	103
5.1.3.11	OB d'état	103
5.1.3.12	OB de mise à jour	104
5.1.3.13	OB de profil	104
5.1.3.14	OB MC-Servo et MC-Interpolator.....	104
5.1.3.15	MC-PreServo	105
5.1.3.16	MC-PostServo.....	106
5.1.3.17	Priorités d'exécution et mise en file d'attente des événements	106
5.1.4	Surveillance et configuration du temps de cycle.....	111
5.1.5	Mémoire de la CPU.....	113
5.1.5.1	Mémento système et mémento de cadence	115
5.1.6	Mémoire tampon de diagnostic.....	118
5.1.7	Horloge temps réel.....	119
5.1.8	Configuration des sorties lors d'un passage de MARCHE à ARRET	120
5.2	Stockage des données, zones de mémoire, E/S et adressage.....	121
5.2.1	Accès aux données du S7-1200.....	121
5.3	Traitement des valeurs analogiques.....	128
5.4	Types de données	129
5.4.1	Types de données Bool, Byte, Word et DWord	131
5.4.2	Types de données entiers.....	132
5.4.3	Types de données réels à virgule flottante	132
5.4.4	Types de données "date et heure".....	133
5.4.5	Types de données "caractère et chaîne de caractères"	135
5.4.6	Type de données "tableau".....	137
5.4.7	Type de données "structure de données".....	138
5.4.8	Type de données API	138
5.4.9	Type de données pointeur variant	139
5.4.10	Accès à une "tranche" d'un type de données de variable	139
5.4.11	Accès à une variable par un type de données ajouté AT	141
5.5	Utilisation d'une carte mémoire.....	143
5.5.1	Insertion d'une carte mémoire dans la CPU	143
5.5.2	Configuration des paramètres de mise en route de la CPU avant copie du projet dans la carte mémoire	147
5.5.3	Utilisation de la carte mémoire en tant que carte "transfert"	147

5.5.4	Utilisation de la carte mémoire en tant que carte "programme"	150
5.5.5	Mise à jour du firmware.....	153
5.6	Récupération en cas d'oubli du mot de passe.....	157
6	Configuration des appareils.....	159
6.1	Insertion d'une CPU	160
6.2	Chargement de la configuration d'une CPU connectée	162
6.3	Ajout de modules à la configuration.....	164
6.4	Commande de configuration.....	165
6.4.1	Avantages et applications d'une commande de configuration	165
6.4.2	Configuration de l'installation centrale et de modules optionnels.....	165
6.4.3	Exemple de commande de configuration	173
6.5	Changer d'appareil.....	177
6.6	Configuration du fonctionnement de la CPU	177
6.6.1	Présentation.....	177
6.6.2	Configuration des temps de filtre des entrées TOR	179
6.6.3	Capture d'impulsions	181
6.7	Configuration du multilinguisme.....	183
6.8	Configuration des paramètres des modules.....	185
6.9	Configuration de la CPU pour la communication.....	187
6.10	Synchronisation de l'heure.....	189
7	Concepts de programmation	191
7.1	Principes de conception d'un système d'automatisation	191
7.2	Organisation de votre programme utilisateur	192
7.3	Utilisation de blocs pour structurer votre programme.....	194
7.3.1	Bloc d'organisation (OB).....	195
7.3.2	Fonction (FC).....	197
7.3.3	Bloc fonctionnel (FB).....	197
7.3.4	Bloc de données (DB).....	199
7.3.5	Création de blocs de code réutilisables	200
7.3.6	Transmission de paramètres aux blocs	201
7.4	Comprendre le concept de cohérence des données.....	204
7.5	Langage de programmation.....	206
7.5.1	Schéma à contacts (CONT).....	206
7.5.2	Logigramme (LOG).....	207
7.5.3	SCL	208
7.5.3.1	Éditeur de programme SCL	208
7.5.3.2	Expressions et opérations SCL	210
7.5.3.3	Adressage indexé avec les instructions PEEK et POKE.....	214
7.5.4	EN et ENO pour CONT, LOG et SCL	216
7.6	Protection	218
7.6.1	Protection d'accès pour la CPU	218
7.6.2	Mémoire de chargement externe.....	221
7.6.3	Protection du savoir-faire.....	222

7.6.4	Protection contre la copie	223
7.7	Chargement d'éléments de votre programme dans la CPU	225
7.8	Synchronisation de la CPU en ligne et du projet hors ligne	229
7.9	Chargement à partir de la CPU en ligne	231
7.9.1	Comparaison entre la CPU en ligne et la CPU hors ligne	231
7.10	Débugage et test du programme	232
7.10.1	Visualisation et forçage de données dans la CPU.....	232
7.10.2	Tables de visualisation et tables de forçage permanent	232
7.10.3	Affichage de l'usage des références croisées	233
7.10.4	Structure d'appel permettant de constater la hiérarchie d'appel	234
8	Instructions de base.....	237
8.1	Opérations logiques sur bits	237
8.1.1	Opérations combinatoires sur bits	237
8.1.2	Instructions Mise à 1 et Mise à 0	240
8.1.3	Instructions Front montant et Front descendant	243
8.2	Temporisations.....	246
8.3	Compteurs.....	255
8.4	Comparaison.....	262
8.4.1	Instructions Comparer valeurs.....	262
8.4.2	IN_RANGE (Valeur dans la plage) et OUT_RANGE (Valeur en dehors de la plage)	263
8.4.3	OK (Contrôler validité) et NOT_OK (Contrôler invalidité)	264
8.4.4	Instructions de comparaison de variante et tableau	265
8.4.4.1	Instructions de comparaison d'égalité et de non-égalité.....	265
8.4.4.2	Instructions de comparaison nulle	266
8.4.4.3	IS_ARRAY (vérifier le TABLEAU).....	267
8.5	Fonctions mathématiques.....	268
8.5.1	CALCULATE (Calculer)	268
8.5.2	Instructions Addition, Soustraction, Multiplication et Division	269
8.5.3	MOD (Calculer le reste de la division)	270
8.5.4	NEG (Créer le complément à deux).....	271
8.5.5	INC (Incrémenter) et DEC (Décrémenter)	272
8.5.6	ABS (Valeur absolue)	272
8.5.7	MIN (Calculer le minimum) et MAX (Calculer le maximum)	273
8.5.8	LIMIT (Définir une limite).....	274
8.5.9	Instructions exponentielle, logarithmique et trigonométriques.....	275
8.6	Transfert.....	277
8.6.1	MOVE (Copier valeur), MOVE_BLK (Copier zone), UMOVE_BLK (Copier zone contiguë) et MOVE_BLK_VARIANT (Copier zone).....	277
8.6.2	Deserialize	281
8.6.3	Serialize	284
8.6.4	FILL_BLK (Compléter zone) et UFILL_BLK (Compléter zone contiguë)	287
8.6.5	SWAP (Permuter octets).....	289
8.6.6	LOWER_BOUND (Lire la limite inférieure d'un ARRAY)	290
8.6.7	UPPER_BOUND (Lire la limite supérieure d'un ARRAY).....	291
8.6.8	Instructions de lecture de mémoire / écriture dans la mémoire.....	293
8.6.8.1	PEEK et POKE (SCL uniquement)	293
8.6.8.2	Lire et écrire des instructions big et little Endian (SCL).....	295

8.6.9	Instructions Variant	296
8.6.9.1	VariantGet (Lire la valeur de variable VARIANT)	296
8.6.9.2	VariantPut (Écrire la valeur dans une variable VARIANT)	297
8.6.9.3	CountOfElements (Interroger le nombre d'éléments ARRAY)	298
8.6.10	Instructions d'héritage	299
8.6.10.1	Instructions FieldRead (Lire champ) et FieldWrite (Ecrire champ)	299
8.7	Conversion	302
8.7.1	CONV (Convertir valeur)	302
8.7.2	Instructions de conversion pour SCL	303
8.7.3	ROUND (Arrondir nombre) et TRUNC (Former un nombre entier)	306
8.7.4	CEIL (Arrondir à l'entier supérieur) et FLOOR (Arrondir à l'entier inférieur)	307
8.7.5	SCALE_X (Mise à l'échelle) et NORM_X (Normaliser)	308
8.7.6	Instructions de conversion de Variante	311
8.7.6.1	VARIANT_TO_DB_ANY (Convertir VARIANT en DB_ANY)	311
8.7.6.2	DB_ANY_TO_VARIANT (Convertir DB_ANY en VARIANT)	312
8.8	Gestion du programme	314
8.8.1	Instructions JMP (Saut si RLO = 1), JMPN (Saut si RLO = 0), et Label (Repère de saut)	314
8.8.2	JMP_LIST (Définir liste de sauts)	315
8.8.3	SWITCH (Branchement conditionnel)	316
8.8.4	RET (Retour de saut)	318
8.8.5	ENDIS_PW (Limiter et valider la légitimation par mot de passe)	319
8.8.6	RE_TRIGR (Redéclencher le temps de surveillance du cycle)	321
8.8.7	STP (Arrêter le programme)	323
8.8.8	Instructions GET_ERROR et GET_ERROR_ID (Interrogation locale des erreurs et des ID d'erreur)	323
8.8.9	RUNTIME (Mesurer le temps d'exécution du programme)	327
8.8.10	Instructions de contrôle SCL	330
8.8.10.1	Présentation des instructions de contrôle SCL	330
8.8.10.2	Instruction IF-THEN	331
8.8.10.3	Instruction CASE	332
8.8.10.4	Instruction FOR	333
8.8.10.5	Instruction WHILE-DO	334
8.8.10.6	Instruction REPEAT-UNTIL	335
8.8.10.7	Instruction CONTINUE	336
8.8.10.8	Instruction EXIT	337
8.8.10.9	Instruction GOTO	338
8.8.10.10	Instruction RETURN	338
8.9	Opérations logiques sur mots	339
8.9.1	Opérations logiques ET, OU et OU EXCLUSIF	339
8.9.2	INV (Former le complément à 1)	339
8.9.3	Instructions DECO (Décoder) et ENCO (Encoder)	340
8.9.4	Instructions SEL (Sélectionner), MUX (Multiplexeur), et DEMUX (Démultiplexeur)	341
8.10	Décalage et rotation	345
8.10.1	Instruction SHR (Décaler à droite) et SHL (Décaler à gauche)	345
8.10.2	Instructions ROR (Rotation à droite) et ROL (Rotation à gauche)	346

9	Instructions avancées	347
9.1	Fonctions date, heure et horloge	347
9.1.1	Instructions pour la date et l'heure	347
9.1.2	Fonctions d'horloge	350
9.1.3	Structure de données TimeTransformationRule	353
9.1.4	SET_TIMEZONE (Sélectionner le fuseau horaire)	354
9.1.5	RTM (Compteurs d'heures de fonctionnement)	355
9.2	Chaînes de caractères et caractères	357
9.2.1	Présentation des données chaînes de caractères (type String)	357
9.2.2	S_MOVE (Déplacer la chaîne de caractères)	357
9.2.3	Instructions de conversion de chaîne	358
9.2.3.1	Instructions S_CONV (Convertir la chaîne de caractères), STRG_VAL (Convertir la chaîne de caractères en valeur numérique) et VAL_STRG (Convertir la valeur numérique en chaîne de caractères)	358
9.2.3.2	Instructions Strg_TO_Chars et Chars_TO_Strg (Convertir de/vers chaîne de caractères et tableau de CHAR)	367
9.2.3.3	Instructions ATH et HTA (Convertir de/vers chaîne de caractères ASCII et nombre hexadécimal)	369
9.2.4	Instructions sur chaîne	371
9.2.4.1	MAX_LEN (Déterminer la longueur maximale d'une chaîne de caractères)	371
9.2.4.2	LEN (Déterminer la longueur d'une chaîne de caractères)	372
9.2.4.3	CONCAT (Concaténer les chaînes de caractères)	372
9.2.4.4	Instructions LEFT, RIGHT et MID (Lire sous-chaînes dans une chaîne de caractères)	374
9.2.4.5	DELETE (Supprimer des caractères dans la chaîne de caractères)	375
9.2.4.6	INSERT (Insérer des caractères dans une chaîne de caractères)	376
9.2.4.7	REPLACE (Remplacer des caractères dans une chaîne de caractères)	377
9.2.4.8	FIND (Trouver des caractères dans une chaîne de caractères)	378
9.2.5	Informations Runtime	379
9.2.5.1	GetSymbolName (Lecture d'un nom de variable au paramètre d'entrée)	379
9.2.5.2	GetSymbolPath (Interroger un nom global composite de l'affectation de paramètre d'entrée)	383
9.2.5.3	GetInstanceName (Lire le nom de l'instance de bloc)	386
9.2.5.4	GetInstancePath (Interroger le nom global composite de l'instance de bloc)	388
9.2.5.5	GetBlockName (Lire le nom du bloc)	390
9.3	Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)	394
9.3.1	Instructions de périphérie décentralisée	394
9.3.2	RDREC et WRREC (Lire l'enregistrement/Écrire l'enregistrement)	395
9.3.3	GETIO (Lire la mémoire image)	398
9.3.4	SETIO (Transférer la mémoire image)	399
9.3.5	GETIO_PART (Lire la zone de mémoire image)	401
9.3.6	SETIO_PART (Transférer la zone de mémoire image)	403
9.3.7	RALRM (Recevoir l'alarme)	404
9.3.8	D_ACT_DP (Activer/désactiver les périphériques PROFINET IO)	408
9.3.9	Paramètre STATUS pour RDREC, WRREC et RALRM	413
9.3.10	Autres instructions	418
9.3.10.1	DPRD_DAT et DPWR_DAT (Lire/écrire des données cohérentes)	418
9.3.10.2	RCVREC (Recevoir l'enregistrement - périphérique I/esclave I)	421
9.3.10.3	PRVREC (Mettre l'enregistrement à disposition - périphérique I/esclave I)	424
9.3.10.4	DPNRM_DG (Lire des données de diagnostic d'un esclave PROFIBUS DP)	426
9.4	PROFIenergy	429

9.5	Alarmes	430
9.5.1	Instructions ATTACH et DETACH (Affecter/dissocier un OB et un événement déclencheur d'alarme)	430
9.5.2	Alarmes cycliques	434
9.5.2.1	SET_CINT (Définir les paramètres de l'alarme cyclique)	434
9.5.2.2	QRY_CINT (Interroger les paramètres de l'alarme cyclique)	436
9.5.3	Alarmes horaires	437
9.5.3.1	SET_TINTL (Définir une alarme horaire)	438
9.5.3.2	CAN_TINT (Annuler l'alarme horaire)	439
9.5.3.3	ACT_TINT (Activer alarme horaire)	440
9.5.3.4	QRY_TINT (Interroger l'état de l'alarme horaire)	441
9.5.4	Alarmes temporisées	442
9.5.5	Instructions DIS_AIRT et EN_AIRT (Retarder/activer les alarmes de priorité supérieure et les erreurs asynchrones)	444
9.6	Messages	446
9.6.1	Gen_UsrMsg (Créer des alarmes de diagnostic utilisateur)	446
9.7	Diagnostic (PROFINET ou PROFIBUS)	449
9.7.1	Opérations de diagnostic	449
9.7.2	RD_SINFO (Lire l'information de déclenchement de l'OB actuel)	450
9.7.3	LED (Lire l'état de la LED)	461
9.7.4	Get_IM_Data (Lire les données d'identification et de maintenance)	462
9.7.5	Get_Name (Lire le nom d'un périphérique PROFINET IO)	464
9.7.6	GetStationInfo (Lire l'adresse IP ou MAC d'un périphérique PROFINET IO)	471
9.7.7	Instruction DeviceStates	479
9.7.7.1	Exemples de configuration de DeviceStates	481
9.7.8	Instruction ModuleStates	485
9.7.8.1	Exemples de configuration de ModuleStates	487
9.7.9	GET_DIAG (Lire l'information de diagnostic)	491
9.7.10	Événements de diagnostic d'une périphérie décentralisée	498
9.8	Impulsion	499
9.8.1	CTRL_PWM (Modulation de largeur d'impulsion)	499
9.8.2	CTRL_PTO (Émettre un train d'impulsions à fréquence prédéfinie)	501
9.8.3	Fonctionnement des sorties d'impulsions	504
9.8.4	Configuration d'une voie d'impulsions pour PWM ou PTO	507
9.9	Recettes et journaux	512
9.9.1	Recettes	512
9.9.1.1	Liste des recettes	512
9.9.1.2	Exemple de recette	513
9.9.1.3	Instructions de transfert des données de recette	517
9.9.1.4	Exemple de programme de recette	521
9.9.2	Journaux de données	524
9.9.2.1	Structure des enregistrements de journaux	524
9.9.2.2	Instructions de gestion des journaux de données	525
9.9.2.3	Utilisation des journaux de données	541
9.9.2.4	Limite de taille des fichiers journaux	542
9.9.2.5	Exemple de programme pour les journaux de données	546
9.10	Gestion des blocs de données	551
9.10.1	CREATE_DB (Créer un bloc de données)	551
9.10.2	Instructions READ_DBL et WRIT_DBL (Lire/écrire un bloc de données en mémoire de chargement)	556

9.10.3	ATTR_DB (Lire les attributs d'un bloc de données).....	559
9.10.4	DELETE_DB (Supprimer le bloc de données).....	561
9.11	Gestion des adresses	563
9.11.1	GEO2LOG (Déterminer l'identification matérielle à partir de l'emplacement)	563
9.11.2	LOG2GEO (Déterminer l'emplacement à partir de l'identification matérielle)	565
9.11.3	IO2MOD (Déterminer l'identification matérielle à partir d'une adresse E/S).....	566
9.11.4	RD_ADDR (Déterminer les adresses E/S à partir de l'identification matérielle).....	568
9.11.5	Type de données système GEOADDR	569
9.12	Codes d'erreur communs pour les instructions avancées	571
10	Instructions technologiques.....	573
10.1	Comptage (compteurs rapides)	573
10.1.1	Instruction CTRL_HSC_EXT (Commande de compteurs rapides).....	574
10.1.1.1	Présentation de l'instruction	574
10.1.1.2	Exemple	575
10.1.1.3	Types de données système (SDT) de l'instruction CTRL_HSC_EXT	578
10.1.2	Utilisation du compteur rapide	584
10.1.2.1	Fonction Synchronisation.....	584
10.1.2.2	Fonction Validation	585
10.1.2.3	Fonction Capture.....	587
10.1.2.4	Fonction Comparaison.....	588
10.1.2.5	Applications.....	589
10.1.3	Configuration d'un compteur rapide	590
10.1.3.1	Type de comptage	592
10.1.3.2	Phase de fonctionnement	593
10.1.3.3	Valeurs initiales	597
10.1.3.4	Fonctions d'entrée.....	597
10.1.3.5	Fonction de sortie.....	598
10.1.3.6	Événements d'alarme	599
10.1.3.7	Affectation d'entrée matérielle	599
10.1.3.8	Affectation d'une sortie matérielle	601
10.1.3.9	Adresses de mémoire d'entrée des compteurs rapides	602
10.1.3.10	Identification matérielle	602
10.1.4	Instruction CTRL_HSC (Commander le compteur rapide) héritée	603
10.1.4.1	Présentation de l'instruction	603
10.1.4.2	Utilisation de CTRL_HSC.....	605
10.1.4.3	Valeur de comptage en cours du compteur	606
10.2	Régulation PID	607
10.2.1	Insertion de l'instruction PID et de l'objet technologique	609
10.2.2	PID_Compact.....	611
10.2.2.1	Instruction PID_Compact	611
10.2.2.2	Limites de la mesure dans l'instruction PID_Compact.....	615
10.2.2.3	Paramètre ErrorBits de l'instruction PID_Compact.....	616
10.2.2.4	Paramètres d'avertissement de l'instruction PID_Compact.....	618
10.2.3	PID_3Step	619
10.2.3.1	Instruction PID_3Step	619
10.2.3.2	Paramètres ErrorBit de l'instruction PID_3Step.....	626
10.2.3.3	Paramètres d'avertissement de l'instruction PID_3Step.....	628
10.2.4	PID_Temp	629
10.2.4.1	Instruction PID_Temp	629
10.2.4.2	Paramètres ErrorBit de l'instruction PID_Temp	640

10.2.4.3	Paramètres Warning de l'instruction PID_Temp	642
10.2.5	Configuration des contrôleurs PID_Compact et PID_3Step	643
10.2.6	Configuration du régulateur PID_Temp	646
10.2.7	Mise en service des régulateurs PID_Compact et PID_3Step	661
10.2.8	Mise en service du régulateur PID_Temp.....	663
10.3	Motion control	674
10.3.1	Phasage	680
10.3.2	Configuration d'un générateur d'impulsions	683
10.3.3	Ouvrir la boucle de commande de mouvement.....	684
10.3.3.1	Configuration de l'axe	684
10.3.3.2	Mise en service	688
10.3.4	Motion control en boucle fermée	694
10.3.4.1	Configuration de l'axe	694
10.3.4.2	ServoOB	701
10.3.4.3	Fonctionnement en régulation de vitesse	703
10.3.4.4	Prise en charge du télégramme 4.....	706
10.3.4.5	Axe de simulation.....	710
10.3.4.6	Adaptation des données	713
10.3.4.7	Commande d'axe utilisant le module TM Pulse.....	724
10.3.5	Configuration de la table de commande TO_CommandTable_PTO	730
10.3.6	Fonctionnement de Motion Control pour S7-1200.....	734
10.3.6.1	Sorties de la CPU utilisées pour Motion Control	734
10.3.6.2	Fins de course matériels et logiciels pour Motion Control	736
10.3.6.3	Référencement	746
10.3.6.4	Limitation d'à-coup.....	753
10.3.7	Instructions Motion Control	754
10.3.7.1	Présentation des instructions MC	754
10.3.7.2	MC_Power (Libérer/bloquer l'axe)	756
10.3.7.3	MC_Reset (Acquitter l'erreur)	759
10.3.7.4	Instruction MC_Home (Référencer l'axe).....	760
10.3.7.5	MC_Halt (Arrêter un axe).....	763
10.3.7.6	MC_MoveAbsolute (Positionner un axe de manière absolue).....	765
10.3.7.7	MC_MoveRelative (Positionner un axe de manière relative).....	768
10.3.7.8	MC_MoveVelocity (Déplacer un axe à une vitesse prédéfinie)	770
10.3.7.9	MC_MoveJog (Déplacer un axe en mode Manuel à vue)	773
10.3.7.10	MC_CommandTable (Exécuter les commandes de l'axe comme séquence de mouvements)	776
10.3.7.11	MC_ChangeDynamic (Modifier les paramètres dynamiques de l'axe).....	779
10.3.7.12	MC_WriteParam (Écrire dans les paramètres d'un objet technologique).....	782
10.3.7.13	Instruction MC_ReadParam (Lire les paramètres d'un objet technologique)	784
10.3.8	Suivi des commandes actives	785
10.3.8.1	Suivi des instructions MC avec paramètre de sortie "Done"	785
10.3.8.2	Surveillance de l'instruction MC_Velocity	790
10.3.8.3	Surveillance de l'instruction MC_MoveJog	794
10.3.9	Valeurs ErrorId et ErrorInfo pour Motion Control	798
11	Communication.....	825
11.1	Liaisons de communication asynchrones	827
11.2	PROFINET	830
11.2.1	Création d'une liaison réseau	832
11.2.2	Configuration du routage local/partenaire.....	833

11.2.3	Affectation d'adresses IP (Internet Protocol)	836
11.2.3.1	Affectation d'adresses IP à des consoles de programmation et des dispositifs réseau	836
11.2.3.2	Vérification de l'adresse IP de votre console de programmation.....	838
11.2.3.3	Affectation d'une adresse IP à une CPU en ligne.....	838
11.2.3.4	Configuration d'une adresse IP pour une CPU dans votre projet.....	840
11.2.4	Test du réseau PROFINET	845
11.2.5	Localisation de l'adresse Ethernet (MAC) sur la CPU	846
11.2.6	Configuration de la synchronisation NTP	848
11.2.7	Temps de mise en route, affectation de nom et d'adresse pour un appareil PROFINET ...	850
11.2.8	Communication ouverte (Open User Communication)	851
11.2.8.1	Protocoles	851
11.2.8.2	TCP et ISO sur TCP	852
11.2.8.3	Services de communication et numéros de port utilisés	853
11.2.8.4	Mode ad hoc	854
11.2.8.5	ID de liaison pour les instructions Open User Communication	854
11.2.8.6	Paramètres pour la liaison PROFINET	857
11.2.8.7	Instructions TSEND_C et TRCV_C.....	862
11.2.8.8	Instructions d'héritage TSEND_C et TRCV_C.....	875
11.2.8.9	Instructions TCON, TDISCON, TSEND et TRCV	884
11.2.8.10	Instructions d'héritage TCON, TDISCON, TSEND et TRCV	895
11.2.8.11	Instruction T_RESET (Couper et rétablir une liaison existante)	906
11.2.8.12	Instruction T_DIAG (Vérifie l'état d'une liaison et lire des informations)	908
11.2.8.13	Instruction TMAIL_C (Envoyer un courriel à l'aide de l'interface Ethernet de la CPU).....	913
11.2.8.14	UDP.....	923
11.2.8.15	TUSEND et TURCV.....	924
11.2.8.16	T_CONFIG	931
11.2.8.17	Paramètres communs des instructions.....	943
11.2.9	Communication avec une console de programmation.....	945
11.2.9.1	Etablissement de la liaison de communication matérielle	945
11.2.9.2	Configuration des appareils	946
11.2.9.3	Affectation d'adresses IP (Internet Protocol)	946
11.2.9.4	Test de votre réseau PROFINET	946
11.2.10	Communication IHM vers automate.....	947
11.2.10.1	Configuration des liaisons réseau logiques entre deux appareils	948
11.2.11	Communication API-API	949
11.2.11.1	Configuration des liaisons réseau logiques entre deux appareils	950
11.2.11.2	Configuration du routage local/partenaire entre deux appareils.....	950
11.2.11.3	Configuration des paramètres d'émission et de réception.....	951
11.2.12	Configuration d'une CPU et d'un périphérique PROFINET IO	954
11.2.12.1	Ajout d'un périphérique PROFINET IO	954
11.2.12.2	Affectation de CPU et de noms d'appareils	955
11.2.12.3	Affectation d'adresses IP (Internet Protocol)	956
11.2.12.4	Configuration du temps de cycle IO.....	956
11.2.13	Configuration d'une CPU et d'un I-device PROFINET.....	958
11.2.13.1	Fonctionnalité I-device	958
11.2.13.2	Propriétés et avantages d'un I-device.....	959
11.2.13.3	Caractéristiques d'un I-device.....	960
11.2.13.4	Echange de données entre système IO de niveau supérieur et système IO de niveau inférieur	962
11.2.13.5	Configurer un I-Device	965
11.2.14	Appareils partagés	968
11.2.14.1	Fonctionnalité appareil partagé.....	968

11.2.14.2	Exemple : Configurer un appareil partagé (configuration GSD).....	971
11.2.14.3	Exemple : Configurer un I-device comme appareil partagé	977
11.2.15	Protocole MRP (Media Redundancy Protocol).....	987
11.2.15.1	Redondance des supports dans les technologies en anneau.....	987
11.2.15.2	Utilisation du protocole MRP (Media Redundancy Protocol)	989
11.2.15.3	Configuration de la redondance des supports.....	993
11.2.16	Routage S7.....	996
11.2.16.1	Routage S7 entre interfaces CPU et CP	997
11.2.16.2	Routage S7 entre deux interfaces CP	997
11.2.17	Désactivation de SNMP.....	998
11.2.17.1	Désactivation de SNMP.....	999
11.2.18	Diagnostic	1001
11.2.19	Instructions de périphérie décentralisée	1001
11.2.20	Instructions de diagnostic	1001
11.2.21	Événements de diagnostic pour la périphérie décentralisée	1001
11.3	PROFIBUS.....	1002
11.3.1	Services de communication des CM PROFIBUS.....	1004
11.3.2	Référence aux manuels utilisateurs des CM PROFIBUS.....	1005
11.3.3	Configuration d'un maître et d'un esclave DP	1005
11.3.3.1	Ajout du module CM 1243-5 (maître DP) et d'un esclave DP	1005
11.3.3.2	Configuration des liaisons réseau logiques entre deux appareils PROFIBUS.....	1006
11.3.3.3	Affectation d'adresses PROFIBUS au module CM 1243-5 et à l'esclave DP	1006
11.3.4	Instructions de périphérie décentralisée	1008
11.3.5	Instructions de diagnostic	1008
11.3.6	Événements de diagnostic pour périphérie décentralisée.....	1008
11.4	Interface AS-i.....	1009
11.4.1	Configuration d'un maître et d'un esclave AS-i.....	1010
11.4.1.1	Ajout du module maître AS-i CM 1243-2 et d'un esclave AS-i.....	1010
11.4.1.2	Configuration des liaisons réseau logiques entre deux appareils AS-i	1011
11.4.1.3	Configuration des propriétés du maître AS-i CM 1243-2.....	1011
11.4.1.4	Affectation d'une adresse AS-i à un esclave AS-i	1012
11.4.2	Echange de données entre le programme utilisateur et les esclaves AS-i	1014
11.4.2.1	Configuration STEP 7 de base	1015
11.4.2.2	Configuration d'esclaves avec STEP 7.....	1016
11.4.3	Instructions de périphérie décentralisée	1018
11.4.4	Utilisation d'outils en ligne AS-i.....	1019
11.5	Communication S7.....	1021
11.5.1	GET et PUT (Lire les données d'une CPU distante/Écrire les données dans une CPU distante)	1021
11.5.2	Création d'une liaison S7	1026
11.5.3	Configuration du routage local/partenaire entre deux appareils.....	1027
11.5.4	Paramétrage de la liaison GET/PUT	1027
11.5.4.1	Paramètres de la liaison	1028
11.5.4.2	Configuration d'une liaison S7 CPU à CPU.....	1030
11.6	Que faire lorsqu'il est impossible d'accéder à la CPU par l'adresse IP.....	1036
12	Serveur Web.....	1037
12.1	Activation du serveur Web	1040
12.2	Configuration d'utilisateurs serveur Web	1042

12.3	Accès aux pages Web depuis un PC.....	1044
12.4	Accès aux pages Web depuis un dispositif mobile.....	1046
12.5	Utilisation d'un module CP pour accéder à des pages Web.....	1047
12.6	Pages Web standard	1048
12.6.1	Disposition des pages Web standard	1048
12.6.2	Pages de base	1049
12.6.3	Ouverture de session et privilèges utilisateur	1050
12.6.4	Introduction	1054
12.6.5	Page de démarrage	1054
12.6.6	Diagnostic	1056
12.6.7	Mémoire tampon de diagnostic.....	1059
12.6.8	Informations sur les modules	1060
12.6.9	Communication	1064
12.6.10	État des variables.....	1068
12.6.11	Tables de visualisation.....	1069
12.6.12	Sauvegarde en ligne	1072
12.6.13	Navigateur de fichiers	1075
12.7	Pages Web personnalisées	1079
12.7.1	Création de pages HTML.....	1080
12.7.2	Commandes AWP prises en charge par le serveur Web du S7-1200	1081
12.7.2.1	Lecture de variables.....	1082
12.7.2.2	Écriture de variables	1083
12.7.2.3	Lecture de variables spéciales.....	1085
12.7.2.4	Écriture de variables spéciales	1087
12.7.2.5	Utilisation d'un alias pour une référence de variable	1088
12.7.2.6	Définition de types Enum	1089
12.7.2.7	Référencement de variables CPU avec un type Enum	1090
12.7.2.8	Création de fragments	1091
12.7.2.9	Importation de fragments	1092
12.7.2.10	Combinaison de définitions.....	1093
12.7.2.11	Gestion des noms de variables contenant des caractères spéciaux.....	1094
12.7.3	Configuration de l'utilisation de pages Web personnalisées	1097
12.7.4	Configuration de la page d'entrée.....	1098
12.7.5	Programmation de l'instruction WWW pour les pages Web personnalisées.....	1099
12.7.6	Chargement des blocs de programme dans la CPU	1101
12.7.7	Accès aux pages Web personnalisées.....	1101
12.7.8	Contraintes spécifiques aux pages Web personnalisées	1102
12.7.9	Exemple de page Web personnalisée	1103
12.7.9.1	Page Web pour le contrôle-commande d'une éolienne.....	1103
12.7.9.2	Lecture et affichage des données de l'automate	1106
12.7.9.3	Utilisation d'un type Enum.....	1106
12.7.9.4	Écriture d'une entrée utilisateur dans l'automate.....	1108
12.7.9.5	Écriture d'une variable spéciale	1109
12.7.9.6	Référence : listage HTML de la page Web de contrôle d'éolienne à distance	1109
12.7.9.7	Configuration de l'exemple de page Web dans STEP 7.....	1114
12.7.10	Configuration de pages Web personnalisées dans plusieurs langues	1115
12.7.10.1	Création de la structure de dossiers	1115
12.7.10.2	Programmation du changement de langue.....	1115
12.7.10.3	Configuration de STEP 7 pour qu'il utilise une structure de page multilingue.....	1119
12.7.11	Commande avancée de pages Web personnalisées	1119

12.8	Contraintes.....	1124
12.8.1	Utilisation de JavaScript.....	1125
12.8.2	Restriction des fonctionnalités lorsque les options Internet n'autorisent pas les cookies ..	1125
12.8.3	Règles de saisie des noms de variable et des valeurs.....	1126
12.8.4	Importation du certificat de sécurité Siemens.....	1126
12.8.5	Importation de journaux en format CSV dans des versions non anglaises/américaines de Microsoft Excel.....	1128
13	Processeur de communication et Modbus TCP	1129
13.1	Utilisation des interfaces de communication série.....	1129
13.2	Polarisation et terminaison d'un connecteur de réseau RS485.....	1130
13.3	Communication point à point (PtP)	1132
13.3.1	PtP, communication Freeport	1132
13.3.2	Communication 3964(R)	1135
13.3.3	Configuration de la communication PtP Freeport.....	1137
13.3.3.1	Gestion du contrôle de flux	1139
13.3.3.2	Configuration des paramètres d'émission	1141
13.3.3.3	Configuration des paramètres de réception	1142
13.3.4	Configuration de la communication 3964(R)	1151
13.3.4.1	Configuration des ports de communication 3964(R)	1151
13.3.4.2	Configuration de la priorité et des paramètres du protocole 3964(R)	1153
13.3.5	Instructions point à point.....	1155
13.3.5.1	Paramètres communs pour les instructions point à point.....	1155
13.3.5.2	Port_Config (Configurer des paramètres de communication dynamiquement).....	1158
13.3.5.3	Send_Config (Configurer dynamiquement les paramètres de transmission série)	1162
13.3.5.4	Receive_Config (Configurer dynamiquement les paramètres de réception série).....	1164
13.3.5.5	P3964_Config (Configurer le protocole 3964(R))	1170
13.3.5.6	Send_P2P (Envoyer les données du tampon d'émission).....	1173
13.3.5.7	Receive_P2P (Activer la réception des messages).....	1177
13.3.5.8	Receive_Reset (Effacer le tampon de réception)	1179
13.3.5.9	Signal_Get (Lire les signaux RS-232).....	1180
13.3.5.10	Signal_Set (Régler les signaux RS-232)	1181
13.3.5.11	Get_Features	1183
13.3.5.12	Set_Features.....	1184
13.3.6	Programmation de la communication point à point	1186
13.3.6.1	Architecture d'interrogation	1187
13.3.7	Exemple : Communication point à point	1188
13.3.7.1	Configuration du module de communication	1189
13.3.7.2	Modes de fonctionnement RS422 et RS485	1192
13.3.7.3	Programmation du programme STEP 7	1196
13.3.7.4	Configuration de l'émulateur de terminal	1197
13.3.7.5	Exécution de l'exemple	1198
13.4	Communication USS (interface série universelle)	1199
13.4.1	Sélection de la version des instructions USS	1202
13.4.2	Conditions requises pour l'utilisation du protocole USS	1203
13.4.3	Opérations USS	1206
13.4.3.1	USS_Port_Scan (Éditer la communication via le réseau USS)	1206
13.4.3.2	USS_Drive_Control (Échanger des données avec l'entraînement).....	1208
13.4.3.3	USS_Read_Param (Lire des paramètres de l'entraînement)	1211
13.4.3.4	USS_Write_Param (Modifier des paramètres dans l'entraînement).....	1213
13.4.4	Codes d'état USS	1215

13.4.5	Exigences générales pour la configuration d'un entraînement USS	1217
13.4.6	Exemple : Configuration et connexion générales à un entraînement USS	1218
13.5	Communication Modbus	1221
13.5.1	Vue d'ensemble de la communication Modbus RTU et Modbus TCP	1221
13.5.2	Modbus TCP	1224
13.5.2.1	Vue d'ensemble	1224
13.5.2.2	Sélection de la version des instructions Modbus TCP	1224
13.5.2.3	Instructions Modbus TCP	1225
13.5.2.4	Exemples Modbus TCP	1246
13.5.3	Modbus RTU	1251
13.5.3.1	Vue d'ensemble	1251
13.5.3.2	Sélection de la version des instructions Modbus RTU	1253
13.5.3.3	Nombre maximum d'esclaves Modbus pris en charge	1254
13.5.3.4	Instructions Modbus RTU	1254
13.5.3.5	Exemples Modbus RTU	1277
13.6	Communication PTP d'héritage (CM/CB 1241 uniquement)	1281
13.6.1	Instructions d'héritage point à point	1282
13.6.1.1	PORT_CFG (Configurer des paramètres de communication dynamiquement)	1282
13.6.1.2	SEND_CFG (Configurer dynamiquement les paramètres de transmission série)	1284
13.6.1.3	RCV_CFG (Configurer dynamiquement les paramètres de réception série)	1286
13.6.1.4	SEND_PTP (Envoyer les données du tampon d'émission)	1291
13.6.1.5	RCV_PTP (Valider la réception des messages)	1293
13.6.1.6	RCV_RST (Effacer le tampon de réception)	1296
13.6.1.7	SGN_GET (Interroger les signaux RS-232)	1297
13.6.1.8	SGN_SET (Régler les signaux RS-232)	1298
13.7	Communication d'héritage USS (CM/CB 1241 uniquement)	1300
13.7.1	Sélection de la version des instructions USS	1301
13.7.2	Conditions requises pour l'utilisation du protocole USS	1302
13.7.3	Instructions d'héritage USS	1305
13.7.3.1	Instruction USS_PORT (Editer communication via réseau USS)	1305
13.7.3.2	Instruction USS_DRV (Données permutées avec l'entraînement)	1306
13.7.3.3	Instruction USS_RPM (Lire des paramètres de l'entraînement)	1309
13.7.3.4	Instruction USS_WPM (Modifier les paramètres dans l'entraînement)	1311
13.7.4	Codes d'état USS d'héritage	1313
13.7.5	Exigences générales pour la configuration d'un entraînement USS d'héritage	1315
13.8	Communication d'héritage Modbus TCP	1316
13.8.1	Vue d'ensemble	1316
13.8.2	Sélection de la version des instructions Modbus TCP	1316
13.8.3	Instructions d'héritage Modbus TCP	1317
13.8.3.1	MB_CLIENT (Communiquer comme client Modbus TCP via PROFINET)	1317
13.8.3.2	MB_SERVER (Communiquer comme serveur Modbus TCP via PROFINET)	1325
13.8.4	Exemples d'héritage Modbus TCP	1331
13.8.4.1	Exemple : Liaisons TCP multiples MB_SERVER d'héritage	1331
13.8.4.2	Exemple : MB_CLIENT 1 d'héritage : Plusieurs demandes avec une liaison TCP commune	1332
13.8.4.3	Exemple : MB_CLIENT 2 d'héritage : Plusieurs demandes avec des liaisons TCP différentes	1333
13.8.4.4	Exemple : MB_CLIENT 3 d'héritage : Demande d'écriture dans la mémoire image des sorties	1334
13.8.4.5	Exemple : MB_CLIENT 4 d'héritage : Coordination de plusieurs demandes	1334

13.9	Communication d'héritage Modbus RTU (CM/CB 1241 uniquement).....	1336
13.9.1	Vue d'ensemble	1336
13.9.2	Sélection de la version des instructions Modbus RTU	1336
13.9.3	Instructions d'héritage Modbus RTU	1337
13.9.3.1	MB_COMM_LOAD (Configurer le port sur le module PtP pour Modbus RTU)	1337
13.9.3.2	MB_MASTER (Communiquer via le port PtP en tant que maître Modbus RTU)	1341
13.9.3.3	MB_SLAVE (Communiquer via le port PtP en tant qu'esclave Modbus RTU)	1347
13.9.4	Exemples d'instructions Modbus RTU héritées	1354
13.9.4.1	Exemple : Programme maître Modbus RTU d'héritage.....	1354
13.9.4.2	Exemple : Programme esclave Modbus RTU d'héritage.....	1356
13.10	Industrial Remote Communication (IRC)	1357
13.10.1	Présentation de CP de commande à distance	1357
13.10.2	Connexion à un réseau GSM	1360
13.10.3	Applications du CP 1242-7	1362
13.10.4	Autres caractéristiques du CP 1242-7	1363
13.10.5	Informations supplémentaires.....	1363
13.10.6	Accessoires.....	1364
13.10.7	Exemples de configuration pour telecontrol.....	1365
14	Communication TeleService (courrier électronique SMTP).....	1371
14.1	Instruction TM_MAIL (Envoyer e-mail)	1371
15	Outils en ligne et de diagnostic.....	1379
15.1	DEL d'état	1379
15.2	Passage en ligne et connexion à une CPU	1383
15.3	Affectation d'un nom à un périphérique PROFINET IO en ligne	1384
15.4	Réglage de l'adresse IP et de l'heure	1386
15.5	Restauration des réglages d'usine.....	1387
15.6	Mise à jour du firmware.....	1389
15.7	Formatage d'une carte mémoire SIMATIC à partir de STEP 7	1391
15.8	Panneau de commande CPU de la CPU en ligne.....	1392
15.9	Surveillance du temps de cycle et de l'utilisation de la mémoire.....	1392
15.10	Affichage des événements de diagnostic dans la CPU.....	1393
15.11	Comparaison de CPU hors ligne et en ligne	1394
15.12	Exécution d'une comparaison hors ligne/en ligne de la topologie.....	1395
15.13	Visualisation et forçage de valeurs dans la CPU.....	1396
15.13.1	Passage en ligne pour visualiser les valeurs dans la CPU	1397
15.13.2	Affichage de l'état dans l'éditeur de programme	1398
15.13.3	Acquisition d'un instantané des valeurs en ligne d'un DB en vue de la restauration	1398
15.13.4	Utilisation d'une table de visualisation pour visualiser et forcer des valeurs dans la CPU.....	1400
15.13.4.1	Utilisation d'un déclenchement lors de la visualisation ou du forçage de variables API	1401
15.13.4.2	Déblocage des sorties à l'état ARRÊT	1402
15.13.5	Forçage permanent de valeurs dans la CPU	1403
15.13.5.1	Utilisation de la table de forçage permanent	1403
15.13.5.2	Fonctionnement de la fonction de forçage permanent	1404

15.14	Chargement dans la CPU à l'état MARCHE	1406
15.14.1	Conditions requises pour un chargement dans la CPU à l'état MARCHE	1407
15.14.2	Modification du programme à l'état "Marche"	1408
15.14.3	Chargement de blocs sélectionnés	1409
15.14.4	Chargement d'un bloc individuel sélectionné alors qu'un autre bloc présente une erreur de compilation	1411
15.14.5	Forcer et télécharger des blocs existants à l'état MARCHE	1412
15.14.6	Réaction du système en cas d'échec de l'opération de chargement.....	1415
15.14.7	Considérations sur le chargement dans la CPU à l'état MARCHE.....	1415
15.15	Traçage et enregistrement de données CPU en fonctions de conditions de déclenchement.....	1418
15.16	Détermination du type de condition de rupture de fil sur un module SM 1231.....	1420
15.17	Sauvegarde et restauration d'une CPU	1423
15.17.1	Options de sauvegarde et de restauration.....	1423
15.17.2	Sauvegarde d'une CPU en ligne.....	1425
15.17.3	Restauration d'une CPU	1427
A	Caractéristiques techniques	1429
A.1	Site Web Industry Online Support de Siemens	1429
A.2	Caractéristiques techniques d'ordre général	1429
A.3	Brochages de port X1 d'interface PROFINET	1440
A.4	CPU 1211C.....	1442
A.4.1	Caractéristiques et fonctions générales.....	1442
A.4.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1211C.....	1445
A.4.3	Entrées et sorties TOR	1449
A.4.4	Entrées analogiques	1450
A.4.4.1	Réponse indicielle des entrées analogiques intégrées de la CPU.....	1451
A.4.4.2	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU	1451
A.4.4.3	Plages de mesure des entrées analogiques pour la tension (CPU).....	1451
A.4.5	Schémas de câblage de la CPU 1211C	1452
A.5	CPU 1212C.....	1456
A.5.1	Caractéristiques et fonctions générales.....	1456
A.5.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1212C.....	1459
A.5.3	Entrées et sorties TOR	1463
A.5.4	Entrées analogiques	1465
A.5.4.1	Réponse indicielle des entrées analogiques intégrées de la CPU.....	1465
A.5.4.2	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU	1466
A.5.4.3	Plages de mesure des entrées analogiques pour la tension (CPU).....	1466
A.5.5	Schémas de câblage de la CPU 1212C	1467
A.6	CPU 1214C.....	1471
A.6.1	Caractéristiques et fonctions générales.....	1471
A.6.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1214C.....	1473
A.6.3	Entrées et sorties TOR	1477
A.6.4	Entrées analogiques	1479
A.6.4.1	Réponse indicielle des entrées analogiques intégrées de la CPU.....	1479
A.6.4.2	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU	1480
A.6.4.3	Plages de mesure des entrées analogiques pour la tension (CPU).....	1480
A.6.5	Schémas de câblage de la CPU 1214C	1481

A.7	CPU 1215C	1487
A.7.1	Caractéristiques et fonctions générales.....	1487
A.7.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1215C.....	1489
A.7.3	Entrées et sorties TOR	1493
A.7.4	Entrées et sorties analogiques	1495
A.7.4.1	Réponse indicielle des entrées analogiques intégrées de la CPU	1495
A.7.4.2	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU	1496
A.7.4.3	Plages de mesure des entrées analogiques pour la tension (CPU).....	1496
A.7.4.4	Caractéristiques des sorties analogiques	1497
A.7.5	Schémas de câblage de la CPU 1215C	1498
A.8	CPU 1217C	1504
A.8.1	Caractéristiques et fonctions générales.....	1504
A.8.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1217C.....	1506
A.8.3	Entrées et sorties TOR	1510
A.8.4	Entrées et sorties analogiques	1514
A.8.4.1	Caractéristiques des entrées analogiques.....	1514
A.8.4.2	Réponse indicielle des entrées analogiques intégrées de la CPU	1514
A.8.4.3	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU	1515
A.8.4.4	Plages de mesure des entrées analogiques pour la tension (CPU).....	1515
A.8.4.5	Caractéristiques des sorties analogiques	1516
A.8.5	Schémas de câblage de la CPU 1217C	1517
A.8.6	Détails de l'entrée différentielle (DI) et exemple d'application de la CPU 1217C.....	1519
A.8.7	Détails de l'entrée différentielle (DI) et exemple d'application de la CPU 1217C.....	1520
A.9	Modules d'entrées-sorties TOR (SM)	1521
A.9.1	Caractéristiques des modules d'entrées TOR SM 1221	1521
A.9.2	Caractéristiques des modules 8 sorties TOR SM 1222.....	1523
A.9.3	Caractéristiques des modules 16 sorties TOR SM 1222.....	1525
A.9.4	Caractéristiques des modules SM 1223 DI/DQ V DC	1531
A.9.5	Caractéristiques des modules SM 1223 DI/DQ V AC	1538
A.10	Modules d'entrées-sorties analogiques (SM)	1541
A.10.1	Caractéristiques des modules d'entrées analogiques SM 1231.....	1541
A.10.2	Caractéristiques des modules de sorties analogiques SM 1232.....	1547
A.10.3	Caractéristiques du module d'entrées/sorties analogiques SM 1234.....	1550
A.10.4	Réponse indicielle des entrées analogiques	1553
A.10.5	Temps d'échantillonnage et temps d'actualisation pour les entrées analogiques.....	1554
A.10.6	Plages de mesures d'entrées analogiques pour tension et courant (SB et SM)	1554
A.10.7	Plages de mesures de sorties analogiques pour tension et courant (SB et SM)	1556
A.11	Modules d'entrées-sorties (SM) Thermocouple et RTD	1558
A.11.1	SM 1231 Thermocouple.....	1558
A.11.1.1	Principe de fonctionnement des thermocouples.....	1562
A.11.1.2	Tableaux de sélection pour les SM 1231 Thermocouple	1563
A.11.2	SM 1231 RTD	1566
A.11.2.1	Tableaux de sélection pour les SM 1231 RTD	1570
A.12	Modules technologiques	1573
A.12.1	SM 1278 4 x IO-Link maître	1573
A.12.1.1	Présentation du SM 1278 4 x IO-Link maître	1576
A.12.1.2	Raccordement.....	1579
A.12.1.3	Paramètres/espace d'adressage	1581
A.12.1.4	Alarmes, erreurs et alarmes système	1584

A.13	Signal Boards (SB) TOR.....	1587
A.13.1	Caractéristiques des SB 1221 entrées TOR, 200 kHz	1587
A.13.2	Caractéristiques des SB 1222 sorties TOR, 200 kHz.....	1590
A.13.3	Caractéristiques des SB 1223 entrées/sorties TOR, 200 kHz.....	1593
A.13.4	Caractéristiques du SB 1223 entrées 2 X 24 V DC/sorties 2 X 24 V DC	1598
A.14	Signal Boards (SB) analogiques	1601
A.14.1	Caractéristiques du SB 1231 1 sortie analogique.....	1601
A.14.2	Caractéristiques du SB 1232 1 sortie analogique.....	1604
A.14.3	Plages de mesure pour les entrées et sorties analogiques	1606
A.14.3.1	Réponse indicielle des entrées analogiques	1606
A.14.3.2	Temps d'échantillonnage et temps d'actualisation pour les entrées analogiques	1606
A.14.3.3	Plages de mesures d'entrées analogiques pour tension et courant (SB et SM)	1607
A.14.3.4	Plages de mesures de sorties analogiques pour tension et courant (SB et SM)	1608
A.14.4	Signal Boards (SB) Thermocouple	1610
A.14.4.1	Caractéristiques du SB 1231 1 entrée analogique Thermocouple	1610
A.14.4.2	Principe de fonctionnement des thermocouples	1611
A.14.5	Signal Boards (SB) RTD	1615
A.14.5.1	Caractéristiques du SB 1231 1 entrée analogique RTD.....	1615
A.14.5.2	Tableaux de sélection pour le SB 1231 RTD.....	1618
A.15	Battery Board BB 1297	1620
A.16	Interfaces de communication	1622
A.16.1	PROFIBUS.....	1622
A.16.1.1	ESCLAVE PROFIBUS DP CM 1242-5	1622
A.16.1.2	Brochage de la prise sub D du CM 1242-5.....	1623
A.16.1.3	Maître PROFIBUS DP CM 1243-5.....	1624
A.16.1.4	Brochage de la prise subD du CM 1243-5.....	1625
A.16.2	CP 1242-7	1626
A.16.2.1	CP 1242-7 GPRS.....	1626
A.16.2.2	Antenne GSM/GPRS ANT794-4MR	1628
A.16.2.3	Antenne en nappe ANT794-3M	1629
A.16.3	CM 1243-2 maître AS-i	1630
A.16.3.1	Données techniques pour l'interface maître AS-i CM 1243-2.....	1630
A.16.3.2	Raccordements électriques au maître AS-i	1631
A.16.4	RS232, RS422 et RS485	1633
A.16.4.1	Caractéristiques CB 1241 RS485.....	1633
A.16.4.2	Caractéristiques du CM 1241 RS232	1635
A.16.4.3	Caractéristiques du module CM 1241 RS422/485	1636
A.17	TeleService (TS Adapter et TS Adapter modulaire)	1638
A.18	Cartes mémoire SIMATIC	1638
A.19	Simulateurs d'entrées	1639
A.20	Module de potentiomètre S7-1200	1641
A.21	Câble d'extension d'E/S	1642
A.22	Produits connexes	1643
A.22.1	Module d'alimentation PM 1207.....	1643
A.22.2	Module commutateur compact CSM 1277	1643
A.22.3	Module CM CANopen	1644
A.22.4	Module de communication RF120C.....	1644
A.22.5	Module SM 1238 Energy Meter	1645

A.22.6	Électronique de pesage SIWAREX	1645
B	Calcul d'un bilan de consommation	1647
C	Numéro de référence	1651
C.1	Modules CPU	1651
C.2	Module d'entrées-sorties (SM), Signal Boards (SB) et Battery boards (BB)	1652
C.3	Communication	1653
C.4	Modules d'entrées-sorties et CPU de sécurité	1655
C.5	Autres modules	1655
C.6	Cartes mémoire	1656
C.7	Appareils IHM Basic.....	1656
C.8	Pièces détachées et autres matériels.....	1657
C.9	Logiciel de programmation	1664
D	Remplacement de l'appareil et compatibilité des pièces de rechange	1665
D.1	Remplacement d'une CPU V3.0 par une CPU V4.2.x.....	1665
D.2	Kits de pièces de rechange du bornier pour S7-1200 V3.0 et antérieure	1672
	Index.....	1675

Présentation du produit

1.1 Introduction à l'automate S7-1200

Le contrôleur S7-1200 offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils afin de répondre à vos besoins en matière d'automatisation. Sa structure compacte, sa configuration souple et son important jeu d'instructions en font une solution idéale pour la commande d'applications très variées.

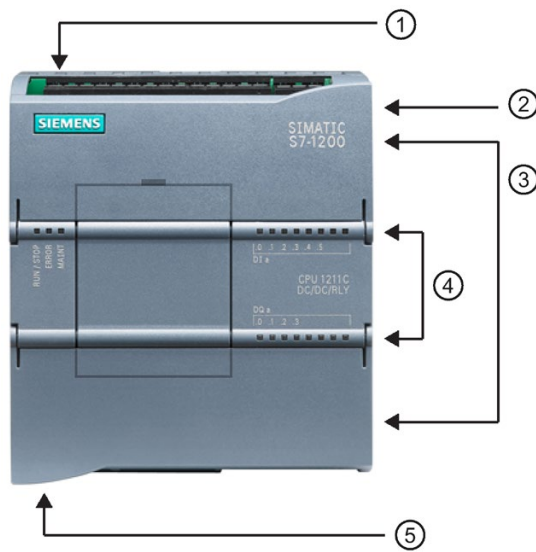
La CPU combine, entre autres, les éléments suivants dans un boîtier compact afin de créer un automate puissant :

- un microprocesseur
- une alimentation intégrée
- des circuits d'entrée et de sortie
- PROFINET intégré
- des E/S Motion Control rapides

Une fois que vous avez chargé votre programme, la CPU contient la logique nécessaire au contrôle et à la commande des appareils dans votre application. La CPU surveille les entrées et modifie les sorties conformément à la logique de votre programme utilisateur qui peut contenir des instructions booléennes, des instructions de comptage, des instructions de temporisation, des instructions mathématiques complexes, des instructions Motion Control, ainsi que des commandes pour communiquer avec d'autres appareils intelligents.

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer par le biais de réseaux et de protocoles tels que :

- PROFIBUS
- GPRS
- LTE
- WAN
- RS485
- RS232
- RS422
- IEC
- DNP3
- USS
- MODBUS



- ① Connecteur d'alimentation
- ② Logement pour carte mémoire sous le volet supérieur
- ③ Connecteurs amovibles pour le câblage utilisateur (derrière les volets)
- ④ DEL d'état pour les E/S intégrées
- ⑤ Connecteur PROFINET (sur la face inférieure de la CPU)

Plusieurs fonctions de sécurité vous aident à protéger l'accès à la CPU et au programme de commande :

- Chaque CPU fournit une protection par mot de passe (Page 218) qui vous permet de configurer l'accès aux fonctions CPU.
- Vous pouvez utiliser la protection du savoir-faire (Page 222) (protection "know-how") pour masquer le code d'un bloc spécifique.
- Vous pouvez utiliser la protection contre la copie (Page 223) pour lier votre programme à une carte mémoire ou une CPU spécifique.

Tableau 1- 1 Comparaison des modèles de CPU

Caractéristique		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Dimensions (mm)		90 x 100 x 75		110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
Mémoire utilisateur	de travail	50 Ko	75 Ko	100 Ko	125 Ko	150 Ko
	de charge-ment	1 Mo	2 Mo	4 Mo		
	rémanente	10 Ko				
E/S intégrées lo-cales	TOR	6 entrées/ 4 sorties	8 entrées/ 6 sorties	14 entrées/ 10 sorties		
	Analo-giques	2 entrées			2 entrées/2 sorties	
Taille de la mé-moire image	Entrées (I)	1024 octets				
	Sorties (Q)	1024 octets				
Mémentos (M)		4096 octets		8192 octets		
Modules d'entrées-sorties (SM) pour extension		Aucun	2	8		
Signal Board (SB), Battery Board (BB) ou Communication Board (CB)		1				

Caractéristique		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Module de communication (CM) (extension vers la gauche)		3				
Compteurs rapides	Total	Jusqu'à 6 configurés pour utiliser des entrées intégrées ou SB				
	1 MHz	-				lb.2 à lb.5
	100/180 kHz	la.0 à la.5				
	30/120 kHz	--	la.6 à la.7	la.6 à lb.5		la.6 à lb.1
	200 kHz ³					
Sorties d'impulsions ²	Total	Jusqu'à 4 configurées pour utiliser des sorties intégrées ou SB				
	1 MHz	--				Qa.0 à Qa.3
	100 kHz	Qa.0 à Qa.3				Qa.4 à Qb.1
	20 kHz	--	Qa.4 à Qa.5	Qa.4 à Qb.1		--
Carte mémoire		Carte mémoire SIMATIC (facultative)				
Journaux de données	Nombre	Maximum 8 ouverts simultanément				
	Taille	500 Mo par journal ou telle que limitée par la mémoire de chargement disponible maximum				
Durée de conservation de l'horloge temps réel		20 jours typ./12 jours min. à 40 °C (supercondensateur sans maintenance)				
Port de communication Ethernet PROFINET		1			2	
Vitesse d'exécution des instructions mathématiques sur réels		2,3 µs/instruction				
Vitesse d'exécution des instructions booléennes		0,08 µs/instruction				

- ¹ La vitesse plus lente s'applique lorsque le HSC est configuré pour fonctionner en quadrature de phase.
- ² Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.
- ³ Jusqu'à 200 kHz sont disponibles avec le SB 1221 DI x 24 V DC 200 kHz et le SB 1221 DI 4 x 5 V DC 200 kHz.

Les différents modèles de CPU disposent de fonctions et éléments divers qui vous aident à créer des solutions efficaces pour vos diverses applications. Reportez-vous aux caractéristiques techniques (Page 1429) pour des informations détaillées sur une CPU spécifique.

Tableau 1- 2 Blocs, temporisations et compteurs pris en charge par le S7-1200

Élément		Description					
Blocs	Type	OB, FB, FC, DB					
	Taille	Modèle de CPU	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
		Blocs de code	50 ko	64 ko	64 ko	64 ko	64 ko
		Blocs de données liés ¹	50 ko	75 ko	100 ko	125 ko	150 ko
	Blocs de données non liés ²	256 ko	256 ko	256 ko	256 ko	256 ko	
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)					
	Profondeur d'imbrication	16 en cas d'appel depuis l'OB de cycle de programme ou de démarrage ; 6 en cas d'appel depuis un OB d'alarme associé à un événement ³					
Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.						
OB	Cycle de programme	Multiple					
	Mise en route	Multiple					
	Alarmes temporisées	4 (1 par événement)					
	Alarmes cycliques	4 (1 par événement)					
	Alarmes de processus	50 (1 par événement)					
	Alarmes d'erreur de temps	1					
	Alarmes de diagnostic	1					
	Débrochage ou enfichage de modules	1					
	Défaillance du châssis ou de la station	1					
	Heure	Multiple					
	Etat	1					
	Mettre à jour	1					
	Profil	1					
Temporisations	Type	CEI					
	Quantité	Limité uniquement par la taille de la mémoire					
	Stockage	Structure dans un DB, 16 octets par temporisation					
Compteurs	Type	CEI					
	Quantité	Limité uniquement par la taille de la mémoire					
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> • SInt, USInt : 3 octets • Int, UInt : 6 octets • DInt, UDInt : 12 octets 					

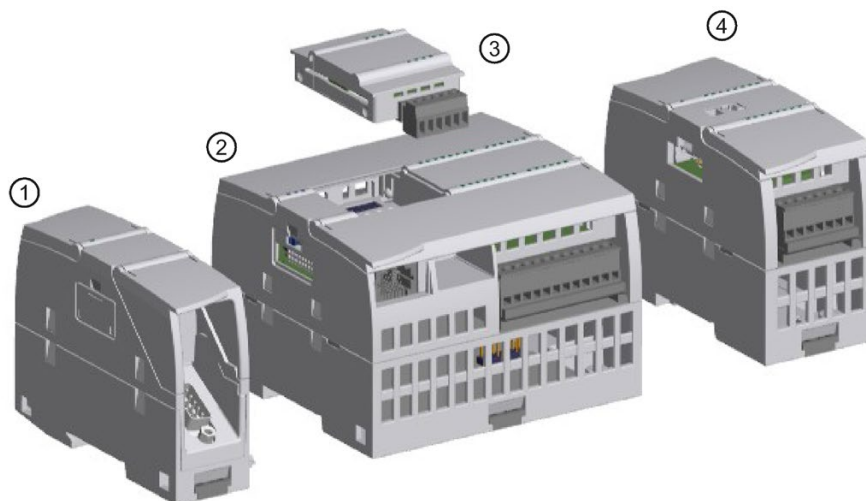
¹ Stockés dans la mémoire de travail et la mémoire de chargement

² Stockés uniquement dans la mémoire de chargement

³ Les programmes de sécurité utilisent deux niveaux d'imbrication. Le programme utilisateur a donc une profondeur d'imbrication de quatre dans les programmes de sécurité.

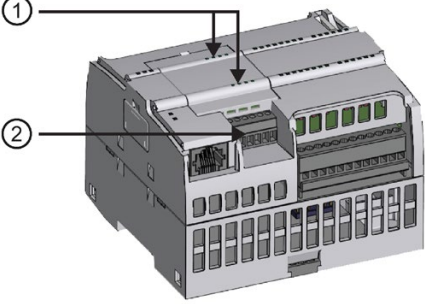
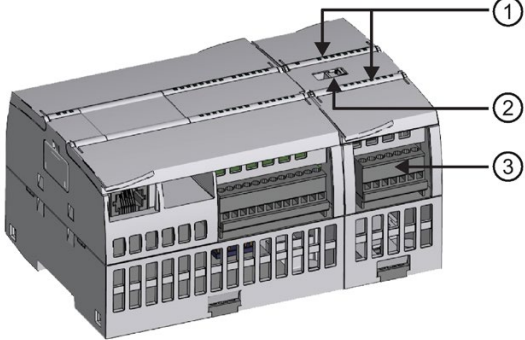
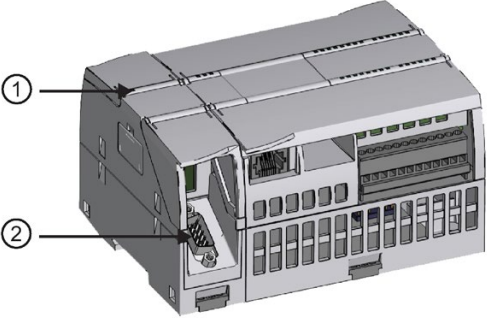
1.2 Possibilités d'extension de la CPU

La gamme S7-1200 offre divers modules et cartes enfichables pour accroître les capacités de la CPU avec des E/S supplémentaires ou d'autres protocoles de communication. Référez-vous aux caractéristiques techniques (Page 1429) pour des informations détaillées sur un module spécifique.



- ① Module de communication (CM) ou processeur de communication (CP) (Page 1622)
- ② CPU (CPU 1211C (Page 1442), CPU 1212C (Page 1456), CPU 1214C (Page 1471), CPU 1215C (Page 1487), CPU 1217C (Page 1504))
- ③ Signal board (SB) (SB TOR (Page 1587), SB analogique (Page 1601)), communication board (CB) (Page 1633) ou Battery Board (BB)CPU (CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C, CPU 1217C) (Page 1620)
- ④ Module d'entrées-sorties (SM) (SM TOR (Page 1521), SM analogique (Page 1541), SM thermocouple (Page 1558), SM RTD (Page 1566), SM technologique) (Page 1573)

Tableau 1- 3 Modules d'extension S7-1200

Type de module	Description
<p>La CPU prend en charge une carte d'extension enfichable :</p> <ul style="list-style-type: none"> • Un Signal Board (SB) fournit des E/S supplémentaires pour votre CPU. Le SB se raccorde à l'avant de la CPU. • Un Communication Board (CB) vous permet d'ajouter un autre port de communication à votre CPU. • Un Battery Board (BB) permet une sauvegarde à long terme de l'horloge temps réel. 	 <p>① DEL d'état sur le Signal Board</p> <p>② Connecteur amovible pour le câblage utilisateur</p>
<p>Les modules d'entrées-sorties (SM) permettent d'ajouter des fonctionnalités à la CPU. Les SM se raccordent sur le côté droit de la CPU.</p> <ul style="list-style-type: none"> • E/S TOR • E/S analogiques • RTD et Thermocouple • SM 1278 IO-Link maître • SM 1238 Energy Meter <p>(https://support.industry.siemens.com/cs/ww/en/view/109483435)</p>	 <p>① DEL d'état</p> <p>② Languette coulissante du connecteur de bus</p> <p>③ Connecteur amovible pour le câblage utilisateur</p>
<p>Les modules de communication (CM) et les processeurs de communication (CP) ajoutent des options de communication à la CPU, telles que la connectivité PROFIBUS ou RS232/RS485 (pour PtP, Modbus ou USS) ou le maître AS-i.</p> <p>Un CP offre la possibilité d'autres types de communication, par exemple la connexion de la CPU par un réseau GPRS, LTE, IEC, DNP3 ou WDC.</p> <ul style="list-style-type: none"> • La CPU accepte jusqu'à trois CM ou CP. • Chaque CM ou CP se raccorde sur le côté gauche de la CPU (ou sur le côté gauche d'un autre CM ou CP). 	 <p>① DEL d'état</p> <p>② Connecteur de communication</p>

1.3 Tableaux IHM basiques

Les pupitres SIMATIC HMI Basic Panels fournissent des appareils à écran tactile pour des tâches de base de commande et de surveillance d'opérateur. Tous les tableaux ont un indice de protection de IP65 et sont certifiés CE, UL, cULus et NEMA 4x.

Les pupitres HMI Basic Panels (Page 1656) disponibles sont décrits ci-dessous :

- KTP400 Basic : écran tactile 4" avec 4 touches configurables, une résolution de 480 x 272 et 800 variables
- KTP700 Basic : écran tactile 7" avec 8 touches configurables, une résolution de 800 x 480 et 800 variables
- KTP700 Basic DP : écran tactile 7" avec 8 touches configurables, une résolution de 800 x 480 et 800 variables
- KTP900 Basic : écran tactile 9" avec 8 touches configurables, une résolution de 800 x 480 et 800 variables
- KTP1200 Basic : écran tactile 12" avec 10 touches configurables, une résolution de 800 x 480 et 800 variables
- KTP 1200 Basic DP : écran tactile 12" avec 10 touches configurables, une résolution de 800 x 400 et 800 variables

Nouvelles fonctions

La version V4.2.3 présente les nouveautés suivantes :

- Amélioration des performances de certaines instructions de bibliothèque
- Lecture et écriture dans des tableaux multidimensionnels à partir de pages Web personnalisées
- Corrections apportées à ce manuel depuis la version précédente

Nouvelles fonctionnalités Modbus

Les instructions Modbus TCP "MB_SERVER" et Modbus RTU "Modbus_Slave" permettent désormais d'exécuter les opérations suivantes :

- Limiter les octets de sortie dans lesquels vous pouvez écrire à l'aide des variables "QB_Start" et "QB_Count"
- Limiter les octets de sortie dans lesquels vous pouvez lire à l'aide des variables "QB_Read_Start" et "QB_Read_Count"
- Limiter les octets d'entrée dans lesquels vous pouvez lire à l'aide des variables "IB_Read_Start" et "IB_Read_Count"

Les instructions Modbus TCP "MB_SERVER" et Modbus RTU "Modbus_Slave" permettent désormais d'exécuter l'opération suivante :

- À partir de la version V5.0 de l'instruction MB_SERVER ou V4.0 de l'instruction Modbus_Slave et de la version V4.2 du firmware de la CPU S7-1200, vous pouvez accéder à des zones de données dans les DB au lieu d'accéder directement aux mémoires images et aux registres de maintien.

L'instruction Modbus TCP "MB_SERVER" permet désormais d'exécuter les opérations suivantes :

- Actualiser le paramètre NDR (New Data Ready, nouvelles données prêtes) dans le même appel qui traite une demande d'écriture Modbus TCP
- Actualiser le paramètre DR (Data Read, données lues) dans le même appel qui traite une demande d'écriture Modbus TCP

Remplacement de votre CPU V3.0 par une CPU V4.x.x

Si vous remplacez une CPU S7-1200 V3.0 par une CPU S7-1200 V4.x.x, prenez connaissance des différences (Page 1665) documentées entre ces versions ainsi que des actions utilisateur requises.

Voir aussi

Manuel de sécurité fonctionnelle du S7-1200
(<https://support.industry.siemens.com/cs/ww/fr/view/104547552/en>)

Logiciel de programmation STEP 7

STEP 7 fournit un environnement convivial pour concevoir, éditer et surveiller la logique nécessaire à la commande de votre application, et notamment les outils pour gérer et configurer tous les appareils dans votre projet, tels que des automates et appareils IHM. Pour vous aider à trouver les informations dont vous avez besoin, STEP 7 fournit un système d'aide en ligne complet.

STEP 7 comprend des langages de programmation standard, ce qui s'avère très pratique et efficace pour la mise au point du programme de commande de votre application.

- CONT (schéma à contacts) (Page 206) est un langage de programmation graphique. Sa représentation se base sur des schémas de circuits.
- LOG (logigramme) (Page 207) est un langage de programmation se fondant sur les symboles logiques graphiques utilisés en algèbre booléenne.
- SCL (Structured Control Language) (Page 208) est un langage de programmation littéral évolué.

Lorsque vous créez un bloc de code, vous sélectionnez le langage de programmation à utiliser par ce bloc. Votre programme utilisateur peut utiliser des blocs de code créés dans n'importe lequel des langages de programmation disponibles.

Remarque

STEP 7 est le composant logiciel de programmation et de configuration du portail TIA. Outre STEP 7, le portail TIA comprend le système WinCC permettant de concevoir et d'exécuter la visualisation runtime du process, et il contient l'aide en ligne pour WinCC et STEP 7.

3.1 Besoins du système

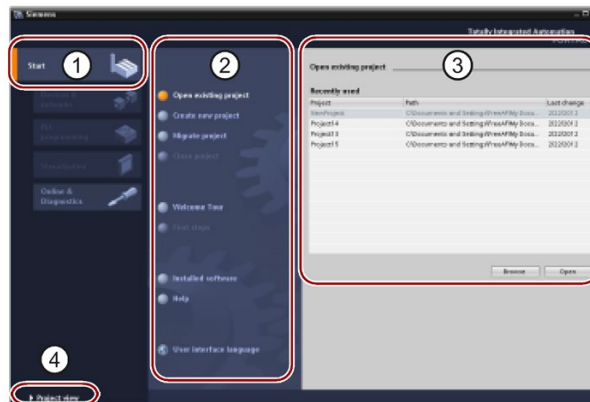
Vous devez installer STEP 7 avec les privilèges Administrateur.

Tableau 3- 1 Besoins du système

Matériel / logiciel	Conditions requises
Type de processeur	Intel® Core™ i5-3320M 3,3 GHz ou version ultérieure
Mémoire vive	8 Go
Espace disponible sur le disque dur	2 Go sur le lecteur système C:\
Systèmes d'exploitation	<p>Vous pouvez utiliser STEP 7 avec les systèmes d'exploitation suivants (64 bit uniquement) :</p> <ul style="list-style-type: none"> • Microsoft Windows 7 Home Premium SP1 ou version ultérieure (STEP 7 Basic uniquement, non supporté pour STEP 7 Professional) • Microsoft Windows 7 ou version ultérieure (Professional SP1, Enterprise SP1, Ultimate SP1) • Microsoft Windows 8.1 (STEP 7 Basic uniquement, non supporté pour STEP 7 Professional) • Microsoft Windows 8.1 (Professional, Enterprise) • Microsoft Server 2008 R2 StdE SP1 (STEP 7 Professional uniquement) • Microsoft Server 2012 R2 StdE
Carte graphique	32 Mo RAM Profondeur de couleur 24 bits
Résolution de l'écran	1920 x 1080 (recommandé)
Réseau	Ethernet 10 Mbit/s ou plus, pour la communication entre STEP 7 et la CPU
Lecteur optique	DVD-ROM

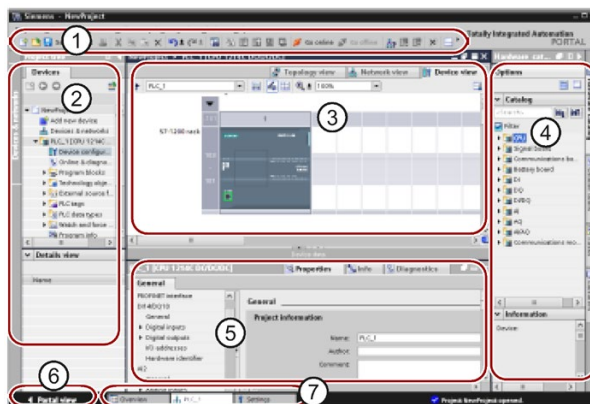
3.2 Différentes vues pour faciliter votre travail

STEP 7 offre un environnement convivial pour développer la logique du contrôleur, configurer la visualisation IHM et établir la communication réseau. Pour permettre d'augmenter votre productivité, STEP 7 offre deux vues différentes du projet : un ensemble orienté tâche de portails qui sont organisés selon la fonctionnalité des outils (vue du portail) et une vue orientée projet des éléments dans le projet (vue du projet). Choisissez la vue qui permet un travail le plus efficace possible. Avec un simple clic, vous pouvez faire le va-et-vient entre la vue du portail et la vue du projet.



Vue du portail

- ① Portails des différentes tâches
- ② Tâches du portail sélectionné
- ③ Panneau de sélection de l'action
- ④ Bascule dans la vue du projet



Vue du projet

- ① Menus et barre d'outils
- ② Navigateur du projet
- ③ Zone de travail
- ④ Task Cards
- ⑤ Fenêtre d'inspection
- ⑥ Bascule dans la vue du portail
- ⑦ Barre d'édition

Comme tous ces composants sont regroupés à un endroit, vous pouvez facilement accéder à chaque élément de votre projet. La zone de travail est composée de trois vues dans trois onglets différents :

- Vue de l'appareil : Affiche l'appareil que vous avez ajouté ou sélectionné et les modules associés.
- Vue de réseau : Affiche la CPU et les liaisons dans votre réseau.
- Vue topologique : Affiche la topologie PROFINET du réseau comprenant les appareils, les composants passifs, les ports, les interconnexions et le diagnostic des ports

Vous pouvez également exécuter des tâches de configuration dans chaque vue. La fenêtre d'inspection montre les propriétés et informations de l'objet que vous avez sélectionné dans la zone de travail. Lorsque vous sélectionnez différents objets, la fenêtre d'inspection affiche les propriétés que vous pouvez configurer. La fenêtre d'inspection contient des onglets vous permettant de voir les informations de diagnostic et autres messages.

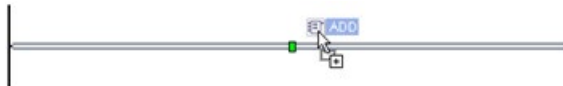
En affichant tous les éditeurs ouverts, la barre d'édition vous permet de travailler plus rapidement et efficacement. Pour basculer d'un éditeur ouvert à un autre, il suffit de cliquer sur l'éditeur correspondant. Vous pouvez également aligner deux éditeurs verticalement ou horizontalement. Cette fonction vous permet d'utiliser la fonction glisser-déplacer entre les éditeurs.

Le système d'information de STEP 7 fournit un système d'aide en ligne complet pour tous les outils de configuration, de programmation et de contrôle avec STEP 7. Vous pouvez vous y reporter pour obtenir des explications plus détaillées que celles fournies dans ce manuel.

3.3 Simplicité des outils

3.3.1 Insertion d'instructions dans votre programme utilisateur

STEP 7 fournit des Task Cards qui contiennent les instructions pour votre programme. Les opérations sont regroupées selon leur fonction.



Pour créer votre programme, vous amenez les opérations de la Task Card dans un réseau.

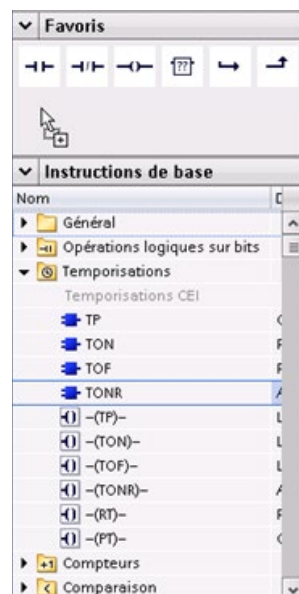


3.3.2 Accès aux instructions depuis la barre d'outils "Favoris"

STEP 7 fournit une barre d'outils "Favoris" pour permettre un accès rapide aux instructions que vous utilisez fréquemment. Effectuez un clic simple sur l'icône de l'opération à insérer dans votre réseau !



(Pour les "Favoris" dans l'arborescence d'instructions, double-cliquez sur l'icône)



Vous pouvez facilement personnaliser les "Favoris" en ajoutant de nouvelles opérations.

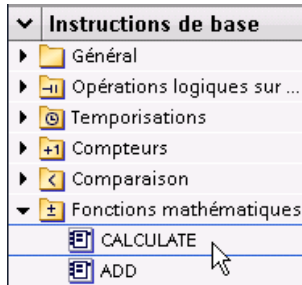
Il vous suffit de glisser-déplacer une opération sur vos "Favoris".

Un simple clic permet alors d'accéder à l'opération !

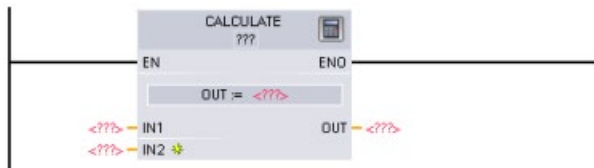


3.3.3 Création d'une équation complexe avec une instruction simple

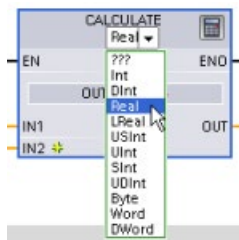
L'instruction Calculate (Page 268) vous permet de créer une fonction mathématique qui fonctionne sur des paramètres à entrées multiples pour produire le résultat, selon l'équation que vous définissez.



Dans l'arborescence d'opération Basic, agrandissez le fichier de fonctions Mathématiques. Double-cliquez sur l'opération Calculate pour insérer l'opération dans votre programme utilisateur.



L'opération Calculate non-configurée fournit deux paramètres d'entrée et un paramètre de sortie.

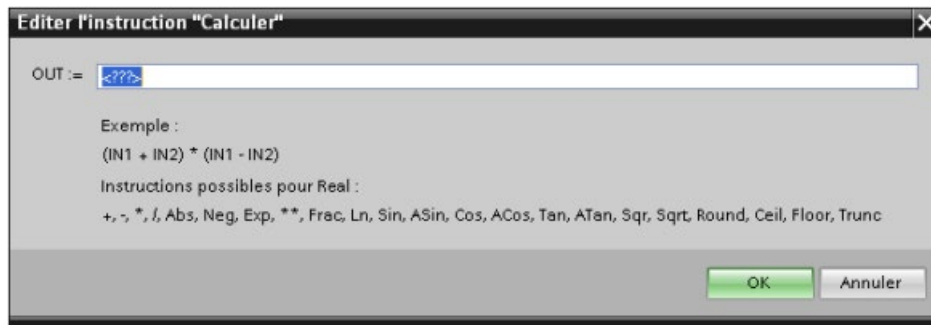


Cliquez sur "???" et sélectionnez les types de données pour les paramètres d'entrée et de sortie. (Les paramètres d'entrée et de sortie doivent tous être du même type de données.)

Pour cet exemple, sélectionnez le type de données "Real".



Cliquez sur l'icone "Éditer équation" pour saisir l'équation.



Pour cet exemple, saisissez l'équation suivante pour établir une valeur brute analogique. (Les désignations "Entrée" et "Sortie" correspondent aux paramètres de l'opération Calculer.)

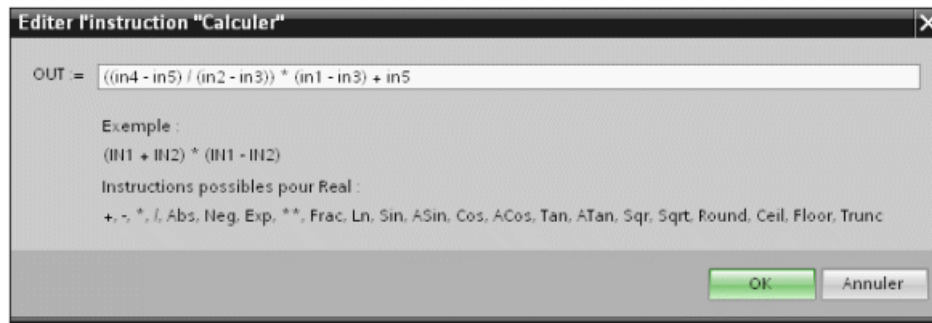
$$\text{Out}_{\text{value}} = ((\text{Out}_{\text{high}} - \text{Out}_{\text{low}}) / (\text{In}_{\text{high}} - \text{In}_{\text{low}})) * (\text{In}_{\text{value}} - \text{In}_{\text{low}}) + \text{Out}_{\text{low}}$$

$$\text{Out} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$

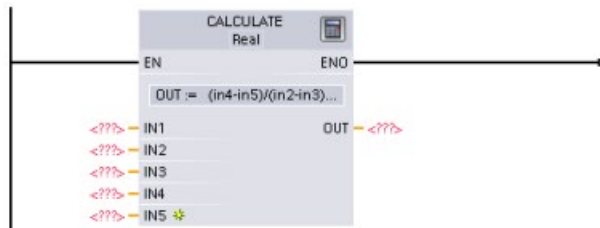
Si :	Out _{value}	(Out)	Valeur de sortie graduée
	In _{value}	(in1)	Valeur d'entrée analogique
	In _{high}	(in2)	Limite supérieure pour la valeur d'entrée graduée
	In _{low}	(in3)	Limite inférieure pour la valeur d'entrée graduée
	Out _{high}	(in4)	Limite supérieure pour la valeur de sortie graduée
	Out _{low}	(in5)	Limite inférieure pour la valeur de sortie normalisée

Dans le champ "Éditer Calculer", saisissez l'équation avec les noms de paramètre :

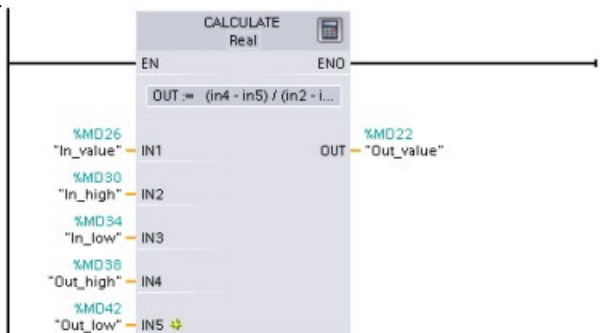
$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$



Lorsque vous cliquez sur "OK", l'instruction Calculate crée les entrées nécessaires à l'opération.



Saisissez les noms de variables pour les valeurs qui correspondent aux paramètres.



3.3.4 Ajout d'entrées ou de sorties à une instruction CONT ou LOG

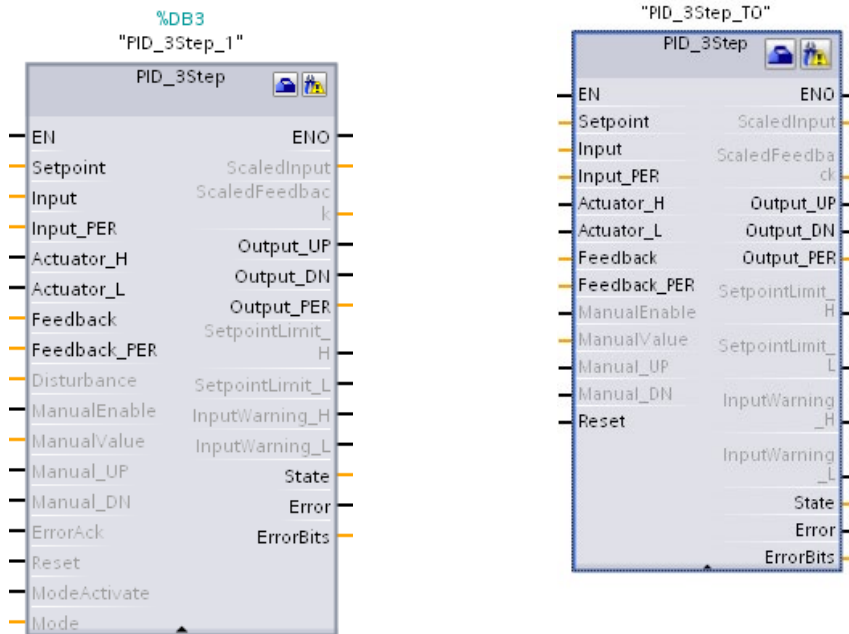


Certaines instructions vous permettent de créer des entrées ou sorties supplémentaires.

- Pour ajouter une entrée ou une sortie, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne de l'un des paramètres IN ou OUT existants et sélectionnez la commande "Insérer entrée".
- Pour supprimer une entrée ou une sortie, cliquez avec le bouton droit de la souris sur la ligne de l'un des paramètres IN ou OUT existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

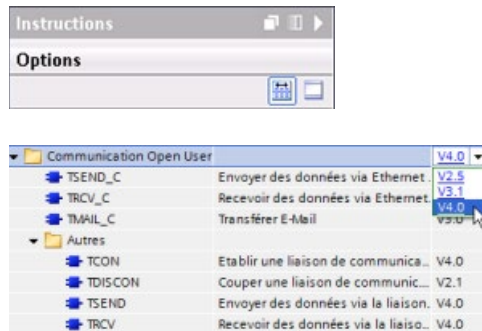
3.3.5 Instructions extensibles

Quelques-unes des instructions les plus complexes sont extensibles, n'affichant que les entrées et sorties importantes. Pour afficher toutes les entrées et sorties, cliquez sur la flèche au bas de l'instruction.



3.3.6 Sélection d'une version d'une instruction

Les cycles de développement et de mise à disposition pour certains jeux d'instructions (tels que Modbus, PID et commande de mouvement) ont créé de multiples versions disponibles de ces instructions. Pour vous aider à garantir la compatibilité et la migration avec des projets plus anciens, STEP 7 vous permet de choisir la version d'une instruction à insérer dans votre programme utilisateur.



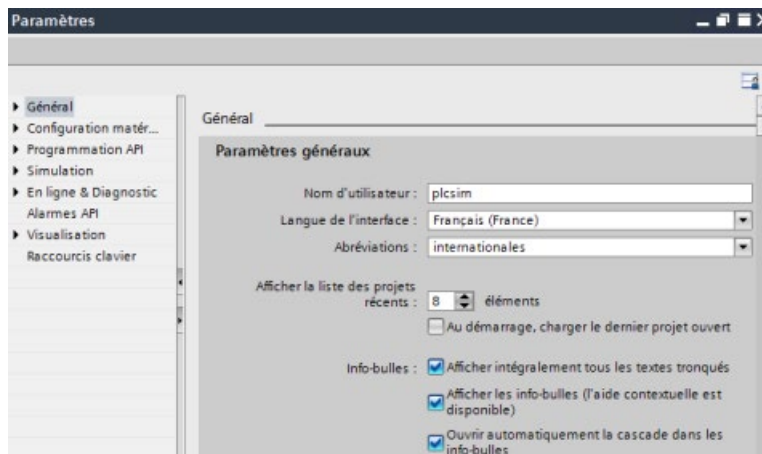
Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les entêtes et colonnes de l'arborescence d'instructions.

Pour changer la version de l'instruction, sélectionnez la version appropriée dans la liste déroulante.

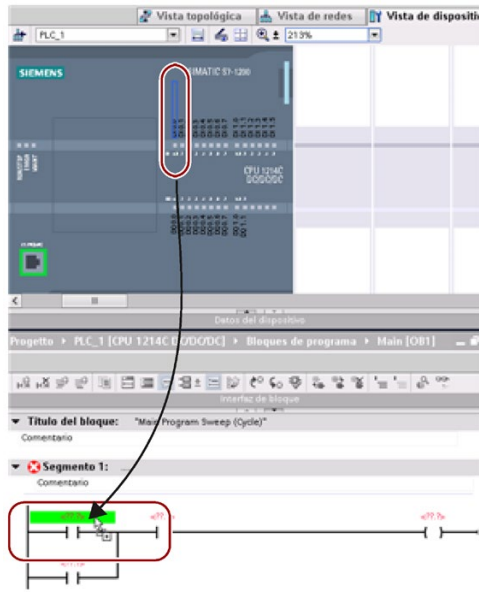
3.3.7 Modification de l'apparence et de la configuration de STEP 7

Vous pouvez sélectionner divers paramètres, tels que l'apparence de l'interface, la langue ou le dossier d'enregistrement de votre travail.

Pour modifier ces paramètres, sélectionnez la commande "Paramètres" dans le menu "Outils".



3.3.8 Glisser-déplacer entre les éditeurs

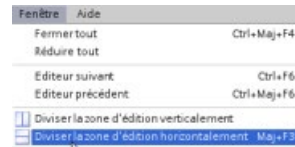


Pour vous aider à réaliser des tâches rapidement et facilement, STEP 7 vous permet de faire glisser des éléments d'un éditeur à un autre. Vous pouvez, par exemple, amener une entrée de la CPU sur l'adresse d'une opération dans votre programme utilisateur

Vous devez effectuer un agrandissement d'au moins 200 % pour pouvoir sélectionner les entrées ou les sorties de la CPU.

Notez que les noms des variables sont affichés non seulement dans la table des variables de l'API, mais également dans la CPU.

Pour afficher simultanément deux éditeurs, choisissez les commandes de menu "Fractionner éditeur" ou sélectionnez les boutons correspondants dans la barre d'outils.



Pour basculer entre les éditeurs ouverts, cliquez sur les icônes correspondantes dans la barre d'édition.



3.3.9 Changement de l'état de fonctionnement de la CPU

La CPU ne possède pas de commutateur physique pour changer l'état de fonctionnement (ARRET ou MARCHE).

Cliquez sur les boutons "Démarrer CPU" ou "Arrêter CPU" de la barre d'outils pour changer l'état de fonctionnement de la CPU.



Lorsque vous configurez la CPU dans la configuration d'appareil (Page 159), vous configurez le comportement au démarrage dans les propriétés de la CPU (Page 177).

Le portail "En ligne & diagnostic" comprend un panneau de commande pour modifier l'état de fonctionnement de la CPU en ligne. Pour pouvoir utiliser le panneau de commande de la CPU, vous devez être connecté en ligne à la CPU. La Task Card "Outils en ligne" affiche un panneau de commande indiquant l'état de fonctionnement de la CPU en ligne. Ce panneau de commande vous permet également de changer l'état de fonctionnement de la CPU en ligne.



Servez-vous du bouton sur le panneau de commande pour changer l'état de fonctionnement (ARRET ou MARCHE). Le panneau de commande dispose également d'un bouton MRES pour réinitialiser la mémoire.

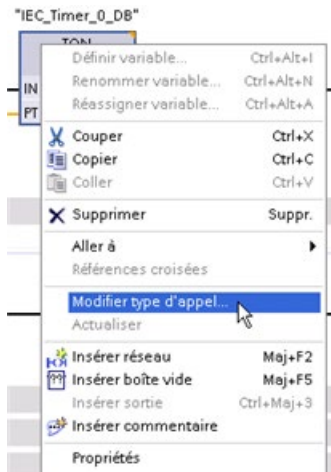
La couleur de l'indicateur MARCHE/ARRET signale l'état de fonctionnement en cours de la CPU. Le jaune correspond à l'état ARRET, le vert à l'état MARCHE.

Dans la configuration d'appareil de STEP 7 (Page 159), vous pouvez également configurer l'état de fonctionnement par défaut à la mise sous tension de la CPU (Page 89).

Remarque

Il est en outre possible de changer l'état de fonctionnement de la CPU à partir du serveur Web (Page 1037) ou dans l'outil SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/fr/view/98161300>).

3.3.10 Modification du type d'appel pour un DB



STEP 7 permet de créer et de changer aisément l'association d'un DB à une instruction ou à un FB se trouvant dans un FB.

- Vous pouvez permuter l'association entre différents DB.
- Vous pouvez permuter l'association entre un DB mono-instance et un DB multi-instance.
- Vous pouvez créer un DB d'instance (s'il manque ou n'est pas disponible).

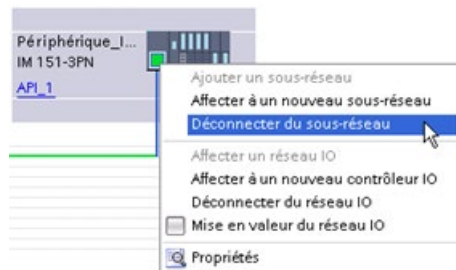
Vous pouvez accéder à la commande "Modifier type d'appel" en cliquant avec le bouton droit de la souris sur l'instruction ou le FB dans l'éditeur de programmes ou en sélectionnant la commande "Appel de bloc" dans le menu "Outils".



La boîte de dialogue "Options d'appel" vous permet de sélectionner un DB mono-instance ou un DB multi-instance. Vous pouvez également sélectionner des DB spécifiques dans une liste déroulante de DB disponibles.

3.3.11 Déconnexion temporaire d'appareils d'un réseau

Dans la vue de réseau, vous pouvez déconnecter des appareils individuels du sous-réseau. Comme la configuration de l'appareil n'est pas supprimée du projet, vous pouvez aisément restaurer la connexion à l'appareil.



Cliquez avec le bouton droit de la souris sur le port d'interface de l'appareil du réseau et sélectionnez la commande "Déconnecter du sous-réseau" dans le menu contextuel.

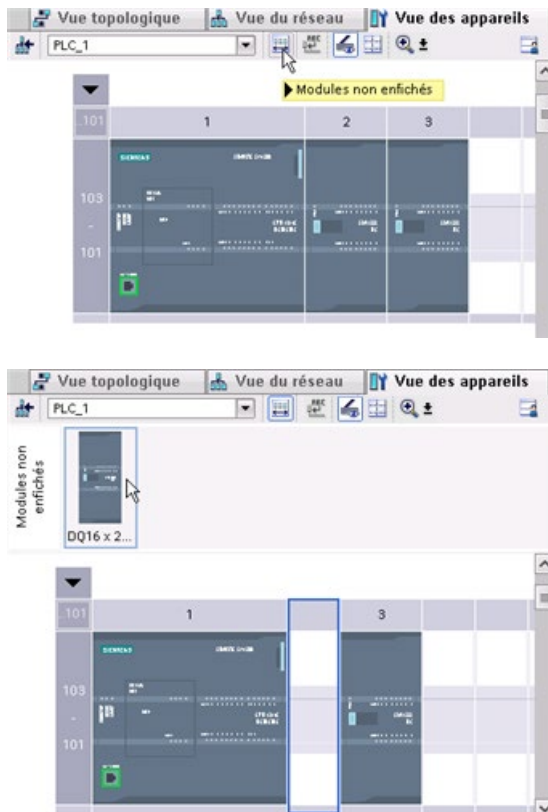
STEP 7 reconfigure les liaisons réseau, mais ne supprime pas l'appareil déconnecté du projet. Alors que la connexion réseau est supprimée, les adresses d'interface ne sont pas modifiées.



Vous devez faire passer la CPU à l'état ARRET lorsque vous chargez les nouvelles connexions réseau.

Pour reconnecter l'appareil, il vous suffit de créer une nouvelle connexion réseau au port de cet appareil.

3.3.12 Désenfichage virtuel d'appareils de la configuration



STEP 7 propose une zone de stockage pour les modules "retirés". Vous avez la possibilité d'amener un module du châssis afin d'en enregistrer la configuration. Ces modules retirés sont enregistrés dans votre projet, ce qui permettra de les réinsérer ultérieurement sans avoir à redéfinir les paramètres de configuration.

L'une des applications de cette fonctionnalité est la maintenance temporaire. Considérez l'hypothèse où vous auriez à attendre avant de pouvoir remplacer un module et où vous décideriez d'utiliser temporairement un module différent comme solution à court terme. Vous auriez la possibilité d'amener le module configuré du châssis dans les "Modules non enfichés", puis d'insérer le module temporaire.

3.4 Compatibilité amont

STEP 7 V15 prend en charge la configuration et la programmation de la CPU S7-1200 V4.2.x et permet l'utilisation de toutes les nouvelles fonctions (Page 35).

Vous pouvez charger des projets pour des CPU S7-1200 V4.0 et V4.1 de STEP 7 (V13 SP1 ou plus) dans une CPU S7-1200 V4.2.x. Votre configuration et votre programme seront limités au jeu de fonctions et d'instructions pris en charge par la version précédente de la CPU S7-1200 et votre version de STEP 7.

Cette compatibilité amont vous permet d'exécuter sur les nouveaux modèles de CPU S7-1200 V4.2 des programmes que vous aviez conçus et programmés pour des versions antérieures.

ATTENTION

Risques liés à la copie et à l'insertion de logique de programme provenant de versions antérieures de STEP 7

La copie de logique de programme d'une ancienne version de STEP 7, STEP 7 V12 par exemple, dans STEP 7 V15 peut provoquer un comportement imprévisible lors de l'exécution du programme ou un échec de la compilation. En effet, différentes versions de STEP 7 réalisent les éléments de programme de différentes manières. Le compilateur ne détecte pas toujours les différences si vous avez apporté les modifications par copier-coller d'une version antérieure dans STEP 7 V15. L'exécution d'une logique de programme imprévisible peut entraîner la mort ou des blessures graves si vous ne corrigez pas le programme.

Si vous utilisez une logique de programme d'une version de STEP 7 antérieure à STEP 7 V15, une mise à niveau du projet entier à STEP 7 V15 est requise. Vous pourrez alors copier, couper, coller et éditer la logique de programme selon vos besoins. Dans STEP 7 V15, vous pouvez ouvrir un projet de STEP 7 V13 SP1 ou plus. STEP 7 effectue les conversions de compatibilité nécessaires et met le programme à niveau de manière appropriée. De telles corrections et conversions de mise à niveau sont nécessaires pour assurer une compilation et une exécution correctes du programme. Si votre projet est antérieur à STEP 7 V13 SP1, une mise à niveau progressive à la version STEP 7 V15 est requise.

Vous ne pouvez pas charger de projets pour des CPU S7-1200 V1.0, V2.0 ou V3.0 dans une CPU S7-1200 V4.2.x. La rubrique Remplacement de l'appareil et compatibilité des pièces de rechange (Page 1665) donne des conseils pour la mise à niveau d'anciens projets à un projet que vous pouvez charger.

Remarque

Projets avec des versions de CPU S7-1200 V1.x

Vous ne pouvez pas ouvrir un projet STEP 7 contenant des CPU S7-1200 V1.x dans STEP 7 V15. Pour pouvoir utiliser votre projet existant, vous devez l'ouvrir avec STEP 7 V13 SP1 (toutes mises à niveau) et convertir les CPU S7-1200 V1.x à la version V2.0 ou plus. Vous pourrez alors ouvrir avec STEP 7 V15 le projet sauvegardé contenant les CPU converties.

Installation

4.1 Conseils pour l'installation d'appareils S7-1200

Le matériel S7-1200 est conçu pour être facile à installer. Vous pouvez monter l'automate S7-1200 sur un panneau ou sur un profilé support et l'orienter horizontalement ou verticalement. La petite taille du S7-1200 permet une optimisation de l'espace.

Les normes relatives au matériel électrique classent le système SIMATIC S7-1200 comme Equipement ouvert. Vous devez installer le S7-1200 dans un boîtier, une armoire ou une salle électrique auxquels seules les personnes autorisées doivent avoir accès.

Le S7-1200 doit être installé dans un environnement sec. On considère que les circuits TBTS/TBTP offrent une protection contre les chocs électriques dans les endroits secs.

L'installation doit offrir une résistance mécanique, une protection contre l'inflammabilité et une stabilité agréées pour un équipement ouvert dans la catégorie de votre emplacement selon les codes électriques et de construction applicables.

Une contamination conductrice occasionnée par la poussière, l'humidité ou la pollution atmosphérique peut entraîner des dysfonctionnements et des défauts électriques dans l'API.

Si vous placez l'API dans une zone susceptible de renfermer une pollution conductrice, l'API doit être protégé dans une enceinte ayant un indice de protection approprié. L'indice IP54 est généralement utilisé pour les enceintes de matériel électrique dans des environnements souillés et il se peut qu'il soit approprié pour votre application.

ATTENTION

Une installation inappropriée du S7-1200 peut provoquer des défauts électriques ou un fonctionnement inattendu des machines.

Les défauts électriques ou un fonctionnement inattendu des machines peuvent entraîner la mort, des blessures graves et des dommages matériels importants.

Il convient de respecter toutes les instructions d'installation et de maintien d'un environnement de fonctionnement approprié afin d'assurer le fonctionnement sûr de l'équipement.

Ne placez pas les appareils S7-1200 à proximité de sources de chaleur, de haute tension et de bruit électrique

En règle générale pour la mise en place des appareils de votre système, séparez toujours les appareils générant une tension élevée et un bruit électrique important des appareils de type logique basse tension, tels que le S7-1200.

Lors de l'implantation du S7-1200 dans votre panneau, tenez compte des appareils sources de chaleur et placez les appareils de type électronique dans les zones plus fraîches de votre armoire. En effet, réduire l'exposition aux températures élevées allonge la durée de vie des appareils électroniques.

Tenez également compte de l'acheminement du câblage pour les appareils dans le panneau. Evitez de placer les câbles de signaux et de communication de faible tension dans la même goulotte que le câblage d'alimentation en courant alternatif et le câblage pour courant continu à commutation rapide et haute énergie.

Ménagez un dégagement adéquat pour le refroidissement et le câblage

Les appareils S7-1200 sont conçus pour un refroidissement par convection naturelle. Pour que le refroidissement se fasse correctement, vous devez laisser un espace libre d'au moins 25 mm au-dessus et en dessous des appareils. Vous devez également avoir une profondeur d'au moins 25 mm entre l'avant des modules et l'intérieur de l'enceinte.

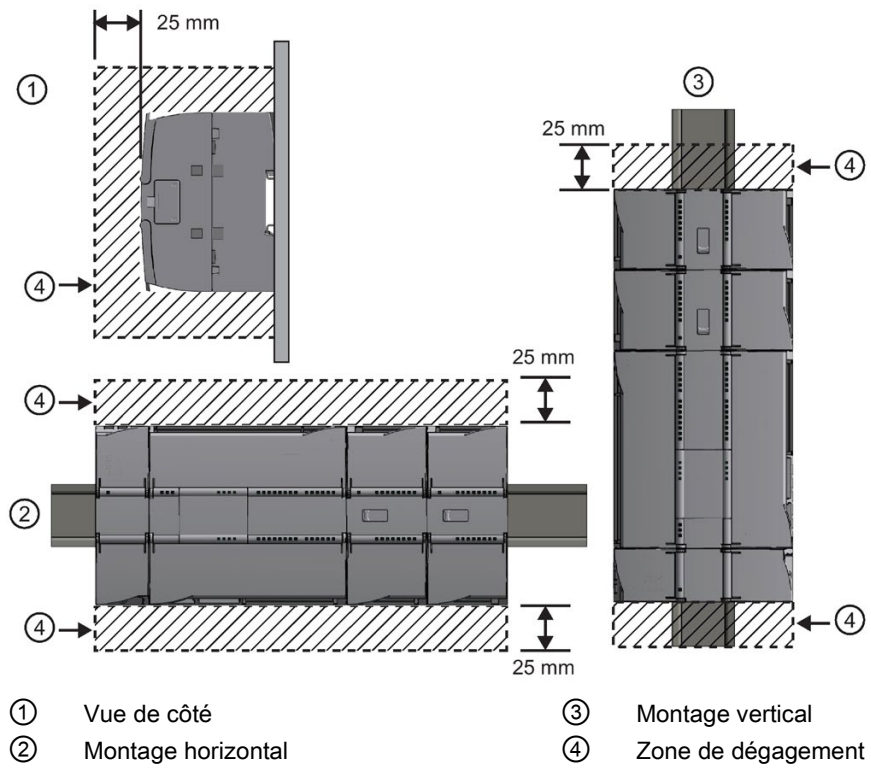
 **PRUDENCE**

En cas de montage vertical, la température ambiante maximale autorisée est diminuée de 10 °C.

Orientez un système S7-1200 monté verticalement comme illustré dans la figure suivante.

Assurez-vous que le système S7-1200 est monté correctement.

Lorsque vous planifiez la disposition de votre système S7-1200, réservez suffisamment d'espace pour le câblage et les connexions de câbles de communication.



① Vue de côté
② Montage horizontal

③ Montage vertical
④ Zone de dégagement

4.2 Bilan de consommation

Votre CPU possède une alimentation interne fournissant du courant à la CPU, aux modules d'entrées-sorties, au Signal Board et aux modules de communication ainsi qu'à d'autres équipements consommant du courant 24 V CC.

Reportez-vous aux Caractéristiques techniques (Page 1429) pour plus d'informations sur le bilan de consommation constitué par le courant 5 V CC fourni par votre CPU et par les besoins en courant 5 V CC des modules d'entrées-sorties, des Signal Boards et des modules de communication. Reportez-vous au "Calcul d'un bilan de consommation" (Page 1647) pour déterminer combien de courant la CPU peut mettre à la disposition de votre configuration.

La CPU dispose également d'une alimentation de capteur 24 V CC pouvant fournir du courant 24 V CC aux entrées, aux bobines de relais sur les modules d'entrées-sorties ou à d'autres équipements. Si vos besoins en courant 24 V CC dépassent ce que fournit l'alimentation de capteur, vous devez ajouter à votre système un module d'alimentation 24 V CC externe. Vous trouverez dans les Caractéristiques techniques (Page 1429) le bilan de consommation pour l'alimentation de capteur 24 V CC de votre CPU particulière.

Si vous avez besoin d'un module d'alimentation 24 V CC externe, assurez-vous que ce module n'est pas connecté en parallèle avec l'alimentation de capteur de la CPU. En effet, il est recommandé, pour une meilleure protection contre les bruits électriques, que les communs (M) des différentes alimentations soient connectés.

 **ATTENTION**

Connecter une alimentation 24 V CC externe en parallèle avec l'alimentation de capteur 24 V CC peut entraîner un conflit entre les deux alimentations, chacune cherchant à établir son propre niveau de tension de sortie préféré.

Ce conflit peut réduire la durée de vie ou provoquer une défaillance immédiate de l'une ou des deux alimentations, ayant pour effet un fonctionnement imprévisible du système d'automatisation pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

L'alimentation de capteur CC et toute alimentation externe doivent fournir du courant à des points différents.

Certains ports d'entrée d'alimentation 24 V CC dans le système S7-1200 sont interconnectés, avec un circuit logique commun connectant plusieurs bornes M. Par exemple, les circuits suivants sont interconnectés lorsqu'ils sont signalés comme "non isolés" dans les fiches techniques : l'alimentation 24 V CC de la CPU, l'entrée d'alimentation pour la bobine de relais d'un SM ou l'alimentation pour une entrée analogique non isolée. Toutes les bornes M non isolées doivent être connectées au même potentiel de référence externe.

 ATTENTION

Connecter des bornes M non isolées à des potentiels de référence différents provoque des flux de courant indésirables qui peuvent être à l'origine de dégâts ou d'un fonctionnement imprévisible dans l'automate et tout équipement connecté.

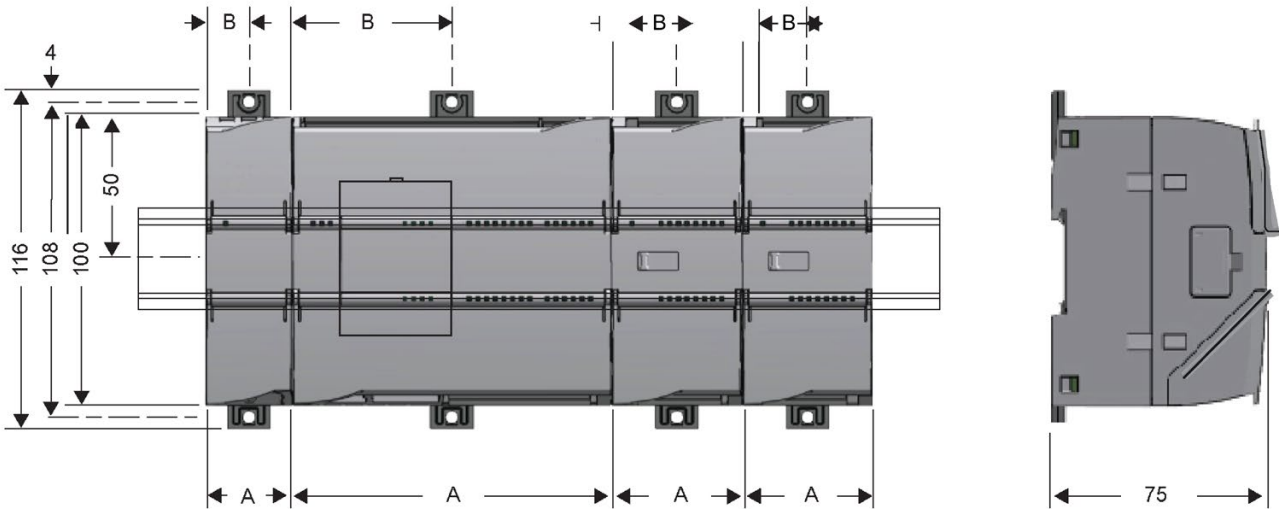
Le non-respect de ces conseils peut être à l'origine de dégâts ou d'un fonctionnement imprévisible pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Vérifiez toujours que toutes les bornes M non isolées dans un système S7-1200 sont connectées au même potentiel de référence.

4.3 Procédures d'installation et de désinstallation

4.3.1 Dimensions de montage pour les appareils S7-1200

CPU 1211C, CPU 1212C, CPU 1214C
(measurements in mm)



CPU 1215C, CPU 1217C
(measurements in mm)

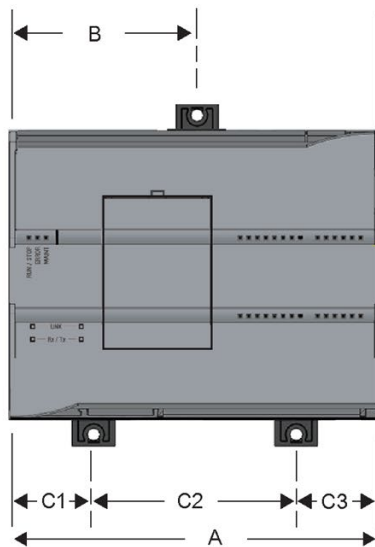


Tableau 4- 1 Dimensions de montage (mm)

Appareils S7-1200		Largeur A (mm)	Largeur B (mm)	Largeur C (mm)
CPU	CPU 1211C et CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (haut)	Bas : C1 : 32,5 C2 : 65 C3 : 32,5
	CPU 1217C	150	75	Bas : C1 : 37,5 C2 : 75 C3 : 37,5
Modules d'en- trées-sorties	TOR, 8 et 16 entrées/sorties Analogiques, 2, 4 et 8 entrées/sorties Thermocouple, 4 et 8 entrées/sorties RTD, 4 entrées/sorties SM 1278 maître IO-Link	45	22,5	--
	TOR, 8 sorties Relais (inverseur)	70	35	--
	Analogiques, 16 entrées/sorties RTD, 8 entrées/sorties	70	35	--
	Module SM 1238 Energy Meter	45	22,5	--
Interfaces de communication	CM 1241 RS232 et CM 1241 RS422/485 CM 1243-5 maître PROFIBUS et CM 1242-5 esclave PROFIBUS CM 1242-2 maître AS-i CP 1242-7 GPRS V2 CP 1243-7 LTE-US CP 1243-7 LTE-EU CP 1243-1 CP 1243-8 IRC RF120C	30	15	--
	Adaptateur TS (TeleService) Adapter IE Advanced ¹ Adaptateur TS (TeleService) Adapter IE Basic ¹ Adaptateur TS Adapter Module TS	30 30	15 15	-- --

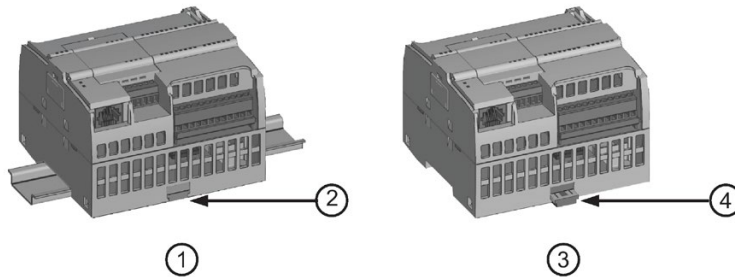
¹ Avant d'installer l'adaptateur TS (TeleService) Adapter IE Advanced ou IE Basic, vous devez d'abord connecter l'adaptateur TS Adapter et un module TS. La largeur totale ("largeur A") est de 60 mm.

Chaque CPU, SM, CM et CP peut être indifféremment monté sur un profilé support ou encastré dans un panneau. Servez-vous des barrettes de fixation sur le module pour fixer l'appareil sur le profilé support. Ces barrettes s'encliquettent également en position sortie afin de fournir des points de vissage pour monter l'unité directement sur un panneau. Le diamètre intérieur de l'alésage pour les barrettes de fixation de l'appareil est de 4,3 mm.

Une zone thermique de 25 mm doit également être ménagée au-dessus et en dessous de l'unité pour la libre circulation de l'air.

Installation et désinstallation des appareils S7-1200

Il est très aisé d'installer la CPU sur un profilé support ou sur un panneau. Des barrettes de fixation sont fournies pour fixer l'appareil sur le profilé support. Ces barrettes s'encliquettent également en position sortie afin de fournir un point de vissage pour le montage sur panneau de l'unité.



- ① Montage sur profilé support
- ② Barrette de fixation en position rentrée
- ③ Montage sur panneau
- ④ Barrette en position sortie pour le montage sur panneau

Assurez-vous, avant d'installer ou de démonter tout appareil électrique, que cet appareil a été mis hors tension. Assurez-vous également que tout équipement associé a été mis hors tension.

<p>⚠ ATTENTION</p> <p>L'installation ou le démontage du S7-1200 ou d'équipements reliés alors qu'ils sont sous tension peut provoquer un choc électrique ou un fonctionnement imprévisible du matériel.</p> <p>La non-désactivation de l'alimentation complète du S7-1200 et des appareils qui y sont reliés pendant les procédures d'installation ou de désinstallation peut entraîner la mort, des blessures graves ou des dommages matériels importants en raison du choc électrique ou du fonctionnement imprévisible du matériel.</p> <p>Prenez toujours toutes les mesures de sécurité nécessaires et assurez-vous que l'alimentation de l'automate S7-1200 est coupée avant de tenter d'installer ou de démonter des CPU S7-1200 ou des matériels reliés.</p>
--

Assurez-vous toujours, lorsque vous remplacez ou installez un appareil S7-1200, que vous utilisez le module correct ou un appareil équivalent.

 ATTENTION

Une installation incorrecte d'un module S7-1200 peut entraîner un fonctionnement imprévisible du programme dans le S7-1200.

Le remplacement d'un appareil S7-1200 par un autre modèle, sans respecter l'orientation correcte et l'ordre initial, peut entraîner la mort, des blessures graves ou des dommages matériels importants en raison d'un fonctionnement imprévisible de l'équipement.

Remplacez toujours un appareil S7-1200 par le même modèle et assurez-vous que vous l'orientez et le positionnez correctement.

 ATTENTION

Ne déconnectez pas les équipements en cas d'atmosphère inflammable ou combustible.

En effet, la déconnexion d'équipements en présence d'une atmosphère inflammable ou combustible peut provoquer un incendie ou une explosion, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Respectez toujours les mesures de sécurité appropriées en présence d'atmosphère inflammable ou combustible.

Remarque

Des décharges électrostatiques peuvent endommager l'appareil ou son emplacement sur la CPU.

Pour éviter tout risque lorsque vous manipulez l'appareil, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette.

4.3.2 Installation et désinstallation de la CPU

Vous pouvez monter la CPU sur un panneau ou sur un profilé support.

Remarque

Reliez les modules de communication éventuels à la CPU et installez l'ensemble en tant qu'unité. Installez les modules d'entrées-sorties séparément une fois la CPU installée.

Tenez compte des points suivants lorsque vous installez les unités sur le profilé support ou sur un panneau :

- En cas de montage sur profilé support, assurez-vous que la barrette de fixation supérieure est en position rentrée et que la barrette inférieure est en position sortie sur la CPU et les CM reliés.
- Une fois les appareils installés sur le profilé support, faites glisser les barrettes de fixation inférieure dans leur position rentrée pour verrouiller les appareils sur le profilé support.
- En cas de montage sur panneau, assurez-vous que les barrettes de fixation sont en position sortie.

Procédez comme suit pour installer la CPU sur un panneau :

1. Positionnez, percez et taraudez les trous de fixation (M4) en vous aidant des dimensions indiquées dans le tableau Dimensions de montage (mm) (Page 58).
2. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.
3. Sortez les barrettes de fixation du module. Assurez-vous que les barrettes de fixation en haut et en bas de la CPU sont bien en position sortie.
4. Fixez le module sur le panneau à l'aide d'une vis M4 à tête cylindrique avec rondelle plate élastique. N'utilisez pas de vis à tête plate.

Remarque

Vous déterminerez le type de vis en fonction du matériau sur lequel vous la fixerez. Vous devez appliquer un couple approprié jusqu'à ce que la rondelle élastique soit aplatie. Veillez à ne pas appliquer un couple excessif aux vis de montage. N'utilisez pas de vis à tête plate.

Remarque

Les butées de rail DIN peuvent être utiles si votre CPU se trouve dans un environnement soumis à de fortes vibrations ou si vous l'avez installée verticalement. Utilisez une équerre d'extrémité (8WA1808 ou 8WA1805) sur le rail DIN pour être sûr que les modules restent connectés. Si votre système se situe dans un environnement à fortes vibrations, le montage de la CPU sur panneau offrira un niveau de protection plus élevé contre les vibrations.

Tableau 4- 2 Installation de la CPU sur un profilé support

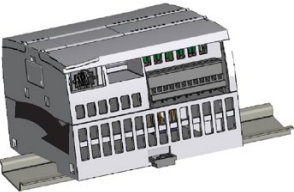
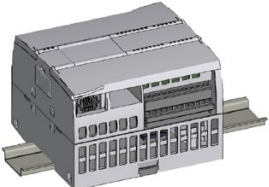
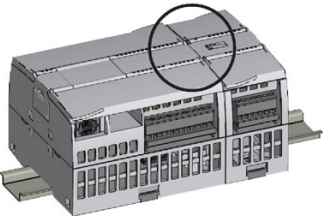
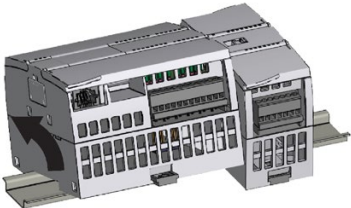
Tâche	Marche à suivre
	<ol style="list-style-type: none"> 1. Installez le profilé support. Vissez le profilé support tous les 75 mm sur le panneau de montage. 2. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 3. Accrochez la CPU sur le haut du profilé support. 4. Tirez la barrette de fixation située au bas de la CPU pour que la CPU puisse s'enclencher sur le profilé support.
	<ol style="list-style-type: none"> 5. Faites pivoter la CPU vers le bas sur le profilé support. 6. Repoussez les barrettes de fixation pour verrouiller la CPU sur le profilé support.

Tableau 4- 3 Désinstallation de la CPU d'un profilé support

Tâche	Marche à suivre
	<ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Déconnectez les connecteurs d'E/S, les fils et les câbles de la CPU (Page 69). 3. Démontez la CPU et tout module de communication relié en tant qu'unité. Tous les modules d'entrées-sorties doivent rester en place.
	<ol style="list-style-type: none"> 4. Si un module SM est connecté à la CPU, rentrez le connecteur de bus : <ul style="list-style-type: none"> – Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties. – Appuyez pour dégager le connecteur de la CPU. – Faites glisser la languette complètement à droite. 5. Démontez la CPU : <ul style="list-style-type: none"> – Tirez la barrette de fixation pour libérer la CPU du profilé support. – Faites pivoter la CPU vers le haut pour la dégager du profilé support et retirez-la du système.

4.3.3 Installation et désinstallation d'un SB, d'un CB ou d'un BB

Tableau 4- 4 Installation d'un SB, d'un CB ou d'un BB 1297

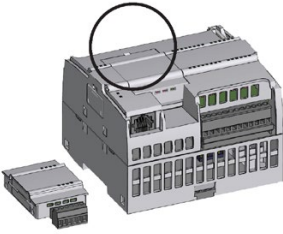
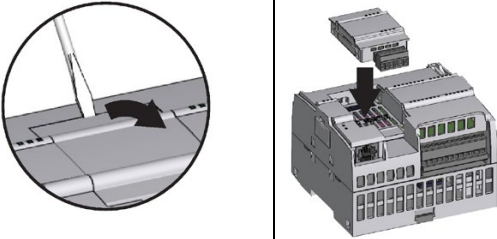
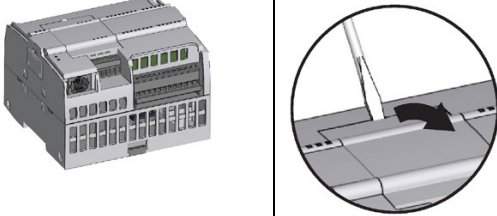
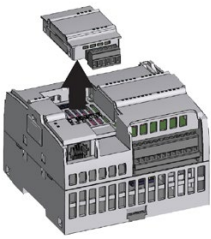
Tâche	Marche à suivre
	<ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Retirez les caches des borniers supérieur et inférieur de la CPU. 3. Placez un tournevis dans l'encoche à l'arrière du cache sur le haut de la CPU. 4. Faites doucement levier verticalement vers le haut et retirez le cache de la CPU. 5. Insérez le module verticalement dans sa position de montage dans la partie supérieure de la CPU.
	<ol style="list-style-type: none"> 6. Appuyez fermement sur le module jusqu'à ce qu'il s'enclenche en position. 7. Remplacez les caches des borniers.

Tableau 4- 5 Désinstallation d'un SB, d'un CB ou d'un BB 1297

Tâche	Marche à suivre
	<ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Retirez les caches des borniers supérieur et inférieur de la CPU. 3. Retirez, le cas échéant, le connecteur du Signal Board en le dégageant doucement avec un tournevis. 4. Placez un tournevis dans l'encoche sur le haut du module. 5. Faites doucement levier pour dégager le module de la CPU. 6. Sans utiliser de tournevis, retirez le module verticalement de sa position de montage dans la partie supérieure de la CPU.
	<ol style="list-style-type: none"> 7. Remplacez le cache sur la CPU. 8. Remplacez les caches des borniers.

Installation ou remplacement de la pile dans le Battery Board BB 1297

Le BB 1297 fonctionne avec une pile de type CR1025. La pile n'est pas fournie avec le BB 1297 et doit être achetée séparément. Procédez comme suit pour installer ou remplacer la pile :

1. Placez une nouvelle pile dans le BB 1297, avec le pôle positif de la pile vers le haut et le pôle négatif à côté de la carte de câblage imprimée.
2. Le BB 1297 est prêt à être installé dans la CPU. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension et suivez les instructions d'installation données plus haut pour installer le BB 1297.

Procédez comme suit pour remplacer la pile dans le BB 1297 :

1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. Retirez le BB 1297 de la CPU en suivant les instructions de démontage données plus haut.
2. Retirez l'ancienne pile avec précaution à l'aide d'un petit tournevis. Extrayez la pile de dessous l'attache.
3. Installez une nouvelle pile de remplacement CR1025 avec le pôle positif de la pile vers le haut et le pôle négatif à côté de la carte de câblage imprimée.
4. Réinstallez le Battery Board BB 1297 en suivant les instructions d'installation données plus haut.

ATTENTION

Installer une pile non spécifiée dans le BB 1297, ou connecter de toute autre manière une pile non spécifiée au circuit peut provoquer un incendie ou endommager les composants et peut entraîner un fonctionnement imprévisible des machines.

Un incendie ou un fonctionnement imprévisible des machines peut entraîner la mort, des blessures graves ou des dommages matériels.

Utilisez uniquement une pile CR1025 spécifiée pour effectuer une sauvegarde de l'horloge temps réel.

4.3.4 Installation et désinstallation d'un SM

Tableau 4- 6 Installation d'un SM

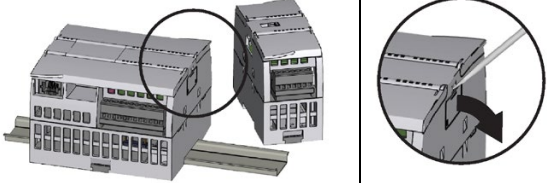
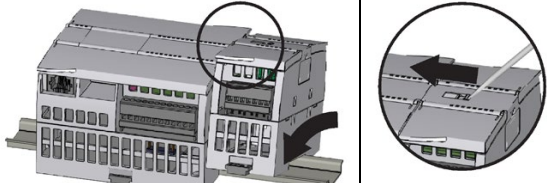
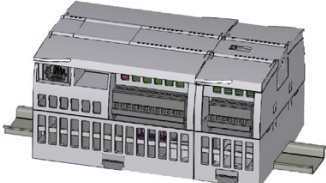
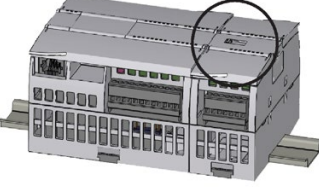

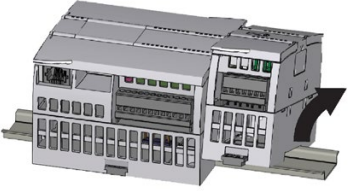
Tâche	Marche à suivre
	<p>Installez votre module d'entrées-sorties après avoir installé la CPU.</p> <ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Retirez le cache pour le connecteur du côté droit de la CPU : <ul style="list-style-type: none"> – Insérez un tournevis dans l'encoche au-dessus du cache. – Détachez doucement le côté supérieur du cache puis retirez le cache. 3. Conservez-le en vue de sa réutilisation.
	<p>Raccordez le SM à la CPU :</p> <ol style="list-style-type: none"> 1. Positionnez le SM à côté de la CPU. 2. Accrochez le SM sur le haut du profilé support. 3. Tirez la barrette de fixation inférieure pour que le SM puisse s'enclencher sur le profilé support. 4. Faites pivoter le SM vers le bas à côté de la CPU et repoussez la barrette de fixation inférieure pour verrouiller le SM sur le profilé support.
	<p>L'insertion du connecteur de bus réalise les connexions mécaniques et électriques pour le SM.</p> <ol style="list-style-type: none"> 1. Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties. 2. Faites glisser la languette complètement vers la gauche pour insérer le connecteur de bus dans la CPU. <p>Procédez de la même façon pour installer un module d'entrées-sorties sur un autre module d'entrées-sorties.</p>

Tableau 4- 7 Désinstallation d'un SM

Tâche	Marche à suivre
	<p>Vous pouvez démonter n'importe quel SM sans démonter la CPU ni d'autres SM en place.</p> <ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Retirez les connecteurs d'E/S et le câblage du SM (Page 69). 3. Retirez le connecteur de bus : <ul style="list-style-type: none"> – Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties. – Appuyez pour dégager le connecteur de la CPU. – Faites glisser la languette complètement à droite. <p>S'il y a un autre SM à droite, répétez cette procédure pour ce SM.</p>
	
	<p>Démontez le SM :</p> <ol style="list-style-type: none"> 1. Tirez la barrette de fixation inférieure pour libérer le SM du profilé support. 2. Faites pivoter le SM vers le haut pour le dégager du profilé support. Retirez le SM du système. 3. Si nécessaire, couvrez le connecteur de bus sur la CPU de son cache pour éviter la contamination. <p>Procédez de la même façon pour démonter un module d'entrées-sorties d'un autre module d'entrées-sorties.</p>

4.3.5 Installation et désinstallation d'un CM ou d'un CP

Reliez les modules de communication éventuels à la CPU et installez l'ensemble en tant qu'unité, comme illustré au paragraphe Installation et désinstallation de la CPU (Page 62).

Tableau 4- 8 Installation d'un CM ou d'un CP

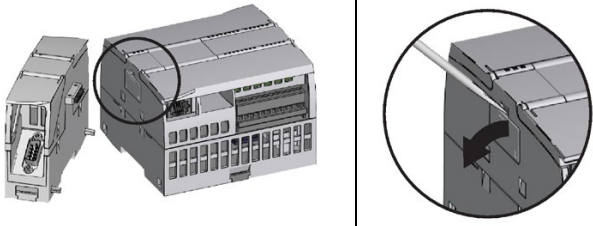
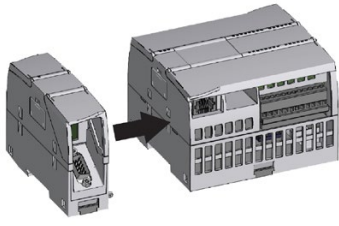
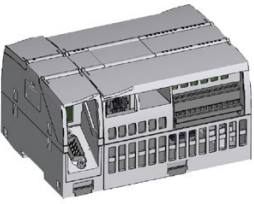
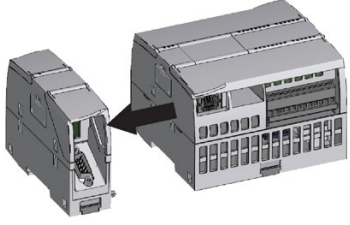
Tâche	Marche à suivre
	<ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Reliez le CM à la CPU avant d'installer l'ensemble en tant qu'unité sur le profilé support ou le panneau. 3. Retirez le cache de bus sur le côté gauche de la CPU : <ul style="list-style-type: none"> – Insérez un tournevis dans l'encoche au-dessus du cache de bus. – Détachez doucement le côté supérieur du cache. 4. Retirez le cache de bus. Conservez-le en vue de sa réutilisation. 5. Raccordez le CM ou le CP à la CPU : <ul style="list-style-type: none"> – Aligned le connecteur de bus et les montants du CM avec les trous de la CPU. – Appuyez fermement les unités l'une contre l'autre jusqu'à ce que les montants s'enclenchent. 6. Installez la CPU et le CP sur un profilé support ou dans un panneau.
	

Tableau 4- 9 Désinstallation d'un CM ou d'un CP

Tâche	Marche à suivre
	<p>Démontez la CPU et le CM du profilé support ou du panneau en tant qu'unité.</p> <ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Retirez les connecteurs d'E/S et tous les fils et câbles de la CPU et des CM. 3. En cas de montage sur profilé support, tirez les barrettes de fixation inférieures sur la CPU et les CM en position sortie. 4. Démontez la CPU et les CM du profilé support ou du panneau. 5. Maintenez fermement la CPU et le CM et tirez pour les séparer.
	

IMPORTANT

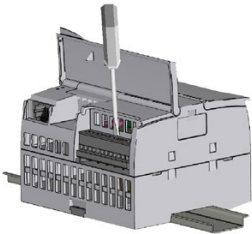
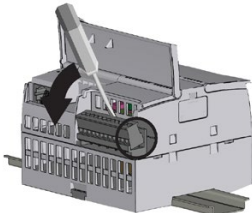
Séparez les modules sans utiliser d'outil.

N'utilisez pas d'outil pour séparer les modules, car cela risquerait de les endommager.

4.3.6 Démontage et remontage du bornier de connexion S7-1200

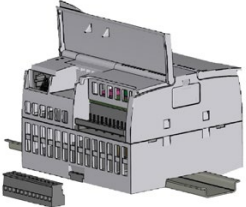
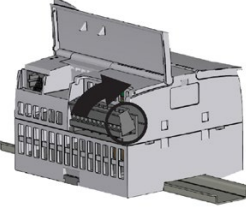
Les CPU, les SB et les SM comportent des connecteurs amovibles pour faciliter le câblage.

Tableau 4- 10 Démontage du bornier de connexion

Tâche	Marche à suivre
	<p>Préparez le système au démontage du bornier de connexion en mettant la CPU hors tension et en ouvrant le cache au-dessus du bornier de connexion.</p> <ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Inspectez le dessus du connecteur afin de localiser l'encoche pour la pointe du tournevis. 3. Insérez un tournevis dans l'encoche. 4. Faites doucement levier sur le haut du connecteur pour le détacher de la CPU. Le connecteur se dégagera avec un bruit sec. 5. Saisissez le connecteur et retirez-le de la CPU.
	

4.3 Procédures d'installation et de désinstallation

Tableau 4- 11 Installation du bornier de connexion

Tâche	Marche à suivre
	<p>Préparez les composants en vue de l'installation du bornier de connexion en mettant la CPU hors tension et en ouvrant le cache pour le bornier.</p> <ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Alignez le connecteur avec les broches sur l'unité. 3. Alignez la face de câblage du connecteur à l'intérieur du bord du socle de connecteur. 4. Appuyez fermement et faites pivoter le connecteur jusqu'à ce qu'il s'enclenche.
	<p>Vérifiez soigneusement que le connecteur est correctement aligné et bien enfoncé.</p>

4.3.7 Installation et désinstallation du câble d'extension

Le câble d'extension S7-1200 offre une souplesse accrue lors de la configuration de la disposition de votre système S7-1200. Un seul câble d'extension est autorisé par système CPU. Vous installez le câble d'extension soit entre la CPU et le premier SM, soit entre deux SM quelconques.

Tableau 4- 12 Installation et désinstallation du connecteur mâle du câble d'extension

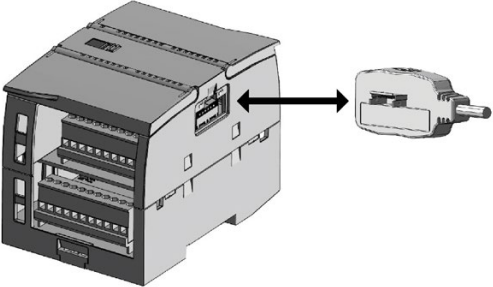
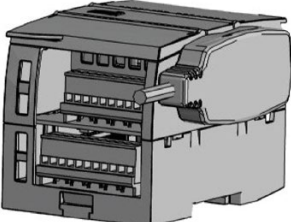
Tâche	Marche à suivre
	<p>Pour installer le connecteur mâle :</p> <ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Enfoncez le connecteur dans le connecteur de bus du côté droit du module d'entrées-sorties ou de la CPU. <p>Pour désinstaller le connecteur mâle :</p> <ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Tirez sur le connecteur mâle pour l'extraire du module d'entrées-sorties ou de la CPU.
	

Tableau 4- 13 Installation du connecteur femelle du câble d'extension

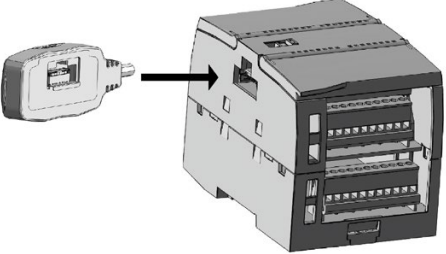
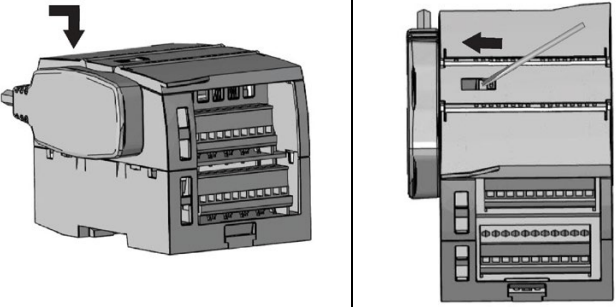
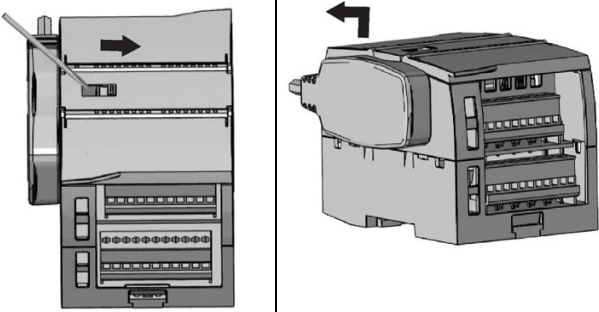
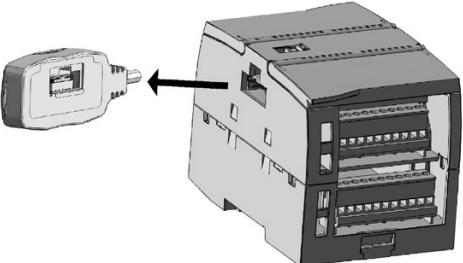
Tâche	Marche à suivre
	<ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Placez le connecteur femelle sur le connecteur de bus du côté gauche du module d'entrées-sorties. 3. Faites glisser l'extension en crochet du connecteur femelle dans le boîtier au niveau du connecteur de bus et appuyez légèrement vers le bas pour enclencher le crochet.
	<ol style="list-style-type: none"> 4. Fixez le connecteur en place : <ul style="list-style-type: none"> – Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties. – Faites glisser la languette complètement vers la gauche. <p>Pour enclencher le connecteur, vous devez faire glisser la languette jusqu'au bout vers la gauche. Le connecteur doit être verrouillé en place.</p>

Tableau 4- 14 Désinstallation du connecteur femelle du câble d'extension

Tâche	Marche à suivre
	<ol style="list-style-type: none"> 1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension. 2. Déverrouillez le connecteur : <ul style="list-style-type: none"> – Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties. – Appuyez légèrement et faites glisser la languette complètement vers la droite. 3. Soulevez légèrement le connecteur pour désenclencher l'extension en crochet. 4. Retirez le connecteur femelle.
	

Remarque

Installation du câble d'extension dans un environnement avec vibrations

Si le câble d'extension est raccordé à des modules qui bougent ou qui ne sont pas bien fixés, la connexion par encliquetage de l'extrémité mâle du câble peut progressivement se défaire.

Utilisez une attache de câble pour fixer l'extrémité mâle du câble au profilé support (ou à un autre emplacement) afin d'obtenir une décharge de traction supplémentaire.

Évitez d'exercer une force excessive lorsque vous tirez le câble pendant l'installation. Assurez-vous que la connexion câble-module est correctement positionnée une fois l'installation achevée.

4.3.8 Adaptateur TS Adapter (TeleService)

4.3.8.1 Connexion de l'adaptateur TeleService

Avant d'installer l'adaptateur TS (TeleService) Adapter IE Basic ou l'adaptateur TS (TeleService) Adapter IE Advanced, vous devez d'abord connecter l'adaptateur TS Adapter et un module TS.

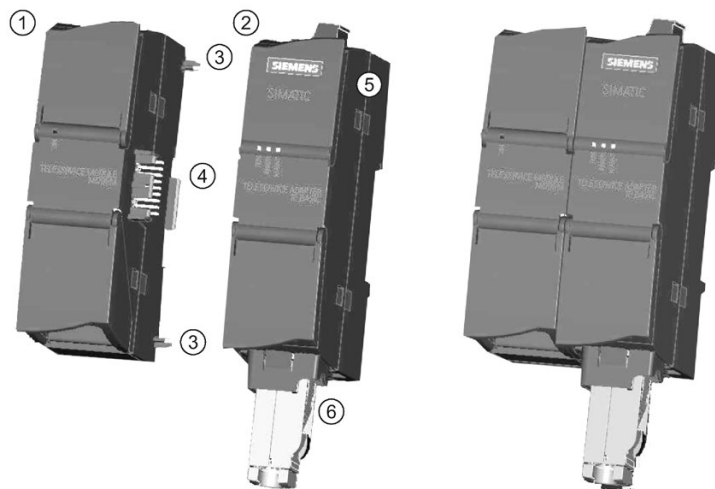
Modules TS disponibles :

- Module TS RS232
- Module TS Modem
- Module TS GSM
- Module TS RNIS

Remarque

Le module TS peut être endommagé si vous touchez les contacts de son connecteur enfichable ④.

Suivez les directives pour composants CSDE afin d'éviter que le module TS ne soit endommagé par des décharges électrostatiques. Avant de connecter un module TS et l'adaptateur TS Adapter, assurez-vous que tous deux sont inactifs.



- | | | | |
|---|-----------------------|---|------------------------------------|
| ① | Module TS | ④ | Connecteur enfichable du module TS |
| ② | Adaptateur TS Adapter | ⑤ | Ouverture impossible |
| ③ | Eléments | ⑥ | Port Ethernet |

Remarque

Avant de connecter un module TS et l'unité de base TS Adapter, assurez-vous que les broches de contact ④ ne dévient pas.

Lors de la connexion, veillez à ce que le connecteur mâle et les broches de guidage soient positionnés correctement.

Connectez uniquement un module TS à l'adaptateur TS Adapter. N'essayez pas de connecter par la force l'adaptateur TS Adapter à un autre dispositif, tel qu'une CPU S7-1200. Ne modifiez pas la structure mécanique du connecteur et n'ôtez et n'endommagez pas les broches de guidage.

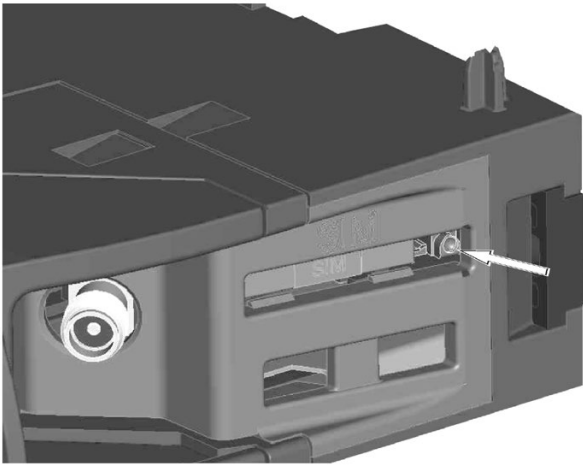
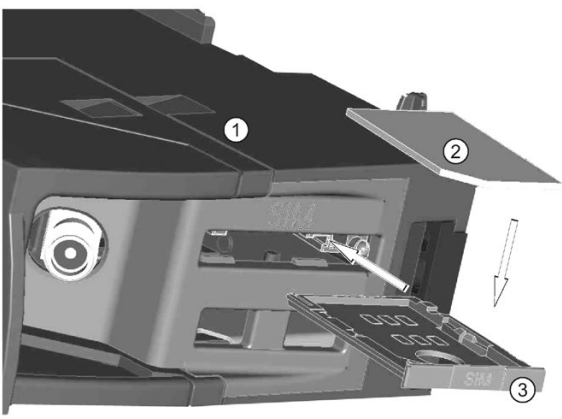
4.3.8.2 Installation de la carte SIM

Localisez l'emplacement de la carte SIM sur la face inférieure du TS module GSM.

Remarque

La carte SIM doit uniquement être débrochée ou enfichée lorsque le TS module GSM est hors tension.

Tableau 4- 15 Installation de la carte SIM

Tâche	Marche à suivre
	Utilisez un objet pointu pour enfoncer le bouton d'éjection du porte-carte SIM (dans le sens de la flèche) et retirez le porte-carte SIM.
	Placez la carte SIM dans le porte-carte SIM comme illustré et remplacez le porte-carte SIM dans son emplacement. ① TS Module GSM ② Carte SIM ③ Porte-carte SIM

Remarque

Assurez-vous que la carte SIM soit correctement positionnée dans le porte-carte. Sinon, la connexion entre la carte SIM et le module s'avèrera impossible et le bouton d'éjection ne permettra éventuellement plus d'éjecter le porte-carte.

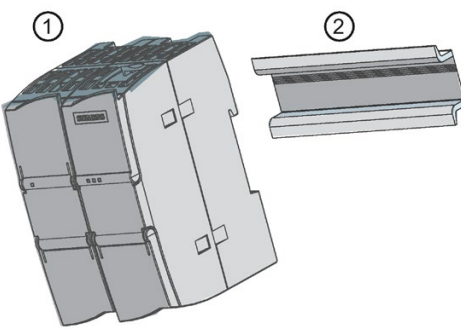
4.3.8.3 Installation de l'ensemble adaptateur TS sur un profilé support

Conditions requises : L'adaptateur TS Adapter doit être raccordé à un module TS et le profilé support doit être installé.

Remarque

Si vous montez l'ensemble TS verticalement ou dans un environnement à fortes vibrations, le module TS peut se détacher de l'adaptateur TS Adapter. Utilisez une équerre d'extrémité 8WA1808 sur le rail DIN pour être sûr que les modules restent connectés.

Tableau 4- 16 Montage et démontage de l'adaptateur TS Adapter

Tâche	Marche à suivre
	<p>Montage :</p> <ol style="list-style-type: none"> 1. Accrochez l'ensemble formé par l'adaptateur TS et le module TS ① sur le profilé support ②. 2. Faites pivoter l'ensemble vers le bas jusqu'à ce qu'il s'enclenche. 3. Repoussez la barrette de fixation sur chaque module pour fixer chaque module au profilé support. <p>Démontage :</p> <ol style="list-style-type: none"> 1. Retirez le câble analogique et le câble Ethernet de la face inférieure de l'adaptateur TS. 2. Mettez l'adaptateur TS hors tension. 3. Servez-vous d'un tournevis pour dégager les barrettes de fixation sur les deux modules. 4. Faites pivoter l'ensemble vers le haut pour le retirer du profilé support.

ATTENTION

Consignes de sécurité pour l'installation et la désinstallation du TS Adapter

Avant la mise hors tension, déconnectez la mise à la terre de l'adaptateur TS Adapter en retirant le câble analogique et le câble Ethernet. Le non-respect de cette précaution peut entraîner la mort, des blessures graves ou des dommages matériels importants en raison du fonctionnement imprévisible du matériel.

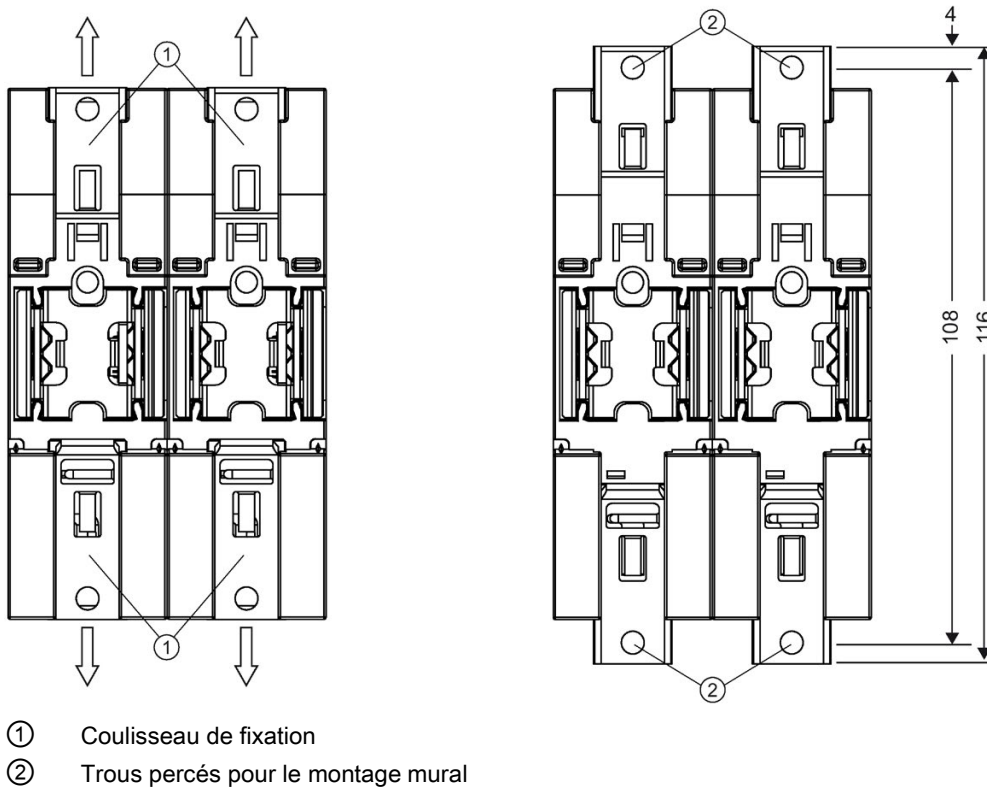
Observez toujours ces consignes de sécurité lorsque vous installez ou désinstallez l'adaptateur TS Adapter.

4.3.8.4 Montage de l'adaptateur TS sur un panneau

Conditions requises : L'adaptateur TS Adapter doit être raccordé à un module TS.

1. Déplacez le coulisseau de fixation ① situé sur la face arrière de l'adaptateur TS et du module TS dans le sens de la flèche jusqu'à ce qu'il s'enclenche.
2. Vissez l'adaptateur TS et le module TS sur le mur de montage prévu à la position repérée par ②.

La figure suivante montre l'adaptateur TS Adapter vu de derrière avec les coulisseaux de fixation ① dans les deux positions.



4.4 Conseils de câblage

Une mise à la terre et un câblage corrects de tout l'équipement électrique sont importants pour garantir un fonctionnement optimal de votre système et pour fournir une protection supplémentaire contre le bruit électrique pour votre application et le S7-1200. Vous trouverez les schémas de câblage du S7-1200 dans les caractéristiques techniques (Page 1429).

Conditions requises

Assurez-vous, avant de mettre à la terre ou de câbler tout appareil électrique, que cet appareil a été mis hors tension. Assurez-vous également que tout équipement associé a été mis hors tension.

Respectez toutes les normes électriques nationales et régionales en vigueur lors du câblage du S7-1200 et des matériels reliés. Installez et exploitez tous les équipements en conformité avec toutes les normes nationales et régionales en vigueur. Demandez l'aide des autorités locales pour déterminer les normes et les réglementations qui s'appliquent à votre cas particulier.

ATTENTION

L'installation ou le câblage du S7-1200 ou d'équipements reliés alors qu'ils sont sous tension peut provoquer un choc électrique ou un fonctionnement imprévisible du matériel.

La non-désactivation de l'alimentation complète du S7-1200 et des appareils qui y sont reliés pendant les procédures d'installation ou de désinstallation peut entraîner la mort, des blessures graves ou des dommages matériels importants en raison du choc électrique ou du fonctionnement imprévisible du matériel.

Prenez toujours toutes les mesures de sécurité nécessaires et assurez-vous que l'alimentation de l'automate S7-1200 est coupée avant de tenter d'installer ou de démonter le S7-1200 ou des matériels reliés.

Tenez toujours compte de la sécurité lorsque vous planifiez la mise à la terre et le câblage de votre système S7-1200. Les appareils de commande électroniques, tels que le S7-1200, peuvent subir des défaillances provoquant un fonctionnement inattendu des équipements commandés ou contrôlés. Vous devez, pour cette raison, toujours mettre en oeuvre des dispositifs de sécurité indépendants du S7-1200 pour éviter les blessures ou les dégâts matériels éventuels.

ATTENTION

Les appareils de commande peuvent présenter des défaillances dans des situations non sûres et provoquer un fonctionnement inattendu des appareils pilotés

pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Vous devez donc utiliser une fonction d'arrêt d'urgence, des dispositifs de sécurité électromécaniques ou d'autres sécurités redondantes qui soient indépendants du S7-1200.

Conseils pour l'isolation

Les barrières d'alimentation en courant alternatif du S7-1200 et les barrières d'E/S vers les circuits en courant alternatif ont été conçues et homologuées comme fournissant une séparation sûre entre les tensions de ligne courant alternatif et les circuits basse tension. Ces barrières incluent une isolation double ou renforcée, ou une isolation de base plus complémentaire, selon diverses normes. Les composants qui traversent ces barrières, tels que coupleurs optiques, condensateurs, transformateurs et relais, ont été homologués comme fournissant une séparation sûre. Seuls les circuits à valeur nominale de tension de ligne CA comprennent une isolation de sécurité envers les autres circuits. Seules les barrières d'isolation entre les circuits 24 V CC sont fonctionnelles et votre sécurité ne doit pas dépendre de ces barrières.

La sortie d'alimentation de capteur, les circuits de communication et les circuits logiques internes d'un S7-1200 à alimentation en courant alternatif incluse sont fournis comme TBTS (très basse tension de sécurité) selon EN 61131-2.

Pour conserver le caractère sûr des circuits basse tension du S7-1200, les connexions externes aux ports de communication, les circuits analogiques et tous les circuits d'alimentation et d'E/S 24 V CC nominal doivent être alimentés par des sources homologuées satisfaisant aux exigences TBTS, TBTP, classe 2, tension limitée ou courant limité, selon diverses normes.

ATTENTION

L'utilisation d'alimentations non isolées ou à une seule isolation pour alimenter les circuits basse tension à partir d'une ligne en courant alternatif peut provoquer l'apparition de tensions dangereuses sur des circuits censés être sûrs au toucher, tels que des circuits de communication ou un câblage de capteur basse tension.

De telles tensions élevées imprévisibles peuvent entraîner la mort, des blessures graves et des dommages matériels importants.

Utilisez exclusivement des convertisseurs de puissance haute tension-basse tension qui sont homologués comme sources de circuits à tension limitée à toucher sûr.

Conseils pour la mise à la terre du S7-1200

La meilleure façon de mettre à la terre votre application est de vous assurer que tous les branchements de commun et de terre de votre S7-1200 et des matériels reliés sont mis à la terre en un point unique. Ce point unique doit être relié directement à la masse de terre de votre système.

Tous les fils de terre doivent être aussi courts que possible et doivent avoir une section importante, par exemple 2 mm² (14 AWG).

Lorsque vous choisissez des points de référence à la terre, pensez également aux règles de sécurité correspondantes et à l'utilisation correcte des appareils de protection par coupure.

Conseils pour le câblage du S7-1200

Lorsque vous planifiez le câblage pour votre S7-1200, prévoyez un commutateur de coupure unique qui supprime simultanément le courant pour l'alimentation de la CPU S7-1200, pour tous les circuits d'entrée et pour tous les circuits de sortie. Fournissez une protection contre les surtensions, telle qu'un fusible ou un disjoncteur, afin de limiter les courants de défaut sur le câblage d'alimentation. Pensez également à fournir une protection supplémentaire en plaçant un fusible ou une autre limitation de courant dans chaque circuit de sortie.

Équipez les fils pouvant être menacés par la foudre d'une protection appropriée contre les surtensions. Pour plus d'informations, voir Immunité aux pointes de tension (Page 1429) dans les caractéristiques techniques générales.

Évitez de placer les câbles de signaux et de communication de faible tension dans la même goulotte que les câbles d'alimentation en courant alternatif et les câbles pour courant continu à commutation rapide et haute énergie. Posez toujours les fils par paires : conducteur neutre ou commun avec conducteur sous tension ou de signaux.

Utilisez le câble le plus court possible et assurez-vous qu'il est correctement dimensionné pour transporter le courant requis.

Le fil et le câble doivent avoir une température nominale de 30 °C plus élevée que la température ambiante autour du S7-1200 (par exemple, un minimum de 85 °C pour les conducteurs nominaux pour une température ambiante de 55 °C). Vous devez déterminer un autre type de câblage et des autres exigences matérielles à partir des valeurs nominales des circuits électriques et de votre environnement d'installation.

Utilisez des fils blindés pour assurer une protection optimale contre le bruit électrique. Typiquement, la mise à la terre du blindage au niveau du S7-1200 donne les meilleurs résultats. Vous devez mettre à la terre des blindages de câbles de communication avec les coques des connecteurs de communication du S7-1200 en utilisant des connecteurs qui s'enclenchent dans le blindage de câbles, ou en reliant les blindages de câbles de communication à une mise à la terre séparée. Vous devez mettre à la terre d'autres blindages de câbles en utilisant des bornes ou une bande cuivre autour du blindage pour fournir une connexion de zone de surface élevée au point de mise à la terre.

Lorsque vous câblez des circuits d'entrée alimentés par une source externe, incluez un dispositif de protection contre les surtensions dans ce circuit. Une protection externe n'est pas nécessaire pour les circuits alimentés par l'alimentation de capteur 24 V CC du S7-1200, car cette alimentation de capteur comprend déjà une limitation de courant.

Tous les modules S7-1200 comportent des connecteurs amovibles pour le câblage utilisateur. Pour éviter que les connexions ne se desserrent, assurez-vous que le connecteur est correctement enfoncé et que le fil est bien enfiché dans le connecteur.

Le S7-1200 comporte des barrières d'isolation à certains points pour empêcher des flux de courant indésirables dans votre installation. Tenez compte de ces barrières d'isolation lorsque vous concevez le câblage de votre système. Vous trouverez dans les caractéristiques techniques (Page 1504) des informations sur la quantité d'isolation fournie et sur l'emplacement de ces barrières d'isolation. Seuls les circuits à valeur nominale de tension de ligne CA comprennent une isolation de sécurité envers les autres circuits. Seules les barrières d'isolation entre les circuits 24 V CC sont fonctionnelles et votre sécurité ne doit pas dépendre de ces barrières.

4.4 Conseils de câblage

Le tableau ci-dessous contient un récapitulatif des règles de câblage pour les CPU S7-1200, SM et SB :

Tableau 4- 17 Règles de câblage pour CPU S7-1200, SM et SB

Règles de câblage pour ...	CPU et connecteur SM	Connecteur SB
Diamètres de conducteurs connectables pour conducteurs standard	2 mm ² à 0,3 mm ² (14 AWG à 22 AWG)	1,3 mm ² à 0,3 mm ² (16 AWG à 22 AWG)
Nombre de conducteurs par connexion	1 ou association de 2 conducteurs jusqu'à 2 mm ² (total)	1 ou association de 2 conducteurs jusqu'à 1,3 mm ² (total)
Longueur de dénudage	6,4 mm	6,3 à 7 mm
Couple de serrage* (maximum)	0,56 N-m (5 inch-pounds)	0,33 N-m (3 inch-pounds)
Outil	Tournevis à tête plate de 2,5 à 3,0 mm	Tournevis à tête plate de 2,0 à 2,5 mm

* Afin de ne pas endommager le connecteur, veillez à ne pas trop serrer les vis.

Remarque

Des embouts sur les fils toronnés empêchent que des fils nus ne causent un court-circuit. Des embouts plus longs que la longueur de dénudage recommandée doivent comporter des colliers isolants afin de prévenir des courts-circuits dus à un mouvement latéral des conducteurs. Les sections maximales recommandées pour les conducteurs dénudés s'appliquent également aux embouts.

Voir aussi

Caractéristiques techniques (Page 1429)

Conseils pour les charges de lampe

Les charges de lampe, y compris les charges de DEL, endommagent les contacts de relais en raison du fort courant de choc à l'activation. Ce courant de choc représente nominalement de 10 à 15 fois le courant de régime permanent pour une lampe au tungstène. Un relais d'interposition ou un limiteur de surtension remplaçable est donc recommandé pour des charges de lampe qui seront commutées de nombreuses fois pendant la durée de vie de l'application.

Conseils pour les charges inductives

Utilisez des circuits de protection avec charges inductives pour écrêter l'accroissement de la tension lorsqu'une sortie de commande est désactivée. Les circuits de protection par écrêtage protègent les sorties d'une défaillance prématurée due aux transitoires haute tension qui se produisent lors de l'interruption du flux de courant à travers une charge inductive.

En outre, ces circuits limitent les interférences électriques engendrées lors de la commutation de charges inductives. Les perturbations de haute fréquence induites par des charges inductives faiblement écrêtées peuvent perturber le fonctionnement de l'API. Placer un circuit de protection par écrêtage externe de façon à ce qu'il soit électriquement aux bornes de la charge et physiquement à proximité de la charge est la façon la plus efficace de réduire le bruit électrique.

Les sorties CC du S7-1200 comprennent des circuits de protection internes par écrêtage qui conviennent aux charges inductives dans la plupart des applications. Comme les contacts de sortie relais du S7-1200 peuvent servir pour commuter une charge en courant continu aussi bien qu'une charge en courant alternatif, aucune protection interne n'est fournie.

Une bonne solution de protection par écrêtage consiste à utiliser des contacteurs ou autres charges inductives pour lesquels le fabricant fournit des circuits de protection intégrés dans l'appareil de chargement ou comme accessoires en option. Cependant, certains circuits de protection par écrêtage fournis par le fabricant peuvent ne pas être adaptés à votre application. Un circuit de protection par écrêtage supplémentaire peut s'avérer nécessaire pour obtenir une réduction du bruit et une longévité des contacts optimales.

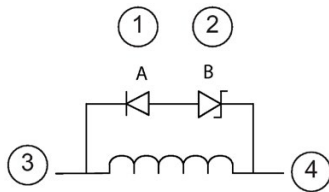
Pour les charges CA, une varistance à oxyde métallique (MOV) ou un autre dispositif de blocage de tension peut être utilisé avec un circuit RC parallèle. L'efficacité est toutefois moins grande que s'il est utilisé seul. Une protection par écrêtage MOV sans circuit RC parallèle entraîne souvent d'importantes perturbations de haute fréquence allant jusqu'à la tension de blocage.

Une surtension transitoire de coupure bien contrôlée n'aura pas de fréquence supérieure à 10 kHz, et elle sera de préférence inférieure à 1 kHz. La tension de crête pour les lignes CA doit être comprise dans une plage de +/- 1 200 V par rapport à la terre. La tension de crête négative pour les charges CC utilisant le circuit interne de protection par écrêtage de l'API sera à environ 40 V en dessous de la tension d'alimentation 24 V CC. Le circuit de protection externe devrait limiter la pointe de tension dans une plage de 36 V par rapport à la tension du réseau pour décharger le circuit de protection par écrêtage interne.

Remarque

L'efficacité d'un circuit de protection par écrêtage étant toutefois dépendante de l'application, vous devez vérifier qu'il convient à votre cas particulier. Assurez-vous que tous les éléments ont des valeurs nominales correctes et utilisez un oscilloscope pour observer les transitoires de coupure.

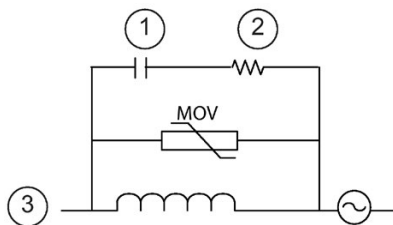
Circuit de protection par écrêtage typique pour courant continu ou sorties relais qui commutent des charges inductives en courant continu



- ① Diode 1N4001 ou équivalent
- ② Zener 8,2 V (sorties CC),
Zener 36 V (sorties relais)
- ③ Sortie
- ④ M, référence 24 V

Dans la plupart des applications, l'ajout d'une diode (A) en parallèle avec une charge inductive en courant continu suffit, mais si votre application requiert des temps de désactivation plus rapides, nous vous recommandons d'ajouter une diode Zener (B). Veillez à dimensionner votre diode Zener correctement pour la quantité de courant dans votre circuit de sortie.

Circuit de protection par écrêtage typique pour sorties relais qui commutent des charges inductives en courant alternatif



- ① Voir le tableau pour la valeur C
- ② Voir le tableau pour la valeur R
- ③ Sortie

Assurez-vous que la tension de fonctionnement de la varistance à oxyde métallique (MOV) est supérieure d'au moins 20 % à la tension de ligne nominale.

Choisissez des résistances non inductives d'impulsion nominale et des condensateurs recommandés pour les applications d'impulsion (habituellement un film métallique). Vérifiez que les composants sont conformes aux exigences de puissance moyenne, de puissance de crête et de tension de crête.

Si vous constituez votre propre circuit de protection par écrêtage, le tableau suivant vous suggère les valeurs de résistance et de condensateur pour une série de charges CA. Ces valeurs se basent sur des calculs effectués avec des paramètres idéaux pour ces composants. Dans le tableau, I rms fait référence au courant en régime permanent de la charge quand celle-ci est entièrement activée.

Tableau 4- 18 Valeurs de résistance du circuit de protection par écrêtage CA et de condensateur

Charge inductive			Valeurs de protection par écrêtage		
E rms	230 V CA	120 V CA	Résistance		Condensateur
Amps	VA	VA	Ω	W (puissance nominale)	nF
0,02	4,6	2,4	15 000	0,1	15
0,05	11,5	6	5 600	0,25	470
0,1	23	12	2 700	0,5	100
0,2	46	24	1 500	1	150
0,5	115	60	560	2,5	470
1	230	120	270	5	1 000
2	460	240	150	10	1 500

Conditions remplies par les valeurs du tableau :

Étape de transition de désactivation maximale < 500 V

Tension de crête de la résistance < 500 V

Tension de crête du condensateur < 1 250 V

Courant de protection par écrêtage < 8 % du courant de charge (50 Hz)

Courant de protection par écrêtage < 11 % du courant de charge (60 Hz)

Condensateur $dV/dt < 2 \text{ V}/\mu\text{s}$

Dissipation de l'impulsion du condensateur : $\int (dv/dt)^2 dt < 10\,000 \text{ V}^2/\mu\text{s}$

Fréquence de résonance < 300 Hz

Puissance de résistance pour une fréquence de commutation de 2 Hz maximum

Facteur de puissance de 0,3 supposé pour une charge inductive typique

Conseils pour les entrées et sorties différentielles

Les entrées et sorties différentielles ont un comportement différent de celui des entrées et sorties standard. Il y a deux broches par entrée ou sortie différentielle. Pour déterminer si une entrée ou une sortie différentielle est sous tension ou hors tension, vous devez mesurer la différence de tension entre ces deux broches.

Voir les caractéristiques détaillées de la CPU 1217C à l'annexe A (Page 1504).

5.1 Exécution du programme utilisateur

La CPU fournit les types suivants de blocs de code qui vous permettent de créer une structure efficace pour votre programme utilisateur :

- Les blocs d'organisation (OB) définissent la structure du programme. Certains OB ont des événements déclencheurs et un comportement prédéfinis, mais vous pouvez également créer des OB à événements déclencheurs personnalisés.
- Les fonctions (FC) et blocs fonctionnels (FB) contiennent le code de programme qui correspond à des tâches ou combinaisons de paramètres spécifiques. Chaque FC ou FB fournit un jeu de paramètres d'entrée et de sortie pour partager les données avec le bloc appelant. Un FB utilise également un bloc de données associé - appelé DB d'instance - pour conserver les valeurs de données pour cette instance d'appel de FB. Vous pouvez appeler un FB plusieurs fois et ce, avec un DB d'instance unique chaque fois. Utiliser des DB d'instance différents pour appeler le même FB n'affecte les valeurs de données dans aucun des DB d'instance.
- Les blocs de données (DB) mémorisent des données qui peuvent être utilisées par les blocs de programme.

L'exécution du programme utilisateur commence avec un ou plusieurs blocs d'organisation (OB) de démarrage optionnels qui s'exécutent une fois au passage à l'état MARCHE et qui sont suivis par un ou plusieurs OB du cycle de programme qui s'exécutent cycliquement. Vous pouvez également associer un OB à un événement d'alarme - événement standard ou événement d'erreur. Ces OB s'exécutent à chaque fois que l'événement standard ou d'erreur correspondant se produit.

Une fonction (FC) ou un bloc fonctionnel (FB) est un bloc de code qui peut être appelé dans un OB ou dans un autre FB ou une autre FC, en respectant les profondeurs d'imbrication suivantes :

- 16 en cas d'appel depuis l'OB de cycle de programme ou de démarrage
 - 6 en cas d'appel depuis un OB d'alarme associé à un événement
- Remarque : Les programmes de sécurité utilisent deux niveaux d'imbrication. Le programme utilisateur a donc une profondeur d'imbrication de quatre dans les programmes de sécurité.


Les FC ne sont pas associées à un bloc de données (DB) particulier. Les FB sont liés directement à un DB qu'ils utilisent pour transmettre les paramètres et stocker les valeurs intermédiaires et les résultats.

La taille du programme utilisateur, des données et de la configuration est limitée par la mémoire de chargement et la mémoire de travail disponibles dans la CPU. Il n'y a pas de limite spécifique au nombre de blocs de chaque type (OB, FC, FB ou DB). Cependant, le nombre total de blocs est limité à 1024.

Chaque cycle comprend l'écriture dans les sorties, la lecture des entrées, l'exécution des instructions du programme utilisateur et l'exécution du traitement d'arrière-plan. On parle parfois de cycle d'exploration.

Votre solution d'automatisation S7-1200 peut être composée d'un châssis central contenant la CPU S7-1200 et des modules supplémentaires. Le terme "châssis central" désigne soit le châssis ou le panneau d'installation de la CPU et les modules associés. Les modules (SM, SB, BB, CB, CM ou CP) sont détectés et pris en compte seulement lors de la mise sous tension.

- L'enfichage ou le débrogage d'un module dans le châssis central sous tension (à chaud) ne sont pas pris en charge. N'enfichez jamais de module dans le châssis central et n'en débroguez jamais lorsque la CPU est sous tension.

 ATTENTION
Consignes de sécurité pour l'enfichage ou le débrogage d'un module
La non-désactivation de l'alimentation complète de la CPU avant l'enfichage ou le débrogage d'un module (SM, SB, BB, CD, CM ou CP) du châssis central peut être à l'origine de dégâts ou d'un fonctionnement imprévisible pouvant entraîner la mort, des blessures graves et des dommages matériels importants.
Mettez toujours la CPU et le châssis central hors tension et prenez toutes les mesures de sécurité nécessaires avant d'insérer un module dans le châssis central ou d'en débroguer un.

- Vous pouvez insérer ou retirer une carte mémoire SIMATIC lorsque la CPU est sous tension. Toutefois, enficher ou retirer une carte mémoire lorsque la CPU est à l'état MARCHE provoque le passage de cette dernière à l'état ARRET.

IMPORTANT
Risques en cas de retrait d'une carte mémoire lorsque la CPU est à l'état MARCHE.
L'insertion ou le retrait d'une carte mémoire lorsque la CPU est à l'état MARCHE provoque le passage de cette dernière à l'état ARRET, ce qui peut provoquer des dommages aux équipements ou au processus piloté.
Lorsque vous insérez ou retirez une carte mémoire, la CPU passe immédiatement à l'état ARRET. Avant d'insérer ou de retirer une carte mémoire, assurez-vous toujours que la CPU ne pilote pas activement une machine ou un processus. Installez toujours un circuit d'arrêt d'urgence pour votre application ou votre processus.

- Si vous enfichez ou débroguez un module dans un châssis de périphérie décentralisée (AS-i, PROFINET ou PROFIBUS) alors que la CPU est à l'état MARCHE, la CPU génère une entrée dans la mémoire tampon de diagnostic, exécute l'OB de débrogage/enfichage de modules, le cas échéant, et reste à l'état MARCHE par défaut.

Mise à jour de la mémoire image et des mémoires images partielles

La CPU actualise les entrées/sorties TOR et analogiques locales en synchronisme avec le cycle à l'aide d'une zone de mémoire interne appelée mémoire image du processus. La mémoire image contient un cliché instantané des entrées et sorties physiques (de la CPU, du Signal Board et des modules d'entrées-sorties).

Vous pouvez configurer des entrées/sorties à mettre à jour dans la mémoire image avec le cycle ou lorsqu'un événement d'alarme spécifique se produit. Vous pouvez également configurer une entrée/sortie à exclure de l'actualisation de la mémoire image. Par exemple, votre processus peut ne requérir que certaines valeurs de données lorsqu'un événement telle qu'une alarme de processus se produit. En configurant l'actualisation de la mémoire image pour ces entrées/sorties à associer à une zone que vous affectez à un OB d'alarme de processus, vous évitez que la CPU ne mette à jour inutilement des valeurs de données à chaque cycle lorsque votre processus ne nécessite pas une mise à jour continue.

Pour l'entrée/sortie actualisée avec chaque cycle, la CPU exécute les tâches suivantes à chaque cycle d'exploration :

- La CPU écrit les sorties de la mémoire image des sorties dans les sorties physiques.
- La CPU lit les entrées physiques juste avant l'exécution du programme utilisateur et stocke leurs valeurs dans la mémoire image des entrées. Ainsi, ces valeurs restent cohérentes pendant toute l'exécution des instructions utilisateur.
- La CPU exécute la logique des instructions utilisateur et actualise les valeurs des sorties dans la mémoire image des sorties sans les écrire dans les sorties physiques réelles.

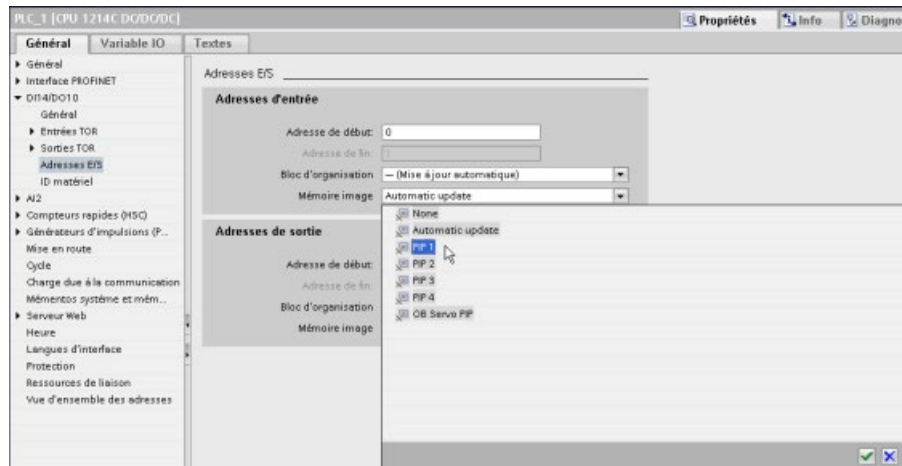
Ce procédé fournit une logique cohérente durant l'exécution des instructions utilisateur pour un cycle donné et empêche la gigue des sorties physiques dont l'état peut changer à de nombreuses reprises dans la mémoire image des sorties.

Pour vérifier si votre processus actualise les entrées/sorties automatiquement à chaque cycle, ou après le déclenchement d'alarmes, S7-1200 offre cinq mémoires images partielles. La première mémoire image partielle, PIP0, est affectée aux entrées/sorties à actualiser automatiquement à chaque cycle. C'est le paramètre par défaut. Vous pouvez utiliser les quatre mémoires restantes, PIP1, PIP2, PIP3 et PIP4, pour associer l'actualisation de la mémoire image des entrées/sorties à divers événements d'alarme. Vous affectez les entrées/sorties aux mémoires images partielles dans la Configuration des appareils et vous associez des mémoires images partielles à des événements d'alarme en créant des OB d'alarme (Page 195) ou en éditant les propriétés d'OB (Page 195).

Par défaut, lorsque vous insérez un module dans la vue des appareils, STEP 7 règle l'actualisation de la mémoire image des entrées/sorties de celui-ci sur "Actualisation automatique". Sur les entrées/sorties configurées pour une "Actualisation automatique", la CPU gère automatiquement l'échange de données entre le module et la zone de mémoire image pendant chaque cycle.

Procédez comme suit pour affecter des entrées ou sorties TOR ou analogiques à une mémoire image partielle ou pour en exclure de l'actualisation automatique de la mémoire image :

1. Affichez l'onglet Propriétés de l'appareil approprié dans la Configuration des appareils.
2. Affichez si nécessaire les détails sous "Général" pour localiser les entrées/sorties souhaitées.
3. Sélectionnez "Adresses E/S".
4. Ou sélectionnez un OB spécifique dans la liste déroulante du "Bloc d'organisation".
5. Dans la liste déroulante "Mémoire image", modifiez "Actualisation automatique" à "PIP1", "PIP2", "PIP3", "PIP4" ou "Aucune". Sélectionner "Aucune" signifie que vous pouvez uniquement lire et écrire directement des valeurs des entrées/sorties à l'aide d'instructions. Pour réintégrer les entrées et sorties dans l'actualisation automatique de la mémoire image, réactivez l'option "Actualisation automatique".



Vous pouvez lire directement des valeurs d'entrées physiques et écrire directement des valeurs de sorties physiques lors de l'exécution d'une instruction. Une lecture directe accède à l'état en cours de l'entrée physique et n'actualise pas la mémoire image des entrées, que l'entrée soit ou non configurée pour être stockée dans la mémoire image. Une écriture directe dans la sortie physique actualise à la fois la mémoire image des sorties (si la sortie est configurée pour être stockée dans la mémoire image) et la sortie physique. Ajoutez le suffixe ":P" à l'adresse d'entrée/sortie si vous voulez que le programme accède aux données d'E/S directement par l'entrée ou la sortie physique au lieu de la mémoire image.

Remarque

Utilisation de mémoires images partielles

Si vous affectez les entrées/sorties à l'une des mémoires images partielles PIP1 à PIP4 sans affecter un OB à celle-ci, la CPU n'actualise jamais la mémoire image des entrées/sorties concernées. Affecter des entrées/sorties à une PIP qui ne dispose pas d'une affectation OB correspondante est la même chose que d'affecter la mémoire image à "Aucune". Vous pouvez lire directement des valeurs d'entrées/de sorties physiques avec une instruction de lecture ou écrire directement des valeurs d'entrées/de sorties physiques avec une instruction d'écriture. La CPU n'actualise pas la mémoire image.

La CPU prend en charge la périphérie décentralisée pour les réseaux PROFINET, PROFIBUS et AS-i (Page 825).

5.1.1 Etats de fonctionnement de la CPU

La CPU a trois états de fonctionnement : l'état ARRET (STOP), l'état MISE EN ROUTE et l'état MARCHE (RUN). Des DEL d'état en face avant de la CPU signalent l'état de fonctionnement en cours.

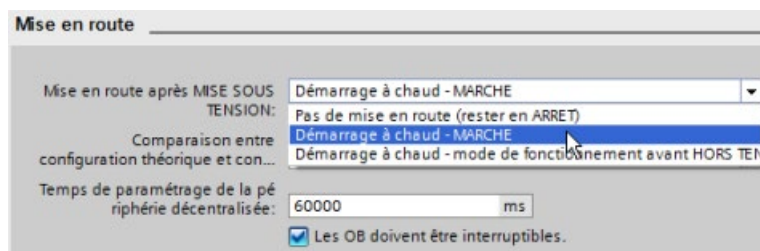
- A l'état ARRET, la CPU n'exécute pas le programme. Vous pouvez y charger un projet.
- A l'état MISE EN ROUTE, les OB de démarrage (s'il y en a) sont exécutés une fois. La CPU ne traite pas les événements d'alarme pendant la mise en route.
- A l'état MARCHE, les OB de cycle de programme s'exécutent de manière répétée. Des événements d'alarme peuvent survenir à tout moment à l'état MARCHE et peuvent déclencher l'exécution des OB d'alarme correspondants. Vous pouvez charger certaines parties d'un projet dans la CPU à l'état MARCHE (Page 1406).

La CPU accepte un démarrage à chaud pour le passage à l'état MARCHE. Le démarrage à chaud ne comprend pas d'effacement général de la mémoire. La CPU initialise toutes les données système et utilisateur non rémanentes lors d'un démarrage à chaud et conserve les valeurs de toutes les données utilisateur rémanentes.

Un effacement général efface toute la mémoire de travail, efface les zones de mémoire rémanentes et non rémanentes, copie la mémoire de chargement dans la mémoire de travail et définit les sorties selon la "Réaction à l'arrêt de la CPU" configurée. Un effacement général n'efface pas la mémoire tampon de diagnostic ni les valeurs sauvegardées de manière permanente de l'adresse IP.

Vous pouvez configurer le paramètre "Démarrage après mise sous tension" de la CPU. Ce paramètre de configuration apparaît dans "Configuration d'appareil" pour la CPU, sous "Mise en route". A la mise sous tension, la CPU exécute une séquence de vérifications du diagnostic de mise en route et effectue l'initialisation du système. Pendant l'initialisation du système, la CPU efface tous les mémentos (M) non rémanents et réinitialise le contenu de tous les DB non rémanents aux valeurs initiales provenant de la mémoire de chargement. La CPU conserve les mémentos (M) rémanents et le contenu des DB rémanents puis passe à l'état de fonctionnement approprié. La détection de certaines erreurs empêche le passage de la CPU à l'état MARCHE. La CPU accepte les configurations suivantes :

- Pas de mise en route (rester en ARRÊT)
- Démarrage à chaud - MARCHE
- Démarrage à chaud - mode de fonctionnement avant la mise hors tension



IMPORTANT

Des défauts réparables peuvent provoquer le passage de la CPU à l'état ARRÊT.

La CPU peut passer à l'état ARRÊT suite à des défauts réparables tels que :

- Défaillance d'un module d'entrées-sorties remplaçable
- Défauts temporaires tels que perturbation sur la ligne d'alimentation ou fonctionnement erratique à la mise sous tension

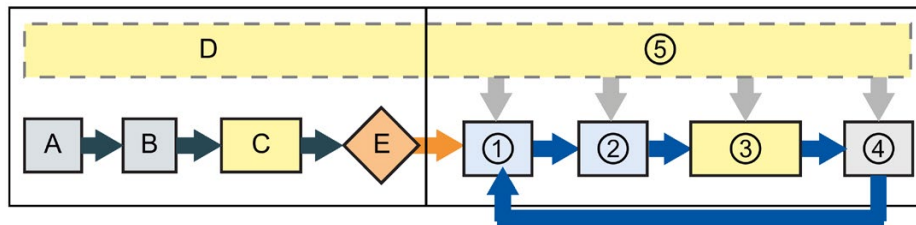
De telles situations peuvent entraîner des dommages matériels.

Si vous avez configuré "Démarrage à chaud - mode de fonctionnement avant la mise hors tension" pour la CPU, la CPU passe à l'état de fonctionnement précédant la coupure de courant ou le défaut. Si la CPU était à l'état ARRÊT au moment de la coupure de courant ou du défaut, elle passe à l'état ARRÊT lors de la mise sous tension. La CPU reste à l'état ARRÊT jusqu'à ce qu'elle reçoive une commande de passage à l'état MARCHE. Si la CPU était à l'état MARCHE au moment de la coupure de courant ou du défaut, elle passe à l'état MARCHE lors de la mise sous tension suivante. La CPU passe à l'état MARCHE à condition qu'elle ne détecte aucune erreur qui empêcherait le passage à l'état MARCHE.

Configurez "Démarrage à chaud - MARCHE" pour les CPU destinées à fonctionner indépendamment d'une liaison STEP 7. Ce mode de démarrage permet à la CPU de repasser à l'état MARCHE lors de la mise hors tension puis sous tension suivante.

Vous pouvez utiliser les commandes "ARRÊT" et "MARCHE" (Page 1392) des outils en ligne du logiciel de programmation pour changer l'état de fonctionnement en cours. Vous pouvez aussi insérer une instruction STP (Page 323) dans votre programme pour faire passer la CPU à l'état ARRÊT. Cette instruction vous permet d'arrêter l'exécution de votre programme en fonction de la logique.

- A l'état ARRÊT, la CPU traite toute demande de communication de manière appropriée et effectue un auto-diagnostic. Elle n'exécute pas le programme utilisateur. Il n'y a pas d'actualisation automatique de la mémoire image du processus.
- Dans les états MISE EN ROUTE et MARCHE, la CPU effectue les tâches présentées dans la figure suivante :



MISE EN ROUTE

- A Copie l'état des entrées physiques dans la mémoire image des entrées.
- B Initialise la zone de mémoire (image) des sorties Q avec zéro, la dernière valeur ou la valeur de remplacement configurée. Met les sorties PB, PN et AS-i à zéro
- C Initialise les mémentos et les blocs de données non rémanents à leur valeur initiale et valide les événements d'alarme cyclique et d'alarme horaire configurés.
Exécute les OB de démarrage.
- D Enregistre tous les événements d'alarme dans la file d'attente en vue de leur traitement après passage à l'état MARCHE.
- E Valide l'écriture de la mémoire image des sorties dans les sorties physiques.

MARCHE

- ① Ecrit la mémoire image des sorties dans les sorties physiques.
- ② Copie l'état des entrées physiques dans la mémoire image des entrées.
- ③ Exécute les OB de cycle de programme.
- ④ Réalise des tests d'auto-diagnostic.
- ⑤ Traite les alarmes et la communication à n'importe quel moment du cycle.

Remarque

La communication, y compris la communication IHM, ne peut pas interrompre les OB autres que les OB de cycle de programme.

Traitement de MISE EN ROUTE

A chaque fois que l'état de fonctionnement passe de ARRET à MARCHE, la CPU efface la mémoire image des entrées, initialise la mémoire image des sorties et traite les OB de démarrage. Tout accès en lecture à la mémoire image des entrées par des instructions dans les OB de démarrage lit zéro et non l'état en cours de l'entrée physique. Ainsi, pour lire l'état en cours de l'entrée physique pendant la mise en route, vous devez effectuer une lecture directe. Les OB de démarrage ainsi que tous les FB et FC associés sont ensuite exécutés. En présence de plus d'un OB de démarrage, la CPU exécute chaque OB par ordre de numéro d'OB, l'OB de plus petit numéro étant exécuté en premier.

Chaque OB de démarrage comprend des informations de démarrage qui vous aident à déterminer la validité des données rémanentes et l'horloge temps réel. Vous pouvez programmer dans les OB de démarrage des instructions qui examinent ces valeurs de démarrage afin de déclencher l'action appropriée. Les informations de démarrage suivantes sont prises en charge par les OB de démarrage :

Tableau 5- 1 Informations de démarrage prises en charge par les OB de démarrage

Entrée	Type de données	Description
LostRetentive	Bool	Ce bit est vrai si les zones de stockage de données rémanentes ont été perdues.
LostRTC	Bool	Ce bit est vrai si l'horloge temps réel a été perdue.

La CPU exécute également les tâches suivantes pendant le traitement de mise en route :

- Les alarmes sont mises en file d'attente mais ne sont pas traitées pendant la phase de mise en route.
- Aucune surveillance du temps de cycle n'est effectuée pendant la phase de mise en route.
- Il est possible de modifier la configuration de HSC (compteur rapide), PWM (modulation de largeur des impulsions) et des modules PtP (communication point à point) pendant la mise en route.
- Mais l'exécution réelle de HSC, PWM et des modules de communication point à point n'est possible qu'à l'état MARCHE.

Une fois l'exécution des OB de démarrage achevée, la CPU passe à l'état MARCHE et traite les tâches de commande lors d'un cycle continu.

5.1.2 Traitement du cycle à l'état MARCHÉ

A chaque cycle, la CPU écrit dans les sorties, lit les entrées, exécute le programme utilisateur, actualise les modules de communication et réagit aux événements d'alarme utilisateur et aux demandes de communication. Les demandes de communication sont traitées périodiquement pendant tout le cycle.

Ces actions (à l'exception des événements d'alarme utilisateur) sont assurées régulièrement et à la suite les unes des autres. Les événements d'alarme utilisateur activés sont gérés selon leur priorité dans l'ordre où ils se produisent. Pour les événements d'alarme, la CPU lit les entrées, exécute l'OB, puis écrit dans les sorties en utilisant la mémoire image partielle (MIP) associée, le cas échéant.

Le système garantit que le cycle s'achèvera au bout d'une durée appelée temps de cycle maximum. Dans le cas contraire, un événement d'erreur de temps est généré.

- Chaque cycle commence par l'extraction des valeurs en cours des sorties TOR et analogiques de la mémoire image et leur écriture dans les sorties physiques de la CPU, du SB et des SM configurées pour l'actualisation automatique des E/S (configuration par défaut). Lorsqu'on accède à une sortie physique par une instruction, la mémoire image des sorties et la sortie physique elle-même sont toutes deux actualisées.
- Chaque cycle se poursuit par la lecture des valeurs en cours des entrées TOR et analogiques dans la CPU, le SB et les SM configurées pour l'actualisation automatique des E/S (configuration par défaut), puis par l'écriture de ces valeurs dans la mémoire image des entrées. Lorsqu'on accède à une entrée physique par une instruction, l'instruction accède à la valeur de l'entrée physique mais la mémoire image des entrées n'est pas actualisée.
- Une fois les entrées lues, le programme utilisateur est exécuté de la première instruction jusqu'à l'instruction finale. Cela comprend tous les OB du cycle de programme avec leurs FC et FB associés. Les OB du cycle de programme sont exécutés par ordre de numéro d'OB, l'OB de plus petit numéro étant exécuté en premier.

Le traitement de la communication se produit périodiquement pendant tout le cycle, interrompant éventuellement l'exécution du programme utilisateur.

Les contrôles d'auto-diagnostic comprennent des contrôles périodiques du système ainsi que des contrôles de l'état des modules d'E/S.

Des alarmes peuvent se produire à tout endroit du cycle ; elles sont déclenchées sur événement. Lorsqu'un événement se produit, la CPU interrompt le cycle et appelle l'OB qui a été configuré pour traiter cet événement. Lorsque l'OB a fini de traiter l'événement, la CPU reprend l'exécution du programme utilisateur à l'endroit où elle s'était interrompue.

5.1.3 Blocs d'organisation (OB)

Les OB pilotent l'exécution du programme utilisateur. Des événements spécifiques dans la CPU déclenchent l'exécution d'un bloc d'organisation. Les OB ne peuvent pas s'appeler les uns les autres. Une FC ou un FB ne peut pas appeler un OB. Seul un événement tel qu'une alarme de diagnostic ou un intervalle de temps peut déclencher l'exécution d'un OB. La CPU gère les OB en fonction de leur classe de priorité respective, les OB de priorité supérieure étant exécutés avant les OB de priorité inférieure. La classe de priorité la plus faible est 1 (pour le cycle du programme principal) et la classe de priorité la plus élevée est 26.

5.1.3.1 OB de cycle de programme

Les OB du cycle de programme s'exécutent cycliquement lorsque la CPU est à l'état MARCHE. Le bloc principal du programme est un OB de cycle de programme. C'est là que vous placez les instructions qui commandent votre programme et où vous appelez d'autres blocs utilisateur. Vous pouvez avoir plusieurs OB de cycle de programme que la CPU exécute par ordre numérique. "Main" (OB 1) est le bloc par défaut.

Evénements cycle de programme

L'événement cycle de programme se produit une fois à chaque cycle. Pendant le cycle de programme, la CPU écrit dans les sorties, lit les entrées et exécute les OB de cycle de programme. L'événement cycle de programme est obligatoire et est toujours activé. Vous pouvez n'avoir aucun OB de cycle de programme ou avoir plusieurs OB sélectionnés pour l'événement cycle de programme. Une fois que l'événement cycle de programme s'est produit, la CPU exécute l'OB de cycle de programme ayant le plus petit numéro (généralement "Main" OB 1). La CPU exécute les autres OB de cycle de programme de façon séquentielle (par ordre numérique) à l'intérieur du cycle de programme. L'exécution du programme est cyclique de sorte que l'événement cycle de programme se produit aux instants suivants :

- Lorsque l'exécution du dernier OB de démarrage s'achève
- Lorsque l'exécution du dernier OB de cycle de programme s'achève

Tableau 5-2 Informations de déclenchement pour un OB de cycle de programme

Entrée	Type de données	Description
Initial_Call	Bool	Vrai pour l'appel initial de l'OB
Remanence	Bool	Vrai si des données rémanentes sont disponibles

5.1.3.2 OB de démarrage

Les OB de démarrage s'exécutent une fois lorsque la CPU passe de l'état ARRET à l'état MARCHE, y compris lors de la mise sous tension en mode MARCHE et en transitions commandées de ARRET à MARCHE. Une fois les OB de démarrage achevés, le "Cycle de programme" principal commence son exécution.

Evénements de démarrage

L'événement démarrage se produit une fois lors d'une transition de ARRET à MARCHE et provoque l'exécution des OB de démarrage par la CPU. Vous pouvez configurer plusieurs OB pour l'événement démarrage. Les OB de démarrage s'exécutent par ordre numérique.

Tableau 5- 3 Information de déclenchement pour un OB de démarrage

Entrée	Type de données	Description
LostRetentive	Bool	Vrai si des données rémanentes sont perdues
LostRTC	Bool	Vrai si date et heure sont perdues

5.1.3.3 OB d'alarme temporisée

Les OB d'alarme temporisée s'exécutent après un temps de retard que vous avez configuré.

Evénements alarme temporisée

Vous pouvez configurer des événements alarme temporisée de sorte qu'ils se produisent à l'expiration d'un temps de retard indiqué. Vous définissez le temps de retard au moyen de l'instruction SRT_DINT. Les événements alarme temporisée interrompent le cycle de programme afin d'exécuter l'OB d'alarme temporisée correspondant. Vous ne pouvez associer qu'un seul OB d'alarme temporisée à un événement alarme temporisée. La CPU accepte quatre événements alarme temporisée.

Tableau 5- 4 Information de déclenchement pour un OB d'alarme temporisée

Entrée	Type de données	Description
Sign	Word	Identificateur transmis pour déclencher l'appel de SRT_DINT

5.1.3.4 OB d'alarme cyclique

Les OB d'alarme cyclique s'exécutent à un rythme indiqué. Vous pouvez configurer jusqu'à 4 événements alarme cyclique, un OB correspondant à chaque événement alarme cyclique.

Événements alarme cyclique

Les événements alarme cyclique vous permettent de configurer l'exécution d'un OB d'alarme avec une périodicité configurée. Vous configurez la périodicité initiale lors de la création de l'OB d'alarme cyclique. Un événement alarme cyclique interrompt le cycle de programme et exécute l'OB d'alarme cyclique correspondant. Notez que l'événement alarme cyclique est dans une classe de priorité supérieure à celle de l'événement cycle de programme.

Vous ne pouvez associer qu'un seul OB d'alarme cyclique à un événement alarme cyclique.

Vous pouvez affecter un déphasage à chaque alarme cyclique de sorte que l'exécution d'alarmes cycliques peut être décalée de la valeur du déphasage. Par exemple, si vous avez un événement alarme cyclique 5 ms et un événement alarme cyclique 10 ms, les deux événements se produisent au même moment toutes les dix ms. Si vous déphasez l'événement 5 ms de 1 à 4 ms et l'événement 10 ms de 0 ms, les deux événements ne se produisent pas au même moment.

Le déphasage par défaut est 0. Pour modifier le déphasage initial ou la périodicité d'un événement contextuel, suivez les étapes suivantes :

1. Faites un clic droit sur l'OB d'alarme cyclique dans l'arborescence du projet.
2. Sélectionnez "Propriétés" dans le menu contextuel.
3. Cliquez sur "Alarme cyclique" dans la boîte de dialogue "Alarme cyclique [OB 30]" et saisissez les nouvelles valeurs initiales.

Le décalage de phases maximal est la plus petite des deux valeurs suivantes : 6 000 ms (6 secondes) ou temps de cycle imparti.

Vous pouvez également connaître et modifier la périodicité et le déphasage depuis votre programme à l'aide des instructions Interroger alarme cyclique (QRY_CINT) et Régler alarme cyclique (SET_CINT). Les valeurs de périodicité et de déphasage définies à l'aide de l'instruction SET_CINT ne perdurent pas en cas de mise hors tension puis sous tension ou de passage à l'état ARRET ; la périodicité et le déphasage reprennent leur valeur initiale après une mise hors tension puis sous tension ou un passage à l'état ARRET. La CPU prend en charge un total de quatre événements alarme cyclique.

5.1.3.5 OB d'alarme de processus

Les OB d'alarme de processus s'exécutent lorsque l'événement d'alarme de processus correspondant se produit. Un OB d'alarme de processus interrompt l'exécution cyclique normale du programme en réaction à un signal provenant d'un événement du processus.

Evénements alarme de processus

Un changement dans le matériel, par exemple un front montant ou descendant sur une entrée ou un événement HSC (compteur rapide), déclenche les événements alarme de processus. Le S7-1200 prend en charge un OB d'alarme pour chaque événement alarme de processus. Vous validez les événements alarme de processus dans la configuration des appareils et vous affectez un OB à un événement dans la configuration des appareils ou au moyen d'une instruction ATTACH dans le programme utilisateur. La CPU accepte plusieurs événements alarme de processus. Le modèle de CPU et le nombre d'entrées déterminent les événements exacts disponibles.

Les limites sur les événements alarme de processus sont définies comme suit :

Fronts :

- Événements front montant : 16 au maximum
- Événements front descendant : 16 au maximum

Evénements HSC :

- CV=PV : 6 au maximum
- Inversion du sens : 6 au maximum
- Réinitialisation externe : 6 au maximum

Tableau 5- 5 Information de déclenchement d'un OB d'alarme matérielle

Entrée	Type de données	Description
LADDR	HW_IO	Identification matérielle du module qui a déclenché l'alarme matérielle
USI	WORD	Identification pour de futures extensions (16#0001 to 16#FFFF), réservé pour une utilisation future
IChannel	USINT	Numéro de la voie qui a déclenché l'alarme
EventType	BYTE	Identification du type d'évènement spécifique au module associé à l'évènement déclenchant l'alarme, par exemple, un front montant ou descendant.

Les bits dans EventType dépendent du module déclencheur comme décrit ci-dessous :

Modules / sous-modules	Valeur	Évènement de processus
Périphérie intégrée sur CPU ou SB	16#0	Front montant
	16#1	Front descendant
HSC	16#0	HSC CV=RV1
	16#1	HSC changement de direction
	16#2	Réinitialisation HSC
	16#3	HSC CV=RV2

5.1.3.6 OB d'erreur de temps

S'il est configuré, l'OB d'erreur de temps (OB 80) s'exécute lorsque le cycle dépasse le temps de cycle maximum ou qu'un événement erreur de temps se produit. S'il est déclenché, il s'exécute en interrompant l'exécution normale du programme cyclique ou tout autre OB d'évènement.

L'apparition de l'un de ces événements génère une entrée décrivant l'évènement dans la mémoire tampon de diagnostic. L'entrée de mémoire tampon de diagnostic est générée indépendamment de l'existence de l'OB d'erreur de temps.

Evénements erreur de temps

L'apparition d'une quelconque des différentes situations d'erreur de temps provoque un événement erreur de temps :

- Le cycle dépasse le temps de cycle maximum.
L'erreur "Temps de cycle maximum dépassé" se produit si le cycle du programme ne s'achève pas pendant le temps de cycle maximum défini. Voir la rubrique "Surveillance et configuration du temps de cycle (Page 111)" pour plus d'informations sur la condition de temps de cycle maximum, la procédure de configuration du temps de cycle maximum dans les propriétés de la CPU et la procédure de réinitialisation de la temporisation du cycle.
- La CPU ne peut pas démarrer l'OB demandé parce qu'une deuxième alarme (cyclique ou horaire) a été démarrée avant que la CPU n'ait fini d'exécuter le premier OB d'alarme.
- Débordement de file d'attente
L'erreur "Débordement de file d'attente" apparaît si les alarmes se produisent plus vite que la CPU ne peut les traiter. La CPU limite le nombre d'événements en attente (dans une file) grâce à une file d'attente différente pour chaque type d'évènement. Si un événement se produit alors que la file d'attente correspondante est pleine, la CPU génère un événement erreur de temps.

Tous les événements erreur de temps déclenchent l'exécution de l'OB d'erreur de temps s'il existe. Si l'OB d'erreur de temps n'existe pas, la CPU passe à l'état ARRÊT.

Le programme utilisateur peut allonger le temps d'exécution du cycle de programme jusqu'à dix fois le temps de cycle maximum configuré en exécutant l'instruction RE_TRIGR (Page 321) pour redémarrer la surveillance du temps de cycle. Toutefois, si deux erreurs "Temps de cycle maximum dépassé" se produisent pendant le même cycle de programme sans réinitialisation de la temporisation du cycle, la CPU passe à l'état ARRET que l'OB d'erreur de temps existe ou non. Référez-vous au paragraphe "Surveillance du temps de cycle" du "Manuel système S7-1200" (Page 111).

L'OB d'erreur de temps comprend des informations de déclenchement qui vous aident à déterminer quel événement et quel OB ont généré l'erreur de temps. Vous pouvez programmer dans l'OB des instructions pour examiner ces valeurs de déclenchement et prendre les mesures appropriées.

Tableau 5- 6 Informations de déclenchement pour l'OB d'erreur de temps (OB 80)

Entrée	Type de données	Description
fault_id	BYTE	16#01 : temps de cycle maximum dépassé 16#02 : impossible de démarrer l'OB requis 16#07 et 16#09 : débordement de file d'attente
csg_OBnr	OB_ANY	Numéro de l'OB en cours d'exécution lorsque l'erreur s'est produite
csg_prio	UINT	Priorité de l'OB à l'origine de l'erreur

Pour inclure un OB d'erreur de temps dans votre projet, vous devez ajouter une alarme d'erreur de temps en double-cliquant sur "Ajouter nouveau bloc" sous "Blocs de programme" dans l'arborescence, puis en choisissant "Bloc d'organisation" et enfin "Erreur de temps".

La priorité pour une nouvelle CPU V4.0 est fixée à 22. Si vous remplacez une CPU V3.0 par une CPU V4.0 (Page 1665), la priorité est 26, c'est-à-dire la priorité en vigueur pour V3.0. Dans tous les cas, le champ de priorité peut être édité et vous pouvez fixer la priorité à une valeur comprise entre 22 et 26.

5.1.3.7 OB d'alarme de diagnostic

L'OB d'alarme de diagnostic s'exécute lorsque la CPU détecte une erreur de diagnostic ou si un module apte au diagnostic identifie une erreur et que vous avez activé l'alarme de diagnostic pour ce module. L'OB d'alarme de diagnostic interrompt l'exécution cyclique normale du programme. Vous pouvez inclure une instruction STP dans l'OB d'alarme de diagnostic pour faire passer la CPU à l'état ARRET si vous souhaitez que votre CPU passe à l'état ARRET lors de la réception de ce type d'erreur.

Si vous n'incluez pas d'OB d'alarme de diagnostic dans votre programme, la CPU ne tient pas compte de l'erreur et reste à l'état MARCHE.

Événements erreur de diagnostic

Certains appareils analogiques (locaux), PROFINET, PROFIBUS et TOR (locaux) sont capables de détecter et de signaler des erreurs de diagnostic. L'apparition ou l'élimination d'une quelconque des différentes situations d'erreur de diagnostic provoque un événement erreur de diagnostic. Les erreurs de diagnostic suivantes sont prises en charge :

- Pas de courant utilisateur
- Limite supérieure dépassée
- Limite inférieure dépassée
- Rupture de fil
- Court-circuit

Les événements erreur de diagnostic déclenchent l'exécution de l'OB d'alarme de diagnostic (OB 82) s'il existe. S'il n'existe pas, la CPU ne tient pas compte de l'erreur.

Pour inclure un OB d'alarme de diagnostic dans votre projet, vous devez ajouter une alarme de diagnostic en double-cliquant sur "Ajouter nouveau bloc" sous "Blocs de programme" dans l'arborescence, puis en choisissant "Bloc d'organisation" et enfin "Alarme de diagnostic".

Remarque

Erreurs de diagnostic pour les appareils analogiques locaux multivoies (E/S, RTD et Thermocouple)

L'OB d'alarme de diagnostic ne peut traiter qu'une erreur de diagnostic de voie à la fois.

Si deux voies d'un appareil multivoie présentent une erreur, la deuxième erreur ne déclenche l'OB d'alarme de diagnostic que dans les conditions suivantes : la première erreur de voie est corrigée, l'exécution de l'OB d'alarme de diagnostic déclenchée par la première erreur est achevée et la deuxième erreur persiste.

L'OB d'alarme de diagnostic comprend des informations de déclenchement qui vous aident à déterminer si l'événement est dû à l'apparition ou à l'élimination d'une erreur, ainsi que l'appareil et la voie ayant signalé l'erreur. Vous pouvez programmer dans l'OB d'alarme de diagnostic des instructions pour examiner ces valeurs de déclenchement et prendre les mesures appropriées.

Remarque

L'information de déclenchement de l'OB d'alarme de diagnostic fait référence à l'ensemble du sous-module si aucun événement de diagnostic n'est en attente

Dans la version V3.0, l'information de déclenchement pour un événement d'erreur de diagnostic disparaissant indiquait toujours l'origine de l'événement. Dans la version V4.0, l'information de déclenchement fait référence à l'ensemble du sous-module (16#8000) s'il n'y a plus de diagnostic en attente pour le sous-module après un événement disparaissant, même si l'événement à l'origine de l'OB était une voie spécifique.

Par exemple, si une rupture de fil déclenche un événement de diagnostic sur la voie 2, que le défaut est ensuite corrigé et l'événement de diagnostic effacé, l'information de déclenchement ne fera pas référence à la voie 2, mais au sous-module (16#8000).

Tableau 5- 7 Informations de déclenchement pour l'OB d'alarme de diagnostic

Entrée	Type de données	Description
IOWstate	WORD	Etat E/S de l'appareil : <ul style="list-style-type: none"> • Bit 0 = 1 si la configuration est correcte et = 0 si la configuration n'est plus correcte. • Bit 4 = 1 si une erreur est présente (rupture de fil, par exemple). Bit 4 = 0 en l'absence d'erreur. • Bit 5 = 1 si la configuration n'est pas correcte et = 0 si la configuration est à nouveau correcte. • Bit 7 = 1 si une erreur d'accès à la périphérie s'est produite. Reportez-vous à LADDR qui contient l'identificateur matériel de l'E/S en erreur d'accès. Bit 6 = 0 en l'absence d'erreur.
LADDR	HW_ANY	Identificateur matériel de l'appareil ou de l'unité fonctionnelle ayant signalé l'erreur ¹
Channel	UINT	Numéro de voie
MultiError	BOOL	VRAI en présence de plusieurs erreurs

¹ L'entrée LADDR contient l'identificateur matériel de l'appareil ou de l'unité fonctionnelle qui a renvoyé l'erreur. L'identificateur matériel est affecté automatiquement lorsque des composants sont insérés dans la vue des appareils ou du réseau et il apparaît dans l'onglet Constantes des variables API. Un nom est également défini automatiquement pour l'identificateur matériel. Ces entrées dans l'onglet Constantes des variables API ne peuvent pas être modifiées.

5.1.3.8 OB de débrogage/enfichage de modules

L'OB de débrogage/enfichage de modules s'exécute lorsque le module ou sous-module d'une périphérie décentralisée configurée et non désactivée (PROFIBUS, PROFINET, AS-i) génère un événement lié au débrogage ou à l'enfichage d'un module.

Evénement de débrogage/enfichage de modules

Les conditions suivantes génèrent un événement débrogage/enfichage de modules :

- Quelqu'un a débrogé ou enfiché un module configuré.
- Un module configuré n'existe pas physiquement dans le châssis d'extension.
- Un module incompatible se trouve dans un châssis d'extension qui ne correspond pas au module configuré.
- Un module compatible avec un module configuré se trouve dans un châssis d'extension, mais la configuration n'accepte pas de remplacement.
- Un module ou un sous-module comporte des erreurs de paramétrage

Si vous n'avez pas programmé cet OB, la CPU reste à l'état MARCHE lorsque l'une de ces situations se produit avec un module de périphérie décentralisée configuré et non désactivé.

Que vous ayez programmé ou non cet OB, la CPU passe à l'état ARRET lorsque l'une de ces situations se produit avec un module dans le châssis central.

Tableau 5- 8 Information de déclenchement pour OB de débrogage/enfichage de modules

Entrée	Type de données	Description
LADDR	HW_IO	Identificateur matériel
Event_Class	Byte	16#38: module enfiché 16#29 : module débrogé
Fault_ID	Byte	Code d'erreur

5.1.3.9 OB de défaillance du châssis ou de la station

L'OB "Défaillance du châssis ou de la station" s'exécute lorsque la CPU détecte la défaillance ou la perte de communication d'un châssis ou d'une station décentralisée.

Evénement défaillance du châssis ou de la station

La CPU génère un événement défaillance du châssis ou de la station lorsqu'elle détecte l'une des situations suivantes :

- La défaillance d'un système maître DP ou d'un système PROFINET IO (dans le cas soit d'un événement apparaissant soit d'un événement disparaissant)
- La défaillance d'un esclave DP ou d'un périphérique IO (dans le cas soit d'un événement apparaissant soit d'un événement disparaissant)
- La défaillance de certains des sous-modules d'un I-device PROFINET

Si vous n'avez pas programmé cet OB, la CPU passe à l'état MARCHE lorsque l'une de ces situations se produit.

Tableau 5- 9 Informations de déclenchement pour l'OB de défaillance du châssis ou de la station

Entrée	Type de données	Description
LADDR	HW_IO	Identificateur matériel
Event_Class	Byte	16#38 : Événement disparaissant 16#39 : Événement apparaissant
Fault_ID	Byte	Identificateur du défaut

5.1.3.10 OB d'alarme horaire

Les OB d'alarme horaire s'exécutent sur la base de conditions d'horloge configurées. La CPU prend en charge deux OB d'alarme horaire.

Evénements alarme horaire

Vous pouvez configurer un événement alarme horaire afin qu'il se produise une fois à la date et à l'heure indiquées ou cycliquement selon l'une des périodicités suivantes :

- Toutes les minutes : l'alarme se produit toutes les minutes.
- Toutes les heures : l'alarme se produit toutes les heures.
- Journalière : l'alarme se produit chaque jour à une heure précise (heure et minute).
- Hebdomadaire : l'alarme se produit chaque semaine à une heure et un jour de la semaine indiqués (par exemple, chaque mardi à 16h30).
- Mensuelle : l'alarme se produit tous les mois à une heure et un jour du mois indiqués. Le numéro du jour doit être compris entre 1 et 28 inclus.
- A la fin du mois : l'alarme se produit le dernier jour de chaque mois à l'heure indiquée.
- Annuelle : l'alarme se produit tous les ans, à la date indiquée (mois et jour). Vous ne pouvez pas indiquer le 29 février comme date.

Tableau 5- 10 Informations de déclenchement pour un OB d'alarme horaire

Entrée	Type de données	Description
CaughtUp	Bool	L'appel de l'OB est rattrapé car l'heure a été avancée.
SecondTimes	Bool	L'appel de l'OB est lancé une deuxième fois, car l'heure a été retardée.

5.1.3.11 OB d'état

Les OB d'état s'exécutent si un esclave DPV1 ou PNIO déclenche une alarme d'état. Cela peut se produire si un composant (module ou châssis) d'un esclave DPV1 ou PNIO change son état de fonctionnement, par exemple de MARCHE à ARRET.

Evénements état

Pour plus d'informations sur les événements qui déclenchent une alarme d'état, consultez la documentation du fabricant des esclaves DPV1 ou PNIO.

Tableau 5- 11 Informations de déclenchement pour un OB d'état

Entrée	Type de données	Description
LADDR	HW_IO	Identificateur matériel
Slot	UInt	Numéro d'emplacement
Spécifier	Word	Spécificateur d'alarme

5.1.3.12 OB de mise à jour

L'OB de mise à jour s'exécute si un esclave DPV1 ou PNIO déclenche une alarme de mise à jour.

Événements mise à jour

Pour plus d'informations sur les événements qui déclenchent une alarme de mise à jour, consultez la documentation du fabricant des esclaves DPV1 ou PNIO.

Tableau 5- 12 Informations de déclenchement pour l'OB de mise à jour

Entrée	Type de données	Description
LADDR	HW_IO	Identificateur matériel
Slot	UInt	Numéro d'emplacement
Specifier	Word	Spécificateur d'alarme

5.1.3.13 OB de profil

Les OB de profil s'exécutent si un esclave DPV1 ou PNIO déclenche une alarme spécifique au profil.

Événements profil

Pour plus d'informations sur les événements qui déclenchent une alarme de profil, consultez la documentation du fabricant des esclaves DPV1 ou PNIO.

Tableau 5- 13 Informations de déclenchement pour un OB de profil

Entrée	Type de données	Description
LADDR	HW_IO	Identificateur matériel
Slot	UInt	Numéro d'emplacement
Specifier	Word	Spécificateur d'alarme

5.1.3.14 OB MC-Servo et MC-Interpolator

Quand vous créez un objet technologique et que vous configurez la "Connexion de l'entraînement analogique" ou "PROFIdrive » comme interface d'entraînement, STEP 7 crée automatiquement les blocs d'organisation MC-Servo et MC-Interpolator protégés en écriture. Vous n'avez pas besoin d'éditer des propriétés OB ou de créer cet OB directement. La CPU utilise ces blocs d'organisation pour la régulation. Pour plus de détails, reportez-vous au système d'information de STEP 7.

5.1.3.15 MC-PreServo

Vous pouvez programmer l'OB MC-PreServo afin qu'il contienne la logique que le programme STEP 7 exécutera juste avant l'exécution de l'OB MC-Servo.

Événements MC-PreServo

Le bloc d'organisation MC-PreServo permet de lire l'information de cycle d'application configurée en microsecondes.

Tableau 5- 14 Information de déclenchement pour l'OB MC-PreServo

Entrée	Type de données	Description
Initial_Call	BOOL	TRUE signale le premier appel de cet OB lors du passage de ARRÊT à MARCHE.
PIP_Input	BOOL	TRUE indique que la mémoire image des entrées associée est à jour.
PIP_Output	BOOL	TRUE signale que la CPU a transféré la mémoire image des sorties associée dans les sorties dans les temps après le dernier cycle.
IO_System	USINT	Numéro du système périphérique décentralisé qui déclenche l'alarme
Event_Count	INT	n : nombre de cycles perdus -1 : nombre de cycles perdus inconnu (par exemple, parce que le cycle a changé)
Synchronous	BOOL	Réservé
CycleTime	UDINT	Affichage du cycle d'application configuré pour l'OB MC-Servo en microsecondes

5.1.3.16 MC-PostServo

Vous pouvez programmer l'OB MC-PostServo afin qu'il contienne la logique que le programme STEP 7 exécutera juste après l'exécution de l'OB MC-Servo.

Événements MC-PostServo

Le bloc d'organisation MC-PreServo permet de lire l'information de cycle d'application configurée en microsecondes.

Tableau 5- 15 Information de déclenchement pour l'OB MC-PostServo

Entrée	Type de données	Description
Initial_Call	BOOL	TRUE signale le premier appel de cet OB lors du passage de ARRÊT à MARCHE.
PIP_Input	BOOL	TRUE indique que la mémoire image des entrées associée est à jour.
PIP_Output	BOOL	TRUE signale que la CPU a transféré la mémoire image des sorties associée dans les sorties dans les temps après le dernier cycle.
IO_System	USINT	Numéro du système périphérique décentralisé qui déclenche l'alarme
Event_Count	INT	n : nombre de cycles perdus -1 : nombre de cycles perdus inconnu (par exemple, parce que le cycle a changé)
Synchronous	BOOL	Réservé
CycleTime	UDINT	Affichage du cycle d'application configuré pour l'OB MC-Servo en microsecondes

5.1.3.17 Priorités d'exécution et mise en file d'attente des événements

Le traitement de la CPU est commandé par des événements. Un événement déclenche un OB d'alarme qu'il faut exécuter. Vous pouvez définir l'OB d'alarme pour un événement lors de la création du bloc, lors de la configuration de l'appareil ou à l'aide d'une opération ATTACH ou DETACH. Certains événements se produisent sur une base régulière, tels les événements cycle de programme ou les événements alarme cyclique. D'autres événements ne se produisent qu'une fois, tels l'événement démarrage et les événements alarme temporisée. Certains événements se produisent lorsque le matériel déclenche un événement, par exemple un événement front sur une entrée ou un événement compteur rapide. Des événements tels que l'erreur de diagnostic et l'erreur de temps ne se produisent que lorsqu'une erreur se produit. Les priorités et files d'attente d'événements sont utilisées pour déterminer l'ordre de traitement des OB d'alarme d'événement.

La CPU traite des événements par ordre de priorité dans lequel 1 est la priorité la plus faible et 26 est la priorité la plus élevée. Avant le V4.0 de la CPU S7-1200, chaque type d'OB appartenait à une classe de priorité fixe (1 à 26). A partir de la version V4.0, vous pouvez affecter une classe de priorité à chaque OB que vous configurez. Vous configurez le nombre de priorités dans les attributs des propriétés OB.

Etats d'exécution interruptible et non-interruptible

Les OB (Page 94) s'exécutent dans l'ordre de priorité des événements qui les déclenchent. Dans les propriétés de démarrage de la configuration d'appareil de la CPU (Page 177), vous pouvez configurer l'exécution de l'OB pour qu'elle soit interruptible ou non interruptible. Notez que les OB de ce cycle de programme sont toujours interruptibles, mais vous pouvez configurer tous les autres OB pour qu'ils soient interruptibles ou non interruptibles.

Si vous définissez l'état interruptible, alors si un OB s'exécute et qu'un événement de priorité plus haute se produit avant que l'OB achève son exécution, l'OB en cours s'interrompt pour permettre à l'OB d'un événement ayant la priorité la plus élevée de s'exécuter. L'événement ayant la priorité la plus élevée s'exécute, et à son achèvement, l'OB qui a été interrompu reprend. Lorsque des événements multiples se produisent alors qu'un OB alarme s'exécute, la CPU traite ces événements dans l'ordre de priorité.

Si vous ne définissez pas un état interruptible, alors un OB s'exécute jusqu'à son achèvement lorsqu'il est déclenché indépendamment de tout autre événement qui se déclenche pendant sa durée de fonctionnement.

Envisagez les deux cas suivants où des événements d'alarme déclenchent un OB cyclique et un OB temporisé. Dans les deux cas, l'OB d'alarme temporisée (OB 201) n'a pas d'affectation de mémoire image partielle (Page 85) et s'exécute avec la priorité 4. L'OB cyclique (OB 200) a une affectation de mémoire image partielle dans MIP1 et s'exécute avec la priorité 2. Les illustrations suivantes montrent la différence d'exécution entre des états d'exécution interruptible et non interruptible :

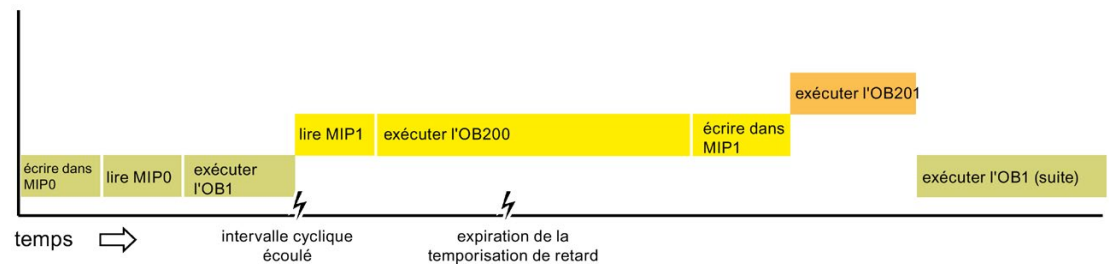


Figure 5-1 Cas 1 : Exécution d'OB non interruptible

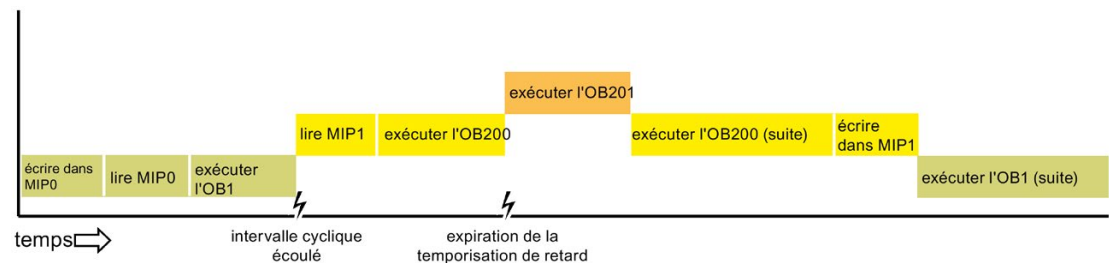


Figure 5-2 Cas 2 : exécution d'OB interruptible

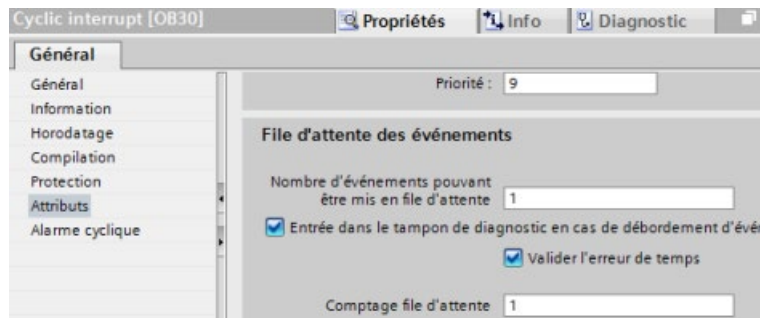
Remarque

Si vous configurez l'état d'exécution d'OB pour être non interruptible, alors un OB d'erreur de temps ne peut pas interrompre d'autres OB que des OB de cycle de programme. Avant la V4.0 de la CPU du S7-1200, un OB d'erreur de temps pouvait interrompre n'importe quel OB d'exécution. A partir de la V4.0, vous devez configurer l'exécution d'un OB pour être interruptible si vous voulez qu'un OB d'erreur de temps (ou n'importe quel autre OB d'une priorité plus élevée) soit capable d'interrompre l'exécution des OB qui ne sont pas des OB de cycle de programme.

Comprendre les priorités d'exécution et la mise en file d'attente des événements

La CPU limite le nombre d'événements en attente provenant d'une source unique, avec une file d'attente différente pour chaque type d'événement. Lorsque la limite d'événements en attente pour un type d'événement donné est atteinte, l'événement suivant qui survient est perdu. Vous pouvez utiliser une OB d'alarme d'erreur de temps (Page 98) pour répondre à des débordements de file d'attente.

Notez que STEP 7 vous permet de configurer des paramètres de mise en file d'attente des événements spécifiques pour l'OB d'alarme cyclique et l'OB d'alarme horaire.



Pour plus d'informations sur le comportement en surcharge de la CPU et la mise en fil d'attente des événements, reportez-vous au système d'information de STEP 7.

Chaque événement de CPU a une priorité associée. En général, les événements sont traités par la CPU selon l'ordre de priorité (priorité la plus élevée en premier). Les événements de même priorité sont traités selon le principe "premier arrivé, premier servi".

Tableau 5- 16 Événements d'OB

Événement	Quantité autorisée	Priorité d'OB par défaut
Cycle de programme	1 événement de cycle de programme Plusieurs OB autorisés	1 ⁴
Mise en route	1 événement de démarrage ¹ Plusieurs OB autorisés	1 ⁴
Alarme temporisée	Jusqu'à 4 événements horaires 1 OB par événement	OB 20 : 3 OB 21 : 4 OB 22 : 5 OB 23 : 6 OB 123 à OB 32767 : 3
Alarme cyclique	Jusqu'à 4 événements 1 OB par événement	OB 30 : 8 OB 31 : 9 OB 32 : 10 OB 33 : 11 OB 34 : 12 OB 35 : 13 OB 36 : 14 OB 37 : 16 OB 38 : 17 OB 123 à OB 32767 : 7
Alarme de processus	Jusqu'à 50 événements d'alarme de processus ² 1 OB par événement, mais vous pouvez utiliser le même OB pour des événements multiples	18
		18
Erreur de temps	1 événement (seulement s'il est configuré) ³	22 ou 26 ⁴
Erreur de diagnostic	1 événement (seulement s'il est configuré)	5
Débrochage ou enfi-chage de modules	1 événement	6
Défaillance du châssis ou de la station	1 événement	6
Heure	Jusqu'à 2 événements	2
Etat	1 événement	4
Mettre à jour	1 événement	4
Profil	1 événement	4

5.1 Exécution du programme utilisateur

Événement	Quantité autorisée	Priorité d'OB par défaut
MC-Servo	1 événement	25
MC-Interpolator	1 événement	24

- 1 L'événement démarrage et l'événement cycle de programme ne se produisent jamais en même temps, car l'événement démarrage s'exécute jusqu'à son achèvement avant que l'événement cycle de programme ne commence.
- 2 Vous pouvez avoir plus de 50 OB d'événements alarme de processus si vous utilisez les instructions DETACH et ATTACH.
- 3 Vous pouvez configurer la CPU afin qu'elle reste à l'état MARCHE si le temps de cycle de balayage dépasse le temps de balayage maximal ou vous pouvez utiliser l'instruction RE_TRIGR pour réinitialiser le temps de cycle. Toutefois, la CPU passera à l'état ARRET la deuxième fois que le temps de cycle dépasse le temps de cycle maximal.
- 4 La priorité pour une nouvelle CPU V4.0 ou V4.1 est fixée à 22. Si vous remplacez une CPU V3.0 par une CPU V4.0 ou V4.1, la priorité est 26 : la priorité en vigueur pour la V3.0. Dans tous les cas, le champ de priorité peut être édité et vous pouvez fixer la priorité à une valeur comprise entre 22 et 26.

Pour plus d'informations, voir la rubrique "Remplacement d'une CPU V3.0 par une CPU V4.2.x (Page 1665)".

En outre, la CPU reconnaît d'autres événements qui n'ont pas d'OB associés. La table suivante décrit ces événements et les actions de la CPU correspondantes :

Tableau 5- 17 événements supplémentaires

Événement	Description	action CPU
Erreur d'accès aux E/S	Erreur lecture/écriture E/S directe	La CPU enregistre la première occurrence dans le tampon de diagnostic et reste à l'état MARCHE. Vous pouvez accéder à la cause de l'erreur au moyen de l'instruction GET_ERROR_ID (Page 323).
Erreur de temps de cycle max.	La CPU dépasse le double du temps de cycle configuré	La CPU enregistre la première occurrence dans le tampon de diagnostic et reste à l'état MARCHE.
Erreur d'accès à la périphérie	Erreur d'E/S pendant la mise à jour de la mémoire image	La CPU enregistre la première occurrence dans le tampon de diagnostic et reste à l'état MARCHE.
Erreur de programmation	Erreurs d'exécution de programme	<ul style="list-style-type: none"> • Si le traitement d'erreur local dans le bloc a été activé, le système inscrit la cause de l'erreur dans la structure d'erreur. Vous pouvez accéder à la cause de l'erreur au moyen de l'instruction GET_ERROR_ID (Page 323). • En cas de traitement global des erreurs, le système inscrit un événement déclencheur d'erreur d'accès dans le tampon de diagnostic et reste à l'état MARCHE.

Temps d'attente

Le temps d'attente d'un événement d'alarme (le temps qui s'écoule entre la notification à la CPU qu'un événement s'est produit et le moment où la CPU exécute la première instruction dans l'OB gérant cet événement) est d'environ 175 µs, à condition qu'un OB de cycle de programme soit le seul programme de service d'événement actif au moment de l'événement d'alarme.

5.1.4 Surveillance et configuration du temps de cycle

Le temps de cycle est le temps nécessaire au système d'exploitation de la CPU pour exécuter la phase cyclique de l'état MARCHE. La CPU fournit deux méthodes de surveillance du temps de cycle :

- Temps de cycle maximum
- Temps de cycle minimum

La surveillance du cycle commence une fois l'événement démarrage achevé. La configuration de cette fonction se fait dans la "Configuration d'appareil" pour la CPU, sous "Temps de cycle".

La CPU surveille le temps de cycle et réagit si le temps de cycle dépasse le temps de cycle maximum configuré. La CPU génère une erreur et réagit comme suit si le temps de cycle dépasse le temps de cycle maximum configuré :

- Si le programme utilisateur contient un OB d'erreur de temps (Page 98), la CPU l'exécute.
- Si le programme utilisateur ne contient pas d'OB d'erreur de temps, l'événement d'erreur de temps génère une entrée dans le tampon de diagnostic. La CPU passe à l'état ARRÊT.

L'instruction RE_TRIGR (Page 321) (redémarrer surveillance du temps de cycle) vous permet de réinitialiser la temporisation qui mesure le temps de cycle. Si le temps écoulé pour l'exécution du cycle de programme en cours est dix fois inférieur au temps de cycle maximum configuré, l'instruction RE_TRIGR redémarre la surveillance du temps de cycle et met ENO à VRAI. Sinon, l'instruction RE_TRIGR ne redémarre pas la surveillance du temps de cycle. ENO est mis à FAUX.

Typiquement, le cycle s'exécute aussi rapidement qu'il le peut et le cycle suivant commence dès que le cycle en cours s'achève. La durée d'un cycle peut varier d'un cycle à l'autre en fonction du programme utilisateur et des tâches de communication. Pour éliminer cette variation, la CPU prend en charge un temps de cycle minimum optionnel. Si vous activez cette fonction optionnelle et que vous fixez un temps de cycle minimum en ms, après l'exécution de l'OB du cycle de programme, la CPU attend que le temps de cycle minimum s'écoule avant de répéter le cycle de programme.

Si la CPU achève le cycle normal en moins de temps que le temps de cycle minimum spécifié, elle passe le temps restant du cycle à effectuer le diagnostic d'exécution et/ou à traiter des demandes de communication.

5.1 Exécution du programme utilisateur

Si la CPU n'achève pas le cycle pendant le temps de cycle minimum indiqué, elle continue à exécuter le cycle normalement jusqu'à son achèvement (traitement de la communication inclus) et ne crée aucune réaction système en réponse à ce dépassement du temps de cycle minimum. Le tableau suivant présente les plages et les valeurs par défaut pour les fonctions de surveillance du temps de cycle :

Tableau 5- 18 Plage pour le temps de cycle

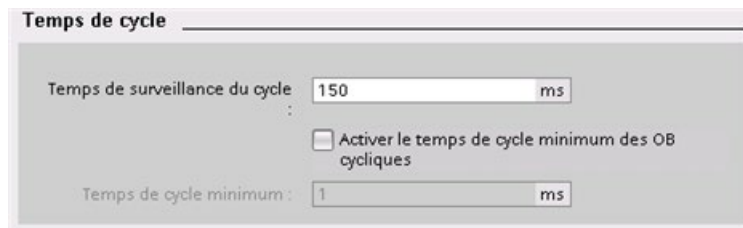
Temps de cycle	Plage (ms)	Valeur par défaut
Temps de cycle maximum ¹	1 à 6000	150 ms
Temps de cycle minimum ²	1 à temps de cycle maximum	Désactivé

- ¹ Le temps de cycle maximum est toujours activé. Configurez un temps de cycle compris entre 1 ms et 6 000 ms. La valeur par défaut est 150 ms.
- ² Le temps de cycle minimum est facultatif ; il est désactivé par défaut. Si nécessaire, configurez un temps de cycle compris entre 1 ms et le temps de cycle maximum.

Configuration du temps de cycle et de la charge due à la communication

Vous vous servez des propriétés CPU dans la configuration des appareils pour configurer les paramètres suivants :

- Cycle : Vous pouvez entrer un temps de surveillance de cycle maximum. Vous pouvez également activer et entrer un temps de cycle minimum.



- Charge due à la communication : Vous pouvez configurer un pourcentage du temps à réserver aux tâches de communication.



Reportez-vous au paragraphe "Surveillance du temps de cycle" (Page 111) pour plus d'informations sur le cycle.

5.1.5 Mémoire de la CPU

Gestion de la mémoire

La CPU fournit les zones de mémoire suivantes pour stocker le programme utilisateur, les données et la configuration :

- La mémoire de chargement est une mémoire rémanente pour le programme utilisateur, les données et la configuration. Lorsqu'un projet est chargé dans la CPU, la CPU le stocke d'abord dans la mémoire de chargement. Cette zone se trouve soit sur une carte mémoire (en présence d'une telle carte) soit dans la CPU. La CPU conserve cette zone de mémoire non volatile en cas de coupure de courant. La carte mémoire offre plus d'espace de stockage que la mémoire de chargement intégrée dans la CPU.
- La mémoire de travail est une mémoire volatile pour certains éléments du projet utilisateur pendant l'exécution du programme utilisateur. La CPU copie certains éléments du projet de la mémoire de chargement dans la mémoire de travail. Cette zone de mémoire volatile est perdue en cas de coupure de courant et est restaurée par la CPU au retour de la tension.
- La mémoire rémanente constitue un stockage permanent pour une quantité limitée de valeurs de mémoire de travail. La CPU utilise la zone de mémoire rémanente pour stocker les valeurs d'adresses d'opérande utilisateur choisies pendant une coupure de courant. Si une mise hors tension ou une coupure de courant se produit, la CPU restaurera ces valeurs rémanentes à la mise sous tension suivante.

Pour afficher l'utilisation de la mémoire pour un bloc de programme compilé, cliquez sur le bloc en question avec le bouton droit de la souris dans le dossier "Blocs de programme", dans l'arborescence de projet STEP 7 et sélectionnez "Ressources" dans le menu contextuel. Les propriétés de compilation affichent la mémoire de chargement et la mémoire de travail pour le bloc compilé.

Pour afficher l'utilisation de la mémoire dans la CPU en ligne, double-cliquez sur "En ligne & Diagnostic" dans STEP 7, affichez le détail de "Diagnostic" et sélectionnez "Mémoire".

Mémoire rémanente

Vous pouvez éviter la perte de données en cas de coupure de courant en définissant certaines données comme rémanentes. La CPU vous permet de configurer les données suivantes comme rémanentes :

- **Mémentos (M) :** Vous pouvez définir la taille de la mémoire rémanente pour les mémentos dans la table des variables API ou dans le tableau d'affectation. Les mémentos rémanents commencent toujours à MB0 et comprennent un nombre indiqué d'octets consécutifs. Indiquez cette valeur dans la table des variables API ou dans le tableau d'affectation en cliquant sur le bouton "Rémanence" de la barre d'outils. Entrez le nombre d'octets de mémentos à définir comme rémanents à compter de MB0.

Remarque : Pour n'importe quel bloc, vous pouvez afficher le tableau d'affectation en sélectionnant un bloc dans le dossier Blocs de programme puis en sélectionnant la commande de menu **Outils > Tableau d'affectation**.

- **Variables d'un bloc fonctionnel :** Si un FB est de type à "accès au bloc optimisé", l'éditeur d'interface pour ce FB comprend une colonne "Rémanence". Dans cette colonne, vous pouvez sélectionner "Rémanente", "Non rémanente" ou "Activer dans l'IDB" individuellement pour chaque variable. Lorsque vous placez un tel FB dans le programme, le DB d'instance correspondant à ce FB comporte également cette colonne "Rémanence". Vous ne pouvez modifier l'état de rémanence d'une variable dans l'éditeur d'interface de DB d'instance que si vous avez sélectionné "Définir dans IDB" (définir dans le bloc de données d'instance) dans la colonne "Rémanence" de la variable concernée dans le FB optimisé.

Si un FB **n'est pas** de type à "accès au bloc optimisé", l'éditeur d'interface pour ce FB ne comporte pas de colonne "Rémanence". Lorsque vous placez un tel FB dans le programme, le DB d'instance correspondant à ce FB comprend une colonne "Rémanence" dont le contenu peut être modifié. Dans ce cas, sélectionner l'option "Rémanente" pour n'importe quelle variable entraîne son activation pour **toutes** les variables. De même, désactiver cette option pour n'importe quelle variable entraîne sa désactivation pour **toutes** les variables.

Pour afficher ou modifier l'attribut d'optimisation d'un FB, ouvrez les propriétés du FB et sélectionnez les attributs.

- **Variables d'un bloc de données global :** Le comportement d'un DB global quant à l'affectation de l'état de rémanence est similaire à celui d'un FB. Selon le paramétrage d'accès au bloc, vous pouvez définir l'état de rémanence soit pour des variables individuelles soit pour toutes les variables d'un bloc de données global.
 - Si vous avez sélectionné "Optimisé" lors de la création du DB, vous pouvez définir l'état de rémanence pour chaque variable individuelle.
 - Si vous avez sélectionné "Standard - compatible avec S7-300/400" lors de la création du DB, le paramétrage de l'état de rémanence s'applique à toutes les variables du DB : soit toutes les variables sont rémanentes, soit aucune variable n'est rémanente.

La CPU permet le stockage de 10 240 octets de données rémanentes. Pour voir combien d'octets sont disponibles, cliquez sur le bouton "Rémanence" de la barre d'outils dans la table des variables API ou dans le tableau d'affectation. Les informations affichées mentionnent certes la plage de rémanence pour les mémentos, mais également, dans la deuxième ligne, la mémoire restante totale disponible pour les mémentos et les DB combinés. Notez que, pour que cette valeur soit exacte, vous devez compiler tous les blocs de données ayant des variables rémanentes.

Remarque

Le chargement d'un programme dans la CPU n'efface pas ni ne modifie les valeurs existantes en mémoire rémanente. Pour effacer la mémoire rémanente avant un chargement, réinitialisez la CPU aux réglages d'usine avant d'y charger le programme.

5.1.5.1 Mémento système et mémento de cadence

Vous vous servez des propriétés CPU pour activer les octets "mémento système" et "mémento de cadence". La logique de votre programme peut faire référence aux bits individuels de ces fonctions par leur nom de variable.

- Vous pouvez affecter un octet dans la zone de mémoire M au mémento système. L'octet de mémento système fournit les quatre bits suivants auxquels votre programme utilisateur peut faire référence par les noms de variables suivants :
 - Premier cycle (nom de variable "FirstScan") : Ce bit est mis à 1 pendant la durée du premier cycle une fois l'OB de démarrage terminé. Une fois l'exécution du premier cycle achevée, le bit "Premier cycle" est mis à 0.
 - Etat de diagnostic modifié : (Nom de la variable : "DiagStatusUpdate") : Ce bit est mis à 1 pendant un cycle après que la CPU a consigné un événement de diagnostic. Comme la CPU ne définit pas le bit "DiagStatusUpdate" avant la fin de la première exécution des OB de cycle de programme, votre programme utilisateur ne peut pas détecter s'il y a eu une modification du diagnostic pendant l'exécution des OB de démarrage ou la première exécution des OB de cycle de programme.
 - Toujours 1 (high) (nom de variable "AlwaysTRUE") : Ce bit est toujours à 1.
 - Toujours 0 (low) (nom de variable "AlwaysFALSE") : Ce bit est toujours à 0.
- Vous pouvez affecter un octet dans la zone de mémoire M au mémento de cadence. Chaque bit de l'octet configuré comme mémento de cadence génère une impulsion en signaux carrés. L'octet de mémento de cadence fournit 8 fréquences différentes, de 0,5 Hz (lent) à 10 Hz (rapide). Vous pouvez utiliser ces bits comme bits de commande, en particulier en combinaison avec des instructions sur front, pour déclencher des actions dans le programme utilisateur sur une base cyclique.

La CPU initialise ces octets lors du passage de l'état ARRET à l'état MISE EN ROUTE. Les bits du memento de cadence changent de manière synchrone avec l'horloge CPU aux états MISE EN ROUTE et MARCHE.

! PRUDENCE

Risques liés à l'écrasement des bits du memento système ou du memento de cadence

L'écrasement des bits du memento système ou du memento de cadence peut altérer les données dans ces fonctions et provoquer un fonctionnement incorrect du programme utilisateur, pouvant entraîner la mort et des blessures graves.

Comme le memento de cadence et le memento système ne sont pas réservés en mémoire M, des instructions ou des tâches de communication peuvent écrire dans ces adresses et altérer les données.

Evitez d'écrire des données dans ces adresses afin de garantir le bon fonctionnement de ces fonctions et prévoyez toujours un circuit d'arrêt d'urgence pour votre processus ou votre installation.

La fonction memento système configure un octet dont les bits sont activés (valeur = 1) pour un événement spécifique.

Bits de memento système

Activer l'utilisation de l'octet de memento système

Adresse de l'octet de memento système (MBx) :

Premier cycle :

Diagramme de diagnostic modifié :

Toujours 1 (high) :

Toujours 0 (low) :

Tableau 5- 19 Memento système

7	6	5	4	3	2	1	0
Réservés Valeur 0				Toujours désactivé Valeur 0	Toujours activé Valeur 1	Indicateur Etat de diagnostic <ul style="list-style-type: none"> 1: Modification 0: Pas de modification 	Indicateur Premier cycle <ul style="list-style-type: none"> 1: Premier cycle après la mise en route 0: Pas le premier cycle

La fonction memento de cadence configure un octet qui met les différents bits à 1 et à 0 à intervalles fixes. Chaque bit de cadence génère une impulsion en signaux carrés sur le bit M correspondant. Ces bits peuvent être utilisés comme bits de commande, en particulier en combinaison avec des instructions sur front, pour déclencher des actions dans le code utilisateur sur une base cyclique.

Bits de memento de cadence

Activer l'utilisation de l'octet de memento de cadence

Adresse de l'octet de memento de cadence (MBx) :

Cadence 10 Hz :

Cadence 5 Hz :

Cadence 2.5 Hz :

Cadence 2 Hz :

Cadence 1.25 Hz :

Cadence 1 Hz :

Cadence 0.625 Hz :

Cadence 0.5 Hz :

Tableau 5- 20 Memento de cadence

Numéro du bit	7	6	5	4	3	2	1	0
Nom de variable								
Période (s)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Fréquence (Hz)	0.5	0.625	1	1.25	2	2.5	5	10

Comme le memento de cadence opère de manière asynchrone par rapport au cycle de la CPU, son état peut changer plusieurs fois au cours d'un cycle long.

5.1.6 Mémoire tampon de diagnostic

La CPU prend en charge un tampon de diagnostic qui contient une entrée pour chaque événement de diagnostic. Chaque entrée inclut la date et l'heure auxquelles l'événement s'est produit, une catégorie d'événement, ainsi qu'une description de l'événement. Les entrées sont affichées par ordre chronologique, l'événement le plus récent se trouvant en haut. Les 50 événements les plus récents (maximum) sont disponibles dans ce journal. Lorsque le journal est plein, l'événement le plus ancien dans le journal est remplacé par un nouvel événement. En cas de coupure de courant, les événements sont sauvegardés.

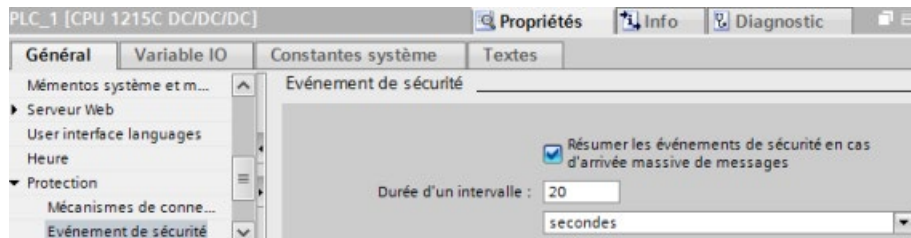
Les types d'événements suivants sont enregistrés dans la mémoire tampon de diagnostic :

- Chaque événement de diagnostic système, par exemple les erreurs de CPU et les erreurs de module
- Chaque changement d'état de la CPU (chaque mise sous tension, chaque passage à ARRET, chaque passage à MARCHE)

Vous devez être en ligne pour accéder à la mémoire tampon de diagnostic (Page 1393). Dans la vue "En ligne & Diagnostic, localisez la mémoire tampon de diagnostic sous "Diagnostic > Tampon de diagnostic".

Réduction du nombre d'événements de diagnostic de sécurité

Certains événements de sécurité génèrent des entrées répétées dans le tampon de diagnostic. Ces messages peuvent remplir le tampon de diagnostic et éventuellement masquer d'autres messages d'événements. Vous pouvez configurer l'API de telle manière qu'il limite le nombre de messages de diagnostic provenant d'événements de sécurité. Vous faites vos choix dans la configuration d'appareil de la CPU en fonction de l'intervalle de temps pendant lequel vous voulez supprimer les messages récurrents :



Si vous décidez de résumer les événements de sécurité pendant un intervalle de temps, vous pouvez définir la durée de cet intervalle en secondes, minutes ou heures avec une valeur numérique comprise dans la plage allant de 1 à 255.

Si vous choisissez de limiter les événements de sécurité, vous limiterez les types d'événements suivants :

- Liaison en ligne avec le mot de passe correct ou incorrect
- Détection de données de communication manipulées
- Données manipulées détectées sur la carte mémoire
- Détection d'un fichier de mise à jour du firmware manipulé
- Niveau de protection modifié (protection d'accès) chargé dans la CPU
- Légitimation par mot de passe restreinte ou activée (par instruction ou affichage CPU)
- Accès en ligne refusé, car le nombre maximum possible de tentatives d'accès simultanées est dépassé
- Dépassement de temps lorsqu'une liaison en ligne existante est inactive
- Ouverture d'une session sur le serveur Web avec le mot de passe correct ou incorrect
- Création d'une sauvegarde de la CPU
- Restauration de la configuration de la CPU

5.1.7 Horloge temps réel

La CPU prend en charge une horloge temps réel. Un supercondensateur fournit l'énergie nécessaire pour que l'horloge continue à fonctionner lorsque la CPU est hors tension. Le supercondensateur se recharge lorsque la CPU est sous tension. Une mise sous tension de la CPU d'au moins 24 heures charge suffisamment le supercondensateur pour assurer le fonctionnement de l'horloge pendant 20 jours typiquement.

STEP 7 règle l'horloge temps réel sur l'heure système qui a une valeur par défaut au déballage ou suite à une réinitialisation aux réglages d'usine. Pour utiliser l'horloge temps réel, vous devez la régler. Les horodatages tels ceux pour les entrées de la mémoire de diagnostic, les journaux de données et les entrées de journaux se basent sur l'heure système. Vous réglez l'heure grâce à la fonction "Réglage de l'heure" (Page 1386) dans la vue "En ligne & Diagnostic" de la CPU en ligne. STEP 7 calcule alors l'heure système à partir de l'heure que vous avez réglée plus ou moins le décalage du système d'exploitation Windows par rapport au temps universel coordonné (UTC). Régler l'heure sur l'heure locale en cours donne une heure système UTC si les paramètres de votre système d'exploitation Windows pour le fuseau horaire et l'heure d'été correspondent à vos paramètres régionaux.

STEP 7 comporte des instructions (Page 350) pour lire et écrire l'heure système (RD_SYS_T et WR_SYS_T), pour lire l'heure locale (RD_LOC_T) et pour régler le fuseau horaire (SET_TIMEZONE). L'instruction RD_LOC_T calcule l'heure locale à l'aide des décalages de fuseau horaire et d'heure d'été que vous paramétrez dans la configuration "Heure" des propriétés générales de la CPU (Page 177). Ces paramètres vous permettent de définir votre fuseau horaire pour l'heure locale, d'activer éventuellement l'heure d'été et d'indiquer les dates et heures de début et de fin de l'heure d'été. Vous pouvez également utiliser l'instruction SET_TIMEZONE pour configurer ces paramètres.

5.1.8 Configuration des sorties lors d'un passage de MARCHE à ARRET

Vous pouvez configurer le comportement des sorties TOR et analogiques lorsque la CPU est à l'état ARRET. Vous pouvez paramétrer n'importe quelle sortie d'une CPU, d'un SB ou d'un SM de sorte que sa valeur est figée ou qu'une valeur de remplacement est utilisée :

- Remplacement par une valeur de sortie spécifique (par défaut) : Vous entrez une valeur de remplacement pour chaque sortie (voie) de cette CPU, ce SB ou ce SM.

La valeur de remplacement par défaut pour les voies de sortie TOR est "désactivé" et la valeur de remplacement par défaut pour les voies de sortie analogiques est 0.

- Maintenir les sorties à leur dernier état : Les sorties conservent la valeur qu'elles avaient lors du passage de MARCHE à ARRET. Après une mise sous tension, les sorties sont définies à la valeur de remplacement par défaut.

Vous configurez le comportement des sorties dans la Configuration d'appareil. Sélectionnez les différents appareils et utilisez l'onglet "Propriétés" pour configurer les sorties de chaque appareil.

Remarque

Certains modules de périphérie décentralisée offrent des paramétrages supplémentaires pour la réaction à l'état d'arrêt de la CPU. Faites un choix à partir de la liste de choix dans Configuration de l'appareil pour ces modules.

Lorsque la CPU passe de MARCHE à ARRET, elle conserve la mémoire image et écrit les valeurs appropriées pour les sorties TOR et analogiques en fonction de la configuration.

5.2 Stockage des données, zones de mémoire, E/S et adressage

5.2.1 Accès aux données du S7-1200

STEP 7 facilite la programmation symbolique. Vous créez des noms symboliques ou "variables" pour les adresses des données, soit sous forme de variables API associées à des adresses mémoire et à des E/S, soit sous forme de variables locales utilisées dans un bloc de code. Pour utiliser ces variables dans votre programme utilisateur, il vous suffit d'entrer le nom de la variable comme paramètre de l'instruction.

Pour une meilleure compréhension de la manière dont la CPU organise et adresse les zones de mémoire, les paragraphes suivants expliquent l'adressage "absolu" référencé par les variables API. La CPU offre plusieurs possibilités pour stocker les données pendant l'exécution du programme utilisateur :

- Mémoire globale : La CPU fournit diverses zones de mémoire spécialisées, à savoir les entrées (I), les sorties (Q) et les mémentos (M). Cette mémoire est accessible à tous les blocs de code sans restriction.
- Table de variables API : Vous pouvez entrer des mnémoniques pour des adresses de mémoire spécifiques dans la table de variables API STEP 7. Ces variables sont globales pour le programme STEP 7 et permettent de programmer avec des noms qui ont une signification dans votre application.
- Blocs de données (DB) : Vous pouvez inclure des blocs de données (DB) dans votre programme utilisateur afin de sauvegarder les données des blocs de code. Les données sauvegardées sont conservées une fois l'exécution du bloc de code associé achevée. Un DB "global" contient des données pouvant être utilisées par tous les blocs de code alors qu'un DB d'instance contient les données d'un FB spécifique et a une structure correspondant aux paramètres du FB.
- Mémoire temporaire : Lors de l'appel d'un bloc de code, le système d'exploitation de la CPU alloue de la mémoire temporaire - ou locale (L) - utilisable pendant l'exécution de ce bloc. Lorsque l'exécution de ce bloc de code s'achève, la CPU réalloue la mémoire locale pour l'exécution d'autres blocs de code.

Chaque emplacement de mémoire différent a une adresse unique. Votre programme utilisateur utilise ces adresses pour accéder aux informations contenues dans l'emplacement de mémoire. Les références aux zones de mémoire d'entrée (I) ou de sortie (Q), telles que I0.3 ou Q1.7, permettent d'accéder à la mémoire image. Pour accéder directement à l'entrée ou à la sortie physique, ajoutez ":P" à la référence (par exemple, I0.3:P, Q1.7:P ou "Arret:P").

Tableau 5- 21 Zones de mémoire

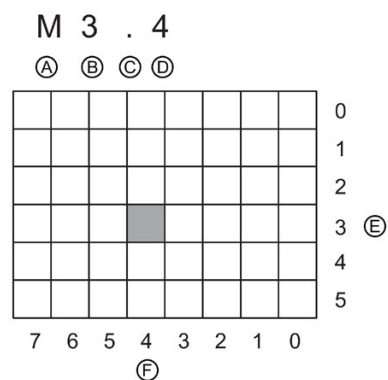
Zone de mémoire	Description	Forçage permanent	Rémanence
I Mémoire image des entrées I_:P ¹ (entrée physique)	Copiée des entrées physiques au début du cycle	Non	Non
	Lecture directe des entrées physiques sur la CPU, le SB et le SM	Oui	Non
Q Mémoire image des sorties Q_:P ¹ (sortie physique)	Copiée dans les sorties physiques au début du cycle	Non	Non
	Ecriture directe dans les entrées physiques sur la CPU, le SB et le SM	Oui	Non
M Mémentos	Mémoire de commande et de données	Non	Oui (facultatif)
L Mémoire temporaire	Données temporaires pour un bloc, locales à ce bloc	Non	Non
DB Bloc de données	Mémoire de données ainsi que mémoire de paramètres pour les FB	Non	Oui (facultatif)

¹ Pour accéder directement (en lecture ou en écriture) aux entrées et sorties physiques, ajoutez ":P" à l'adresse ou à la variable (par exemple, I0.3:P, Q1.7:P ou "Arret:P").

Chaque emplacement de mémoire différent a une adresse unique. Votre programme utilisateur utilise ces adresses pour accéder aux informations contenues dans l'emplacement de mémoire. L'adresse absolue est constituée des éléments suivants :

- Identificateur de zone de mémoire (comme I, Q ou M)
- Taille des données en accès ("B" pour Byte, "W" pour Word ou "D" for DWord)
- Adresse de début des données (comme octet 3 ou mot 3)

Lors de l'accès à un bit dans l'adresse pour une valeur booléenne, vous ne précisez pas la taille. Vous saisissez uniquement la zone de mémoire, l'adresse d'octet et l'adresse de bit pour cette donnée (I0.0, Q0.1 ou M3.4, par exemple).



- | | | | |
|---|---|---|------------------------------|
| A | Identificateur de zone de mémoire | E | Octets de la zone de mémoire |
| B | Adresse d'octet : octet 3 | F | Bits de l'octet sélectionné |
| C | Séparateur ("octet.bit") | | |
| D | Emplacement du bit dans l'octet (bit 4 sur 8) | | |

Dans l'exemple, la zone de mémoire et l'adresse d'octet (M = zone des mémentos et 3 = Byte 3) sont suivies par un point (".") pour séparer l'adresse du bit (bit 4).

Accès aux données dans les zones de mémoire de la CPU

STEP 7 facilite la programmation symbolique. Typiquement, des variables sont créées dans les variables API, dans un bloc de données ou dans l'interface d'un OB, d'une FC ou d'un FB. Ces variables comprennent un nom, un type de données, un décalage et un commentaire. En plus, vous pouvez indiquer une valeur initiale dans un bloc de données. Vous pouvez utiliser ces variables lors de la programmation en entrant le nom de variable au niveau du paramètre de l'instruction. Vous pouvez optionnellement entrer l'opérande absolu (zone de mémoire, taille et décalage) au niveau du paramètre de l'instruction. Les exemples dans les paragraphes suivants montrent comment entrer des opérandes absolus. Le caractère % est inséré automatiquement devant l'opérande absolu par l'éditeur de programmes. Vous pouvez activer divers modes d'affichage dans l'éditeur de programmes : affichage symbolique, affichage symbolique et absolu ou affichage absolu.

I (mémoire image des entrées) : La CPU lit les entrées de périphérie (entrées physiques) juste avant l'exécution de l'OB cyclique de chaque cycle et écrit ces valeurs dans la mémoire image des entrées. Vous pouvez accéder à la mémoire image des entrées par bits, octets, mots ou doubles mots. L'accès en lecture et l'accès en écriture sont tous deux autorisés mais, typiquement, les entrées de la mémoire image sont uniquement lues.

Tableau 5- 22 Adressage absolu pour la mémoire I

Bit	I[adresse d'octet].[adresse de bit]	I0.1
Octet, mot ou double mot	I[taille][adresse d'octet de départ]	IB4, IW5 ou ID12

En ajoutant un ":P" à l'adresse, vous pouvez lire directement les entrées TOR et analogiques de la CPU, du SB, du SM ou du module décentralisé. La différence entre un accès via I_:P et un accès via I est que les données proviennent directement des entrées en accès et non de la mémoire image des entrées. On parle d'accès en "lecture directe" pour cet accès I_:P, car les données sont extraites directement de la source et non d'une copie qui a été faite lors de la dernière actualisation de la mémoire image des entrées.

Comme les entrées physiques reçoivent leur valeur directement des appareils de terrain connectés à ces entrées, écrire dans ces entrées est interdit. Ainsi, les accès I_:P sont en lecture seule alors que les accès I peuvent être en lecture ou en écriture.

Les accès I_:P sont également limités à la taille des entrées prises en charge par une CPU, un SB ou un SM unique, arrondie à l'octet le plus proche. Par exemple, si les entrées d'un SB 2 DI / 2 DQ sont configurées pour commencer en I4.0, il est possible d'y accéder en tant que I4.0:P et I4.1:P ou en tant que IB4:P. Les accès de I4.2:P à I4.7:P ne sont pas refusés mais n'ont aucun sens puisque les entrées correspondantes ne sont pas utilisées. Les accès à IW4:P et ID4:P sont interdits puisqu'ils dépassent le décalage d'octet associé au SB.

Les accès via I_:P n'affectent pas la valeur correspondante sauvegardée dans la mémoire image des entrées.

Tableau 5- 23 Adressage absolu pour la mémoire I (accès direct)

Bit	I[adresse d'octet].[adresse de bit]:P	I0.1:P
Octet, mot ou double mot	I[taille][adresse d'octet de départ]:P	IB4:P, IW5:P ou ID12:P

Q (mémoire image des sorties) : La CPU copie dans les sorties physiques les valeurs sauvegardées dans la mémoire image des sorties. Vous pouvez accéder à la mémoire image des sorties par bits, octets, mots ou doubles mots. L'accès en lecture et l'accès en écriture sont tous deux autorisés pour les sorties de la mémoire image.

Tableau 5- 24 Adressage absolu pour la mémoire Q

Bit	Q[adresse d'octet].[adresse de bit]	Q1.1
Octet, mot ou double mot	Q[taille][adresse d'octet de départ]	QB5, QW10, QD40

En ajoutant un ":P" à l'adresse, vous pouvez écrire directement dans les sorties TOR et analogiques physiques de la CPU, du SB, du SM ou du module décentralisé. La différence entre un accès via Q_:P et un accès via Q est que les données sont directement écrites dans les sorties en accès en plus d'être écrites dans la mémoire image des sorties. On parle parfois d'accès en "écriture directe" pour cet accès Q_:P, car les données sont envoyées directement à leur destination qui n'a pas besoin d'attendre la prochaine actualisation à partir de la mémoire image des sorties.

Comme les sorties physiques pilotent directement les appareils de terrain qui y sont connectés, la lecture de ces sorties est interdite. Ainsi, les accès Q_:P sont en écriture seule alors que les accès Q peuvent être en lecture ou en écriture.

Les accès Q_:P sont également limités à la taille des sorties prises en charge par une CPU, un SB ou un SM unique, arrondie à l'octet le plus proche. Par exemple, si les sorties d'un SB 2 DI / 2 DQ sont configurées pour commencer en Q4.0, il est possible d'y accéder en tant que Q4.0:P et Q4.1:P ou en tant que QB4:P. Les accès de Q4.2:P à Q4.7:P ne sont pas refusés mais n'ont aucun sens puisque les sorties correspondantes ne sont pas utilisées. Les accès à QW4:P et QD4:P sont interdits puisqu'ils dépassent le décalage d'octet associé au SB.

Les accès via Q_:P affectent à la fois la sortie physique et la valeur correspondante sauvegardée dans la mémoire image des sorties.

Tableau 5- 25 Adressage absolu pour la mémoire Q (accès direct)

Bit	Q[adresse d'octet].[adresse de bit]:P	Q1.1:P
Octet, mot ou double mot	Q[taille][adresse d'octet de départ]:P	QB5:P, QW10:P ou QD40:P

M (mémentos) : Utilisez la zone des mémentos (mémoire M) pour les relais de commande et les données afin de stocker l'état intermédiaire d'une opération ou d'autres informations de commande. Vous pouvez accéder à la zone des mémentos par bits, octets, mots ou doubles mots. L'accès en lecture et l'accès en écriture sont tous deux autorisés pour la mémoire M.

Tableau 5- 26 Adressage absolu pour la mémoire M

Bit	M[adresse d'octet].[adresse de bit]	M26.7
Octet, mot ou double mot	M[taille][adresse d'octet de départ]	MB20, MW30, MD50

Temp (mémoire temporaire) : La CPU affecte la mémoire temporaire selon les besoins. La CPU alloue la mémoire temporaire au bloc de code et initialise les adresses de mémoire à 0 au moment où elle démarre le bloc de code (pour un OB) ou appelle le bloc de code (pour une FC ou un FB).

La mémoire temporaire est similaire à la mémoire M à une exception majeure près. En effet, la mémoire M a une portée globale alors que la mémoire temporaire a une portée locale.

- Mémoire M : Tout OB, FC ou FB peut accéder aux données dans la mémoire M, ce qui signifie que les données sont globalement disponibles pour tous les éléments du programme utilisateur.
- Mémoire temporaire : La CPU limite l'accès aux données en mémoire temporaire à l'OB, à la FC ou au FB qui a créé ou déclaré l'adresse de mémoire temporaire. Les adresses de mémoire temporaire restent locales et des blocs de code différents ne partagent pas leur mémoire temporaire même lorsque le bloc de code appelle un autre bloc de code. Par exemple, lorsqu'un OB appelle une FC, la FC ne peut pas accéder à la mémoire temporaire de l'OB qui l'a appelée.

La CPU fournit de la mémoire temporaire (locale) pour chaque niveau de priorité d'OB :

- 16 Ko pour le démarrage et le cycle du programme, FB et FC associés inclus
- 6 Ko pour chaque fil d'événement d'alarme supplémentaire, FB et FC associés inclus

Vous accédez à la mémoire temporaire par adressage symbolique uniquement.

Vous pouvez voir la taille de mémoire (locale) temporaire utilisée par les blocs dans votre programme à travers la structure d'appel du programme dans STEP 7. À partir de l'arbre du projet, sélectionnez Information sur le programme et sélectionnez l'onglet Structure d'appel. Vous pouvez voir tous les OB présents dans votre programme et faire un zoom avant pour découvrir les blocs qu'ils appellent. Vous pouvez voir l'allocation de données locales pour chaque bloc. Vous pouvez accéder à l'affichage de la structure d'appel depuis la commande de menu **Outils > Structure d'appel** dans STEP 7.

DB (bloc de données) : Utilisez les blocs de données pour sauvegarder divers types de données, notamment l'état intermédiaire d'une opération ou d'autres informations de commande, les paramètres de FB et des structures de données nécessaires pour de nombreuses instructions telles que temporisations et compteurs. Vous pouvez accéder aux blocs de données par bits, octets, mots ou doubles mots. L'accès en lecture et l'accès en écriture sont tous deux autorisés pour les blocs de données en lecture/écriture. Seul l'accès en lecture est autorisé pour les blocs de données en lecture seule.

Tableau 5- 27 Adressage absolu pour la mémoire DB

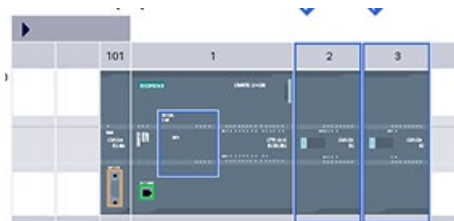
Bit	DB[numéro de bloc de données].DBX[adresse d'octet].[adresse de bit]	DB1.DBX2.3
Octet, mot ou double mot	DB[numéro de bloc de données].DB[taille][adresse d'octet de départ]	DB1.DBB4, DB10.DBW2, DB20.DBD8

Remarque

Lorsque vous indiquez une adresse absolue en CONT ou LOG, STEP 7 insère le caractère "%" devant cette adresse pour signaler qu'il s'agit d'une adresse absolue. Lors de la programmation, vous pouvez saisir une adresse absolue soit avec, soit sans le caractère "%" (par exemple, %I0.0 ou I.0). Si vous omettez le caractère "%", STEP 7 l'ajoutera.

En SCL, vous devez entrer le signe "%" avant l'adresse pour indiquer qu'il s'agit d'une adresse absolue. Si "%" manque, STEP 7 génère une erreur variable non définie au moment de la compilation.

Configuration des E/S dans la CPU et les modules E/S



Lorsque vous ajoutez une CPU et des modules d'E/S à votre configuration d'appareil, STEP 7 affecte automatiquement les adresses I et Q. Vous pouvez modifier l'adressage par défaut en sélectionnant le champ d'adresse dans la configuration d'appareil et en entrant de nouveaux nombres.

- STEP 7 attribue les entrées et sorties numériques dans des groupes de 8 points (1 octet), que le module utilise tous les points ou non.
- STEP 7 alloue les entrées et les sorties analogiques par groupes de 2, chaque entrée ou sortie analogique occupant 2 octets (16 bits).

Vue d'ensemble des appareils						
Module	Empla.	Adresse E	Adresse S	Type	N° de r	
	103					
	102					
RS485_1	101			CM 1241 (RS485)	6ES7	
PLC_1	1			CPU 1214C DCDCI	6ES7	
DI14/DO10	1.1	0...1	0...1	DI14/DO10		
AI2	1.2	64...67		AI2		
AO1 x 12Bi	1.3		80...81	Signal Board AO1	6ES7	
HSC_1	1.16	1000.....		Compteur rapide (.)		
HSC_2	1.17			Compteur rapide (.)		
HSC_3	1.18			Compteur rapide (.)		
HSC_4	1.19			Compteur rapide (.)		
HSC_5	1.20			Compteur rapide (.)		
HSC_6	1.21			Compteur rapide (.)		
Pulse_1	1.32			Générateur d'impu.		
Pulse_2	1.33			Générateur d'impu.		
Interface P. X1				Interface PROFINET		
DI8 x DC24V.	2	8		SM 1221 DI8 x DC	6ES7	

La figure montre un exemple de CPU 1214C avec deux SM et un SB. Dans cet exemple, vous pourriez remplacer l'adresse 8 du module DI8 par 2. L'outil vous aide en modifiant les plages d'adresses qui ont une taille incorrecte ou sont en conflit avec d'autres adresses.

5.3 Traitement des valeurs analogiques

Les modules d'entrées-sorties analogiques fournissent des signaux d'entrée ou attendent des valeurs de sortie qui représentent soit une plage de tension, soit une plage de courant. Il s'agit des plages ± 10 V, ± 5 V, ± 2.5 V ou 0 à 20 mA. Les valeurs renvoyées par les modules sont des valeurs entières, 0 à 27648 représentant la plage nominale pour le courant et -27648 à 27648 la plage nominale pour la tension. Toute valeur en dehors de ces plages représente soit un débordement haut, soit un débordement bas. Reportez-vous aux tableaux sur la représentation des entrées analogiques (Page 1554) et la représentation des sorties analogiques (Page 1556) pour plus de détails concernant les types de valeurs en dehors de la plage.

Dans votre programme de commande, vous aurez probablement besoin de ces valeurs en unités de mesure physiques, par exemple pour représenter un volume, une température, un poids ou une autre valeur de quantité. S'il s'agit d'une entrée analogique, vous devez d'abord normaliser la valeur analogique à une valeur réelle (à virgule flottante) comprise entre 0,0 et 1,0. Vous devez ensuite la mettre à l'échelle des valeurs minimum et maximum des unités physiques qu'elle représente. Dans le cas de valeurs en unités physiques que vous avez besoin de convertir en valeurs de sortie analogiques, vous devez d'abord normaliser la valeur en unités physiques à une valeur comprise entre 0,0 et 1,0, puis la mettre à l'échelle entre 0 et 27648 ou -27648 et 27648 en fonction de la plage du module analogique. STEP 7 fournit les instructions NORM_X et SCALE_X (Page 308) pour réaliser ces opérations. Vous pouvez également utiliser l'instruction CALCULATE (Page 268) pour mettre les valeurs analogiques (Page 42) à l'échelle.

Exemple : traitement des valeurs analogiques

Considérez, par exemple, une entrée analogique dont la plage actuelle s'étend de 0 à 20 mA. Le module d'entrées analogiques renvoie des valeurs comprises dans la plage de 0 à 27 648 pour les valeurs mesurées. Pour cet exemple, considérez que vous utilisez cette valeur d'entrée analogique pour mesurer une plage de température allant de 50 °C à 100 °C. Quelques exemples de valeurs auraient les significations suivantes :

Valeur d'entrée analogique	Unités physiques
0	50 °C
6 192	62,5 °C
12 384	75 °C
18 576	87,5 °C
27648	100 °C

Dans cet exemple, le calcul permettant de déterminer les unités physiques à partir de la valeur d'entrée analogique est le suivant :

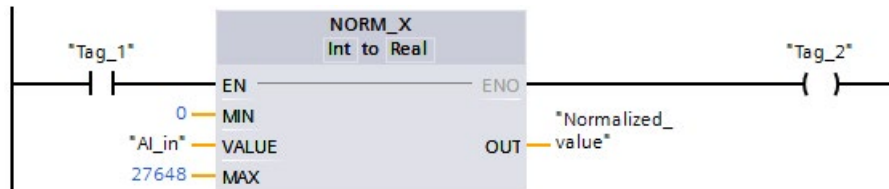
$$\text{Valeur des unités physiques} = 50 + (\text{valeur d'entrée analogique}) * (100 - 50) / (27648 - 0)$$

Pour le cas général, l'équation serait :

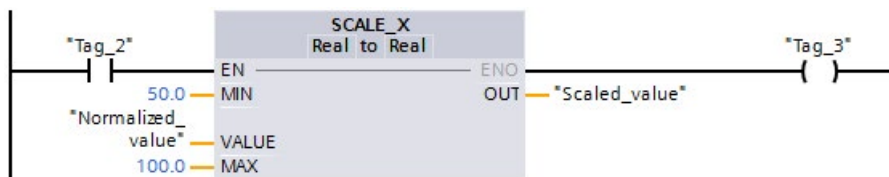
$$\begin{aligned} \text{Valeur d'unités physiques} = & (\text{fourchette basse d'unités physiques}) + \\ & (\text{Valeur d'entrée analogique}) * \\ & (\text{Fourchette haute d'unités physiques} - \text{fourchette basse d'unités physiques}) / \\ & (\text{Plage d'entrées analogiques maximum} - \text{plage d'entrées analogiques minimum}) \end{aligned}$$

Dans les applications API, la méthode traditionnelle consiste à normaliser la valeur d'entrée analogique en un nombre à virgule flottante compris entre 0,0 et 1,0. Ensuite, vous mettez à l'échelle la valeur en résultant en un nombre à virgule flottante situé dans la plage de vos unités physiques. Pour une question de simplicité, les instructions CONT suivantes utilisent des valeurs constantes pour les plages ; vous choisirez peut-être d'utiliser des variables :

Réseau 1



Réseau 2



5.4 Types de données

Les types de données servent à indiquer à la fois la taille d'un élément de données et la façon dont les données doivent être interprétées. Chaque paramètre d'une instruction accepte au moins un type de données et certains paramètres acceptent plusieurs types de données. Maintenez le curseur au-dessus du champ de paramètre d'une instruction pour voir les types de données acceptés par un paramètre particulier.

Un paramètre formel est un identificateur qui repère l'emplacement des données qu'une instruction doit utiliser (par exemple, l'entrée IN1 d'une instruction ADD). Un paramètre effectif est l'adresse de mémoire (précédée d'un caractère "%") ou la constante contenant les données que l'instruction doit utiliser (exemple : %MD400 "nombre_de_widgets"). Le type de données du paramètre effectif que vous indiquez doit correspondre à l'un des types de données acceptés par le paramètre formel indiqué par l'instruction.

Lorsque vous indiquez un paramètre effectif, vous devez indiquer soit une variable (mnémorique), soit une adresse de mémoire (directe) absolue. Les variables sont associées à un mnémorique (nom de la variable) avec un type de données, une zone de mémoire, un décalage en mémoire et un commentaire et elles peuvent être créées soit dans l'éditeur de variables API, soit dans l'éditeur d'interface d'un bloc (OB, FC, FB et DB). Si vous entrez une adresse absolue n'ayant pas de variable associée, vous devez utiliser une taille appropriée qui correspond à un type de données accepté et une variable par défaut sera créée lors de l'entrée.

Tous les types de données excepté String, Struct, Array et DTL sont disponibles à la fois dans l'éditeur de variables API et dans les éditeurs d'interface de bloc. String, Struct, Array et DTL sont disponibles uniquement dans les éditeurs d'interface de bloc. Vous pouvez également entrer une valeur constante pour de nombreux paramètres d'entrée.

- Bit et séquences de bits (Page 131) : Bool (valeur booléenne ou binaire), Byte (octet de 8 bits), Word (mot de 16 bits), DWord (double mot de 32 bits)
- Entier (Page 132)
 - USInt (entier de 8 bits non signé), SInt (entier de 8 bits signé)
 - UInt (entier de 16 bits non signé), Int (entier de 16 bits signé)
 - UDInt (entier de 32 bits non signé), DInt (entier de 32 bits signé)
- Réel à virgule flottante (Page 132) : Real (valeur réelle ou virgule flottante de 32 bits), LReal (valeur réelle ou virgule flottante de 64 bits)
- Date et heure (Page 133) : Time (durée CEI de 32 bits), Date (date de 16 bits), TOD (heure de 32 bits), DTL (structure date et heure de 12 octets)
- Caractère et chaîne de caractères (Page 135) : Char (caractère individuel de 8 bits), String (chaîne de longueur variable allant jusqu'à 254 caractères)
- Tableau (Page 137)
- Structure de données (Page 138): Struct
- Types de données API (Page 138)
- Types de données Variant (Page 139)

Bien que non disponibles en tant que types de données, les formats numériques BCD suivants sont pris en charge par les instructions de conversion :

Tableau 5- 28 Taille et plage du format DCB

Format	Taille (bits)	Plage numérique	Exemples d'entrée de constantes
BCD16	16	-999 à 999	123, -123
BCD32	32	-9999999 à 9999999	1234567, -1234567

5.4.1 Types de données Bool, Byte, Word et DWord

Tableau 5- 29 Types de données pour bit et séquence de bits

Type de données	Taille en bits	Type de nombre	Plage de nombres	Exemples de constantes	Exemples d'adresses
Bool	1	Booléen	FALSE ou TRUE	TRUE	I1.0 Q0.1 M50.7 DB1.DBX2.3 nom_variable
		Binaire	2#0 ou 2#1	2#0	
		Entier non signé	0 ou 1	1	
		Octal	8#0 ou 8#1	8#1	
		Hexadécimal	16#0 ou 16#1	16#1	
Byte	8	Binaire	2#0 à 2#1111_1111	2#1000_1001	IB2 MB10 DB1.DBB4 nom_variable
		Entier non signé	0 à 255	15	
		Entier signé	-128 à 127	-63	
		Octal	8#0 à 8#377	8#17	
		Hexadécimal	B#16#0 à B#16#FF, 16#0 à 16#FF	B#16#F, 16#F	
Word	16	Binaire	2#0 à 2#1111_1111_1111_1111	2#1101_0010_1001_0110	MW10 DB1.DBW2 nom_variable
		Entier non signé	0 à 65535	61680	
		Entier signé	-32768 à 32767	72	
		Octal	8#0 à 8#177_777	8#170_362	
		Hexadécimal	W#16#0 à W#16#FFFF, 16#0 à 16#FFFF	W#16#F1C0, 16#A67B	
DWord	32	Binaire	2#0 à 2#1111_1111_1111_1111_1111_1111	2#1101_0100_1111_1110_1000_1100	MD10 DB1.DBD8 nom_variable
		Entier non signé*	0 à 4_294_967_295	15_793_935	
		Entier signé*	-2_147_483_648 à 2_147_483_647	-400000	
		Octal	8#0 à 8#37_777_777_777	8#74_177_417	
		Hexadécimal	DW#16#0000_0000 à DW#16#FFFF_FFFF, 16#0000_0000 à 16#FFFF_FFFF	DW#16#20_F30A, 16#B_01F6	

* Le trait de soulignement "_" est un séparateur des milliers qui améliore la lisibilité des nombres de plus de huit chiffres.

5.4.2 Types de données entiers

Tableau 5- 30 Types de données entiers (U = non signé, S = court, D= double)

Type de données	Taille en bits	Plage de nombres	Exemples de constantes	Exemples d'adresses
USInt	8	0 à 255	78, 2#01001110	MB0, DB1.DBB4, nom_variable
SInt	8	-128 à 127	+50, 16#50	
UInt	16	0 à 65 535	65295, 0	MW2, DB1.DBW2, nom_variable
Int	16	-32 768 à 32 767	30000, +30000	
UDInt	32	0 à 4 294 967 295	4042322160	MD6, DB1.DBD8, nom_variable
DInt	32	-2 147 483 648 à 2 147 483 647	-2131754992	

5.4.3 Types de données réels à virgule flottante

Les nombres réels (ou nombres à virgule flottante) sont représentés sous forme de nombres de 32 bits à simple précision (type Real) ou de nombres de 64 bits à double précision (type LReal) comme décrit dans la norme ANSI/IEEE 754-1985. Les nombres à virgule flottante simple précision sont précis jusqu'à 6 chiffres significatifs et les nombres à virgule flottante double précision sont précis jusqu'à 15 chiffres significatifs. Vous pouvez donc indiquer 6 (Real) et 15 (LReal) chiffres significatifs au maximum lorsque vous entrez une constante à virgule flottante afin de conserver la précision.

Tableau 5- 31 Types de données réels à virgule flottante (L=long)

Type de données	Taille en bits	Plage de nombres	Exemples de constantes	Exemples d'adresses
Real	32	-3,402823e+38 à -1,175 495e-38, ±0, +1,175 495e-38 à +3,402823e+38	123.456, -3.4, 1.0e-5	MD100, DB1.DBD8, nom_variable
LReal	64	-1,7976931348623158e+308 à -2,2250738585072014e-308, ±0, +2,2250738585072014e-308 à +1,7976931348623158e+308	12345,123456789e40, 1,2E+40	nom_DB.nom_var Règles : <ul style="list-style-type: none"> • L'adressage direct n'est pas pris en charge. • Peut être défini dans une table d'interface de bloc OB, FB ou FC

Les calculs impliquant une longue série de valeurs comprenant des nombres très grands et très petits peuvent produire des résultats qui manquent de précision. Cela peut arriver si ces nombres diffèrent de 10 puissance x, x étant supérieur à 6 (Real) ou à 15 (LReal). Exemple pour Real : 100 000 000 + 1 = 100 000 000.

5.4.4 Types de données "date et heure"

Tableau 5- 32 Types de données "date et heure"

Type de données	Taille	Plage	Exemples d'entrée de constantes
TIME	32 bits	T#-24d_20h_31m_23s_648ms à T#24d_20h_31m_23s_647ms Sauvegardé en tant que : -2 147 483 648 ms à +2 147 483 647 ms	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms 10d20h30m20s630ms
DATE	16 bits	D#1990-1-1 à D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31
TIME_OF_DAY	32 bits	TOD#0:0:0.0 à TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (date et heure long)	12 octets	Min. : DTL#1970-01-01-00:00:00.0 Max. : DTL#2262-04-11:23:47:16.854 775 807	DTL#2008-12-16- 20:30:20.250

TIME

Les données TIME sont sauvegardées sous forme d'entier double signé interprété en tant que millisecondes. Le format d'éditeur peut utiliser des informations pour le jour (d), les heures (h), les minutes (m), les secondes (s) et les millisecondes (ms).

Il n'est pas nécessaire de préciser toutes les unités de durée. T#5h10s et 500h sont, par exemple, valables.

La valeur combinée de toutes les valeurs d'unité indiquées ne doit pas dépasser les limites supérieure et inférieure en millisecondes pour le type de données TIME (-2 147 483 648 ms à +2 147 483 647 ms).

DATE

Les données DATE sont sauvegardées sous forme d'entier non signé interprété en tant que nombre de jours ajoutés à la date de base, 01/01/1990, pour obtenir la date précisée. Le format d'éditeur doit indiquer une année, un mois et un jour.

TOD

Les données TOD (TIME_OF_DAY) sont sauvegardées sous forme d'entier double non signé interprété en tant que nombre de millisecondes depuis minuit pour l'heure indiquée (minuit = 0 ms). Les heures (24 heures/jour), les minutes et les secondes doivent être indiquées. L'indication des fractions de seconde est facultative.

DTL

Le type de données DTL (date et heure long) utilise une structure de 12 octets qui mémorise les informations sur la date et l'heure. Vous pouvez définir des données DTL soit dans la mémoire temporaire d'un bloc, soit dans un DB. Il faut entrer une valeur pour chaque élément dans la colonne Valeur initiale de l'éditeur de DB.

Tableau 5- 33 Taille et plage pour DTL

Longueur (octets)	Format	Plage de valeurs	Exemple d'entrée de valeur
12	Horloge et calendrier année-mois-jour:heures:minutes:secondes.nanosecondes	Min. : DTL#1970-01-01-00:00:00.0 Max. : DTL#2554-12-31-23:59:59.999 999	DTL#2008-12-16-20:30:20.250

Chaque élément de la structure DTL contient un type de données et une plage de valeurs différents. Le type de données d'une valeur indiquée doit correspondre au type de données des éléments correspondants.

Tableau 5- 34 Eléments de la structure DTL

Octet	Élément	Type de données	Plage de valeurs
0	Année	UINT	1970 à 2554
1			
2	Mois	USINT	1 à 12
3	Jour	USINT	1 à 31
4	Jour de la semaine ¹	USINT	1 (dimanche) à 7 (samedi) ¹
5	Heures	USINT	0 à 23
6	Minutes	USINT	0 à 59
7	Secondes	USINT	0 à 59
8	Nanosecondes	UDINT	0 à 999 999 999
9			
10			
11			

¹ Le format année-mois-jour:heures:minutes:secondes.nanosecondes ne comprend pas les jours de la semaine.

5.4.5 Types de données "caractère et chaîne de caractères"

Tableau 5- 35 Types de données "caractère et chaîne de caractères"

Type de données	Taille	Plage	Exemples d'entrée de constantes
Char	8 bits	16#00 à 16#FF	'A', 't', '@', 'ä', 'Σ'
WChar	16 bits	16#0000 à 16#FFFF	'A', 't', '@', 'ä', 'Σ', Caractères asiatiques, caractères cyrilliques et autres
String	n+ 2 octets	n = (0 à 254 octets)	"ABC"
WString	n+ 2 mots	n = (0 à 65534 mots)	"ä123@XYZ.COM"

Char et WChar

Un Char occupe un octet en mémoire et contient un caractère unique codé au format ASCII, comprenant les codes de caractères ASCII étendus. Un WChar occupe un mot en mémoire et peut contenir n'importe quelle représentation de caractère à deux octets.

La syntaxe de l'éditeur utilise une apostrophe avant et après le caractère. Vous pouvez utiliser des caractères visibles et des caractères de commande.

String et WString

La CPU prend en charge le type de données String pour sauvegarder une séquence de caractères d'un octet. Le type de données String contient le nombre de caractères total (nombre de caractères dans la chaîne) et le nombre de caractères en cours. Le type String fournit jusqu'à 256 octets pour le stockage du nombre de caractères total maximum (1 octet), du nombre de caractères en cours (1 octet) et de 254 caractères au maximum dans la chaîne. Chaque octet dans un type de données String peut avoir n'importe quelle valeur comprise entre 16#00 et 16#FF.

Le type de données WString offre des chaînes de valeurs à un mot (à deux octets) plus longues. Le premier mot contient le nombre de caractères total maximum ; le mot suivant contient le nombre total de caractères, et la chaîne suivante peut contenir jusqu'à 65 534 mots. Chaque mot dans un type de données WString peut avoir n'importe quelle valeur comprise entre 16#0000 et 16#FFFF.

Vous pouvez utiliser des chaînes littérales (constantes) pour les paramètres d'instruction de type IN avec des apostrophes. Par exemple, 'ABC' est une chaîne de trois caractères qui pourrait être utilisée en tant qu'entrée pour le paramètre IN de l'instruction S_CONV. Vous pouvez également créer des variables chaînes de caractères en sélectionnant le type de données "String" ou "WString" dans les éditeurs d'interface des blocs OB, FC, FB et DB. Vous ne pouvez pas créer de chaîne dans l'éditeur de variables API.

Vous pouvez indiquer la taille maximale d'une chaîne en octets (String) ou en mots (WString) à l'aide de crochets après le mot-clé "String" ou "WString" une fois que vous avez sélectionné l'un de ces types de données depuis la liste déroulante des types de données. "MaChaîne String[10]" indiquerait par exemple une taille maximale de 10 octets pour MaChaîne. Si vous ne mentionnez pas de taille maximale entre crochets, 254 est pris par défaut pour une chaîne et 65534 pour une WString. "MyWString WString[1000]" indiquerait une WString de 1 000 mots.

5.4 Types de données

L'exemple suivant définit une chaîne String ayant un nombre maximum de caractères égal à 10 et un nombre de caractères en cours égal à 3. Cela signifie que la chaîne contient actuellement 3 caractères à un octet, mais pourrait être allongée jusqu'à contenir 10 caractères à un octet.

Tableau 5- 36 Exemple de type de données String

Nombre total de caractères	Nombre de caractères en cours	Caractère 1	Caractère 2	Caractère 3	...	Caractère 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	...	Octet 11

L'exemple suivant définit une chaîne WString ayant un nombre maximum de caractères égal à 500 et un nombre de caractères en cours égal à 300. Cela signifie que la chaîne contient actuellement 300 caractères à un mot, mais pourrait être allongée jusqu'à contenir 500 caractères à un mot.

Tableau 5- 37 Exemple de type de données WString

Nombre total de caractères	Nombre de caractères en cours	Caractère 1	Caractères 2 à 299	Caractère 300	...	Caractère 500
500	300	'à' (16#0084)	Mots de caractères ASCII	'M' (16#004D)	...	-
Mot 0	Mot 1	Mot 2	Mots 3 à 300	Mot 301	...	Mot 501

Des caractères de commande ASCII peuvent être utilisés dans des données de type Char, Wchar, String et WString. Le tableau suivant présente des exemples de syntaxe avec caractères de commande.

Tableau 5- 38 Caractères de commande ASCII autorisés

Caractères de commande	Valeur ASCII hexa (Char)	Valeur ASCII hexa (WChar)	Fonction de commande	Exemples
\$L ou \$l	16#0A	16#000A	Changement de ligne	'\$LTexte', '\$0ATexte'
\$N ou \$n	16#0A et 16#0D	16#000A et 16#000D	Saut de ligne La nouvelle ligne montre deux caractères dans la chaîne.	'\$NTexte', '\$0A\$0DTexte'
\$P ou \$p	16#0C	16#000C	Changement de page	'\$PTexte', '\$0CTexte'
\$R ou \$r	16#0D	16#000D	Retour chariot	'\$RTexte', '\$0DTexte'
\$T ou \$t	16#09	16#0009	Tabulation	'\$TTexte', '\$09Texte'
\$\$	16#24	16#0024	Signe dollar	'100\$\$', '100\$24'
\$'	16#27	16#0027	Apostrophe	'\$'Texte\$', '\$27Texte\$27'

5.4.6 Type de données "tableau"

Tableaux

Vous pouvez créer un tableau qui contient plusieurs éléments de même type de données. Il est possible de créer des tableaux dans les éditeurs d'interface des blocs OB, FC, FB et DB. Vous ne pouvez pas créer de tableau dans l'éditeur de variables API.

Pour créer un tableau dans l'éditeur d'interface de bloc, donnez un nom au tableau et choisissez le type de données "Array [lo .. hi] of type", puis éditez "lo", "hi" et "type" de la manière suivante :

- lo : indice de départ (le plus bas) pour votre tableau
- hi : indice de fin (le plus élevé) pour votre tableau
- type : un type de données, tels que BOOL, SINT, UDINT

Tableau 5- 39 Règles pour le type de données ARRAY

Type de données	Syntaxe d'un tableau		
ARRAY	nom [indice1_min..indice1_max, indice2_min..indice2_max] of <type de données>		
	<ul style="list-style-type: none"> • Tous les éléments du tableau doivent avoir le même type de données. • L'indice peut être négatif, mais la limite inférieure doit être inférieure ou égale à la limite supérieure. • Un tableau peut avoir de une à six dimensions. • Les déclarations des indices min..max des différentes dimensions sont séparées par des virgules. • Les tableaux imbriqués (tableaux de tableaux) ne sont pas autorisés. • La taille d'un tableau en mémoire est égale à (taille d'un élément * nombre total d'éléments dans le tableau) 		
	Indice de tableau	Types de données autorisés pour les indices	Règles pour les indices de tableau
Constante ou variable	USInt, SInt, UInt, Int, UDInt, DInt	<ul style="list-style-type: none"> • Valeurs limites : -32768 à +32767 • Correct : constantes et variables mélangées • Correct : expressions constantes • Incorrect : expressions variables 	

Exemple : déclarations de tableaux ARRAY[1..20] of REAL

Une dimension, 20 éléments

ARRAY[-5..5] of INT

Une dimension, 11 éléments

ARRAY[1..2, 3..4] of CHAR

Deux dimensions, 4 éléments

Exemple : adresses de tableaux ARRAY1[0]

Tableau ARRAY1, élément 0

ARRAY2[1,2]

Tableau ARRAY2, élément [1,2]

ARRAY3[i,j]

Si i=3 et j=4, on adresse l'élément [3, 4] du tableau ARRAY3.

5.4.7 Type de données "structure de données"

Vous pouvez utiliser le type de données "Struct" pour définir une structure de données constituée de types de données différents. Le type de données Struct peut servir à gérer un groupe de données de processus apparentées en tant qu'unité de données unique. Vous donnez un nom à la structure et vous déclarez son organisation de données interne dans l'éditeur de bloc de données ou dans un éditeur d'interface de bloc.

Vous pouvez regrouper des tableaux et des structures dans une structure plus grande. Une structure peut être imbriquée jusqu'à huit niveaux de profondeur. Par exemple, vous pouvez créer une structure de structures qui contiennent des tableaux.

5.4.8 Type de données API

L'éditeur de type de données API vous permet de définir des structures de données utilisables à plusieurs reprises dans votre programme. Pour créer un type de données API, ouvrez la branche "Types de données API" de l'arborescence de projet et double-cliquez sur "Ajouter nouveau type de données". Dans le type de données API nouvellement créé, servez-vous de deux clics simples pour modifier le nom par défaut et d'un double clic pour ouvrir l'éditeur de type de données API.

Pour créer une structure de type de données API personnalisée, vous utilisez les mêmes méthodes d'édition que dans l'éditeur de bloc de données. Ajoutez de nouvelles lignes pour tout type de données dont vous avez besoin pour créer la structure de données voulue.

Si un nouveau type de données API est créé, son nom apparaît dans les listes déroulantes de sélection de type de données dans l'éditeur de DB et dans l'éditeur d'interface de bloc de code.

Les types de données API peuvent être utilisés comme suit :

- Comme types de données dans une interface de bloc de code ou dans des blocs de données
- Comme modèles pour la création de plusieurs blocs de données globaux utilisant la même structure de données
- Comme types de données pour des déclarations de variables API dans les zones de mémoire I et Q de la CPU

Un type de données API pourrait, par exemple, être une recette pour mélanger des couleurs. Vous pouvez alors affecter ce type de données API à plusieurs blocs de données, les variables dans chaque bloc de données étant ajustées de manière à créer une couleur spécifique.

5.4.9 Type de données pointeur variant

Le type de données Variant peut désigner des variables de différents types de données ou des paramètres. Le pointeur Variant peut désigner des structures et des éléments individuels de structures. Le pointeur Variant n'occupe aucun espace en mémoire.

Tableau 5- 40 Propriétés du pointeur Variant

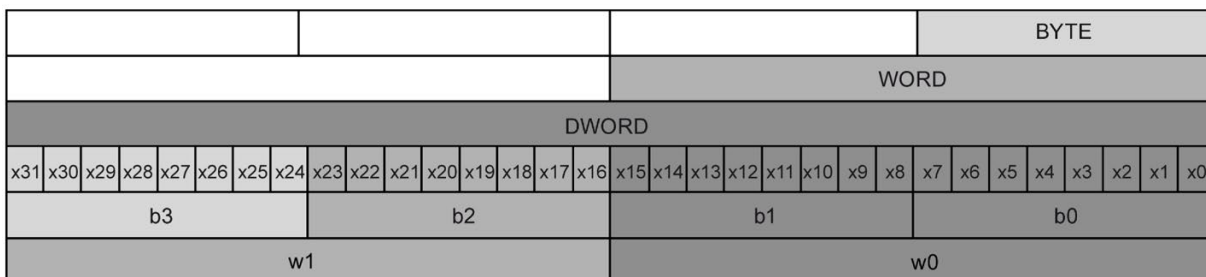
Longueur (octets)	Représentation	Format	Exemple de saisie
0	Symbolique	Opérande	MaVariable
		Nom_DB.Nom_struct.Nom_élément	MonDB.Struct1.pression1
	Absolue	Opérande	%MW10
		Numéro_DB.Opérande Type Longueur	P#DB10.DBX10.0 INT 12

5.4.10 Accès à une "tranche" d'un type de données de variable

Vous pouvez accéder aux variables API et aux variables de blocs de données au niveau bit, octet ou mot selon leur taille. Voici la syntaxe pour accéder à une telle tranche de données :

- "<nom variable API>".xn (accès bit)
- "<nom variable API>".bn (accès octet)
- "<nom variable API>".wn (accès mot)
- "<nom bloc de données>".<nom variable>.xn (accès bit)
- "<nom bloc de données>".<nom variable>.bn (accès octet)
- "<nom bloc de données>".<nom variable>.wn (accès mot)

Il est possible d'accéder à une variable double mot par les bits 0 à 31, les octets 0 à 3 ou les mots 0 et 1. Il est possible d'accéder à une variable mot par les bits 0 à 15, les octets 0 à 1 ou le mot 0. Il est possible d'accéder à une variable octet par les bits 0 à 7 ou l'octet 0. Les tranches bit, octet et mot peuvent être utilisées partout où on attend des bits, octets ou mots comme opérandes.



Remarque

Les types de données auxquels il est possible d'accéder par tranche sont les types Byte, Char, Conn_Any, Date, DInt, DWord, Event_Any, Event_Att, Hw_Any, Hw_Device, HW_Interface, Hw_Io, Hw_Pwm, Hw_SubModule, Int, OB_Any, OB_Att, OB_Cyclic, OB_Delay, OB_WHINT, OB_PCYCLE, OB_STARTUP, OB_TIMEERROR, OB_Tod, Port, Rtm, SInt, Time, Time_Of_Day, UDInt, UInt, USInt et Word. Il est possible d'accéder aux variables API de type Real par tranche, mais pas aux variables de blocs de données de type Real.

Exemples

Dans la table de variables API, "DW" est une variable déclarée de type DWORD. Les exemples suivants montrent l'accès à cette variable par tranche de bit, octet et mot :

	CONT	LOG	SCL
Accès bit			<pre>IF "DW".x11 THEN ... END_IF;</pre>
Accès octet			<pre>IF "DW".b2 = "DW".b3 THEN ... END_IF;</pre>
Accès mot			<pre>out:= "DW".w0 AND "DW".w1;</pre>

5.4.11 Accès à une variable par un type de données ajouté AT

L'ajout d'un type de données à une variable via AT vous permet d'accéder à une variable de bloc déjà déclarée avec une déclaration de type de données différent ajoutée. Vous pouvez, par exemple, accéder aux bits individuels d'une variable de type Byte, Word ou DWord au moyen d'un tableau de booléens (Array of Bool). Les ajouts de type de données par AT sont disponibles pour les types de variable suivants :

- Variables dans un bloc à accès standard
- Variables rémanentes dans un bloc optimisé

Déclaration

Pour ajouter un type de données à un paramètre, déclarez un paramètre supplémentaire directement après le paramètre concerné et sélectionnez le type de données "AT". L'éditeur crée l'ajout et vous pouvez alors choisir le type de données, la structure ou le tableau que vous voulez utiliser comme type de données ajouté.

Exemple

Cet exemple montre les paramètres d'entrée d'un FB à accès standard. Un tableau de booléens est ajouté comme type de données à la variable B1 de type octet :

B1	Byte	0.0
OV	AT*B1*	Array[0..7] of Bool
OV[0]	Bool	0.0
OV[1]	Bool	0.1
OV[2]	Bool	0.2
OV[3]	Bool	0.3
OV[4]	Bool	0.4
OV[5]	Bool	0.5
OV[6]	Bool	0.6
OV[7]	Bool	0.7

Dans cet autre exemple, une variable de type DWord reçoit une structure comme nouveau type de données. Cette structure comprend un mot, un octet et deux booléens :

DW1	DWord	2.0
DW1_Struct	AT*DW1*	Struct
W1	Word	0.0
B1	Byte	2.0
BO1	Bool	3.0
BO2	Bool	3.1

La colonne Décalage de l'interface de bloc montre l'emplacement des types de données ajoutés par rapport à la variable initiale.

Vous pouvez adresser les types de données ajoutés directement dans la logique du programme :

CONT	LOG	SCL
		<pre>IF #OV[1] THEN ... END_IF;</pre>
		<pre>IF #DW1_Struct.W1 = W#16#000C THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.B1;</pre>
		<pre>IF #OV[4] AND #DW1_Struct.BO2 THEN ... END_IF;</pre>

Règles

- L'ajout d'un type de données à des variables est possible dans les FB et FC à accès standard (non optimisé).
- Dans les FB et FC optimisés, l'ajout d'un type de données est possible pour les variables rémanentes.
- Vous pouvez ajouter un type de données à des paramètres pour tous les types de blocs et toutes les sections de déclaration.
- Vous pouvez utiliser un paramètre auquel on a ajouté un type de données comme n'importe quel autre paramètre du bloc.
- Vous ne pouvez pas ajouter de type de données aux paramètres de type VARIANT.
- La taille du paramètre d'ajout doit être inférieure ou égale à la taille du paramètre auquel on ajoute le type de données.
- Vous devez déclarer la variable d'ajout de type de données immédiatement après la variable à laquelle on ajoute le type de données et sélectionner le mot-clé "AT" comme sélection initiale du type de données.

5.5 Utilisation d'une carte mémoire

Remarque

La CPU accepte uniquement les cartes mémoire SIMATIC (Page 1638) préformatées.

Avant de copier un programme sur la carte mémoire formatée, effacez de la carte mémoire tout programme précédemment sauvegardé.

Vous pouvez utiliser une carte mémoire soit comme carte transfert, soit comme carte programme. Les cartes transfert et les cartes programme contiennent tous les blocs de code et de données, les objets technologiques éventuels et la configuration des appareils. Elles **ne contiennent pas** les tables de forçage permanent, les tables de visualisation ou les tables de variables API, par exemple.

- Utilisez une carte transfert (Page 147) pour copier un programme dans la mémoire de chargement interne de la CPU sans recourir à STEP 7.

Vous pouvez utiliser une carte transfert vide pour accéder à une CPU protégée par mot de passe en cas d'oubli du mot de passe (Page 157).

- Utilisez une carte programme (Page 150) comme mémoire de chargement externe pour la CPU.

Vous utiliserez également une carte mémoire pour télécharger les mises à jour du firmware (Page 153).

5.5.1 Insertion d'une carte mémoire dans la CPU

IMPORTANT

Protégez la carte mémoire et son emplacement contre les décharges électrostatiques

Des décharges électrostatiques peuvent endommager la carte mémoire ou son emplacement sur la CPU.

Pour éviter tout risque lorsque vous manipulez la carte mémoire, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette. Conservez la carte mémoire dans une boîte conductrice.



Vérifiez que la carte mémoire n'est pas protégée en écriture. Faites glisser le commutateur de protection pour l'éloigner de la position "verrou".

Notez que si vous insérez une carte mémoire protégée en écriture dans la CPU, STEP 7 le signalera en affichant un message de diagnostic lors de la mise sous tension suivante. La mise sous tension de la CPU s'achèvera sans problème, mais les instructions impliquant des recettes et des journaux, par exemple, renverront des erreurs si la carte est protégée en écriture.

 **ATTENTION**

Vérifiez qu'aucun processus ne s'exécute dans la CPU avant d'insérer la carte mémoire.

Si vous insérez une carte mémoire (qu'elle soit configurée comme carte programme, carte transfert ou comme carte de mise à jour du firmware) dans une CPU à l'état MARCHE, la CPU passe immédiatement à l'état ARRET, ce qui peut perturber le fonctionnement du processus et cela peut entraîner la mort ou des blessures graves.

Avant d'insérer ou de retirer une carte mémoire, assurez-vous toujours que la CPU ne pilote pas activement une machine ou un processus. Installez toujours un circuit d'arrêt d'urgence pour votre application ou votre processus.

Remarque

N'insérez pas de cartes transfert de programme V3.0 dans des CPU S7-1200 V4.x.

Les cartes transfert de programme de version 3.0 ne sont pas compatibles avec les CPU S7-1200 V4.x. Insérer une carte mémoire qui contient un programme V3.0 provoque une erreur de CPU.

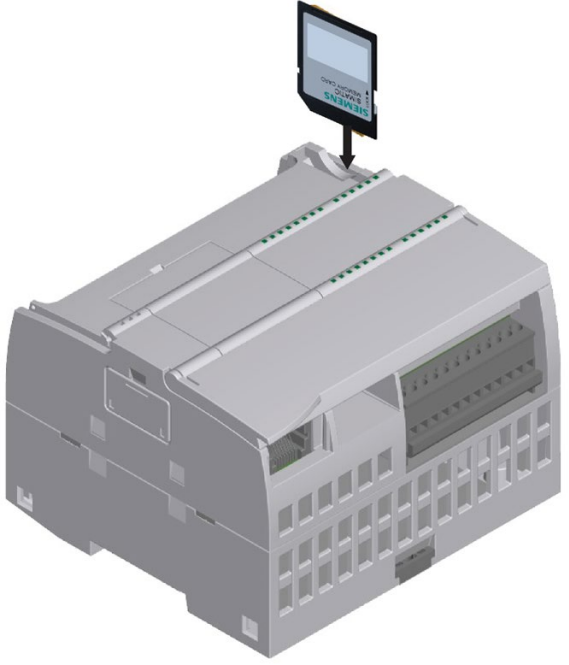
Si vous insérez une version invalide de carte transfert de programme (Page 147), retirez la carte, effectuez une transition de ARRET à MARCHE, un effacement général de la mémoire (MRES) ou mettez la CPU hors tension puis sous tension. Après avoir remédié à la situation d'erreur de la CPU, vous pouvez charger un programme CPU V4.x valide.

Pour remplacer un programme V3.0 par un programme V4.x, vous devez changer l'appareil dans la Configuration matérielle de TIA Portal.

Remarque

Si vous insérez une carte mémoire alors que la CPU est à l'état ARRET, la mémoire tampon de diagnostic affiche un message indiquant que l'évaluation de la carte mémoire a commencé. La CPU évaluera la carte mémoire la prochaine fois que vous ferez passer la CPU à l'état MARCHE, que vous effectuerez un effacement général de la CPU via MRES ou que vous mettrez la CPU hors tension puis sous tension.

Tableau 5- 41 Insertion d'une carte mémoire

	<p>Pour insérer une carte mémoire, ouvrez le volet supérieur de la CPU et insérez la carte mémoire dans la fente. Un connecteur de type "push-push" facilite l'insertion et l'extraction de la carte.</p> <p>La carte mémoire est dotée d'un détrompeur pour garantir une installation correcte.</p>
---	--

Comportement de la CPU lorsque vous insérez une carte mémoire

Lorsque vous insérez une carte mémoire dans la CPU, celle-ci exécute les opérations suivantes :

1. Elle passe à l'état ARRÊT (si elle n'y est pas déjà).
2. Elle affiche un message proposant l'un des choix suivants :
 - mise hors tension puis sous tension
 - passage à l'état MARCHE
 - exécution d'un effacement général
3. Elle évalue la carte.

Méthode d'évaluation de la carte mémoire par la CPU

Si vous n'avez pas configuré la CPU de manière à "inhiber la copie de la mémoire de chargement interne dans la mémoire de chargement externe" dans les propriétés de protection de la configuration d'appareil (Page 221), la CPU détermine quel type de carte mémoire vous avez inséré :

- **Carte mémoire vide** : Une carte mémoire vide ne contient pas de fichier de tâche (S7_JOB.S7S). Si vous insérez une carte mémoire vide, la CPU ajoute un fichier de tâche programme. Elle copie alors la mémoire de chargement interne dans la mémoire de chargement externe (le fichier programme sur la carte mémoire) et efface la mémoire de chargement interne.
- **Carte programme vide** : Une carte programme vide contient un fichier de tâche programme qui est vide. Dans ce cas, la CPU copie la mémoire de chargement interne dans la mémoire de chargement externe (le fichier programme sur la carte mémoire) et efface la mémoire de chargement interne.

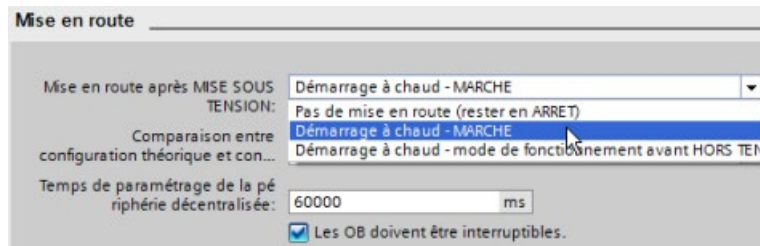
Si vous avez configuré la CPU de manière à "Inhiber la copie de la mémoire de chargement interne dans la mémoire de chargement externe" dans les propriétés de protection de la configuration d'appareil, la CPU se comporte comme suit :

- **Carte mémoire vide** : Une carte mémoire vide ne contient pas de fichier de tâche (S7_JOB.S7S). Si vous insérez une carte mémoire vide, la CPU ne fait rien. Elle ne crée pas de fichier de tâche programme et ne copie pas la mémoire de chargement interne dans la mémoire de chargement externe (le fichier programme sur la carte mémoire). Elle n'efface pas la mémoire de chargement interne.
- **Carte programme vide** : Une carte programme vide contient un fichier de tâche programme qui est vide. Dans ce cas, la CPU n'exécute aucune action. Elle ne copie pas la mémoire de chargement interne dans la mémoire de chargement externe (le fichier programme sur la carte mémoire). Elle n'efface pas la mémoire de chargement interne.

Si vous insérez dans la CPU une carte programme (Page 150), une carte transfert (Page 147) ou une carte contenant une mise à jour du firmware (Page 153), le paramètre de configuration pour "Inhiber la copie de la mémoire de chargement interne dans la mémoire de chargement externe" n'a pas d'effet sur la manière dont la CPU évalue la carte mémoire.

5.5.2 Configuration des paramètres de mise en route de la CPU avant copie du projet dans la carte mémoire

Lorsque vous copiez un programme dans une carte transfert ou une carte programme, le programme comprend les paramètres de mise en route pour la CPU. Assurez-vous toujours, avant de copier le programme dans la carte mémoire, que vous avez configuré l'état de fonctionnement de la CPU au retour de la tension. Choisissez si la CPU doit démarrer à l'état ARRET, à l'état MARCHE ou à l'état qu'elle avait avant la mise hors puis sous tension.



5.5.3 Utilisation de la carte mémoire en tant que carte "transfert"

IMPORTANT

Protégez la carte mémoire et son emplacement contre les décharges électrostatiques

Des décharges électrostatiques peuvent endommager la carte mémoire ou son emplacement sur la CPU.

Pour manipuler la carte mémoire en toute sécurité, veillez :

- à être en contact avec un tapis conducteur mis à la terre
- ou à porter un bracelet spécial avec chaînette mis à la terre.

Conservez la carte mémoire dans une boîte conductrice.

Création d'une carte transfert

Pensez à configurer les paramètres de démarrage de la CPU (Page 147) avant de copier un programme dans la carte transfert. Procédez comme suit pour créer une carte transfert :

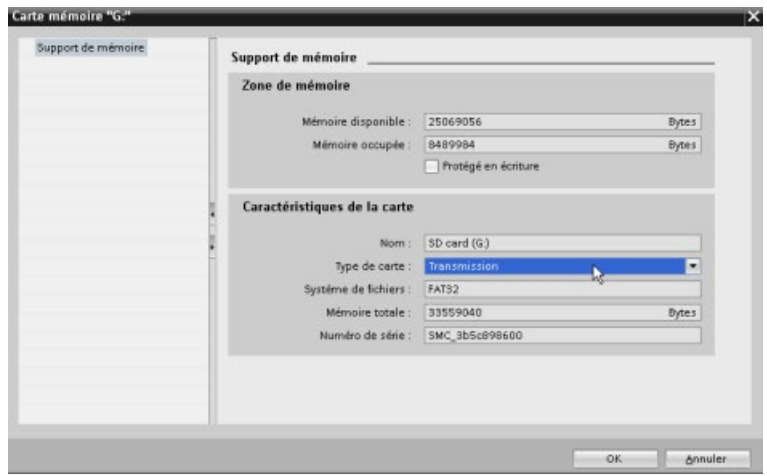
1. Insérez une carte mémoire SIMATIC vierge qui n'est pas protégée en écriture dans un lecteur/graveur de carte SD relié à votre ordinateur. (Si la carte est protégée en écriture, faites glisser le commutateur de protection pour l'éloigner de la position "verrou".)

Si vous réutilisez une carte mémoire SIMATIC contenant un programme utilisateur, des journaux, des recettes ou une mise à jour du firmware, vous **devez** effacer les fichiers programme avant de réutiliser la carte. Utilisez l'Explorateur Windows pour afficher le contenu de la carte mémoire et effacez le fichier "S7_JOB.S7S" ainsi que tout dossier existant (tel que "SIMATIC.S7S", "FWUPDATE.S7S", "Data Logs" et "Recipes").

IMPORTANT
N'effacez pas les fichiers cachés "__LOG__" et "crdinfo.bin" sur la carte mémoire.
Les fichiers "__LOG__" et "crdinfo.bin" sont requis pour la carte mémoire. Si vous supprimez ces fichiers, vous ne pourrez plus utiliser la carte mémoire avec la CPU.

2. Dans l'arborescence du projet (vue du projet), affichez le détail du dossier "SIMATIC Card Reader" et sélectionnez votre lecteur de carte.
3. Ouvrez la boîte de dialogue "Carte mémoire" en cliquant avec le bouton droit de la souris sur la lettre de lecteur correspondant à la carte mémoire dans le lecteur de carte et en sélectionnant "Propriétés" dans le menu contextuel.
4. Sélectionnez "Transfert" dans le menu déroulant "Type de carte" de la boîte de dialogue "Carte mémoire".

STEP 7 crée alors la carte transfert vide. Si vous créez une carte transfert vide, par exemple pour restaurer une CPU dont vous avez oublié le mot de passe (Page 157), retirez la carte transfert du lecteur de carte.



5. Ajoutez le programme en sélectionnant la CPU (par exemple, API_1 [CPU 1214C DC/DC/DC]) dans l'arborescence du projet et en la faisant glisser dans la carte mémoire. Une autre méthode consiste à copier-coller la CPU dans la carte mémoire. Copier la CPU dans la carte mémoire ouvre la boîte de dialogue "Aperçu du chargement".

6. Dans la boîte de dialogue "Aperçu du chargement", cliquez sur le bouton "Charger" pour copier la CPU dans la carte mémoire.
7. Cliquez sur le bouton "Terminer" lorsqu'un message vous signale que la CPU (programme) a été chargée sans erreur.

Utilisation d'une carte transfert

ATTENTION

Vérifiez qu'aucun processus n'est actif dans la CPU avant d'insérer la carte mémoire.

L'insertion d'une carte mémoire fait passer la CPU à l'état ARRET, ce qui peut affecter le fonctionnement d'une machine ou d'un processus en ligne avec comme conséquence éventuelle la mort, des blessures graves et des dommages matériels importants.

Avant d'insérer une carte transfert, assurez-vous que la CPU est à l'état ARRET et que votre processus est dans un état sûr.

Remarque

N'insérez pas de cartes transfert de programme V3.0 dans des modèles de CPU ultérieurs.

Les cartes transfert de programme de version 3.0 ne sont pas compatibles avec les modèles de CPU S7-1200 ultérieurs. Insérer une carte mémoire qui contient un programme V3.0 provoque une erreur de CPU.

Si vous insérez une version invalide de carte transfert de programme, retirez la carte, effectuez une transition de ARRET à MARCHE, un effacement général de la mémoire (MRES) ou mettez la CPU hors tension puis sous tension. Après avoir remédié à la situation d'erreur de la CPU, vous pouvez charger un programme CPU valide.

Procédez comme suit pour transférer le programme dans une CPU :

1. Insérez la carte transfert dans la CPU (Page 143). Si la CPU est à l'état MARCHE, elle passe à l'état ARRET. La DEL de maintenance (MAINT) clignote pour signaler que la carte mémoire doit être évaluée. À cet instant, le programme existant est encore dans la CPU.
2. Mettez la CPU hors tension puis sous tension pour évaluer la carte mémoire. Les autres méthodes pour redémarrer la CPU consistent à exécuter une transition ARRET-MARCHE ou un effacement général (MRES) à partir de STEP 7.
3. Après le redémarrage, la CPU évalue la carte mémoire et copie le programme dans sa mémoire de chargement interne.

La DEL RUN/STOP clignote alternativement en vert et en jaune pour indiquer que le programme est en cours de copie. L'opération de copie est achevée lorsque la DEL RUN/STOP passe au feu fixe (jaune) et que la DEL MAINT clignote (en jaune). Vous pouvez alors retirer la carte mémoire.

4. Redémarrez la CPU (en la remettant sous tension ou par l'une des méthodes alternatives de redémarrage) pour évaluer le nouveau programme transféré en mémoire de chargement interne.

La CPU passe alors au mode (MARCHE ou ARRET) que vous avez configuré pour la mise en route dans le projet.

Remarque

Vous devez retirer la carte transfert avant de faire passer la CPU à l'état MARCHE.

5.5.4 Utilisation de la carte mémoire en tant que carte "programme"

IMPORTANT

Des décharges électrostatiques peuvent endommager la carte mémoire ou son emplacement sur la CPU.

Pour éviter tout risque lorsque vous manipulez la carte mémoire, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette. Conservez la carte mémoire dans une boîte conductrice.



Vérifiez que la carte mémoire n'est pas protégée en écriture. Faites glisser le commutateur de protection pour l'éloigner de la position "verrou".

Avant de copier des éléments de programme sur la carte programme, effacez de la carte mémoire tout programme précédemment sauvegardé.

Création d'une carte programme

Lorsque vous l'utilisez comme carte programme, la carte mémoire constitue la mémoire de chargement externe de la CPU. Si vous retirez la carte programme, la mémoire de chargement interne de la CPU est vide.

Remarque

Si vous insérez une carte mémoire vierge dans la CPU et effectuez une évaluation de carte mémoire, soit en mettant la CPU hors tension puis sous tension, en effectuant un passage de l'état ARRET à l'état MARCHE ou en exécutant un effacement général (MRES), le programme et les valeurs de forçage dans la mémoire de chargement interne de la CPU sont copiés dans la carte mémoire. La carte mémoire est maintenant une carte programme. Une fois la copie achevée, le programme dans la mémoire de chargement interne de la CPU est effacé. La CPU passe alors au mode de mise en route configuré (MARCHE ou ARRET).

Pensez toujours à configurer les paramètres de mise en route de la CPU (Page 147) avant de copier un projet dans la carte programme. Procédez comme suit pour créer une carte programme :

1. Insérez une carte mémoire SIMATIC vierge qui n'est pas protégée en écriture dans un lecteur/graveur de carte SD relié à votre ordinateur. (Si la carte est protégée en écriture, faites glisser le commutateur de protection pour l'éloigner de la position "verrou".)

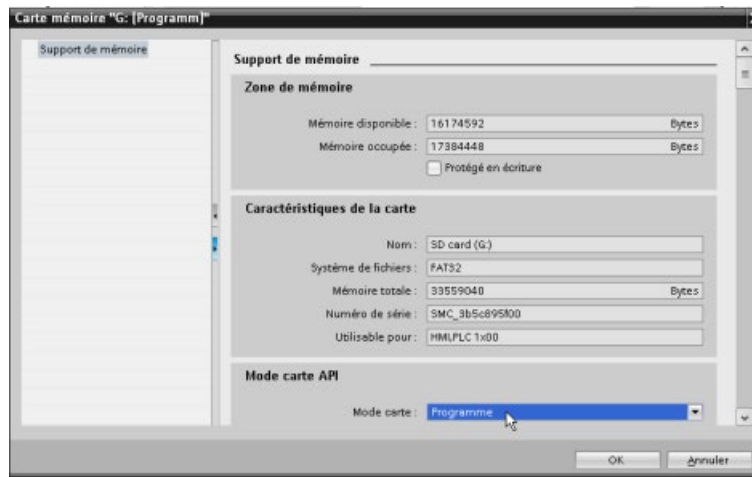
Si vous réutilisez une carte mémoire SIMATIC contenant un programme utilisateur, des journaux, des recettes ou une mise à jour du firmware, vous **devez** effacer les fichiers programme avant de réutiliser la carte. Utilisez l'Explorateur Windows pour afficher le contenu de la carte mémoire et effacez les fichiers et dossiers suivants le cas échéant :

- S7_JOB.S7S
- SIMATIC.S7S
- FWUPDATE.S7S
- DataLogs
- Recipes

IMPORTANT
N'effacez pas les fichiers cachés "__LOG__" et "crdinfo.bin" sur la carte mémoire.
Les fichiers "__LOG__" et "crdinfo.bin" sont requis pour la carte mémoire. Si vous supprimez ces fichiers, vous ne pourrez plus utiliser la carte mémoire avec la CPU.

2. Dans l'arborescence du projet (vue du projet), affichez le détail du dossier "Card Reader/Mémoire USB" et sélectionnez votre lecteur de carte.
3. Ouvrez la boîte de dialogue "Carte mémoire" en cliquant avec le bouton droit de la souris sur la lettre de lecteur correspondant à la carte mémoire dans le lecteur de carte et en sélectionnant "Propriétés" dans le menu contextuel.

4. Sélectionnez "Programme" dans le menu contextuel de la boîte de dialogue "Carte mémoire".



5. Ajoutez le programme en sélectionnant la CPU (par exemple, API_1 [CPU 1214C DC/DC/DC]) dans l'arborescence du projet et en la faisant glisser dans la carte mémoire. Une autre méthode consiste à copier-coller la CPU dans la carte mémoire. Copier la CPU dans la carte mémoire ouvre la boîte de dialogue "Aperçu du chargement".
6. Dans la boîte de dialogue "Aperçu du chargement", cliquez sur le bouton "Charger" pour copier la CPU dans la carte mémoire.
7. Cliquez sur le bouton "Terminer" lorsqu'un message vous signale que la CPU (programme) a été chargée sans erreur.

Utilisation d'une carte programme comme mémoire de chargement pour votre CPU

ATTENTION

Risques lors de l'insertion d'une carte programme

Vérifiez qu'aucun processus n'est actif dans la CPU avant d'insérer la carte mémoire.

L'insertion d'une carte mémoire fait passer la CPU à l'état ARRET, ce qui peut affecter le fonctionnement d'une machine ou d'un processus en ligne avec comme conséquence éventuelle la mort, des blessures graves et des dommages matériels importants.

Avant d'insérer une carte mémoire, assurez-vous que la CPU est hors ligne et dans un état sûr.

Procédez comme suit pour utiliser une carte programme avec votre CPU :

1. Insérez la carte programme dans la CPU. Si la CPU est à l'état MARCHE, elle passe à l'état ARRET. La DEL de maintenance (MAINT) clignote pour signaler que la carte mémoire doit être évaluée.
2. Mettez la CPU hors tension puis sous tension pour évaluer la carte mémoire. Les autres méthodes pour redémarrer la CPU consistent à exécuter une transition ARRET-MARCHE ou un effacement général (MRES) à partir de STEP 7.
3. Après que la CPU a redémarré et a évalué la carte programme, elle efface sa mémoire de chargement interne.

La CPU passe alors au mode (MARCHE ou ARRET) que vous avez configuré pour la mise en route de la CPU.

La carte programme doit rester dans la CPU. Retirer la carte programme laisse la CPU sans aucun programme dans la mémoire de chargement interne.

ATTENTION

Risques lors du retrait d'une carte programme

Si vous retirez la carte programme, la CPU perd sa mémoire de chargement externe et génère une erreur. La CPU passe à l'état ARRET et la DEL de défaut se met à clignoter.

Les appareils de commande peuvent présenter des défaillances dans des situations non sûres et provoquer un fonctionnement inattendu des appareils pilotés pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

En retirant la carte programme, sachez que vous retirez le programme de la CPU.

5.5.5 Mise à jour du firmware

Vous pouvez utiliser une carte mémoire SIMATIC pour effectuer une mise à jour du firmware.

IMPORTANT

Protégez la carte mémoire et son emplacement contre les décharges électrostatiques

Des décharges électrostatiques peuvent endommager la carte mémoire ou son emplacement sur la CPU.

Pour éviter tout risque lorsque vous manipulez la carte mémoire, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette. Conservez la carte mémoire dans une boîte conductrice.

Vous utilisez une carte mémoire SIMATIC pour télécharger les mises à jour du firmware depuis Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/fr>). À partir de ce site web, naviguez dans "Téléchargements". De là, recherchez le type spécifique de module que vous devez mettre à jour.

Vous pouvez aussi accéder directement à la page Web des téléchargements S7-1200 (<https://support.industry.siemens.com/cs/ww/fr/ps/13683/dl>).

Remarque

Vous ne pouvez pas mettre à jour une CPU S7-1200 V3.0 ou de version antérieure à S7-1200 V4.0 ou V4.1 par mise à jour de firmware.

Vous pouvez également mettre à jour le firmware au moyen de l'une des autres méthodes suivantes :

- Utilisation des Outils en ligne et de diagnostic de STEP 7 (Page 1389)
- Utilisation de la page Web standard "Information sur les modules" du serveur Web (Page 1060)
- Utilisation de SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/fr/view/98161300>)

IMPORTANT

N'utilisez pas l'utilitaire de formatage de Windows ou tout autre utilitaire de formatage pour reformater la carte mémoire.
--

Si une carte mémoire Siemens est reformatée avec l'utilitaire de formatage de Microsoft Windows, la carte mémoire ne peut plus être utilisée par une CPU S7-1200.

Pour charger la mise à jour du firmware sur votre carte mémoire, exécutez les étapes suivantes :

1. Insérez une carte mémoire SIMATIC vierge qui n'est pas protégée en écriture dans un lecteur/graveur de carte SD relié à votre ordinateur (si la carte est protégée en écriture, faites glisser le commutateur de protection pour l'éloigner de la position "verrou").

Si vous réutilisez une carte mémoire SIMATIC contenant un programme utilisateur ou une autre mise à jour du firmware, vous devez effacer un certain nombre de fichiers sur la carte mémoire.

Pour réutiliser une carte mémoire, vous **devez** effacer le fichier "S7_JOB.S7S" et tout dossier "Archives de variables" existant ou tout autre dossier (tel que "SIMATIC.S7S" ou "FWUPDATE.S7S") avant de télécharger la mise à jour du firmware. Utilisez Windows Explorer pour afficher le contenu de la carte mémoire et effacer le fichier ou les dossiers.

IMPORTANT

N'effacez pas les fichiers cachés "__LOG_" et "crdinfo.bin" sur la carte mémoire.

Les fichiers "__LOG_" et "crdinfo.bin" sont nécessaires pour la carte mémoire. Si vous supprimez ces fichiers, vous ne pourrez plus utiliser la carte mémoire avec la CPU.

2. Sélectionnez le fichier zip de la mise à jour du firmware correspondant à votre module et chargez-le sur votre ordinateur. Double-cliquez sur le fichier, sélectionnez le chemin cible qui sera pris comme répertoire racine de la carte mémoire SIMATIC et lancez l'extraction. Une fois l'extraction terminée, le répertoire racine de la carte mémoire contiendra un répertoire "FWUPDATE.S7S" et le fichier "S7_JOB.S7S".
3. Ejectez la carte du lecteur/graveur de carte en toute sécurité.

Procédez comme suit pour installer la mise à jour du firmware :

 **ATTENTION**

Vérifiez qu'un processus ne soit pas en cours sur la CPU avant d'installer la mise à jour du firmware.

L'installation de la mise à jour du firmware fait passer la CPU à l'arrêt, ce qui peut affecter l'exécution d'un processus ou d'une machine en ligne, dont la mise en route inattendue peut entraîner la mort, des blessures graves et des dommages matériels importants.

Avant d'insérer la carte mémoire, assurez-vous que la CPU se trouve hors ligne et dans un état sûr.

1. Insérez la carte mémoire dans la CPU. Si la CPU est à l'état MARCHE, elle passe à l'état ARRET. La DEL de maintenance clignote pour signaler que la carte mémoire doit être évaluée.
2. Mettez la CPU hors tension puis sous tension avant de lancer la mise à jour du firmware. Les autres méthodes pour redémarrer la CPU consistent à exécuter une transition ARRET-MARCHE ou un effacement général (MRES) à partir de STEP 7.

Remarque

Pour achever la mise à niveau du firmware du module, assurez-vous que l'alimentation 24 V CC externe est maintenue pour le module.

Après le redémarrage de la CPU, la mise à jour du firmware démarre. La DEL RUN/STOP clignote alternativement en vert et en jaune pour indiquer que la mise à jour est copiée. Lorsque la DEL RUN/STOP passe au feu fixe (jaune) et que la DEL MAINT clignote, l'opération de copie est terminée. Vous devez alors retirer la carte mémoire.

3. Après avoir retiré la carte mémoire, redémarrez la CPU (soit en la remettant sous tension, soit par les méthodes alternatives de redémarrage) pour charger le nouveau firmware.

Le programme utilisateur et la configuration matérielle ne sont pas affectés par la mise à jour du firmware. Après sa mise sous tension, la CPU passe à l'état de mise en route configuré. (si le mode de mise en route paramétré pour votre CPU était "Démarrage à chaud dans le mode avant la mise hors tension", la CPU sera sur ARRET, car le dernier état de fonctionnement de la CPU était ARRET.)

Remarque

Mise à jour de multiples modules connectés à la CPU

Si votre configuration matérielle contient de multiples modules qui correspondent à un seul fichier de mise à jour du firmware sur la carte mémoire, la CPU applique les mises à jour à tous les modules appropriés (CM, SM et SB) dans l'ordre de configuration c'est-à-dire dans l'ordre croissant de la position des modules dans la configuration de l'appareil de STEP 7.

Si vous avez téléchargé plusieurs mises à jour du firmware sur la carte mémoire pour plusieurs modules, la CPU applique les mises à jour dans l'ordre dans lequel vous les avez téléchargées sur la carte mémoire.

5.6 Récupération en cas d'oubli du mot de passe

Si vous avez oublié le mot de passe d'une CPU protégée par mot de passe, utilisez une carte transfert vide pour effacer le programme protégé par mot de passe. La carte transfert vide efface la mémoire de chargement interne de la CPU. Vous pouvez alors charger un nouveau programme utilisateur de STEP 7 dans la CPU.

Pour plus d'informations sur la création et l'utilisation d'une carte transfert vide, reportez-vous au paragraphe sur les cartes transfert (Page 147).

ATTENTION

Vérifiez qu'aucun processus n'est actif dans la CPU avant d'insérer la carte mémoire

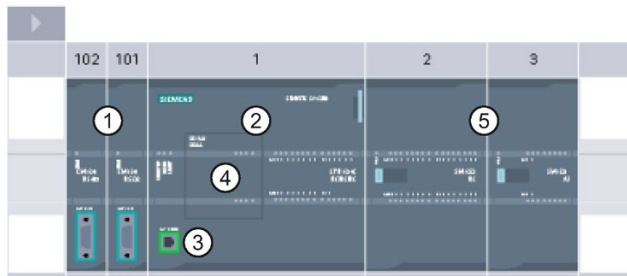
Si vous insérez une carte transfert dans une CPU à l'état MARCHE, la CPU passe à l'état ARRET. Les appareils de commande peuvent présenter des défaillances dans des situations non sûres et provoquer un fonctionnement inattendu des appareils pilotés pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Avant d'insérer une carte transfert, assurez-vous que la CPU est à l'état ARRET et que votre processus est dans un état sûr.

Vous devez retirer la carte transfert avant de faire passer la CPU à l'état MARCHE.

Configuration des appareils

Vous créez la configuration des appareils pour votre automate en ajoutant une CPU et des modules supplémentaires à votre projet.



- ① Module de communication (CM) et processeur de communication (CP) : jusqu'à 3, insérés aux emplacements 101, 102 et 103
- ② CPU : emplacement 1
- ③ Port PROFINET de la CPU
- ④ Signal Board (SB), Communication Board (CB) ou Battery Board (BB) : au plus 1, inséré dans la CPU
- ⑤ Module d'entrées-sorties (SM) TOR ou analogiques : jusqu'à 8, insérés aux emplacements 2 à 9
(les CPU 1214C, CPU 1215C et CPU 1217C en autorisent 8, la CPU 1212C en autorise 2, la CPU 1211C n'en autorise aucun)

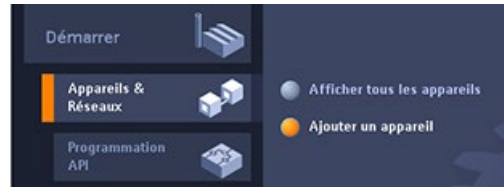
Commande de configuration

La configuration d'appareil du S7-1200 accepte également une "commande de configuration (Page 165)" où vous pouvez configurer une configuration maximum pour un projet comprenant les modules que vous pourriez ne pas utiliser en réalité. Cette fonction, parfois également appelée "gestion des options", vous permet de configurer une configuration maximum que vous pourriez utiliser avec des variations dans les modules installés dans des applications multiples.

6.1 Insertion d'une CPU

Vous pouvez insérer une CPU dans votre projet depuis la vue du portail ou la vue du projet de STEP 7 :

- Dans la vue du portail, sélectionnez "Appareils & réseaux" et cliquez sur "Ajouter un appareil".
- Dans la vue du projet, double-cliquez sur "Ajouter un appareil" sous le nom de projet.

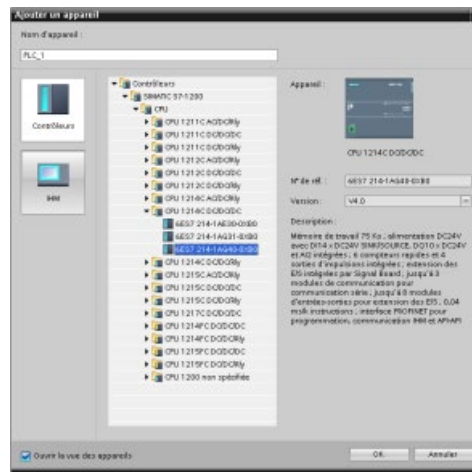


Veillez à bien insérer le modèle et la version de firmware corrects à partir de la liste. Sélectionner la CPU dans la boîte de dialogue "Ajouter un appareil" crée le châssis et la CPU.

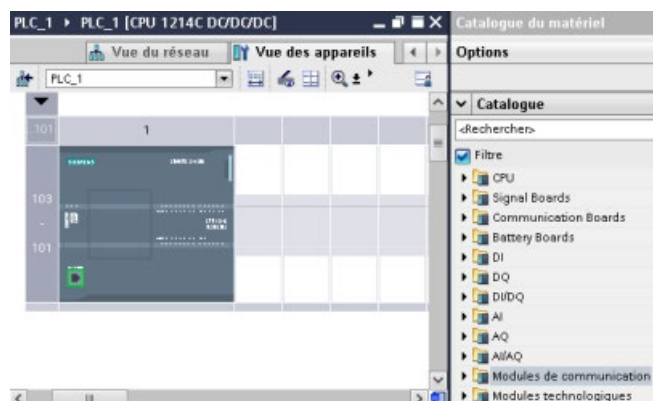
Remarque

Avec les versions V14 et ultérieures de STEP 7, vous ne pouvez pas ajouter de CPU S7-1200 V1.0 à votre projet.

Boîte de dialogue "Ajouter un appareil"

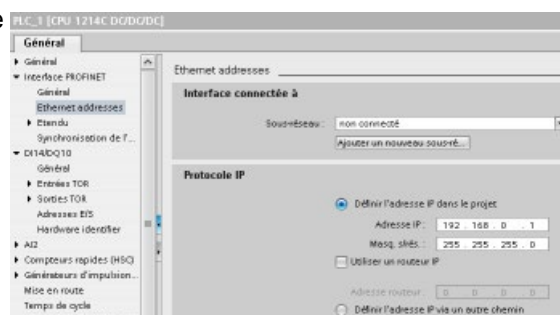


Vue des appareils de la configuration matérielle



Sélectionner la CPU dans la vue des appareils provoque l'affichage des propriétés de la CPU dans la fenêtre d'inspection.

La CPU ne dispose pas d'une adresse IP préconfigurée. Vous devez affecter manuellement une adresse IP à la CPU pendant la configuration des appareils. Si votre CPU est connectée à un routeur sur le réseau, vous entrez aussi l'adresse IP du routeur.



6.2 Chargement de la configuration d'une CPU connectée

STEP 7 offre deux méthodes de chargement de la configuration matérielle d'une CPU connectée :

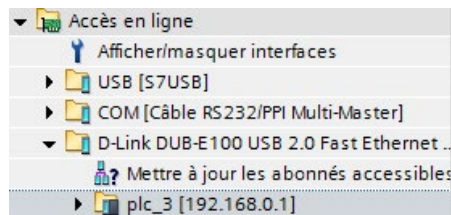
- Chargement de l'appareil connecté comme un nouveau poste
- Configuration d'une CPU non spécifiée et détection de la configuration matérielle de la CPU connectée

À noter, toutefois, que la première méthode charge à la fois la configuration matérielle et le logiciel de la CPU connectée.

Chargement d'un appareil comme un nouveau poste

Pour charger un appareil connecté comme un nouveau poste, suivez les étapes ci-dessous :

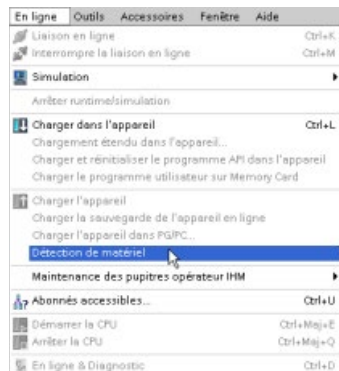
1. Affichez votre interface de communication depuis le nœud "Accès en ligne" de l'arborescence du projet.
2. Double-cliquez sur "Mettre à jour les abonnés accessibles".
3. Sélectionnez l'API dans les appareils détectés.



4. Dans le menu En ligne de STEP 7, sélectionnez la commande de menu "Charger l'appareil comme un nouveau poste (matériel et logiciel)".

STEP 7 charge à la fois la configuration matérielle et les blocs de programme.

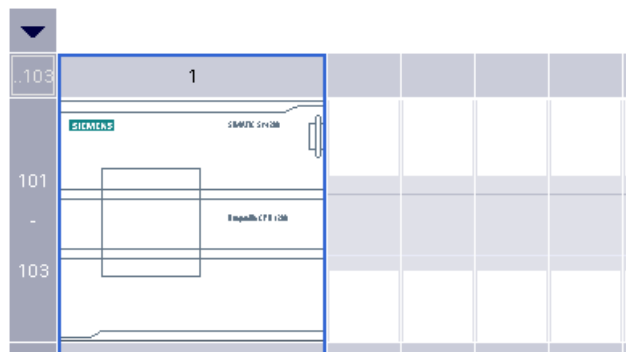
Détection de la configuration matérielle d'une CPU non spécifiée



Lorsque vous êtes connecté à une CPU, vous pouvez charger la configuration de cette CPU, y compris tous les modules, dans votre projet. Il vous suffit de créer un nouveau projet et de sélectionner la "CPU non spécifiée" au lieu d'une CPU spécifique (vous pouvez également ne pas passer par la configuration matérielle en sélectionnant "Créer un programme API" dans "Mise en route". STEP 7 crée ensuite automatiquement une CPU non spécifiée.)

Dans l'éditeur de programmes, vous sélectionnez la commande "Détection du matériel" du menu "En ligne".

Dans l'éditeur de configuration des appareils, vous sélectionnez l'option de détection de configuration de l'appareil connecté.

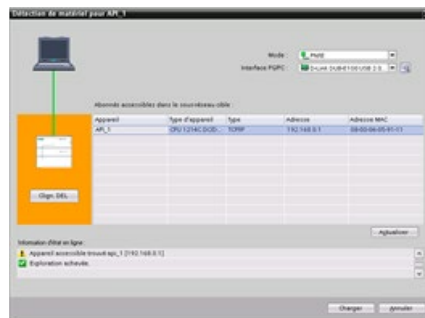


L'appareil n'est pas spécifié.

→ Veuillez utiliser [Catalogue du matériel](#) pour spécifier la CPU.

→ ou [déterminez](#) la configuration de l'appareil connecté.

Une fois que vous avez sélectionné la CPU dans la boîte de dialogue en ligne et cliqué sur le bouton Charger, STEP 7 télécharge la configuration matérielle depuis la CPU, en incluant tous les modules (SM, SB ou CM). Vous pouvez alors configurer les paramètres de la CPU et des modules (Page 177).



6.3 Ajout de modules à la configuration

Utiliser le catalogue de matériel pour ajouter des modules à la CPU :

- Le module d'entrées-sorties (SM) fournit des points d'E/S numériques ou analogiques supplémentaires. Ces modules se raccordent sur le côté droit de la CPU.
- Le Signal Board (SB) fournit seulement quelques points d'E/S supplémentaires pour la CPU. Le SB s'installe à l'avant de la CPU.
- Le Battery Board 1297 fournit une sauvegarde à long terme de l'horloge temps réel. Il s'installe à l'avant de la CPU.
- Le Communication Board (CB) fournit un port de communication supplémentaire (comme RS485). Le Communication Board s'installe à l'avant de la CPU.
- Le module de communication (CM) et le processeur de communication (CP) fournissent un port de communication supplémentaire, comme PROFIBUS ou GPRS. Ces modules se raccordent sur le côté gauche de la CPU.

Pour insérer un module dans la configuration des appareils, sélectionnez le module concerné dans le catalogue du matériel et double-cliquez ou faites glisser le module dans l'emplacement mis en évidence. Pour que les modules soient opérationnels, vous devez les ajouter à la configuration des appareils et charger dans la CPU la configuration matérielle pour les modules.

Tableau 6- 1 Ajouter un module à la configuration de l'appareil

Module	Sélectionnez le module	Insérez le module	Résultat
SM			
SB, BB ou CB			
CM ou CP			

Grâce à la fonction "commande de configuration" (Page 165), vous pouvez ajouter des modules d'entrées-sorties et des signal boards à votre configuration d'appareil qui pourraient ne pas correspondre au matériel réel d'une application spécifique, mais qui seront utilisés dans des applications associées qui partagent un programme utilisateur commun, un modèle de CPU et peut-être certains des modules configurés.

6.4 Commande de configuration

6.4.1 Avantages et applications d'une commande de configuration

Une commande de configuration peut être une solution utile lorsque vous créez une solution d'automatisation (machine) que vous prévoyez d'utiliser avec des adaptations dans des installations multiples.

Vous pouvez charger un programme utilisateur et une configuration d'appareil STEP 7 vers différentes configurations API installées. Vous avez simplement à effectuer quelques adaptations simples pour faire correspondre le projet STEP 7 à l'installation réelle.

6.4.2 Configuration de l'installation centrale et de modules optionnels

La commande de configuration avec STEP 7 et le S7-1200 vous permet de configurer une configuration maximum pour une machine standard et de faire fonctionner des versions (options) qui utilisent un sous-ensemble de cette configuration. Le manuel PROFINET avec STEP 7 (<https://support.industry.siemens.com/cs/ww/fr/view/49948856>) appelle ces types de projets des "projets de machine standard".

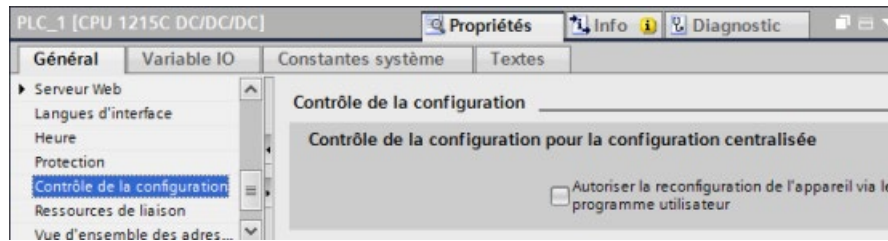
Un enregistrement de données de commande que vous programmez dans le bloc de programme de démarrage informe la CPU des modules manquants dans l'installation réelle par rapport à la configuration, ou des modules qui sont situés à des emplacements différents par rapport à la configuration. La commande de configuration n'a pas d'impact sur le paramétrage des modules.

La commande de configuration vous offre la possibilité de modifier l'installation à condition que vous puissiez déduire la configuration réelle de la configuration maximum de l'appareil dans STEP 7.

Pour activer la commande de configuration et structurer l'enregistrement de commande requis, suivez les étapes ci-dessous :

1. En option, réinitialisez la CPU avec les réglages d'usine pour garantir qu'un enregistrement de commande incompatible ne soit pas présent dans la CPU.
2. Sélectionnez la CPU dans la configuration d'appareil dans STEP 7.

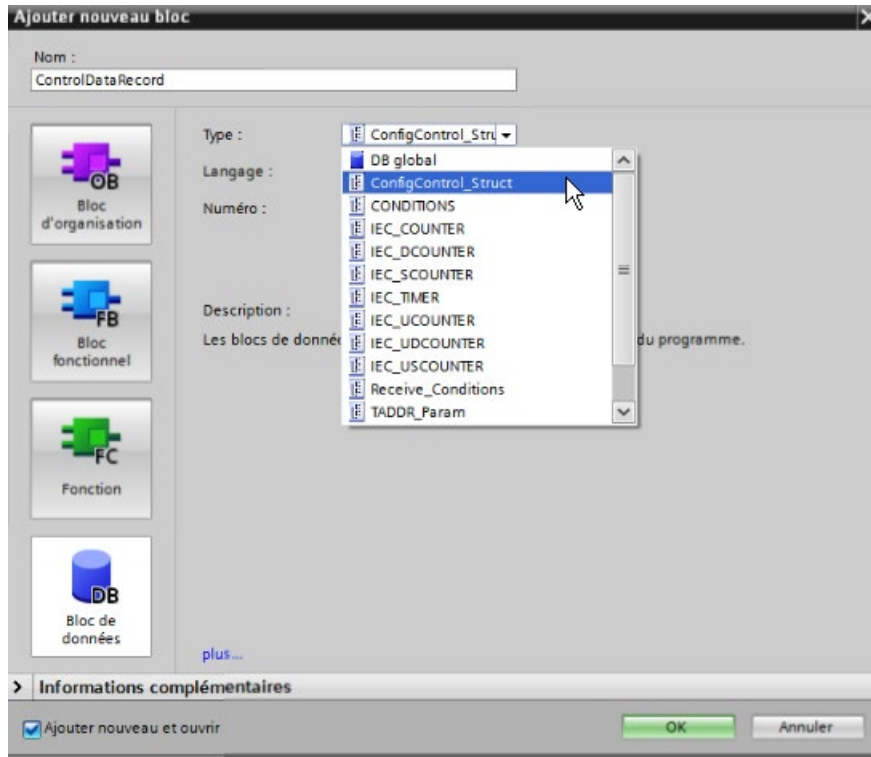
- Depuis le nœud Commande de configuration dans les propriétés de la CPU, cochez la case "Permettre la reconfiguration de l'appareil avec le programme utilisateur".



- Créez un type de données API pour contenir l'enregistrement de commande. Configurez-le comme une structure comprenant quatre USInt pour les informations de la commande de configuration et des USInt supplémentaires pour correspondre aux emplacements d'une configuration maximum d'appareil S7-1200, de la manière suivante :

ConfigControl_Struct				
	Nom	Type de données	Valeur par déf.	Commentaire
1	▼ ConfigControl	Struct		
2	■ Block_length	USInt	16	Length of control data record, including header
3	■ Block_ID	USInt	196	Data record number
4	■ Version	USInt	5	
5	■ Subversion	USInt	0	
6	■ Slot_1	USInt	255	Assignment for CPU annex card/Actual annex card
7	■ Slot_2	USInt	255	Configured slot 2 / Assigned "real" slot
8	■ Slot_3	USInt	255	Configured slot 3 / Assigned "real" slot
9	■ Slot_4	USInt	255	Configured slot 4 / Assigned "real" slot
10	■ Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
11	■ Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
12	■ Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
13	■ Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
14	■ Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
15	■ Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
16	■ Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
17	■ Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

5. Créez un bloc de données du type de données API que vous avez créé.



6. Dans ce bloc de données, configurez Block_length, Block_ID, Version, et Subversion comme indiqué ci-dessous. Configurez les valeurs des emplacements en fonction de leur présence ou de leur absence et de leur position dans votre installation réelle :
 - 0 : le module configuré n'est pas présent dans la configuration réelle. (L'emplacement est vide.)
 - 1 à 9, 101 à 103 : Position réelle de l'emplacement pour l'emplacement configuré
 - 255 : La configuration de l'appareil STEP 7 ne comprend pas de module à cet emplacement.

Remarque

Contrôle de la configuration pas disponible pour les HSC et PTO sur le Signal Board

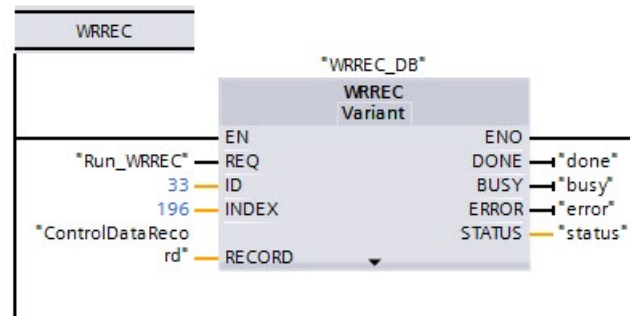
Si vous avez dans la CPU un Signal Board que vous avez configuré pour les HSC ou PTO, vous ne devez pas le désactiver avec un "0" à l'emplacement 1 de l'enregistrement de contrôle de la configuration. Les appareils HSC et PTO configurés de la CPU sont obligatoires en ce qui concerne le contrôle de la configuration.

ControlDataRecord				
	Nom	Type de données	Valeur de départ	Commentaire
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/ Actual annex...
8	Slot_2	USInt	255	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	255	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	255	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

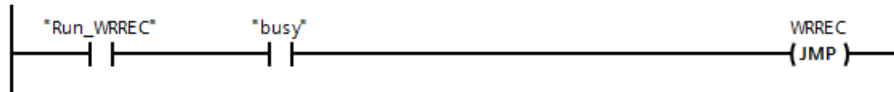
Pour obtenir une explication sur la manière d'attribuer les valeurs des emplacements, voir Exemple de commande de configuration (Page 173).

7. Dans l'OB de démarrage, appelez l'instruction WRREC (Écrire l'enregistrement) avancée pour transférer l'enregistrement de commande que vous avez créé dans l'indice 196 de l'ID matérielle 33. Utilisez un repère et une instruction JMP (Saut) pour attendre que l'instruction WRREC soit terminée.

Réseau 1 :



Réseau 2 :



Remarque

Le contrôle de la configuration n'est pas en vigueur tant que l'instruction WRREC n'a pas transféré l'enregistrement de commande dans l'OB de démarrage. Si vous avez activé le contrôle de la configuration et que la CPU ne dispose pas de l'enregistrement de commande, elle passera à l'état ARRÊT lorsqu'elle quittera l'état MISE EN ROUTE. Veillez à programmer l'OB de démarrage pour pouvoir transférer l'enregistrement de commande.

Agencement des modules

Le tableau suivant représente l'attribution des numéros d'emplacement :

Emplacement	Modules
1	Signal board ou communication board (carte annexe de la CPU)
2 à 9	Modules d'entrées-sorties
101 à 103	Modules de communication

Enregistrement de commande

Un enregistrement de commande 196 contient l'attribution d'emplacements et représente la configuration réelle, comme qu'indiqué ci-dessous :

Octet	Élément	Valeur	Explication
0	Longueur de bloc	16	En-tête
1	ID de bloc	196	
2	Version	5	
3	Sous-version	0	
4	Attribution de carte annexe de la CPU	Carte annexe réelle, 0, ou 255*	Élément de commande Décrit dans chaque élément quel emplacement réel dans l'appareil est attribué à l'emplacement configuré.
5	Attribution de l'emplacement configuré 2	Emplacement réel, 0, ou 255*	
...	
12	Attribution de l'emplacement configuré 9	Emplacement réel, 0, ou 255*	A la différence des modules d'entrées-sorties, l'emplacement réel des modules de communication physiquement présents doit être le même que l'emplacement configuré.
13	Attribution de l'emplacement configuré 101	Emplacement réel ou 255*	
14	Attribution de l'emplacement configuré 102	Emplacement réel ou 255*	
15	Attribution de l'emplacement configuré 103	Emplacement réel ou 255*	

***Valeurs des emplacements :**

0 : le module configuré n'est pas présent dans la configuration réelle. (L'emplacement est vide.)

1 à 9, 101 à 103 : Position réelle de l'emplacement pour l'emplacement configuré

255 : La configuration de l'appareil STEP 7 ne comprend pas de module à cet emplacement.

Remarque

Alternative pour créer un type de variable API

Comme alternative de création d'un type de variable API personnalisé, vous pouvez créer un bloc de données directement avec tous les éléments de structure d'un enregistrement de commande. Vous pourriez même configurer de multiples structures dans ce bloc de données afin qu'elles servent comme configurations d'enregistrement de commande multiples. L'une ou l'autre implémentation est une façon efficace de transférer l'enregistrement de commande pendant le démarrage.

Règles

Observez les règles suivantes :

- La commande de configuration n'accepte pas les changements de position concernant les modules de communication. Les positions des emplacements d'enregistrements de commande pour les emplacements 101 à 103 doivent correspondre à l'installation réelle. Si vous n'avez pas configuré un module pour l'emplacement dans votre configuration d'appareil, entrez 255 pour cette position d'emplacement dans l'enregistrement de commande. Si vous avez configuré un module pour l'emplacement, indiquez l'emplacement configuré comme emplacement réel pour cette position d'emplacement.
- Les modules F-E/S ne prennent pas en charge la commande de configuration. Les positions des emplacements d'enregistrement de commande d'un module F-E/S doivent égaler la position d'emplacement configurée pour le module F-E/S. Si vous tentez de déplacer ou de supprimer un module F-E/S en utilisant l'enregistrement de commande, tous les modules F-E/S effectivement installés signaleront une erreur de "paramétrage" et rejeteront les échanges.
- Vous ne pouvez pas avoir des emplacements vides (inutilisés) insérés entre des emplacements remplis (utilisés). Par exemple, si la configuration réelle a un module à l'emplacement 4, elle doit également avoir des modules aux emplacements 2 et 3. De même, si la configuration réelle a un module de communication à l'emplacement 102, elle doit également avoir un module à l'emplacement 101.
- Si vous avez activé la commande de configuration, la CPU n'est pas prête à fonctionner sans enregistrement de commande. La CPU repasse de mise en route à ARRET si un OB de démarrage ne transfère pas un enregistrement de commande valide. La CPU n'initialise pas les E/S centrales dans ce cas et entre la cause de l'état ARRET dans la mémoire tampon de diagnostic.
- La CPU sauvegarde un enregistrement de commande transféré avec succès dans la mémoire rémanente, ce qui signifie qu'il n'est pas nécessaire d'écrire à nouveau l'enregistrement de commande 196 lors d'un redémarrage si vous n'avez pas modifié la configuration.
- Chaque emplacement réel doit être présent une seule fois dans l'enregistrement de commande.
- Vous ne pouvez attribuer un emplacement réel qu'à un emplacement configuré.

Remarque

Modification d'une configuration

L'écriture d'un enregistrement de commande avec une configuration modifiée déclenche la réaction automatique suivante de la CPU : Réinitialisation de la mémoire avec redémarrage ultérieur avec cette configuration modifiée.

Conséquence de cette réaction, la CPU efface l'enregistrement de commande initial et sauvegarde le nouvel enregistrement de commande de manière rémanente.

Comportement en fonctionnement

Pour l'affichage en ligne et pour l'affichage dans la mémoire tampon de diagnostic (module OK ou module défectueux), STEP 7 utilise la configuration de l'appareil et non pas la configuration réelle qui diffère.

Exemple : Un module génère des données de diagnostic. Ce module est configuré à l'emplacement 4, mais est inséré en réalité à l'emplacement 3. La vue en ligne indique que l'emplacement 4 configuré est défectueux. Dans la configuration réelle, le module à l'emplacement 3 indique une erreur grâce à la signalisation par DEL.

Si vous avez configuré des modules comme manquants dans l'enregistrement de commande (0 entrée), le système d'automatisation se comporte de la manière suivante :

- Les modules désignés comme non présents dans l'enregistrement de commande ne fournissent pas de diagnostic et leur état est toujours OK. L'état de la valeur est OK.
- Un accès direct en écriture aux sorties ou un accès en écriture à la mémoire image des sorties qui ne sont pas présentes n'entraîne pas d'effet ; la CPU ne signale pas d'erreur d'accès.
- Un accès direct en lecture aux entrées ou un accès en lecture à la mémoire image des entrées qui ne sont pas présentes entraîne une valeur "0" pour chaque entrée ; la CPU ne signale pas d'erreur d'accès.
- Ecrire un enregistrement dans un module qui n'est pas présent n'entraîne pas d'effet ; la CPU ne signale pas d'erreur.
- Essayer de lire un enregistrement dans un module qui n'est pas présent entraîne une erreur, car la CPU ne peut pas renvoyer d'enregistrement valide.

Messages d'erreur

La CPU renvoie les messages d'erreur suivants si une erreur se produit pendant l'écriture de l'enregistrement de commande :

Code d'erreur	Signification
16#80B1	Longueur invalide ; les informations sur la longueur dans l'enregistrement de commande ne sont pas exactes.
16#80B5	Paramètres de commande de configuration non attribués
16#80E2	L'enregistrement a été transféré dans le mauvais contexte OB. L'enregistrement doit être transféré dans l'OB de démarrage.
16#80B0	Le type de bloc (octet 2) de l'enregistrement de commande n'est pas égal à 196.
16#80B8	Erreur de paramètre ; paramètres invalides des modules d'entrées-sorties, par exemple : <ul style="list-style-type: none"> • L'enregistrement de commande essaie de modifier la configuration d'un module de communication ou une carte annexe de communication. La configuration réelle des modules de communication et une carte annexe de communication doit correspondre à la configuration STEP 7. • La valeur attribuée d'un emplacement non configuré dans le projet STEP 7 n'est pas égale à 255. • La valeur attribuée d'un emplacement configuré se situe en dehors de la plage. • La configuration attribuée possède un emplacement "interne" vide, par exemple, l'emplacement n est attribué et l'emplacement n-1 n'est pas attribué.

6.4.3 Exemple de commande de configuration

Cet exemple décrit une configuration composée d'une CPU et de trois modules d'E/S. Le module à l'emplacement 3 n'est pas présent dans la première installation réelle, donc vous devez utiliser une commande de configuration pour le "masquer".

Dans la deuxième installation, l'application comprend le module qui a été initialement masqué mais l'inclut maintenant dans le dernier emplacement. Un enregistrement de commande modifié fournit les informations concernant les affectations d'emplacements des modules.

Exemple : Installation réelle avec un module configuré mais inutilisé

La configuration de l'appareil contient tous les modules qui peuvent être présents dans une installation réelle (configuration maximum). Dans ce cas, le module qui se trouve dans l'emplacement 3 dans la configuration de l'appareil n'est pas présent dans l'installation réelle.



Figure 6-1 Configuration d'appareil d'installation maximum avec trois modules d'entrées-sorties



Figure 6-2 Installation réelle avec module configuré dans l'emplacement 3 absent, et module configuré pour l'emplacement 4 dans l'emplacement réel 3

Pour indiquer l'absence du module manquant, vous devez configurer l'emplacement 3 dans l'enregistrement de commande avec 0.

ControlDataRecord				
	Nom	Type de données	Valeur de départ	Commentaire
1	▼ Static			
2	▼ ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex card
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	0	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

Exemple : Installation réelle avec module ajouté ultérieurement à un emplacement différent

Dans le deuxième exemple, le module dans l'emplacement 3 de la configuration de l'appareil est présent dans l'installation réelle mais se trouve dans l'emplacement 4.

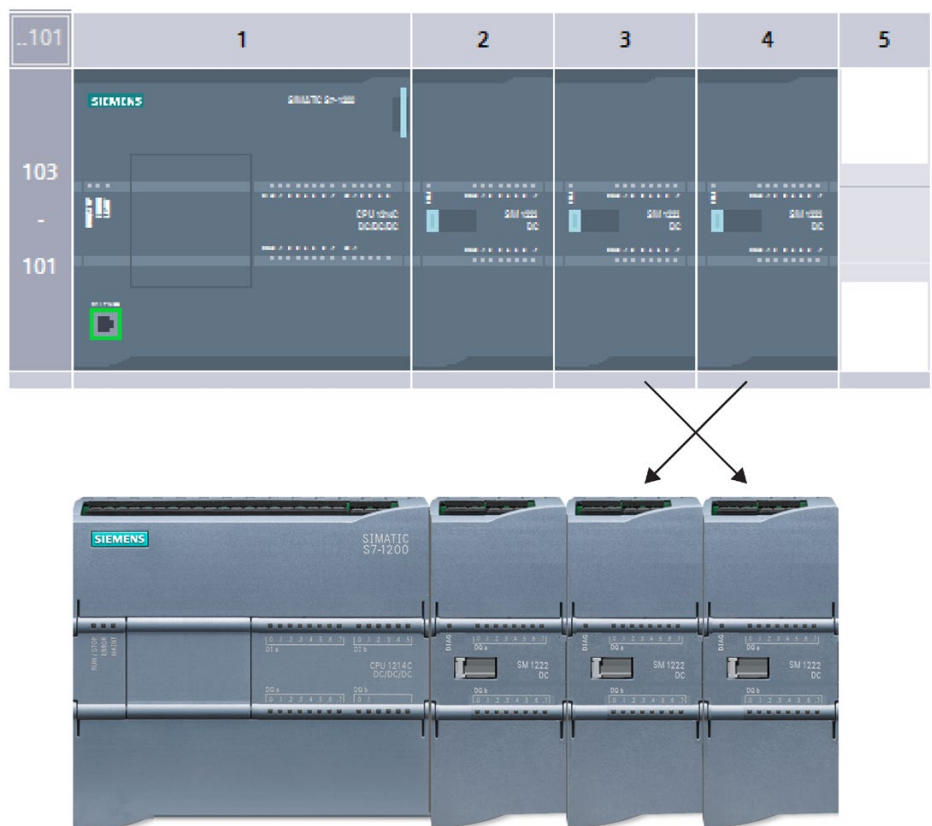


Figure 6-3 Configuration de l'appareil par rapport à l'installation réelle avec les modules dans les emplacements 3 et 4 permutés

Pour mettre en correspondance la configuration de l'appareil avec l'installation réelle, éditez l'enregistrement de commande pour affecter les modules aux positions d'emplacements correctes.

ControlDataRecord				
	Nom	Type de données	Valeur de départ	Commentaire
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex ...
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	4	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

6.5 Changer d'appareil

Vous pouvez changer le type d'appareil d'une CPU ou d'un module configuré. Depuis Configuration de l'appareil, cliquez sur l'appareil avec le bouton droit de la souris et choisissez la commande "Remplacer l'appareil" dans le menu contextuel. Depuis la boîte de dialogue, naviguez jusqu'à la CPU ou au module que vous souhaitez remplacer puis sélectionnez-le. La boîte de dialogue Changer l'appareil vous fournit des informations sur la compatibilité entre les deux appareils.

Des remarques sur le changement d'appareils entre différentes versions de CPU figurent au paragraphe Remplacement d'une CPU V3.0 par une CPU V4.2.x (Page 1665).

6.6 Configuration du fonctionnement de la CPU

6.6.1 Présentation

Pour configurer les paramètres de fonctionnement de la CPU, sélectionnez la CPU dans la vue des appareils (CPU entourée de bleu) et servez-vous de l'onglet "Propriétés" de la fenêtre d'inspection.

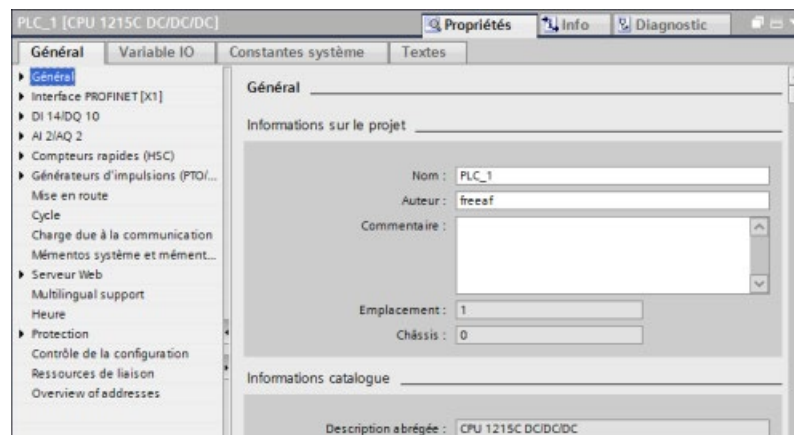


Tableau 6- 2 Propriétés de la CPU

Propriété	Description
Interface PROFINET	Définit l'adresse IP pour la CPU et la synchronisation d'horloge.
DI, DO et AI	Configure le comportement des E/S TOR et analogiques locales (intégrées), par exemple les temps de filtre des entrées TOR et la réaction des sorties TOR à un arrêt de la CPU.
Compteurs rapides (Page 573) et générateurs d'impulsions (Page 504)	<p>Valide et configure les compteurs rapides (HSC) et les générateurs d'impulsions utilisés pour les fonctions PTO (sortie de trains d'impulsions) et PWM (modulation de largeur des impulsions).</p> <p>En cas de configuration des sorties de la CPU ou du Signal Board comme générateurs d'impulsions (pour les instructions PWM ou Motion Control), les adresses des sorties correspondantes sont supprimées de la mémoire Q et ne peuvent pas être utilisées à d'autres fins dans le programme utilisateur. Si votre programme utilisateur écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.</p>
Mise en route (Page 89)	<p>Démarrage après mise sous tension : Sélectionne le comportement de la CPU après une mise sous tension, par exemple démarrage à l'état ARRET ou passage à l'état MARCHE après un démarrage à chaud.</p> <p>Compatibilité matérielle prise en charge : Configure la stratégie de remplacement pour tous les composants système (SM, SB, CM, CP et CPU) :</p> <ul style="list-style-type: none"> • Autoriser un module de remplacement compatible • Autoriser tout module de remplacement (option par défaut) <p>Chaque module contient en interne les conditions de compatibilité de remplacement requises basées sur le nombre d'E/S, la compatibilité électrique et d'autres points de comparaison correspondants. Par exemple, un SM à 16 voies serait un module de remplacement compatible pour un SM à 8 voies, mais un SM à 8 voies ne le serait pas pour un SM à 16 voies Si vous sélectionnez "Autoriser un module de remplacement compatible", STEP 7 applique les règles de remplacement ; sinon, STEP 7 permet tous les remplacements.</p> <p>Temps d'affectation des paramètres pour la périphérie décentralisée : Configure une durée maximale (par défaut : 60000 ms) pour que la périphérie décentralisée soit mise en ligne. (Les CM et les CP reçoivent les paramètres de communication et d'alimentation de la CPU pendant la mise en route. Ce temps d'affectation fournit du temps pour la mise en ligne des E/S connectées au CM ou au CP.)</p> <p>La CPU passe à l'état MARCHE dès que la périphérie décentralisée est en ligne, sans tenir compte du temps d'affectation. Si la périphérie décentralisée n'a pas été mise en ligne à l'expiration de ce temps, la CPU passe quand même à l'état MARCHE, mais sans la périphérie décentralisée.</p> <p>Remarque : Si votre configuration utilise un CM 1243-5 (maître PROFIBUS), ne donnez pas une valeur inférieure à 15 secondes (15 000 ms) à ce paramètre afin de garantir que le module puisse être mis en ligne.</p> <p>Les OB doivent être interruptibles : Configure si l'exécution (de tous les OB) dans la CPU est interruptible ou non interruptible (Page 106).</p>
Cycle (Page 111)	Définit un temps de cycle maximum ou un temps de cycle minimum fixe.
Charge due à la communication	Réserve un pourcentage du temps CPU aux tâches de communication.
Mémento système et mémento de cadence (Page 115)	Valide un octet pour les fonctions "mémento système" et valide un octet pour les fonctions "mémento de cadence" (où chaque bit bascule entre 1 et 0 à une fréquence prédéfinie).
Serveur Web (Page 1037)	Valide et configure la fonction serveur Web.
Heure	Permet de sélectionner le fuseau horaire et de configurer l'heure d'été.

Propriété	Description
Multilinguisme (Page 183)	Définit une langue de projet pour le serveur Web à utiliser pour afficher les textes d'entrée de la mémoire tampon de diagnostic pour chacune des langues d'affichage possibles de l'interface utilisateur du serveur Web.
Protection (Page 218)	Définit la protection en lecture/écriture et les mots de passe pour accéder à la CPU.
Contrôle de la configuration (Page 165)	Permet de définir une configuration d'appareil maître que vous pouvez gérer pour différentes configurations d'appareil réelles.
Ressources de liaison (Page 827)	Fournit un récapitulatif des ressources de liaison de communication disponibles pour la CPU et le nombre de ressources de liaison qui ont été configurées.
Présentation des adresses	Fournit un récapitulatif des adresses d'E/S qui ont été configurées pour la CPU.

6.6.2 Configuration des temps de filtre des entrées TOR

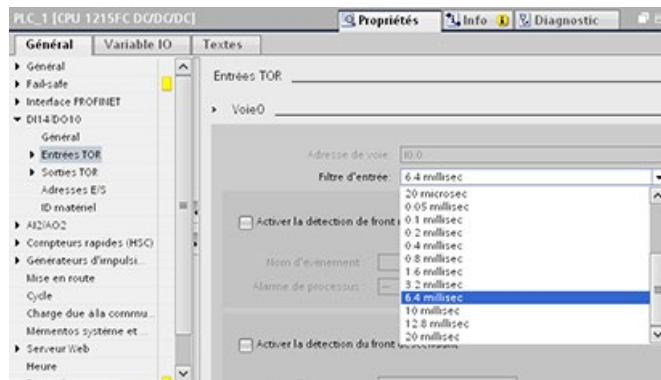
Les filtres d'entrées TOR empêchent votre programme de réagir à des variations rapides indésirables des signaux d'entrée, qui peuvent résulter du rebondissement des contacts de commutation ou du bruit électrique. Le temps de filtre par défaut de 6,4 ms bloque les transitions indésirables des contacts mécaniques typiques. Divers points de votre application peuvent nécessiter des temps de filtre plus courts pour détecter et réagir aux entrées de capteurs rapides ou des temps de filtre plus longs pour bloquer un rebondissement lent des contacts ou un bruit impulsionnel plus long.

Un temps de filtre d'entrée de 6,4 ms signifie qu'un changement de signal unique, de 0 à 1 ou de 1 à 0, doit durer environ 6,4 ms pour être détecté et qu'une impulsion haute ou basse unique durant moins de 6,4 ms approximativement n'est pas détectée. Si un signal d'entrée passe de 0 à 1 plus rapidement que le temps de filtre, la valeur de l'entrée peut changer dans le programme utilisateur lorsque la durée cumulée des impulsions ayant la nouvelle valeur sur les impulsions ayant l'ancienne valeur dépasse le temps de filtre.

Le filtre d'entrée TOR fonctionne comme suit :

- Lorsqu'un "1" est entré, le système incrémente en s'arrêtant au temps de filtre. L'entrée dans la mémoire image passe de "0" à "1" lorsque le décompte atteint le temps de filtre.
- Lorsqu'un "0" est entré, le système décrémente en s'arrêtant à "0". L'entrée dans la mémoire image passe de "1" à "0" lorsque le décompte atteint "0".

- Si l'entrée alterne sans cesse, le compteur incrémente partiellement et décrémente partiellement. La mémoire image changera lorsque le cumul net des décomptes atteindra soit le temps de filtre, soit "0".
- Un signal changeant rapidement avec plus de "0" que de "1" penchera finalement vers "0" alors que, s'il y a plus de "1" que de "0", la mémoire image prendra finalement la valeur "1".



Chaque entrée présente une configuration de filtre unique qui s'applique à toutes les utilisations : entrées du processus, alarmes, capture d'impulsions et entrées HSC. Pour configurer les temps de filtre des entrées, sélectionnez "Entrées TOR".

Le temps de filtre par défaut pour les entrées TOR est de 6,4 ms. Vous pouvez sélectionner un temps de filtre dans la liste déroulante des filtres d'entrée. Plage de temps de filtre valide comprise entre 0,1 us et 20,0 ms.

⚠ ATTENTION

Risques liés à la modification du temps de filtre pour une voie d'entrée TOR

Si vous modifiez le temps de filtre pour une voie d'entrée TOR par rapport à un paramétrage précédent, une nouvelle valeur d'entrée de niveau "0" peut devoir rester à "0" pendant une durée allant jusqu'à 20,0 ms pour que le filtre réagisse pleinement aux nouvelles entrées. Pendant ce temps, les événements d'impulsion "0" courts de moins de 20,0 ms peuvent ne pas être détectés ni comptés.

Cette modification des temps de filtre peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Pour garantir la prise en compte immédiate d'un nouveau temps de filtre, mettez la CPU hors tension puis sous tension.

Configuration du temps de filtre des entrées TOR utilisées comme HSC

Pour les entrées utilisées comme compteurs rapides (HSC), réglez le temps de filtre d'entrée à une valeur appropriée afin d'éviter la perte de valeurs de comptage.

Siemens recommande les paramétrages suivants :

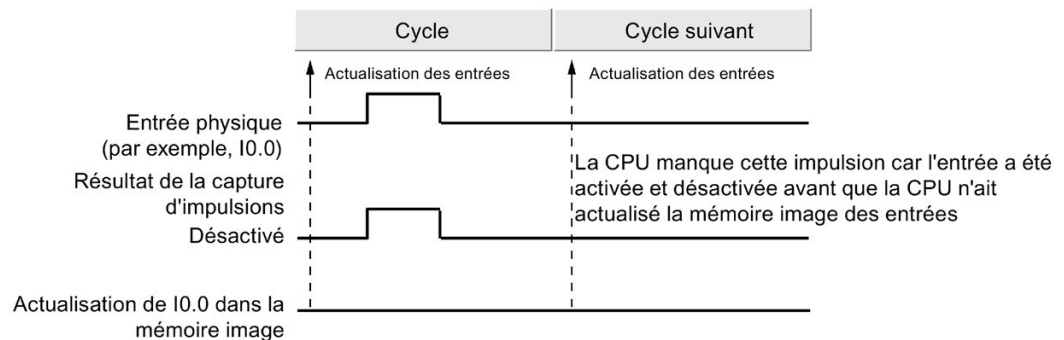
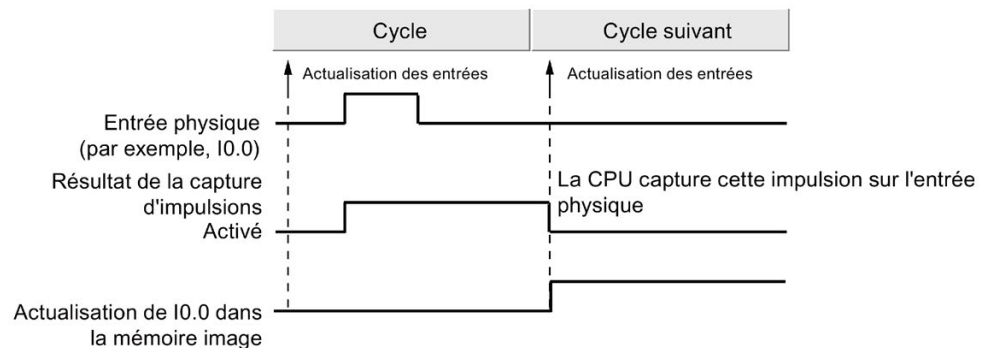
Type de HSC	Temps de filtre d'entrée recommandé
1 MHz	0,1 microseconde
100 kHz	0,8 microseconde
30 kHz	3,2 microsecondes

6.6.3 Capture d'impulsions

La CPU S7-1200 propose une fonction de capture d'impulsions pour les entrées TOR. Cette fonction vous permet de capturer les impulsions montantes ou descendantes d'une durée tellement courte qu'elles pourraient passer inaperçues lorsque la CPU lit les entrées TOR au début du cycle.

Lorsque vous validez la capture d'impulsions pour une entrée, un changement d'état de l'entrée est verrouillé et maintenu jusqu'à l'actualisation cyclique suivante de l'entrée. On a ainsi la garantie qu'une impulsion de brève durée est acquise et maintenue jusqu'à ce que la CPU lise les entrées.

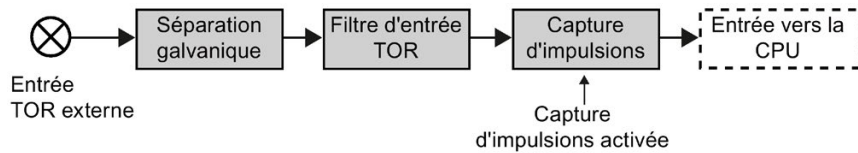
Les figures ci-dessous illustrent le fonctionnement de la CPU S7-1200 lorsque la capture d'impulsions est validée et inhibée :



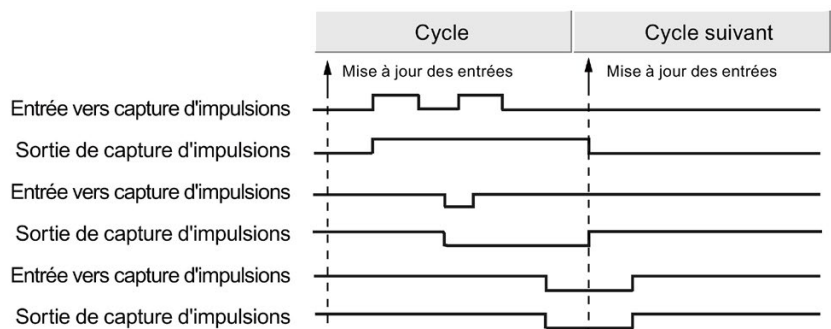
Remarque

Comme la fonction de capture d'impulsions opère sur l'entrée après le passage par le filtre d'entrée, vous devez ajuster le temps de filtre d'entrée de façon que le filtre ne supprime pas l'impulsion.

La figure ci-dessous présente un schéma fonctionnel du circuit d'entrée TOR :



La figure ci-dessous représente la réaction d'une fonction de capture d'impulsions activée dans différentes situations d'entrée. Si un cycle donné comporte plusieurs impulsions, seule la première impulsion est lue. Si un cycle comporte plusieurs impulsions, nous vous conseillons d'utiliser les événements d'interruption sur front montant ou descendant :



6.7 Configuration du multilinguisme

Les paramètres de multilinguisme permettent d'affecter une des deux langues de projet à chaque langue d'interface utilisateur pour le serveur Web (Page 1037) du S7-1200. Vous pouvez également ne pas configurer de langue de projet pour une langue d'interface.

Qu'est-ce qu'une langue de projet ?

La langue de projet est la langue que TIA Portal utilise pour afficher des textes de projet personnalisés comme commentaires de réseau et de bloc.

Vous sélectionnez des langues de projet dans TIA Portal avec la commande **Accessoires > Langues du projet** pour le projet sélectionné dans le navigateur de projet.

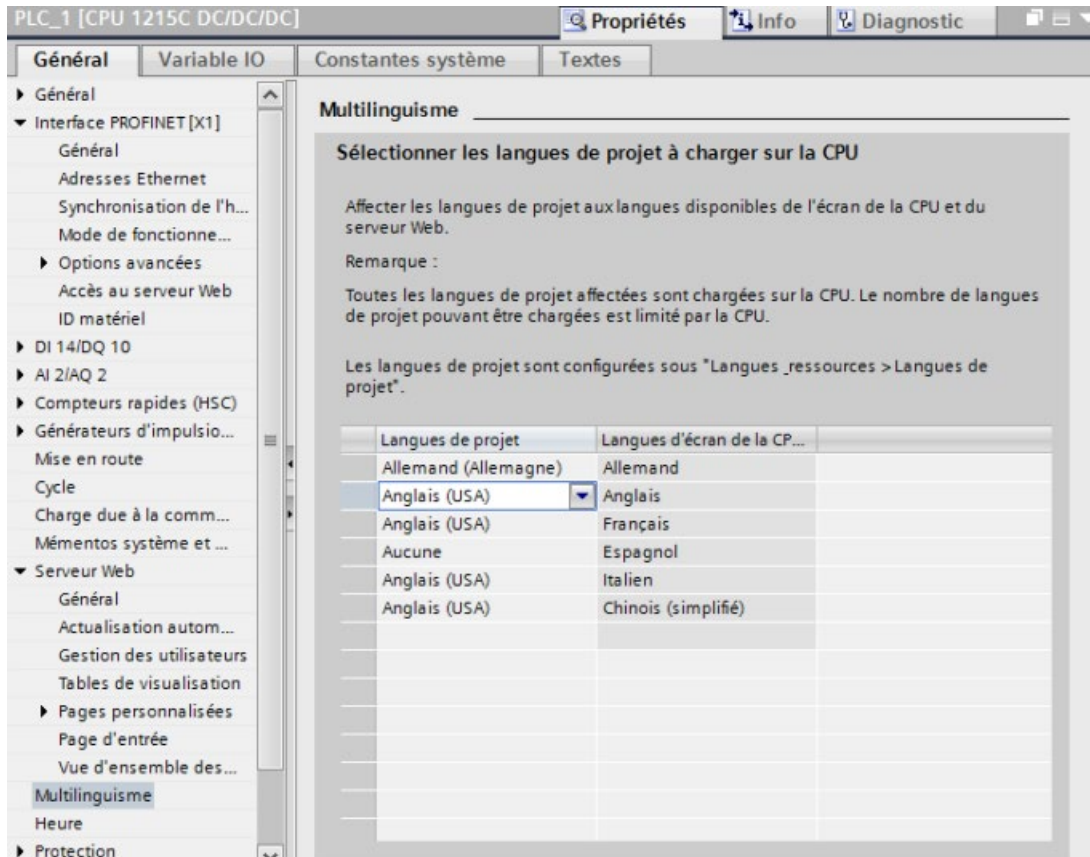
Vous pouvez alors configurer des textes personnalisés tels que des commentaires de réseau et de bloc dans chaque langue de projet avec la commande **Accessoires > Textes du projet**. Lorsque, par la suite, vous changerez la langue de l'interface utilisateur de TIA Portal, les commentaires de réseau, les commentaires de bloc et d'autres textes de projet multilingues s'afficheront dans la langue de projet correspondante. Vous définissez la langue de l'interface utilisateur de TIA Portal avec la commande **Outils > Paramètres > Langue du projet**.

Les langues et textes du projet peuvent également être configurés à partir du nœud **Langues & ressources** du navigateur de projet.

Le serveur Web peut utiliser une ou deux des langues de projet STEP 7 pour l'affichage des messages de la mémoire tampon de diagnostic.

Correspondance entre langue du projet et langue de l'interface utilisateur dans le serveur Web

Le serveur Web prend en charge les mêmes langues d'interface utilisateur que TIA Portal, mais il n'accepte que deux langues de projet au plus. Vous pouvez configurer le serveur Web afin qu'il utilise l'une des deux langues de projet pour les entrées textuelles de la mémoire tampon de diagnostic en fonction de la langue de son interface utilisateur. Vous configurez ces paramètres dans les propriétés de "Multilinguisme" dans la configuration d'appareil de la CPU. (Les commentaires de réseau et de bloc et d'autres textes multilingues ne sont pas visibles dans le serveur Web.)



Dans la page Multilinguisme des propriétés, les langues d'interface à droite ne peuvent pas être modifiées. Ce sont les langues prédéfinies qui sont disponibles pour les interfaces utilisateur de TIA Portal et du serveur Web. L'option "Affecter une langue de projet" est configurable et peut être définie à l'une de vos deux langues de projet configurées ou à "Aucune". Comme la CPU S7-1200 ne prend en charge que deux langues de projet, vous ne pouvez pas configurer une langue de projet identique à la langue d'interface pour toutes les langues d'interface prises en charge.

Dans la configuration ci-dessus, le serveur Web affiche les entrées de la mémoire tampon de diagnostic (Page 1059) en allemand lorsque la langue d'interface utilisateur est l'allemand, n'affiche aucun texte pour les événements de la mémoire tampon de diagnostic lorsque la langue d'interface utilisateur est l'espagnol et affiche les entrées de la mémoire tampon de diagnostic en anglais pour toutes les autres langues de l'interface utilisateur.

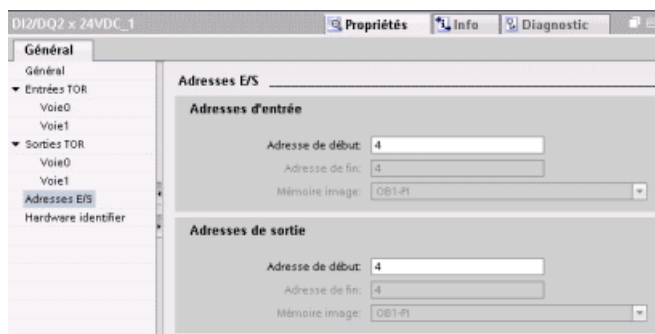
6.8 Configuration des paramètres des modules

Pour configurer les paramètres de fonctionnement d'un module, sélectionnez le module dans la vue des appareils et servez-vous de l'onglet "Propriétés" de la fenêtre d'inspection.

Configuration d'un module d'entrées-sorties (SM) ou d'un Signal Board (SB)

La configuration d'appareil pour les module d'entrées-sorties les Signal Board fournit les moyens de configurer ce qui suit :

- E/S TOR : Vous pouvez configurer des entrées pour la détection de front montant ou de front descendant (en associant chaque entrée à un événement et une alarme de processus) ou pour la capture d'impulsions (afin qu'elles restent activées après une impulsion momentanée) jusqu'à l'actualisation suivante de la mémoire image des entrées. Les sorties peuvent être figées ou utiliser une valeur de remplacement.
- E/S analogiques : Configurez les paramètres des entrées individuelles, par exemple le type de mesure (tension ou courant), la plage et le lissage, et activez le diagnostic de débordement bas ou haut. Les sorties analogiques fournissent des paramètres relatifs au type de sortie (tension ou courant) et au diagnostic, de court-circuit (pour les sorties de tension) ou de limite supérieure/inférieure par exemple. Vous ne configurez pas les plages des entrées et sorties analogiques en unités physiques dans la boîte de dialogue Propriétés, car ce point doit être géré dans la logique de votre programme comme décrit à la rubrique "Traitement des valeurs analogiques (Page 128)".
- Adresses E/S : Vous permet de configurer l'adresse de début pour le jeu d'entrées et de sorties du module. Vous pouvez également affecter les entrées et sorties à une mémoire image partielle (PIP0, PIP1, PIP2, PIP3, PIP4) ou à l'actualisation automatique ou pour n'utiliser aucune mémoire image partielle. Voir "Exécution du programme utilisateur " (Page 85) pour obtenir une explication sur la mémoire image et les mémoires images partielles.



Configuration d'une interface de communication (CM, CP ou CB)

Vous configurez les paramètres pour le réseau en fonction du type d'interface de communication.



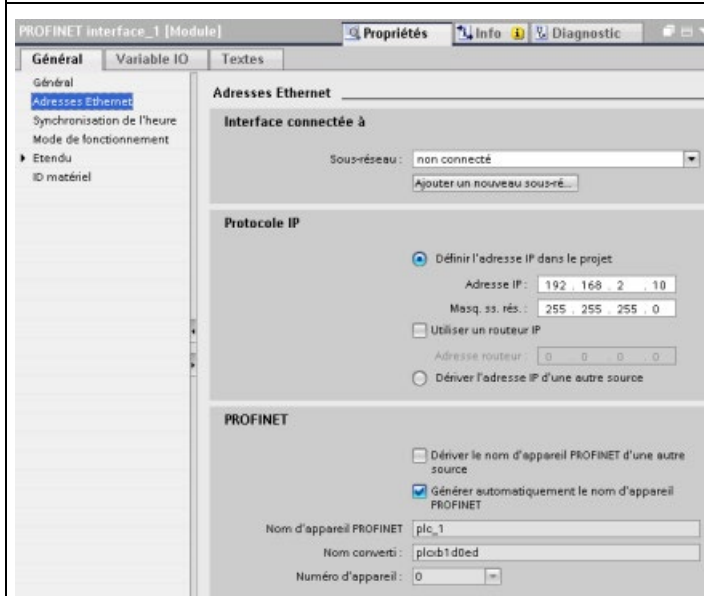
6.9 Configuration de la CPU pour la communication

Le S7-1200 est conçu pour apporter une solution à vos besoins de communication et de mise en réseau par la prise en charge des réseaux les plus simples comme les plus complexes. Le S7-1200 offre également les outils qui vous permettent de communiquer avec les autres appareils, tels que les imprimantes et les balances de pesée qui utilisent leurs propres protocoles de communications.



Utilisez la "Vue du réseau" de la configuration des appareils pour créer les liaisons réseau entre les appareils dans votre projet. Une fois la liaison réseau créée, servez-vous de l'onglet "Propriétés" de la fenêtre d'inspection pour configurer les paramètres du réseau.

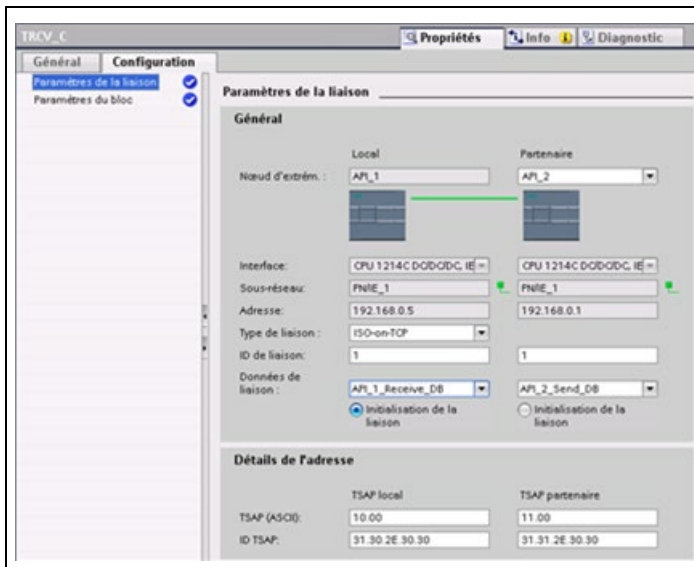
Reportez-vous à "Création d'une liaison réseau" (Page 832) pour plus d'informations.



Dans la fenêtre Propriétés, sélectionnez l'entrée de configuration "Adresses Ethernet". STEP 7 affiche la boîte de dialogue de configuration de l'adresse Ethernet qui associe le projet logiciel à l'adresse IP de la CPU qui recevra ce projet.

Remarque : La CPU S7-1200 ne dispose pas d'une adresse IP préconfigurée. Vous devez affecter manuellement une adresse IP à la CPU.

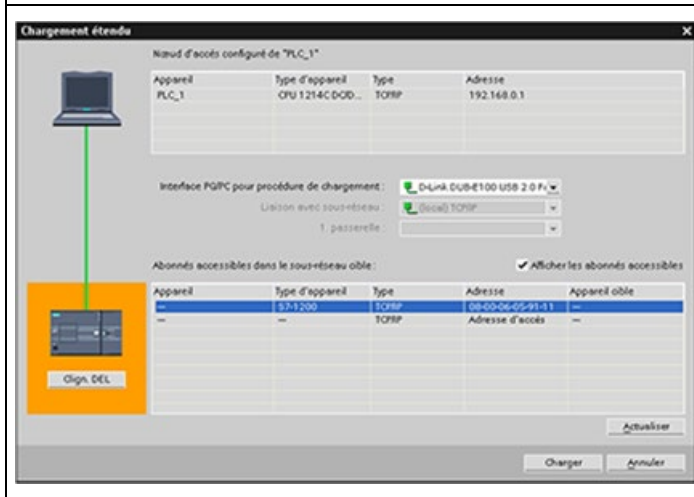
Reportez-vous à "Affectation d'adresses IP (Internet Protocol)" (Page 836) pour plus d'informations.



Pour les protocoles Ethernet TCP, ISO sur TCP et UDP, utilisez les "Propriétés" de l'instruction (TSEND_C, TRCV_C ou TCON) pour configurer les liaisons "locale/partenaire".

La figure montre les "Propriétés de liaison" d'une connexion ISO sur TCP dans l'onglet "Configuration".

Reportez-vous à "Configuration du routage local/partenaire" (Page 833) pour plus d'informations.



Une fois la configuration achevée, chargez le projet dans la CPU. Toutes les adresses IP sont configurées lorsque vous chargez le projet dans la CPU.

Reportez-vous à "Test du réseau PROFINET" (Page 845) pour plus d'informations.

Remarque

Pour effectuer une connexion à votre CPU, votre carte d'interface réseau (NIC) et la CPU doivent se trouver sur le même type de réseau et sur le même sous-réseau. Vous pouvez soit configurer votre carte d'interface réseau pour qu'elle corresponde à l'adresse IP par défaut de la CPU, soit modifier l'adresse IP de la CPU pour qu'elle corresponde au type de réseau et au sous-réseau de votre carte d'interface réseau.

Reportez-vous à "Affectation d'adresses IP (Internet Protocol)" (Page 836) pour plus d'informations sur la manière de procéder.

6.10 Synchronisation de l'heure

Le but de la synchronisation des horloges temps réel est de disposer d'une horloge maître qui synchronise toutes les autres horloges locales. L'horloge maître synchronise les horloges locales une première fois et les resynchronise périodiquement pour éviter les effets d'une dérive au cours du temps.

Dans le cas du S7-1200 et de ses composants de base locaux, seuls la CPU et certains des modules CP comportent des horloges temps réel qui pourraient nécessiter une synchronisation. Vous pouvez configurer l'horloge temps réel de la CPU pour qu'elle soit synchronisée sur une horloge maître externe. L'horloge maître externe peut fournir l'heure grâce à un serveur NTP ou par le biais d'un CP enfiché dans le châssis local du S7-1200 et connecté à un système SCADA comprenant une horloge maître.

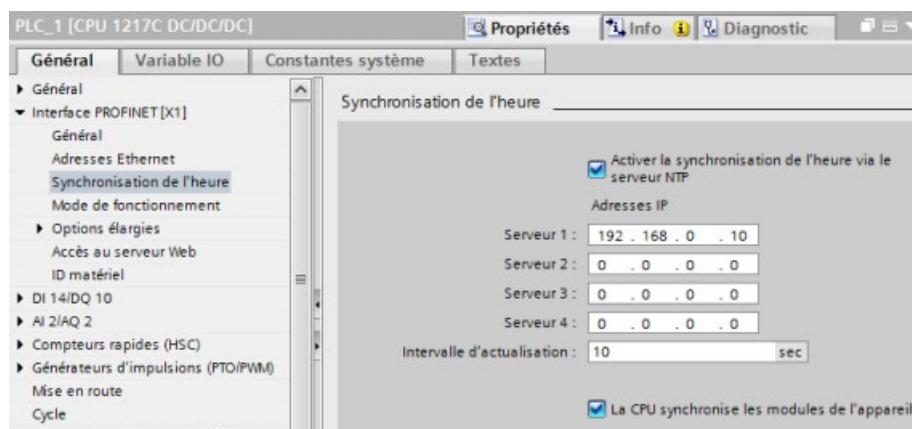
Pour plus d'informations sur les CP S7-1200 qui prennent en charge la fonction de synchronisation de l'heure, voir CP S7-1200 (<https://support.industry.siemens.com/cs/fr/fr/ps>) sur la page Support produit du site Industry Online Support de Siemens.

Réglage de l'horloge temps réel

Il existe trois manières de régler l'horloge temps réel dans la CPU S7-1200 :

- Au moyen du serveur NTP (Page 848)
- Au moyen de STEP 7
- Dans le programme utilisateur
- Au moyen d'un pupitre IHM

Pour configurer la synchronisation de l'heure des modules CP sur l'horloge de la CPU, cochez la case "La CPU synchronise les modules de l'appareil" comme illustré ci-dessous :



Par défaut, ni l'option de synchronisation de l'heure via le serveur NTP, ni l'option de synchronisation des horloges CP avec l'horloge de la CPU ne sont activées.

Vous configurez la synchronisation de l'horloge de la CPU et la synchronisation des horloges CP indépendamment l'une de l'autre. En conséquence, vous pouvez activer la synchronisation des horloges CP par la CPU si l'horloge CPU est réglée par l'une des méthodes indiquées précédemment.

6.10 Synchronisation de l'heure

Vous pouvez sélectionner l'intervalle d'actualisation avec le serveur NTP. Par défaut, l'intervalle d'actualisation du serveur NTP est défini à 10 secondes.

Lorsque vous activez la synchronisation de l'heure dans un module, STEP 7 vous demande de sélectionner l'option "La CPU synchronise les modules de l'appareil" si vous n'avez pas déjà coché la case correspondante dans la boîte de dialogue "Synchronisation de l'heure" de la CPU. STEP 7 vous avertit également si vous avez configuré plus d'une horloge maître source pour la synchronisation de l'heure (vous avez, par exemple, activé la synchronisation de l'heure sur plus d'un CP ou à la fois sur la CPU et sur un module).

Remarque

Lorsque vous activez la synchronisation de l'heure sur un CP, c'est le CP qui règle l'horloge de la CPU.

Si vous activez l'option "La CPU synchronise les modules de l'appareil" dans la boîte de dialogue "Synchronisation de l'heure" de la CPU, la CPU est l'horloge maître. Les modules CP se synchronisent alors sur l'horloge de la CPU.

Remarque

Ne configurez qu'une seule source pour la synchronisation de l'heure de la CPU. La réception de synchronisations horaires pour la CPU en provenance de plus d'une source (serveur NTP ou module CP, par exemple) peut provoquer des conflits lors des mises à jour de l'heure. Des synchronisations horaires provenant de plusieurs sources peuvent avoir une incidence négative sur les instructions et les événements basés sur l'heure.

Concepts de programmation

7.1 Principes de conception d'un système d'automatisation

Lorsque vous concevez un système d'automatisation, vous devez choisir parmi diverses méthodes et divers critères. Les principes généraux suivants sont valables pour de nombreux projets de conception. Il vous faudra, bien sûr, suivre les procédures en vigueur dans votre société ainsi que les usages découlant de votre formation et de votre lieu de travail.

Tableau 7- 1 Principes de conception d'un système d'automatisation

Étapes recommandées	Tâches
Subdivision de votre processus ou de votre installation en unités	Segmentez votre processus ou votre installation en parties indépendantes les unes des autres. Ces subdivisions déterminent les limites entre les automates et ont une influence sur les descriptions fonctionnelles et l'affectation des ressources.
Création des descriptions fonctionnelles	Décrivez le fonctionnement de chaque partie du processus ou de l'installation, par exemple entrées/sorties, description fonctionnelle de l'exploitation, états devant être atteints avant de permettre l'actionnement des actionneurs (tels que valves solénoïdes, moteurs et entraînements), description de l'interface opérateur et toute interface avec d'autres sections du processus ou de l'installation.
Conception des circuits de sécurité	<p>Identifiez tout équipement pouvant nécessiter de la logique câblée pour la sécurité. N'oubliez pas que des défaillances dangereuses peuvent se produire dans les appareils de commande, ce qui peut entraîner un démarrage ou un changement inattendu dans le fonctionnement d'une machine. Lorsqu'il existe alors des risques de blessures sur les personnes ou de dommages matériels importants, songez à implémenter des dispositifs de sécurité électromécaniques prioritaires (qui opèrent indépendamment de l'automate) afin d'éviter des dysfonctionnements dangereux. La conception des circuits de sécurité comprend les tâches suivantes :</p> <ul style="list-style-type: none"> • Identifiez tout fonctionnement incorrect ou inattendu des actionneurs qui pourrait être dangereux. • Identifiez les conditions qui assurent que l'exploitation est sans danger et déterminez comment détecter ces conditions indépendamment de l'automate. • Identifiez comment l'automate affecte le processus lorsque la tension est appliquée puis coupée et identifiez également comment et quand des erreurs sont détectées. Servez-vous de ces informations uniquement pour la conception des fonctionnements normaux et anormaux prévisibles. Vous ne devez pas vous baser sur ce scénario du "meilleur des cas" pour les problèmes de sécurité. • Concevez les dispositifs de sécurité manuels ou électromécaniques prioritaires qui bloquent les fonctionnements dangereux indépendamment de l'automate. • Fournissez à l'automate les informations d'état appropriées en provenance des circuits indépendants afin que le programme et toute interface opérateur disposent des informations nécessaires. • Identifiez toute autre mesure de sécurité nécessaire à un déroulement sûr du processus.
Planification de la sécurité du système	Déterminez le niveau de protection (Page 218) nécessaire pour accéder à votre processus. Vous pouvez protéger les CPU et les blocs de programme par un mot de passe afin d'empêcher les accès non autorisés.

Étapes recommandées	Tâches
Définition des postes d'opération	<p>Créez les schémas suivants des postes d'opération, basés sur les exigences des descriptions fonctionnelles :</p> <ul style="list-style-type: none"> • Schéma d'ensemble indiquant l'emplacement de chaque poste d'opération par rapport au processus ou à l'installation • Schéma mécanique des dispositifs pour le poste d'opération, tels qu'afficheurs, commutateurs et lampes • Schémas électriques avec les E/S associées de l'automate et des modules d'entrées-sorties
Création des schémas de configuration	<p>Créez des schémas de configuration de l'équipement de commande basés sur les exigences des descriptions fonctionnelles :</p> <ul style="list-style-type: none"> • Schéma d'ensemble indiquant l'emplacement de chaque automate par rapport au processus ou à l'installation • Schéma mécanique de chaque automate et de tout module d'E/S incluant les armoires et autres équipements • Schémas électriques pour chaque automate et module d'E/S incluant les numéros de référence des appareils, les adresses de communication et les adresses d'E/S
Création d'une liste de mnémoniques	<p>Créez une liste de mnémoniques avec les adresses absolues. N'indiquez pas seulement les signaux d'E/S physiques, mais également les autres éléments (tels que les noms de variables) qui seront utilisés dans votre programme.</p>

7.2 Organisation de votre programme utilisateur

Lorsque vous créez un programme utilisateur pour les tâches d'automatisation, vous insérez les instructions pour le programme dans des blocs de code :

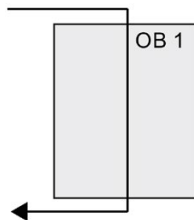
- Un bloc d'organisation (OB) réagit à un événement spécifique dans la CPU et peut interrompre l'exécution du programme utilisateur. Le bloc par défaut pour l'exécution cyclique du programme utilisateur (OB 1) fournit la structure de base pour votre programme utilisateur. Si vous incluez d'autres OB dans votre programme, ces OB interrompent l'exécution de l'OB 1. Les autres OB exécutent des fonctions spécifiques, telles que des tâches de démarrage, la gestion des alarmes et des erreurs ou l'exécution d'un code spécifique à intervalles spécifiques.
- Un bloc fonctionnel (FB) est un sous-programme qui est exécuté lorsqu'il est appelé dans un autre bloc de code (OB, FB ou FC). Le bloc appelant transmet des paramètres au FB et identifie également un bloc de données spécifique (DB) qui contient les données pour l'appel spécifique ou instance de ce FB. Changer le DB d'instance permet à un FB générique de commander le fonctionnement d'un ensemble d'appareils. Ainsi, par exemple, un FB peut piloter plusieurs pompes ou vannes avec des DB d'instance différents contenant les paramètres de fonctionnement spécifiques de chaque pompe ou vanne.
- Une fonction (FC) est un sous-programme qui est exécuté lorsqu'il est appelé dans un autre bloc de code (OB, FB ou FC). Une FC ne comporte pas de DB d'instance associé. Le bloc appelant transmet des paramètres à la FC. Les valeurs de sortie de la FC doivent être écrites dans une adresse de mémoire ou dans un DB global.

Choix du type de structure pour votre programme utilisateur

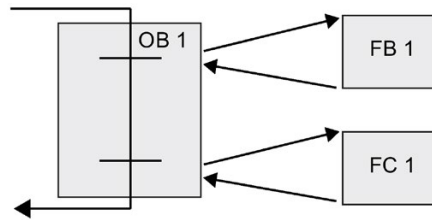
Selon les besoins de votre application, vous pouvez choisir soit une structure linéaire soit une structure modulaire pour votre programme utilisateur :

- Un programme linéaire exécute toutes les instructions pour vos tâches d'automatisation séquentiellement les unes après les autres. Avec un programme linéaire, vous placerez typiquement toutes les instructions dans l'OB d'exécution cyclique du programme (OB 1).
- Un programme modulaire appelle des blocs de code spécifiques qui exécutent des tâches spécifiques. Pour créer une structure modulaire, vous divisez la tâche d'automatisation complexe en petites tâches subordonnées qui correspondent aux fonctions technologiques du processus. Chaque bloc de code fournit le segment de programme pour une tâche subordonnée. Vous structurez votre programme en appelant l'un des blocs de code à partir d'un autre bloc.

Structure linéaire :



Structure modulaire :



En créant des blocs de code génériques pouvant être réutilisés à l'intérieur du programme utilisateur, vous simplifiez la conception et l'implémentation du programme utilisateur.

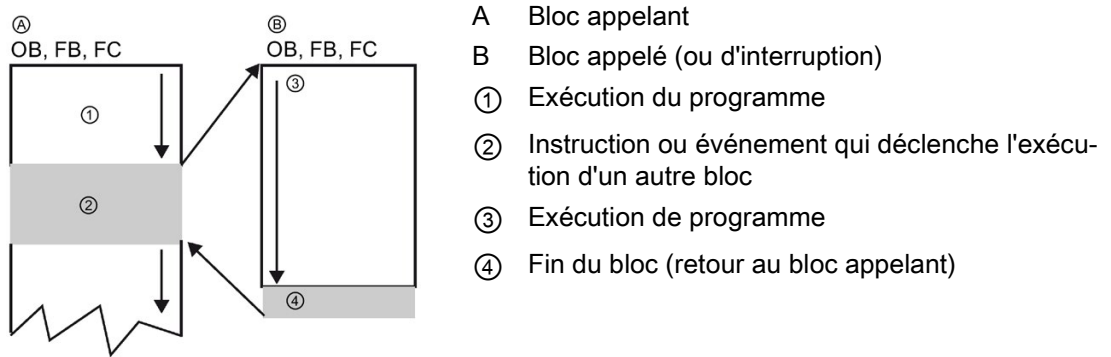
L'utilisation de blocs de code génériques présente un certain nombre d'avantages :

- Vous pouvez créer des blocs de code réutilisables pour des tâches standard, telles que la commande d'une pompe ou d'un moteur. Vous pouvez également stocker ces blocs de code génériques dans une bibliothèque qui peut être utilisée par d'autres applications ou solutions.
- Lorsque vous structurez le programme utilisateur en composants modulaires liés à des tâches fonctionnelles, votre programme devient plus facile à comprendre et à gérer. Non seulement les composants modulaires vous aident à standardiser la conception du programme, mais ils rendent également l'actualisation ou la modification du code plus rapide et plus facile.
- La création de composants modulaires simplifie le débogage de votre programme. En structurant le programme complet en tant qu'ensemble de segments de programme modulaires, vous pouvez tester la fonctionnalité de chaque bloc de code dès qu'il est élaboré.
- La création de composants modulaires liés à des fonctions technologiques spécifiques permet de simplifier et d'accélérer la mise en service de l'application achevée.

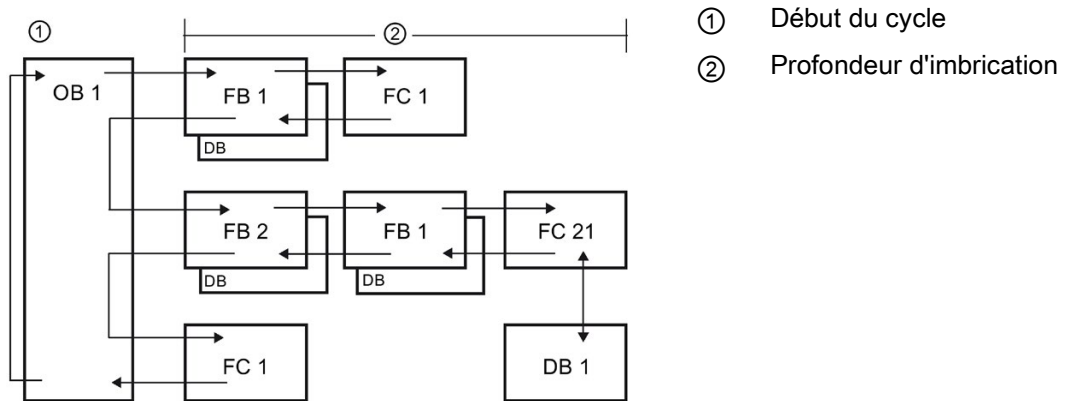
7.3 Utilisation de blocs pour structurer votre programme

En concevant des FB et des FC qui exécutent des tâches génériques, vous créez des blocs de code modulaires. Vous structurez ensuite votre programme en faisant appeler ces modules réutilisables par d'autres blocs de code. Le bloc appelant transmet des paramètres spécifiques de l'appareil au bloc appelé.

Lorsqu'un bloc de code appelle un autre bloc de code, la CPU exécute le code du programme dans le bloc appelé. Une fois l'exécution du bloc appelé achevée, la CPU reprend l'exécution du bloc appelant. Le traitement se poursuit par l'exécution de l'instruction qui suit l'appel de bloc.



Vous pouvez imbriquer les appels de bloc pour obtenir une structure plus modulaire. Dans l'exemple suivant, la profondeur d'imbrication est 3 : l'OB de cycle de programme plus 3 niveaux d'appels de blocs de code.



Remarque : la profondeur d'imbrication maximale est de six. Les programmes de sécurité utilisent deux niveaux d'imbrication. Le programme utilisateur a donc une profondeur d'imbrication de quatre dans les programmes de sécurité.

7.3.1 Bloc d'organisation (OB)

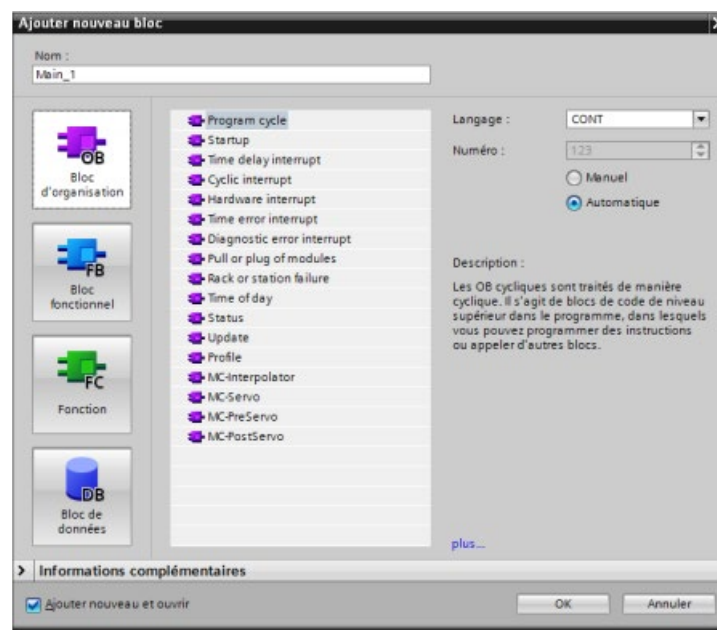
Les blocs d'organisation permettent de structurer votre programme. Ils servent d'interface entre le système d'exploitation et le programme utilisateur. Les OB sont déclenchés sur événement. La CPU exécute un OB à la suite d'un événement, tel qu'une alarme de diagnostic ou un intervalle de temps. Certains OB ont des événements déclencheurs et un comportement prédéfinis.

L'OB du cycle de programme contient votre programme principal. Vous pouvez inclure plus d'un OB de cycle de programme dans votre programme utilisateur. A l'état MARCHE, les OB de cycle de programme s'exécutent au niveau de priorité le plus bas et peuvent être interrompus par tous les autres types d'événements. L'OB de démarrage n'interrompt pas l'OB de cycle de programme, car la CPU exécute l'OB de démarrage avant de passer à l'état MARCHE.

Une fois le traitement des OB de cycle de programme achevé, la CPU réexécute immédiatement les OB de cycle de programme. Ce traitement cyclique est le type "normal" de traitement utilisé par les automates programmables. Pour de nombreuses applications, le programme utilisateur entier se trouve dans un OB de cycle de programme unique.

Vous pouvez créer d'autres OB qui exécuteront des fonctions spécifiques, telles que la gestion des alarmes et des erreurs ou l'exécution d'un code spécifique à intervalles spécifiques. Ces OB interrompent l'exécution des OB du cycle de programme.

Utilisez la boîte de dialogue "Ajouter nouveau bloc" pour créer de nouveaux OB dans votre programme utilisateur.



La gestion des alarmes se fait toujours sur événement. Lorsqu'un tel événement se produit, la CPU interrompt l'exécution du programme utilisateur et appelle l'OB qui a été configuré pour gérer cet événement. Une fois l'exécution de l'OB d'interruption achevée, la CPU reprend l'exécution du programme utilisateur à l'endroit où elle s'était interrompue.

La CPU détermine l'ordre de traitement des événements des alarmes par priorité. Vous pouvez affecter de multiples événements d'alarme à la même classe de priorité. Pour plus d'informations, reportez-vous aux rubriques blocs d'organisation (Page 94) et exécution du programme utilisateur (Page 85).

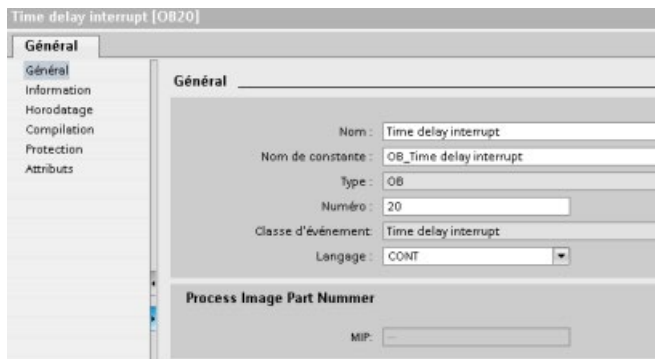
Création d'OB supplémentaires

Vous pouvez créer plusieurs OB pour votre programme utilisateur, et ce même pour les événements OB de cycle de programme et OB de démarrage. Utilisez la boîte de dialogue "Ajouter nouveau bloc" pour créer un OB et entrer le nom de votre OB.

Si vous créez plusieurs OB de cycle de programme pour votre programme utilisateur, la CPU exécute chaque OB de cycle de programme par ordre numérique, en commençant par l'OB de cycle de programme de plus petit numéro (l'OB 1, par exemple). Par exemple, une fois le premier OB de cycle de programme (tel l'OB 1) achevé, la CPU exécute l'OB de cycle de programme de numéro immédiatement supérieur.

Configuration des propriétés d'un OB

Vous pouvez modifier les propriétés d'un OB. Par exemple, vous pouvez configurer le nombre d'OB ou le langage de programmation.



Remarque

Notez que vous pouvez affecter à un OB un numéro de référence de la mémoire image qui correspond à PIP0, PIP1, PIP2, PIP3 ou PIP4. Si vous entrez un nombre pour le numéro de référence de la mémoire image, la CPU crée cette mémoire image partielle. Reportez-vous à la rubrique "Exécution du programme utilisateur (Page 85)" pour une explication des mémoires images partielles.

7.3.2 Fonction (FC)

Une fonction (FC) est un bloc de code qui exécute typiquement une opération spécifique sur un ensemble de valeurs d'entrée. La FC stocke les résultats de cette opération dans des adresses de mémoire. Vous avez p. ex. recours à des FC pour effectuer des opérations standard et réutilisables (telles que des calculs mathématiques) ou des fonctions technologiques (telles que des contrôles individuels utilisant des opérations logiques sur bits) Une FC peut également être appelée plusieurs fois en différents points d'un programme. Cette réutilisation simplifie la programmation de tâches revenant souvent.

Une FC ne comporte pas de bloc de données (DB) d'instance associé. La FC se sert de la pile des données locales pour les données temporaires utilisées pour effectuer l'opération. Les données temporaires ne sont pas sauvegardées. Pour stocker les données de manière permanente, affectez la valeur de sortie à une adresse de mémoire globale, un memento M ou un DB global par exemple.

7.3.3 Bloc fonctionnel (FB)

Un bloc fonctionnel (FB) est un bloc de code qui utilise un bloc de données d'instance pour ses paramètres et ses données statiques. Les FB ont une mémoire de variables qui se situe dans un bloc de données (DB) appelé DB d'instance. Le DB d'instance fournit un bloc de mémoire qui est associé à cette instance (ou appel) du FB et qui contient les données une fois le FB achevé. Vous pouvez associer différents DB d'instance à différents appels du FB. Les DB d'instance vous permettent d'utiliser un FB générique pour commander plusieurs appareils. Vous structurez votre programme en insérant dans un bloc de code l'appel d'un FB et d'un DB d'instance. La CPU exécute alors le code dans ce FB et sauvegarde les paramètres du bloc et les données locales statiques dans le DB d'instance. Une fois l'exécution du FB achevée, la CPU revient au bloc de code qui a appelé le FB. Le DB d'instance conserve les valeurs pour cette instance du FB. Ces valeurs sont disponibles pour des appels ultérieurs du bloc fonctionnel soit dans le même cycle, soit dans d'autres cycles.

Blocs de code réutilisables à mémoire associée

Vous utiliserez typiquement un FB pour commander l'exécution de tâches ou le fonctionnement d'appareils qui durent plus d'un cycle. Pour stocker les paramètres de fonctionnement afin qu'ils soient rapidement accessibles d'un cycle au suivant, chaque FB dans votre programme utilisateur comporte un ou plusieurs DB d'instance. Lorsque vous appelez un FB, vous indiquez aussi un DB d'instance qui contient les paramètres du bloc et les données locales statiques pour cet appel ou "instance" du FB. Le DB d'instance conserve ces valeurs après que l'exécution du FB s'est achevée.

En concevant le FB pour des tâches de commande génériques, vous pouvez le réutiliser pour plusieurs appareils en sélectionnant des DB d'instance différents pour différents appels du FB.

Un FB sauvegarde les paramètres d'entrée, de sortie et d'entrée/sortie et les paramètres statiques dans un DB d'instance.

Vous pouvez également modifier et charger l'interface de bloc fonctionnel dans la CPU à l'état MARCHE (Page 1412).

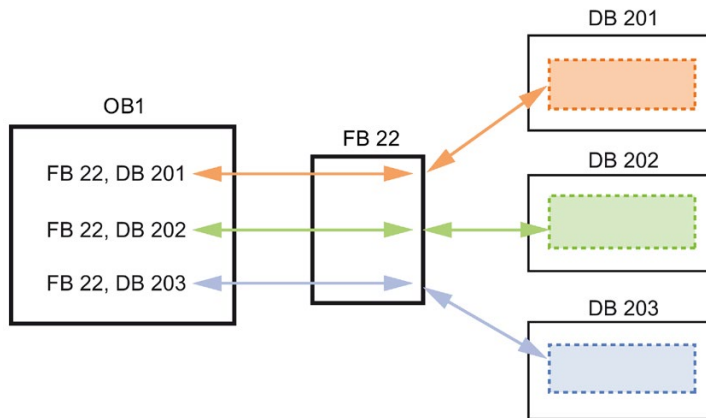
Définition de la valeur initiale dans le DB d'instance

Le DB d'instance comporte une valeur par défaut et une valeur initiale pour chaque paramètre. La valeur initiale fournit la valeur devant être utilisée lors de l'exécution du FB. La valeur initiale peut être modifiée lors de l'exécution de votre programme utilisateur.

L'interface de FB comporte également une colonne "Valeur par défaut" qui vous permet de définir une nouvelle valeur initiale pour le paramètre lors de l'écriture du code de programme. Cette valeur par défaut dans le FB est ensuite transférée dans la valeur initiale du DB d'instance associé. Si vous ne définissez pas de nouvelle valeur initiale pour un paramètre dans l'interface de FB, la valeur par défaut du DB d'instance est copiée dans la valeur initiale.

Utilisation d'un FB unique avec plusieurs DB

La figure suivante montre un OB qui appelle un FB trois fois, avec un bloc de données différent à chaque appel. Cette structure permet à un FB générique de commander plusieurs appareils similaires, tels des moteurs, en affectant un bloc de données d'instance différent à chaque appel pour les différents appareils. Chaque DB d'instance mémorise les données (par exemple, la vitesse, le temps d'accélération et le temps de fonctionnement total) pour un appareil distinct.



Dans cet exemple, le FB 22 commande trois appareils distincts, le DB 201 contenant les données de fonctionnement du premier appareil, le DB 202 celles du deuxième appareil et le DB 203 celles du troisième appareil.

7.3.4 Bloc de données (DB)

Vous créez des blocs de données (DB) dans votre programme utilisateur pour sauvegarder les données des blocs de code. Tous les blocs de programme dans le programme utilisateur peuvent accéder aux données d'un DB global. En revanche, un DB d'instance contient des données pour un bloc fonctionnel spécifique (FB).

Les données sauvegardées dans un DB ne sont pas effacées à la fin de l'exécution du bloc de code associé. Il existe deux types de DB :

- Un DB global contient des données pour les blocs de code dans votre programme. Tous les OB, FB et FC peuvent accéder aux données dans un DB global.
- Un DB d'instance contient les données d'un FB spécifique. La structure des données dans un DB d'instance reflète les paramètres (Input, Output et InOut) et les données statiques du FB. La mémoire Temp pour le FB n'est pas stockée dans le DB d'instance.

Remarque

Bien que le DB d'instance reflète les données d'un FB spécifique, tout bloc de code peut accéder aux données à l'intérieur d'un DB d'instance.

Vous pouvez également modifier et charger des blocs de données à l'état MARCHE (Page 1412).

Blocs de données en lecture seule

Vous pouvez configurer un DB comme étant en lecture seule :

1. Cliquez avec le bouton droit de la souris sur le DB dans le navigateur du projet et sélectionnez "Propriétés" dans le menu contextuel.
2. Sélectionnez "Attributs" dans la boîte de dialogue "Propriétés".
3. Sélectionnez l'option "Bloc de données protégé en écriture dans l'appareil" et cliquez sur "OK".

Blocs de données optimisés et standard

Vous pouvez également configurer un bloc de données de sorte qu'il soit standard ou optimisé. Un DB standard est compatible avec les outils de programmation STEP 7 Classic et les CPU S7-300 et S7-400 classiques. Les blocs de données avec accès optimisé n'ont pas de structure fixe définie. Les éléments de données ne contiennent qu'un nom symbolique dans la déclaration, sans adresse fixe à l'intérieur du bloc. La CPU enregistre les éléments automatiquement dans la zone de mémoire disponible du bloc afin qu'il n'y ait pas d'emplacement vide dans la mémoire. Cela permet une utilisation optimale de la mémoire libre.

Procédez comme suit pour paramétrer l'accès optimisé à un bloc de données :

1. Affichez le détail du dossier Blocs de programme dans l'arborescence du projet STEP 7.
2. Cliquez avec le bouton droit de la souris sur le bloc de données et sélectionnez "Propriétés" dans le menu contextuel.
3. Comme attribut, sélectionnez "Accès au bloc optimisé".

Notez que l'accès au bloc optimisé est l'option par défaut pour les nouveaux blocs de données. Si vous désélectionnez "Accès au bloc optimisé", le bloc utilise l'accès standard.

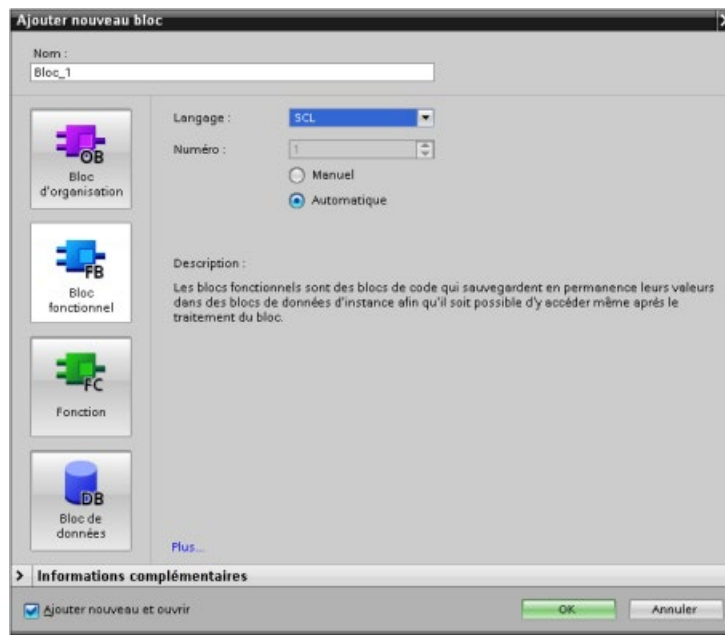
Remarque

Type d'accès au bloc pour un FB et son DB d'instance

Si votre paramétrage du FB est "Accès au bloc optimisé", alors assurez-vous que le paramétrage du DB d'instance pour cet FB soit également "Accès au bloc optimisé". De manière similaire, si vous n'avez pas sélectionné "Accès au bloc optimisé" pour le FB de sorte que le FB soit de type accès standard, alors assurez-vous que le DB d'instance est également standard, ou accès au bloc non optimisé.

Si vous n'avez pas de types d'accès au bloc compatibles, les modifications apportées aux valeurs de paramètres IN/OUT du FB à partir d'une IHM pendant l'exécution du FB peuvent être perdues.

7.3.5 Création de blocs de code réutilisables



Utilisez la boîte de dialogue "Ajouter nouveau bloc" sous "Blocs de programme" dans le navigateur du projet pour créer des OB, des FB, des FC et des DB globaux.

Lors de la création d'un bloc de code, vous sélectionnez le langage de programmation pour ce bloc. Vous ne sélectionnez pas de langage pour les DB qui servent uniquement au stockage de données.

Lorsque vous cochez la case (par défaut) "Ajouter nouveau et ouvrir", le bloc de code s'ouvre dans la vue du projet.

Vous pouvez stocker des objets que vous voulez réutiliser dans des bibliothèques. Pour chaque projet, il y a une bibliothèque de projets qui est connectée au projet. Outre la bibliothèque de projet, vous pouvez créer n'importe quel nombre de bibliothèques internes qui peuvent être utilisées sur plusieurs projets. Les bibliothèques étant compatibles les unes avec les autres, des éléments de bibliothèque peuvent être copiés et retirés d'une bibliothèque sur une autre.

Les bibliothèques sont utilisées, par exemple, pour créer des modèles pour des blocs que vous collez d'abord dans la bibliothèque de projet, puis que vous développez davantage à cet endroit. Enfin, vous copiez les blocs de la bibliothèque du projet sur une bibliothèque globale. Vous rendez la bibliothèque globale disponible pour d'autres collègues qui travaillent sur votre projet. Ils utilisent les blocs et les adaptent en outre aux besoins individuels, si nécessaire.

Pour avoir des précisions sur les mises en œuvre de bibliothèques, reportez-vous aux rubriques de bibliothèques Aide en ligne sur STEP 7.

7.3.6 Transmission de paramètres aux blocs

Les Blocs fonctionnels (FB) et les Fonctions (FC) ont trois types d'interface différents :

- IN
- IN/OUT
- OUT

Les FB et les FC reçoivent des paramètres à travers les types d'interface IN et IN/OUT. Les blocs traitent les paramètres et renvoient les valeurs à l'appelant à travers les types d'interface IN/OUT et OUT.

Le programme utilisateur transfère les paramètres en utilisant l'une des deux méthodes.

Appel par valeur

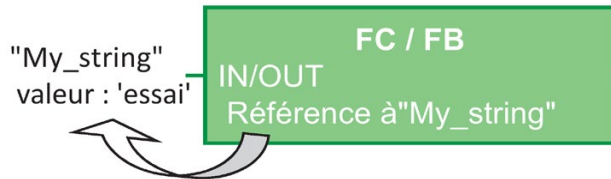
Lorsque le programme utilisateur transmet un paramètre à une fonction comme "appel par valeur", le programme utilisateur copie la valeur de paramètre réelle dans le paramètre d'entrée du bloc pour le type d'interface IN. Cette opération requiert de la mémoire supplémentaire pour la valeur copiée.



Lorsque le programme utilisateur appelle le bloc, il copie les valeurs.

Appel par référence

Lorsque le programme utilisateur transmet un paramètre à une fonction comme "appel par valeur", le programme utilisateur référence l'adresse du paramètre réel pour le type d'interface IN/OUT et ne copie pas la valeur. Cette opération ne requiert pas de mémoire supplémentaire.



Lorsque le programme utilisateur appelle le bloc, il référence l'adresse des paramètres réels.

Remarque

En règle générale, utilisez le type d'interface IN/OUT pour les variables structurées (par exemple, ARRAY, STRUCT et STRING) afin d'éviter d'alourdir la mémoire de données requise inutilement.

Optimisation de bloc et transmission de paramètres

Le programme utilisateur transmet les paramètres FC comme "appel par valeur" pour les types de données simples (par exemple, INT, DINT et REAL). Il transmet des types de données complexes (par exemple, STRUCT, ARRAY, et STRING) comme "appel par référence".

Le programme utilisateur transmet normalement les paramètres du FB dans le bloc de données d'instance (DB) associé au FB :

- Le programme utilisateur transmet les paramètres de données simples (par exemple, INT, DINT et REAL) comme "appel par valeur" en copiant les paramètres vers/à partir du DB d'instance.
- Le programme utilisateur copie les types de données complexes (par exemple, STRUCT, ARRAY, et STRING) vers et à partir du DB d'instance pour les types de paramètres IN et OUT.
- Le programme utilisateur transmet les types de données complexes comme "appel par référence" pour le type d'interface IN/OUT.

Les DB peuvent être créés soit comme "Optimisé" ou "Standard" (non optimisé). Les blocs de données optimisés sont plus compacts que les blocs de données non optimisés. De plus, l'ordre des éléments de données à l'intérieur du DB est différent pour les DB optimisés par rapport aux DB non optimisés. Reportez-vous au paragraphe "Blocs optimisés" de la Directive de programmation S7 pour le S7-1200/1500, STEP 7 (Portail TIA), 03/2014 (<https://support.industry.siemens.com/cs/ww/fr/view/81318674>) pour en savoir plus sur les blocs optimisés.

Vous créez des FB et des FC pour traiter les données, qu'elles soient optimisées ou non. Vous pouvez sélectionner la case "Accès au bloc optimisé" comme l'un des attributs du bloc. Le programme utilisateur optimise les blocs de programme par défaut, et les blocs de programme attendent que les données transmises au bloc soient au format optimisé.

Lorsque le programme utilisateur transmet un paramètre complexe (par exemple, un STRUCT) à une fonction, le système vérifie le paramétrage d'optimisation du bloc de données contenant la structure et le paramétrage d'optimisation du bloc de programme. Si vous optimisez à la fois le bloc de données et la fonction, alors le programme utilisateur transmet le STRUCT comme "appel par référence". Il en est de même si vous sélectionnez non optimisé à la fois pour le bloc de données et la fonction.

Cependant, si vous choisissez une optimisation différente pour la fonction et le bloc de données (ce qui signifie que vous avez optimisé un bloc mais pas l'autre bloc), le STRUCT doit être converti au format attendu par la fonction. Par exemple, si vous sélectionnez non optimisé pour le bloc de données et optimisé pour la fonction, alors un STRUCT dans le bloc de données doit être converti à un format optimisé avant que la fonction ne puisse traiter le STRUCT. Le système effectue cette conversion en créant une "copie" du STRUCT et en la convertissant au format optimisé que la fonction attend.

En résumé, lorsque le programme utilisateur transmet un type de données complexe (par exemple, un STRUCT) à une fonction comme paramètre IN/OUT, la fonction s'attend à ce que le programme utilisateur transmette le STRUCT comme "appel par référence".

- Si vous sélectionnez optimisé ou non optimisé à la fois pour le bloc de données contenant le STRUCT et la fonction, le programme utilisateur transmet les données comme "appel par référence".
- Si vous ne configurez pas le bloc de données et la fonction avec les mêmes paramétrages d'optimisation (l'un est optimisé et l'autre est non optimisé), le système doit faire une copie du STRUCT avant de le transmettre à la fonction. Puisque le système doit faire cette copie de la structure, cela convertit "l'appel par référence" efficacement en un "appel par valeur".

Effet des paramétrages d'optimisation sur les programmes utilisateur

La copie du paramètre peut entraîner un problème dans un programme utilisateur si un IHM ou un OB d'interruption modifie les éléments de la structure. Par exemple, il existe un paramètre IN/OUT d'une fonction (normalement transmise comme "appel par référence"), mais les paramétrages d'optimisation du bloc de données et de la fonction sont différents :

1. Lorsque le programme utilisateur est prêt à appeler la fonction, le système doit faire une "copie" de la structure pour modifier le format des données afin de correspondre à la fonction.
2. Le programme utilisateur appelle la fonction en référence à la "copie" de la structure.
3. Un OB d'interruption se produit alors que la fonction est en cours d'exécution, et l'OB d'interruption modifie une valeur dans la structure d'origine.
4. La fonction se termine et, puisque la structure est un paramètre IN/OUT, le système copie les valeurs à nouveau vers la structure d'origine dans le format d'origine.

L'effet visant à créer une copie de la structure pour modifier le format entraîne la perte des données écrites par l'OB d'interruption. La même chose peut se produire lors de l'écriture d'une valeur avec un IHM. L'IHM peut interrompre le programme utilisateur et écrire une valeur de la même manière qu'un OB d'interruption.

Il y a plusieurs manières de corriger ce problème :

- La meilleure solution à ce problème consiste à faire correspondre les paramètres d'optimisation du bloc de programme et du bloc de données lors de l'utilisation de types de données complexes (par exemple, un STRUCT). Cela permet de garantir que le programme utilisateur transmet toujours les paramètres comme "appel par référence".
- Une autre solution consiste à savoir qu'un OB d'interruption ou un IHM ne modifie pas directement un élément dans la structure. L'OB ou l'IHM peut modifier une autre variable, vous pouvez ensuite copier cette variable dans la structure à un point spécifique dans le programme utilisateur.

7.4 Comprendre le concept de cohérence des données

La CPU assure la cohérence des données pour tous les types de données simples (par exemple, Word et DWord) et pour toutes les structures système (par exemple, IEC_TIMERS et DTL). La lecture ou l'écriture de la valeur ne peuvent pas être interrompues. Ainsi, la CPU protège l'accès à une valeur double mot jusqu'à ce que les quatre octets du double mot aient été lus ou écrits. Pour garantir que les OB de cycle de programme et les OB d'alarme n'écrivent pas en même temps dans la même adresse de mémoire, la CPU n'exécute pas d'OB d'alarme tant que l'opération de lecture ou d'écriture dans l'OB de cycle de programme n'est pas achevée.

Si votre programme utilisateur partage plusieurs valeurs en mémoire entre un OB de cycle de programme et un OB d'alarme, votre programme utilisateur doit également s'assurer que ces valeurs sont modifiées ou lues de manière cohérente. Vous pouvez utiliser les instructions DIS_AIRT (Inhiber le traitement des alarmes) et EN_AIRT (Valider le traitement des alarmes) dans votre OB de cycle de programme pour protéger les accès aux valeurs partagées.

- Insérez une instruction DIS_AIRT dans le bloc de code pour garantir qu'aucun OB d'alarme ne peut être exécuté pendant l'opération de lecture ou d'écriture.
- Insérez les instructions qui lisent ou écrivent les valeurs qui pourraient être modifiées par un OB d'alarme.
- Insérez une instruction EN_AIRT à la fin de la séquence pour annuler DIS_AIRT et permettre l'exécution de l'OB d'alarme.

Une demande de communication émise par un appareil IHM ou une autre CPU peut également interrompre l'OB de cycle de programme. Les demandes de communication peuvent donc elles aussi constituer une menace pour la cohérence des données. La CPU s'assure que les types de données simples sont toujours lus et écrits de manière cohérente par les instructions du programme utilisateur. Comme le programme utilisateur est interrompu périodiquement par des opérations de communication, il n'est pas possible de garantir que plusieurs valeurs dans la CPU seront toutes mises à jour en même temps par l'IHM. Ainsi, les valeurs affichées sur l'écran d'une IHM donnée pourraient provenir de cycles différents de la CPU.

Les instructions point à point, les instructions PROFINET (telles que TSEND_C et TRCV_C), les instructions de périphérie décentralisée PROFINET (Page 394) et les instructions de périphérie décentralisée PROFIBUS (Page 394) transfèrent des tampons de données qui peuvent être interrompus. Garantissez la cohérence des données des tampons en évitant de lire ou d'écrire dans les tampons à la fois dans l'OB de cycle de programme et dans les OB d'alarme. S'il est nécessaire de modifier les valeurs des tampons pour ces instructions dans un OB d'alarme, utilisez une instruction DIS_AIRT pour retarder toute alarme (OB d'alarme ou alarme de communication en provenance d'une IHM ou d'une autre CPU) jusqu'à l'exécution d'une instruction EN_AIRT.

Remarque

L'utilisation de l'instruction DIS_AIRT retarde le traitement des OB d'alarme jusqu'à exécution de l'instruction EN_AIRT, affectant ainsi le temps d'attente d'alarme de votre programme utilisateur (durée s'écoulant entre l'apparition d'un événement et l'exécution de l'OB d'alarme correspondant).

7.5 Langage de programmation

STEP 7 fournit les langages de programmation standard suivants pour le S7-1200 :

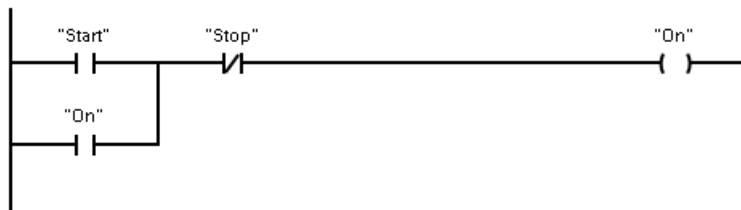
- CONT (schéma à contacts) est un langage de programmation graphique. Sa représentation se base sur des schémas (Page 206) de circuit.
- LOG (logigramme) est un langage de programmation se fondant sur les symboles logiques graphiques utilisés en algèbre (Page 207) booléenne.
- SCL (Structured Control Language) est un langage de programmation littéral évolué (Page 208).

Lorsque vous créez un bloc de code, vous sélectionnez le langage de programmation à utiliser par ce bloc.

Votre programme utilisateur peut utiliser des blocs de code créés dans n'importe lequel des langages de programmation disponibles.

7.5.1 Schéma à contacts (CONT)

Les éléments d'un schéma de circuit, contacts à ouverture et à fermeture et bobines par exemple, sont reliés pour former des réseaux.



Pour créer la logique pour des opérations complexes, vous pouvez insérer des branches formant des circuits parallèles. Les branches parallèles s'ouvrent vers le bas ou se connectent directement à la barre conductrice. Vous terminez les branches vers le haut.

CONT fournit des instructions sous forme de boîtes pour des fonctions variées, telles que les fonctions mathématiques, de temporisation, de comptage et de transfert.

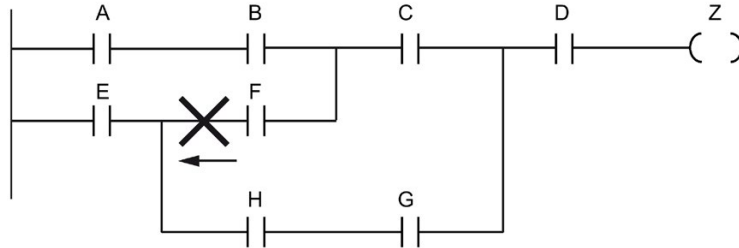
STEP 7 ne limite pas le nombre d'instructions (lignes et colonnes) dans un réseau CONT.

Remarque

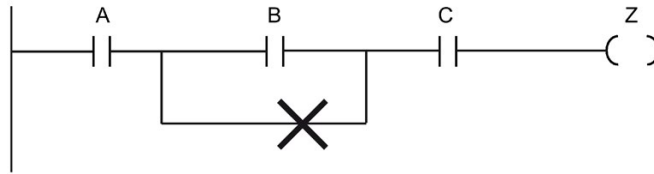
Chaque réseau CONT doit se terminer par une bobine ou une boîte d'instruction.

Vous devez tenir compte des règles suivantes lors de la création d'un réseau CONT :

- Vous ne pouvez pas créer de branche qui entraînerait un flux de courant en sens inverse.

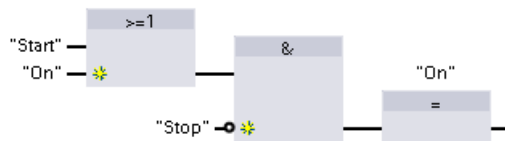


- Vous ne pouvez pas créer de branche qui provoquerait un court-circuit.



7.5.2 Logigramme (LOG)

Comme CONT, LOG est un langage de programmation graphique. La représentation de la logique repose sur les symboles logiques graphiques utilisés en algèbre booléenne.



Pour créer la logique pour des opérations complexes, insérez des branches parallèles entre les boîtes.

Les fonctions mathématiques et autres fonctions complexes peuvent être représentées directement avec des boîtes logiques.

STEP 7 ne limite pas le nombre d'instructions (lignes et colonnes) dans un réseau LOG.

7.5.3 SCL

SCL (Structured Control Language) est un langage de programmation évolué basé sur PASCAL pour les CPU SIMATIC S7. SCL prend en charge la structure de blocs de STEP 7 (Page 194). Votre projet peut contenir des blocs de programme dans l'un quelconque des trois langages de programmation suivants : SCL, CONT et LOG.

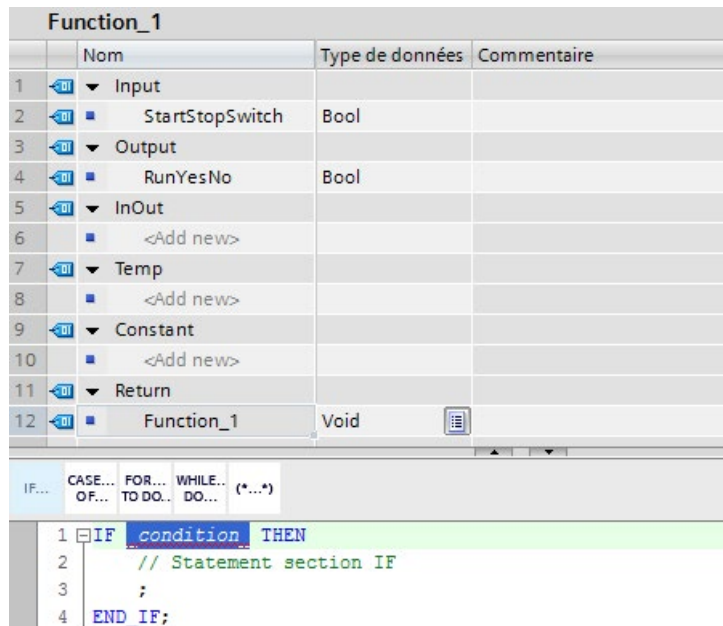
Les instructions SCL utilisent des opérateurs de programmation standard, par exemple pour l'affectation (:=) et les fonctions mathématiques (+ pour l'addition, - pour la soustraction, * pour la multiplication et / pour la division). SCL utilise également des opérations de gestion de programme PASCAL standard, telles que IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO et RETURN. Vous pouvez utiliser n'importe quelle référence PASCAL pour les éléments syntaxiques du langage de programmation SCL. Un grand nombre des autres instructions pour SCL, comme les temporisations et les compteurs, correspondent aux instructions CONT et LOG. Pour plus d'informations sur des instructions spécifiques, reportez-vous aux descriptions correspondantes dans les chapitres sur les instructions de base (Page 237) et les instructions avancées (Page 347).

7.5.3.1 Éditeur de programme SCL

Vous pouvez spécifier, lors de la création de n'importe quel type de bloc (OB, FB ou FC), que ce bloc utilise le langage de programmation SCL. STEP 7 fournit un éditeur de programmes SCL qui inclut les éléments suivants :

- Section d'interface pour définir les paramètres du bloc de code
- Section de code pour le code de programme
- Arborescence d'instructions contenant les instructions SCL prises en charge par la CPU

Vous entrez le code SCL pour votre instruction directement dans la section de code. L'éditeur comprend des boutons pour les structures de code usuelles et les commentaires. Pour des instructions plus complexes, faites simplement glisser les instructions SCL depuis l'arborescence d'instructions et déposez-les dans votre programme. Vous pouvez également utiliser un éditeur de texte quelconque pour créer un programme SCL et importer ensuite ce fichier dans STEP 7.



Vous pouvez déclarer les types de paramètres suivants dans la section d'interface du bloc de code SCL :

- Input, Output, InOut et Ret_Val. Ces paramètres définissent les variables d'entrée, les variables de sortie et la valeur en retour pour le bloc de code. Le nom de variable que vous entrez ici est utilisé localement lors de l'exécution du bloc de code. En règle générale, vous n'utilisez pas le nom de variable globale de la table de variables.
- Static (FB uniquement ; la figure ci-dessus concerne une FC) : Le bloc de code utilise des variables statiques pour stocker des résultats intermédiaires statiques dans le bloc de données d'instance. Le bloc conserve les données statiques jusqu'à leur écrasement, qui peut avoir lieu après plusieurs cycles. Les noms des blocs que ce bloc appelle comme multi-instance sont également stockés dans les données locales statiques.
- Temp : ces paramètres sont les variables temporaires utilisées lors de l'exécution du bloc de code.
- Constant: il s'agit de valeurs de constantes nommés pour votre bloc de code.

Si vous appelez le bloc de code SCL depuis un autre bloc de code, les paramètres du bloc de code SCL apparaissent comme entrées ou sorties.



Dans cet exemple, les variables pour "Start" et "On" (depuis la table de variables du projet) correspondent à "StartStopSwitch" et "RunYesNo" dans la table de déclaration du programme SCL.

7.5.3.2 Expressions et opérations SCL

Construction d'une expression SCL

Une expression SCL est une formule permettant de calculer une valeur. Cette expression est composée d'opérandes et d'opérateurs (tels que *, /, + ou -). Les opérandes peuvent être des variables, des constantes ou des expressions.

L'évaluation de l'expression se fait dans un certain ordre qui est défini par les facteurs suivants :

- Chaque opérateur a une priorité prédéfinie, l'opération de priorité la plus haute étant exécutée en premier.
- Si les opérateurs sont d'égale priorité, ils sont traités de gauche à droite.
- Vous utilisez des parenthèses pour désigner une série d'opérateurs devant être évalués ensemble.

Le résultat d'une expression peut être utilisé pour affecter une valeur à une variable utilisée par votre programme, comme condition devant être utilisée par une instruction de contrôle ou en tant que paramètres pour une autre instruction SCL ou pour appeler un bloc de code.

Tableau 7- 2 Opérateurs dans SCL

Type	Opération	Opérateur	Priorité
Parenthèses	(<i>expression</i>)	(,)	1
Mathématique	Puissance	**	2
	Signe (plus unaire)	+	3
	Signe (moins unaire)	-	3
	Multiplication	*	4
	Division	/	4
	Modulo	MOD	4
	Addition	+	5
	Soustraction	-	5
Comparaison	Inférieur à	<	6
	Inférieur ou égal à	<=	6
	Supérieur à	>	6
	Supérieur ou égal à	>=	6
	Egal à	=	7
	Différent de	<>	7
Instructions logiques sur bits	Négation (unaire)	NOT	3
	ET logique	AND ou &	8
	OU exclusif logique	XOR	9
	OU logique	OR	10
Affectation	Affectation	:=	11

En tant que langage de programmation évolué, SCL utilise des instructions standard pour les tâches de base :

- Instruction d'affectation : :=
- Fonctions mathématiques : +, -, * et /
- Adressage de variables globales : "<nom variable>" (nom de variable ou de bloc de données entre guillemets)
- Adressage de variables locales : #<nom variable> (nom de variable précédé du symbole #)

Les exemples suivants montrent différentes expressions pour différents usages :

<code>"C" := #A+#B;</code>	Affecte la somme de deux variables locales à une variable globale.
<code>"Data_block_1".Tag := #A;</code>	Affectation à une variable de bloc de données
<code>IF #A > #B THEN "C" := #A;</code>	Condition pour l'instruction IF-THEN
<code>"C" := SQRT (SQR (#A) + SQR (#B));</code>	Paramètres pour l'instruction SQRT

Les opérateurs arithmétiques peuvent traiter divers types de données numériques. Le type de données du résultat est déterminé par le type de données des opérandes les plus significatifs. Par exemple, une multiplication qui utilise un opérande INT et un opérande REAL donne une valeur REAL en résultat.

Instructions de contrôle

Une instruction de contrôle est un type spécifique d'expression SCL qui exécute les tâches suivantes :

- Aiguillage dans un programme
- Sections de répétition du code de programme SCL
- Saut à d'autres parties du programme SCL
- Exécution conditionnelle

Parmi les instructions de contrôle SCL, on trouve IF-THEN, CASE-OF, FOR-TO-DO, WHILE-DO, REPEAT-UNTIL, CONTINUE, GOTO et RETURN.

Une seule instruction occupe typiquement une ligne de code. Vous pouvez entrer plusieurs instructions sur une ligne ou vous pouvez subdiviser une instruction en plusieurs lignes de code pour la rendre plus facile à lire. Les séparateurs (tels que tabulations, sauts de ligne et espaces supplémentaires) ne sont pas pris en compte lors du contrôle de syntaxe. Une instruction END met fin à l'instruction de contrôle.

L'exemple suivant montre une instruction de contrôle FOR-TO-DO (les deux formes de codage sont syntaxiquement correctes).

```
FOR x := 0 TO max DO sum := sum + value(x) ; END_FOR;
FOR x := 0 TO max DO
    sum := sum + value(x) ;
END_FOR;
```

Une instruction de contrôle peut également être munie d'un repère. Un repère est délimité par un deux-points au début de l'instruction :

```
repère: <instruction>;
```

L'aide en ligne de STEP 7 fournit des informations de référence complètes sur le langage de programmation SCL.

Conditions

Une condition est une expression de comparaison ou une expression logique dont le résultat est de type BOOL (avec la valeur TRUE ou FALSE). L'exemple suivant montre différents types de conditions :

#Température > 50	Expression de relation
#Compteur <= 100	
#CHAR1 < 'S'	
(#Alpha <> 12) AND NOT #Beta	Expression de comparaison et expression logique
5 + #Alpha	Expression arithmétique

Une condition peut utiliser des expressions arithmétiques :

- La condition de l'expression est TRUE si le résultat est une valeur quelconque différente de zéro.
- La condition de l'expression est FALSE si le résultat est égal à zéro.

Appel d'autres blocs de code depuis votre programme SCL

Pour appeler un autre bloc de code dans votre programme utilisateur, il vous suffit d'entrer le nom (ou l'adresse absolue) du FB ou de la FC concernée avec ses paramètres. Pour un FB, vous devez indiquer le DB d'instance à appeler avec le FB.

<nom DB> (liste de paramètres)	Appel comme instance unique
<#nom instance> (liste de paramètres)	Appel comme multi-instance
"MyDB" (MyInput:=10, MyInOut:="Tag1");	

<nom FC> (liste de paramètres)	Appel standard
<opérande>:=<nom FC> (liste de paramètres)	Appel dans une expression
"MyFC" (MyInput:=10, MyInOut:="Tag1");	

Vous pouvez aussi faire glisser des blocs du navigateur du projet dans l'éditeur de programme SCL, puis achever l'affectation des paramètres.

Ajout de commentaires de bloc au code SCL

Vous pouvez inclure un commentaire de bloc dans votre code SCL en écrivant le texte du commentaire entre (* et *). Vous pouvez insérer autant de lignes de commentaire que vous le désirez entre le (* et le *). Votre bloc de programme SCL peut contenir de nombreux commentaires. Pour faciliter la programmation, l'éditeur SCL propose un bouton de commentaire de bloc avec les instructions de contrôle usuelles :



Adressage

Comme CONT et LOG, SCL permet l'utilisation de variables (adressage symbolique) ou d'adresses absolues dans le programme utilisateur. SCL permet également d'utiliser une variable comme indice de tableau.

Adressage absolu

`%I0.0`
`%MB100`

Faire précéder les adresses absolues du symbole "%". Si "%" manque, STEP 7 génère une erreur variable non définie au moment de la compilation.

Adressage symbolique

`"PLC_Tag_1"`
`"Data_block_1".Tag_1`
`"Data_block_1".MyArray[#i]`

Variable dans la table de variables API
Variable dans un bloc de données
Élément de tableau dans un tableau bloc de données

7.5.3.3 Adressage indexé avec les instructions PEEK et POKE

SCL fournit les instructions PEEK et POKE qui vous permettent de lire et d'écrire dans les blocs de données, les E/S ou la mémoire. Vous indiquez des paramètres pour les décalages d'octet ou de bit spécifiques pour l'opération.

Remarque

Pour utiliser les instructions PEEK et POKE avec des blocs de données, vous devez utiliser des blocs de données standard (non optimisés). Notez également que les instructions PEEK et POKE ne font que transférer des données. Elles n'ont pas connaissance des types de données aux diverses adresses.

```
PEEK(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_);
```

Lit l'octet désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple avec un bloc de données :

```
%MB100 := PEEK(area:=16#84,
               dbNumber:=1, byteOffset:=#i);
```

Exemple avec l'octet d'entrée IB3 :

```
%MB100 := PEEK(area:=16#81,
               dbNumber:=0, byteOffset:=#i); // when
#i = 3
```

```
PEEK_WORD(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_);
```

Lit le mot désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple :

```
%MW200 := PEEK_WORD(area:=16#84,
                    dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_);
```

Lit le double mot désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple :

```
%MD300 := PEEK_DWORD(area:=16#84,
                     dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_);
```

Lit la valeur booléenne désignée par bitOffset et byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée

Exemple :

```
%MB100.0 := PEEK_BOOL(area:=16#84,
                      dbNumber:=1, byteOffset:=#ii,
                      bitOffset:=#j);
```

```
POKE(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_,
      value:=_in_);
```

```
POKE_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_,
           value:=_in_);
```

```
POKE_BLK(area_src:=_in_,
          dbNumber_src:=_in_,
          byteOffset_src:=_in_,
          area_dest:=_in_,
          dbNumber_dest:=_in_,
          byteOffset_dest:=_in_,
          count:=_in_);
```

Ecrit la valeur (octet, mot ou double mot) au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire référencée

Exemple avec un bloc de données :

```
POKE(area:=16#84, dbNumber:=2,
      byteOffset:=3, value:"Tag_1");
```

Exemple avec l'octet de sortie QB3 :

```
POKE(area:=16#82, dbNumber:=0,
      byteOffset:=3, value:"Tag_1");
```

Ecrit la valeur booléenne aux décalages bitOffset et byteOffset désignés du bloc de données ou de la zone d'E/S ou de mémoire référencée

Exemple :

```
POKE_BOOL(area:=16#84, dbNumber:=2,
           byteOffset:=3, bitOffset:=5, value:=0);
```

Ecrit un nombre ("count") d'octets en commençant au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire source référencée au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire de destination référencée

Exemple :

```
POKE_BLK(area_src:=16#84, dbNumber_src:=#src_db, byteOffset_src:=#src_byte, area_dest:=16#84, dbNumber_dest:=#src_db, byteOffset_dest:=#src_byte, count:=10);
```

Pour les instructions PEEK et POKE, les valeurs suivantes sont possibles pour les paramètres "area", "area_src" et "area_dest". Pour les zones autres que les blocs de données, le paramètre dbNumber doit être égal à 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

7.5.4 EN et ENO pour CONT, LOG et SCL

Détermination du "flux de courant" (EN et ENO) pour une instruction

Certaines instructions (telles que les instructions mathématiques et de transfert) comportent des paramètres EN et ENO. Ces paramètres font référence au flux de courant en CONT et LOG et déterminent si l'instruction est ou non exécutée pendant ce cycle. SCL vous permet également d'activer le paramètre ENO pour un bloc de code.

- EN (pour Enable In) est une entrée booléenne. Un flux de courant (EN = 1) doit être présent au niveau de cette entrée pour que la boîte d'instruction soit exécutée. Si l'entrée EN d'une boîte CONT est connectée directement à la barre conductrice gauche, l'instruction sera toujours exécutée.
- ENO (pour Enable Out) est une sortie booléenne. Si un flux de courant est présent à l'entrée EN d'une boîte et que la fonction correspondante s'exécute sans erreur, la sortie ENO transmet le flux de courant (ENO = 1) à l'élément suivant. Si une erreur est détectée lors de l'exécution de la boîte d'instruction, le flux de courant s'arrête (ENO = 0) à la boîte qui a généré l'erreur.

Tableau 7- 3 Opérandes pour EN et ENO

Editeur de programmes	Entrées/sorties	Opérandes	Type de données
CONT	EN, ENO	Flux de courant	Bool
LOG	EN	I, I:P, Q, M, DB, Temp, flux de courant	Bool
	ENO	Flux de courant	Bool
SCL	EN ¹	TRUE, FALSE	Bool
	ENO ²	TRUE, FALSE	Bool

¹ L'utilisation de EN est disponible uniquement pour les FB.

² L'utilisation de ENO avec le bloc de code SCL est facultative. Vous devez configurer le compilateur SCL afin qu'il active ENO à l'achèvement du bloc de code.

Configuration de SCL pour l'activation de ENO

Procédez comme suit pour configurer le compilateur SCL pour l'activation de ENO :

1. Sélectionnez la commande "Paramètres" dans le menu "Outils".
2. Affichez le détail des propriétés "Programmation API" et sélectionnez "SCL (Structured Control Language)".
3. Sélectionnez l'option "Activer ENO automatiquement".

Utilisation de ENO dans le code du programme

Vous pouvez également utiliser ENO dans le code de votre programme, par exemple en affectant ENO à une variable API ou en évaluant ENO dans un bloc local.

Exemples :

```
"MyFunction"  
  ( IN1 := ... ,  
    IN2 := ... ,  
    OUT1 => #myOut,  
    ENO => #statusFlag ); // la variable API statusFlag contient la  
valeur de ENO  
  
"MyFunction"  
  ( IN1 := ...  
    IN2 := ... ,  
    OUT1 => #myOut,  
    ENO => ENO ); // le memento d'état de bloc de "MyFunction"  
                // est stocké dans le bloc local  
  
IF ENO = TRUE THEN  
  // exécuter le code uniquement si MyFunction renvoie ENO avec la  
valeur vrai
```

Effet des paramètres Ret_Val et Status sur ENO

Certaines instructions, telles que les instructions de communication ou les instructions de conversion de chaîne, fournissent un paramètre de sortie qui contient des informations sur le traitement de l'instruction. Par exemple, certaines instructions fournissent un paramètre Ret_Val (valeur en retour) qui est typiquement un type de données Int contenant des informations d'état dans une plage allant de -32768 à +32767. D'autres instructions fournissent un paramètre Status qui est typiquement un type de données Word contenant des informations d'état dans une plage allant des valeurs hexadécimales 16#0000 à 16#FFFF. La valeur numérique contenue dans un paramètre Ret_Val ou Status détermine l'état de ENO pour l'instruction correspondante.

- Ret_Val : Une valeur comprise entre 0 et 32767 met typiquement ENO à 1 (ou VRAI). Une valeur comprise entre -32768 et -1 met typiquement ENO à 0 (ou FAUX). Pour évaluer Ret_Val, passez en représentation hexadécimale.
- Status : Une valeur comprise entre 16#0000 et 16#7FFF met typiquement ENO à 1 (ou VRAI). Une valeur comprise entre 16#8000 et 16#FFFF met typiquement ENO à 0 (ou FAUX).

Les instructions s'exécutant sur plus d'un cycle comportent souvent un paramètre Busy (Bool) qui signale que l'instruction est active mais que l'exécution n'est pas encore achevée. Souvent, ces instructions fournissent aussi un paramètre Done (Bool) et un paramètre Error (Bool). Done signale que l'instruction s'est achevée sans erreur et Error que l'instruction s'est achevée avec une erreur.

- Lorsque Busy vaut 1 (ou VRAI), ENO est égal à 1 (ou VRAI).
- Lorsque Done vaut 1 (ou VRAI), ENO est égal à 1 (ou VRAI).
- Lorsque Error vaut 1 (ou VRAI), ENO est égal à 0 (ou FAUX).

Voir aussi

OK (Contrôler validité) et NOT_OK (Contrôler invalidité) (Page 264)

7.6 Protection

7.6.1 Protection d'accès pour la CPU

La CPU fournit quatre niveaux de sécurité permettant de limiter l'accès à des fonctions spécifiques. Lorsque vous configurez le niveau de sécurité et le mot de passe pour une CPU, vous limitez les fonctions et les zones de mémoire qui sont accessibles sans saisie d'un mot de passe.

Chaque niveau permet d'accéder à certaines fonctions sans mot de passe. Par défaut, il n'y a pas de restriction d'accès ni de protection par mot de passe pour la CPU. Pour limiter l'accès à une CPU, vous configurez les propriétés de la CPU et vous entrez le mot de passe.

L'entrée du mot de passe via un réseau ne met pas en cause la protection par mot de passe de la CPU. La protection par mot de passe ne concerne pas l'exécution des instructions du programme utilisateur, fonctions de communication comprises. La saisie du mot de passe correct permet d'accéder à toutes les fonctions à ce niveau.

Les communications API-API (par le biais d'instructions de communication dans les blocs de code) ne sont pas réduites par le niveau de sécurité dans la CPU.

Tableau 7- 4 Niveaux de sécurité pour la CPU

Niveau de sécurité	Restrictions d'accès
Accès intégral (aucune protection)	Autorise un accès intégral sans protection par mot de passe.
Accès en lecture	Permet l'accès IHM, la comparaison des blocs de code hors ligne/en ligne et toutes les formes de communication API-API sans protection par mot de passe. Un mot de passe est nécessaire pour modifier la CPU (écriture dans la CPU). Aucun mot de passe n'est nécessaire pour changer l'état de fonctionnement de la CPU (MARCHE/ARRÊT).
Accès IHM	Permet l'accès IHM et toutes les formes de communication API-API sans protection par mot de passe. Un mot de passe est nécessaire pour lire les données dans la CPU, pour comparer les blocs de code hors ligne/en ligne, modifier la CPU (écriture) et changer le mode de la CPU (MARCHE/ARRÊT).
Pas d'accès (protection intégrale)	N'autorise aucun accès sans protection par mot de passe. Un mot de passe est requis pour l'accès à une IHM, pour lire les données dans la CPU, comparer les blocs de code hors ligne/en ligne et pour modifier (écrire dans) la CPU.

Notez que vous pouvez définir une adresse IP d'urgence (temporaire) (Page 1036) pour la CPU à tous les niveaux de sécurité.

Il y a distinction entre majuscules et minuscules dans le mot de passe. Procédez comme suit pour configurer le niveau de protection et les mots de passe :

1. Sélectionnez la CPU dans la "Configuration des appareils".
2. Dans la fenêtre d'inspection, sélectionnez l'onglet "Propriétés".
3. Sélectionnez la propriété "Protection" pour choisir le niveau de protection et entrer des mots de passe.

Protection

Protection

Sélectionnez le niveau d'accès pour l'API.

Niveau d'accès	Accès			Permission d'accès	
	IHM	Lire	Ecrire	Mot de passe	Confirmation
<input type="radio"/> Accès complet (pas de protection)	✓	✓	✓	*****	*****
<input type="radio"/> Accès en lecture	✓	✓		*****	*****
<input checked="" type="radio"/> Accès IHM	✓				
<input type="radio"/> Aucun accès (protection complète)					

Quand vous téléchargez cette configuration dans la CPU, l'utilisateur a un accès IHM et peut avoir accès aux fonctions IHM sans mot de passe. Pour lire des données ou comparer les blocs de code hors ligne/en ligne, l'utilisateur doit entrer le mot de passe configuré pour "l'Accès en lecture" ou le mot de passe pour l'"Accès intégral (aucune protection)". Pour écrire des données, l'utilisateur doit entrer le mot de passe configuré pour "l'Accès intégral (aucune protection)".

 **ATTENTION**

Accès non autorisé à une CPU protégée

Les utilisateurs disposant de droits d'accès intégraux à la CPU ont des droits de lecture et d'écriture des variables API. Indépendamment du niveau d'accès pour la CPU, les utilisateurs de serveur Web peuvent avoir des droits de lecture et d'écriture de variables API. L'accès de personnes non autorisées à la CPU ou l'affectation de valeurs invalides aux variables API peut perturber le fonctionnement du processus, ce qui peut entraîner la mort, des blessures graves et/ou des dommages matériels importants.

Les utilisateurs non autorisés peuvent effectuer des modifications d'état de fonctionnement, des écritures sur les données API et des mises à jour de firmware. Siemens vous recommande d'observer les pratiques de sécurité suivantes :

- Niveaux d'accès à la CPU protégés par mot de passe et identifiants utilisateur de serveur Web (Page 1042) avec des mots de passe forts. Les mots de passe forts contiennent au moins dix caractères, lettres, chiffres et caractères spéciaux mélangés, ne sont pas des mots du dictionnaire et ne sont pas des noms ou identifiants pouvant être déduits de vos données personnelles. Ne divulguez pas le mot de passe et changez-en fréquemment.
- N'autorisez l'accès au serveur Web qu'avec le protocole HTTPS.
- Ne pas étendre les droits minimaux par défaut de l'utilisateur "Tout le monde" du serveur Web.
- Effectuez des recherches d'erreur et des vérifications de plage sur vos variables dans la logique de programme, car les utilisateurs des pages Web ont la possibilité de donner des valeurs invalides aux variables API.

Mécanismes de liaison :

Pour avoir accès aux partenaires de connexion à distance avec les instructions PUT/GET, l'utilisateur doit également avoir une permission.

Par défaut, l'option "Permettre l'accès avec la communication PUT/GET" n'est pas activée. Dans ce cas, l'accès en lecture et en écriture aux données de la CPU n'est possible que pour des connexions de communication qui nécessitent une configuration ou une programmation pour la CPU locale et pour le partenaire de communication. L'accès par le biais des instructions BSEND/BRCV est possible, par exemple.

Les connexions pour lesquelles la CPU locale n'est qu'un serveur (ce qui veut dire qu'aucune configuration/programmation de la communication avec le partenaire de communication n'existe au niveau de la CPU locale), sont par conséquent impossibles pendant le fonctionnement de la CPU, par exemple :

- accès PUT/GET, FETCH/WRITE ou FTP par le biais de modules de communication
- accès PUT/GET à partir d'autres CPU S7
- accès IHM par le biais de la communication PUT/GET

Si vous voulez autoriser l'accès aux données de la CPU du côté client, autrement dit, si vous ne voulez pas restreindre les services de communication de la CPU, suivez les étapes suivantes :

1. configurez le niveau d'accès de protection pour qu'il soit n'importe quel niveau sauf "Aucun accès (protection intégrale)".
2. Cochez la case "Permettre l'accès avec la communication PUT/GET".



Quand vous téléchargez cette configuration sur la CPU, la CPU permet la communication PUT/GET à partir de partenaires distants

7.6.2 Mémoire de chargement externe

Il est également possible d'empêcher des copies de la mémoire de chargement interne dans une mémoire de chargement externe (carte mémoire SIMATIC). Procédez comme suit pour empêcher la copie de la mémoire de chargement interne dans une mémoire de chargement externe :

1. Dans la configuration d'appareil de la CPU dans STEP 7, sélectionnez "Protection" dans les propriétés Général.
2. Dans la section "Mémoire de chargement externe", sélectionnez Inhiber la copie de la mémoire de chargement interne dans la mémoire de chargement externe".

Vous trouverez également une description de la manière dont cette propriété affecte l'insertion d'une carte mémoire dans la CPU à la rubrique Insertion d'une carte mémoire dans la CPU (Page 143).

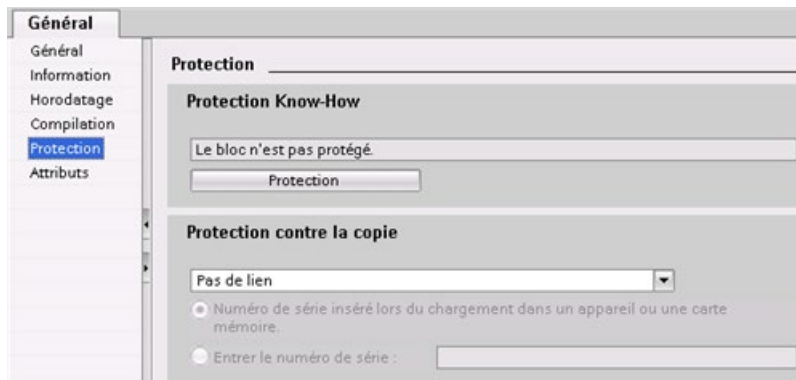
7.6.3 Protection du savoir-faire

Une protection "know-how" vous permet d'empêcher l'accès non autorisé à un ou plusieurs blocs de code (OB, FB, FC ou DB) dans votre programme. Vous créez un mot de passe pour limiter l'accès au bloc de code. La protection par mot de passe empêche de lire ou de modifier le bloc de code sans autorisation. Sans mot de passe, vous pouvez lire uniquement les informations suivantes sur le bloc de code :

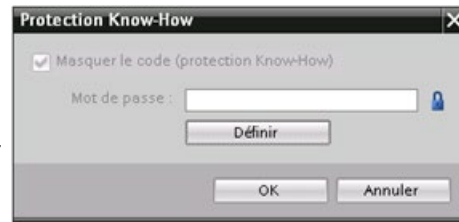
- Titre, commentaire et propriétés du bloc
- Paramètres transmis (IN, OUT, IN_OUT, renvoi)
- Structure d'appel du programme
- Variables globales dans les références croisées (sans information sur l'endroit où elles sont utilisées) ; les variables locales sont masquées

Lorsque vous activez la protection "know-how" pour un bloc, le code à l'intérieur de ce bloc n'est accessible qu'après saisie du mot de passe.

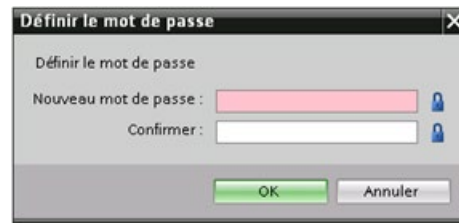
Utilisez la Task Card "Propriétés" du bloc de code pour configurer la protection "know-how" pour ce bloc. Après avoir ouvert le bloc de code, sélectionnez "Protection" dans les propriétés.



1. Dans les propriétés du bloc de code, cliquez sur le bouton "Protection" pour afficher la boîte de dialogue "Protection Know-How".
2. Cliquez sur le bouton "Définir" pour entrer le mot de passe.



Après avoir entré et confirmé le mot de passe, cliquez sur "OK".



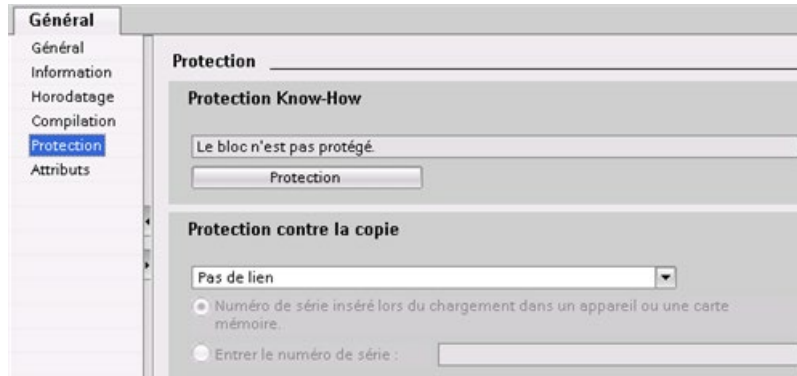
7.6.4 Protection contre la copie

Une fonction de sécurité supplémentaire vous permet de lier des blocs de programme à une utilisation avec une carte mémoire ou une CPU spécifique. Cette fonction est particulièrement utile pour protéger votre propriété intellectuelle. Lorsque vous liez un bloc de programme à un appareil spécifique, vous restreignez le programme ou le bloc de code pour une utilisation uniquement avec une carte mémoire ou une CPU spécifique. Cette fonction vous permet également de distribuer un bloc de code ou de programme électroniquement (par exemple, par Internet ou par courriel) ou en envoyant une carte mémoire. La protection contre la copie est disponible pour des OB (Page 195), des FB (Page 197) et des FC (Page 197). La CPU S7-1200 accepte trois types de protection de bloc :

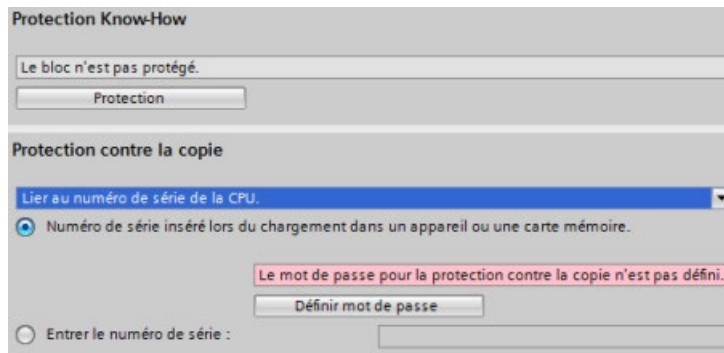
- Liaison avec le numéro de série d'une CPU
- Liaison avec le numéro de série d'une carte mémoire
- Liaison dynamique avec un mot de passe obligatoire

Utilisez la Task Card "Propriétés" du bloc de code pour lier le bloc à une CPU ou une carte mémoire spécifique.

1. Après avoir ouvert le bloc de code, sélectionnez "Protection".



2. Dans la liste déroulante sous la tâche "Protection contre la copie", sélectionnez le type de protection contre la copie que vous souhaitez utiliser.



3. Pour une liaison avec le numéro de série d'une CPU ou d'une carte mémoire, sélectionnez soit d'insérer le numéro de série lors du téléchargement, soit d'entrer le numéro de série de la carte mémoire ou de la CPU.

Remarque

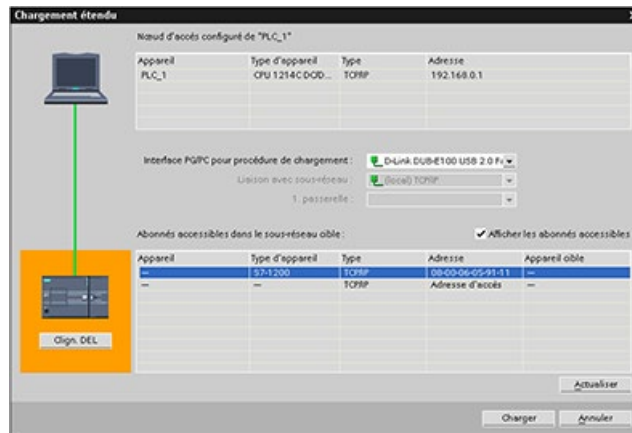
Il y a distinction entre majuscules et minuscules dans le numéro de série.

Pour une liaison dynamique avec un mot de passe obligatoire, définissez le mot de passe que vous devez utiliser pour télécharger ou copier le bloc.

Lorsque vous téléchargez (Page 225) ultérieurement un bloc avec une liaison dynamique, vous devez entrer le mot de passe pour pouvoir télécharger le bloc. A noter que le mot de passe de protection contre la copie et le mot de passe de protection know-how (Page 222) sont deux mots de passe distincts.

7.7 Chargement d'éléments de votre programme dans la CPU

Vous pouvez charger les éléments de votre projet de la console de programmation dans la CPU. Lorsque vous chargez un projet dans la CPU, la CPU stocke le programme utilisateur (OB, FC, FB et DB) dans la mémoire de chargement interne ou, en présence d'une carte mémoire SIMATIC, dans la mémoire de chargement externe (la carte).



Vous pouvez charger votre projet de la console de programmation dans votre CPU à partir de l'un des endroits suivants :

- Navigateur du projet : Cliquez avec le bouton droit de la souris sur l'élément de programme, puis cliquez sur la commande "Charger" du menu contextuel.
- Menu en ligne : Cliquez sur la commande "Charger dans l'appareil".
- Barre d'outils : Cliquez sur l'icône "Charger dans l'appareil".
- Configuration de l'appareil : Cliquez avec le bouton droit de la souris sur la CPU et sélectionnez les éléments à charger.

A noter que si vous avez appliqué une liaison dynamique avec mot de passe obligatoire (Page 223) à n'importe lequel des blocs de programmes, vous devez entrer le mot de passe pour les blocs protégés afin de pouvoir les charger. Si vous avez configuré ce type de protection contre la copie pour des blocs multiples, vous devez entrer le mot de passe pour chacun des blocs protégés afin de pouvoir les charger.

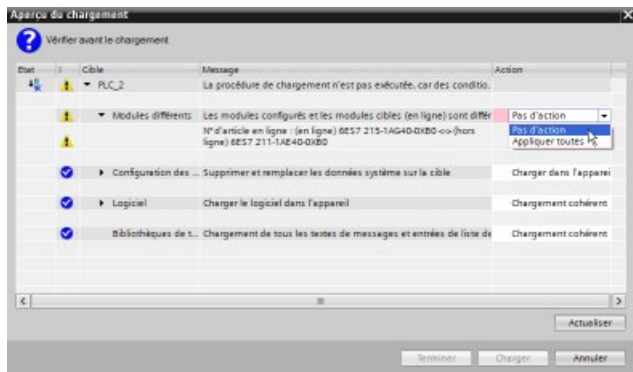
Remarque

Le chargement d'un programme dans la CPU n'efface pas ni ne modifie les valeurs existantes en mémoire rémanente. Pour effacer la mémoire rémanente avant un chargement, réinitialisez la CPU aux réglages d'usine avant d'y charger le programme.

Vous pouvez également charger un projet de pupitre pour les pupitres HMI Basic (Page 33) du portail TIA dans une carte mémoire de la CPU S7-1200.

Chargement dans l'appareil lorsque la CPU configurée est différente de la CPU connectée

STEP 7 et le S7-1200 permettent un chargement dans l'appareil si la CPU connectée est capable de stocker un chargement de la CPU configurée en termes de mémoire requise pour le projet et de compatibilité des entrées/sorties. Ainsi, vous pouvez charger la configuration et le programme d'une CPU dans une CPU plus grande, par exemple d'une CPU 1211C DC/DC/DC dans une CPU 1215C DC/DC/DC parce que les entrées/sorties sont compatibles et que la mémoire est suffisante. Dans ce cas, l'opération de chargement affiche l'avertissement "Les modules configurés et les modules cibles (en ligne) sont différents" ainsi que les numéros d'article et les versions de firmware dans la boîte de dialogue "Aperçu du chargement". Vous devez choisir "Pas d'action" si vous ne voulez pas poursuivre le chargement dans l'appareil ou bien "Appliquer toutes" si vous voulez que le chargement dans l'appareil se poursuive :



Remarque

Lorsque vous passez en ligne (Page 1383) après le chargement de la CPU configurée dans une CPU connectée différente, vous voyez le projet pour la CPU configurée avec des indicateurs d'état en ligne dans le navigateur du projet. Toutefois, dans la vue En ligne et diagnostic, vous voyez bien le type de module de la CPU réellement connectée.

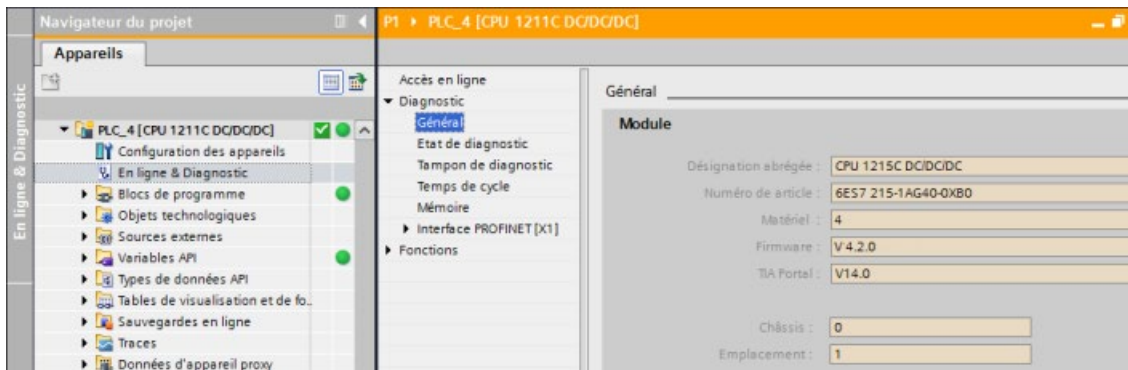


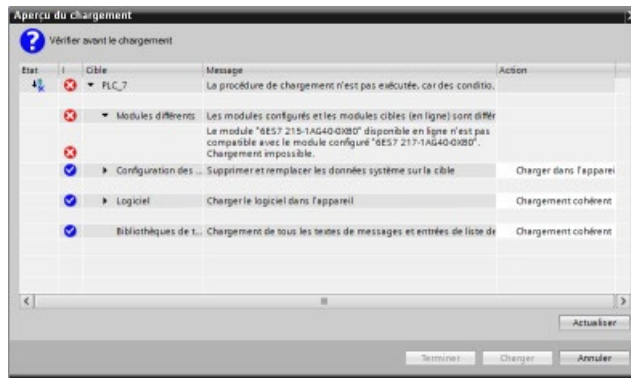
Figure 7-1 Vue en ligne lorsque la CPU configurée est différente de la CPU connectée

Vous pouvez, bien sûr, changer votre appareil (Page 177) dans la configuration d'appareil afin que la CPU configurée soit de même type de module que la CPU connectée. La boîte de dialogue "Remplacer l'appareil" fournit des informations complètes sur la compatibilité lorsque vous tentez de remplacer un appareil.

STEP 7 et le S7-1200 interdisent un chargement dans l'appareil si la CPU connectée n'a pas la capacité de stocker un chargement de la CPU configurée. Vous ne pouvez, par exemple, pas charger la configuration matérielle et le programme dans les cas suivants :

- d'une CPU 1215C DC/DC/DC dans une CPU 1212C DC/DC/DC car la mémoire de travail est insuffisante
- d'une CPU 1211C DC/DC/Relais dans une CPU 1211C DC/DC/DC en raison de la différence des entrées/sorties
- d'une CPU 1217C DC/DC/DC dans n'importe quelle CPU 1211C, CPU 1212C, CPU 1214C ou CPU 1215C en raison de la présence de sorties 1,5 V CC dans la CPU 1217C
- d'une CPU 1214C V4.2.x dans une CPU 1214C V4.0 en raison de l'absence de rétrocompatibilité des versions de firmware

Dans ces cas, la boîte de dialogue "Aperçu du chargement" affiche une erreur :



Récupération en cas d'échec du chargement dans l'appareil

En cas d'échec du chargement, l'onglet "Info" de la fenêtre d'inspection affiche le motif de l'échec. Le tampon de diagnostic fournit également des informations. Après un échec de chargement, procédez comme suit pour réussir le chargement dans l'appareil :

1. Corrigez le problème tel que décrit dans le message d'erreur.
2. Refaites une tentative de chargement.

7.7 Chargement d'éléments de votre programme dans la CPU

Dans de rares cas, le chargement est mené à bien, mais un échec se produit lors d'une mise hors tension puis sous tension ultérieure de la CPU. Dans ce cas, une erreur peut apparaître dans le tampon de diagnostic comme :

- 16# 02:4175 -- Erreur CPU : Erreur d'évaluation de la carte mémoire : Version inconnue ou incompatible du type de carte en cours dans la description de la configuration CPU : Pas de carte mémoire Fonction terminée/annulée, inhibition de nouveau démarrage activée : ...- Carte mémoire manquante, type incorrect, contenu incorrect ou protégé

Si cela se produit et que de nouvelles tentatives de chargement échouent, vous devez effacer la mémoire de chargement interne ou externe :

1. Si vous utilisez la mémoire de chargement interne, réinitialisez la CPU aux paramètres d'usine.
2. Si vous utilisez une carte mémoire SIMATIC, retirez-la et effacez-en le contenu (Page 150) avant de la réinsérer.
3. Chargez la configuration matérielle et le logiciel.

Voir aussi

Synchronisation de la CPU en ligne et du projet hors ligne (Page 229)

7.8 Synchronisation de la CPU en ligne et du projet hors ligne

Lorsque vous chargez des blocs de projet dans la CPU, cette dernière peut détecter si des blocs ou des variables dans la CPU en ligne ont changé depuis le dernier chargement. Dans de tels cas, la CPU vous offre la possibilité de synchroniser les modifications. Cela signifie que vous pouvez charger les modifications de la CPU en ligne dans le projet avant de charger le projet dans la CPU. Les modifications dans la CPU en ligne peuvent être dues à divers facteurs :

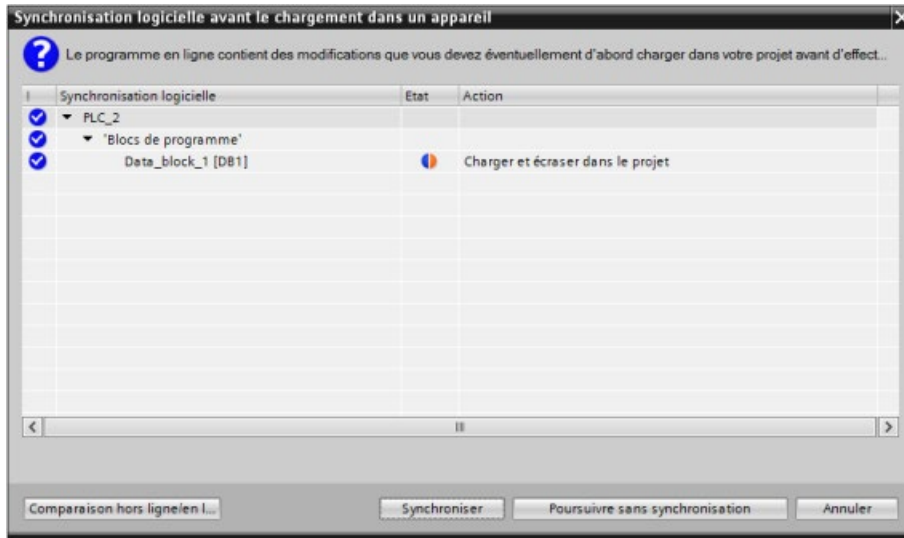
- Modifications apportées aux valeurs initiales des variables de blocs de données pendant l'exécution, par exemple par l'instruction WRIT_DBL (Page 556) ou par le chargement d'une recette
- Chargement dans la CPU depuis un projet "secondaire" (c'est-à-dire un projet autre que celui à l'origine du dernier chargement), une ou plusieurs des conditions suivantes étant également présentes :
 - La CPU en ligne comprend des blocs de programme qui n'existent pas dans le projet.
 - Des variables de blocs de données ou des attributs de blocs diffèrent entre le projet hors ligne et la CPU en ligne.
 - Il existe dans la CPU en ligne des variables API qui n'existent pas dans le projet hors ligne.

Remarque

Si vous éditez des blocs ou des variables dans le projet que vous avez utilisé pour le dernier chargement dans la CPU, vous n'avez aucun choix à faire pour la synchronisation. En effet, STEP 7 et la CPU détectent que les modifications du projet hors ligne sont plus récentes que la CPU en ligne et poursuivent la procédure de chargement normale.

Choix de synchronisation

Lorsque vous chargez un projet dans la CPU, la boîte de dialogue de synchronisation s'affiche si STEP 7 détecte que des blocs de données ou des variables dans la CPU en ligne sont plus récents que les valeurs du projet. Par exemple, si le programme STEP 7 a exécuté WRIT_DBL et modifié la valeur initiale d'une variable dans Data_block_1, STEP 7 affiche la boîte de dialogue de synchronisation suivante lorsque vous exécutez un chargement :



Cette boîte de dialogue énumère les blocs de programme qui présentent des différences. Vous disposez des choix suivants dans cette boîte de dialogue :

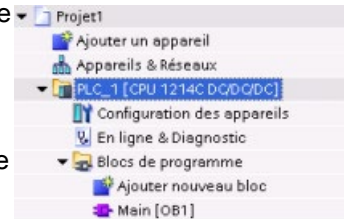
- Comparaison en ligne/hors ligne : Si vous cliquez sur ce bouton, STEP 7 affiche les blocs de programme, les blocs système, les objets technologiques, les variables API et les types de données API du projet comparés à la CPU en ligne (Page 1394). Vous pouvez cliquer sur chaque objet pour afficher une analyse détaillée des différences ainsi que les horodatages. Avec ces informations, vous déciderez quoi faire concernant les différences entre la CPU en ligne et le projet.
- Synchroniser : Si vous cliquez sur ce bouton, STEP 7 charge les blocs de données, les variables et d'autres objets de la CPU en ligne dans le projet. Vous pourrez alors poursuivre le chargement du programme dans la CPU, à moins que l'exécution du programme n'ait à nouveau entraîné une perte de synchronisation entre le projet et la CPU.
- Poursuivre sans synchronisation : Si vous cliquez sur ce bouton, STEP 7 charge le projet dans la CPU.
- Annuler : Cliquez sur ce bouton pour annuler la procédure de chargement dans la CPU.

7.9 Chargement à partir de la CPU en ligne

Vous pouvez également copier les blocs de programme depuis une CPU en ligne ou une carte mémoire reliée à votre console de programmation.

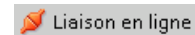
Préparez le projet hors ligne pour les blocs de programme copiés :

1. Ajoutez un appareil CPU qui correspond à la CPU en ligne.
2. Cliquez une fois sur le noeud de la CPU pour rendre le dossier "Blocs de programme" visible.

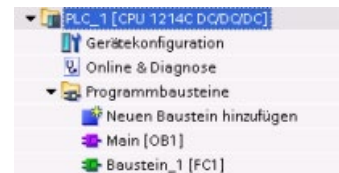


Procédez comme suit pour charger les blocs de programme de la CPU en ligne dans le projet hors ligne :

1. Cliquez sur le dossier "Blocs de programme" dans le projet hors ligne.
2. Cliquez sur le bouton "Liaison en ligne".
3. Cliquez sur le bouton "Charger depuis la CPU".
4. Confirmez votre décision dans la boîte de dialogue Charger depuis la CPU (Page 1383).



Une fois le chargement terminé, STEP 7 affiche tous les blocs de programmes chargés dans le projet.



7.9.1 Comparaison entre la CPU en ligne et la CPU hors ligne

Vous pouvez utiliser l'éditeur de comparaison (Page 1394) dans STEP 7 pour rechercher les différences entre les projets en ligne et hors ligne. Cela peut être utile avant d'effectuer un chargement depuis la CPU.

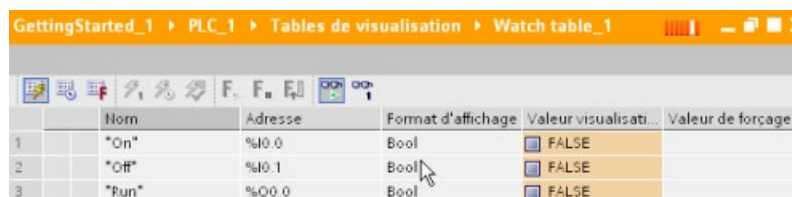
7.10 Débogage et test du programme

7.10.1 Visualisation et forçage de données dans la CPU

Comme indiqué dans le tableau suivant, vous pouvez visualiser et forcer des valeurs dans la CPU en ligne.

Tableau 7- 5 Visualisation et forçage de données avec STEP 7

Editeur	Visualisation	Forçage	Forçage permanent
Table de visualisation	Oui	Oui	Non
Table de forçage permanent	Oui	Non	Oui
Editeur de programmes	Oui	Oui	Non
Table des variables	Oui	Non	Non
Editeur de DB	Oui	Non	Non



Visualisation avec une table de visualisation



Visualisation avec l'éditeur CONT

Reportez-vous au chapitre "Outils en ligne et de diagnostic" pour plus d'informations sur la visualisation et le forçage de données dans la CPU (Page 1396).

7.10.2 Tables de visualisation et tables de forçage permanent

Vous pouvez utiliser des "tables de visualisation" pour visualiser et modifier les valeurs d'un programme utilisateur exécuté par la CPU en ligne. Vous pouvez créer et sauvegarder différentes tables de visualisation dans votre projet afin de prendre en charge des environnements de test variés. Vous pouvez ainsi reproduire des tests pendant la mise en service ou à des fins de dépannage et de maintenance.

Avec une table de visualisation, vous pouvez surveiller la CPU et interagir avec elle alors qu'elle exécute le programme utilisateur. Vous pouvez afficher ou changer les valeurs non seulement pour les variables des blocs de code et de données, mais également pour les zones de mémoire de la CPU, à savoir les entrées et sorties (I et Q), la périphérie d'entrée (I:P), les mémentos (M) et les blocs de données (DB).

Avec la table de visualisation, vous pouvez débloquer les sorties physiques (Q:P) d'une CPU à l'état ARRÊT. Vous pouvez, par exemple, affecter des valeurs spécifiques aux sorties lorsque vous testez le câblage de la CPU.

STEP 7 fournit également une table de forçage permanent pour le forçage permanent d'une variable à une valeur spécifique. Pour plus d'informations sur le forçage, reportez-vous au paragraphe traitant du forçage de valeurs dans la CPU (Page 1404) au chapitre "Outils en ligne et de diagnostic".

Remarque

Les valeurs de forçage permanent sont stockées dans la CPU et non dans la table de visualisation.

Vous ne pouvez pas forcer une entrée (adresse I), mais vous pouvez forcer une entrée de périphérie. Pour forcer une entrée de périphérie, ajoutez ":P" à l'adresse (par exemple, "On:P").

STEP 7 permet également de tracer et d'enregistrer des variables du programme en fonction de conditions de déclenchement (Page 1418).

7.10.3 Affichage de l'usage des références croisées

La fenêtre d'inspection affiche des informations sous forme de références croisées indiquant la manière dont un objet sélectionné est utilisé dans l'ensemble du projet, comme p. ex. le programme utilisateur, la CPU ou tout appareil IHM. La table des "Références croisées" affiche les instances où un objet sélectionné et utilisé, de même que les autres objets qui l'utilisent. La fenêtre d'inspection contient également les blocs qui sont exclusivement accessibles en ligne dans les références croisées. Pour afficher les références croisées, choisissez la commande "Afficher références croisées" (dans la vue du projet, les références croisées font partie du menu "Outils").

Remarque

Il n'est pas nécessaire de fermer l'éditeur pour voir les informations sur les références croisées.

Vous avez la possibilité de trier les entrées des références croisées. La liste des références croisées fournit une vue d'ensemble de l'utilisation des adresses mémoire et des variables dans le programme utilisateur.

- Lorsque vous créez et modifiez un programme, vous obtenez une vue d'ensemble des opérands, variables et appels de blocs utilisés.
- Vous pouvez, à partir des références croisées, sauter directement à l'occurrence d'utilisation des opérands et variables.
- Durant un test du programme ou lors d'un dépannage, vous voyez quelle adresse de mémoire est traitée par quelle commande et dans quel bloc, quelle variable est utilisée dans quelle vue et quel bloc est appelé par quel autre bloc.

Tableau 7- 6 Éléments de la référence croisée

Colonne	Description
Objet	Nom de l'objet utilisant les objets de niveau inférieur ou qui est utilisé par les objets de niveau inférieur
Nombre	Nombre d'utilisations
Endroit d'utilisation	Toute adresse possible, p. ex. adresse de réseau.
Propriétés	Propriétés spécifiques des objets référencés, p. ex. noms de variables dans des déclarations multiinstances
En tant que	Affiche des informations supplémentaires sur l'objet, p. ex. si un DB d'instance est utilisé en tant que modèle ou instance multiple.
Accès	Type d'accès à l'opérande, à savoir en lecture (R) et/ou en écriture (W).
Adresse	Adresse de l'opérande
Type	Information sur le type et le langage utilisés pour créer l'objet.
Chemin	Chemin de l'objet dans l'arborescence

Selon les produits installés, le tableau des références croisées affiche des colonnes supplémentaires ou différentes.

7.10.4 Structure d'appel permettant de constater la hiérarchie d'appel

La structure d'appel décrit la hiérarchie d'appel des blocs dans votre programme utilisateur. Elle fournit une vue d'ensemble des blocs utilisés, des appels d'autres blocs, des relations entre blocs, des données requises pour chaque bloc et de l'état des blocs. Vous pouvez ouvrir l'éditeur de programmes et éditer les blocs de la structure d'appel.

L'affichage de la structure d'appel fournit une liste des blocs utilisés dans le programme utilisateur. STEP 7 met en évidence le premier niveau de la structure d'appel et affiche tout bloc n'étant pas appelé par un autre bloc dans le programme. Le premier niveau de la structure d'appel affiche les OB ainsi que tous les FC, FB et DB qui ne sont pas appelés par un OB. Si un bloc de code appelle un autre bloc, le bloc appelé est représenté en retrait sous le bloc appelant. La structure d'appel affiche uniquement les blocs qui sont appelés par un blocs de code.

Vous avez la possibilité d'afficher exclusivement les blocs entraînant des conflits dans la structure d'appel. Les conditions suivantes entraînent des conflits :

- Blocs exécutant des appels avec des horodatages plus anciens ou plus récents
- Blocs appelant un bloc avec une interface modifiée
- Blocs utilisant une variable avec une adresse et/ou un type de données modifiés
- Blocs n'étant pas appelés directement ou indirectement par un OB.
- Blocs appelant un bloc inexistant ou manquant

Vous pouvez regrouper plusieurs appels de bloc et blocs de données. Une liste déroulante indique les liens aux différentes adresses d'appel.

Vous pouvez également effectuer un contrôle de cohérence pour afficher les conflits d'horodatage. Un changement d'horodatage d'un bloc durant ou après la compilation du programme risque d'entraîner des conflits d'horodatage, qui à leur tour provoquent des incohérences parmi les blocs appelants et les blocs appelés.

- Une recompilation des blocs de code permet de corriger la plupart des conflits d'horodatage et d'interface.
- Si la compilation ne permet pas de remédier aux incohérences, servez-vous du lien dans la colonne "Détails" pour aller à la source du problème dans l'éditeur de programmes. Vous pouvez alors corriger manuellement toutes les incohérences.
- Tous les blocs marqués en rouge doivent être recompilés.

Instructions de base

8.1 Opérations logiques sur bits

8.1.1 Opérations combinatoires sur bits

CONT et LOG sont très efficaces dans le traitement de la logique booléenne. Bien que SCL soit particulièrement efficace pour les calculs mathématiques complexes et pour les structures de commande de projet, vous pouvez aussi l'utiliser pour la logique booléenne.

Contacts CONT

Tableau 8- 1 Contacts à fermeture et contacts à ouverture



CONT	SCL	Description
"IN" 	<pre>IF in THEN Statement; ELSE Statement; END IF;</pre>	Contacts à fermeture et contacts à ouverture : Vous pouvez connecter des contacts à d'autres contacts et créer votre propre logique combinatoire. Si le bit d'entrée que vous indiquez utilise l'identificateur de mémoire I (entrée) ou Q (sortie), la valeur du bit est lue dans la mémoire image du processus. Les signaux physiques dans votre processus de commande sont câblés aux bornes I sur l'automate. La CPU lit les signaux d'entrée câblés et actualise en continu les valeurs d'état correspondantes dans la mémoire image des entrées.
"IN" 	<pre>IF NOT (in) THEN Statement; ELSE Statement; END_IF;</pre>	Vous pouvez lancer la lecture directe d'une entrée physique en indiquant ":P" après le décalage I (par exemple, "%I3.4:P"). En cas de lecture directe, les valeurs de données binaires sont lues directement dans l'entrée physique et non dans la mémoire image. Une lecture directe n'actualise pas la mémoire image.

Tableau 8- 2 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Bool	Bit affecté

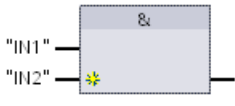
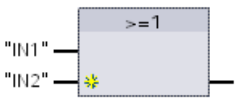
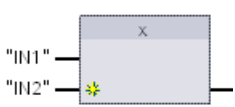
- Le contact à fermeture est fermé (activé) lorsque la valeur du bit affecté est égale à 1.
- Le contact à ouverture est fermé (activé) lorsque la valeur du bit affecté est égale à 0.
- Des contacts connectés en série créent des réseaux de logique ET.
- Des contacts connectés en parallèle créent des réseaux de logique OU.

Boîtes ET, OU et OU EXCLUSIF en LOG

En programmation LOG, les réseaux de contacts CONT sont transformés en réseaux de boîtes ET (&), OU (>=1) et OU EXCLUSIF (x) dans lesquels vous pouvez indiquer des valeurs binaires pour les entrées et sorties des boîtes. Vous pouvez aussi établir des connexions à d'autres boîtes logiques et créer vos propres combinaisons logiques. Une fois que vous avez placé la boîte dans votre réseau, vous pouvez faire glisser l'outil "Insérer entrée" de la barre d'outils "Favoris" ou de l'arborescence d'instructions vers le côté des entrées de la boîte afin d'ajouter des entrées. Vous pouvez aussi cliquer avec le bouton droit de la souris sur le connecteur d'entrée de la boîte et sélectionner "Insérer entrée".

Les entrées et les sorties de la boîte peuvent être connectées à une autre boîte logique ou vous pouvez entrer une adresse binaire ou un mnémonique de bit pour une entrée non connectée. Lors de l'exécution de la boîte d'instruction, les états des entrées en cours sont appliqués à la logique binaire et, si elle est vraie, la sortie de la boîte sera vraie.

Tableau 8-3 Boîtes ET, OU et OU EXCLUSIF

LOG	SCL ¹	Description
	<pre>out := in1 AND in2;</pre>	Toutes les entrées d'une boîte ET doivent être vraies pour que la sortie soit vraie.
	<pre>out := in1 OR in2;</pre>	Il suffit qu'une entrée d'une boîte OU soit vraie pour que la sortie soit vraie.
	<pre>out := in1 XOR in2;</pre>	Un nombre impair d'entrées d'une boîte OU EXCLUSIF doivent être vraies pour que la sortie soit vraie.

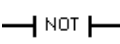
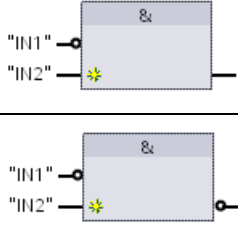
¹ Pour SCL : Vous devez affecter le résultat de l'opération à une variable destinée à être utilisée par une autre instruction.

Tableau 8-4 Types de données pour les paramètres

Paramètre	Type de données	Description
IN1, IN2	Bool	Bit d'entrée

Inverseur logique NOT

Tableau 8- 5 Inverser RLO (Résultat logique)

CONT	LOG	SCL	Description
		NOT	<p>En programmation LOG, vous pouvez faire glisser l'outil "Inverser l'entrée binaire" de la barre d'outils "Favoris" ou de l'arborescence d'instructions vers une entrée ou une sortie afin de créer un inverseur logique sur ce connecteur de boîte.</p> <p>Le contact NOT en CONT inverse l'état logique de l'entrée de flux de courant.</p> <ul style="list-style-type: none"> • S'il n'y a pas de flux de courant entrant dans le contact NOT, il y a un flux de courant sortant. • S'il y a un flux de courant entrant dans le contact NOT, il n'y a pas de flux de courant sortant.

Bobine et boîte d'affectation de sortie

L'instruction Bobine de sortie écrit une valeur pour un bit de sortie. Si le bit de sortie que vous indiquez utilise l'identificateur de mémoire Q, la CPU met le bit de sortie dans la mémoire image du processus à 1 ou à 0 conformément à l'état de flux de courant. Les signaux de sortie pour vos actionneurs de commande sont câblés aux bornes Q de la CPU. A l'état MARCHE, la CPU lit en continu vos signaux d'entrée, traite les états des entrées en fonction de la logique de votre programme, puis réagit en donnant aux sorties de nouvelles valeurs dans la mémoire image des sorties. La CPU transfère le nouvel état des sorties mémorisé dans la mémoire image aux bornes de sortie câblées.

Tableau 8- 6 Affectation et inverser affectation

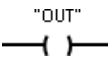
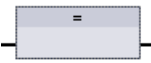
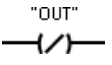

CONT	LOG	SCL	Description
		<code>out := <expression booléenne>;</code>	<p>En programmation LOG, les bobines CONT sont transformées en boîte d'affectation (= et /=) dans lesquelles vous indiquez une adresse de bit pour la sortie de la boîte. Les entrées et sorties de la boîte peuvent être connectées à une autre boîte logique ou vous pouvez entrer une adresse de bit.</p> <p>Vous pouvez demander l'écriture directe dans une sortie physique en indiquant ":P" après le décalage Q (par exemple, "%Q3.4:P"). En cas d'écriture directe, les valeurs de données binaires sont écrites directement dans la mémoire image des sorties et directement dans la sortie physique.</p>
		<code>out := NOT <expression booléenne>;</code>	

Tableau 8- 7 Types de données pour les paramètres

Paramètre	Type de données	Description
OUT	Bool	Bit affecté

8.1 Opérations logiques sur bits

- S'il y a flux de courant à travers une bobine de sortie ou qu'une boîte d'affectation "=" LOG est activée, le bit de sortie est mis à 1.
- S'il n'y a pas de flux de courant à travers une bobine de sortie ou qu'une boîte d'affectation "=" LOG n'est pas activée, le bit de sortie est mis à 0.
- S'il y a flux de courant à travers une bobine de sortie inversée ou qu'une boîte "/=" LOG est activée, le bit de sortie est mis à 0.
- S'il n'y a pas de flux de courant à travers une bobine de sortie inversée ou qu'une boîte "/=" LOG n'est pas activée, le bit de sortie est mis à 1.

8.1.2 Instructions Mise à 1 et Mise à 0

Mise à 1 et Mise à 0 d'1 bit

Tableau 8- 8 Instructions S et R

CONT	LOG	SCL	Description
"OUT" —{S}—		Non disponible	Activer sortie : Lorsque S (Mise à 1) est activé, la valeur de données à l'adresse OUT est mise à 1. Lorsque S n'est pas activé, OUT n'est pas modifié.
"OUT" —{R}—		Non disponible	Désactiver sortie : Lorsque R (Mise à 0) est activé, la valeur de données à l'adresse OUT est mise à 0. Lorsque R n'est pas activé, OUT n'est pas modifié.

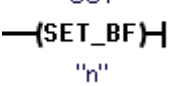

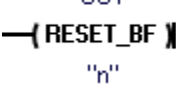

- 1 Pour CONT et LOG : Ces instructions peuvent être placées n'importe où dans le réseau.
- 2 Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

Tableau 8- 9 Types de données pour les paramètres

Paramètre	Type de données	Description
IN (ou connexion à contact/logique de porte)	Bool	Variable binaire de l'adresse à visualiser
OUT	Bool	Variable binaire de l'adresse à mettre à 1 ou à 0

Mise à 1 et Mise à 0 champ de bits

Tableau 8- 10 Instructions SET_BF et RESET_BF :

CONT ¹	LOG	SCL	Description
<p>"OUT"</p>  <p>"n"</p>	<p>"OUT"</p> 	Non disponible	Mise à 1 champ de bits : Lorsque SET_BF est activé, la valeur 1 est affectée à "n" bits en commençant à l'adresse OUT. Lorsque SET_BF n'est pas activé, OUT n'est pas modifié.
<p>"OUT"</p>  <p>"n"</p>	<p>"OUT"</p> 	Non disponible	Mettre à 0 champ de bits : RESET_BF écrit la valeur 0 dans "n" bits en commençant à l'adresse OUT. Lorsque RESET_BF n'est pas activé, OUT n'est pas modifié.

1 Pour CONT et LOG : Ces instructions doivent être à l'extrême droite dans une branche.



2 Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

Tableau 8- 11 Types de données pour les paramètres

Paramètre	Type de données	Description
OUT	Bool	Elément de départ d'un champ de bits à mettre à 1 ou à 0 (exemple : #MyArray[3])
n	Constante (UInt)	Nombre de bits à écrire

Bascules avec mise à 1 prioritaire et mise à 0 prioritaire

Tableau 8- 12 Instructions RS et SR :

CONT/LOG	SCL	Description
<p>"INOUT"</p> 	Non disponible	Basculer avec mise à 0/mise à 1 : L'opération RS est une bascule avec mise à 1 prioritaire où la mise à 1 domine. Si les signaux de mise à 1 (S1) et de mise à 0 (R) sont tous deux vrais, la valeur à l'adresse INOUT sera 1.
<p>"INOUT"</p> 	Non disponible	Basculer avec mise à 1/mise à 0 : L'opération SR est une bascule avec mise à 0 prioritaire où la mise à 0 domine. Si les signaux de mise à 1 (S) et de mise à 0 (R1) sont tous deux vrais, la valeur à l'adresse INOUT sera 0.

1 Pour CONT et LOG : Ces instructions doivent être à l'extrême droite dans une branche.

2 Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

Tableau 8- 13 Types de données pour les paramètres

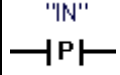
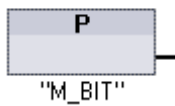
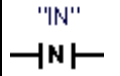
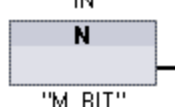
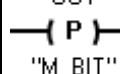
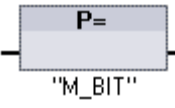
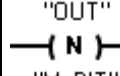
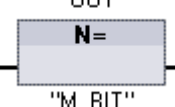
Paramètre	Type de données	Description
S, S1	Bool	Entrée de mise à 1 ; 1 indique la prédominance.
R, R1	Bool	Entrée de mise à 0 ; 1 indique la prédominance.
INOUT	Bool	Variable binaire affectée "INOUT"
Q	Bool	Suit l'état du bit "INOUT".

La variable "INOUT" définit l'adresse de bit qui est mise à 1 ou à 0. La sortie Q facultative prend l'état de signal de l'adresse "INOUT".

Instruction	S1	R	Bit "INOUT"
RS	0	0	Etat précédent
	0	1	0
	1	0	1
	1	1	1
SR	S	R1	
	0	0	Etat précédent
	0	1	0
	1	0	1
	1	1	0

8.1.3 Instructions Front montant et Front descendant



Tableau 8- 14 Détection de front montant et de front descendant

CONT	LOG	SCL	Description
 <p>"IN" "M_BIT"</p>	 <p>"IN" "M_BIT"</p>	Non disponible ¹	<p>Surveiller opérande de front montant.</p> <p>CONT : L'état de ce contact est VRAI lorsqu'un front montant (0 à 1) est détecté sur le bit "IN" affecté. L'état logique du contact est alors combiné à l'état du flux de courant entrant pour définir l'état du flux de courant sortant. Le contact P peut être situé n'importe où dans le réseau, excepté à la fin d'une branche.</p> <p>LOG : L'état logique de la sortie est VRAI lorsqu'un front montant (0 à 1) est détecté sur le bit d'entrée affecté. La boîte P ne peut être située qu'au début d'une branche.</p>
 <p>"IN" "M_BIT"</p>	 <p>"IN" "M_BIT"</p>	Non disponible ¹	<p>Surveiller opérande de front descendant.</p> <p>CONT : L'état de ce contact est VRAI lorsqu'un front descendant (1 à 0) est détecté sur le bit d'entrée affecté. L'état logique du contact est alors combiné à l'état du flux de courant entrant pour définir l'état du flux de courant sortant. Le contact N peut être situé n'importe où dans le réseau, excepté à la fin d'une branche.</p> <p>LOG : L'état logique de la sortie est VRAI lorsqu'un front descendant (1 à 0) est détecté sur le bit d'entrée affecté. La boîte N ne peut être située qu'au début d'une branche.</p>
 <p>"OUT" "M_BIT"</p>	 <p>"OUT" "M_BIT"</p>	Non disponible ¹	<p>Définir un opérande sur le front montant.</p> <p>CONT : Le bit affecté "OUT" est VRAI lorsqu'un front montant (0 à 1) est détecté sur le flux de courant entrant dans la bobine. L'état du flux de courant entrant traverse toujours la bobine en tant qu'état du flux de courant sortant. La bobine P peut être située n'importe où dans le réseau.</p> <p>LOG : Le bit affecté "OUT" est VRAI lorsqu'un front montant (0 à 1) est détecté sur l'état logique au niveau de la connexion d'entrée de la boîte ou sur l'affectation de bit d'entrée si la boîte est située au début d'une branche. L'état logique d'entrée traverse toujours la boîte en tant qu'état logique de sortie. La boîte P= peut être située n'importe où dans la branche.</p>
 <p>"OUT" "M_BIT"</p>	 <p>"OUT" "M_BIT"</p>	Non disponible ¹	<p>Définir un opérande sur le front descendant.</p> <p>CONT : Le bit affecté "OUT" est VRAI lorsqu'un front descendant (1 à 0) est détecté sur le flux de courant entrant dans la bobine. L'état du flux de courant entrant traverse toujours la bobine en tant qu'état du flux de courant sortant. La bobine N peut être située n'importe où dans le réseau.</p> <p>LOG : Le bit affecté "OUT" est VRAI lorsqu'un front descendant (1 à 0) est détecté sur l'état logique au niveau de la connexion d'entrée de la boîte ou sur l'affectation de bit d'entrée si la boîte est située au début d'une branche. L'état logique d'entrée traverse toujours la boîte en tant qu'état logique de sortie. La boîte N= peut être située n'importe où dans la branche.</p>

¹ Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

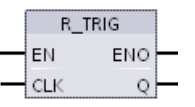
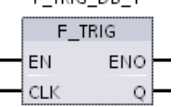
8.1 Opérations logiques sur bits

Tableau 8- 15P_TRIG et N_TRIG

CONT/LOG	SCL	Description
	Non disponible ¹	<p>Surveiller RLO (résultat logique) pour front montant.</p> <p>Le flux de courant ou l'état logique de la sortie Q est VRAI lorsqu'un front montant (0 à 1) est détecté sur l'état de l'entrée CLK (LOG) ou sur le flux de courant entrant CLK (CONT).</p> <p>En CONT, l'instruction P_TRIG ne peut pas être située au début ou à la fin d'un réseau. En LOG, l'instruction P_TRIG peut être située n'importe où, excepté à la fin d'une branche.</p>
	Non disponible ¹	<p>Surveiller RLO pour front descendant.</p> <p>Le flux de courant ou l'état logique de la sortie Q est VRAI lorsqu'un front descendant (1 à 0) est détecté sur l'état de l'entrée CLK (LOG) ou sur le flux de courant entrant CLK (CONT).</p> <p>En CONT, l'instruction N_TRIG ne peut pas être située au début ou à la fin d'un réseau. En LOG, l'instruction N_TRIG peut être située n'importe où, excepté à la fin d'une branche.</p>

¹ Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

Tableau 8- 16 Instructions R_TRIG et F_TRIG

CONT/LOG	SCL	Description
	<pre>"R_TRIG_DB" (CLK:=_in_, Q=>_bool_out_);</pre>	<p>Définir une variable sur le front montant.</p> <p>Le DB d'instance affecté est utilisé pour stocker l'état précédent de l'entrée CLK. Le flux de courant ou l'état logique de la sortie Q est VRAI lorsqu'un front montant (0 à 1) est détecté sur l'état de l'entrée CLK (LOG) ou sur le flux de courant entrant CLK (CONT).</p> <p>En CONT, l'instruction R_TRIG ne peut pas être située au début ou à la fin d'un réseau. En LOG, l'instruction R_TRIG peut être située n'importe où, excepté à la fin d'une branche.</p>
	<pre>"F_TRIG_DB" (CLK:=_in_, Q=>_bool_out_);</pre>	<p>Définir une variable sur le front descendant.</p> <p>Le DB d'instance affecté est utilisé pour stocker l'état précédent de l'entrée CLK. Le flux de courant ou l'état logique de la sortie Q est VRAI lorsqu'un front descendant (1 à 0) est détecté sur l'état de l'entrée CLK (LOG) ou sur le flux de courant entrant CLK (CONT).</p> <p>En CONT, l'instruction F_TRIG ne peut pas être située au début ou à la fin d'un réseau. En LOG, l'instruction F_TRIG peut être située n'importe où, excepté à la fin d'une branche.</p>

Pour R_TRIG et F_TRIG, la boîte de dialogue "Options d'appel" s'ouvre automatiquement lorsque vous insérez l'instruction dans le programme. Dans cette boîte de dialogue, vous pouvez indiquer si le bit de mémoire de front est stocké dans son propre bloc de données (mono-instance) ou en tant que variable locale (multi-instance) dans l'interface de bloc. Si vous créez un bloc de données séparé, vous le trouverez dans le dossier "Ressources programme" sous "Blocs de programme > Blocs système" dans l'arborescence de projet.

Tableau 8- 17 Types de données pour les paramètres (contacts/bobines P et N, P=, N=, P_TRIG and N_TRIG)

Paramètre	Type de données	Description
M_BIT	Bool	Bit de mémoire dans lequel l'état précédent de l'entrée est sauvegardé
IN	Bool	Bit d'entrée dont le front de transition est détecté
OUT	Bool	Bit de sortie qui indique qu'un front de transition a été détecté
CLK	Bool	Flux de courant ou bit d'entrée dont le front de transition est détecté
Q	Bool	Sortie qui indique qu'un front a été détecté

Toutes les instructions sur front utilisent un bit de mémoire (M_BIT : contacts/bobines P/N, P_TRIG/N_TRIG) ou (bit de DB d'instance : R_TRIG, F_TRIG) pour stocker l'état précédent du signal d'entrée surveillé. Un front est détecté par comparaison de l'état de l'entrée avec l'état précédent. Si les états indiquent un changement de l'entrée dans la direction concernée, un front est signalé par écriture de la valeur VRAI dans la sortie. Sinon, FAUX est écrit dans la sortie.

Remarque

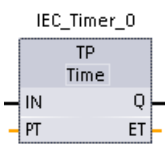
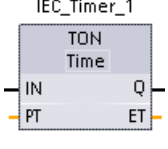
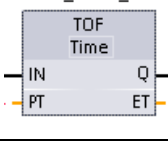
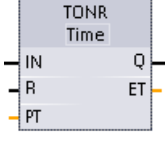
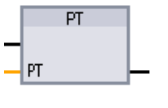
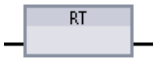
Les instructions sur front évaluent les valeurs de l'entrée et du bit de mémoire à chaque fois qu'elles sont exécutées, première exécution comprise. Vous devez prendre en compte l'état initial de l'entrée et du bit de mémoire dans la conception de votre programme et soit autoriser soit empêcher la détection de front lors du premier cycle.

Comme le bit de mémoire doit être conservé d'une exécution à la suivante, vous devez utiliser pour chaque instruction sur front un bit unique qui ne doit pas servir à un autre endroit dans le programme. Evitez également d'utiliser de la mémoire temporaire ou de la mémoire pouvant être affectée par d'autres fonctions système, telles que l'actualisation des E/S. Utilisez uniquement de la mémoire M, DB global ou statique (dans un DB d'instance) pour les affectations de M_BIT.

8.2 Temporisations

Vous utilisez les temporisations pour créer des retards programmés. Le nombre de temporisations que vous pouvez utiliser dans votre programme utilisateur est limité uniquement par la quantité de mémoire dans la CPU. Chaque temporisation utilise une structure de DB de 16 octets de type de données IEC_Timer pour conserver les données de la temporisation, cette structure étant indiquée au-dessus de la boîte ou de la bobine. STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 8- 18 Temporisations

Boîtes CONT/LOG	Bobines CONT	SCL	Description
 <p>IEC_Timer_0 TP Time IN Q PT ET</p>	<p>TP_DB —(TP)— "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TP (IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	La temporisation TP génère une impulsion de durée prédéfinie.
 <p>IEC_Timer_1 TON Time IN Q PT ET</p>	<p>TON_DB —(TON)— "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TON (IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	La temporisation TON met la sortie Q à 1 après un temps de retard prédéfini.
 <p>IEC_Timer_2 TOF Time IN Q PT ET</p>	<p>TOF_DB —(TOF)— "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TOF (IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	La temporisation TOF met la sortie Q à 0 après un temps de retard prédéfini.
 <p>IEC_Timer_3 TONR Time IN Q R ET PT</p>	<p>TONR_DB —(TONR)— "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TONR (IN:=_bool_in_, R:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	La temporisation TONR met la sortie Q à 1 après un temps de retard prédéfini. Le temps écoulé est accumulé sur plusieurs périodes de temps jusqu'à ce que l'entrée R soit utilisée pour réinitialiser le temps écoulé.
<p>LOG uniquement :</p>  <p>PT PT</p>	<p>TON_DB —(PT)— "PRESET_Tag"</p>	<pre>PRESET_TIMER (PT:=_time_in_, TIMER:=_iec_timer_in_);</pre>	La bobine PT (Charger durée) charge une nouvelle valeur de temps prédéfinie dans la temporisation IEC_Timer spécifiée.
<p>LOG uniquement :</p>  <p>RT</p>	<p>TON_DB —(RT)—</p>	<pre>RESET_TIMER (_iec_timer_in_);</pre>	La bobine RT (Réinitialiser temporisation) réinitialise la temporisation IEC_Timer spécifiée.

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans les exemples SCL, "IEC_Timer_0_DB" est le nom du DB d'instance.

Tableau 8- 19 Types de données pour les paramètres

Paramètre	Type de données	Description
Boîte : IN Bobine : Flux de courant	Bool	TP, TON et TONR : Boîte : 0=désactiver la temporisation, 1=activer la temporisation Bobine : pas de flux de courant=désactiver la temporisation, flux de courant=activer la temporisation TOF : Boîte : 0=activer la temporisation, 1=désactiver la temporisation Bobine : pas de flux de courant=activer la temporisation, flux de courant=désactiver la temporisation
R	Bool	Boîte TONR uniquement : 0=pas de réinitialisation 1=remettre le temps écoulé et le bit Q à 0
Boîte : PT Bobine : "PRESET_Tag"	Time	Boîte ou bobine de temporisation : Entrée valeur de temps prédéfinie
Boîte : Q Bobine : DBdata.Q	Bool	Boîte de temporisation : sortie de boîte Q ou bit Q dans les données du DB de temporisation Bobine de temporisation : vous pouvez accéder au bit Q uniquement dans les données du DB de temporisation
Boîte : ET Bobine : DBdata.ET	Time	Boîte de temporisation : sortie de boîte ET (temps écoulé) ou valeur de temps ET dans les données du DB de temporisation Bobine de temporisation : vous pouvez accéder à la valeur de temps ET uniquement dans les données du DB de temporisation

Tableau 8- 20 Effet des modifications de valeur dans les paramètres PT et IN

Temporisation	Modifications des paramètres de boîte PT et IN et des paramètres de bobine correspondants
TP	<ul style="list-style-type: none"> La modification de PT n'a pas d'effet pendant que la temporisation s'exécute. La modification de IN n'a pas d'effet pendant que la temporisation s'exécute.
TON	<ul style="list-style-type: none"> La modification de PT n'a pas d'effet pendant que la temporisation s'exécute. Le passage de IN à FAUX pendant que la temporisation s'exécute réinitialise et arrête la temporisation.
TOF	<ul style="list-style-type: none"> La modification de PT n'a pas d'effet pendant que la temporisation s'exécute. Le passage de IN à VRAI pendant que la temporisation s'exécute réinitialise et arrête la temporisation.
TONR	<ul style="list-style-type: none"> La modification de PT n'a pas d'effet pendant que la temporisation s'exécute mais a un effet lorsque la temporisation reprend. Le passage de IN à FAUX pendant que la temporisation s'exécute arrête la temporisation mais ne la réinitialise pas. Si IN repasse à VRAI, la temporisation recommence à s'écouler à partir de la valeur de temps accumulée.

Les valeurs PT (temps prédéfini) et ET (temps écoulé) sont sauvegardées dans les données DB IEC_TIMER spécifiées en tant qu'entiers doubles signés représentant des millisecondes. Les données TIME utilisent l'identificateur T# et peuvent être entrées en tant qu'unités de temps simple (T#200ms ou 200) ou unités de temps composées (T#2s_200ms).

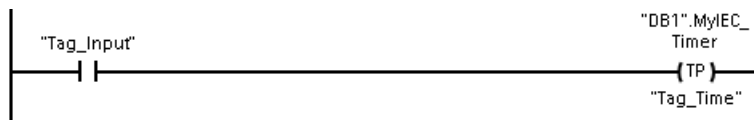
Tableau 8- 21 Taille et plage du type de données de TIME

Type de données	Taille	Plages valides ¹
TIME	32 bits, stockés comme données DInt	T#-24d_20h_31m_23s_648ms à T#24d_20h_31m_23s_647ms Stockés comme -2 147 483 648 ms à +2 147 483 647 ms

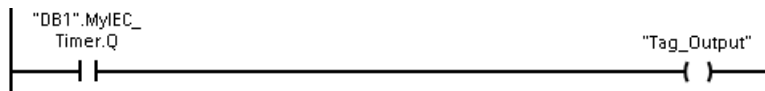
¹ La plage négative du type de données TIME indiquée ci-dessus ne peut pas être utilisée avec les temporisations. Les valeurs négatives de PT (temps prédéfini) sont mises à zéro lorsque l'instruction de temporisation est exécutée. ET (temps écoulé) est toujours une valeur positive.

Exemple de bobine de temporisation

Les bobines de temporisation -(TP)-, -(TON)-, -(TOF)- et -(TONR)- doivent être la dernière instruction dans un réseau CONT. Comme illustré dans l'exemple de temporisation, un contact dans un réseau suivant évalue le bit Q dans les données DB IEC_Timer de la bobine de temporisation. De même, vous devez accéder à l'élément ELAPSED dans les données DB IEC_timer si vous souhaitez utiliser la valeur de temps écoulé dans votre programme.



La temporisation d'impulsion démarre lors d'un front montant de la valeur de bit Tag_Input. La temporisation s'écoule pendant la durée indiquée par la valeur de temps Tag_Time.



Tant que la temporisation s'exécute, l'état de DB1.MyIEC_Timer.Q est égal à 1 et la valeur de Tag_Output est égale à 1. Lorsque la valeur Tag_Time est écoulée, DB1.MyIEC_Timer.Q est égal à 0 et la valeur de Tag_Output est égale à 0.

Bobines Réinitialiser temporisation -(RT)- et Charger durée -(PT)-

Ces bobines peuvent être utilisées avec des boîtes ou des bobines de temporisation et peuvent être placées en position médiane. L'état du flux de courant sortant de la bobine est toujours le même que l'état à l'entrée de la bobine. Lorsque la bobine -(RT)- est activée, l'élément de temps ELAPSED des données DB IEC_Timer indiquées est remis à 0. Lorsque la bobine -(PT)- est activée, l'élément de temps PRESET des données DB IEC_Timer indiquées reçoit la valeur de durée affectée.

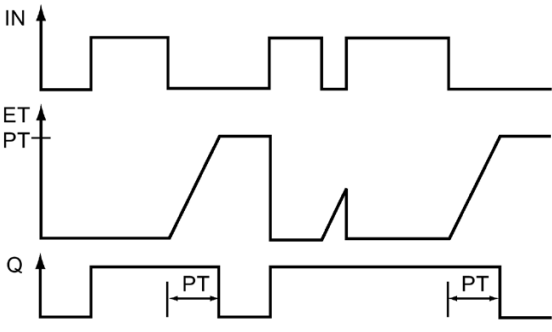
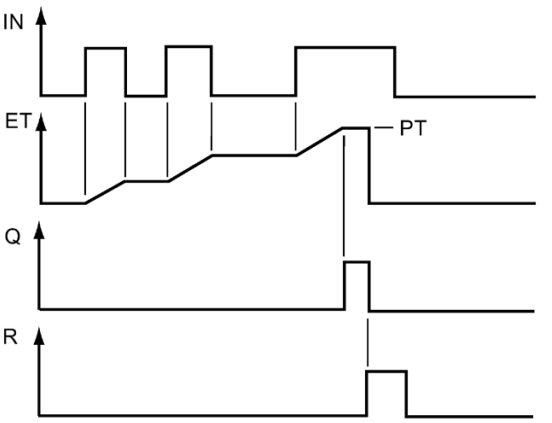
Remarque

Lorsque vous placez des temporisations dans un FB, vous pouvez sélectionner l'option "Bloc de données multi-instance". Les noms des structures de temporisation peuvent être différents avec des structures de données distinctes, mais les données de temporisation sont stockées dans un bloc de données unique et ne nécessitent pas de bloc de données séparé pour chaque temporisation. Cela réduit le temps de traitement et l'espace mémoire nécessaire pour la gestion des temporisations. Il n'y a pas d'interaction entre les structures des données de temporisation dans le DB multi-instance partagé.

Fonctionnement des temporisations

Tableau 8- 22Types de temporisations CEI

Temporisation	Chronogramme
<p>TP : Générer une impulsion</p> <p>La temporisation TP génère une impulsion de durée prédéfinie.</p>	
<p>TON : génère un retard à l'enclenchement</p> <p>La temporisation TON met la sortie Q à 1 après un temps de retard prédéfini.</p>	

Temporisation	Chronogramme
<p>TOF : génère un retard à la retombée</p> <p>La temporisation TOF met la sortie Q à 0 après un temps de retard prédéfini.</p>	
<p>TONR : temporisation</p> <p>La temporisation TONR met la sortie Q à 1 après un temps de retard prédéfini. Le temps écoulé est accumulé sur plusieurs périodes de temps jusqu'à ce que l'entrée R soit utilisée pour réinitialiser le temps écoulé.</p>	

Remarque

Dans la CPU, aucune ressource réservée n'est allouée à une temporisation spécifique. Au lieu de cela, chaque temporisation utilise sa propre structure de temporisation dans la mémoire DB et une temporisation CPU interne s'exécutant en continu pour effectuer la synchronisation.

Lorsqu'une temporisation est démarrée en raison d'un front à l'entrée d'une instruction TP, TON, TOF ou TONR, la valeur de la temporisation CPU interne s'exécutant en continu est copiée dans l'élément START de la structure de DB allouée à cette instruction de temporisation. Cette valeur de démarrage reste inchangée tant que la temporisation continue à s'exécuter et est utilisée plus tard à chaque fois que la temporisation est actualisée. A chaque fois que la temporisation est démarrée, une nouvelle valeur de démarrage est chargée de la temporisation CPU interne dans la structure de la temporisation.

Lorsqu'une temporisation est actualisée, la valeur de démarrage décrite ci-dessus est soustraite de la valeur en cours de la temporisation CPU interne pour déterminer le temps écoulé. Le temps écoulé est alors comparé à la valeur prédéfinie pour déterminer l'état du bit Q de la temporisation. Les éléments ELAPSED et Q sont alors actualisés dans la structure de DB allouée à cette temporisation. Notez que le temps écoulé est limité à la valeur prédéfinie (la temporisation ne continue pas à cumuler le temps écoulé une fois la valeur prédéfinie atteinte).

Une temporisation est actualisée uniquement lorsque :

- une instruction de temporisation (TP, TON, TOF ou TONR) est exécutée,
- l'élément ELAPSED de la structure de temporisation dans le DB est référencé directement par une instruction,
- l'élément Q de la structure de temporisation dans le DB est référencé directement par une instruction.

Programmation des temporisations

Vous devez tenir compte des conséquences suivantes du fonctionnement des temporisations lors de la planification et de la création de votre programme utilisateur :

- Plusieurs actualisations d'une temporisation peuvent avoir lieu dans le même cycle. La temporisation est actualisée à chaque fois que l'instruction de temporisation (TP, TON, TOF, TONR) est exécutée et à chaque fois que l'élément ELAPSED ou Q de la structure de temporisation est utilisé comme paramètre d'une autre instruction exécutée. Cela constitue un avantage si vous voulez disposer des dernières données de temps (sensiblement une lecture directe de la temporisation). Toutefois, si vous désirez avoir des valeurs cohérentes durant tout un cycle de programme, vous devez placer votre instruction de temporisation avant toutes les autres instructions ayant besoin de ces valeurs et utiliser des variables connectées aux sorties Q et ET de l'instruction de temporisation plutôt que les éléments ELAPSED et Q de la structure de DB de temporisation.
- Il peut y avoir des cycles pendant lesquels une temporisation n'est pas actualisée. Il est possible de démarrer une temporisation dans une fonction et de ne plus appeler cette fonction pendant un ou plusieurs cycles. Si aucune autre instruction référençant les éléments ELAPSED ou Q de la structure de temporisation n'est exécutée, la temporisation n'est pas actualisée. Il n'y aura pas d'actualisation tant que l'instruction de temporisation n'est pas réexécutée ou qu'une autre instruction utilisant l'élément ELAPSED ou Q de la structure de temporisation comme paramètre n'est pas exécutée.
- Bien que cela ne soit pas typique, vous pouvez affecter la même structure de temporisation à plusieurs instructions de temporisation. Mais en général, pour éviter des interactions imprévisibles, vous n'utiliserez qu'une instruction de temporisation (TP, TON, TOF, TONR) par structure de temporisation.

- Les temporisations à auto-réinitialisation sont utiles pour déclencher des actions qui doivent se produire périodiquement. Typiquement, vous créez des temporisations à auto-réinitialisation en plaçant un contact à ouverture qui référence le bit de temporisation devant l'instruction de temporisation. Ce réseau de temporisation est typiquement situé au-dessus d'un ou de plusieurs réseaux dépendants qui utilisent le bit de temporisation pour déclencher des actions. Lorsque la temporisation expire (le temps écoulé atteint la valeur prédéfinie), le bit de temporisation est à 1 pour un cycle, permettant ainsi à la logique de réseau dépendante pilotée par le bit de temporisation de s'exécuter. Lors de l'exécution suivante du réseau de temporisation, le contact à ouverture est désactivé, ce qui réinitialise la temporisation et efface le bit de temporisation. Au cycle suivant, le contact à ouverture est activé ce qui redémarre la temporisation. Lors de la création d'une telle temporisation à auto-réinitialisation, n'utilisez pas l'élément Q de la structure de DB de temporisation comme paramètre pour le contact à ouverture devant l'instruction de temporisation, mais utilisez la variable connectée à la sortie Q de l'instruction de temporisation. En effet, il ne faut pas accéder à l'élément Q de la structure de DB de temporisation, car cela provoque une actualisation de la temporisation et que, si la temporisation est actualisée en raison du contact à ouverture, le contact réinitialisera l'instruction de temporisation immédiatement. La sortie Q de l'instruction de temporisation ne sera pas à 1 pour un cycle et les réseaux dépendants ne s'exécuteront pas.

Conservation des données de temporisation après une transition MARCHE-ARRET-MARCHE ou une mise hors tension puis sous tension de la CPU

Si une session MARCHE se termine par passage à l'état ARRET ou mise hors tension puis sous tension de la CPU et qu'une nouvelle session MARCHE est démarrée, les données de temporisation sauvegardées à l'état MARCHE précédent sont perdues à moins que la structure de données de temporisation n'ait été définie comme rémanente (temporisations TP, TON, TOF et TONR).

Lorsque vous acceptez les valeurs par défaut dans la boîte de dialogue d'options d'appel après avoir inséré une temporisation dans l'éditeur de programmes, un DB d'instance qui **ne peut pas être défini comme rémanent** vous est automatiquement affecté. Pour assurer la rémanence de vos données de temporisation, vous devez donc utiliser soit un DB global, soit un DB multi-instance.

Affectation d'un DB global pour sauvegarder les données de temporisation comme données rémanentes

Cette option fonctionne quel que soit l'endroit où se trouve la temporisation (OB, FC ou FB).

1. Créez un DB global :
 - Double-cliquez sur "Ajouter nouveau bloc" dans l'arborescence de projet.
 - Cliquez sur l'icône de bloc de données (DB).
 - Choisissez DB global comme type.
 - Vérifiez que la case "Optimisé" est cochée pour le type de DB si vous désirez pouvoir définir des éléments individuels de ce DB comme rémanents. L'autre option de type de DB "Standard - compatible avec S7-300/400" permet uniquement de définir tous les éléments de DB soit comme rémanents, soit comme non rémanents.
 - Cliquez sur OK.
2. Ajoutez une ou des structures de temporisation au DB :
 - Dans le nouveau DB global, ajoutez une nouvelle variable statique en utilisant le type de données IEC_Timer.
 - Cochez la case dans la colonne "Rémanence" afin que cette structure soit rémanente.
 - Répétez cette procédure pour créer des structures pour toutes les temporisations que vous voulez sauvegarder dans ce DB. Vous pouvez soit placer chaque structure de temporisation dans un DB global unique, soit placer plusieurs structures de temporisation dans le même DB global. Vous pouvez également placer d'autres variables statiques en plus des temporisations dans ce DB global. Insérer plusieurs structures de temporisation dans le même DB global permet de réduire le nombre total de vos blocs.
 - Renommez les structures de temporisation si vous le souhaitez.
3. Ouvrez dans l'éditeur le bloc de programme dans lequel vous voulez placer une temporisation rémanente (OB, FC ou FB).
4. Placez la temporisation à l'emplacement désiré.
5. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur le bouton Annuler.
6. Au-dessus de la nouvelle temporisation, tapez le nom (sans utiliser l'icône d'aide à la navigation) du DB global et de la structure de temporisation que vous avez créés auparavant (exemple : "Data_block_3.Static_1").

Affectation d'un DB multi-instance pour sauvegarder les données de temporisation comme données rémanentes

Cette option ne fonctionne que si vous placez la temporisation dans un FB.

Cette option dépend du fait que les propriétés du FB indiquent ou non "Accès au bloc optimisé" (accès symbolique uniquement). Pour vérifier comment l'attribut d'accès est configuré pour un FB existant, cliquez avec le bouton droit de la souris sur le FB dans l'arborescence de projet, sélectionnez "Propriétés", puis sélectionnez "Attributs".

Si le FB indique "Accès au bloc optimisé" (accès symbolique uniquement) :

1. Ouvrez le FB dans l'éditeur.
2. Placez la temporisation à l'emplacement désiré dans le FB.
3. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur l'icône Multiinstance. L'option Multiinstance n'est disponible que si l'instruction est insérée dans un FB.
4. Dans la boîte de dialogue des options d'appel, renommez la temporisation si vous le souhaitez.
5. Cliquez sur OK. L'instruction de temporisation apparaît dans l'éditeur et la structure IEC_TIMER apparaît dans l'interface de FB sous Statique.
6. Si nécessaire, ouvrez l'éditeur d'interface de FB (vous devrez peut-être cliquer sur la petite flèche pour agrandir la vue).
7. Sous Statique, localisez la structure de temporisation qui vient d'être créée pour vous.
8. Dans la colonne Rémanence de cette structure de temporisation, changez la sélection à "Rémanence". Plus tard, à chaque fois que ce FB sera appelé dans un autre bloc de programme, un DB d'instance sera créé avec cette définition d'interface qui contient la structure de temporisation définie comme rémanente.

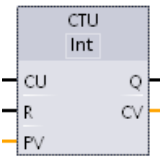
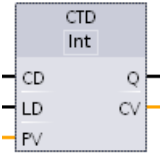
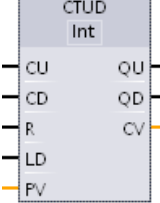
Si le FB n'indique pas "Accès au bloc optimisé", le type d'accès au bloc est standard, ce qui est compatible avec les configurations S7-300/400 classiques et autorise l'accès symbolique et l'accès direct. Procédez comme suit pour affecter un bloc multiinstance à un FB avec accès standard :

1. Ouvrez le FB dans l'éditeur.
2. Placez la temporisation à l'emplacement désiré dans le FB.
3. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur l'icône "Multiinstance". L'option Multiinstance n'est disponible que si l'instruction est insérée dans un FB.
4. Dans la boîte de dialogue des options d'appel, renommez la temporisation si vous le souhaitez.
5. Cliquez sur OK. L'instruction de temporisation apparaît dans l'éditeur et la structure IEC_TIMER apparaît dans l'interface de FB sous Statique.
6. Ouvrez le bloc qui utilisera ce FB.
7. Placez ce FB à l'emplacement désiré. Cela entraîne la création d'un bloc de données d'instance pour ce FB.

8. Ouvrez le bloc de données d'instance qui a été créé lorsque vous avez placé le FB dans l'éditeur.
9. Sous Statique, localisez la structure de temporisation qui vous intéresse. Dans la colonne Rémanence de cette structure de temporisation, cochez la case pour rendre cette structure rémanente.

8.3 Compteurs

Tableau 8- 23 Compteurs

CONT/LOG	SCL	Description
<p>"Counter name"</p> 	<pre>"IEC_Counter_0_DB".CTU (CU:=_bool_in, R:=_bool_in, PV:=_in, Q=>_bool_out, CV=>_out);</pre>	<p>Utilisez les compteurs pour compter des événements de programme internes et des événements de processus externes. Chaque compteur utilise une structure sauvegardée dans un bloc de données afin de conserver les données du compteur. Vous affectez le bloc de données lors du placement du compteur dans l'éditeur.</p> <ul style="list-style-type: none"> • CTU est un compteur de comptage. • CTD est un compteur de décomptage. • CTUD est un compteur de comptage et de décomptage.
<p>"Counter name"</p> 	<pre>"IEC_Counter_0_DB".CTD (CD:=_bool_in, LD:=_bool_in, PV:=_in, Q=>_bool_out, CV=>_out);</pre>	
<p>"Counter name"</p> 	<pre>"IEC_Counter_0_DB".CTUD D(CU:=_bool_in, CD:=_bool_in, R:=_bool_in, LD:=_bool_in, PV:=_in, QU=>_bool_out, QD=>_bool_out, CV=>_out);</pre>	

- 1 Pour CONT et LOG : Sélectionnez le type de données de la valeur de comptage dans la liste déroulante sous le nom de l'instruction.
- 2 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 3 Dans les exemples SCL, "IEC_Counter_0_DB" est le nom du DB d'instance.

Tableau 8- 24 Types de données pour les paramètres

Paramètre	Type de données ¹	Description
CU, CD	Bool	Compter ou décompter, d'une valeur de 1
R (CTU, CTUD)	Bool	Remise à zéro de la valeur de comptage
LD (CTD, CTUD)	Bool	Commande de chargement pour la valeur prédéfinie
PV	SInt, Int, DInt, USInt, UInt, UDInt	Valeur de comptage prédéfinie
Q, QU	Bool	Vrai si CV >= PV
QD	Bool	Vrai si CV <= 0
CV	SInt, Int, DInt, USInt, UInt, UDInt	Valeur de comptage en cours

¹ La plage numérique des valeurs de comptage dépend du type de données que vous sélectionnez. Si la valeur de comptage est un entier non signé, vous pouvez décompter jusqu'à zéro ou compter jusqu'à la limite de plage. Si la valeur de comptage est un entier signé, vous pouvez décompter jusqu'à la limite entière négative et compter jusqu'à la limite entière positive.

Le nombre de compteurs que vous pouvez utiliser dans votre programme utilisateur est limité uniquement par la quantité de mémoire dans la CPU. Les compteurs utilisent la quantité de mémoire suivante :

- Un compteur utilise 3 octets pour les types de données SInt ou USInt.
- Un compteur utilise 6 octets pour les types de données Int ou UInt.
- Un compteur utilise 12 octets pour les types de données DInt ou UDInt.

Ces instructions utilisent des compteurs logiciels dont la vitesse de comptage maximale est limitée par la vitesse d'exécution de l'OB dans lequel ils se trouvent. Il faut exécuter l'OB dans lequel se trouvent les instructions suffisamment souvent pour détecter toutes les transitions des entrées CU ou CD. Utilisez l'instruction CTRL_HSC (Page 573) pour disposer d'opérations de comptage plus rapides.

Remarque

Lorsque vous placez des compteurs dans un FB, vous pouvez sélectionner l'option "DB multi-instance". Les noms des structures de compteur peuvent alors être différents avec des structures de données distinctes, mais les données de compteur sont contenues dans un seul DB et vous n'avez pas besoin d'un DB distinct pour chaque compteur. Cela réduit le temps de traitement et l'espace mémoire nécessaire pour les compteurs. Il n'y a pas d'interaction entre les structures de données de compteur dans le DB multi-instance partagé.

Fonctionnement des compteurs

Tableau 8- 25 Fonctionnement de CTU (comptage)

Compteur	Fonctionnement
<p>Le compteur CTU incrémente de 1 lorsque la valeur du paramètre CU passe de 0 à 1. Le chronogramme illustre le fonctionnement d'un compteur CTU avec une valeur de comptage entière non signée (PV = 3).</p> <ul style="list-style-type: none"> • Si la valeur du paramètre CV (valeur de comptage en cours) est supérieure ou égale à la valeur du paramètre PV (valeur prédéfinie), le paramètre de sortie Q du compteur est égal à 1. • Si la valeur du paramètre de réinitialisation R passe de 0 à 1, la valeur de comptage en cours est remise à 0. 	

Tableau 8- 26 Fonctionnement de CTD (décomptage)

Compteur	Fonctionnement
<p>Le compteur CTD décrémente de 1 lorsque la valeur du paramètre CD passe de 0 à 1. Le chronogramme illustre le fonctionnement d'un compteur CTD avec une valeur de comptage entière non signée (PV = 3).</p> <ul style="list-style-type: none"> • Si la valeur du paramètre CV (valeur de comptage en cours) est inférieure ou égale à 0, le paramètre de sortie Q du compteur est égal à 1. • Si la valeur du paramètre LOAD passe de 0 à 1, la valeur dans le paramètre PV (valeur prédéfinie) est chargée dans le compteur en tant que nouvelle valeur de comptage en cours CV. 	

Tableau 8- 27 Fonctionnement de CTUD (comptage et décomptage)

Compteur	Fonctionnement
<p>Le compteur CTUD incrémente ou décrémente de 1 en cas de passage de 0 à 1 de l'entrée de comptage (CU) ou de décomptage (CD). Le chronogramme illustre le fonctionnement d'un compteur CTUD avec une valeur de comptage entière non signée (PV = 4).</p> <ul style="list-style-type: none"> • Si la valeur du paramètre CV est supérieure ou égale à la valeur du paramètre PV (valeur prédéfinie), le paramètre de sortie QU du compteur est égal à 1. • Si la valeur du paramètre CV est inférieure ou égale à zéro, le paramètre de sortie QD du compteur est égal à 1. • Si la valeur du paramètre LOAD passe de 0 à 1, la valeur dans le paramètre PV est chargée dans le compteur en tant que nouvelle valeur de comptage en cours CV. • Si la valeur du paramètre de réinitialisation R passe de 0 à 1, la valeur de comptage en cours est remise à 0. 	<p>The diagram shows the following signal behavior:</p> <ul style="list-style-type: none"> CU (Count Up): A series of pulses that cause the counter value to increase from 0 to 5. CD (Count Down): A series of pulses that cause the counter value to decrease from 5 to 3. R (Reset): A single pulse that resets the counter value to 0. LOAD: A pulse that loads the value 4 into the counter. CV (Counter Value): A staircase signal that follows the sequence: 0, 1, 2, 3, 4, 5, 4, 3, 4, 5, 0. QU (Upper Limit Flag): A signal that is high when CV ≥ 4 and low otherwise. QD (Lower Limit Flag): A signal that is high when CV ≤ 0 and low otherwise.

Conservation des données de compteur après une transition MARCHE-ARRET-MARCHE ou une mise hors tension puis sous tension de la CPU

Si une session MARCHE se termine par passage à l'état ARRET ou mise hors tension puis sous tension de la CPU et qu'une nouvelle session MARCHE est démarrée, les données de compteur sauvegardées à l'état MARCHE précédent sont perdues à moins que la structure de données de compteur n'ait été définie comme rémanente (compteurs CTU, CTD et CTUD).

Lorsque vous acceptez les valeurs par défaut dans la boîte de dialogue d'options d'appel après avoir inséré un compteur dans l'éditeur de programmes, un DB d'instance qui **ne peut pas être défini comme rémanent** vous est automatiquement affecté. Pour assurer la rémanence de vos données de compteur, vous devez donc utiliser soit un DB global, soit un DB multi-instance.

Affectation d'un DB global pour sauvegarder les données de compteur comme données rémanentes

Cette option fonctionne quel que soit l'endroit où se trouve le compteur (OB, FC ou FB).

1. Créez un DB global :

- Double-cliquez sur "Ajouter nouveau bloc" dans l'arborescence de projet.
- Cliquez sur l'icône de bloc de données (DB).
- Choisissez DB global comme type.
- Vérifiez que la case "Adressage symbolique uniquement" est cochée si vous désirez pouvoir définir des éléments individuels de ce DB comme rémanents.
- Cliquez sur OK.

2. Ajoutez une ou des structures de compteur au DB :

- Dans le nouveau DB global, ajoutez une nouvelle variable statique en utilisant l'un des types de données de compteur. Veillez à tenir compte du type que vous voulez utiliser pour la valeur prédéfinie et la valeur de comptage en cours.
- Cochez la case dans la colonne "Rémanence" afin que cette structure soit rémanente.
- Répétez cette procédure pour créer des structures pour tous les compteurs que vous voulez sauvegarder dans ce DB. Vous pouvez soit placer chaque structure de compteur dans un DB global unique, soit placer plusieurs structures de compteur dans le même DB global. Vous pouvez également placer d'autres variables statiques en plus des compteurs dans ce DB global. Insérer plusieurs structures de compteur dans le même DB global permet de réduire le nombre total de vos blocs.
- Renommez les structures de compteur si vous le souhaitez.

3. Ouvrez dans l'éditeur le bloc de programme dans lequel vous voulez placer un compteur rémanent (OB, FC ou FB).

4. Placez le compteur à l'emplacement désiré.

5. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur le bouton Annuler. Vous devriez maintenant voir un nouveau compteur avec "???" juste au-dessus et en dessous du nom d'instruction.

6. Au-dessus du nouveau compteur, tapez le nom (sans utiliser l'icône d'aide à la navigation) du DB global et de la structure de compteur que vous avez créés auparavant (exemple : "Data_block_3.Static_1"). Le type correspondant de la valeur prédéfinie et de la valeur en cours est alors complété (exemple : UInt pour une structure IEC_UCounter).

Type de données compteur	Type correspondant pour les valeurs prédéfinie et de comptage en cours
IEC_Counter	INT
IEC_SCounter	SINT
IEC_DCounter	DINT
IEC_UCounter	UINT
IEC_USCounter	USINT
IEC_UDCounter	UDINT

Affectation d'un DB multi-instance pour sauvegarder les données de compteur comme données rémanentes

Cette option ne fonctionne que si vous placez le compteur dans un FB.

Cette option dépend du fait que les propriétés du FB indiquent ou non "Accès au bloc optimisé" (accès symbolique uniquement). Pour vérifier comment l'attribut d'accès est configuré pour un FB existant, cliquez avec le bouton droit de la souris sur le FB dans l'arborescence de projet, sélectionnez "Propriétés", puis sélectionnez "Attributs".

Si le FB indique "Accès au bloc optimisé" (accès symbolique uniquement) :

1. Ouvrez le FB dans l'éditeur.
2. Placez le compteur à l'emplacement désiré dans le FB.
3. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur l'icône "Multiinstance". L'option Multiinstance n'est disponible que si l'instruction est insérée dans un FB.
4. Dans la boîte de dialogue des options d'appel, renommez le compteur si vous le souhaitez.
5. Cliquez sur OK. L'instruction de compteur apparaît dans l'éditeur avec le type INT pour la valeur prédéfinie et la valeur de comptage et la structure IEC_COUNTER apparaît dans l'interface de FB sous Statique.
6. Si désiré, changez le type INT dans le compteur en l'un des autres types proposés. La structure de compteur changera de manière correspondante.
7. Si nécessaire, ouvrez l'éditeur d'interface de FB (vous devrez peut-être cliquer sur la petite flèche pour agrandir la vue).
8. Sous Statique, localisez la structure de compteur qui vient d'être créée pour vous.
9. Dans la colonne Rémanence de cette structure de compteur, changez la sélection à "Rémanence". Plus tard, à chaque fois que ce FB sera appelé dans un autre bloc de programme, un DB d'instance sera créé avec cette définition d'interface qui contient la structure de compteur définie comme rémanente.

Si le FB n'indique pas "Accès au bloc optimisé", le type d'accès au bloc est standard, ce qui est compatible avec les configurations S7-300/400 classiques et autorise l'accès symbolique et l'accès direct. Procédez comme suit pour affecter un bloc multiinstance à un FB avec accès standard :

1. Ouvrez le FB dans l'éditeur.
2. Placez le compteur à l'emplacement désiré dans le FB.
3. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur l'icône "Multiinstance". L'option Multiinstance n'est disponible que si l'instruction est insérée dans un FB.
4. Dans la boîte de dialogue des options d'appel, renommez le compteur si vous le souhaitez.
5. Cliquez sur OK. L'instruction de compteur apparaît dans l'éditeur avec le type INT pour la valeur prédéfinie et la valeur de comptage et la structure IEC_COUNTER apparaît dans l'interface de FB sous Statique.
6. Si désiré, changez le type INT dans le compteur en l'un des autres types proposés. La structure de compteur changera de manière correspondante.

7. Ouvrez le bloc qui utilisera ce FB.
8. Placez ce FB à l'emplacement désiré. Cela entraîne la création d'un bloc de données d'instance pour ce FB.
9. Ouvrez le bloc de données d'instance qui a été créé lorsque vous avez placé le FB dans l'éditeur.
10. Sous Statique, localisez la structure de compteur qui vous intéresse. Dans la colonne Rémanence de cette structure de compteur, cochez la case pour rendre cette structure rémanente.

Type indiqué dans l'instruction de compteur
(pour la valeur prédéfinie et la valeur de
comptage)

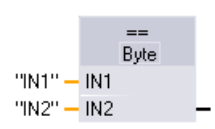
Type de structure correspondante montré
dans l'interface de FB

INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

8.4 Comparaison

8.4.1 Instructions Comparer valeurs

Tableau 8- 28 Instructions de comparaison

CONT	LOG	SCL	Description
<pre>"IN1" == Byte "IN2"</pre>	 <pre>"IN1" IN1 "IN2" IN2</pre>	<pre>out := in1 = in2; or IF in1 = in2 THEN out := 1; ELSE out := 0; END IF;</pre>	<p>Compare deux valeurs ayant le même type de données. Lorsque la comparaison est vraie, le contact de comparaison CONT est activé. Lorsque la comparaison est vraie, la sortie de la boîte de comparaison LOG est VRAIE.</p>

1 Pour CONT et LOG : Cliquez sur le nom de l'instruction ("==" par exemple) pour modifier le type de comparaison dans la liste déroulante. Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.

Tableau 8- 29 Types de données pour les paramètres

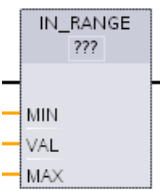
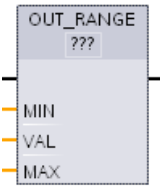
Paramètre	Type de données	Description
IN1, IN2	Byte, Word, DWord, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, String, WString, Char, Time, Date, TOD, DTL, constante	Valeurs à comparer

Tableau 8- 30 Types de comparaison

Type de relation	La comparaison est vraie si ...
=	IN1 est égal à IN2
<>	IN1 est différent de IN2
>=	IN1 est supérieur ou égal à IN2
<=	IN1 est inférieur ou égal à IN2
>	IN1 est supérieur à IN2
<	IN1 est inférieur à IN2

8.4.2 IN_RANGE (Valeur dans la plage) et OUT_RANGE (Valeur en dehors de la plage)

Tableau 8- 31 Instructions Valeur dans la plage et Valeur en dehors de la plage

CONT/LOG	SCL	Description
	<pre>out := IN_RANGE (min, val, max) ;</pre>	<p>Teste si une valeur d'entrée se situe ou non à l'intérieur d'une plage de valeurs indiquée.</p> <p>Lorsque la comparaison est vraie, la sortie de la boîte est VRAIE.</p>
	<pre>out := OUT_RANGE (min, val, max) ;</pre>	

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.

Tableau 8- 32 Types de données pour les paramètres

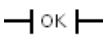
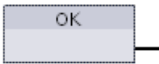
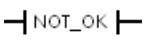
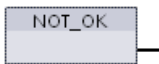
Paramètre	Type de données ¹	Description
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, constante	Entrées du comparateur

¹ Les paramètres d'entrée MIN, VAL et MAX doivent avoir le même type de données.

- La comparaison IN_RANGE est vraie si : $MIN \leq VAL \leq MAX$
- La comparaison OUT_RANGE est vraie si : $VAL < MIN$ ou $VAL > MAX$

8.4.3 OK (Contrôler validité) et NOT_OK (Contrôler invalidité)

Tableau 8- 33 Instructions OK (Contrôler validité) et Not OK (Contrôler invalidité)

CONT	LOG	SCL	Description
"II" 	"II" 	Non disponible	Teste si une référence de donnée d'entrée est un nombre réel valide selon la spécification IEEE 754.
"II" 	"II" 	Non disponible	

- ¹ Pour CONT et LOG : Lorsque le résultat du test est vrai, le contact CONT est activé et transmet le flux de courant. Lorsque le résultat du test est vrai, la sortie de la boîte LOG est VRAIE.

Tableau 8- 34 Types de données pour le paramètre

Paramètre	Type de données	Description
IN	Real, LReal	Données d'entrée

Tableau 8- 35 Fonctionnement

Instruction	Le test concernant le nombre réel est vrai si :
OK	La valeur d'entrée est un nombre réel valide ¹ .
NOT_OK	La valeur d'entrée n'est pas un nombre réel valide ¹ .

- ¹ Une valeur Real ou LReal est invalide s'il s'agit de +/- INF (l'infini), NaN (pas un nombre) ou d'une valeur dénormalisée. Une valeur dénormalisée est un nombre très proche de zéro. La CPU remplace la valeur dénormalisée par un zéro dans les calculs.

8.4.4 Instructions de comparaison de variante et tableau

8.4.4.1 Instructions de comparaison d'égalité et de non-égalité

La CPU S7-1200 fournit des instructions pour interroger le type de données d'une variable vers laquelle un opérande Variant pointe pour une égalité ou une non-égalité avec le type de données de l'autre opérande.

De plus, la CPU S7-1200 fournit des instructions pour interroger le type de données d'un élément de tableau pour une égalité ou une non-égalité avec le type de données de l'autre opérande.

Dans ces instructions, vous comparez <Operand1> et <Operand2>. <Operand1> doit avoir le type de données Variant. <Operand2> peut être un type de données élémentaire d'un type de données API. Dans CONT et LOG, <Operand1> est l'opérande au-dessus de l'instruction. Dans CONT, <Operand2> est l'opérande en dessous de l'instruction.

Pour toutes les instructions, le résultat de la logique (RLO) est 1 (vrai) si le test d'égalité ou de non-égalité est réussi, et 0 (faux) si ce n'est pas le cas.

Les instructions de comparaison de type égalité et non-égalité sont les suivantes :

- EQ_Type (Comparaison d'égalité d'un type de données et du type de données d'une variable)
- NE_Type (Comparaison d'inégalité d'un type de données et du type de données d'une variable)
- EQ_ElemType (Comparaison d'égalité du type de données d'un élément ARRAY et du type de données d'une variable)
- NE_ElemType (Comparaison d'inégalité du type de données d'un élément ARRAY et du type de données d'une variable)

Tableau 8- 36 Instructions EQ et NE

CONT	LOG	SCL	Description
<pre>#Operand1 ┌ EQ_Type ─┘ *Operand2*</pre>	<pre>#Operand1 EQ_Type *Operand2* ─ IN2 OUT ─</pre>	Non disponible	Vérifie si la variable vers laquelle la variante pointe dans Operand1 est du même type de données que la variable dans Operand2.
<pre>#Operand1 ┌ NE_Type ─┘ *Operand2*</pre>	<pre>#Operand1 NE_Type *Operand2* ─ IN2 OUT ─</pre>	Non disponible	Vérifie si la variable vers laquelle la variante pointait dans Operand1 est d'un type de données différent de la variable dans Operand2.
<pre>#Operand1 ┌ EQ_ElemType ─┘ *Operand2*</pre>	<pre>#Operand1 EQ_ElemType *Operand2* ─ IN2 OUT ─</pre>	Non disponible	Vérifie si l'élément du tableau vers lequel la variante pointe dans Operand1 est du même type de données que la variable dans Operand2.
<pre>#Operand1 ┌ NE_ElemType ─┘ *Operand2*</pre>	<pre>#Operand1 NE_ElemType *Operand2* ─ IN2 OUT ─</pre>	Non disponible	Vérifie si l'élément du tableau vers lequel la variante pointe dans Operand1 est d'un type de données différent de la variable dans Operand2.

Tableau 8- 37 Types de données pour les paramètres

Paramètre	Type de données	Description
Operand1	Variant	Premier opérande
Operand2	Chaîne de bits, entiers, nombres à virgule flottante, temporisations, date et heure, chaînes de caractères, TABLEAU, types de données API	Deuxième opérande

8.4.4.2 Instructions de comparaison nulle

Vous pouvez utiliser les instructions IS_NULL et NOT_NULL pour déterminer si l'entrée pointe réellement vers un objet ou non.

Pour les deux instructions, <Operand> doit avoir le type de données Variant.

Tableau 8- 38 Instructions IS_NULL (Interrogation si pointeur NUL) et NOT_NULL (Interrogation si pointeur NON NUL)

CONT	LOG	SCL	Description
#Operand ┌IS_NULL┐	#Operand IS_NULL OUT-	Non disponible	Vérifie si la variable vers laquelle la Variant pointe dans Operand est nulle et n'est donc pas un objet.
#Operand ┌NOT_NULL┐	#Operand NOT_NULL OUT-	Non disponible	Vérifie si la variable vers laquelle la Variant pointe dans Operand n'est pas nulle et pointe donc vers un objet.

Tableau 8- 39 Types de données pour les paramètres

Paramètre	Type de données	Description
Operand	Variant	Opérande pour évaluer nul ou pas nul.

8.4.4.3 IS_ARRAY (vérifier le TABLEAU)

Vous pouvez utiliser l'instruction "Vérifier le TABLEAU" pour interroger si la Variant pointe vers une variable du type de données Array.

L'<opérande> doit avoir le type de données Variant.

Les instructions renvoient 1 (vrai) si l'opérande est un tableau.

Tableau 8- 40 IS_ARRAY (vérifier le TABLEAU)

CONT	LOG	SCL	Description
#Operand IS_ARRAY	#Operand IS_ARRAY OUT -	IS_ARRAY(_variant_in_)	Vérifie si la variable vers laquelle la Variant pointe dans Operand est un tableau.

Tableau 8- 41 Types de données pour les paramètres

Paramètre	Type de données	Description
Operand	Variant	Opérande pour évaluer s'il s'agit d'un tableau.

8.5 Fonctions mathématiques

8.5.1 CALCULATE (Calculer)

Tableau 8- 42 Instruction CALCULATE

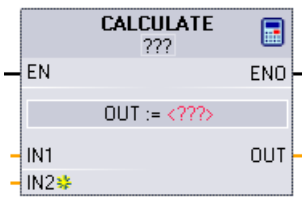
CONT/LOG	SCL	Description
	<p>Utilisez les expressions mathématiques SCL standard pour créer l'équation.</p>	<p>L'instruction CALCULATE permet de créer une fonction mathématique qui opère sur les entrées (IN1, IN2, .. INn) et fournit le résultat dans OUT, conformément à l'équation que vous définissez.</p> <ul style="list-style-type: none"> • Sélectionnez d'abord un type de données. Toutes les entrées et la sortie doivent avoir le même type de données. • Pour ajouter une autre entrée, cliquez sur l'icône à côté de la dernière entrée.

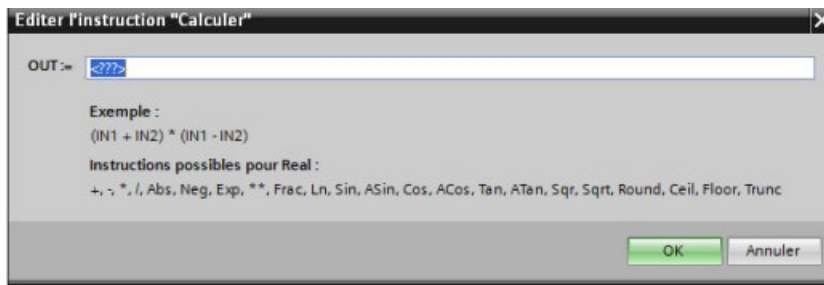
Tableau 8- 43 Types de données pour les paramètres

Paramètre	Type de données ¹
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord

¹ Les paramètres IN et OUT doivent avoir le même type de données (avec conversion implicite des paramètres d'entrée). Par exemple, une valeur SINT pour une entrée sera convertie en valeur INT ou REAL si OUT est un entier (INT) ou un réel (REAL).

Cliquez sur l'icône du calculateur pour ouvrir la boîte de dialogue et définir votre fonction mathématique. Vous entrez votre équation sous forme d'entrées (telles que IN1 et IN2) et d'opérations. Lorsque vous cliquez sur "OK" pour sauvegarder la fonction, le dialogue crée automatiquement les entrées pour l'instruction CALCULATE.

La boîte de dialogue représente un exemple et une liste d'opérations possibles que vous pouvez inclure en fonction du type de données du paramètre OUT :



Remarque

Vous devez également créer une entrée pour chaque constante dans votre fonction. La valeur constante sera alors entrée dans l'entrée associée pour l'instruction CALCULATE.

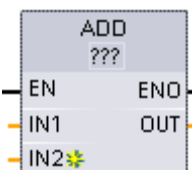
Le fait d'entrer des constantes sous forme d'entrées vous permet de copier l'instruction CALCULATE à d'autres endroits dans votre programme utilisateur sans avoir à changer la fonction. Vous pouvez alors modifier les valeurs ou les variables des entrées pour l'instruction sans modifier la fonction.

Lorsque CALCULATE est exécuté et que toutes les opérations individuelles dans le calcul s'achèvent avec succès, ENO est égal à 1. Sinon, ENO est égal à 0.

Pour un exemple d'instruction CALCULATE, voir "Création d'une équation complexe avec une instruction simple (Page 42)".

8.5.2 Instructions Addition, Soustraction, Multiplication et Division

Tableau 8- 44 Instructions Addition, Soustraction, Multiplication et Division

CONT/LOG	SCL	Description
	<pre>out := in1 + in2; out := in1 - in2; out := in1 * in2; out := in1 / in2;</pre>	<ul style="list-style-type: none"> • ADD : Addition ($IN1 + IN2 = OUT$) • SUB : Soustraction ($IN1 - IN2 = OUT$) • MUL : Multiplication ($IN1 * IN2 = OUT$) • DIV : Division ($IN1 / IN2 = OUT$) <p>Une division sur des entiers tronque la partie fractionnaire du quotient afin de produire une sortie entière.</p>

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 45 Types de données pour les paramètres (CONT et LOG)

Paramètre	Type de données ¹	Description
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, constante	Entrées de l'opération mathématique
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Sortie de l'opération mathématique

¹ Les paramètres IN1, IN2 et OUT doivent avoir le même type de données.



Pour ajouter une entrée ADD ou MUL, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants et sélectionnez la commande "Insérer entrée".

Pour supprimer une entrée, cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

8.5 Fonctions mathématiques

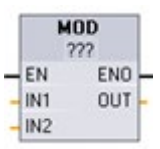
Lorsqu'elle est validée (EN = 1), l'instruction mathématique effectue l'opération spécifiée sur les valeurs d'entrée (IN1 et IN2) et mémorise le résultat dans l'adresse de mémoire indiquée par le paramètre de sortie (OUT). Une fois l'opération achevée avec succès, l'instruction met ENO à 1.

Tableau 8- 46Etat de ENO

ENO	Description
1	Pas d'erreur
0	Le résultat de l'opération mathématique se situerait hors de la plage de nombres valide du type de données sélectionné. La partie la moins significative du résultat qui tient dans la taille de destination est renvoyée.
0	Division par 0 (IN2 = 0) : Le résultat est indéfini et zéro est renvoyé.
0	Real/LReal : Si l'une des valeurs d'entrée est NaN (pas un nombre), NaN est renvoyé.
0	ADD Real/LReal : Si les deux valeurs d'entrée sont l'infini avec des signes différents, il s'agit d'une opération interdite et NaN est renvoyé.
0	SUB Real/LReal : Si les deux valeurs d'entrée sont l'infini avec le même signe, il s'agit d'une opération interdite et NaN est renvoyé.
0	MUL Real/LReal : Si une valeur d'entrée est zéro et l'autre est l'infini, il s'agit d'une opération interdite et NaN est renvoyé.
0	DIV Real/LReal : Si les deux valeurs d'entrée sont zéro ou l'infini, il s'agit d'une opération interdite et NaN est renvoyé.

8.5.3 MOD (Calculer le reste de la division)

Tableau 8- 47 Instruction Modulo (Calculer le reste de la division)

CONT/LOG	SCL	Description
	<pre>out := in1 MOD in2;</pre>	<p>Vous pouvez utiliser l'instruction MOD pour renvoyer le reste d'une division entière. La valeur dans l'entrée IN1 est divisée par la valeur dans l'entrée IN2 et le reste est renvoyé dans la sortie OUT.</p>

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 48 Types de données pour les paramètres

Paramètre	Type de données ¹	Description
IN1 et IN2	SInt, Int, DInt, USInt, UInt, UDIInt, constante	Entrées de l'opération modulo
OUT	SInt, Int, DInt, USInt, UInt, UDIInt	Sortie de l'opération modulo


1 Les paramètres IN1, IN2 et OUT doivent avoir le même type de données.

Tableau 8- 49 Valeurs de ENO

ENO	Description
1	Pas d'erreur
0	Valeur IN2 = 0 ; la valeur zéro est affectée à OUT.

8.5.4 NEG (Créer le complément à deux)

Tableau 8- 50 Instruction NEG (Créer le complément à deux)

CONT/LOG	SCL	Description
	- (in) ;	L'instruction NEG inverse le signe arithmétique de la valeur dans le paramètre IN et mémorise le résultat dans le paramètre OUT.

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 51 Types de données pour les paramètres

Paramètre	Type de données ¹	Description
IN	SInt, Int, DInt, Real, LReal, constante	Entrée de l'opération mathématique
OUT	SInt, Int, DInt, Real, LReal	Sortie de l'opération mathématique



¹ Les paramètres IN et OUT doivent avoir le même type de données.

Tableau 8- 52 Etat de ENO

ENO	Description
1	Pas d'erreur
0	La valeur résultante se situe hors de la plage de nombres valide du type de données sélectionné. Exemple pour SInt : NEG (-128) donne comme résultat +128 qui dépasse le maximum du type de données.

8.5.5 INC (Incrémenter) et DEC (Décrémenter)

Tableau 8- 53 Instructions INC et DEC

CONT/LOG	SCL	Description
	<code>in_out := in_out + 1;</code>	Incrémente une valeur entière signée ou non signée : valeur IN_OUT +1 = valeur IN_OUT
	<code>in_out := in_out - 1;</code>	Décrémente une valeur entière signée ou non signée : valeur IN_OUT - 1 = valeur IN_OUT

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 54 Types de données pour les paramètres


Paramètre	Type de données	Description
IN/OUT	SInt, Int, DInt, USInt, UInt, UDIInt	Entrée et sortie de l'opération mathématique

Tableau 8- 55 Etat de ENO

ENO	Description
1	Pas d'erreur
0	La valeur résultante se situe hors de la plage de nombres valide du type de données sélectionné. Exemple pour SInt : INC (+127) donne comme résultat +128, qui dépasse le maximum du type de données.

8.5.6 ABS (Valeur absolue)

Tableau 8- 56 Instruction ABS (Valeur absolue)

CONT/LOG	SCL	Description
	<code>out := ABS(in);</code>	Calcule la valeur absolue d'un nombre réel ou entier signé dans le paramètre IN et mémorise le résultat dans le paramètre OUT.

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 57 Types de données pour les paramètres

Paramètre	Type de données ¹	Description
IN	SInt, Int, DInt, Real, LReal	Entrée de l'opération mathématique
OUT	SInt, Int, DInt, Real, LReal	Sortie de l'opération mathématique

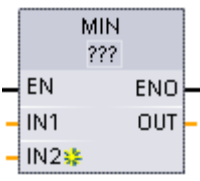
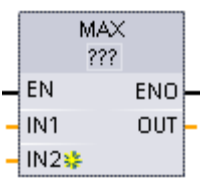
¹ Les paramètres IN et OUT doivent avoir le même type de données.

Tableau 8- 58 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Le résultat de l'opération mathématique se situe hors de la plage de nombres valide du type de données sélectionné. Exemple pour SInt : ABS (-128) donne comme résultat +128, qui dépasse le maximum du type de données.

8.5.7 MIN (Calculer le minimum) et MAX (Calculer le maximum)

Tableau 8- 59 Instructions MIN (Calculer le minimum) et MAX (Calculer le maximum)

CONT/LOG	SCL	Description
	<pre>out:= MIN(in1:=_variant_in_, in2:=_variant_in_ [...in32]);</pre>	L'instruction MIN compare la valeur de deux paramètres IN1 et IN2 et affecte la valeur minimale (inférieure) au paramètre OUT.
	<pre>out:= MAX(in1:=_variant_in_, in2:=_variant_in_ [...in32]);</pre>	L'instruction MAX compare la valeur de deux paramètres IN1 et IN2 et affecte la valeur maximale (supérieure) au paramètre OUT.

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 60 Types de données pour les paramètres

Paramètre	Type de données ¹	Description
IN1, IN2 [...IN32]	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD, constante	Entrées de l'opération mathématique (jusqu'à 32 entrées)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Sortie de l'opération mathématique

¹ Les paramètres IN1, IN2 et OUT doivent avoir le même type de données.



Pour ajouter une entrée, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants et sélectionnez la commande "Insérer entrée".

Pour supprimer une entrée, cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

Tableau 8- 61 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Pour le type de données Real uniquement : <ul style="list-style-type: none"> • Au moins une entrée n'est pas un nombre réel (NaN). • La valeur OUT résultante est +/- INF (l'infini).

8.5.8 LIMIT (Définir une limite)

Tableau 8- 62 Instruction LIMIT (Définir une limite)

CONT/LOG	SCL	Description
	<code>LIMIT(MN:=_variant_in_, IN:=_variant_in_, MX:=_variant_in_, OUT:=_variant_out_);</code>	L'instruction Limit teste si la valeur du paramètre IN se situe à l'intérieur de la plage de valeurs indiquée par les paramètres MIN et MAX and if not, clamps the value at MIN or MAX. et, si ce n'est pas le cas, limite la valeur à MIN ou MAX.

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 63 Types de données pour les paramètres

Paramètre	Type de données ¹	Description
MN, IN et MX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Time, Date, TOD ² , constante	Entrées de l'opération mathématique
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Time, Date, TOD	Sortie de l'opération mathématique

¹ Les paramètres MN, IN, MX et OUT doivent avoir le même type de données.

Si la valeur du paramètre IN se situe à l'intérieur de la plage indiquée, la valeur de IN est mémorisée dans le paramètre OUT. Si la valeur du paramètre IN se situe hors de la plage indiquée, OUT prend la valeur du paramètre MIN (si la valeur de IN est inférieure à la valeur MIN) ou la valeur du paramètre MAX (si la valeur de IN est supérieure à la valeur MAX).

Tableau 8- 64 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Real : Si une ou plusieurs des valeurs de MIN, IN et MAX sont NaN (pas des nombres), NaN est renvoyé.
0	Si MIN est supérieur à MAX, la valeur IN est affectée à OUT.

Exemples SCL :

- MyVal := LIMIT(MN:=10,IN:=53, MX:=40); //Résultat : MyVal = 40
- MyVal := LIMIT(MN:=10,IN:=37, MX:=40); //Résultat : MyVal = 37
- MyVal := LIMIT(MN:=10,IN:=8, MX:=40); //Résultat : MyVal = 10

8.5.9 Instructions exponentielle, logarithmique et trigonométriques

Vous utilisez les instructions à virgule flottante pour programmer des opérations mathématiques avec le type de données Real ou LReal :



- SQR : carré ($IN^2 = OUT$)
- SQRT : racine carrée ($\sqrt{IN} = OUT$)
- LN : logarithme népérien ($LN(IN) = OUT$)
- EXP : fonction exponentielle ($e^{IN} = OUT$), de base $e = 2,71828182845904523536$
- EXPT : élever à la puissance ($IN1^{IN2} = OUT$)

Vous devez toujours sélectionner le même type de données pour les paramètres IN1 et OUT de EXPT, à savoir Real ou LReal. Vous pouvez sélectionner le type de données pour l'exposant IN2 parmi de nombreux types de données.

- FRAC : décimales (partie fractionnaire du nombre à virgule flottante $IN = OUT$)
- SIN: sinus ($\sin(IN \text{ radians}) = OUT$)
- ASIN : arc sinus ($\arcsin(IN) = OUT \text{ radians}$), avec $\sin(OUT \text{ radians}) = IN$
- COS : cosinus ($\cos(IN \text{ radians}) = OUT$)
- ACOS : arc cosinus ($\arccos(IN) = OUT \text{ radians}$), avec $\cos(OUT \text{ radians}) = IN$
- TAN : tangente ($\tan(IN \text{ radians}) = OUT$)
- ATAN : arc tangente ($\arctan(IN) = OUT \text{ radians}$), avec $\tan(OUT \text{ radians}) = IN$

8.5 Fonctions mathématiques

Tableau 8- 65 Exemples d'instructions mathématiques à virgule flottante

CONT/LOG	SCL	Description
	<pre>out := SQR(in);</pre> <p>ou</p> <pre>out := in * in;</pre>	<p>Carré : $IN^2 = OUT$</p> <p>Par exemple : Si IN est égal à 9, alors OUT est égal à 81.</p>
	<pre>out := in1 ** in2;</pre>	<p>Élever à la puissance : $IN1^{IN2} = OUT$</p> <p>Par exemple : Si IN1 est égal à 3 et IN2 est égal à 2, alors OUT est égal à 9.</p>

- 1 Pour CONT et LOG : Cliquez sur "???" (près du nom de l'instruction) et sélectionnez un type de données dans le menu déroulant.
- 2 Pour SCL : Vous pouvez aussi utiliser les opérateurs mathématiques SCL de base pour créer les expressions mathématiques

Tableau 8- 66 Types de données pour les paramètres

Paramètre	Type de données	Description
IN, IN1	Real, LReal, constante	Entrées
IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, constante	Entrée exposant EXPT
OUT	Real, LReal	Sorties

Tableau 8- 67 État de ENO

ENO	Instruction	Situation d'erreur	Résultat (OUT)
1	Toutes	Pas d'erreur	Résultat valide
0	SQR	Le résultat dépasse la plage Real/LReal valide.	+INF
		IN est +/- NaN (pas un nombre)	+NaN
	SQRT	IN est négatif.	-NaN
		IN est +/- INF (l'infini) ou +/- NaN.	+/- INF ou +/- NaN
	LN	IN est 0.0, négatif, -INF ou -NaN.	-NaN
		IN est +INF ou +NaN.	+INF ou +NaN
	EXP	Le résultat dépasse la plage Real/LReal valide.	+INF
		IN est +/- NaN.	+/- NaN
	SIN, COS, TAN	IN est +/- INF ou +/- NaN.	+/- INF ou +/- NaN
	ASIN, ACOS	IN se situe en dehors de la plage valide de -1.0 à +1.0.	+NaN
		IN est +/- NaN.	+/- NaN
	ATAN	IN est +/- NaN.	+/- NaN
	FRAC	IN est +/- INF ou +/- NaN.	+NaN
EXPT	IN1 est +INF et IN2 n'est pas -INF.	+INF	

ENO	Instruction	Situation d'erreur	Résultat (OUT)
		IN1 est négatif ou -INF.	+NaN si IN2 est Real/LReal, -INF sinon
		IN1 ou IN2 est +/- NaN.	+NaN
		IN1 est 0.0 et IN2 est Real/LReal (seulement).	+NaN

8.6 Transfert


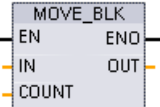
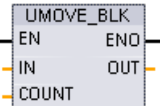
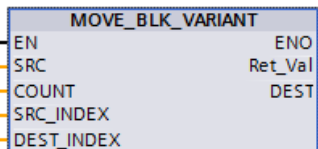
8.6.1 MOVE (Copier valeur), MOVE_BLK (Copier zone), UMOVE_BLK (Copier zone contiguë) et MOVE_BLK_VARIANT (Copier zone)

Utilisez les instructions de transfert pour copier des éléments de données à une nouvelle adresse mémoire et les convertir en un autre type de données. Les données source ne sont pas modifiées par le processus de transfert.

- L'instruction MOVE copie un élément de données unique de l'adresse source indiquée par le paramètre IN dans l'adresse de destination indiquée par le paramètre OUT.
- Les instructions MOVE_BLK et UMOVE_BLK disposent d'un paramètre COUNT supplémentaire. COUNT précise combien d'éléments de données doivent être copiés. Le nombre d'octets par élément copié dépend du type de données affecté aux noms de variable des paramètres IN et OUT dans la table des variables API.

8.6 Transfert

Tableau 8- 68 Instructions MOVE, MOVE_BLK, UMOVE_BLK et MOVE_BLK_VARIANT

CONT/LOG	SCL	Description
	<pre>out1 := in;</pre>	Copie un élément de données stocké à l'adresse indiquée dans une ou plusieurs nouvelles adresses. ¹
	<pre>MOVE_BLK (in:=_variant_in, count:=_uint_in, out=>_variant_out);</pre>	Copie de zone interruptible qui copie une zone d'éléments de données à une nouvelle adresse.
	<pre>UMOVE_BLK (in:=_variant_in, count:=_uint_in, out=>_variant_out);</pre>	Copie de zone contiguë qui copie une zone d'éléments de données à une nouvelle adresse.
	<pre>MOVE_BLK (SRC:=_variant_in, COUNT:=_uint_in, SRC_INDEX:=_dint_in, DEST_INDEX:=_dint_in, DEST=>_variant_out);</pre>	Copie le contenu d'une zone de mémoire source vers une zone de mémoire de destination. Vous pouvez copier un tableau complet ou des éléments d'un tableau vers un autre tableau du même type de données. La taille (nombre d'éléments) du tableau source et de destination peut être différente. Vous pouvez copier des éléments multiples ou uniques dans un tableau. Vous utilisez des types de données Variante pour pointer à la fois vers les tableaux source et de destination.

¹ Instruction MOVE : Pour ajouter une autre sortie en CONT ou LOG, cliquez sur l'icône de création à côté du paramètre de sortie. Pour SCL, utilisez des instructions d'affectation multiple. Vous pouvez également utiliser l'une des structures de boucle.

Tableau 8- 69 Types de données pour l'instruction MOVE

Paramètre	Type de données	Description
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC types de données, types de données API	Adresse de source
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC types de données, types de données API	Adresse de destination



Pour ajouter des sorties MOVE, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants et sélectionnez la commande "Insérer sortie".

Pour supprimer une sortie, cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants (lorsqu'il y a plus de sorties que les deux sorties d'origine) et sélectionnez la commande "Supprimer".

Tableau 8- 70 Types de données pour les instructions MOVE_BLK et UMOVE_BLK

Paramètre	Type de données	Description
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, WChar	Adresse de début de la source
COUNT	UInt	Nombre d'éléments de données à copier
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, WChar	Adresse de début de la destination

Tableau 8- 71 Types de données pour l'instruction MOVE_BLK_VARIANT

Paramètre	Type de données	Description
SRC	Variant (qui pointe vers un tableau ou un élément de tableau spécifique)	Bloc source à partir duquel effectuer une copie
COUNT	UDIInt	Nombre d'éléments de données à copier
SRC_INDEX	DInt	Index de base zéro dans le tableau SRC
DEST_INDEX	DInt	Index de base zéro dans le tableau DEST
RET_VAL	Int	Information d'erreur
DEST	Variant (qui pointe vers un tableau ou un élément de tableau spécifique)	Zone de destination dans laquelle copier le contenu du bloc source

Remarque

Règles pour les opérations de copie de données

- Pour copier le type de données Bool, utilisez SET_BF, RESET_BF, R, S, ou la bobine de sortie (CONT) (Page 240)
- Pour copier un seul type de données simple, utilisez MOVE.
- Pour copier un tableau de type de données simple, utilisez MOVE_BLK ou UMOVE_BLK.
- Pour copier une structure, utilisez MOVE.
- Pour copier une chaîne, utilisez S_MOVE (Page 357).
- Pour copier un seul caractère dans une chaîne, utilisez MOVE.
- Vous ne pouvez pas utiliser les instructions MOVE_BLK et UMOVE_BLK pour copier des tableaux ou des structures dans les zones de mémoire I, Q ou M.

8.6 Transfert

Les instructions MOVE_BLK et UMOVE_BLK diffèrent dans la façon dont les alarmes sont traitées :

- Les événements d'alarme sont **mis en file d'attente et traités** pendant l'exécution de MOVE_BLK. Utilisez l'instruction MOVE_BLK lorsque les données à l'adresse de destination de la copie ne sont pas utilisées dans un sous-programme d'OB d'alarme ou lorsque les données de destination n'ont pas besoin d'être cohérentes si elles sont utilisées. Si le fonctionnement de MOVE_BLK est interrompu, le dernier élément de données copié est complet et cohérent à l'adresse de destination. Le fonctionnement de MOVE_BLK reprend une fois l'OB d'alarme exécuté.
- Les événements d'alarme sont **mis en file d'attente mais ne sont pas traités** tant que l'exécution de UMOVE_BLK n'est pas achevée. Utilisez l'instruction UMOVE_BLK lorsque l'opération de copie doit être achevée et les données de destination cohérentes, avant l'exécution d'un sous-programme d'OB d'alarme. Pour plus d'informations, reportez-vous au paragraphe sur la cohérence des données (Page 204).

ENO est toujours vrai après l'exécution de l'instruction MOVE.

Tableau 8- 72Etat de ENO

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	Tous les COUNT éléments ont été copiés avec succès.
0	La plage source (IN) ou la plage de destination (OUT) dépasse la zone de mémoire disponible.	Les éléments qui entrent sont copiés. Il n'y a pas copie d'éléments partiels.

Tableau 8- 73 Codes d'erreur de l'instruction MOVE_BLK_VARIANT

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
80B4	Les types de données ne correspondent pas.
8151	L'accès au paramètre SRC n'est pas possible.
8152	L'opérande dans le paramètre SRC est un type invalide.
8153	Erreur de génération de code dans le paramètre SRC
8154	L'opérande dans le paramètre SRC est du type de données Bool.
8281	Le paramètre COUNT a une valeur invalide.
8382	La valeur dans le paramètre SRC_INDEX se situe en dehors des limites de la Variant.
8383	La valeur dans le paramètre SRC_INDEX se situe en dehors de la limite supérieure du tableau.
8482	La valeur dans le paramètre DEST_INDEX se situe en dehors des limites de la Variant.
8483	La valeur dans le paramètre DEST_INDEX se situe en dehors de la limite supérieure du tableau.
8534	Le paramètre DEST est protégé en écriture.
8551	L'accès au paramètre DEST n'est pas possible.
8552	L'opérande dans le paramètre DEST est un type invalide.
8553	Erreur de génération de code dans le paramètre DEST
8554	L'opérande dans le paramètre DEST est du type de données Bool.
*Vous pouvez afficher des codes d'erreur sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.	

8.6.2 Deserialize

Vous pouvez utiliser l'instruction "Deserialize" pour convertir la représentation séquentielle d'un type de données API (UDT) en un type de données API et pour remplir l'intégralité de son contenu. Si la comparaison est VRAIE, la sortie de la boîte est VRAIE.

La zone de mémoire qui comprend la représentation séquentielle d'un type de données API doit avoir le type de données Tableau d'octets et vous devez déclarer le bloc de données pour avoir un accès standard (non optimisé). Assurez-vous qu'il y a suffisamment d'espace mémoire avant la conversion.

L'instruction vous permet de reconverter de multiples représentations séquentielles de types de données API convertis en leurs types de données initiaux.

Remarque

Si vous souhaitez uniquement reconverter une représentation séquentielle unique d'un type de données API (UDT), vous pouvez également utiliser l'instruction "TRCV : Recevoir des données via la liaison de communication".

Tableau 8- 74 Instruction DESERIALIZE

CONT / LOG	SCL	Description
	<pre>ret_val := Deserialize(SRC_ARRAY:=_variant_in_, DEST_VARIABLE=>_variant_out); POS:= dint_inout);</pre>	Reconvertit la représentation séquentielle d'un type de données API (UDT) en un type de données API et remplit l'intégralité de son contenu

Tableau 8- 75 Paramètres pour l'instruction DESERIALIZE

Paramètre	Type	Type de données	Description
SRC_ARRAY	IN	Variant	Bloc de données global qui contient le flux de données
DEST_VARIABLE	INOUT	Variant	Variable dans laquelle le type de données API (UDT) converti est sauvegardé
POS	INOUT	Dint	Nombre d'octets utilisés par le type de données API converti
RET_VAL	OUT	Int	Information d'erreur

Tableau 8- 76 Paramètre RET_VAL

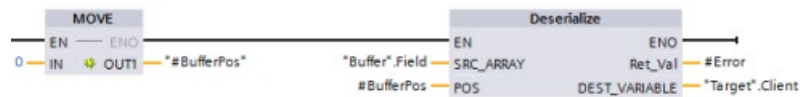
RET_VAL* (W#16#...)	Description
0000	Pas d'erreur
80B0	Les zones mémoire des paramètres SRC_ARRAY et DEST_VARIABLE se recourent.
8136	Le bloc de données dans le paramètre DEST_VARIABLE n'est pas un bloc avec un accès standard.
8150	Le type de données Variant dans le paramètre SRC_ARRAY ne contient pas de valeur.
8151	Erreur de génération de code dans le paramètre SRC_ARRAY.
8153	Il n'y a pas suffisamment de mémoire libre disponible dans le paramètre SRC_ARRAY.
8250	Le type de données Variant dans le paramètre DEST_VARIABLE ne contient pas de valeur.
8251	Erreur de génération de code dans le paramètre DEST_VARIABLE.
8254	Type de données invalide dans le paramètre DEST_VARIABLE.
8382	La valeur dans le paramètre POS se situe en dehors des limites du tableau.

*Vous pouvez visualiser les codes d'erreur sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.

Exemple : Instruction Deserialize

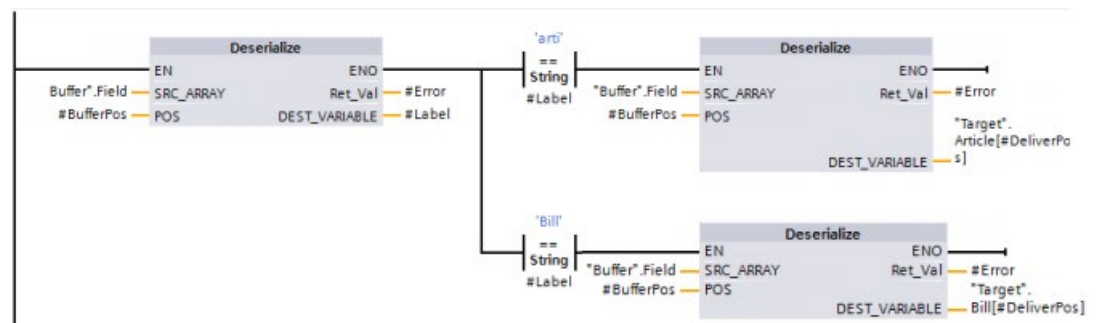
L'exemple suivant montre comment l'instruction fonctionne :

Réseau 1 :



L'instruction "MOVE" copie la valeur "0" dans la variable de bloc de données "#BufferPos". L'instruction Deserialize déséréalise ensuite la représentation séquentielle des données client depuis le bloc de données "Buffer" puis l'écrit dans le bloc de données "Target". L'instruction Deserialize calcule le nombre d'octets utilisés par les données converties et les sauvegarde dans la variable de bloc de données "#BufferPos".

Réseau 2 :



L'instruction "Deserialize" déséréalise la représentation séquentielle du flux de données vers lequel "Buffer" pointe et écrit les caractères dans l'opérande "#Label". La logique compare les caractères à l'aide des instructions de comparaison "arti" et "Bill". Si la comparaison pour "arti" = VRAI, les données sont des données d'article qui doivent être déséréalisées et écrites dans la structure de données "Article" du bloc de données "Target". Si la comparaison pour "Bill" = VRAIE, les données sont des données de facturation qui doivent être déséréalisées et écrites dans la structure de données "Bill" du bloc de données "Target".

Interface du bloc fonctionnel (ou de la fonction) :

	Nom	Type de données
1	Input	
2	DeliverPos	Int
3	Output	
4	InOut	
5	Static	
6	Temp	
7	BufferPos	DInt
8	Error	Int
9	Label	String[4]

Types de données API personnalisés :

La structure des deux type de données API (UDT) pour cet exemple est la suivante :

Article		
	Nom	Type de données
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client		
	Nom	Type de données
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Blocs de données :

Les deux blocs de données pour cet exemple sont les suivants :

Target		
	Nom	Type de données
1	Static	
2	Client	*Client*
3	Article	Array[0..10] of *Article*
4	Bill	Array[0..10] of Int

Buffer		
	Nom	Type de données
1	Static	
2	Field	Array[0..294] of Byte

8.6.3 Serialize

Vous pouvez utiliser l'instruction "Serialize" pour convertir plusieurs types de données API (UDT) en une représentation séquentielle sans aucune perte de structure.

Vous pouvez utiliser l'instruction pour sauvegarder provisoirement de multiples éléments de données structurés de votre programme sur une mémoire tampon, par exemple sur un bloc de données global, et les envoyer sur une autre CPU. La zone mémoire dans laquelle les types de données API convertis sont sauvegardés doit être du type de données TABLEAU d'OCTETS et être déclaré avec un accès standard. Assurez-vous qu'il y a suffisamment d'espace mémoire avant la conversion.

Le paramètre POS contient des informations concernant le nombre d'octets utilisés par les types de données API convertis.

Remarque

Si vous ne souhaitez envoyer qu'un type de données API (UDT) unique, vous pouvez utiliser l'instruction "TSEND : Envoyer des données via la liaison de communication".

Tableau 8- 77 Instruction SERIALIZE

CONT / LOG	SCL	Description
	<pre>ret_val := Serialize(SRC_VARIABLE=>_variant_in_, DEST_ARRAY:=_variant_out_, POS:=_dint_inout_);</pre>	Convertit un type de données API (UDT) en une représentation séquentielle.

Tableau 8- 78 Paramètres pour l'instruction SERIALIZE

Paramètre	Type	Type de données	Description
SRC_VARIABLE	IN	Variant	Type de données API (UDT) qui doit être converti en une représentation sérielle
DEST_ARRAY	INOUT	Variant	Bloc de données dans lequel le flux de données générées doit être archivé
POS	INOUT	DInt	Nombre d'octets utilisés par les types de données API convertis. Le paramètre POS calculé a une base zéro.
RET_VAL	OUT	Int	Information d'erreur

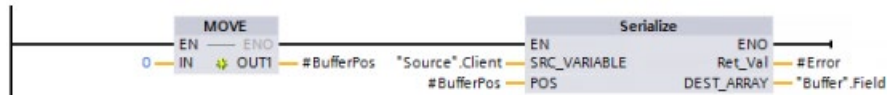
Tableau 8- 79 Paramètre RET_VAL

RET_VAL* (W#16#...)	Description
0000	Pas d'erreur
80B0	Les zones mémoire des paramètres SRC_VARIABLE et DEST_ARRAY se recourent.
8150	Le type de données Variant dans le paramètre SRC_VARIABLE ne contient pas de valeur.
8152	Erreur de génération de code dans le paramètre SRC_VARIABLE.
8236	Le bloc de données dans le paramètre DEST_ARRAY n'est pas un bloc avec un accès standard.
8250	Le type de données Variant dans le paramètre DEST_ARRAY ne contient pas de valeur.
8252	Erreur de génération de code dans le paramètre DEST_ARRAY.
8253	Il n'y a pas suffisamment de mémoire libre disponible dans le paramètre DEST_ARRAY.
8254	Type de données invalide dans le paramètre DEST_VARIABLE.
8382	La valeur dans le paramètre POS se situe en dehors des limites du tableau.
*Vous pouvez visualiser les codes d'erreur sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.	

Exemple : instruction Serialize

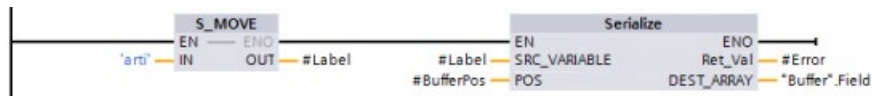
L'exemple suivant montre comment l'instruction fonctionne :

Réseau 1 :



L'instruction "MOVE" copie la valeur "0" dans le paramètre "#BufferPos". L'instruction Serialize sérialise les données client depuis le bloc de données "Source" puis les écrit en représentation séquentielle dans le bloc de données "Buffer". L'instruction sauvegarde le nombre d'octets utilisés par la représentation séquentielle dans le paramètre "#BufferPos".

Réseau 2 :



La logique insère maintenant du texte de séparation afin de simplifier la désérialisation ultérieure de la représentation séquentielle. L'instruction "S_MOVE" copie la chaîne de texte "arti" dans le paramètre "#Label". L'instruction "Serialize" écrit ces caractères après les données client source dans le bloc de données "Buffer". L'instruction ajoute le nombre d'octets dans la chaîne de texte "arti" au nombre déjà stocké dans le paramètre "#BufferPos".

Réseau 3 :



L'instruction "Serialize" sérialise les données d'un article spécifique, qui sont calculées lors de l'exécution, depuis le bloc de données "Source", et les écrit en représentation séquentielle dans le bloc de données "Buffer" après les caractères "arti"

Interface de bloc :

	Nom	Type de données
1	Input	
2	DeliverPos	Int
3	Output	
4	InOut	
5	Static	
6	Temp	
7	BufferPos	DInt
8	Error	Int
9	Label	String[4]

Types de données API personnalisés :

La structure des deux type de données API (UDT) pour cet exemple est la suivante :

Article		
	Nom	Type de données
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client		
	Nom	Type de données
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Blocs de données :

Les deux blocs de données pour cet exemple sont les suivants :

Source		
	Nom	Type de données
1	Static	
2	Client	"Client"
3	Article	Array[0..10] of "Article"

Buffer		
	Nom	Type de données
1	Static	
2	Field	Array[0..294] of Byte

8.6.4 FILL_BLK (Compléter zone) et UFILL_BLK (Compléter zone contiguë)

Tableau 8- 80 Instructions FILL_BLK et UFILL_BLK



CONT/LOG	SCL	Description
	<pre>FILL_BLK (in:=_variant_in, count:=int, out=>_variant_out);</pre>	L'instruction Compléter zone remplit une zone d'adresses avec des copies d'un élément de données indiqué.
	<pre>UFILL_BLK (in:=_variant_in, count:=int, out=>_variant_out);</pre>	L'instruction Compléter zone contiguë remplit une zone d'adresses avec des copies d'un élément de données indiqué.

Tableau 8- 81 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Adresse source des données
COUNT	UDInt, USInt, UInt	Nombre d'éléments de données à copier
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Adresse de destination des données

Remarque

Règles pour les opérations de remplissage de zone

- Pour compléter avec le type de données BOOL, utilisez SET_BF, RESET_BF, R, S ou la bobine de sortie (CONT).
- Pour compléter avec un seul type de données simple, utilisez MOVE.
- Pour compléter un tableau avec un type de données simple, utilisez FILL_BLK ou UFILL_BLK.
- Pour compléter un seul caractère dans une chaîne, utilisez MOVE.
- Vous ne pouvez pas utiliser les instructions FILL_BLK et UFILL_BLK pour compléter des tableaux dans les zones de mémoire I, Q ou M.

Les instructions FILL_BLK et UFILL_BLK copient l'élément de données source IN dans la destination dont l'adresse initiale est indiquée par le paramètre OUT. Le processus de copie est répété de manière à compléter un bloc d'adresses adjacentes jusqu'à ce que le nombre de copies soit égal au paramètre COUNT.

Les instructions FILL_BLK et UFILL_BLK diffèrent dans la façon dont les alarmes sont traitées :


- Les événements d'alarme sont **mis en file d'attente et traités** pendant l'exécution de FILL_BLK. Utilisez l'instruction FILL_BLK lorsque les données à l'adresse de destination de la copie ne sont pas utilisées dans un sous-programme d'OB d'alarme ou lorsque les données de destination n'ont pas besoin d'être cohérentes si elles sont utilisées.
- Les événements d'alarme sont **mis en file d'attente mais ne sont pas traités** tant que l'exécution de UFILL_BLK n'est pas achevée. Utilisez l'instruction UFILL_BLK lorsque l'opération de copie doit être achevée et que les données de destination doivent être cohérentes avant l'exécution d'un sous-programme d'OB d'alarme.

Tableau 8- 82 État de ENO

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	L'élément IN a été copié avec succès dans les COUNT destinations.
0	La plage de destination (OUT) dépasse la zone de mémoire disponible.	Les éléments qui entrent sont copiés. Il n'y a pas copie d'éléments partiels.

8.6.5 SWAP (Permuter octets)

Tableau 8- 83 Instruction SWAP

CONT/LOG	SCL	Description
	<pre>out := SWAP(in);</pre>	Inverse l'ordre des octets dans des éléments de données à deux et quatre octets. L'ordre des bits à l'intérieur de chaque octet n'est pas modifié. ENO est toujours VRAI après l'exécution de l'instruction SWAP.

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 84 Types de données pour les paramètres

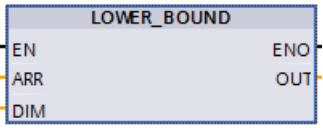
Paramètre	Type de données	Description
IN	Word, DWord	Octets de données dans l'ordre
OUT	Word, DWord	Octets de données dans l'ordre inverse

Exemple 1	Paramètre IN = MB0 (avant exécution)	Paramètre OUT = MB4 (après exécution)
Adresse	MW0 MB1	MW4 MB5
W#16#1234	12 34	34 12
WORD	OPFo OPFa	OPFo OPFa

Exemple 2	Paramètre IN = MB0 (avant exécution)	Paramètre OUT = MB4 (après exécution)
Adresse	MD0 MB1 MB2 MB3	MD4 MB5 MB6 MB7
DW#16# 12345678	12 34 56 78	78 56 34 12
DWORD	OPFo OPFa	OPFo OPFa

8.6.6 LOWER_BOUND (Lire la limite inférieure d'un ARRAY)

Tableau 8- 85 Instruction LOWER_BOUND

CONT/LOG	SCL	Description
	<pre>out := LOWER_BOUND (ARR:=_variant_in_, DIM:=_udint_in_);</pre>	<p>Vous pouvez déclarer des variables avec ARRAY[*] dans l'interface de bloc. Pour ces variables locales, vous pouvez lire les limites du tableau (ARRAY). Il faut indiquer la dimension requise dans le paramètre DIM.</p> <p>L'instruction LOWER_BOUND (Lire la limite inférieure d'un ARRAY) permet de lire la limite inférieure variable d'un tableau.</p>

Paramètres

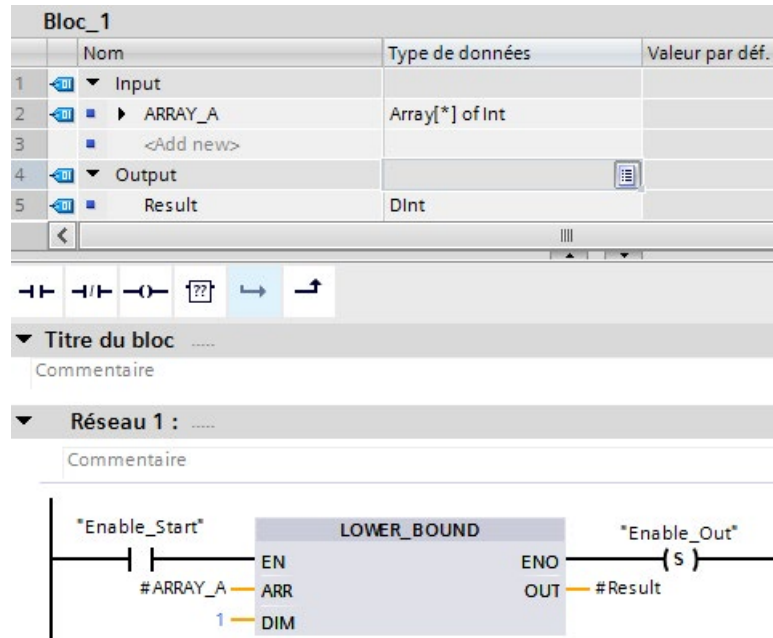
Le tableau suivant présente les paramètres de l'instruction LOWER_BOUND (Lire la limite inférieure d'un ARRAY) :

Paramètres	Déclaration	Type de données	Zone de mémoire	Description
EN	Input	BOOL	I, Q, M, D, L	Entrée de validation
ENO	Output	BOOL	I, Q, M, D, L	La sortie de validation ENO est à l'état logique "0" si l'une des conditions suivantes est vraie : <ul style="list-style-type: none"> • L'entrée de validation EN a l'état logique "0". • La dimension indiquée dans l'entrée DIM n'existe pas.
ARR	Input	ARRAY [*]	FB : section InOut FC : sections Input et InOut	Tableau (ARRAY) dont la limite inférieure variable doit être lue.
DIM	Input	UDINT	I, Q, M, D, L ou constante	Dimension du tableau dont la limite inférieure variable doit être lue.
OUT	Output	DINT	I, Q, M, D, L	Résultat

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Exemple

Dans l'interface de bloc de la fonction (FC), le paramètre d'entrée ARRAY_A est un tableau unidimensionnel à dimensions variables.



Si l'opérande "Enable_Start" renvoie l'état logique "1", la CPU exécute l'instruction LOWER_BOUND . L'instruction lit la limite inférieure variable #ARRAY_A dans le tableau unidimensionnel. Si l'instruction s'exécute sans erreur, elle met à 1 l'opérande "Enable_Out" et inscrit la limite inférieure du tableau dans l'opérande "Result".

8.6.7 UPPER_BOUND (Lire la limite supérieure d'un ARRAY)

Tableau 8- 86 Instruction LOWER_BOUND

CONT/LOG	SCL	Description
<p>The diagram shows the UPPER_BOUND function block with inputs EN, ARR, and DIM on the left, and outputs ENO and OUT on the right.</p>	<pre> out := UPPER_BOUND (ARR:= _variant_in_ , DIM:= _udint_in_); </pre>	<p>Vous pouvez déclarer des variables avec ARRAY[*] dans l'interface de bloc. Pour ces variables locales, vous pouvez lire les limites du tableau (ARRAY). Il faut indiquer la dimension requise dans le paramètre DIM.</p> <p>L'instruction UPPER_BOUND (Lire la limite supérieure d'un ARRAY) permet de lire la limite supérieure variable d'un tableau.</p>

Paramètres

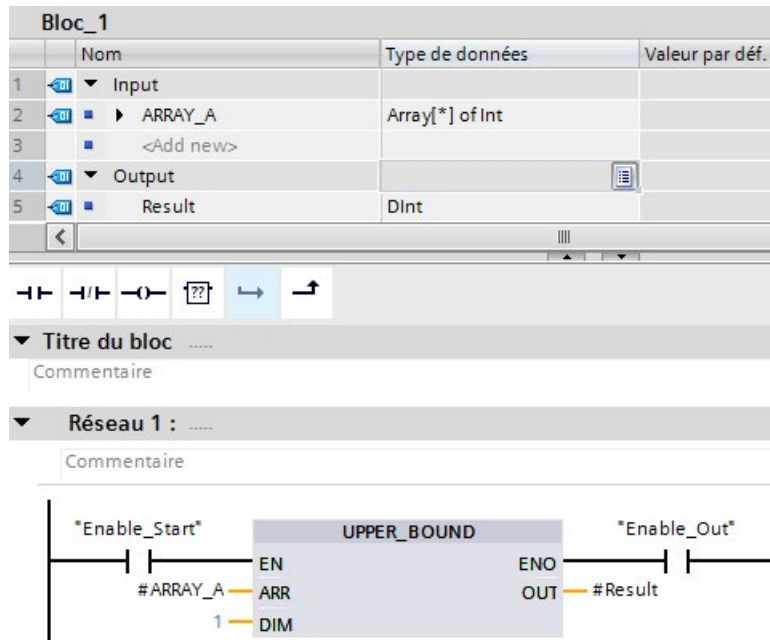
Le tableau suivant présente les paramètres de l'instruction UPPER_BOUND (Lire la limite supérieure d'un ARRAY) :

Paramètres	Déclaration	Type de données	Zone de mémoire	Description
EN	Input	BOOL	I, Q, M, D, L	Entrée de validation
ENO	Output	BOOL	I, Q, M, D, L	Sortie de validation
ARR	Input	ARRAY [*]	FB : section InOut FC : sections Input et InOut	Tableau (ARRAY) dont la limite supérieure variable doit être lue.
DIM	Input	UDINT	I, Q, M, D, L ou constante	Dimension du tableau dont la limite supérieure variable doit être lue.
OUT	Output	DINT	I, Q, M, D, L	Résultat

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Exemple

Dans l'interface de bloc de la fonction (FC), le paramètre d'entrée ARRAY_A est un tableau unidimensionnel à dimensions variables.



Si l'opérande "Enable_Start" renvoie l'état logique "1", la CPU exécute l'instruction. L'instruction lit la limite supérieure variable #ARRAY_A dans le tableau unidimensionnel. Si l'instruction s'exécute sans erreur, elle met à 1 l'opérande "Enable_Out" et inscrit le résultat dans l'opérande "Result".

8.6.8 Instructions de lecture de mémoire / écriture dans la mémoire

8.6.8.1 PEEK et POKE (SCL uniquement)

SCL fournit les instructions PEEK et POKE qui vous permettent de lire et d'écrire dans les blocs de données, les E/S ou la mémoire. Vous indiquez des paramètres pour les décalages d'octet ou de bit spécifiques pour l'opération.

Remarque

Pour utiliser les instructions PEEK et POKE avec des blocs de données, vous devez utiliser des blocs de données standard (non optimisés). Notez également que les instructions PEEK et POKE ne font que transférer des données. Elles n'ont pas connaissance des types de données aux diverses adresses.

```
PEEK(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_);
```

Lit l'octet désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple avec un bloc de données :

```
%MB100 := PEEK(area:=16#84,
               dbNumber:=1, byteOffset:=#i);
```

Exemple avec l'octet d'entrée IB3 :

```
%MB100 := PEEK(area:=16#81,
               dbNumber:=0, byteOffset:=#i); // when
#i = 3
```

```
PEEK_WORD(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_);
```

Lit le mot désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple :

```
%MW200 := PEEK_WORD(area:=16#84,
                    dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_);
```

Lit le double mot désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple :

```
%MD300 := PEEK_DWORD(area:=16#84,
                     dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_);
```

Lit la valeur booléenne désignée par bitOffset et byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée

Exemple :

```
%MB100.0 := PEEK_BOOL(area:=16#84,
                      dbNumber:=1, byteOffset:=#ii,
                      bitOffset:=#j);
```

```
POKE (area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_,
      value:=_in_);
```

Ecrit la valeur (octet, mot ou double mot) au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire référencée

Exemple avec un bloc de données :

```
POKE (area:=16#84, dbNumber:=2,
      byteOffset:=3, value:"Tag_1");
```

Exemple avec l'octet de sortie QB3 :

```
POKE (area:=16#82, dbNumber:=0,
      byteOffset:=3, value:"Tag_1");
```

Ecrit la valeur booléenne aux décalages bitOffset et byteOffset désignés du bloc de données ou de la zone d'E/S ou de mémoire référencée

Exemple :

```
POKE_BOOL (area:=16#84, dbNumber:=2,
            byteOffset:=3, bitOffset:=5, value:=0);
```

Ecrit un nombre ("count") d'octets en commençant au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire source référencée au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire de destination référencée

Exemple :

```
POKE_BLK (area_src:=16#84, dbNumber_src:=#src_db, byteOffset_src:=#src_byte, area_dest:=16#84, dbNumber_dest:=#src_db, byteOffset_dest:=#src_byte, count:=10);
```

```
POKE_BOOL (area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_,
            bitOffset:=_in_,
            value:=_in_);
```

```
POKE_BLK (area_src:=_in_,
           dbNumber_src:=_in_,
           byteOffset_src:=_in_,
           area_dest:=_in_,
           dbNumber_dest:=_in_,
           byteOffset_dest:=_in_,
           count:=_in_);
```

Pour les instructions PEEK et POKE, les valeurs suivantes sont possibles pour les paramètres "area", "area_src" et "area_dest". Pour les zones autres que les blocs de données, le paramètre dbNumber doit être égal à 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

8.6.8.2 Lire et écrire des instructions big et little Endian (SCL)

La CPU S7-1200 fournit des instructions SCL pour lire et écrire des données aux formats little endian et big endian. Le format little endian signifie que l'octet avec le bit le moins significatif se trouve dans l'adresse mémoire la plus basse. Le format big endian signifie que l'octet avec le bit le plus significatif se trouve dans l'adresse mémoire la plus haute.

Les quatre instructions SCL pour la lecture et l'écriture de données aux formats little endian et big endian sont les suivantes :

- READ_LITTLE (Lire des données au format little endian)
- WRITE_LITTLE (Ecrire des données au format little endian)
- READ_BIG (Lire des données au format big endian)
- WRITE_BIG (Ecrire des données au format big endian)

Tableau 8- 87 Lire et écrire des instructions big et little endian

CONT / LOG	SCL	Description
Non disponible	<code>READ_LITTLE (</code> <code> src_array:=_variant_in_</code> <code> dest_Variable =>_out_</code> <code> pos:= dint inout)</code>	Lit des données depuis une zone mémoire et les écrit dans une variable unique au format d'octet little endian.
Non disponible	<code>WRITE_LITTLE (</code> <code> src_variable:=_in_</code> <code> dest_array =>_variant_inout_</code> <code> pos:= dint inout)</code>	Ecrit des données depuis une variable unique vers une zone mémoire au format d'octet little endian.
Non disponible	<code>READ_BIG (</code> <code> src_array:=_variant_in_</code> <code> dest_Variable =>_out_</code> <code> pos:= dint inout)</code>	Lit des données depuis une zone mémoire et les écrit dans une variable unique au format d'octet big endian.
Non disponible	<code>WRITE_BIG (</code> <code> src_variable:=_in_</code> <code> dest_array =>_variant_inout_</code> <code> pos:= dint inout)</code>	Ecrit des données depuis une variable unique vers une zone mémoire au format d'octet big endian.

Tableau 8- 88 Paramètres pour les instructions READ_LITTLE and READ_BIG

Paramètre	Type de données	Description
src_array	Tableau d'octets	Zone mémoire depuis laquelle les données sont lues
dest_Variable	Chaînes de bits, entiers, nombres à virgule flottante, temporisations, date et heure, chaînes de caractères	Variable de destination dans laquelle les données sont écrites
pos	DINT	Position de base zéro depuis laquelle les données commencent à être lues depuis l'entrée src_array.

8.6 Transfert

Tableau 8- 89 Paramètres pour les instructions WRITE_LITTLE and WRITE_BIG

Paramètre	Type de données	Description
src_variable	Chaînes de bits, entiers, nombres à virgule flottante, LDT, TOD, LTOD, DATA, Char, WChar	Données source à partir de la variable
dest_array	Tableau d'octets	Zone mémoire dans laquelle les données sont écrites
pos	DINT	Position de base zéro dans laquelle les données commencent à être écrites dans la sortie dest_array.

Tableau 8- 90 Paramètre RET_VAL

RET_VAL* (W#16#...)	Description
0000	Pas d'erreur
80B4	Le SRC_ARRAY ou le DEST_ARRAY n'est pas un Array of Byte
8382	La valeur dans le paramètre POS se situe en dehors des limites du tableau.
8383	La valeur dans le paramètre POS se situe dans les limites du tableau mais la taille de la zone mémoire dépasse la limite supérieure du tableau.

*Vous pouvez visualiser les codes d'erreur sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.

8.6.9 Instructions Variant

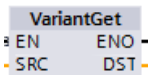
8.6.9.1 VariantGet (Lire la valeur de variable VARIANT)

Vous pouvez utiliser l'instruction "Lire la valeur de variable Variant" pour lire la valeur de la variable vers laquelle le pointeur Variant au niveau du paramètre SRC pointe et l'écrire dans la variable dans le paramètre DST.

Le paramètre SRC doit avoir le type de données Variant. N'importe quel type de données excepté Variant peut être spécifié pour le paramètre DST.

Le type de données de la variable pour le paramètre DST doit correspondre au type de données vers lequel la Variant pointe.

Tableau 8- 91 Instruction VariantGet

CONT / LOG	SCL	Description
	<pre>VariantGet(SRC:=_variant_in_, DST=>_variant_out_) ;</pre>	Lit la variable vers laquelle le paramètre SRC pointe et l'écrit dans la variable dans le paramètre DST

Remarque

Pour copier des structures et des tableaux, vous pouvez utiliser l'instruction "MOVE_BLK_VARIANT : Copier zone".

Tableau 8- 92 Paramètres pour l'instruction VariantGet

Paramètre	Type de données	Description
SRC	Variant	Pointeur désignant les données source
DST	Chaînes de bits, entiers, nombres à virgule flottante, temporisations, date et heure, chaînes de caractères, éléments du TABLEAU, types de données API	Destination dans laquelle les données sont écrites

Tableau 8- 93 État de ENO

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	L'instruction a copié les données de la variable vers laquelle SRC pointe dans la variable DST.
0	L'entrée de validation EN a l'état de signal "0" ou les types de données ne correspondent pas.	L'instruction n'a pas copié de données.

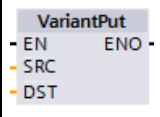
8.6.9.2 VariantPut (Écrire la valeur dans une variable VARIANT)

Vous pouvez utiliser l'instruction "Ecrire la valeur de variable VARIANT" pour écrire la valeur de la variable du paramètre SRC dans la variable du paramètre DST vers lequel la VARIANT pointe.

Le paramètre DST a le type de données VARIANT. N'importe quel type de données excepté VARIANT peut être spécifié pour le paramètre SRC.

Le type de données de la variable du paramètre SRC doit correspondre au type de données vers lequel la VARIANT pointe.

Tableau 8- 94 Instruction VariantPut

CONT / LOG	SCL	Description
	<pre>VariantPut (SRC:= _variant_in_, DST=> _variant_in_);</pre>	<p>Écrit la variable référencée par le paramètre SRC dans la variante vers laquelle le paramètre DST pointe</p>

Remarque

Pour copier des structures et des TABLEAUX, vous pouvez utiliser l'instruction "MOVE_BLK_VARIANT : Copier zone".

Tableau 8- 95 Paramètres pour l'instruction VariantPut

Paramètre	Type de données	Description
SRC	Bit strings, integers, floating-point numbers, timers, date and time, character strings, ARRAY elements, PLC data types	Pointeur désignant les données source
DST	Variant	Destination dans laquelle les données sont écrites

Tableau 8- 96 Etat de ENO

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	L'instruction a copié les données de la variable SRC dans la variable DST.
0	L'entrée de validation EN a l'état de signal "0" ou les types de données ne correspondent pas.	L'instruction n'a pas copié de données.

8.6.9.3 CountOfElements (Interroger le nombre d'éléments ARRAY)

Vous pouvez utiliser l'instruction "Obtenir le nombre d'éléments du TABLEAU" pour savoir combien d'éléments Array se trouvent dans une variable vers laquelle une Variant pointe.

S'il s'agit d'un TABLEAU unidimensionnel, l'instruction renvoie la différence entre la limite supérieure et inférieure +1. S'il s'agit d'un TABLEAU, l'instruction renvoie le produit de toutes les dimensions.

Tableau 8- 97 Instruction CountOfElements

CONT / LOG	SCL	Description
<p>CountOfElements</p> <ul style="list-style-type: none"> • EN • IN <p style="text-align: right;">ENO RET_VAL</p>	<pre>Result := CountOfElements(_variant_in_);</pre>	<p>Compte le nombre d'éléments du tableau dans le tableau vers lequel le paramètre IN pointe.</p>

Remarque

Si la Variant pointe vers un Array of Bool, l'instruction compte les éléments de remplissage à la limite octet la plus proche. Par exemple, l'instruction renvoie 8 comme nombre pour un Array[0..1] of Bool.

Tableau 8- 98 Paramètres pour l'instruction CountOfElements

Paramètre	Type de données	Description
IN	Variant	Variable avec des éléments du tableau à compter
RET_VAL	UDint	Résultat de l'instruction

Tableau 8- 99 Etat de ENO

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	L'instruction renvoie le nombre d'éléments du tableau.
0	L'entrée de validation EN a l'état de signal "0" ou la Variante ne pointe pas vers un tableau.	L'instruction renvoie 0.

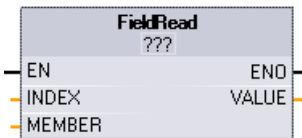

8.6.10 Instructions d'héritage

8.6.10.1 Instructions FieldRead (Lire champ) et FieldWrite (Ecrire champ)

Remarque

STEP 7 V10.5 **n'acceptait pas** de variable comme indice de tableau, ni les tableaux multidimensionnels. Les instructions FieldRead et FieldWrite rendaient possible l'utilisation d'un indice variable pour les tableaux à une dimension. STEP 7 V11 ou ultérieure **prend en charge** les variables comme indices de tableau et les tableaux multidimensionnels. Les instructions FieldRead et FieldWrite sont incluses dans STEP 7 V11 ou ultérieure pour assurer la compatibilité amont avec les programmes qui ont utilisé ces instructions.

Tableau 8- 100 Instructions FieldRead et FieldWrite

CONT/LOG	SCL	Description
	<pre>value := member[index];</pre>	<p>L'instruction FieldRead lit l'élément de valeur d'indice INDEX dans le tableau dont le premier élément est indiqué par le paramètre MEMBER. La valeur de l'élément de tableau est transférée à l'adresse indiquée par le paramètre VALUE.</p>
	<pre>member[index] := value;</pre>	<p>L'instruction FieldWrite transfère la valeur à l'adresse indiquée par le paramètre VALUE dans le tableau dont le premier élément est indiqué par le paramètre MEMBER. La valeur est transférée dans l'élément de tableau dont l'indice est indiqué par le paramètre INDEX.</p>

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 101 Types de données pour les paramètres

Paramètre et type		Type de données	Description
Index	Entrée	DInt	Numéro d'indice de l'élément de tableau à lire ou dans lequel écrire
Member ¹	Entrée	Nombres binaires, entiers, nombres à virgule flottante, temporisations, DATE, TOD, CHAR et WCHAR comme composants d'une variable de ARRAY	Emplacement du premier élément dans un tableau à une dimension défini dans un bloc de données global ou une interface de bloc. Par exemple : Si l'indice du tableau est donné sous la forme [-2..4], l'indice du premier élément est -2 et non 0.
Value ¹	Sortie	Nombres binaires, entiers, nombres à virgule flottante, temporisations, DATE, TOD, CHAR, WCHAR	Pour FieldRead, emplacement dans lequel l'élément de tableau indiqué est copié. Pour FieldWrite, emplacement de la valeur qui est copiée dans l'élément de tableau indiqué.

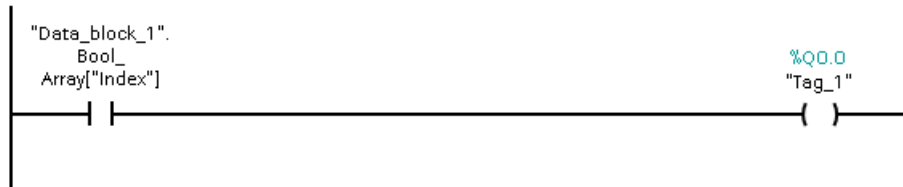
¹ Le type de données de l'élément de tableau indiqué par le paramètre MEMBER et celui du paramètre VALUE doivent être identiques.

La sortie de validation ENO est égale à 0 si l'une des conditions suivantes est vraie :

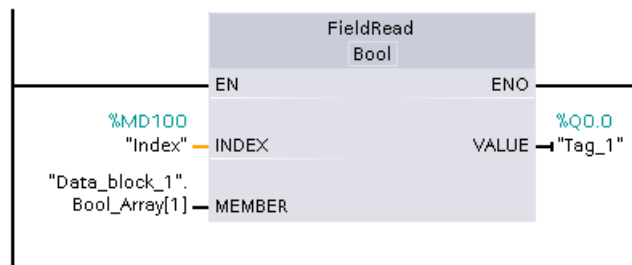
- L'entrée EN a l'état de signal "0".
- L'élément de tableau indiqué par le paramètre INDEX n'est pas défini dans le tableau désigné par le paramètre MEMBER.
- Des erreurs, telles qu'un débordement, se sont produites pendant le traitement.

Exemple : Accès à des données par indexation de tableau

Pour accéder aux éléments d'un tableau au moyen d'une variable, il suffit d'utiliser la variable comme indice de tableau dans la logique du programme. Le réseau ci-dessous, par exemple, définit une sortie en fonction de la valeur booléenne, désignée par la variable API "Index", d'un tableau de booléens dans "Data_block_1".



La logique avec l'indice de tableau variable est équivalente à l'ancienne méthode utilisant l'instruction FieldRead :



Les instructions FieldWrite et FieldRead peuvent être remplacées par la logique d'indexation de tableau par une variable.

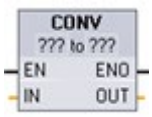
SCL ne dispose pas d'instructions FieldRead ou FieldWrite, mais prend en charge l'adressage indirect d'un tableau avec une variable :

```
#Tag_1 := "Data_block_1".Bool_Array[#Index];
```

8.7 Conversion

8.7.1 CONV (Convertir valeur)

Tableau 8- 102 Instruction Convertir (CONV)

CONT/LOG	SCL	Description
	<pre>out := <data type in>_TO_<data type out>(in);</pre>	Convertit un élément de données d'un type de données dans un autre type de données.

- 1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.
- 2 Pour SCL : Construisez l'instruction de conversion en identifiant le type de données pour le paramètre d'entrée et le paramètre de sortie. Par exemple, DWORD_TO_REAL convertit une valeur double mot en une valeur réelle.

Tableau 8- 103 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Chaîne de bits ¹ , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Valeur d'entrée
OUT	Chaîne de bits ¹ , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Valeur d'entrée convertie au nouveau type de données

- 1 L'instruction ne vous permet pas de sélectionner des chaînes de bits (octet, mot, double mot). Pour entrer un opérande de type Byte, Word ou DWord comme paramètre de l'instruction, sélectionnez un entier non signé ayant la même longueur en bits. Sélectionnez, par exemple, USInt pour un octet, UInt pour un mot ou UDInt pour un double mot.

Une fois que vous avez sélectionné le type de données (convertir de), une liste des conversions possibles s'affiche dans la liste déroulante (convertir en). Les conversions de et en BCD16 sont limitées au type de données Int. Les conversions de et en BCD32 sont limitées au type de données DInt.

Tableau 8- 104 Etat de ENO

ENO	Description	Résultat OUT
1	Pas d'erreur	Résultat valide
0	IN est +/- INF ou +/- NaN.	+/- INF ou +/- NaN
0	Le résultat dépasse la plage valide pour le type de données OUT.	OUT est mis à la valeur de IN .

8.7.2 Instructions de conversion pour SCL

Instructions de conversion pour SCL

Tableau 8- 105 Conversion à partir de Bool, Byte, Word ou DWord

Type de données	Instruction	Résultat
Bool	BOOL_TO_BYTE, BOOL_TO_WORD, BOOL_TO_DWORD, BOOL_TO_INT, BOOL_TO_DINT	La valeur est transférée dans le bit de poids faible du type de données cible.
Byte	BYTE_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	BYTE_TO_WORD, BYTE_TO_DWORD	La valeur est transférée dans l'octet de poids faible du type de données cible.
	BYTE_TO_SINT, BYTE_TO_USINT	La valeur est transférée dans le type de données cible.
	BYTE_TO_INT, BYTE_TO_UINT, BYTE_TO_DINT, BYTE_TO_UDINT	La valeur est transférée dans l'octet de poids faible du type de données cible.
Word	WORD_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	WORD_TO_BYTE	L'octet de poids faible de la valeur source est transféré dans le type de données cible.
	WORD_TO_DWORD	La valeur est transférée dans le mot de poids faible du type de données cible.
	WORD_TO_SINT, WORD_TO_USINT	L'octet de poids faible de la valeur source est transféré dans le type de données cible.
	WORD_TO_INT, WORD_TO_UINT	La valeur est transférée dans le type de données cible.
	WORD_TO_DINT, WORD_TO_UDINT	La valeur est transférée dans le mot de poids faible du type de données cible.
DWord	DWORD_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	DWORD_TO_BYTE, DWORD_TO_WORD, DWORD_TO_SINT	L'octet de poids faible de la valeur source est transféré dans le type de données cible.
	DWORD_TO_USINT, DWORD_TO_INT, DWORD_TO_UINT	Le mot de poids faible de la valeur source est transféré dans le type de données cible.
	DWORD_TO_DINT, DWORD_TO_UDINT, DWORD_TO_REAL	La valeur est transférée dans le type de données cible.

8.7 Conversion

Tableau 8- 106 Conversion à partir d'un entier court (SInt ou USInt)

Type de données	Instruction	Résultat
SInt	SINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	SINT_TO_BYTE	La valeur est transférée dans le type de données cible.
	SINT_TO_WORD, SINT_TO_DWORD	La valeur est transférée dans l'octet de poids faible du type de données cible.
	SINT_TO_INT, SINT_TO_DINT, SINT_TO_USINT, SINT_TO_UINT, SINT_TO_UDINT, SINT_TO_REAL, SINT_TO_LREAL, SINT_TO_CHAR, SINT_TO_STRING	La valeur est convertie.
USInt	USINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	USINT_TO_BYTE	La valeur est transférée dans le type de données cible.
	USINT_TO_WORD, USINT_TO_DWORD, USINT_TO_INT, USINT_TO_UINT, USINT_TO_DINT, USINT_TO_UDINT	La valeur est transférée dans l'octet de poids faible du type de données cible.
	USINT_TO_SINT, USINT_TO_REAL, USINT_TO_LREAL, USINT_TO_CHAR, USINT_TO_STRING	La valeur est convertie.

Tableau 8- 107 Conversion à partir d'un entier (Int ou UInt)

Type de données	Instruction	Résultat
Int	INT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	INT_TO_BYTE, INT_TO_DWORD, INT_TO_SINT, INT_TO_USINT, INT_TO_UINT, INT_TO_UDINT, INT_TO_REAL, INT_TO_LREAL, INT_TO_CHAR, INT_TO_STRING	La valeur est convertie.
	INT_TO_WORD	La valeur est transférée dans le type de données cible.
	INT_TO_DINT	La valeur est transférée dans l'octet de poids faible du type de données cible.
UInt	UINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	UINT_TO_BYTE, UINT_TO_SINT, UINT_TO_USINT, UINT_TO_INT, UINT_TO_REAL, UINT_TO_LREAL, UINT_TO_CHAR, UINT_TO_STRING	La valeur est convertie.
	UINT_TO_WORD, UINT_TO_DATE	La valeur est transférée dans le type de données cible.
	UINT_TO_DWORD, UINT_TO_DINT, UINT_TO_UDINT	La valeur est transférée dans l'octet de poids faible du type de données cible.

Tableau 8- 108 Conversion à partir d'un entier double (Dint ou UDIInt)

Type de données	Instruction	Résultat
DIInt	DINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	DINT_TO_BYTE, DINT_TO_WORD, DINT_TO_SINT, DINT_TO_USINT, DINT_TO_INT, DINT_TO_UINT, DINT_TO_UDINT, DINT_TO_REAL, DINT_TO_LREAL, DINT_TO_CHAR, DINT_TO_STRING	La valeur est convertie.
	DINT_TO_DWORD, DINT_TO_TIME	La valeur est transférée dans le type de données cible.
UDIInt	UDINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	UDINT_TO_BYTE, UDINT_TO_WORD, UDINT_TO_SINT, UDINT_TO_USINT, UDINT_TO_INT, UDINT_TO_UINT, UDINT_TO_DINT, UDINT_TO_REAL, UDINT_TO_LREAL, UDINT_TO_CHAR, UDINT_TO_STRING	La valeur est convertie.
	UDINT_TO_DWORD, UDINT_TO_TOD	La valeur est transférée dans le type de données cible.

Tableau 8- 109 Conversion à partir d'un réel (Real ou LReal)

Type de données	Instruction	Résultat
Real	REAL_TO_DWORD, REAL_TO_LREAL	La valeur est transférée dans le type de données cible.
	REAL_TO_SINT, REAL_TO_USINT, REAL_TO_INT, REAL_TO_UINT, REAL_TO_DINT, REAL_TO_UDINT, REAL_TO_STRING	La valeur est convertie.
LReal	LREAL_TO_SINT, LREAL_TO_USINT, LREAL_TO_INT, LREAL_TO_UINT, LREAL_TO_DINT, LREAL_TO_UDINT, LREAL_TO_REAL, LREAL_TO_STRING	La valeur est convertie.

Tableau 8- 110 Conversion à partir de Time, DTL, TOD ou Date



Type de données	Instruction	Résultat
Time	TIME_TO_DINT	La valeur est transférée dans le type de données cible.
DTL	DTL_TO_DATE, DTL_TO_TOD	La valeur est convertie.
TOD	TOD_TO_UDINT	La valeur est convertie.
Date	DATE_TO_UINT	La valeur est convertie.

Tableau 8- 111 Conversion à partir de Char ou String

Type de données	Instruction	Résultat
Char	CHAR_TO_SINT, CHAR_TO_USINT, CHAR_TO_INT, CHAR_TO_UINT, CHAR_TO_DINT, CHAR_TO_UDINT	La valeur est convertie.
	CHAR_TO_STRING	La valeur est transférée dans le premier caractère de la chaîne.
String	STRING_TO_SINT, STRING_TO_USINT, STRING_TO_INT, STRING_TO_UINT, STRING_TO_DINT, STRING_TO_UDINT, STRING_TO_REAL, STRING_TO_LREAL	La valeur est convertie.
	STRING_TO_CHAR	Le premier caractère de la chaîne est copié dans Char.

8.7.3 ROUND (Arrondir nombre) et TRUNC (Former un nombre entier)

Tableau 8- 112 Instructions ROUND et TRUNC

CONT/LOG	SCL	Description
	<pre>out := ROUND (in);</pre>	<p>Convertit un nombre réel en entier. Dans CONT/LOG, vous cliquez sur "???" dans la boîte d'instruction pour sélectionner le type de données pour la sortie, par exemple "DInt".</p> <p>Dans SCL, le type de données par défaut pour la sortie de l'instruction ROUND est DINT. Pour arrondir à un autre type de données de sortie, entrez le nom de l'instruction avec le nom explicite du type de données, par exemple ROUND_REAL ou ROUND_LREAL.</p> <p>La partie fractionnaire du nombre réel est arrondie à la valeur entière la plus proche (IEEE, arrondir au plus proche). Si le nombre se situe exactement à mi-chemin entre deux entiers (par exemple 10,5), le nombre est arrondi à l'entier pair. Par exemple :</p> <ul style="list-style-type: none"> • ROUND (10.5) = 10 • ROUND (11.5) = 12
	<pre>out := TRUNC (in);</pre>	<p>TRUNC convertit un nombre réel en entier. La partie fractionnaire du nombre réel est tronquée à zéro (IEEE, arrondir à zéro).</p>

¹ Pour CONT et LOG : Cliquez sur "???" (près du nom de l'instruction) et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 113 Types de données pour les paramètres



Paramètre	Type de données	Description
IN	Real, LReal	Entrée en virgule flottante
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Sortie arrondie ou tronquée

Tableau 8- 114 Etat de ENO

ENO	Description	Résultat OUT
1	Pas d'erreur	Résultat valide
0	IN est +/- INF ou +/- NaN.	+/- INF ou +/- NaN

8.7.4 CEIL (Arrondir à l'entier supérieur) et FLOOR (Arrondir à l'entier inférieur)

Tableau 8- 115 Instructions CEIL et FLOOR

CONT/LOG	SCL	Description
	<code>out := CEIL(in);</code>	Convertit un nombre réel (Real ou LReal) au nombre entier le plus proche supérieur ou égal au nombre réel sélectionné ("arrondir à +infini" IEEE).
	<code>out := FLOOR(in);</code>	Convertit un nombre réel (Real ou LReal) au nombre entier le plus proche inférieur ou égal au nombre réel sélectionné ("arrondir à -infini" IEEE).

¹ Pour CONT et LOG : Cliquez sur "???" (près du nom de l'instruction) et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 116 Types de données pour les paramètres

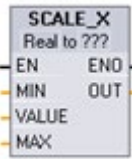
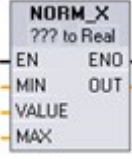
Paramètre	Type de données	Description
IN	Real, LReal	Entrée en virgule flottante
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Sortie convertie

Tableau 8- 117 Etat de ENO

ENO	Description	Résultat OUT
1	Pas d'erreur	Résultat valide
0	IN est +/- INF ou +/- NaN.	+/- INF ou +/- NaN

8.7.5 SCALE_X (Mise à l'échelle) et NORM_X (Normaliser)

Tableau 8- 118 Instructions SCALE_X et NORM_X

CONT/LOG	SCL	Description
	<pre>out :=SCALE_X(min:=_in_, value:=_in_, max:=_in_);</pre>	<p>Met à l'échelle le paramètre réel normalisé VALUE, avec $0.0 \leq VALUE \leq 1.0$, dans le type de données et la plage de valeurs indiqués par les paramètres MIN et MAX :</p> $OUT = VALUE (MAX - MIN) + MIN$
	<pre>out :=NORM_X(min:=_in_, value:=_in_, max:=_in_);</pre>	<p>Normalise le paramètre VALUE à l'intérieur de la plage de valeurs indiquée par les paramètres MIN et MAX :</p> $OUT = (VALUE - MIN) / (MAX - MIN),$ <p>avec ($0.0 \leq OUT \leq 1.0$)</p>

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 119 Types de données pour les paramètres

Paramètre	Type de données ¹	Description
MIN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Valeur d'entrée pour le minimum de la plage
VALUE	SCALE_X : Real, LReal NORM_X : SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Valeur d'entrée à mettre à l'échelle ou à normaliser
MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Valeur d'entrée pour le maximum de la plage
OUT	SCALE_X : SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal NORM_X : Real, LReal	Valeur de sortie mise à l'échelle ou normalisée

¹ Pour SCALE_X : Les paramètres MIN, MAX et OUT doivent avoir le même type de données.
Pour NORM_X : Les paramètres MIN, VALUE et MAX doivent avoir le même type de données.

Remarque

Le paramètre VALUE de SCALE_X doit être limité à $0.0 \leq \text{VALUE} \leq 1.0$

Si le paramètre VALUE est inférieur à 0.0 ou supérieur à 1.0 :

- La mise à l'échelle linéaire peut produire des valeurs OUT qui sont inférieures à la valeur du paramètre MIN ou supérieures à la valeur du paramètre MAX pour des valeurs OUT qui se situent dans la plage de valeurs du type de données OUT. Dans ces cas, l'exécution de SCALE_X met ENO à VRAI.
- Il est possible de générer des nombres mis à l'échelle qui ne sont pas dans la plage du type de données OUT. Dans ces cas, la valeur du paramètre OUT est définie à une valeur intermédiaire égale à la partie la moins significative du nombre réel mis à l'échelle avant la conversion finale au type de données OUT. Dans ce cas, l'exécution de SCALE_X met ENO à FAUX.

Le paramètre VALUE de NORM_X doit être limité à $\text{MIN} \leq \text{VALUE} \leq \text{MAX}$

Si le paramètre VALUE est inférieur à MIN ou supérieur à MAX, la mise à l'échelle linéaire peut produire des valeurs OUT normalisées qui sont inférieures à 0.0 ou supérieures à 1.0. Dans ce cas, l'exécution de NORM_X met ENO à VRAI.

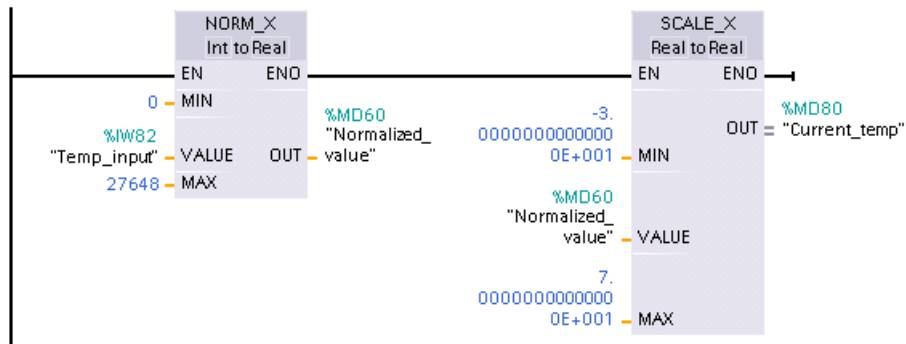
Tableau 8- 120 Etat de ENO

ENO	Situation d'erreur	Résultat OUT
1	Pas d'erreur	Résultat valide
0	Le résultat dépasse la plage valide pour le type de données OUT.	Résultat intermédiaire : partie la moins significative d'un nombre réel avant la conversion finale au type de données OUT.
0	Paramètre MAX \leq MIN	SCALE_X : La partie la moins significative du nombre réel VALUE complète la taille de OUT. NORM_X : VALUE dans le type de données VALUE étendu pour remplir une taille de double mot.
0	Paramètre VALUE = +/- INF ou +/- NaN	VALUE est écrit dans OUT.

Exemple (CONT) : Normalisation et mise à l'échelle d'une valeur d'entrée analogique

Une entrée analogique d'un module d'entrées-sorties ou Signal Board analogique avec entrée en courant se situe dans la plage 0 à 27648 des valeurs valides. Supposons qu'une entrée analogique représente une température, avec la valeur 0 de l'entrée analogique correspondant à -30,0 °C et la valeur 27648 correspondant à 70,0 °C.

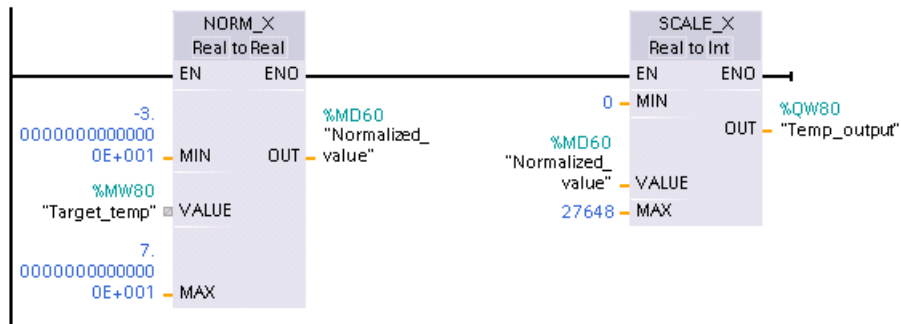
Pour transformer la valeur analogique en unités physiques correspondantes, normalisez l'entrée à une valeur comprise entre 0,0 et 1,0, puis mettez-la à l'échelle entre -30,0 et 70,0. La valeur résultante est la température représentée par l'entrée analogique en degrés Celsius :



Notez que si l'entrée analogique était celle d'un module d'entrées-sorties ou Signal Board analogique en tension, la valeur MIN pour l'instruction NORM_X serait -27648 et non 0.

Exemple (CONT) : Normalisation et mise à l'échelle d'une valeur de sortie analogique

Une sortie analogique à définir dans un module d'entrées-sorties ou Signal Board analogique avec sortie en courant doit se situer dans la plage 0 à 27648 des valeurs valides. Supposons qu'une sortie analogique représente un réglage de température, avec la valeur 0 de la sortie analogique correspondant à -30,0 °C et la valeur 27648 correspondant à 70,0 °C. Pour convertir une valeur de température comprise entre -30,0 et 70,0 en mémoire à une valeur pour la sortie analogique qui se situe dans la plage 0 à 27648, vous devez normaliser la valeur en unités physiques à une valeur comprise entre 0,0 et 1,0, puis la mettre à l'échelle à la plage de la sortie analogique, c'est-à-dire 0 à 27648 :



Notez que si la sortie analogique était destinée à un module d'entrées-sorties ou Signal Board analogique en tension, la valeur MIN pour l'instruction SCALE_X serait -27648 et non 0.

Vous trouverez des informations supplémentaires sur les représentations des entrées analogiques (Page 1554) et des sorties analogiques (Page 1556) en tension et en courant dans les Caractéristiques techniques.

8.7.6 Instructions de conversion de Variante

8.7.6.1 VARIANT_TO_DB_ANY (Convertir VARIANT en DB_ANY)

L'instruction "VARIANT to DB_ANY" lit l'opérande dans le paramètre IN et le convertit en type de données DB_ANY. Le paramètre IN est du type de données Variant et représente soit un bloc de données d'instance, soit un bloc de données ARRAY. Lorsque vous créez le programme, vous n'avez pas besoin de savoir quel bloc de données correspond au paramètre IN. L'instruction lit le numéro du bloc de données pendant l'exécution et l'écrit dans l'opérande du paramètre RET_VAL .

Tableau 8- 121 Instruction VARIANT_TO_DB_ANY

CONT / LOG	SCL	Description
Non disponible	<pre>RET_VAL := VARIANT_TO_DB_ANY(in := _variant_in_, err => _int_out_);</pre>	Lit l'opérande dans le paramètre IN de type Variant et le sauvegarde dans le résultat de la fonction, qui est de type DB_ANY.

Tableau 8- 122 Paramètres pour l'instruction VARIANT_TO_DB_ANY

Paramètre	Type de données	Description
IN	Variant	Variante qui représente un bloc de données d'instance ou un bloc de données de tableau
RET_VAL	DB_ANY	Type de données DB_ANY de sortie qui contient le numéro du bloc de données converti
ERR	Int	Information d'erreur

Tableau 8- 123 Etat de ENO

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	L'instruction convertit l'entrée de type Variant et la sauvegarde dans la sortie de type DB_ANY de la fonction.
0	L'entrée de validation EN a l'état de signal "0" ou le paramètre IN est invalide.	L'instruction ne fait rien.

Tableau 8- 124 Codes de sortie d'erreur pour l'instruction VARIANT_TO_DB_ANY

Err (W#16#...)	Description
0000	Pas d'erreur
252C	Le type de données Variant dans le paramètre IN a la valeur 0. La CPU passe à l'état Arrêt (STOP).
8131	Le bloc de données n'existe pas ou est trop court (premier accès).
8132	Le bloc de données est trop court et n'est pas un bloc de données Array (deuxième accès).
8134	Le bloc de données est protégé en écriture
8150	Le type de données Variant dans le paramètre IN indique la valeur "0". Pour recevoir ce message d'erreur, la propriété du bloc "Gérer les erreurs dans le bloc" doit être activée. Sinon, la CPU passe à l'état STOP et envoie le code d'erreur 16#252C
8154	Le bloc de données a un type de données incorrect.
*Vous pouvez afficher des codes d'erreur sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.	

8.7.6.2 DB_ANY_TO_VARIANT (Convertir DB_ANY en VARIANT)

L'instruction "DB_ANY to VARIANT" permet de lire le numéro d'un bloc de données qui satisfait aux exigences indiquées ci-dessous. L'opérande dans le paramètre IN a le type de données DB_ANY, ce qui signifie que vous n'avez pas besoin de savoir pendant la création du programme quel bloc de données doit être lu. L'instruction lit le numéro du bloc de données lors de l'exécution et l'écrit dans le résultat de la fonction RET_VAL au moyen d'un pointeur VARIANT.

Tableau 8- 125 Instruction DB_ANY_TO_VARIANT

CONT / LOG	SCL	Description
Non disponible	<pre>RET_VAL := DB_ANY_TO_VARIANT(in := _db_any_in_, err => _int_out_);</pre>	Lit le numéro du bloc de données dans le paramètre Variant IN et le sauvegarde dans le résultat de fonction, qui est du type Variant

Tableau 8- 126 Paramètres pour l'instruction DB_ANY_TO_VARIANT

Paramètre	Type de données	Description
IN	DB_ANY	Variant qui contient le numéro du bloc de données
RET_VAL	Variant	Type de données DB_ANY de sortie qui contient le numéro du bloc de données converti
ERR	Int	Information d'erreur

Tableau 8- 127 Etat de ENO

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	L'instruction convertit le numéro du bloc de données dans la variante et le sauvegarde dans la sortie de fonction DB_ANY
0	L'entrée de validation EN a l'état de signal "0" ou le paramètre IN est invalide.	L'instruction ne fait rien.

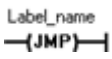



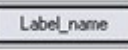
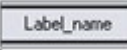
Tableau 8- 128 Codes de sortie d'erreur pour l'instruction DB_ANY_TO_VARIANT

Err (W#16#...)	Description
0000	Pas d'erreur
8130	Le numéro du bloc de données est 0.
8131	Le bloc de données n'existe pas ou est trop court.
8132	Le bloc de données est trop court et n'est pas un bloc de données du Tableau.
8134	Le bloc de données est protégé en écriture.
8154	Le bloc de données a un type de données incorrect.
8155	Code de type inconnu
*Vous pouvez afficher des codes d'erreur sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.	

8.8 Gestion du programme

8.8.1 Instructions JMP (Saut si RLO = 1), JMPN (Saut si RLO = 0), et Label (Repère de saut)

Tableau 8- 129 Instructions JMP, JMPN et LABEL

CONT	LOG	SCL	Description
		Voir l'instruction GOTO (Page 338).	Saut si RLO (Résultat logique) = 1 : S'il y a flux de courant vers la bobine JMP (CONT) ou si l'entrée de la boîte JMP est vraie (LOG), l'exécution du programme continue avec la première instruction suivant le repère de saut indiqué.
			Saut si RLO = 0 : S'il n'y a pas de flux de courant vers la bobine JMPN (CONT) ou si l'entrée de la boîte JMPN est fausse (LOG), l'exécution du programme continue avec la première instruction suivant le repère de saut indiqué.
			Destination pour une instruction de saut JMP ou JMPN.

¹ Vous créez vos noms de repère en tapant directement dans l'instruction LABEL. Utilisez l'icône d'aide des paramètres pour sélectionner les noms de repère disponibles pour le champ de nom de repère de JMP et JMPN. Vous pouvez également saisir un nom de repère directement dans l'instruction JMP ou JMPN.

Tableau 8- 130 Types de données pour les paramètres

Paramètre	Type de données	Description
Label_name	Identificateur de repère	Identificateur pour les instructions de saut et repère de destination de saut correspondant

- Chaque repère doit être univoque à l'intérieur d'un bloc de code.
- Vous pouvez sauter à l'intérieur d'un bloc de code, mais vous ne pouvez pas sauter d'un bloc de code à un autre bloc de code.
- Vous pouvez sauter en avant ou en arrière.
- Vous pouvez sauter au même repère à partir de plus d'un point dans le même bloc de code.

8.8.2 JMP_LIST (Définir liste de sauts)

Tableau 8- 131 Instruction JMP_LIST


CONT/LOG	SCL	Description
	<pre> CASE k OF 0: GOTO dest0; 1: GOTO dest1; 2: GOTO dest2; [n: GOTO destn;] END_CASE; </pre>	<p>L'instruction JMP_LIST agit comme un répartiteur de sauts pour piloter l'exécution de sections de programme. Un saut est exécuté au repère de programme correspondant à la valeur de l'entrée K. L'exécution du programme se poursuit avec les instructions qui suivent le repère de saut de destination. Si la valeur de l'entrée K dépasse le nombre de repères - 1, aucun saut n'est effectué et le traitement se poursuit avec le réseau suivant du programme.</p>

Tableau 8- 132 Types de données pour les paramètres

Paramètre	Type de données	Description
K	UInt	Valeur de commande du répartiteur de sauts
DEST0, DEST1, ..., DESTn.	Repères de programme	Repères de destination de saut correspondant à des valeurs spécifiques du paramètre K. Si la valeur de K est égale à 0, un saut est effectué au repère de programme affecté à la sortie DEST0. Si la valeur de K est égale à 1, un saut est effectué au repère de programme affecté à la sortie DEST1 et ainsi de suite. Si la valeur de l'entrée K dépasse le nombre de repères - 1, aucun saut n'est effectué et le traitement se poursuit avec le réseau suivant du programme.

Pour CONT et LOG : Lorsque vous insérez initialement la boîte JMP_LIST dans votre programme, elle ne comporte que deux sorties de repère de saut. Vous pouvez ajouter ou supprimer des destinations de saut.



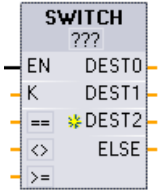
Cliquez sur l'icône de création à l'intérieur de la boîte (à gauche du dernier paramètre DEST) pour ajouter de nouvelles sorties de repères de saut.



- Cliquez avec le bouton droit de la souris sur une ligne de sortie et sélectionnez la commande "Insérer sortie".
- Cliquez avec le bouton droit de la souris sur une ligne de sortie et sélectionnez la commande "Supprimer".

8.8.3 SWITCH (Branchement conditionnel)

Tableau 8- 133 Instruction SWITCH

CONT/LOG	SCL	Description
	Non disponible	L'instruction SWITCH agit comme un répartiteur de sauts pour piloter l'exécution de sections de programme. La valeur de l'entrée K est comparée aux valeurs affectées aux entrées de comparaison indiquées et un saut est effectué au repère de programme correspondant au premier test de comparaison qui est vrai. Si aucune des comparaisons n'est vraie, il y a saut au repère affecté à ELSE. L'exécution du programme se poursuit avec les instructions qui suivent le repère de saut de destination.

- 1 Pour CONT et LOG : Cliquez sous le nom de la boîte et sélectionnez un type de données dans le menu déroulant.
- 2 Pour SCL : Utilisez un jeu de comparaisons IF-THEN.

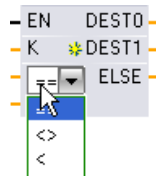
Tableau 8- 134 Types de données pour les paramètres

Paramètre	Type de données ¹	Description
K	UInt	Entrée de valeur de comparaison commune
==, <>, <, <=, >, >=	SInt, Int, DInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Entrées de valeurs de comparaison distinctes pour des types de comparaison spécifiques
DEST0, DEST1, ..., DESTn, ELSE	Repères de programme	Repères de destination de saut correspondant à des comparaisons spécifiques : L'entrée de comparaison la plus proche en dessous de l'entrée K est traitée en premier et provoque un saut au repère affecté à DEST0 si la comparaison entre la valeur K et cette entrée est vraie. Le test de comparaison suivant utilise l'entrée du dessous suivante et provoque un saut au repère affecté à DEST1 si la comparaison est vraie, et ainsi de suite avec les comparaisons restantes. Si aucune des comparaisons n'est vraie, il y a saut au repère affecté à la sortie ELSE.

¹ L'entrée K et les entrées de comparaison (==, <>, <, <=, >, >=) doivent avoir le même type de données.

Ajout et suppression d'entrées et indication des types de comparaison

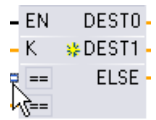
Lorsque la boîte SWITCH CONT ou LOG est initialement insérée dans votre programme, elle comporte deux entrées de comparaison. Vous pouvez affecter des types de comparaison et ajouter des entrées/destinations de saut comme illustré ci-dessous.



Cliquez sur un opérateur de comparaison dans la boîte et sélectionnez un nouvel opérateur dans la liste déroulante.



Cliquez sur l'icône de création à l'intérieur de la boîte (à gauche du dernier paramètre DEST) pour ajouter de nouveaux paramètres de destination pour la comparaison.



- Cliquez avec le bouton droit de la souris sur une ligne d'entrée et sélectionnez la commande "Insérer entrée".
- Cliquez avec le bouton droit de la souris sur une ligne d'entrée et sélectionnez la commande "Supprimer".

Tableau 8- 135 Sélection de type de données et opérations de comparaison autorisées pour la boîte SWITCH

Type de données	Comparaison	Syntaxe de l'opérateur
Byte, Word, DWord	Egal	==
	Différent	<>
SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, TOD, Date	Egal	==
	Différent	<>
	Supérieur ou égal	>=
	Inférieur ou égal	<=
	Supérieur à	>
	Inférieur à	<

Règles de placement de la boîte SWITCH

- Aucune connexion d'instruction CONT/LOG devant l'entrée de comparaison n'est autorisée.
- Il n'existe pas de sortie ENO. On ne peut donc avoir qu'une instruction SWITCH dans un réseau et l'instruction SWITCH doit être la dernière opération du réseau.

8.8.4 RET (Retour de saut)

L'instruction RET optionnelle sert à mettre fin à l'exécution du bloc en cours. Si et seulement si il y a flux de courant vers la bobine RET (CONT) ou si l'entrée de la boîte RET est vraie (CONT), l'exécution du programme du bloc en cours est interrompue en ce point et les instructions au-delà de l'instruction RET ne sont pas exécutées. Si le bloc en cours est un OB, le paramètre "Return_Value" n'est pas pris en compte. Si le bloc en cours est une FC ou un FB, la valeur du paramètre "Return_Value" est transmise au programme appelant en tant que valeur ENO de la boîte appelée.

Vous n'avez pas besoin d'utiliser une instruction RET comme dernière instruction dans un bloc ; cela est fait automatiquement pour vous. Vous pouvez avoir plusieurs instructions RET à l'intérieur d'un seul bloc.

Pour SCL, reportez-vous à l'instruction RETURN (Page 338).

Tableau 8- 136 Instruction de gestion du programme RET

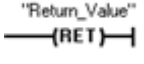

CONT	LOG	SCL	Description
		RETURN ;	Met fin à l'exécution du bloc en cours.

Tableau 8- 137 Types de données pour les paramètres

Paramètre	Type de données	Description
Return_Value	Bool	Le paramètre "Return_value" de l'instruction RET est affecté à la sortie ENO de la boîte d'appel de bloc dans le bloc appelant.

Modèle d'étapes pour l'utilisation de l'instruction RET dans une FC :

1. Créez un nouveau projet et insérez-y une FC.
2. Editez la FC :
 - Ajoutez des instructions depuis l'arborescence d'instructions.
 - Ajoutez une instruction RET en indiquant l'une des valeurs suivantes pour le paramètre "Return_Value" :
VRAI, FAUX ou une adresse de mémoire contenant la valeur de retour requise.
 - Ajoutez d'autres instructions.
3. Appelez la FC depuis le programme principal [OB1].

L'entrée EN de la boîte FC dans l'OB 1 doit être vraie pour que l'exécution de la FC commence.

La valeur indiquée par l'instruction RET dans la FC sera présente au niveau de la sortie ENO de la boîte FC dans le bloc OB 1 après l'exécution de la FC pour laquelle le flux de courant vers l'instruction RET est vrai.

8.8.5 ENDIS_PW (Limiter et valider la légitimation par mot de passe)

Tableau 8- 138 Instruction ENDIS_PW

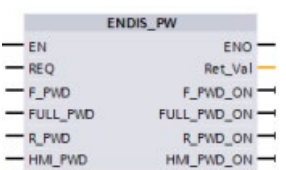
CONT/LOG	SCL	Description
	<pre> ENDIS_PW (req:=_bool_in_, f_pwd:=_bool_in_, full_pwd:=_bool_in_, r_pwd:=_bool_in_, hmi_pwd:=_bool_in_, f_pwd_on=>_bool_out_, full_pwd_on=>_bool_out_, r_pwd_on=>_bool_out_, hmi_pwd_on=>_bool_out_); </pre>	<p>L'instruction ENDIS_PW permet d'autoriser ou de rejeter les connexions d'un client à une CPU S7-1200, et ce même lorsque le client peut fournir le mot de passe correct.</p> <p>Cette instruction ne rejette pas les mots de passe du serveur Web.</p>

Tableau 8- 139 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Exécuter la fonction si REQ=1
F_PWD	IN	Bool	Mot de passe de sécurité : Autoriser (=1) ou rejeter (=0)
FULL_PWD	IN	Bool	Mot de passe pour l'accès intégral : Autoriser (=1) ou rejeter (=0) le mot de passe pour l'accès illimité
R_PWD	IN	Bool	Mot de passe pour l'accès en lecture : Autoriser (=1) ou rejeter (=0)
HMI_PWD	IN	Bool	Mot de passe IHM : Autoriser (=1) ou rejeter (=0)
F_PWD_ON	OUT	Bool	Etat du mot de passe de sécurité : Autoriser (=1) ou rejeter (=0)
FULL_PWD_ON	OUT	Bool	État du mot de passe pour l'accès intégral : Autoriser (=1) ou rejeter (=0)
R_PWD_ON	OUT	Bool	Etat du mot de passe Lecture seule : Autoriser (=1) ou rejeter (=0)
HMI_PWD_ON	OUT	Bool	Etat du mot de passe IHM : Autoriser (=1) ou rejeter (=0)
Ret_Val	OUT	Word	Résultat de la fonction

Appeler ENDIS_PW avec REQ=1 provoque le rejet des types de mots de passe dont la valeur du paramètre d'entrée est FALSE. Chaque type de mot de passe peut être autorisé ou rejeté de manière indépendante. Par exemple, si le mot de passe de sécurité est autorisé et que tous les autres mots de passe sont rejetés, vous pouvez limiter l'accès à la CPU à un petit groupe d'employés.

ENDIS_PW est exécuté de manière synchrone dans un cycle de programme et les paramètres de sortie des mots de passe montrent toujours l'état d'autorisation en cours du mot de passe indépendamment du paramètre d'entrée REQ. Tous les mots de passe que vous autorisez doivent pouvoir être changés en rejeté/autorisé. Faute de quoi une erreur est renvoyée et tous les mots de passe qui étaient autorisés avant l'exécution de ENDIS_PW sont autorisés. Cela signifie que, dans une CPU standard (où le mot de passe de sécurité n'est pas configuré) F_PWD doit toujours être mis à 1 pour que la valeur en retour soit égale à 0. Dans ce cas, F_PWD_ON est toujours à 1.

Remarque

- L'exécution de ENDIS_PW peut bloquer l'accès des appareils IHM si le mot de passe IHM est rejeté.
- Les sessions clients qui étaient autorisées avant l'exécution de ENDIS_PW ne sont pas affectées par l'exécution de ENDIS_PW.

Après la mise sous tension, l'accès à la CPU est restreint par des mots de passe préalablement définis dans la configuration de protection normale de la CPU. La possibilité de rejeter un mot de passe valide doit être remise en place en exécutant une nouvelle fois ENDIS_PW. Toutefois, si l'instruction ENDIS_PW est immédiatement exécutée et que des mots de passe nécessaires sont rejetés, l'accès à TIA Portal peut être verrouillé. Vous pouvez utiliser une instruction de temporisation pour repousser l'exécution d'ENDIS_PW et laisser le temps de saisir les mots de passe, avant que ceux-ci ne soient rejetés.

Remarque

Restauration d'une CPU qui verrouille la communication avec TIA Portal

Consultez la rubrique "Restauration du fait d'un mot de passe perdu (Page 157)" pour obtenir des détails sur la manière de supprimer la mémoire de chargement interne d'un API en utilisant une carte mémoire.

Un changement d'état de fonctionnement à ARRET dû à des erreurs, à l'exécution de STP ou à STEP 7 ne supprime pas la protection. La protection s'applique jusqu'à la mise hors tension puis sous tension de la CPU. Pour obtenir des détails, référez-vous au tableau suivant.

Action	Mode de fonctionnement	Gestion de mot de passe par ENDIS_PW
Après un effacement général de STEP 7	ARRET	Actif : les mots de passe rejetés le restent.
Après mise sous tension ou changement de carte mémoire	STOP	Off : aucun mot de passe n'est rejeté.
Après l'exécution d'ENDIS_PW dans un OB de cycle de programme ou de démarrage	MISE EN ROUTE, MARCHÉ	Actif : certains mots de passe sont rejetés suivant les paramètres d'ENDIS_PW.
Après un changement de l'état de fonctionnement de MARCHÉ ou MISE EN ROUTE à ARRET par l'instruction STP, une erreur ou STEP 7	ARRET	Actif : les mots de passe rejetés le restent.

Remarque

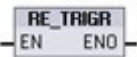
Protégez les niveaux d'accès à la CPU par des mots de passe complexes. Les mots de passe forts contiennent au moins dix caractères, lettres, chiffres et caractères spéciaux mélangés, ne sont pas des mots du dictionnaire et ne sont pas des noms ou identifiants pouvant être déduits de vos données personnelles. Ne divulguez pas le mot de passe et changez-en fréquemment.

Tableau 8- 140 Codes d'erreur

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	L'instruction n'est pas supportée.
80D0	Le mot de passe de sécurité n'est pas configuré.
80D1	Le mot de passe pour l'accès en lecture/écriture n'est pas configuré.
80D2	Le mot de passe pour l'accès en lecture n'est pas configuré.
80D3	Le mot de passe pour l'accès IHM n'est pas configuré.

8.8.6 RE_TRIGR (Redéclencher le temps de surveillance du cycle)

Tableau 8- 141 RE_TRIGR, instruction

CONT/LOG	SCL	Description
	RE_TRIGR () ;	RE_TRIGR (Redémarrer surveillance du temps de cycle) sert à allonger le temps maximum autorisé avant que la temporisation de surveillance du cycle ne génère une erreur.

Utilisez l'instruction RE_TRIGR pour redémarrer la temporisation de surveillance du cycle pendant un cycle unique. Cela a pour effet d'allonger le temps de cycle maximum autorisé d'une période de temps de cycle maximum, à partir de la dernière exécution de la fonction RE_TRIGR.

Remarque

Avant la version de firmware 2.2 des CPU S7-1200, RE_TRIGR ne pouvait être exécuté qu'à partir d'un OB de cycle de programme et pouvait être utilisé pour allonger le temps de cycle API indéfiniment. ENO = FALSE et la temporisation de surveillance du cycle n'est pas réinitialisée lorsque RE_TRIGR était exécuté à partir d'un OB de démarrage, d'un OB d'alarme ou d'un OB d'erreur.

A partir de la version de firmware 2.2, RE-TRIGR peut être exécuté dans n'importe quel OB (OB de démarrage, d'alarme et d'erreur compris). Toutefois, le cycle API ne peut être allongé que de 10 fois le temps de cycle maximum configuré au plus.

Définition du temps de cycle maximum de l'automate

Configurez la valeur du temps de cycle maximum sous "Temps de cycle" dans la configuration des appareils.

Tableau 8- 142 Valeurs du temps de cycle

Surveillance du temps de cycle	Valeur minimale	Valeur maximale	Valeur par défaut
Temps de cycle maximum	1 ms	6000 ms	150 ms

Expiration de la temporisation de surveillance du cycle

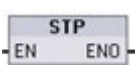
Une erreur est générée si la temporisation de cycle maximum expire avant que le cycle ne soit achevé. Si le programme utilisateur contient un OB d'erreur de temps (OB 80), la CPU exécute l'OB d'erreur de temps dans lequel vous pouvez insérer de la logique de programme pour créer une réaction spécifique.

Si le programme utilisateur ne contient pas d'OB d'erreur de temps, la première expiration n'est pas prise en compte et la CPU reste à l'état MARCHE. Si une deuxième expiration du temps de cycle maximum se produit pendant le même cycle du programme (2 fois la valeur du temps de cycle maximum), une erreur provoquant le passage de la CPU à l'état ARRET est générée.

A l'état ARRET, l'exécution de votre programme s'arrête alors que la communication système de la CPU et le diagnostic système se poursuivent.

8.8.7 STP (Arrêter le programme)

Tableau 8- 143 Instruction STP

CONT/LOG	SCL	Description
	<pre>STP () ;</pre>	STP met la CPU à l'état ARRET. Lorsque la CPU est à l'état ARRET, l'exécution de votre programme et les actualisations physiques depuis la mémoire image s'arrêtent.

Pour plus d'informations, reportez-vous à Configuration des sorties lors d'un passage de MARCHE à ARRET (Page 120).

Si EN = VRAI, la CPU passe à l'état ARRET, l'exécution du programme s'arrête et l'état de ENO est sans objet. Sinon, EN = ENO = 0.

8.8.8 Instructions GET_ERROR et GET_ERROR_ID (Interrogation locale des erreurs et des ID d'erreur)

Les instructions de traitement d'erreur fournissent des informations sur les erreurs d'exécution des blocs de programme. Si vous ajoutez une instruction GET_ERROR ou GET_ERROR_ID à votre bloc de code, vous pouvez gérer les erreurs de programme à l'intérieur de votre bloc de programme.

GET_ERROR

Tableau 8- 144 Instruction GET_ERROR

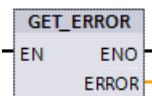
CONT/LOG	SCL	Description
	<pre>GET_ERROR (_out_) ;</pre>	Signale qu'une erreur d'exécution du bloc de programme s'est produite et remplit une structure de données d'erreur prédéfinie avec des informations d'erreur détaillées.

Tableau 8- 145 Types de données pour les paramètres

Paramètre	Type de données	Description
ERROR	ErrorStruct	Structure de données d'erreur : Vous pouvez renommer la structure mais pas les éléments à l'intérieur de la structure.

Tableau 8- 146 Eléments de la structure de données ErrorStruct

Eléments de la structure		Type de données	Description					
ERROR_ID		Word	Code d'erreur					
FLAGS		Byte	Indique si une erreur s'est produite pendant un appel de bloc. <ul style="list-style-type: none"> 16#01: Erreur pendant un appel de bloc 16#00: Aucune erreur pendant un appel de bloc 					
REACTION		Byte	Réaction par défaut : <ul style="list-style-type: none"> 0: Ne pas tenir compte de l'erreur (erreur d'écriture) 1: Poursuivre avec la valeur de remplacement "0" (erreur de lecture) 2: Sauter l'instruction (erreur système) 					
CODE_ADDRESS		CREF	Information sur l'adresse et le type de bloc					
	BLOCK_TYPE	Byte	Type de bloc où l'erreur s'est produite : <ul style="list-style-type: none"> 1: OB 2: FC 3: FB 					
	CB_NUMBER	UInt	Numéro du bloc de code					
	OFFSET	UDInt	Référence à la mémoire interne					
MODE		Byte	Mode d'accès : Les informations suivantes peuvent être transmises selon le type d'accès :					
			Mode	(A)	(B)	(C)	(D)	(E)
			0					
			1					Déca- lage
			2			Zone		
			3	Adresse	Portée		Numéro	
			4			Zone		Déca- lage
			5			Zone	N° DB	Déca- lage
			6	N° Ptr /Acc		Zone	N° DB	Déca- lage
7	N° Ptr /Acc	N° empl./Portée	Zone	N° DB	Déca- lage			
OPERAND_NUMBER		UInt	Numéro d'opérande de la commande machine					
POINTER_NUMBER_LOCATION		UInt	(A) Pointeur interne					
SLOT_NUMBER_SCOPE		UInt	(B) Zone de stockage en mémoire interne					
DATA_ADDRESS		NREF	Information sur l'adresse d'un opérande					

Éléments de la structure		Type de données	Description
	AREA	Byte	(C) Zone de mémoire <ul style="list-style-type: none"> • L : 16#40 – 4E, 86, 87, 8E, 8F, C0 – CE • I : 16#81 • Q : 16#82 • M : 16#83 • DB : 16#84, 85, 8A, 8B
	DB_NUMBER	UInt	(D) Numéro du bloc de données
	OFFSET	UDInt	(E) Adresse relative de l'opérande

GET_ERROR_ID

Tableau 8- 147 Instruction GetErrorID

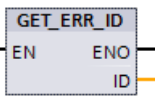
CONT/LOG	SCL	Description
	<pre>GET_ERR_ID ();</pre>	Signale qu'une erreur d'exécution du bloc de programme s'est produite et transmet l'ID (code identificateur) de l'erreur.

Tableau 8- 148 Types de données pour les paramètres

Paramètre	Type de données	Description
ID	Word	Valeurs de l'identificateur d'erreur pour l'élément ERROR_ID de ErrorStruct

Tableau 8- 149 Valeurs de Error_ID

ERROR_ID hexa-décimal	ERROR_ID décimal	Erreur d'exécution du bloc de programme
0	0	Pas d'erreur
2520	9504	Chaîne altérée
2522	9506	Erreur de lecture, opérande hors plage
2523	9507	Erreur d'écriture, opérande hors plage
2524	9508	Erreur de lecture, zone invalide
2525	9509	Erreur d'écriture, zone invalide
2528	9512	Erreur de lecture, alignement de données (alignement de bits incorrect)
2529	9513	Erreur d'écriture, alignement de données (alignement de bits incorrect)
252C	9516	Pointeur non initialisé
2530	9520	DB protégé en écriture
2533	9523	Pointeur invalide utilisé
2538	9528	Erreur d'accès : DB inexistant
2539	9529	Erreur d'accès : Mauvais DB utilisé
253A	9530	DB global inexistant
253C	9532	Mauvaise version ou FC inexistante
253D	9533	Instruction inexistante
253E	9534	Mauvaise version ou FB inexistant
253F	9535	Instruction inexistante
2550	9552	Erreur d'accès : DB inexistant
2575	9589	Erreur de profondeur d'imbrication de programme
2576	9590	Erreur d'allocation de données locales
2942	10562	Entrée physique inexistante
2943	10563	Sortie physique inexistante

Fonctionnement

Par défaut, la CPU réagit à une erreur d'exécution de bloc en consignait une erreur dans la mémoire tampon de diagnostic. Toutefois, si vous placez une ou plusieurs instructions GET_ERROR ou GET_ERROR_ID dans un bloc de code, ce bloc pourra gérer les erreurs se produisant à l'intérieur du bloc. Dans ce cas, la CPU ne consigne pas d'erreur dans la mémoire tampon de diagnostic. Au lieu de cela, l'information d'erreur est transmise à la sortie de l'instruction GET_ERROR ou GET_ERROR_ID. Vous pouvez lire les informations d'erreur détaillées avec l'instruction GET_ERROR ou uniquement l'identificateur de l'erreur avec l'instruction GET_ERROR_ID. Normalement, la première erreur est la plus importante, les erreurs suivantes n'étant que des conséquences de la première erreur.

La première exécution d'une instruction GET_ERROR ou GET_ERROR_ID à l'intérieur d'un bloc renvoie la première erreur détectée pendant l'exécution du bloc. Cette erreur peut s'être produite à n'importe quel endroit entre le début du bloc et l'exécution de GET_ERROR ou GET_ERROR_ID. Les exécutions suivantes de GET_ERROR ou GET_ERROR_ID renvoient la première erreur depuis l'exécution précédente de GET_ERROR ou GET_ERROR_ID. L'historique des erreurs n'est pas sauvegardé et l'exécution de l'une ou l'autre instruction réarme le système d'automatisation pour la capture de l'erreur suivante.

Vous pouvez ajouter le type de données ErrorStruct utilisé par l'instruction GET_ERROR dans l'éditeur de bloc de données et les éditeurs d'interface de bloc afin que la logique de votre programme puisse accéder à ces valeurs. Sélectionnez ErrorStruct dans la liste déroulante des types de données pour ajouter cette structure. Vous pouvez créer plusieurs structures ErrorStruct en leur donnant des noms univoques. Il n'est pas possible de renommer les éléments d'une structure ErrorStruct.

Situation d'erreur signalée par ENO

Si EN = TRUE et GET_ERROR ou GET_ERROR_ID s'exécute, alors :

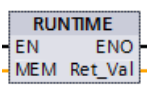
- ENO = VRAI signale une erreur d'exécution de bloc de code et la présence d'une erreur de données ;
- ENO = FAUX signale qu'aucune erreur d'exécution de bloc de code ne s'est produite.

Vous pouvez connecter de la logique de programme de réaction aux erreurs à ENO qui s'active si une erreur s'est produite. En présence d'une erreur, le paramètre de sortie mémorise les données d'erreur là où votre programme peut y accéder.

Vous pouvez utiliser GET_ERROR et GET_ERROR_ID pour envoyer des informations d'erreur du bloc en cours d'exécution (bloc appelé) à un bloc appelant. Placez l'instruction dans le dernier réseau du programme du bloc appelé afin de renvoyer l'état d'exécution final au bloc appelant.

8.8.9 RUNTIME (Mesurer le temps d'exécution du programme)

Tableau 8- 150 Instruction RUNTIME

CONT / LOG	SCL	Description
	<pre>Ret_Val := RUNTIME(_lread_inout_);</pre>	Mesure le temps d'exécution du programme complet, des blocs individuels ou des séquences de commande.

Si vous souhaitez mesurer le temps d'exécution de votre programme complet, appelez l'instruction "Mesurer le temps d'exécution du programme" dans l'OB 1. La mesure du temps d'exécution commence avec le premier appel et la sortie RET_VAL renvoie le temps d'exécution du programme après le deuxième appel. Le temps d'exécution mesuré comprend tous les processus CPU qui peuvent se produire pendant l'exécution du programme, par exemple, des interruptions causées par une communication ou des événements de niveau supérieur. L'instruction "Mesurer le temps d'exécution du programme" lit un compteur interne de la CPU et écrit la valeur dans le paramètre IN-OUT MEM. L'instruction calcule le temps d'exécution du programme actuel selon la fréquence de comptage interne et l'écrit dans la sortie RET_VAL.

Si vous souhaitez mesurer le temps d'exécution de blocs individuels ou de séquence de commande individuelles, vous avez besoin de trois réseaux distincts. Appelez l'instruction "Mesurer le temps d'exécution du programme" dans un réseau individuel à l'intérieur de votre programme. Vous définissez le point de départ de la mesure du temps d'exécution avec ce premier appel de l'instruction. Ensuite, vous appelez le bloc de programme requis ou la séquence de commande dans le réseau suivant. Dans un autre réseau, appelez l'instruction "Mesurer le temps d'exécution du programme" une deuxième fois et affectez la même mémoire au paramètre IN-OUT MEM comme vous l'avez fait lors du premier appel de l'instruction. L'instruction "Mesurer le temps d'exécution du programme" dans le troisième réseau lit un compteur CPU interne et calcule le temps d'exécution actuel du bloc de programme ou la séquence de commande selon la fréquence de comptage interne et l'écrit dans la sortie RET_VAL.

L'instruction "Mesurer le temps d'exécution du programme" utilise un compteur haute fréquence interne pour calculer la durée. Si le compteur présente un débordement, l'instruction renvoie les valeurs $\leq 0,0$. Ignorez ces valeurs du temps d'exécution.

Remarque

La CPU ne peut pas déterminer exactement le temps d'exécution d'une séquence de commande, car la séquence d'instructions au sein d'une séquence de commande change pendant la compilation optimisée du programme.

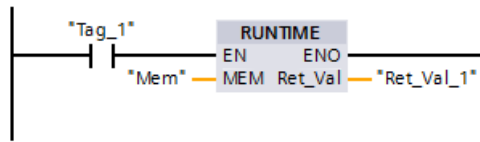
Tableau 8- 151 Types de données pour les paramètres

Paramètre	Type de données	Description
MEM	LReal	Point de départ de la mesure du temps d'exécution
RET_VAL	LReal	Temps d'exécution mesuré en secondes

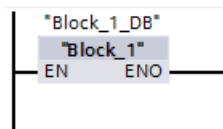
Exemple : Instruction RUNTIME

L'exemple suivant montre l'utilisation de l'instruction RUNTIME pour mesurer le temps d'exécution d'un bloc fonctionnel :

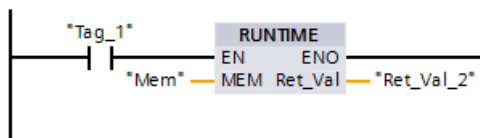
Réseau 1 :



Réseau 2 :



Réseau 3 :



Lorsque l'opérande "Tag_1" dans le réseau 1 a l'état de signal "1", l'instruction RUNTIME s'exécute. Le point de départ de la mesure du temps d'exécution est défini avec le premier appel de l'instruction et mis en mémoire tampon comme référence pour le deuxième appel de l'instruction dans l'opérande "Mem"

Le bloc fonctionnel FB1 s'exécute dans le réseau 2.

Une fois que le bloc de programme FB1 est terminé et que l'opérande "Tag_1" a l'état de signal "1", l'instruction RUNTIME dans le réseau 3 s'exécute. Le deuxième appel de l'instruction calcule le temps d'exécution du bloc de programme et écrit le résultat dans la sortie RET_VAL_2.

8.8.10 Instructions de contrôle SCL

8.8.10.1 Présentation des instructions de contrôle SCL

Le langage SCL (Structured Control Language) fournit trois types d'instructions de contrôle pour structurer votre programme utilisateur :

- Instructions de sélection : Une instruction de sélection vous permet d'aiguiller le déroulement du programme vers différentes séquences d'instructions.
- Boucles : Vous pouvez programmer l'exécution de boucles à l'aide d'instructions d'itération. Une instruction d'itération précise quelles parties d'un programme doivent être répétées en fonction de certaines conditions.
- Sauts dans le programme : Un saut dans le programme provoque le saut immédiat à un repère de saut indiqué et donc à une autre instruction dans le même bloc.

Ces instructions de gestion du programme utilisent la syntaxe du langage de programmation PASCAL.

Tableau 8- 152 Types d'instructions de contrôle SCL

Instruction de contrôle		Description
Instruction de sélection	Instruction IF-THEN (Page 331)	Permet d'aiguiller le déroulement du programme vers une branche d'une alternative selon qu'une condition est vraie ou fausse.
	Instruction CASE (Page 332)	Permet l'exécution sélective d'une variante parmi <i>n</i> variantes en fonction de la valeur d'une variable.
Boucle	Instruction FOR (Page 333)	Répète une séquence d'instructions tant que la variable de contrôle reste dans la plage de valeurs indiquée.
	Instruction WHILE-DO (Page 334)	Répète une séquence d'instructions tant qu'une condition d'exécution est satisfaite.
	Instruction REPEAT-UNTIL (Page 335)	Répète une séquence d'instructions jusqu'à ce qu'une condition d'achèvement apparaisse.
Saut dans le programme	Instruction CONTINUE (Page 336)	Sert à abandonner l'exécution de l'itération en cours de la boucle.
	Instruction EXIT (Page 337)	Sert à quitter une boucle en n'importe quel point, que la condition d'abandon soit remplie ou non.
	Instruction GOTO (Page 338)	Provoque le saut immédiat du programme à un repère indiqué.
	Instruction RETURN (Page 338)	Permet de quitter le bloc en cours d'exécution et de retourner au bloc appelant.

8.8.10.2 Instruction IF-THEN

Une instruction IF-THEN est une instruction conditionnelle qui contrôle le déroulement du programme en exécutant un groupe d'instructions en fonction de l'évaluation de la valeur Bool d'une expression logique. Vous pouvez utiliser des crochets pour imbriquer ou structurer l'exécution de plusieurs instructions IF-THEN.

Tableau 8- 153 Eléments de l'instruction IF-THEN

SCL	Description
<pre>IF "condition" THEN instruction_A; instruction_B; instruction_C; ;</pre>	<p>Si "condition" est VRAI ou 1, le programme exécute les instructions suivantes jusqu'à ce qu'il rencontre l'instruction END_IF.</p> <p>Si "condition" est FAUX ou 0, le programme va à l'instruction END_IF (à moins qu'il ne contienne des instructions ELSIF ou ELSE optionnelles).</p>
<pre>[ELSIF "condition-n" THEN instruction_N; ;]</pre>	<p>L'instruction ELSIF¹ optionnelle fournit des conditions additionnelles à évaluer. Par exemple, si "condition" dans l'instruction IF-THEN est FAUX, le programme évalue "condition-n". Si "condition-n" est VRAI, le programme exécute "instruction_N".</p>
<pre>[ELSE instruction_X; ;]</pre>	<p>L'instruction ELSE optionnelle fournit des instructions à exécuter lorsque "condition" dans l'instruction IF-THEN est FAUX.</p>
<pre>END_IF;</pre>	<p>L'instruction END_IF met fin à l'instruction IF-THEN.</p>

¹ Vous pouvez inclure plusieurs instructions ELSIF à l'intérieur d'une instruction IF-THEN.

Tableau 8- 154 Variables pour l'instruction IF-THEN

Variables	Description
"condition"	Obligatoire. L'expression logique est soit vraie (TRUE ou 1), soit fausse (FALSE ou 0).
"instruction_A"	Facultatif. Une ou plusieurs instructions à exécuter lorsque "condition" est VRAI.
"condition-n"	Facultatif. Expression logique que l'instruction optionnelle ELSIF doit évaluer.
"instruction_N"	Facultatif. Une ou plusieurs instructions à exécuter lorsque "condition-n" de l'instruction ELSIF est VRAI.
"instruction_X"	Facultatif. Une ou plusieurs instructions à exécuter lorsque "condition" de l'instruction IF-THEN est FAUX.

L'exécution d'une instruction IF est régie par les règles suivantes :

- La première séquence d'instructions dont l'expression logique est vraie (TRUE) est exécutée. Les séquences d'instructions restantes ne sont pas exécutées.
- Si aucune expression booléenne n'est vraie, c'est la séquence d'instructions introduite par ELSE qui est exécutée (ou aucune séquence d'instructions s'il n'y a pas de branche ELSE).
- Le nombre d'instructions ELSIF possibles est illimité.

Remarque

L'avantage d'utiliser une ou plusieurs branches ELSIF plutôt qu'une suite d'instructions IF réside dans le fait que les expressions logiques qui suivent une expression évaluée comme vraie ne sont plus évaluées. Cela permet de réduire la durée d'exécution du programme.

8.8.10.3 Instruction CASE

Tableau 8- 155 Eléments de l'instruction CASE

SCL	Description
<pre> CASE "Valeur_test" OF "ListeValeurs": Instruction[; Instruction, ...] "ListeValeurs": Instruction[; Instruction, ...] [ELSE Instruction-Else[; Instruction-Else, ...]] END_CASE;</pre>	<p>L'instruction CASE exécute un groupe d'instructions parmi plusieurs groupes en fonction de la valeur d'une expression.</p>

Tableau 8- 156 Paramètres

Paramètre	Description
"Valeur_test"	Obligatoire. Toute expression numérique de type de données Int
"ListeValeurs"	Obligatoire. Valeur unique ou liste de valeurs ou de plages de valeurs séparées par des virgules (utilisez deux points pour définir une plage de valeurs : 2..8). L'exemple suivant illustre les différents types de listes de valeurs : <pre> 1 : Instruction_A; 2, 4: Instruction_B; 3, 5..7,9: Instruction_C;</pre>
Instruction	Obligatoire. Une ou plusieurs instructions qui sont exécutées lorsque "Valeur_test" correspond à une valeur quelconque dans la liste de valeurs.
Instruction-Else	Facultatif. Une ou plusieurs instructions qui sont exécutées s'il n'existe aucune occurrence de "Valeur_test" dans la liste "ListeValeurs".

L'exécution d'une instruction CASE est régie par les règles suivantes :

- L'expression Valeur_test doit renvoyer une valeur de type Int.
- Lors de l'exécution d'une instruction CASE, le programme vérifie si la valeur de l'expression Valeur_test figure dans la liste de valeurs spécifiée. Si tel est le cas, la section d'instructions affectée à la liste est exécutée.
- Si la valeur de l'expression de sélection est introuvable, c'est la section d'instructions suivant ELSE qui est exécutée ou aucune instruction s'il n'y a pas de branche ELSE.

Exemple : Instructions CASE imbriquées

Il est possible d'imbriquer des instructions CASE. Chaque instruction CASE imbriquée doit être associée à une instruction END_CASE.

```

CASE "var1" OF
    1 : #var2 := 'A';
    2 : #var2 := 'B';
ELSE
    CASE "var3" OF

        65..90: #var2 := 'Majuscule';
        97..122: #var2 := 'Minuscule';

    ELSE

        #var2:= 'SpecialCharacter';

    END_CASE;
END_CASE;

```

8.8.10.4 Instruction FOR

Tableau 8- 157 Eléments de l'instruction FOR

SCL	Description
<pre> FOR "variable_contrôle" := "début" TO "fin" [BY "incrément"] DO instruction; ; END_FOR; </pre>	<p>Une instruction FOR sert à répéter une séquence d'instructions aussi longtemps qu'une variable de contrôle se trouve dans la plage de valeurs spécifiée. La définition d'une boucle avec FOR comporte aussi l'indication d'une valeur initiale et d'une valeur finale. Ces deux valeurs doivent avoir le même type que la variable de contrôle.</p> <p>Vous pouvez imbriquer des boucles FOR. L'instruction END_FOR se réfère à la dernière instruction FOR exécutée.</p>

Tableau 8- 158 Paramètres

Paramètre	Description
"variable_contrôle"	Obligatoire. Entier (Int ou DInt) servant de compteur de boucle
"début"	Obligatoire. Expression simple qui indique la valeur initiale de la variable de contrôle.
"fin"	Obligatoire. Expression simple qui indique la valeur finale de la variable de contrôle.
"incrément"	Facultatif. Quantité de laquelle la variable de contrôle est modifiée après chaque boucle. "incrément" doit avoir le même type de données que la variable de contrôle. Si aucun incrément n'est indiqué, la valeur des variables d'exécution est augmentée de 1 après chaque boucle. Vous ne pouvez pas modifier la valeur de "incrément" pendant l'exécution de l'instruction FOR.

L'instruction FOR s'exécute comme suit :

- Au début de la boucle, la variable de contrôle prend la valeur initiale (affectation initiale) puis, à chaque itération de la boucle, elle est incrémentée (en cas d'incrément positif) ou décrétementée (en cas d'incrément négatif) de l'incrément indiqué, et ce jusqu'à ce que la valeur finale soit atteinte.
- Après chaque exécution de la boucle, la condition (valeur finale atteinte) est vérifiée afin de constater si elle est ou non remplie. Si la condition de fin n'est pas remplie, la séquence d'instructions est réexécutée. Sinon, la boucle s'arrête et l'exécution se poursuit avec l'instruction qui suit immédiatement la boucle.

Règles de formulation des instructions FOR :

- La variable de contrôle doit être exclusivement du type de données Int ou DInt.
- Vous pouvez omettre l'instruction BY [incrément]. Si aucun incrément est indiqué, la valeur +1 est prise par défaut.

Utilisez l'instruction Instruction EXIT (Page 337) pour quitter la boucle quel que soit l'état de l'expression "condition". L'instruction EXIT exécute l'instruction se trouvant immédiatement après l'instruction END_FOR.

Utilisez l'Instruction CONTINUE (Page 336) pour sauter les instructions suivantes dans la boucle FOR et aller immédiatement vérifier si la condition d'abandon de la boucle est remplie ou non.

8.8.10.5 Instruction WHILE-DO

Tableau 8- 159 Instruction WHILE

SCL	Description
<pre> WHILE "condition" DO Instruction; Instruction; ...; END WHILE;</pre>	<p>L'instruction WHILE exécute une série d'instructions tant qu'une condition donnée est vraie (TRUE).</p> <p>Vous pouvez imbriquer des boucles WHILE. L'instruction END_WHILE se réfère à la dernière instruction WHILE exécutée.</p>

Tableau 8- 160 Paramètres

Paramètre	Description
"condition"	Obligatoire. Expression logique qui peut prendre la valeur VRAI ou FAUX (une condition "nulle" est interprétée comme fausse).
Instruction	Facultatif. Une ou plusieurs instructions qui sont exécutées tant que la condition est vraie.

Remarque

L'instruction WHILE évalue l'état de "condition" avant d'exécuter les instructions. Pour exécuter les instructions au moins une fois quel que soit l'état de "condition", utilisez l'instruction REPEAT (Page 335).

L'exécution d'une instruction WHILE est régie par les règles suivantes :

- La condition d'exécution est évaluée avant chaque itération du corps de boucle.
- Le corps de boucle suivant DO est répété tant que la condition d'exécution a la valeur VRAI.
- Dès que la condition d'exécution prend la valeur FAUX, la boucle est sautée et l'instruction suivant la boucle est exécutée.

Utilisez l'instruction Instruction EXIT (Page 337) pour quitter la boucle quel que soit l'état de l'expression "condition". L'instruction EXIT exécute l'instruction se trouvant immédiatement après l'instruction END_WHILE.

Utilisez l'instruction CONTINUE pour sauter les instructions suivantes dans la boucle WHILE et aller immédiatement vérifier que la condition d'abandon de la boucle est remplie ou non.

8.8.10.6 Instruction REPEAT-UNTIL

Tableau 8- 161 Instruction REPEAT

SCL	Description
<pre> REPEAT Instruction; ; UNTIL "condition" END_REPEAT;</pre>	<p>L'instruction REPEAT exécute un groupe d'instructions jusqu'à ce qu'une condition donnée soit vraie (TRUE).</p> <p>Vous pouvez imbriquer des boucles REPEAT. L'instruction END_REPEAT se réfère toujours à la dernière instruction REPEAT exécutée.</p>

Tableau 8- 162 Paramètres

Paramètre	Description
Instruction	Facultatif. Une ou plusieurs instructions qui sont exécutées jusqu'à ce que la condition soit vraie.
"condition"	Obligatoire. Une ou plusieurs expressions des deux types suivants : Expression numérique ou chaîne de caractères qui peut prendre la valeur VRAI ou FAUX. Une condition "nulle" est interprétée comme fausse.

Remarque

Avant d'évaluer l'état de "condition", l'instruction REPEAT exécute une fois les instructions dans la boucle (même si "condition" est FAUX). Pour que l'état de "condition" soit évalué avant l'exécution des instructions, utilisez l'instruction WHILE (Page 334).

Utilisez l'instruction Instruction EXIT (Page 337) pour quitter la boucle quel que soit l'état de l'expression "condition". L'instruction EXIT exécute l'instruction se trouvant immédiatement après l'instruction END_REPEAT.

Utilisez l'instruction Instruction CONTINUE (Page 336) pour sauter les instructions suivantes dans la boucle REPEAT et continuer la boucle par la vérification que la condition d'abandon est remplie ou non.

8.8.10.7 Instruction CONTINUE

Tableau 8- 163 Instruction CONTINUE

SCL	Description
CONTINUE Instruction; ;	L'instruction CONTINUE saute les instructions suivantes d'un programme de boucle (FOR, WHILE, REPEAT) et va immédiatement vérifier si la condition d'abandon de la boucle est remplie ou non. Si ce n'est pas le cas, la boucle continue.

L'exécution d'une instruction CONTINUE est régie par les règles suivantes :

- Cette instruction met immédiatement fin à l'exécution du corps de la boucle.
- Selon que la condition de répétition de la boucle est remplie ou non, le corps de la boucle est réexécuté ou bien l'instruction d'itération est abandonnée et l'instruction venant immédiatement après est exécutée.
- Dans une instruction FOR, la variable de contrôle est incrémentée de l'incrément indiqué aussitôt après une instruction CONTINUE.

Utilisez l'instruction CONTINUE uniquement à l'intérieur d'une boucle. Dans les boucles imbriquées, CONTINUE fait toujours référence à la boucle dans laquelle il est directement inclus. CONTINUE est typiquement utilisé en combinaison avec une instruction IF.

S'il s'agit d'abandonner la boucle quel que soit le résultat du test d'achèvement, utilisez l'instruction EXIT.

Exemple : instruction CONTINUE

L'exemple suivant montre l'utilisation de l'instruction CONTINUE pour éviter une division par 0 lors du calcul du pourcentage d'une valeur :

```

FOR i := 0 TO 10 DO
IF valeur[i] = 0 THEN CONTINUE; END_IF;
    p := partie / valeur[i] * 100;
    s := INT_TO_STRING(p);
    pourcentage := CONCAT(IN1:=s, IN2:="%");
END_FOR;
    
```

8.8.10.8 Instruction EXIT

Tableau 8- 164 Instruction EXIT

SCL	Description
EXIT;	L'instruction EXIT permet de quitter une boucle (FOR, WHILE ou REPEAT) en n'importe quel point, que la condition d'achèvement soit remplie ou non.

L'exécution d'une instruction EXIT est régie par les règles suivantes :

- Cette instruction provoque l'abandon instantané de l'instruction d'itération qui entoure immédiatement l'instruction EXIT.
- L'exécution du programme se poursuit après la fin de la boucle (par exemple, après END_FOR).

Utilisez l'instruction EXIT à l'intérieur d'une boucle. Dans les boucles imbriquées, l'instruction EXIT rend la main à la boucle de niveau immédiatement supérieur.

Exemple : instruction EXIT

```
FOR i := 0 TO 10 DO
CASE valeur[i, 0] OF
  1..10: valeur [i, 1]:="A";
  11..40: valeur [i, 1]:="B";
  41..100: valeur [i, 1]:="C";
ELSE
EXIT;
END_CASE;
END_FOR;
```

8.8.10.9 Instruction GOTO

Tableau 8- 165 Instruction GOTO

SCL	Description
<pre>GOTO JumpLabel; instruction; ... ; JumpLabel: instruction;</pre>	<p>L'instruction GOTO évite des instructions en sautant directement à un repère dans le même bloc.</p> <p>Le repère de saut ("JumpLabel") et l'instruction GOTO doivent être dans le même bloc. Le nom d'un repère de saut ne peut être affecté qu'une fois à l'intérieur d'un bloc. Chaque repère de saut peut être la cible de plusieurs instructions GOTO.</p>

Il n'est pas possible d'entrer dans une boucle par un saut (FOR, WHILE ou REPEAT). Il est possible de quitter une boucle par un saut.

Exemple : instruction GOTO

Dans l'exemple suivant, selon la valeur de l'opérande "valeur_variable", l'exécution du programme reprend au point défini par le repère de saut correspondant. Si "valeur_variable" est égal à 2, l'exécution du programme se poursuit au repère de saut "MyLabel2" et évite "MyLabel1".

```
CASE "valeur_variable" OF
1 : GOTO MyLabel1;
2 : GOTO MyLabel2;
ELSE GOTO MyLabel3;
END_CASE;
MyLabel1: "variable_1" := 1;
MyLabel2: "variable_2" := 1;
MyLabel3: "variable_4" := 1;
```

8.8.10.10 Instruction RETURN

Tableau 8- 166 Instruction RETURN

SCL	Description
<pre>RETURN;</pre>	<p>L'instruction RETURN quitte sans condition le bloc de code en cours d'exécution. L'exécution du programme revient au bloc appelant ou au système d'exploitation (lors de la sortie d'un OB).</p>

Exemple : instruction RETURN

```
IF "Erreur" <> 0 THEN
RETURN;
END_IF;
```

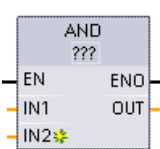
Remarque

Après exécution de la dernière instruction, le bloc de code revient automatiquement au bloc appelant. N'insérez pas d'instruction RETURN à la fin du bloc de code.

8.9 Opérations logiques sur mots

8.9.1 Opérations logiques ET, OU et OU EXCLUSIF

Tableau 8- 167 Opérations logiques ET, OU et OU EXCLUSIF

CONT/LOG	SCL	Description
	<code>out := in1 AND in2;</code>	AND : ET logique
	<code>out := in1 OR in2;</code>	OR : OU logique
	<code>out := in1 XOR in2;</code>	XOR : OU EXCLUSIF logique

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.



Pour ajouter une entrée , cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants et sélectionnez la commande "Insérer entrée".

Pour supprimer une entrée, cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

Tableau 8- 168 Types de données pour les paramètres


Paramètre	Type de données	Description
IN1, IN2	Byte, Word, DWord	Entrées logiques
OUT	Byte, Word, DWord	Sortie logique

1 La sélection du type de données donne le même type de données aux paramètres IN1, IN2 et OUT.

Les valeurs binaires correspondantes de IN1 et IN2 sont combinées pour produire un résultat logique binaire qui est transmis au paramètre OUT. ENO est toujours VRAI après l'exécution de ces instructions.

8.9.2 INV (Former le complément à 1)

Tableau 8- 169 Instruction INV

CONT/LOG	SCL	Description
	Non disponible	Calcule le complément à un binaire du paramètre IN. Le complément à un est formé en inversant chaque valeur de bit dans le paramètre IN (changement de chaque 0 en 1 et de chaque 1 en 0). ENO est toujours VRAI après l'exécution de cette instruction.

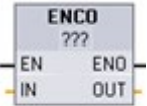
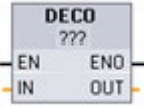
1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 170 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Élément de données à inverser
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Sortie inversée

8.9.3 Instructions DECO (Décoder) et ENCO (Encoder)

Tableau 8- 171 Instructions ENCO et DECO

CONT/LOG	SCL	Description
	<pre>out := ENCO(_in_);</pre>	<p>Code un motif binaire en un nombre binaire.</p> <p>L'instruction ENCO convertit le paramètre IN en un nombre binaire correspondant à la position du bit mis à 1 le moins significatif du paramètre IN et renvoie le résultat dans le paramètre OUT. Si le paramètre IN est soit 0000 0001 soit 0000 0000, la valeur 0 est renvoyée dans le paramètre OUT. Si la valeur du paramètre IN est 0000 0000, ENO est mis à FAUX.</p>
	<pre>out := DECO(_in_);</pre>	<p>Décode un nombre binaire en un motif binaire.</p> <p>L'instruction DECO décode un nombre binaire dans le paramètre IN en mettant à 1 la position de bit correspondante dans le paramètre OUT (tous les autres bits sont mis à 0). ENO est toujours VRAI après l'exécution de l'instruction DECO.</p> <p>Remarque : Le type de données par défaut pour l'instruction DECO est DWORD. Dans SCL, modifiez le nom de l'instruction en DECO_BYTE ou DECO_WORD pour décoder une valeur d'octet ou de mot, respectivement, et indiquez une variable ou une adresse octet ou mot.</p>

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 8- 172 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	ENCO : Byte, Word, DWord DECO : UInt	ENCO : Motif binaire à encoder DECO : Valeur à décoder
OUT	ENCO : Int DECO : Byte, Word, DWord	ENCO : Valeur encodée DECO : Motif binaire décodé

Tableau 8- 173 Etat de ENO

ENO	Situation d'erreur	Résultat (OUT)
1	Pas d'erreur	Numéro de bit valide
0	IN est zéro.	OUT est mis à zéro.

La sélection du type de données Byte, Word ou DWord pour le paramètre OUT de DECO restreint la plage utile du paramètre IN. Si la valeur du paramètre IN dépasse la plage utile, une opération modulo est exécutée pour extraire les bits les moins significatifs comme illustré ci-dessous.

Plage du paramètre IN de DECO :

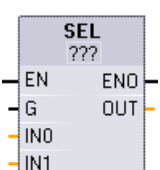
- 3 bits (valeurs 0-7) IN sont utilisés pour mettre à 1 une position de bit dans un octet OUT (Byte OUT).
- 4 bits (valeurs 0-15) IN sont utilisés pour mettre à 1 une position de bit dans un mot OUT (Word OUT).
- 5 bits (valeurs 0-31) IN sont utilisés pour mettre à 1 une position de bit dans un double mot OUT (DWord OUT).

Tableau 8- 174 Exemples

Valeur IN de DECO			Valeur OUT de DECO (Décoder une position de bit unique)
Byte OUT 8 bits	Min. IN	0	00000001
	Max. IN	7	10000000
Word OUT 16 bits	Min. IN	0	0000000000000001
	Max. IN	15	1000000000000000
DWord OUT 32 bits	Min. IN	0	00000000000000000000000000000001
	Max. IN	31	10000000000000000000000000000000

8.9.4 Instructions SEL (Sélectionner), MUX (Multiplexeur), et DEMUX (Démultiplexeur)

Tableau 8- 175 Instruction SEL (Sélectionner)

CONT/LOG	SCL	Description
	<pre>out := SEL(g:=_bool_in, in0:=-_variant_in, in1:=-_variant_in);</pre>	SEL affecte une valeur parmi deux valeurs d'entrée au paramètre OUT en fonction de la valeur du paramètre G.

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

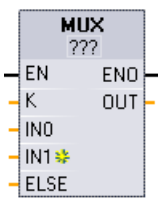
Tableau 8- 176 Types de données pour l'instruction SEL

Paramètre	Type de données ¹	Description
G	Bool	<ul style="list-style-type: none"> • 0 sélectionne IN0 • 1 sélectionne IN1
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Entrées
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Sortie

¹ Les variables d'entrée et la variable de sortie doivent avoir le même type de données.

Codes d'erreur : ENO est toujours vrai après l'exécution de l'instruction SEL.

Tableau 8- 177 Instruction MUX (Multiplexeur)

CONT/LOG	SCL	Description
	<pre> out := MUX(k:=_unit_in, in1:=variant_in, in2:=variant_in, [... in32:=variant_in,] in_else:=variant_in); </pre>	MUX copie une valeur parmi plusieurs valeurs d'entrée dans le paramètre OUT en fonction de la valeur du paramètre K. Si la valeur du paramètre K dépasse (IN n - 1), la valeur du paramètre ELSE est copiée dans le paramètre OUT.

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.



Pour ajouter une entrée , cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants et sélectionnez la commande "Insérer entrée".

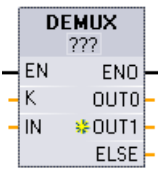
Pour supprimer une entrée, cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

Tableau 8- 178 Types de données pour l'instruction MUX

Paramètre	Type de données	Description
K	UInt	<ul style="list-style-type: none"> • 0 sélectionne IN1 • 1 sélectionne IN2 • n sélectionne INn
IN0, IN1, .. IN n	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Entrées
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Valeur de substitution d'entrée (optionnelle)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Sortie

¹ Les variables d'entrée et la variable de sortie doivent avoir le même type de données.

Tableau 8- 179 Instruction DEMUX (Démultiplexeur)

CONT/LOG	SCL	Description
	<pre> DEMUX (k:=_unit_in, in:=variant_in, out1:=variant_in, out2:=variant_in, [...out32:=variant_in,] outelse:=variant_in); </pre>	DEMUX copie la valeur de l'adresse affectée au paramètre IN dans une sortie parmi plusieurs sorties. La valeur du paramètre K indique la sortie sélectionnée comme destination de la valeur IN. Si la valeur de K est supérieure au nombre (OUTn - 1), la valeur dans IN est copiée dans le paramètre ELSE.

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.



Pour ajouter une sortie, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants et sélectionnez la commande "Insérer sortie".

Pour supprimer une sortie, cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants (lorsqu'il y a plus de sorties que les deux sorties d'origine) et sélectionnez la commande "Supprimer".

Tableau 8- 180 Types de données pour l'instruction DEMUX

Paramètre	Type de données ¹	Description
K	UInt	Valeur de sélecteur : <ul style="list-style-type: none"> • 0 sélectionne OUT1 • 1 sélectionne OUT2 • n sélectionne OUTn
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Entrée
OUT0, OUT1, .. OUTn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Sorties
ELSE	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Sortie de substitution lorsque K est supérieur à (OUTn - 1)

¹ La variable d'entrée et les variables de sortie doivent avoir le même type de données.


Tableau 8- 181 Etat de ENO pour les instructions MUX et DEMUX

ENO	Situation d'erreur	Résultat OUT
1	Pas d'erreur	MUX : La valeur IN sélectionnée est copiée dans OUT. DEMUX : La valeur IN est copiée dans la sortie OUT sélectionnée.
0	MUX : K est supérieur au nombre d'entrées -1	<ul style="list-style-type: none"> • Pas de valeur ELSE fournie : OUT reste inchangé • Valeur ELSE fournie : la valeur de ELSE est affectée à OUT
	DEMUX : K est supérieur au nombre de sorties -1	<ul style="list-style-type: none"> • Pas de valeur ELSE fournie : les sorties restent inchangées • Valeur ELSE fournie : la valeur de IN est copiée dans ELSE

8.10 Décalage et rotation

8.10.1 Instruction SHR (Décaler à droite) et SHL (Décaler à gauche)

Tableau 8- 182 Instructions SHR et SHL

CONT/LOG	SCL	Description
	<pre> out := SHR (in:= _variant_in_, n:= _uint_in_); out := SHL (in:= _variant_in_, n:= _uint_in_); </pre>	Utilisez les instructions de décalage (SHL et SHR) pour décaler le motif binaire du paramètre IN. Le résultat est affecté au paramètre OUT. Le paramètre N indique le nombre de positions de bit à décaler. <ul style="list-style-type: none"> • SHR : Décaler le motif binaire vers la droite • SHL : Décaler le motif binaire vers la gauche

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.

Tableau 8- 183 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Entiers	Motif binaire à décaler
N	USInt, UDint	Nombre de positions de bit à décaler
OUT	Entiers	Motif binaire après l'opération de décalage

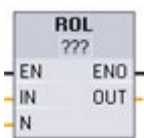
- Pour N=0, aucun décalage n'est effectué. La valeur IN est affectée à OUT.
- Des zéros sont insérés dans les positions de bit vidées par l'opération de décalage.
- Si le nombre de positions à décaler (N) dépasse le nombre de bits dans la valeur cible (8 pour Byte, 16 pour Word, 32 pour DWord), toutes les valeurs de bit d'origine sont décalées et remplacées par des zéros (zéro est affecté à OUT).
- ENO est toujours VRAI pour les instructions de décalage.

Tableau 8- 184 Exemple : SHL pour des données de type Word

Décaler les bits d'un mot vers la gauche en insérant des zéros par la droite (N = 1)			
IN	1110 0010 1010 1101	Valeur OUT avant le premier décalage :	1110 0010 1010 1101
		Après le premier décalage vers la gauche :	1100 0101 0101 1010
		Après le deuxième décalage vers la gauche :	1000 1010 1011 0100
		Après le troisième décalage vers la gauche :	0001 0101 0110 1000
		:	

8.10.2 Instructions ROR (Rotation à droite) et ROL (Rotation à gauche)

Tableau 8- 185 Instructions ROR et ROL

CONT/LOG	SCL	Description
	<pre> out := ROL(in:=_variant_in_, n:=_uint_in); out := ROR(in:=_variant_in_, n:=_uint_in); </pre>	<p>Les instructions de rotation (ROR et ROL) permettent d'opérer une rotation sur le motif binaire du paramètre IN. Le résultat est affecté au paramètre OUT. Le paramètre N définit le nombre de positions de bit à faire tourner.</p> <ul style="list-style-type: none"> • ROR : Rotation du motif binaire vers la droite • ROL : Rotation du motif binaire vers la gauche

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.

Tableau 8- 186 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Entiers	Motif binaire devant faire l'objet de la rotation
N	USInt, UDint	Nombre de positions de bit à faire tourner
OUT	Entiers	Motif binaire après l'opération de rotation

- Pour N=0, aucune rotation n'est effectuée. La valeur IN est affectée à OUT.
- Les données binaires objet de la rotation qui sortent d'un côté de la valeur cible sont insérées de l'autre côté de la valeur cible de sorte qu'aucune valeur binaire d'origine n'est perdue.
- Si le nombre de positions de bit à faire tourner (N) dépasse le nombre de bits dans la valeur cible (8 pour Byte, 16 pour Word, 32 pour DWord), la rotation est quand même exécutée.
- ENO est toujours VRAI après l'exécution des instructions de rotation.

Tableau 8- 187 Exemple : ROR pour des données de type Word

Rotation de bits sortant par la droite et entrant par la gauche (N = 1)			
IN	0100 0000 0000 0001	Valeur OUT avant la première rotation :	0100 0000 0000 0001
		Après la première rotation vers la droite :	1010 0000 0000 0000
		Après la deuxième rotation vers la droite :	0101 0000 0000 0000

Instructions avancées

9.1 Fonctions date, heure et horloge

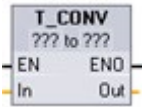
9.1.1 Instructions pour la date et l'heure

Utilisez les instructions sur la date et l'heure pour des calculs sur le calendrier et l'heure.

- T_CONV convertit une valeur (types de données "date et heure") et (types de données d'une taille de byte, word et dword)
- T_ADD additionne des valeurs Time et DTL : (Time + Time = Time) ou (DTL + Time = DTL)
- T_SUB soustrait des valeurs Time et DTL : (Time - Time = Time) ou (DTL - Time = DTL)
- T_DIFF fournit la différence entre deux valeurs DTL en tant que valeur Time : DTL - DTL = Time
- T_COMBINE combine une valeur Date et une valeur Time_and_Date pour créer une valeur DTL.

Pour plus d'informations sur le format des données DTL et Time, reportez-vous au paragraphe sur les types de données "date et heure" (Page 133).

Tableau 9- 1 Instruction T_CONV (Convertir et extraire les temps)

CONT/LOG	Exemple SCL	Description
	<pre> out := DINT_TO_TIME (in:= _variant_in); out := TIME_TO_DINT (in:= _variant_in); </pre>	<p>T_CONV convertit une valeur (types de données "date et heure") et (types de données d'une taille de byte, word et dword).</p>

- 1 Pour les boîtes CONT et LOG : Cliquez sur "???" et sélectionnez les types de données source/cible dans le menu déroulant.
- 2 Pour SCL : Faites glisser T_CONV dans l'éditeur de programmes à partir de l'arborescence d'instruction, puis sélectionnez les types de données source/cible.

Tableau 9- 2 Types de données valides pour les conversions T_CONV

Type de données IN (ou OUT)	Types de données OUT (ou IN)
TIME (millisecondes)	DInt, Int, SInt, UDInt, UInt, USInt, TOD SCL uniquement : Byte, Word, Dword
DATE (nombre de jours depuis le 1er janv. 1990)	DInt, Int, SInt, UDInt, UInt, USInt, DTL SCL uniquement : Byte, Word, Dword
TOD (millisecondes depuis minuit - 24:00:00.000)	DInt, Int, SInt, UDInt, UInt, USInt, TIME, DTL SCL uniquement : Byte, Word, Dword

Remarque

Utilisation de T_CONV pour convertir une taille de données plus grande en une taille de données plus petite

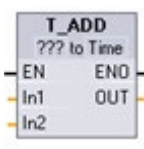
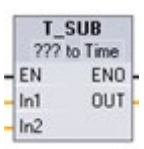
Les valeurs de données peuvent être tronquées lorsque vous convertissez un type de données plus grand en un type de données plus petit avec moins d'octets. Si cette erreur se produit, ENO est mis à 0.

Conversion de/en type de données DTL

DTL (Date and Time Long) contient les données année, mois, date et heure. Les données DTL peuvent être converties en types de données DATE et TOD et vice versa. Toutefois, la conversion DTL en données DATE ne touche que les valeurs année, mois et jour. La conversion DTL en données TOD ne touche que les valeurs heure, minutes et secondes.

Lorsque T_CONV convertit en DTL, les éléments de données qui ne sont pas touchés dans le format DTL restent inchangés.

Tableau 9- 3 Instructions T_ADD (Additionner les temps) et T_SUB (Soustraire les temps)

CONT/LOG	SCL	Description
	<pre>out := T_ADD(in1:=_variant_in, in2:=_time_in);</pre>	<p>T_ADD additionne la valeur d'entrée IN1 (de type de données DTL ou Time) à la valeur d'entrée IN2 de type Time. Le paramètre OUT fournit le résultat de type DTL ou Time. Deux opérations sont possibles selon les types de données :</p> <ul style="list-style-type: none"> • Time + Time = Time • DTL + Time = DTL
	<pre>out := T_SUB(in1:=_variant_in, in2:=_time_in);</pre>	<p>T_SUB soustrait la valeur IN2 de type Time de la valeur IN1 (de type DTL ou Time). Le paramètre OUT fournit la valeur de différence en tant que type de données DTL ou Time. Deux opérations sont possibles selon les types de données :</p> <ul style="list-style-type: none"> • Time - Time = Time • DTL - Time = DTL

¹ Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.

Tableau 9- 4 Types de données pour les paramètres de T_ADD et T_SUB

Paramètre et type		Type de données	Description
IN1 ¹	IN	DTL, Time	Valeur DTL ou Time
IN2	IN	Time	Valeur Time à ajouter ou à soustraire
OUT	OUT	DTL, Time	Somme ou différence DTL ou Time

¹ Sélectionnez le type de données de IN1 dans la liste déroulante disponible sous le nom de l'instruction. La sélection du type de données de IN1 définit également le type de données du paramètre OUT.

Tableau 9- 5 Instruction T_DIFF (Différence de temps)


CONT/LOG	SCL	Description
	<pre>out := T_DIFF(in1:= _DTL_in, in2:= _DTL_in);</pre>	<p>T_DIFF soustrait la valeur DTL (IN2) de la valeur DTL (IN1). Le paramètre OUT fournit la valeur de différence en tant que type de données Time.</p> <ul style="list-style-type: none"> DTL - DTL = Time

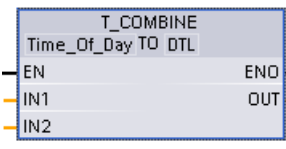
Tableau 9- 6 Types de données pour les paramètres de T_DIFF

Paramètre et type		Type de données	Description
IN1	IN	DTL	Valeur DTL
IN2	IN	DTL	Valeur DTL à soustraire
OUT	OUT	Time	Différence de temps de type Time

Codes d'erreur : ENO = 1 signifie qu'aucune erreur ne s'est produite. ENO = 0 et paramètre OUT = 0 pour les erreurs :

- Valeur DTL invalide
- Valeur Time invalide

Tableau 9- 7 Instruction T_COMBINE (Combiner les temps)


CONT/LOG	SCL	Description
	<pre>out := CONCAT_DATE_TOD (In1 := _date_in, In2 := _tod_in);</pre>	<p>T_COMBINE combine une valeur Date et une valeur Time_of_Day pour créer une valeur DTL.</p>

¹ Notez que l'instruction avancée T_COMBINE correspond à la fonction CONCAT_DATE_TOD dans SCL.

Tableau 9- 8 Types de données pour les paramètres de T_COMBINE

Paramètre et type		Type de données	Description
IN1	IN	Date	La valeur Date à combiner doit être comprise entre DATE#1990-01-01 et DATE#2089-12-31.
IN2	IN	Time_of_Day	Valeur Time_of_Day à combiner
OUT	OUT	DTL	Valeur DTL

9.1.2 Fonctions d'horloge

<p> ATTENTION</p> <p>Accès possible d'un pirate à vos réseaux par la synchronisation NTP</p> <p>Si un pirate parvient à accéder à vos réseaux par la synchronisation NTP (Network Time Protocol), il aura la possibilité de perturber la commande de votre processus en décalant l'heure système de la CPU. De telles perturbations dans la commande du processus peuvent entraîner la mort, des blessures graves ou des dommages matériels.</p> <p>La fonction client NTP de la CPU S7-1200 est désactivée par défaut. Lorsqu'elle est activée, elle permet uniquement aux adresses IP configurées d'agir en tant que serveur NTP. Par défaut, la CPU désactive cette fonction et vous devez la configurer pour permettre les corrections à distance de l'heure système de la CPU.</p> <p>La CPU S7-1200 prend en charge les alarmes horaires et les instructions d'horloge qui sont dépendantes de la précision de l'heure système de la CPU. Si vous configurez NTP et acceptez la synchronisation de l'heure par un serveur, vous devez vous assurer que le serveur est fiable. Vous risquez sinon une brèche de sécurité permettant à un utilisateur inconnu d'altérer la commande de votre processus par le décalage de l'heure système de la CPU.</p> <p>Pour obtenir des informations et des recommandations sur la sécurité, voir nos "Recommandations d'opération pour la sécurité industrielle" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) sur le site de service et d'assistance :</p>
--

Utilisez les instructions d'horloge pour régler et lire l'horloge système de la CPU. Le type de données DTL (Page 133) sert à fournir des valeurs de date et d'heure.

Tableau 9- 9 Instructions pour l'heure système


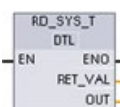
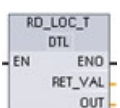
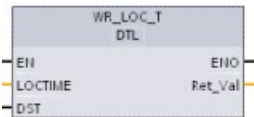
CONT/LOG	SCL	Description
	<pre>ret_val := WR_SYS_T (in:= DTL in);</pre>	WR_SYS_T (Régler l'heure) règle l'horloge temps réel de la CPU à l'aide de la valeur DTL dans le paramètre IN. Cette valeur de temps n'inclut pas les décalages pour le fuseau horaire local et l'heure d'été.
	<pre>ret_val := RD_SYS_T (out=>_DTL_out);</pre>	RD_SYS_T (Lire l'heure) lit l'heure système en cours dans la CPU. Cette valeur de temps n'inclut pas les décalages pour le fuseau horaire local et l'heure d'été.
	<pre>ret_val := RD_LOC_T (out=>_DTL_out);</pre>	RD_LOC_T (Lire l'heure locale) fournit l'heure locale en cours de la CPU sous forme de type de données DTL. Cette valeur correspond au fuseau horaire local avec les ajustements appropriés pour l'heure d'été (si configurée).
	<pre>ret_val := WR_LOC_T (LOCTIME:=DTL_in_, DST: _in_;</pre>	<p>WR_LOC_T (Ecrire l'heure locale) permet de régler la date et l'heure de l'horloge de la CPU. Vous entrez les indications sur les date et heure comme heure locale au paramètre LOCTIME à l'aide du type de données DTL. L'instruction calcule au moyen de la structure DB "TimeTransformationRule (Page 353)" l'heure système. La granularité de l'indication temporelle pour l'heure locale et le temps système est spécifique au produit et s'élève à au moins une milliseconde. Les valeurs d'entrée au paramètre LOCTIME, qui sont inférieures à ce que la CPU prend en charge, sont arrondies à l'entier supérieur lors du calcul du temps système.</p> <p>Remarque : Vous devez utiliser la configuration d'appareil de la CPU pour régler les propriétés horaires (fuseau horaire, activation de l'heure d'été, début de l'heure d'été et fin de l'heure d'été). faute de quoi, WR_LOC_T ne pourra interpréter le passage à l'heure d'été.</p>

Tableau 9- 10Types de données pour les paramètres

Paramètre et type		Type de données	Description
IN	IN	DTL	Heure à régler dans l'horloge système de la CPU
OUT	OUT	DTL	RD_SYS_T : heure système en cours de la CPU RD_LOC_T : heure locale en cours, comprenant le cas échéant tout ajustement lié à l'heure d'été
LOCTIME	IN	DTL	WR_LOC_T : heure locale
DST	IN	BOOL	WR_LOC_T : heure d'été (Daylight Saving Time) évaluée uniquement pendant l'heure "double" lors du passage à l'heure d'été. <ul style="list-style-type: none"> TRUE = heure d'été (première heure) FALSE = heure d'hiver (deuxième heure)
RET_VAL	OUT	Int	Code d'erreur d'exécution

9.1 Fonctions date, heure et horloge

- L'heure locale se calcule à l'aide des décalages de fuseau horaire et d'heure d'été que vous avez paramétrés dans l'onglet général "Heure" de la configuration d'appareil.
- La configuration du fuseau horaire constitue un décalage par rapport à l'heure UTC ou GMT.
- La configuration de l'heure d'été précise le mois, la semaine, le jour et l'heure auxquels l'heure d'été entre en vigueur.
- La configuration de l'heure standard précise également le mois, la semaine, le jour et l'heure auxquels l'heure standard entre en vigueur.
- Le décalage de fuseau horaire est toujours appliqué à la valeur de l'heure système. Le décalage d'heure d'été n'est appliqué que lorsque l'heure d'été est en vigueur.

Remarque

Configuration du passage à l'heure d'été et à l'heure d'hiver

La propriété "Heure" pour "Début de l'heure d'été" de la configuration d'appareil de la CPU doit être votre heure locale.

Codes d'erreur : ENO = 1 signifie qu'aucune erreur ne s'est produite. ENO = 0 signifie qu'une erreur d'exécution s'est produite et un code d'erreur est fourni par la valeur RET_VAL.

RET_VAL (W#16#....)	Description
0000	L'heure locale en cours est l'heure d'hiver.
0001	L'heure d'été a été configurée et l'heure locale en cours est l'heure d'été.
8080	Heure locale non disponible ou valeur LOCTIME non valable.
8081	Valeur de l'année incorrecte ou valeur temporelle indiquée au paramètre LOCTIME non valable
8082	Valeur non valable pour l'indication du mois (octet 2 au format DTL)
8083	Valeur non valable pour l'indication du jour (octet 3 au format DTL)
8084	Valeur non valable pour l'indication de l'heure (octet 5 au format DTL)
8085	Valeur non valable pour l'indication de la minute (octet 6 au format DTL)
8086	Valeur non valable pour l'indication de la seconde (octet 7 au format DTL)
8087	Valeur non valable pour l'indication de la nanoseconde (octets 8 à 11 au format DTL)
8089	Valeur temporelle inexistante (heure écoulée lors du passage à l'heure d'été)
80B0	Défaillance de l'horloge temps réel
80B1	La structure "TimeTransformationRule" n'a pas été définie.

9.1.3 Structure de données TimeTransformationRule

Description

Les règles de changement d'heure entre heure d'hiver et heure d'été sont définies dans la structure TimeTransformationRule. La structure se présente comme suit :

Nom	Type de données	Description
TimeTransformationRule	STRUCT	
Bias	INT	Différence de temps entre l'heure locale et l'heure UTC [minutes] Plage : -1439 à 1439
DaylightBias	INT	Différence de temps entre l'heure d'été et l'heure d'hiver [minutes] Plage : 0 à 60
DaylightStartMonth	USINT	Mois de passage à l'heure d'été Plage : 1 à 12
DaylightStartWeek	USINT	Semaine de passage à l'heure d'été 1 = Première occurrence du jour de la semaine dans le mois, ..., 5 = Dernière occurrence du jour de la semaine dans le mois
DaylightStartWeekday	USINT	Jour de passage à l'heure d'été : 1 = dimanche
DaylightStartHour	USINT	Heure de passage à l'heure d'été : Plage : 0 à 23
DaylightStartMinute	USINT	Minute de passage à l'heure d'été Plage : 0 à 59
StandardStartMonth	USINT	Mois de passage à l'heure d'hiver Plage : 1 à 12
StandardStartWeek	USINT	Semaine de passage à l'heure d'hiver 1 = Première occurrence du jour de la semaine dans le mois, ..., 5 = Dernière occurrence du jour de la semaine dans le mois
StandardStartWeekday	USINT	Jour de passage à l'heure d'hiver : 1 = dimanche
StandardStartHour	USINT	Heure de passage à l'heure d'hiver Plage : 0 à 23
StandardStartMinute	USINT	Minute de passage à l'heure d'hiver Plage : 0 à 59
TimeZoneName	STRING[80]	Nom du fuseau : "(GMT+01:00) Amsterdam, Berlin, Berne, Rome, Stockholm, Vienne"

9.1.4 SET_TIMEZONE (Sélectionner le fuseau horaire)

Tableau 9- 11 Instruction SET_TIMEZONE

CONT/LOG	SCL	Description
	<pre>"SET_TIMEZONE_DB" (REQ:=_bool_in, Timezone:=_struct_in, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	Définit les paramètres de fuseau horaire local et d'heure d'été utilisés pour convertir l'heure système CPU en heure locale.

¹ Dans l'exemple SCL, "SET_TIMEZONE_DB" est le nom du DB d'instance.

Tableau 9- 12 Types de données pour les paramètres

Paramètre et type	Type de données	Description
REQ IN	Bool	REQ=1 : la fonction est exécutée
Timezone IN	TimeTransformationRule	Règles pour la conversion de l'heure système en heure locale
DONE OUT	Bool	Fonction achevée
BUSY OUT	Bool	Fonction occupée
ERROR OUT	Bool	Erreur détectée
STATUS OUT	Word	Résultat de la fonction / message d'erreur

Pour configurer manuellement les paramètres de fuseau horaire pour la CPU, utilisez les propriétés "Heure" de l'onglet "Général" de la configuration d'appareil.

Utilisez l'instruction SET_TIMEZONE pour configurer l'heure locale par programme. Les paramètres de la structure "TimeTransformationRule (Page 353)" définissent le fuseau horaire local et les informations de temps pour la commutation automatique entre heure d'hiver et heure d'été.

Codes d'erreur : ENO = 1 signifie qu'aucune erreur ne s'est produite. ENO = 0 signifie qu'une erreur d'exécution s'est produite et un code d'erreur est fourni dans la sortie STATUS.

STATUS (W#16#...)	Description
0	Pas d'erreur
7000	Pas de traitement de tâche actif
7001	Début du traitement de la tâche. Paramètre BUSY = 1, DONE = 0
7002	7002 Appel intermédiaire (REQ non significatif) : instruction déjà active ; BUSY a la valeur "1".
808x	Erreur au niveau du x-ième élément : 8084 indique, par exemple, que DaylightStartWeekif n'a pas une valeur comprise entre 1 et 5.

9.1.5 RTM (Compteurs d'heures de fonctionnement)

Tableau 9- 13 Instruction RTM

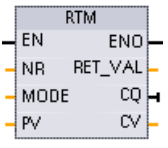
CONT/LOG	SCL	Description
	<pre>RTM(NR:=_uint_in_, MODE:=_byte_in_, PV:=_dint_in_, CQ=>_bool_out_, CV=>_dint_out_);</pre>	L'instruction RTM (Compteurs d'heures de fonctionnement) permet de régler, de démarrer, d'arrêter et de lire les compteurs d'heures de fonctionnement dans la CPU.

Tableau 9- 14 Types de données pour les paramètres

Paramètre et type		Type de données	Description
NR	IN	UInt	Numéro du compteur d'heures de fonctionnement : (valeurs possibles : 0..9)
MODE	IN	Byte	Numéro de mode d'exécution RTM : <ul style="list-style-type: none"> 0 = Lire les valeurs (l'état est alors écrit dans CQ et la valeur en cours dans CV) 1 = Démarrer (à la dernière valeur du compteur) 2 = Arrêter 4 = Régler (à la valeur indiquée dans PV) 5 = Régler (à la valeur indiquée dans PV) puis démarrer 6 = Régler (à la valeur indiquée dans PV) puis arrêter 7 = Sauvegarder toutes les valeurs RTM de la CPU dans la carte mémoire
PV	IN	DInt	Valeur d'heures prédéfinie pour le compteur d'heures de fonctionnement indiqué
RET_VAL	OUT	Int	Résultat de la fonction / message d'erreur
CQ	OUT	Bool	Etat du compteur d'heures de fonctionnement (1 = en cours d'exécution)
CV	OUT	DInt	Valeur d'heures de fonctionnement en cours pour le compteur indiqué

La CPU exploite jusqu'à dix compteurs d'heures de fonctionnement pour suivre les heures de fonctionnement des sous-systèmes de commande critiques. Vous devez démarrer les compteurs d'heures individuels avec une instruction RTM pour chaque compteur. Tous les compteurs d'heures de fonctionnement s'arrêtent lorsque la CPU passe de l'état MARCHE à l'état ARRÊT. Vous pouvez également arrêter des compteurs d'heures individuellement à l'aide du mode d'exécution 2 de RTM.

Lorsqu'une CPU passe de l'état ARRÊT à l'état MARCHE, vous devez redémarrer les compteurs d'heures avec une instruction RTM pour chaque compteur démarré. Lorsque la valeur d'un compteur d'heures de fonctionnement est supérieure à 2147483647 heures, le comptage s'arrête et l'erreur "Débordement" est émise. Vous devez exécuter l'instruction RTM une fois pour chaque compteur d'heures que vous voulez réinitialiser ou modifier.

9.1 Fonctions date, heure et horloge

Une coupure de courant ou une mise hors tension de la CPU déclenche une procédure d'arrêt qui enregistre les valeurs en cours des compteurs d'heures de fonctionnement en mémoire rémanente. A la remise sous tension de la CPU, les valeurs des compteurs d'heures de fonctionnement sauvegardées sont rechargées dans les compteurs d'heures sans perte des totaux d'heures de fonctionnement précédents. Les compteurs d'heures de fonctionnement doivent être redémarrés pour comptabiliser de nouvelles heures de fonctionnement.

Vous pouvez également utiliser le mode d'exécution 7 de RTM dans votre programme pour sauvegarder les valeurs des compteurs d'heures de fonctionnement en carte mémoire. L'état de tous les compteurs d'heures à l'instant où le mode 7 de RTM est exécuté est alors stocké dans la carte mémoire. Ces valeurs enregistrées peuvent devenir incorrectes avec le temps du fait des démarrages et arrêts éventuels des compteurs d'heures pendant une session d'exécution du programme. Vous devez périodiquement actualiser les valeurs en carte mémoire pour capturer les événements de fonctionnement importants. L'avantage de stocker les valeurs RTM en carte mémoire est que vous pouvez insérer la carte mémoire dans une CPU de remplacement où votre programme est les valeurs RTM sauvegardées seront alors disponibles. Si vous n'aviez pas sauvegardé les valeurs RTM dans la carte mémoire, elles seraient perdues (dans une CPU de remplacement).

Remarque

Évitez les appels excessifs d'opérations d'écriture en carte mémoire dans votre programme

Réduisez les opérations d'écriture en carte mémoire flash au minimum pour allonger la durée de vie de la carte mémoire.

Tableau 9- 15 Codes d'erreur

RET_VAL (W#16#....)	Description
0	Pas d'erreur
8080	Numéro du compteur d'heures de fonctionnement incorrect
8081	Une valeur négative a été transmise au paramètre PV.
8082	Débordement du compteur d'heures de fonctionnement
8091	Le paramètre d'entrée MODE contient une valeur incorrecte.
80B1	Impossible de sauvegarder les valeurs en carte mémoire (MODE = 7)

9.2 Chaînes de caractères et caractères

9.2.1 Présentation des données chaînes de caractères (type String)

Type de données STRING

Les données chaînes de caractères (de type String) sont stockées sous forme d'un en-tête de deux octets suivi de 254 octets au maximum de codes de caractères ASCII. Un en-tête de chaîne de caractères contient deux longueurs. Le premier octet correspond à la longueur maximale qui est indiquée entre crochets lorsque vous initialisez une chaîne (254 par défaut). Le deuxième octet de l'en-tête est la longueur en cours qui est le nombre de caractères valides dans la chaîne. La longueur en cours doit être inférieure ou égale à la longueur maximale. Le nombre d'octets occupés par le format String est supérieur de 2 octets à la longueur maximale.

Initialisation des données String

Les données d'entrée et de sortie String doivent être initialisées en tant que chaînes valides en mémoire avant exécution de toute instruction sur chaîne.

Données String valides

Une chaîne valide a une longueur maximale qui doit être supérieure à zéro mais inférieure à 255. La longueur en cours doit être inférieure ou égale à la longueur maximale.

Il n'est pas possible d'affecter des chaînes aux zones de mémoire I ou Q.

Pour plus d'informations, reportez-vous à Format du type de données String (Page 135).

9.2.2 S_MOVE (Déplacer la chaîne de caractères)

Tableau 9- 16 Instruction Déplacer la chaîne de caractères

CONT/LOG	SCL	Description
	<pre>out := in;</pre>	Copier la chaîne de caractères source IN dans la destination OUT. L'exécution de S_MOVE n'affecte pas le contenu de la chaîne source.

Tableau 9- 17 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	String	Chaîne de caractères source
OUT	String	Adresse cible

Si la longueur effective de la chaîne dans l'entrée IN dépasse la longueur maximale d'une chaîne stockée dans la sortie OUT, seule la partie de la chaîne IN qui tient dans la chaîne OUT est copiée.

9.2.3 Instructions de conversion de chaîne

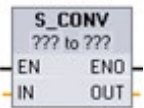
9.2.3.1 Instructions S_CONV (Convertir la chaîne de caractères), STRG_VAL (Convertir la chaîne de caractères en valeur numérique) et VAL_STRG (Convertir la valeur numérique en chaîne de caractères)

Vous pouvez convertir des chaînes de caractères numériques en valeurs numériques ou des valeurs numériques en chaînes de caractères numériques à l'aide des instructions suivantes :

- S_CONV convertit une chaîne numérique en valeur numérique ou bien une valeur numérique en chaîne numérique.
- STRG_VAL convertit une chaîne numérique en valeur numérique avec des options de format.
- VAL_STRG convertit une valeur numérique en chaîne numérique avec des options de format.

S_CONV (Convertir la chaîne de caractères)

Tableau 9- 18 Instruction de conversion de chaîne

CONT/LOG	SCL	Description
	<pre>out := <Type>_TO_<Type>(in);</pre>	Convertit une chaîne de caractères en sa valeur correspondante ou une valeur en sa chaîne de caractères correspondante. L'instruction S_CONV ne comporte pas d'options de formatage de la sortie. Cela rend S_CONV plus simple à utiliser mais moins souple que les instructions STRG_VAL et VAL_STRG.

- 1 Pour CONT / LOG : Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.
- 2 Pour SCL : Sélectionnez S_CONV dans les instructions avancées et répondez aux messages concernant les types de données pour la conversion. STEP 7 propose alors l'instruction de conversion appropriée.

Tableau 9- 19 Types de données (chaîne en valeur)

Paramètre et type		Type de données	Description
IN	IN	String, WString	Chaîne de caractères d'entrée
OUT	OUT	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valeur numérique de sortie

La conversion du paramètre chaîne IN commence au premier caractère et continue jusqu'à la fin de la chaîne ou jusqu'au premier caractère rencontré qui n'est pas "0" à "9", "+", "-" ou ".". La valeur résultante est fournie à l'adresse indiquée dans le paramètre OUT. Si la valeur numérique en sortie ne tient pas dans la plage du type de données de OUT, le paramètre OUT est mis à 0 et ENO est défini à FAUX. Sinon, le paramètre OUT contient un résultat valide et ENO est défini à VRAI.

Règles de format concernant la chaîne d'entrée :

- Si un séparateur décimal est utilisé dans la chaîne IN, vous devez utiliser le caractère ".".
- Les virgules "," utilisées comme séparateurs des milliers à gauche du séparateur décimal sont autorisées mais ne sont pas prises en compte.
- Les espaces en tête ne sont pas pris en compte.

S_CONV (Conversion valeur en chaîne)

Tableau 9- 20 Types de données (valeur en chaîne)

Paramètre et type		Type de données	Description
IN	IN	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Valeur numérique d'entrée
OUT	OUT	String, WString	Chaîne de caractères de sortie

Un paramètre IN entier, entier non signé ou nombre à virgule flottante est converti en sa chaîne de caractères correspondante dans OUT. Le paramètre OUT doit référencer une chaîne valide pour que la conversion soit exécutée. Une chaîne valide comprend la longueur de chaîne maximale dans le premier octet, la longueur de chaîne en cours dans le deuxième octet et les caractères de la chaîne en cours dans les octets suivants. La chaîne convertie remplace les caractères dans la chaîne OUT en commençant au premier caractère et l'octet de longueur en cours de la chaîne OUT est ajusté. L'octet de longueur maximale de la chaîne OUT n'est pas modifié.

Le nombre de caractères remplacés dépend du type de données et de la valeur numérique du paramètre IN. Le nombre de caractères remplacés doit tenir dans la longueur de la chaîne du paramètre OUT. La longueur de chaîne maximale (premier octet) de la chaîne OUT doit être supérieure ou égale au nombre maximum attendu de caractères convertis. Le tableau ci-après montre des exemples de conversion valeur en chaîne S_CONV.

Règles de format concernant la chaîne de sortie :

- Les valeurs écrites dans le paramètre OUT n'utilisent pas de signe "+" en tête.
- La représentation en virgule fixe est utilisée (et non la notation exponentielle).
- Le point "." est utilisé pour représenter le séparateur décimal lorsque le paramètre IN est de type de données Real.
- Les valeurs sont alignées à droite dans la chaîne de sortie et précédées d'espaces qui complètent les positions de caractère vides.


Tableau 9- 21 Longueurs maximales des chaînes de caractères pour chaque type de données

Type de données de IN	Positions de caractère allouées par S_CONV	Exemple de chaîne convertie ¹	Longueur de chaîne totale incluant les octets de longueur maximale et en cours
USInt	4	"x255"	6
SInt	4	"-128"	6
UInt	6	"x65535"	8
Int	6	"-32768"	8
UDInt	11	"x4294967295"	13
DInt	11	"-2147483648"	13
Real	14	"x-3.402823E+38" "x-1.175495E-38" "x+1.175495E-38" "x+3.402823E+38"	16
LReal	21	"-1.7976931348623E+308" "-2.2250738585072E-308" "+2.2250738585072E-308" "+1.7976931348623E+308"	23

¹ Les caractères "x" représentent des espaces qui complètent les positions vides du champ aligné à droite alloués pour la valeur convertie.

STRG_VAL (Convertir la chaîne de caractères en valeur numérique)

Tableau 9- 22 Instruction Chaîne en valeur

CONT/LOG	SCL	Description
	<pre>"STRG_VAL" (in:=_string_in, format:=_word_in, p:=uint_in, out=>_variant_out);</pre>	Convertit une chaîne de caractères numérique en sa représentation entière ou virgule flottante correspondante.

¹ Pour CONT / LOG : Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.

Tableau 9- 23 Types de données pour l'instruction STRG_VAL

Paramètre et type		Type de données	Description
IN	IN	String, WString	Chaîne de caractères ASCII à convertir
FORMAT	IN	Word	Options de format de sortie
P	IN	UInt, Byte, USInt	IN : Indice pointant sur le premier caractère à convertir (premier caractère = 1)
OUT	OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valeur numérique convertie

La conversion commence dans la chaîne IN au décalage de caractère P et continue jusqu'à la fin de la chaîne ou jusqu'au premier caractère rencontré qui n'est pas "+", "-", ".", ",", "e", "E" ou "0" à "9". Le résultat est placé à l'adresse indiquée dans le paramètre OUT.

Les données String doivent être initialisées avant l'exécution en tant que chaîne valide en mémoire.

Le paramètre FORMAT pour l'instruction STRG_VAL est défini ci-dessous. Les positions de bit inutilisées doivent être mises à zéro.

Tableau 9- 24 Format de l'instruction STRG_VAL

Bit 16							Bit 8	Bit 7								Bit 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f	r

f = format de notation
 1 = notation exponentielle
 0 = notation à virgule fixe

r = format du séparateur décimal
 1 = "," (virgule)
 0 = "." (point)

Tableau 9- 25 Valeurs du paramètre FORMAT


FORMAT (W#16#)	Format de notation	Représentation du séparateur décimal
0000 (par défaut)	Virgule fixe	."
0001		","
0002	Exponentielle	."
0003		","
0004 à FFFF	Valeurs illicites	

Règles pour la conversion STRG_VAL :

- Si le point "." est utilisé comme séparateur décimal, les virgules "," à gauche du séparateur décimal sont considérées comme caractères de séparation des milliers. Les virgules sont autorisées mais ne sont pas prises en compte.
- Si la virgule "," est utilisée comme séparateur décimal, les points "." à gauche du séparateur décimal sont considérés comme caractères de séparation des milliers. Ces points sont autorisés mais ne sont pas pris en compte.
- Les espaces en tête ne sont pas pris en compte.

VAL_STRG (Convertir la valeur numérique en chaîne de caractères)

Tableau 9- 26 Instruction Valeur en chaîne

CONT/LOG	SCL	Description
	<pre>"VAL_STRG" (in:=_variant_in, size:=_usint_in, prec:=_usint_in, format:=_word_in, p:=uint_in, out=>_string_out);</pre>	Convertit une valeur entière, entière non signée ou virgule flottante en sa représentation chaîne de caractères correspondante.

1 Pour CONT / LOG : Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.

Tableau 9- 27 Types de données pour l'instruction VAL_STRG

Paramètre et type		Type de données	Description
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valeur à convertir
SIZE	IN	USInt	Nombre de caractères à écrire dans la chaîne OUT
PREC	IN	USInt	Précision ou taille de la partie fractionnaire. Cela n'inclut pas le séparateur décimal.
FORMAT	IN	Word	Options de format de sortie
P	IN	UInt, Byte, USInt	IN : Indice pointant sur le premier caractère de la chaîne OUT à remplacer (premier caractère = 1)
OUT	OUT	String, WString	Chaîne convertie

Cette instruction convertit la valeur représentée par le paramètre IN en une chaîne référencée par le paramètre OUT. Le paramètre OUT doit être une chaîne valide pour que la conversion soit exécutée.

La chaîne convertie remplace les caractères dans la chaîne OUT en commençant au décalage de caractères P et en allant jusqu'au nombre de caractères indiqué par le paramètre SIZE. Le nombre de caractères dans SIZE doit tenir dans la longueur de la chaîne OUT, le comptage commençant à la position de caractère P. Si le paramètre SIZE est de zéro, les caractères seront écrasés à la position P dans la chaîne OUT sans restriction. Cette instruction est utile pour insérer des nombres dans une chaîne de texte. Vous pouvez, par exemple, insérer le nombre "120" dans la chaîne "Pression pompe = 120 psi".

Le paramètre PREC indique la précision ou le nombre de chiffres pour la partie fractionnaire de la chaîne. Si la valeur du paramètre IN est un entier, PREC indique la position du séparateur décimal. Par exemple, si la valeur de données est 123 et que PREC est égal à 1, le résultat sera "12.3". La précision maximale acceptée pour le type de données Real est de 7 chiffres.

Si le paramètre P est supérieur à la taille en cours de la chaîne OUT, des espaces sont ajoutés jusqu'à la position P et le résultat est ajouté à la fin de la chaîne. La conversion s'arrête si la longueur maximale de la chaîne OUT est atteinte.

Le paramètre FORMAT pour l'instruction VAL_STRG est défini ci-dessous. Les positions de bit inutilisées doivent être mises à zéro.

Tableau 9- 28 Format de l'instruction VAL_STRG

Bit 16								Bit 8	Bit 7								Bit 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	s	f	r

- | | |
|----------------------------------|---|
| s = signe du nombre | 1= utiliser les caractères "+" et "-" pour le signe
0 = utiliser le caractère "-" uniquement |
| f = format de notation | 1= notation exponentielle
0 = notation à virgule fixe |
| r = format du séparateur décimal | 1 = "," (virgule)
0 = "." (point) |

Tableau 9- 29 Valeurs du paramètre FORMAT

FORMAT (WORD)	Signe du nombre	Format de notation	Représentation du séparateur décimal
W#16#0000	"-" uniquement	Virgule fixe	" ."
W#16#0001			" ,"
W#16#0002		Exponentielle	" ."
W#16#0003	" ,"		
W#16#0004	"+" et "-"	Virgule fixe	" ."
W#16#0005			" ,"
W#16#0006		Exponentielle	" ."
W#16#0007			" ,"
W#16#0008 à W#16#FFFF	Valeurs illicites		

Règles de format concernant la chaîne OUT :

- Des espaces sont ajoutés à l'extrême gauche de la chaîne lorsque la chaîne convertie est inférieure à la taille précisée.
- Lorsque le bit de signe du paramètre FORMAT est FAUX, les valeurs de type de données entier non signé et entier signé sont écrites dans la mémoire tampon de sortie sans être précédées du signe "+". Le signe "-" est utilisé si nécessaire.
<espaces en tête><chiffres sans zéros en tête>'<chiffres PREC>
- Lorsque le bit de signe est VRAI, les valeurs de type de données entier non signé et entier signé sont toujours écrites dans la mémoire tampon de sortie avec un caractère de signe en tête.
<espaces en tête><signe><chiffres sans zéros en tête>'<chiffres PREC>

9.2 Chaînes de caractères et caractères

- Lorsque la notation exponentielle est paramétrée dans FORMAT, les valeurs de type de données Real sont écrites comme suit dans la mémoire tampon de sortie :
<espaces en tête><signe><chiffre> '.' <chiffres PREC>'E' <signe><chiffres sans zéro en tête>
- Lorsque la notation en virgule fixe est paramétrée dans FORMAT, les valeurs de type de données entier, entier non signé et réel sont écrites comme suit dans la mémoire tampon de sortie :
<espaces en tête><signe><chiffres sans zéros en tête>.'<chiffres PREC>
- Les zéros en tête à gauche du séparateur décimal sont supprimés (excepté le chiffre à côté du séparateur décimal).
- Les valeurs à droite du séparateur décimal sont arrondies pour correspondre au nombre de chiffres à droite du séparateur décimal indiqué par le paramètre PREC.
- La taille de la chaîne de sortie doit être au minimum de trois octets supérieure au nombre de chiffres à droite du séparateur décimal.
- Les valeurs sont alignées à droite dans la chaîne de sortie.

Situations d'erreur signalées par ENO

Lorsque l'opération de conversion rencontre une erreur, l'instruction renvoie les résultats suivants :

- ENO est mis à 0.
- OUT est mis à 0 ou défini comme montré dans les exemples de conversion de chaîne en valeur.
- OUT reste inchangé ou est défini comme montré dans les exemples où OUT est une chaîne.

Tableau 9- 30Etat de ENO

ENO	Description
1	Pas d'erreur
0	Paramètre illicite ou invalide ; par exemple, accès à un DB qui n'existe pas.
0	Chaîne illicite : la longueur maximale de la chaîne est 0 ou 255.
0	Chaîne illicite : la longueur en cours est supérieure à la longueur maximale.
0	La valeur numérique convertie est trop grande pour le type de données spécifié pour OUT.
0	La taille de chaîne maximale du paramètre OUT doit être suffisamment grande pour recevoir le nombre de caractères indiqué par le paramètre SIZE, en commençant à la position de caractère indiquée par le paramètre P.
0	Valeur P illicite : P = 0 ou P est supérieur à la longueur de chaîne en cours.
0	Le paramètre SIZE doit être supérieur au paramètre PREC.

Tableau 9- 31 Exemple de conversion de chaîne en valeur S_CONV

Chaîne IN	Type de données de OUT	Valeur OUT	ENO
"123"	Int ou DInt	123	VRAI
"-00456"	Int ou DInt	-456	VRAI
"123.45"	Int ou DInt	123	VRAI
"+2345"	Int ou DInt	2345	VRAI
"00123AB"	Int ou DInt	123	VRAI
"123"	Real	123.0	VRAI
"123.45"	Real	123.45	VRAI
"1.23e-4"	Real	1.23	VRAI
"1.23E-4"	Real	1.23	VRAI
"12,345.67"	Real	12345.67	VRAI
"3.4e39"	Real	3.4	VRAI
"-3.4e39"	Real	-3.4	VRAI
"1.17549e-38"	Real	1.17549	VRAI
"12345"	SInt	0	FAUX
"A123"	Sans objet	0	FAUX
""	N/D	0	FAUX
"++123"	N/D	0	FAUX
"+-123"	N/D	0	FAUX

Tableau 9- 32 Exemples de conversion de valeur en chaîne S_CONV

Type de données	Valeur IN	Chaîne OUT ¹	ENO
UInt	123	"xxx123"	VRAI
UInt	0	"xxxxx0"	VRAI
UDInt	12345678	"xxx12345678"	VRAI
Real	+9123.456	"xx+9.123456E+3"	VRAI
LReal	+9123.4567890123	"xx+9.1234567890123E+3"	VRAI
Real	-INF	"xxxxxxxxxxxINF"	FAUX
Real	+INF	"xxxxxxxxxxxINF"	FAUX
Real	NaN	"xxxxxxxxxxxNaN"	FAUX

¹ Les caractères "x" représentent des espaces qui complètent les positions vides du champ aligné à droite alloués pour la valeur convertie.

Tableau 9- 33Exemple : Conversion STRG_VAL

Chaîne IN	FORMAT (W#16#....)	Type de données de OUT	Valeur OUT	ENO
"123"	0000	Int ou DInt	123	VRAI
"-00456"	0000	Int ou DInt	-456	VRAI
"123.45"	0000	Int ou DInt	123	VRAI
"+2345"	0000	Int ou DInt	2345	VRAI
"00123AB"	0000	Int ou DInt	123	VRAI
"123"	0000	Real	123.0	VRAI
"-00456"	0001	Real	-456.0	VRAI
"+00456"	0001	Real	456.0	VRAI
"123.45"	0000	Real	123.45	VRAI
"123.45"	0001	Real	12345.0	VRAI
"123.45"	0000	Real	12345.0	VRAI
"123.45"	0001	Real	123.45	VRAI
".00123AB"	0001	Real	123.0	VRAI
"1.23e-4"	0000	Real	1.23	VRAI
"1.23E-4"	0000	Real	1.23	VRAI
"1.23E-4"	0002	Real	1.23E-4	VRAI
"12,345.67"	0000	Real	12345.67	VRAI
"12,345.67"	0001	Real	12.345	VRAI
"3.4e39"	0002	Real	+INF	VRAI
"-3.4e39"	0002	Real	-INF	VRAI
"1.1754943e-38" (et inférieur)	0002	Real	0.0	VRAI
"12345"	N/D	SInt	0	FAUX
"A123"	Sans objet	Sans objet	0	FAUX
""	N/D	N/D	0	FAUX
"++123"	N/D	N/D	0	FAUX
"+-123"	N/D	N/D	0	FAUX

Les exemples suivants de conversions VAL_STRG sont basés sur une chaîne OUT initialisée comme suit :

"Current Temp = xxxxxxxxxxx c"

Le caractère "x" représente des espaces alloués pour la valeur convertie.

Tableau 9- 34 Exemple : Conversion VAL_STRG

Type de données	Valeur IN	P	SIZE	FORMAT (W#16#....)	PREC	Chaîne OUT	ENO
UInt	123	16	10	0000	0	Current Temp = xxxxxxx123 C	VRAI
UInt	0	16	10	0000	2	Current Temp = xxxxxx0.00 C	VRAI
UDInt	12345678	16	10	0000	3	Current Temp = x12345.678 C	VRAI
UDInt	12345678	16	10	0001	3	Current Temp = x12345,678 C	VRAI
Int	123	16	10	0004	0	Current Temp = xxxxxxx+123 C	VRAI
Int	-123	16	10	0004	0	Current Temp = xxxxxx-123 C	VRAI
Real	-0.00123	16	10	0004	4	Current Temp = xxx- 0.0012 C	VRAI
Real	-0.00123	16	10	0006	4	Current Temp = - 1.2300E-3 C	VRAI
Real	-INF	16	10	N/D	4	Current Temp = xxxxxx-INF C	FAUX
Real	+INF	16	10	N/D	4	Current Temp = xxxxxx+INF C	FAUX
Real	NaN	16	10	N/D	4	Current Temp = xxxxxxxNaN C	FAUX
UDInt	12345678	16	6	N/D	3	Current Temp = xxxxxxxxxx C	FAUX

9.2.3.2 Instructions Strg_TO_Chars et Chars_TO_Strg (Convertir de/vers chaîne de caractères et tableau de CHAR)

Strg_TO_Chars copie une chaîne de caractères ASCII dans un tableau d'octets de caractères.

Chars_TO_Strg copie un tableau d'octets de caractères ASCII dans une chaîne de caractères.

Remarque

Seuls les types de tableaux de base zéro (Array [0..n] of Char ou Array [0..n] of Byte) sont autorisés comme paramètre d'entrée Chars pour l'instruction Chars_TO_Strg ou comme paramètre d'entrée/sortie Chars pour l'instruction Strg_TO_Chars .

Tableau 9- 35 Instruction Strg_TO_Chars

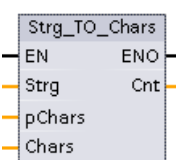
CONT/LOG	SCL	Description
	<pre>Strg_TO_Chars (Strg:=_string_in_, pChars:=_dint_in_, Cnt=>_uint_out_, Chars:=_variant_inout_);</pre>	<p>La chaîne d'entrée complète Strg est copiée dans un tableau de caractères indiqué par le paramètre IN_OUT Chars.</p> <p>L'opération écrase les octets en commençant à l'élément du tableau indiqué par le paramètre pChars.</p> <p>Il est possible d'utiliser des chaînes de toutes les longueurs maximales acceptées (1..254).</p> <p>Aucun délimiteur de fin n'est écrit ; c'est à vous de vous en charger. Pour placer un délimiteur de fin juste après le dernière caractère écrit dans le tableau, utilisez le numéro d'élément de tableau suivant [pChars+Cnt].</p>

Tableau 9- 36 Types de données pour les paramètres (Strg_TO_Chars)

Paramètre et type		Type de données	Description
Strg	IN	String, WString	Chaîne de caractères source
pChars	IN	DInt	Numéro d'élément de tableau pour le premier caractère de la chaîne écrit dans le tableau cible
Chars	IN_OUT	VARIANT	Le paramètre Chars est un pointeur désignant un tableau de caractères de base zéro [0..n] à copier depuis la chaîne d'entrée. Le tableau peut être déclaré dans un DB ou sous forme de variables locales dans l'interface de bloc. Exemple : "DB1".MyArray désigne les éléments de MyArray [0..10] of Char dans le DB1.
Cnt	OUT	UInt	Nombre de caractères copiés

Tableau 9- 37 Instruction Chars_TO_Strg

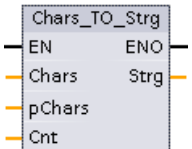
CONT/LOG	SCL	Description
	<pre>Chars_TO_Strg(Chars:=_variant_in_, pChars:=_dint_in_, Cnt:=_uint_in_, Strg=>_string_out_);</pre>	<p>L'ensemble ou une partie d'un tableau de caractères est copié dans une chaîne.</p> <p>La chaîne de sortie doit être déclarée avant l'exécution de Chars_TO_Strg. La chaîne est alors écrasée par l'exécution de l'instruction Chars_TO_Strg.</p> <p>Il est possible d'utiliser des chaînes de toutes les longueurs maximales acceptées (1..254).</p> <p>La longueur de chaîne maximale n'est pas modifiée par l'instruction Chars_TO_Strg . La copie du tableau dans la chaîne s'arrête lorsque la longueur de chaîne maximale est atteinte.</p> <p>Une valeur de caractère nul (\$00) ou 16#00 dans le tableau de caractères constitue un délimiteur qui met fin à la copie de caractères dans la chaîne.</p>

Tableau 9- 38 Types de données pour les paramètres (Chars_TO_Strg)

Paramètre et type		Type de données	Description
Chars	IN	VARIANT	Le paramètre Chars est un pointeur désignant un tableau de caractères de base zéro [0..n] à convertir en chaîne. Le tableau peut être déclaré dans un DB ou sous forme de variables locales dans l'interface de bloc. Exemple : "DB1".MyArray désigne les éléments de MyArray [0..10] of Char dans le DB1.
pChars	IN	DInt	Numéro d'élément du premier caractère du tableau à copier. L'élément [0] du tableau est pris par défaut.
Cnt	IN	UInt	Nombre de caractères à copier ; 0 signifie tous les caractères.
Strg	OUT	String, WString	Chaîne cible

Tableau 9- 39 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Chars_TO_Strg : tentative de copie dans la chaîne de sortie d'un nombre d'octets de caractères supérieur au nombre autorisé par l'octet de longueur maximale dans la déclaration de la chaîne
0	Chars_TO_Strg : valeur caractère nul (16#00) trouvée dans le tableau d'entrée d'octets de caractères
0	Strg_TO_Chars : tentative de copie dans le tableau de sortie d'un nombre d'octets de caractères supérieur au nombre autorisé par la limite de numéro d'élément

9.2.3.3 Instructions ATH et HTA (Convertir de/vers chaîne de caractères ASCII et nombre hexadécimal)

Utilisez les instructions ATH (ASCII en hexadécimal) et HTA (Hexadécimal en ASCII) pour effectuer des conversions entre octets de caractères ASCII (caractères 0 à 9 et majuscules A à F uniquement) et les quartets hexadécimaux correspondants.

Tableau 9- 40 Instruction ATH

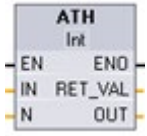
CONT/LOG	SCL	Description
	<pre>ret_val := ATH(in:=_variant_in_, n:=_int_in_, out=>_variant_out_);</pre>	Convertit des caractères ASCII en chiffres hexadécimaux condensés.

Tableau 9- 41 Types de données pour l'instruction ATH

Type de paramètre	Type de données	Description	
IN	IN	Variant	Pointeur désignant le tableau d'octets de caractères ASCII
N	IN	UInt	Nombre d'octets de caractères ASCII à convertir
RET_VAL	OUT	Word	Code d'erreur d'exécution
OUT	OUT	Variant	Pointeur désignant le tableau d'octets hexadécimaux convertis

La conversion commence à l'emplacement indiqué par le paramètre IN et se poursuit sur N octets. Le résultat est placé à l'emplacement indiqué par OUT. Seuls les caractères ASCII valides 0 à 9, a à f minuscules et A à F majuscules peuvent être convertis. Tout autre caractère est converti en un zéro.

Les caractères codés ASCII de 8 bits sont convertis en quartets hexadécimaux. Deux caractères ASCII peuvent être convertis en un octet unique contenant deux quartets hexadécimaux.

9.2 Chaînes de caractères et caractères

Les paramètres IN et OUT correspondent à des tableaux d'octets et non à des données de type String hexadécimales. Les caractères ASCII sont convertis et placés dans la sortie hexadécimale dans l'ordre où ils sont lus. S'il y a un nombre impair de caractères ASCII, le quartet de droite du dernier chiffre hexadécimal converti est complété avec des zéros.

Tableau 9- 42Exemples : Conversion d'ASCII en hexadécimal (ATH)

Octets caractères dans IN	N	Valeur OUT	ENO
'0a23'	4	W#16#0A23	VRAI
'123AFx1a23'	10	16#123AF01023	FAUX
'a23'	3	W#16#A230	VRAI

Tableau 9- 43Instruction HTA

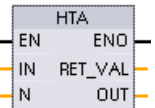
CONT/LOG	SCL	Description
	<pre>ret_val := HTA(in:=_variant_in_, n:=_uint_in_, out=>_variant_out_);</pre>	Convertit des chiffres hexadécimaux condensés en octets de caractères ASCII correspondants.

Tableau 9- 44 Types de données pour l'instruction HTA

Paramètre et type	Type de données	Description	
IN	IN	Variant	Pointeur désignant le tableau d'octets d'entrée
N	IN	UInt	Nombre d'octets à convertir (chaque octet d'entrée comporte deux quartets ce qui produit 2N caractères ASCII)
RET_VAL	OUT	Word	Code d'erreur d'exécution
OUT	OUT	Variant	Pointeur désignant le tableau d'octets de caractères ASCII

La conversion commence à l'emplacement indiqué par le paramètre IN et se poursuit sur N octets. Chaque quartet est converti en un unique caractère ASCII de 8 bits, ce qui produit 2N octets de caractères ASCII de sortie. Les 2N octets de la sortie sont écrits sous forme de caractères ASCII de 0 à 9 et A à F majuscules. Le paramètre OUT correspond à un tableau d'octets et non à une chaîne de caractères.

Les quartets des octets hexadécimaux sont convertis en caractères dans l'ordre où ils sont lus (le quartet de gauche d'un octet hexadécimal est converti le premier, suivi par le quartet de droite de ce même octet).

Tableau 9- 45Exemples : Conversion d'hexadécimal en ASCII (HTA)

Valeur IN	N	Octets caractères dans OUT	ENO (ENO est toujours VRAI après l'exécution de HTA)
W#16#0123	2	'0123'	VRAI
DW#16#123AF012	4	'123AF012'	VRAI

Tableau 9- 46 Codes d'erreur de ATH and HTA

RET_VAL (W#16#...)	Description	ENO
0000	Pas d'erreur	VRAI
0007	Caractère d'entrée ATH invalide : Un caractère qui n'était pas un caractère ASCII 0 à 9, a à f minuscule ou A à F majuscule a été trouvé.	FAUX
8101	Pointeur d'entrée illicite ou invalide ; par exemple, accès à un DB qui n'existe pas.	FAUX
8120	La chaîne d'entrée a un format invalide, c'est-à-dire max=0, max=255, valeur en cours>max ou longueur allouée dans le pointeur < max.	FAUX
8182	Mémoire tampon d'entrée trop petite pour N	FAUX
8151	Type de données interdit pour la mémoire tampon d'entrée	FAUX
8301	Pointeur de sortie illicite ou invalide ; par exemple, accès à un DB qui n'existe pas.	FAUX
8320	La chaîne de sortie a un format invalide, c'est-à-dire max=0, max=255, valeur en cours>max ou longueur allouée dans le pointeur < max.	FAUX
8382	Mémoire tampon de sortie trop petite pour N	FAUX
8351	Type de données interdit pour la mémoire tampon de sortie	FAUX

9.2.4 Instructions sur chaîne

Votre programme de commande peut utiliser les instructions sur chaîne et sur caractère suivantes pour créer des messages destinés aux afficheurs et aux journaux du processus.

9.2.4.1 MAX_LEN (Déterminer la longueur maximale d'une chaîne de caractères)

Tableau 9- 47 Instruction longueur maximale

CONT/LOG	SCL	Description
	<pre>out := MAX_LEN(in);</pre>	<p>MAX_LEN (Longueur de chaîne maximale) fournit la valeur de longueur maximale affectée à la chaîne IN sur la sortie OUT. Si des erreurs se produisent pendant le traitement de l'instruction, une chaîne vide est transmise.</p> <p>Les types de données String et WString contiennent deux longueurs : le premier octet (ou mot) indique la longueur de chaîne maximale et le deuxième octet (ou mot) indique la longueur en cours (correspond au nombre actuel de caractères valides).</p> <ul style="list-style-type: none"> La longueur maximale de la chaîne de caractères est indiquée entre crochets pour chaque instruction String ou WString. Le nombre d'octets occupés par une String est supérieur de 2 octets à la longueur maximale. Le nombre de mots occupés par une WString est supérieur de 2 mots à la longueur maximale. La longueur en cours représente le nombre de caractères réellement utilisés. La longueur en cours doit être inférieure ou égale à la longueur maximale. La longueur réelle est en octets pour une String et en mots pour une WString. <p>Utilisez l'instruction MAX_LEN i pour obtenir la longueur maximale de la chaîne de caractères et l'instruction LEN pour obtenir la longueur en cours d'une chaîne.</p>

Tableau 9- 48 Types de données pour les paramètres

Paramètre et type		Type de données	Description
IN	IN	String, WString	Chaîne d'entrée
OUT	OUT	DInt	Nombre maximum de caractères autorisés dans la chaîne IN

9.2.4.2 LEN (Déterminer la longueur d'une chaîne de caractères)

Tableau 9- 49 Instruction Longueur


CONT/LOG	SCL	Description
	<pre>out := LEN(in);</pre>	LEN (Longueur) transmet la longueur en cours de la chaîne IN à la sortie OUT. Une chaîne vide a une longueur de zéro.

Tableau 9- 50 Types de données pour les paramètres

Paramètre et type		Type de données	Description
IN	IN	String, WString	Chaîne d'entrée
OUT	OUT	Int, DInt, Real, LReal	Nombre de caractères valides dans la chaîne d'entrée IN

Tableau 9- 51 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas de chaîne invalide	Longueur de chaîne valide
0	La longueur en cours de IN dépasse la longueur maximale de IN.	La longueur en cours est mise à 0.
	La longueur maximale de IN n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN est 255 (longueur interdite).	

9.2.4.3 CONCAT (Concaténer les chaînes de caractères)

Tableau 9- 52 Instruction Concaténer des chaînes

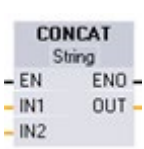
CONT/LOG	SCL	Description
	<pre>out := CONCAT(in1, in2);</pre>	CONCAT (Concaténer des chaînes) combine les chaînes IN1 et IN2 pour former une nouvelle chaîne dans OUT. Après la concaténation, la chaîne IN1 est la partie gauche et la chaîne IN2 la partie droite de la chaîne combinée.

Tableau 9- 53 Types de données pour les paramètres

Paramètre et type		Type de données	Description
IN1	IN	String, WString	Chaîne d'entrée 1
IN2	IN	String, WString	Chaîne d'entrée 2
OUT	OUT	String, WString	Chaîne combinée (chaîne 1 + chaîne 2)

Tableau 9- 54 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	La chaîne résultante après concaténation est plus grande que la longueur maximale de la chaîne OUT.	Les caractères de la chaîne résultante sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1, la longueur en cours de IN2 dépasse la longueur maximale de IN2 ou la longueur en cours de OUT dépasse la longueur maximale de OUT (chaîne invalide).	La longueur en cours est mise à 0.
	La longueur maximale de IN1, IN2 ou OUT n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN1 ou IN2 est 255, ou la longueur maximale de OUT est 0 ou 255 (type de données String)	
	La longueur maximale de IN1 ou IN2 est 65534, ou la longueur maximale de OUT est 0 ou 65534 (type de données WString)	

9.2.4.4 Instructions LEFT, RIGHT et MID (Lire sous-chaînes dans une chaîne de caractères)

Tableau 9- 55 Instructions Lire les caractères de gauche d'une chaîne, Lire les caractères de droite d'une chaîne et Lire les caractères du milieu d'une chaîne

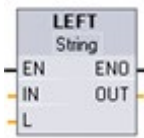
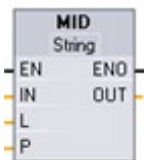
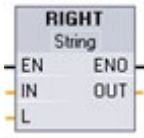
CONT/LOG	SCL	Description
	<pre>out := LEFT(in, L);</pre>	<p>LEFT (Lire les caractères de gauche d'une chaîne) renvoie une sous-chaîne constituée des L premiers caractères du paramètre chaîne IN.</p> <ul style="list-style-type: none"> • Si L est supérieur à la longueur en cours de la chaîne IN, la chaîne IN entière est renvoyée dans OUT. • Si l'entrée est une chaîne vide, une chaîne vide est renvoyée dans OUT.
	<pre>out := MID(in, L, p);</pre>	<p>MID (Lire les caractères du milieu d'une chaîne) renvoie la partie médiane d'une chaîne. La sous-chaîne centrale fait L caractères de long et commence à la position de caractère P (inclusive).</p> <p>Si la somme de L et de P dépasse la longueur en cours de la chaîne IN, une sous-chaîne commençant à la position de caractère P et allant jusqu'à la fin de la chaîne IN est renvoyée.</p>
	<pre>out := RIGHT(in, L);</pre>	<p>RIGHT (Lire les caractères de droite d'une chaîne) renvoie les L derniers caractères d'une chaîne.</p> <ul style="list-style-type: none"> • Si L est supérieur à la longueur en cours de la chaîne IN, la chaîne IN entière est renvoyée dans le paramètre OUT. • Si l'entrée est une chaîne vide, une chaîne vide est renvoyée dans OUT.

Tableau 9- 56 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN	IN	String, WString
L	IN	Int
P	IN	Int
OUT	OUT	String, WString

Chaîne d'entrée

Longueur de la sous-chaîne à créer :

- LEFT utilise le nombre de caractères les plus à gauche dans la chaîne.
- RIGHT utilise le nombre de caractères les plus à droite dans la chaîne.
- MID utilise le nombre de caractères à partir de la position P dans la chaîne.

MID uniquement : Position du premier caractère de la sous-chaîne à copier
P= 1 pour la position de caractère de début de la chaîne IN

Chaîne de sortie

Tableau 9- 57 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	<ul style="list-style-type: none"> L ou P est inférieur ou égal à 0. P est supérieur à la longueur maximale de IN. La longueur en cours de IN dépasse la longueur maximale de IN ou la longueur en cours de OUT dépasse la longueur maximale de OUT. La longueur maximale de IN ou OUT n'entre pas dans la zone de mémoire allouée. La longueur maximale de IN ou OUT est 0 ou 255 (type de données String) ou 0 ou 65534 (type de données WString) 	La longueur en cours est mise à 0.
	La longueur (L) de la sous-chaîne à copier est supérieure à la longueur maximale de la chaîne OUT.	Les caractères sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	MID uniquement : L ou P est inférieur ou égal à 0.	La longueur en cours est mise à 0.
	MID uniquement : P est supérieur à la longueur maximale de IN.	La longueur en cours est mise à 0.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1 ou la longueur en cours de IN2 dépasse la longueur maximale de IN2 (chaîne invalide).	La longueur en cours est mise à 0.
	La longueur maximale de IN1, IN2 ou OUT n'entre pas dans la zone de mémoire allouée.	
La longueur maximale de IN1, IN2 ou OUT est une longueur interdite : 0 ou 255 (type de données String) ou 0 ou 65534 (type de données WString)		

9.2.4.5 DELETE (Supprimer des caractères dans la chaîne de caractères)

Tableau 9- 58 Instruction Supprimer des caractères dans une chaîne


CONT/LOG	SCL	Description
	<pre>out := DELETE(in, L, p);</pre>	<p>Supprime L caractères dans la chaîne IN. La suppression des caractères commence à la position P (inclusive) et la sous-chaîne restante est renvoyée dans le paramètre OUT.</p> <ul style="list-style-type: none"> Si L est égal à zéro, la chaîne d'entrée est renvoyée dans OUT. Si la somme de L et de P est supérieure à la longueur de la chaîne d'entrée, la chaîne est effacée jusqu'à la fin.

Tableau 9- 59 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN	IN	String, WString
L	IN	Int
P	IN	Int
OUT	OUT	String, WString

Tableau 9- 60 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	P est supérieur à la longueur actuelle de IN.	IN est copié dans OUT sans qu'aucun caractère ne soit supprimé.
	La chaîne résultante après suppression des caractères est plus grande que la longueur maximale de la chaîne OUT.	Les caractères de la chaîne résultante sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	L est inférieur à 0 ou P est inférieur ou égal à 0.	La longueur en cours est mise à 0.
	La longueur en cours de IN dépasse la longueur maximale de IN ou la longueur en cours de OUT dépasse la longueur maximale de OUT.	
	La longueur maximale de IN ou OUT n'entre pas dans la zone de mémoire allouée.	
La longueur maximale de IN ou OUT est 0 ou 255.		

9.2.4.6 INSERT (Insérer des caractères dans une chaîne de caractères)

Tableau 9- 61 Instruction Insérer des caractères dans une chaîne

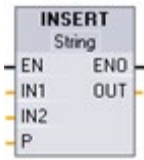
CONT/LOG	SCL	Description
	<pre>out := INSERT(in1, in2, p);</pre>	Insère la chaîne IN2 dans la chaîne IN1. L'insertion commence après le caractère à la position P.

Tableau 9- 62 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN1	IN	String, WString Chaîne d'entrée 1
IN2	IN	String, WString Chaîne d'entrée 2
P	IN	Int Dernière position de caractère dans la chaîne IN1 avant le point d'insertion de la chaîne IN2. Le premier caractère de la chaîne IN1 correspond au numéro de position 1.
OUT	OUT	String, WString Chaîne résultante

Tableau 9- 63 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	P est supérieur à la longueur de IN1.	IN2 est concaténé à IN1 immédiatement après le dernier caractère de IN1.
	P est inférieur à 0.	La longueur en cours est mise à 0.
	La chaîne résultante après insertion est plus grande que la longueur maximale de la chaîne OUT.	Les caractères de la chaîne résultante sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1, la longueur en cours de IN2 dépasse la longueur maximale de IN2 ou la longueur en cours de OUT dépasse la longueur maximale de OUT (chaîne invalide).	La longueur en cours est mise à 0.
	La longueur maximale de IN1, IN2 ou OUT n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN1 ou IN2 est 255, ou la longueur maximale de OUT est 0 ou 255 (type de données String)	
La longueur maximale de IN1 ou IN2 est 65534, ou la longueur maximale de OUT est 0 ou 65534 (type de données WString)		

9.2.4.7 REPLACE (Remplacer des caractères dans une chaîne de caractères)

Tableau 9- 64 Instruction Remplacer des caractères dans une chaîne

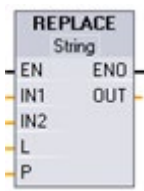
CONT/LOG	SCL	Description
	<pre> out := REPLACE (in1:=_string_in_, in2:=_string_in_, L:=_int_in_, p:=_int_in_); </pre>	Remplace L caractères dans la chaîne IN1. La substitution commence à la position de caractère P (inclusive) de la chaîne IN1, les caractères de substitution provenant de la chaîne IN2.

Tableau 9- 65 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN1	IN	String, WString
IN2	IN	String, WString
L	IN	Int
P	IN	Int
OUT	OUT	String, WString

Si le paramètre L est égal à zéro, la chaîne IN2 est insérée à la position P de la chaîne IN1 sans qu'aucun caractère de la chaîne IN1 ne soit effacé.

9.2 Chaînes de caractères et caractères

Si P est égal à 1, les L premiers caractères de la chaîne IN1 sont remplacés par les caractères de la chaîne IN2.

Tableau 9- 66 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	P est supérieur à la longueur de IN1.	IN2 est concaténé à IN1 immédiatement après le dernier caractère de IN1.
	P pointe à l'intérieur de IN1 mais il reste moins de L caractères dans IN1.	IN2 remplace les derniers caractères de IN1 en commençant à la position P.
	La chaîne résultante après substitution est plus grande que la longueur maximale de la chaîne OUT.	Les caractères de la chaîne résultante sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	La longueur maximale de IN1 est 0.	IN2 caractères sont copiés dans OUT.
	L est inférieur à 0 ou P est inférieur ou égal à 0.	La longueur en cours est mise à 0.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1, la longueur en cours de IN2 dépasse la longueur maximale de IN2 ou la longueur en cours de OUT dépasse la longueur maximale de OUT.	
	La longueur maximale de IN1, IN2 ou OUT n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN1 ou IN2 est 255, ou la longueur maximale de OUT est 0 ou 255 (type de données String)	
La longueur maximale de IN1 ou IN2 est 65534, ou la longueur maximale de OUT est 0 ou 65534 (type de données WString)		

9.2.4.8 FIND (Trouver des caractères dans une chaîne de caractères)

Tableau 9- 67 Instruction Trouver des caractères dans une chaîne

CONT/LOG	SCL	Description
	<pre> out := FIND (in1:=_string_in_, in2:=_string_in_); </pre>	Renvoie la position de la sous-chaîne indiquée par IN2 à l'intérieur de la chaîne IN1. La recherche commence par la gauche. La position de caractère de la première occurrence de la chaîne IN2 est renvoyée dans OUT. Si la chaîne IN2 est introuvable dans la chaîne IN1, zéro est renvoyé.

Tableau 9- 68 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN1	IN	String, WString
IN2	IN	String, WString
OUT	OUT	Int

Tableau 9- 69 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Position de caractère valide
0	IN2 est plus grande que IN1.	La position de caractère est mise à 0.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1 ou la longueur en cours de IN2 dépasse la longueur maximale de IN2 (chaîne invalide).	
	La longueur maximale de IN1 ou IN2 n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN1 ou IN2 est 255 (type de données String) ou 65535 (type de données WString)	

9.2.5 Informations Runtime

9.2.5.1 GetSymbolName (Lecture d'un nom de variable au paramètre d'entrée)

Tableau 9- 70 Instruction GetSymbolName

CONT/LOG	SCL	Description
<p style="text-align: center;">GetSymbolName</p>	<pre>OUT := GetSymbolName (variable:=_parameter_in_, size:=_dint_in_);</pre>	<p>L'instruction GetSymbolName renvoie une chaîne correspondant au nom d'une variable dans l'interface de bloc.</p> <p>Votre programme peut appeler cette instruction plusieurs fois avec des variables différentes. La valeur de processus de la variable n'est pas pertinente.</p> <p>L'instruction renvoie le nom lu dans le paramètre OUT.</p>

Paramètres

Le tableau suivant présente les paramètres de l'instruction GetSymbolName :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
VARIABLE	Input	PARAMETER	Sections de paramètres Input, Output, InOut	Variable de l'interface de bloc locale dont le nom doit être renvoyé sous forme de valeur chaîne
SIZE	Input	DINT	I, Q, M, D, L	Limite le nombre de caractères en sortie dans le paramètre OUT : <ul style="list-style-type: none"> • SIZE > 0 : GetSymbolName renvoie les premiers caractères SIZE du nom. • SIZE = 0 : GetSymbolName renvoie le nom en entier. • SIZE < 0 : GetSymbolName renvoie les derniers caractères SIZE du nom.
OUT	Return	WSTRING	I, Q, M, D, L	Sortie du nom de variable fourni par le paramètre d'entrée

Vous indiquez les paramètres d'entrée de l'interface de bloc dans le paramètre VARIABLE. N'utilisez pas de variable API ni de variable de bloc de données pour ce paramètre, mais uniquement un paramètre d'interface.

Le paramètre SIZE permet de limiter la longueur du nom de variable lu. Si l'instruction tronque le nom, elle signale cette troncature par le caractère "..." (caractère Unicode 16#2026) à la fin du nom. N'oubliez pas que le caractère "..." a lui-même une longueur de 1.

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Exemple : signification du paramètre SIZE

L'exemple suivant illustre la signification du paramètre SIZE. Le nom de variable suivant est lu dans l'interface de bloc : "MyPLCTag" (les guillemets au début et à la fin appartiennent au nom).

SIZE	GetSymbolName renvoie	Explication
1	'...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Marque signalant que le nom a été tronqué : ... Dernier caractère de WSTRING : '
2	""...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Premier caractère du nom et marque signalant que le nom a été tronqué : "... Dernier caractère de WSTRING : '
3	""M...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Deux premiers caractères du nom et marque signalant que le nom a été tronqué : "M... Dernier caractère de WSTRING : '
6	""MyPL...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Cinq premiers caractères du nom et marque signalant que le nom a été tronqué : "MyPL... Dernier caractère de WSTRING : '
0	""MyPLCTag"	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Tous les caractères du nom : "MyPLCTag" Dernier caractère de WSTRING : '

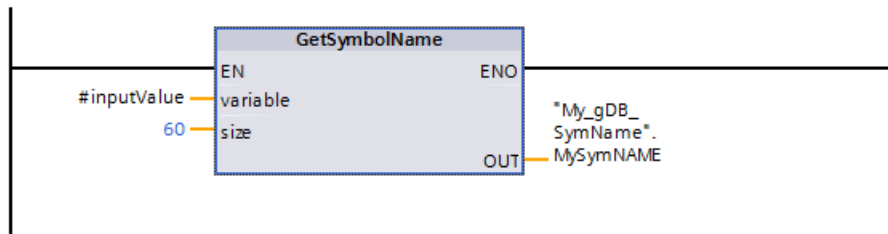
Exemple : lecture d'un mnémonique

L'exemple suivant permet de lire le nom d'une variable qui est connectée par le paramètre d'entrée d'un bloc.

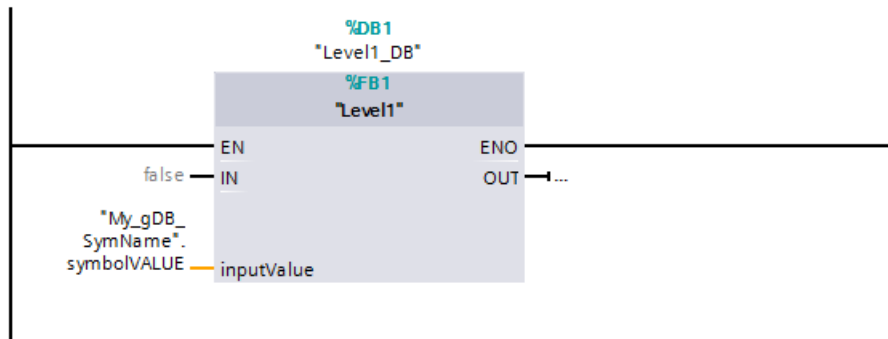
Créez deux variables dans un bloc de données global pour stocker les données.

My_gDB_SymName			
	Nom	Type de données	Valeur de départ
1	Static		
2	MySymNAME	WString	WSTRING#"
3	symbolVALUE	Byte	16#42

Créez dans le bloc Level1 un paramètre d'entrée inputValue avec le type de données BYTE. Appelez l'instruction GetSymbolName dans le bloc Level1. Interconnectez les paramètres de l'instruction comme suit.



Connectez le paramètre inputValue du bloc Level1 comme suit.



L'instruction GetSymbolName est exécutée dans le bloc Level1. La connexion du paramètre d'entrée inputValue du bloc Level1 est déterminée à l'aide du paramètre d'entrée VARIABLE de l'instruction. La variable symbolVALUE est alors lue et transmise sous forme de chaîne de caractères au paramètre de sortie OUT ("MySymNAME"). La longueur de la chaîne de caractères est limitée à 60 caractères conformément à la valeur du paramètre d'entrée SIZE.

My_gDB_SymName				
	Nom	Type de don...	Valeur de départ	Valeur de visualisation
1	Static			
2	MySymNAME	WString	WSTRING#"	WSTRING#"My_gDB_SymName".symbolVALUE'
3	symbolVALUE	Byte	16#42	16#42

9.2.5.2 GetSymbolPath (Interroger un nom global composite de l'affectation de paramètre d'entrée)

Tableau 9- 71 Instruction GetSymbolPath

CONT/LOG	SCL	Description
	<pre>OUT := GetSymbolPath(variable:=_parameter_in_, size:=_dint_in_);</pre>	<p>L'instruction GetSymbolPath permet de lire le nom global composé d'un paramètre d'entrée dans l'interface locale d'un bloc (FB ou FC). Le nom est constitué du chemin de stockage et du nom de la variable.</p> <p>Votre programme peut appeler cette instruction plusieurs fois avec des variables différentes. La valeur de processus de la variable n'est pas pertinente.</p> <p>L'instruction renvoie le nom lu dans le paramètre OUT.</p>

Paramètres

Le tableau suivant présente les paramètres de l'instruction GetSymbolPath :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
VARIABLE	Input	PARAMETER	Sections de paramètres Input, Output, InOut	Sélection de l'interface locale pour laquelle vous souhaitez lire le nom global de la fourniture de paramètre d'entrée.
SIZE	Input	DINT	I, Q, M, D, L ou constante	Limite le nombre de caractères en sortie dans le paramètre OUT <ul style="list-style-type: none"> • SIZE > 0 : GetSymbolPath renvoie les SIZE premiers caractères du nom. • SIZE = 0 : GetSymbolPath renvoie le nom en entier. • SIZE < 0 : GetSymbolPath renvoie les SIZE derniers caractères du nom.
OUT	Output	WSTRING	I, Q, M, D, L	Sortie du nom de variable de la fourniture de paramètres d'entrée.

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Utilisation

Tenez compte des conseils suivants lors de l'utilisation de l'instruction GetSymbolPath :

- Indiquez dans le paramètre VARIABLE de l'instruction l'interface de bloc par laquelle le nom de la variable d'entrée doit être lu :
 - Si une variable de bloc de données fournit le paramètre d'entrée, GetSymbolPath transmet le nom du DB, les structures contenues et le nom de la variable.
 - Si une variable API fournit les paramètres d'entrée, GetSymbolPath transmet le nom de cette variable.
 - Si une constante fournit le paramètre d'entrée, GetSymbolPath transmet la valeur de la constante.
- Le paramètre SIZE permet de limiter la longueur du nom de variable lu. Si le nom a été tronqué, cela est signalé par le caractère "..." (caractère Unicode 16#2026) à la fin du nom. N'oubliez pas que le caractère "..." a lui-même une longueur de 1.

Exemple : Signification du paramètre SIZE

L'exemple suivant illustre la signification du paramètre SIZE. GetSymbolPath a lu le nom de variable suivant dans l'interface de bloc : "MyPLCTag" (les guillemets au début et à la fin appartiennent au nom).

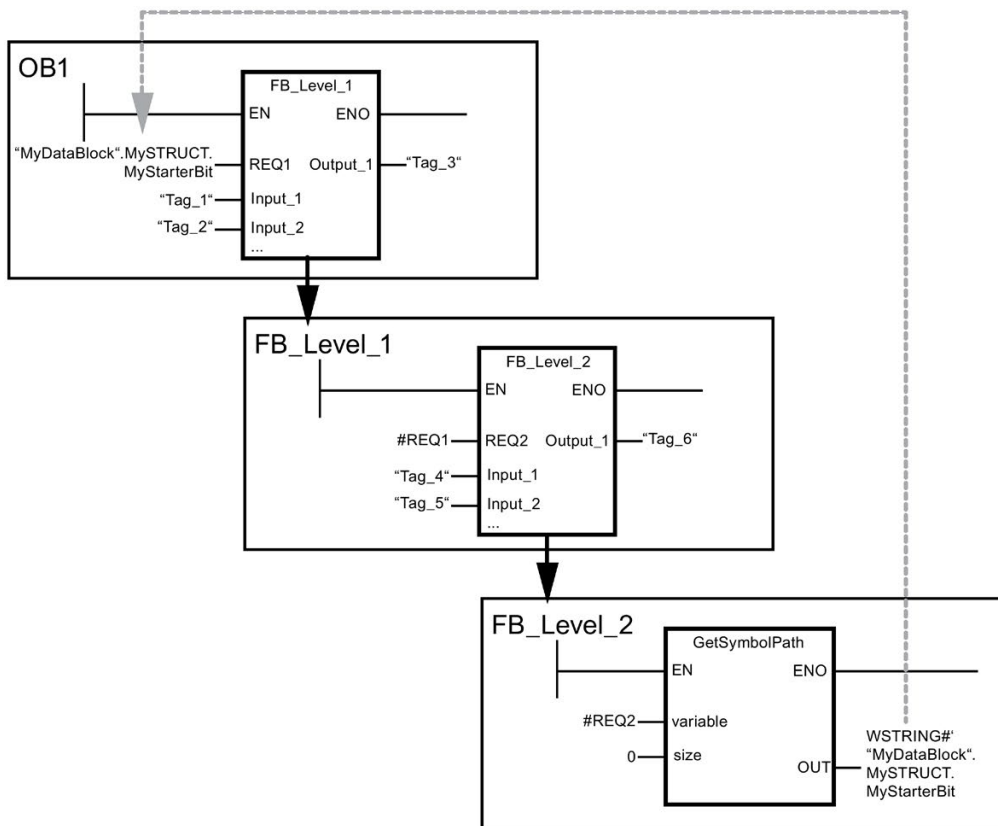
SIZE	GetSymbolPath renvoie	Explication
1	'...'	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Marque signalant que le nom a été tronqué : ... • Dernier caractère de WSTRING : '
2	""...'	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Premier caractère du nom et marque signalant que le nom a été tronqué : "... • Dernier caractère de WSTRING : '
3	""M...'	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Deux premiers caractères du nom et marque signalant que le nom a été tronqué : "M... • Dernier caractère de WSTRING : '
6	""MyPL...'	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Cinq premiers caractères du nom et marque signalant que le nom a été tronqué : "MyPL... • Dernier caractère de WSTRING : '
0	""MyPLCTag"	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Tous les caractères du nom : "MyPLCTag" • Dernier caractère de WSTRING : '

Exemple : Appel de GetSymbolPath sur plusieurs niveaux d'appel de bloc

L'exemple suivant montre l'utilisation de GetSymbolPath sur plusieurs niveaux d'appel :

- Le bloc d'organisation OB 1 appelle le bloc FB_Level_1, qui appelle à son tour le bloc FB_Level_2
- Le bloc FB_Level_2 exécute GetSymbolPath pour lire le chemin du paramètre au niveau de l'interface REQ2.
- Comme REQ2 est fourni par l'interface REQ1, l'instruction détermine le chemin du paramètre d'entrée de REQ1.
- La variable MyStarterBit est le paramètre d'entrée REQ1. Le bit se trouve dans la structure MySTRUCT du bloc de données MyDataBlock.

GetSymbolPath lit cette information et transmet le chemin ("MyDataBlock".MySTRUCT.MyStarterBit) au paramètre OUT.



9.2.5.3 GetInstanceName (Lire le nom de l'instance de bloc)

Tableau 9- 72 Instruction GetInstanceName

CONT/LOG	SCL	Description
<div style="border: 1px solid black; padding: 2px;"> <p style="text-align: center; margin: 0;">GetInstanceName</p> <p style="margin: 0;">- EN ENO -</p> <p style="margin: 0;">- size OUT -</p> </div>	<pre>OUT := GetInstanceName (size := _dint_in_);</pre>	L'instruction GetInstanceName permet de lire le nom du bloc de données d'instance à l'intérieur d'un bloc fonctionnel.

Paramètres

Le tableau suivant présente les paramètres de l'instruction GetInstanceName :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
SIZE	Input	DINT	I, Q, M, D, L ou constante	Limite le nombre de caractères en sortie dans le paramètre OUT <ul style="list-style-type: none"> SIZE > 0 : GetInstanceName renvoie les SIZE premiers caractères du nom. SIZE = 0 : GetInstanceName renvoie le nom en entier. SIZE < 0 : GetInstanceName renvoie les SIZE derniers caractères du nom.
OUT	Output	WSTRING	I, Q, M, D, L	Nom lu du bloc de données d'instance

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Exemple : signification du paramètre SIZE

Le paramètre SIZE permet de limiter la longueur du nom d'instance lu. Si l'instruction a tronqué le nom, elle signale cette troncature par le caractère "... " (caractère Unicode 16#2026) à la fin du nom. N'oubliez pas que le caractère "... " a lui-même une longueur de 1.

L'exemple suivant illustre la signification du paramètre SIZE. GetInstanceName a lu le nom d'instance suivant dans l'interface de bloc : "Level1_DB" (les guillemets au début et à la fin appartiennent au nom).

SIZE	GetSymbolPath renvoie	Explication
1	'...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Marque signalant que le nom a été tronqué : ... Dernier caractère de WSTRING : '
2	""...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Premier caractère du nom et marque signalant que le nom a été tronqué : "... Dernier caractère de WSTRING : '
3	""L...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Deux premiers caractères du nom et marque signalant que le nom a été tronqué : "L... Dernier caractère de WSTRING : '
6	""Leve...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Cinq premiers caractères du nom et marque signalant que le nom a été tronqué : "Leve... Dernier caractère de WSTRING : '
0	""Level1_DB""	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Tous les caractères du nom : "Level1_DB" Dernier caractère de WSTRING : '

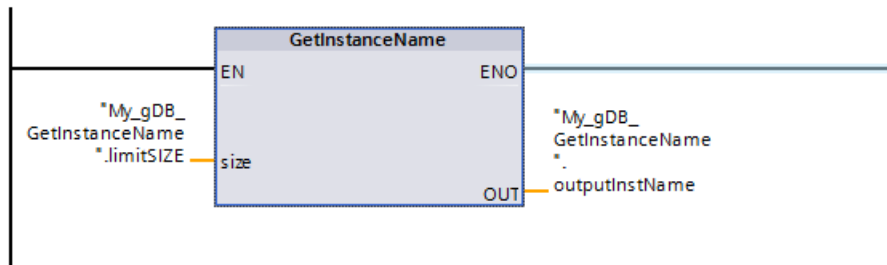
GetInstanceName écrit le nom du bloc de données d'instance dans le paramètre OUT. Si le nom du bloc de données d'instance est plus long que la longueur maximale de WSTRING, l'instruction le tronque.

Exemple : lecture du nom d'un bloc de données d'instance

L'exemple suivant montre comment lire le nom d'un bloc de données d'instance.
 Créez deux variables dans un bloc de données global pour stocker les données.
 Définissez les paramètres de l'instruction comme suit.

My_gDB_GetInstanceName			
	Nom	Type de données	Valeur de départ
1	Static		
2	limitSIZE	DInt	0
3	outputInstName	WString	WSTRING#"

Le bloc Level1_gin exécute l'instruction GetInstanceName qui détermine le bloc de données d'instance associé au bloc Level1_gin et transmet ce nom sous forme de chaîne de caractères au paramètre de sortie OUT (outputInstName). Selon la valeur 0 du paramètre SIZE (limitSIZE), la longueur de la chaîne de caractères est illimitée.



My_gDB_GetInstanceName				
	Nom	Type de données	Valeur de départ	Valeur de visualisation
1	Static			
2	limitSIZE	DInt	0	0
3	outputInstName	WString	WSTRING#"	WSTRING#"Level_1_DB"

9.2.5.4 GetInstancePath (Interroger le nom global composite de l'instance de bloc)

Tableau 9- 73 Instruction GetInstancePath

CONT/LOG	SCL	Description
<div style="border: 1px solid gray; padding: 5px;"> <p style="text-align: center;">GetInstancePath</p> <p> - EN ENO - - size OUT - </p> </div>	<pre> OUT := GetInstancePath(size:=_dint_in_); </pre>	L'instruction GetInstancePath permet de lire le nom global composé de l'instance de bloc à l'intérieur d'un bloc fonctionnel. Le nom global composé de l'instance de bloc est le chemin de la hiérarchie d'appel complète lorsque le programme appelle plusieurs instances.

Paramètres

Le tableau suivant présente les paramètres de l'instruction GetInstancePath :

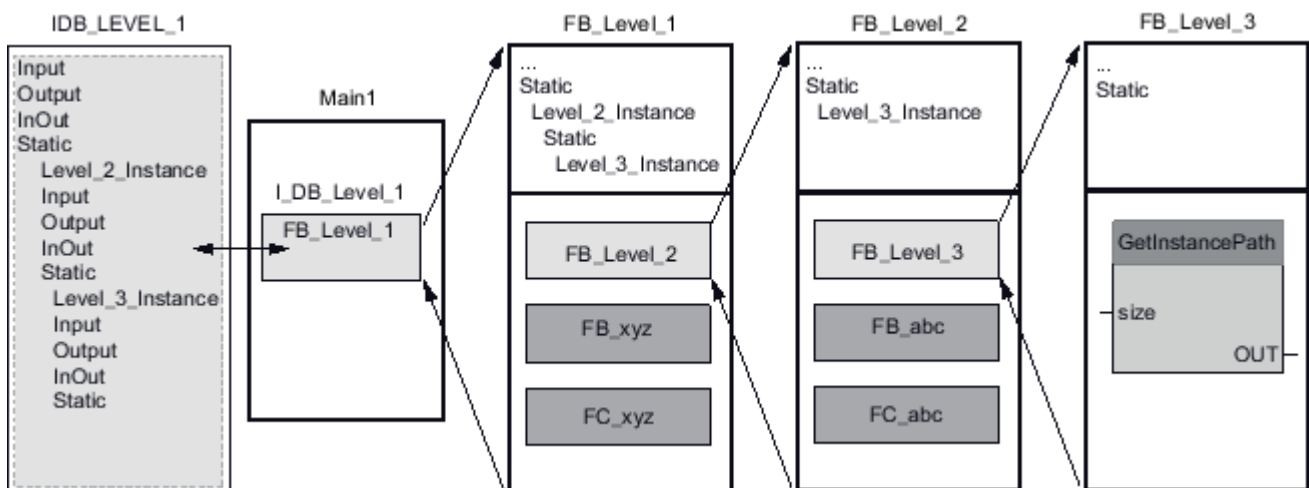
Paramètre	Déclaration	Type de données	Zone de mémoire	Description
SIZE	Input	DINT	I, Q, M, D, L ou constante	Limite le nombre de caractères en sortie dans le paramètre OUT. <ul style="list-style-type: none"> • SIZE > 0 : GetInstancePath renvoie les SIZE premiers caractères du nom. • SIZE = 0 : GetInstancePath renvoie le nom en entier. • SIZE < 0 : GetInstancePath renvoie les SIZE derniers caractères du nom.
OUT	Output	WSTRING	I, Q, M, D, L	Nom global lu de l'instance de bloc. Si le nom global de l'instance de bloc est plus long que la longueur maximale de WSTRING (254 caractères), GetInstancePath le tronque.

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Exemple : appel de GetInstancePath pour obtenir le chemin d'un appel de FB multiinstance

Dans l'exemple suivant, le bloc fonctionnel FB_Level_3 appelle l'instruction GetInstancePath.

- Le bloc fonctionnel FB_Level_3 enregistre ses données dans le bloc fonctionnel appelant FB_Level_2.
- Le bloc fonctionnel FB_Level_2 enregistre à son tour ses données dans le bloc fonctionnel appelant FB_Level_1.
- Le bloc fonctionnel FB_Level_1 enregistre à son tour ses données dans son bloc de données d'instance IDB_LEVEL_1. Grâce à l'utilisation de multiinstances, le bloc de données d'instance de FB_Level_1 contient toutes les données des trois blocs fonctionnels.



L'instruction GetInstancePath renvoie les valeurs suivantes pour cet exemple en fonction de la valeur du paramètre SIZE :

SIZE	GetInstancePath renvoie	Explication
1	'...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : ' Marque signalant que le nom a été tronqué : ... Dernier caractère de WSTRING : '
2	""...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : '' Premier caractère du nom et marque signalant que le nom a été tronqué : "... Dernier caractère de WSTRING : ''
3	""l...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : '' Deux premiers caractères du nom et marque signalant que le nom a été tronqué : "l... Dernier caractère de WSTRING : ''
6	""IDB_...'	<ul style="list-style-type: none"> Premier caractère de WSTRING : '' Cinq premiers caractères du nom et marque signalant que le nom a été tronqué : "IDB_... Dernier caractère de WSTRING : ''
0	""IDB_LEVEL_1".Level_2_ Instance.Level_3_Instance'	<ul style="list-style-type: none"> Premier caractère de WSTRING : '' Tous les caractères du nom : "IDB_LEVEL_1".Level_2_Instance.Level_3_Instance Dernier caractère de WSTRING : ''

Remarque

Utilisation de GetInstancePath dans les blocs fonctionnels avec instance unique

Si le bloc fonctionnel dans lequel vous appelez GetInstancePath enregistre ses données dans son propre bloc de données d'instance, GetInstancePath transmettra le nom de l'instance unique comme nom global. Le résultat dans le paramètre OUT correspond dans ce cas à l'instruction GetInstanceName (Page 386).

9.2.5.5 GetBlockName (Lire le nom du bloc)

Tableau 9- 74 Instruction GetBlockName

CONT/LOG	SCL	Description
GetBlockName - EN ENO - - SIZE RET_VAL -	RET_VAL := GetBlockName (size:=_dint_in_);	L'instruction GetBlockName permet de lire le nom du bloc dans lequel elle est appelée.

Paramètres

Le tableau suivant présente les paramètres de l'instruction GetBlockName :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
SIZE	Input	UINT	I, Q, M, D, L ou constante	Limite le nombre de caractères en sortie dans le paramètre RET_VAL <ul style="list-style-type: none"> • SIZE > 0 : GetBlockName renvoie les SIZE premiers caractères du nom. • SIZE = 0 : GetBlockName renvoie le nom en entier. • SIZE < 0 : GetBlockName renvoie les SIZE derniers caractères du nom.
RET_VAL	Output	WSTRING	I, Q, M, D, L	Nom lu du bloc de données d'instance

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Exemple : Signification du paramètre SIZE

Pour limiter la longueur du nom de bloc à un certain nombre de caractères, indiquez la longueur maximale dans le paramètre SIZE. Si l'instruction GetBlockName tronque le nom, elle signale cette troncature par le caractère "..." (caractère Unicode 16#2026) à la fin du nom. N'oubliez pas que le caractère "..." a lui-même une longueur de 1.

L'exemple suivant illustre la signification du paramètre SIZE. GetBlockName a lu le nom de bloc suivant : "Level1_gbn" (les guillemets au début et à la fin appartiennent au nom).

SIZE	GetBlockName renvoie	Explication
1	'...'	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Marque signalant que le nom a été tronqué : ... • Dernier caractère de WSTRING : '
2	""...'	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Premier caractère du nom et marque signalant que le nom a été tronqué : "... • Dernier caractère de WSTRING : '
3	""L...'	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Deux premiers caractères du nom et marque signalant que le nom a été tronqué : "L... • Dernier caractère de WSTRING : '
6	""Leve...'	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Cinq premiers caractères du nom et marque signalant que le nom a été tronqué : "Leve... • Dernier caractère de WSTRING : '
0	""Level1_gbn"	<ul style="list-style-type: none"> • Premier caractère de WSTRING : ' • Tous les caractères du nom : "Level1_gbn" • Dernier caractère de WSTRING : '

GetBlockName écrit le nom du bloc dans le paramètre RET_VAL. Si le nom du bloc est plus long que la longueur maximale de WSTRING, l'instruction le tronque.

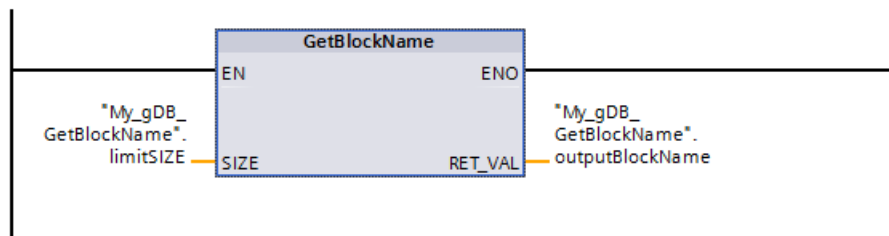
Exemple : lecture du nom d'un bloc

L'exemple suivant montre comment lire le nom d'un bloc.

1. Créez deux variables dans un bloc de données global pour stocker les données.

My_gDB_GetBlockName			
	Nom	Type de données	Valeur de départ
1	Static		
2	limitSIZE	DInt	0
3	outputBlockName	WString	WSTRING#"

2. Définissez les paramètres de l'instruction comme suit :



Le bloc Level1_gbn exécute l'instruction GetBlockName qui lit le nom du bloc Level1_gbn et transmet ce nom sous forme de chaîne de caractères au paramètre de sortie RET_VAL (outputBlockName). Comme le paramètre SIZE est égal à 0 (limitSIZE), la longueur de la chaîne de caractères est illimitée.

My_gDB_GetBlockName				
	Nom	Type de données	Valeur de départ	Valeur de visualisation
1	Static			
2	limitSIZE	DInt	0	0
3	outputBlockName	WString	WSTRING#"	WSTRING#"Level_1_gbn"

9.3 Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)

9.3.1 Instructions de périphérie décentralisée

Vous pouvez utiliser les instructions de périphérie décentralisée suivantes avec PROFINET, PROFIBUS ou AS-i :

- Instruction RDREC (Page 395): Lit un enregistrement de numéro INDEX dans un module ou un appareil.
- Instruction WRREC (Page 395): Transfère un enregistrement de numéro INDEX dans un module ou un appareil défini par ID.
- Instruction GETIO (Page 398) : Lit de manière cohérente toutes les entrées d'un esclave DP norme / périphérique PROFINET IO.
- Instruction SETIO (Page 399) : Transfère de manière cohérente les données de la zone source définie par le paramètre OUTPUTS dans l'esclave DP norme / le périphérique PROFINET IO adressé.
- Instruction GETIO_PART (Page 401) : Lit de manière cohérente une zone liée des entrées d'un module d'E/S.
- Instruction SETIO_PART (Page 403) : Écrit de manière cohérente les données de la zone source définie par le paramètre OUTPUTS dans les sorties d'un module d'E/S.
- Instruction RALRM (Page 404): Permet de recevoir une alarme en provenance d'un module ou d'un appareil avec toutes les informations correspondantes et de transmettre ces informations aux paramètres de sortie.
- Instruction DPRD_DAT (Page 418): Vous permet de lire des zones de données cohérentes de plus de 64 octets à partir d'un module ou d'un appareil avec l'instruction DPRD_DAT.
- Instruction DPWR_DAT (Page 418): Vous permet d'écrire des zones de données cohérentes de plus de 64 octets dans un module ou un appareil avec l'instruction DPWR_DAT.

L'instruction D_ACT_DP (Page 408) permet d'activer et de désactiver de manière ciblée des périphériques PROFINET IO configurés. Vous pouvez également déterminer si chaque périphérique PROFINET IO affecté est actuellement activé ou désactivé.

Remarque

Remarque : L'instruction D_ACT_DP n'est utilisable qu'avec des périphériques PROFINET IO. Vous ne pouvez pas l'utiliser avec des esclaves PROFIBUS DP.

L'instruction DPNRM_DG (Page 426) permet de lire les données de diagnostic en cours d'un esclave DP dans le format précisé par EN 50 170 Volume 2, PROFIBUS.

Remarque

L'instruction DPNRM_DG n'est utilisable qu'avec PROFIBUS.

9.3.2 RDREC et WRREC (Lire l'enregistrement/Écrire l'enregistrement)

Vous pouvez utiliser les instructions RDREC (Lire l'enregistrement) et WRREC (Ecrire l'enregistrement) avec PROFINET, PROFIBUS et AS-i.

Tableau 9- 75 Instructions RDREC et WRREC

CONT/LOG	SCL	Description
<p>"RDREC_DB"</p>	<pre>"RDREC_DB" (req:=_bool_in_, ID:=_word_in_, index:=_dint_in_, mlen:=_uint_in_, valid=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_dword_out_, len=>_uint_out_, rec- ord:= variant inout);</pre>	<p>Utilisez l'instruction RDREC pour lire l'enregistrement de numéro INDEX dans le composant désigné par le paramètre ID, par exemple un châssis central ou un composant décentralisé (PROFIBUS DP ou PROFINET IO). Indiquez dans MLEN le nombre maximum d'octets à lire. La longueur de la zone cible RECORD doit donc être d'au moins MLEN octets.</p>
<p>"WRREC_DB"</p>	<pre>"WRREC_DB" (req:=_bool_in_, ID:=_word_in_, index:=_dint_in_, len:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_dword_out_, rec- ord:= variant inout);</pre>	<p>Utilisez l'instruction WRREC pour transférer l'enregistrement RECORD de numéro INDEX dans un esclave DP/périphérique PROFINET IO désigné par ID, par exemple un module dans le châssis central ou un composant décentralisé (PROFIBUS DP ou PROFINET IO). Indiquez dans LEN la longueur en octets de l'enregistrement à transmettre. La longueur de la zone source RECORD doit donc être d'au moins LEN octets.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "RDREC_DB" et "WRREC_DB" sont les noms des DB d'instance.

Tableau 9- 76 Types de données pour les paramètres de RDREC et WRREC

Paramètre et type		Type de données	Description
REQ	IN	Bool	REQ = 1 : Transférer un enregistrement
ID	IN	HW_IO (Word)	<p>Adresse logique de l'esclave DP/composant PROFINET IO (module ou sous-module) :</p> <ul style="list-style-type: none"> • Pour un module de sorties, le bit 15 doit être à 1 (par exemple, pour l'adresse 5 : ID:= DW#16#8005). • Pour un module d'entrées-sorties, la plus petite des deux adresses doit être indiquée. <p>Remarque : Dans la version V3.0, l'ID d'appareil peut être déterminé de l'une des deux façons suivantes :</p> <ul style="list-style-type: none"> • Via les sélections suivantes dans la vue du réseau : <ul style="list-style-type: none"> – Appareil (boîte grise) – "Propriétés" de l'appareil – "Identificateur matériel" <p>Remarque : Tous les appareils n'affichent toutefois pas leur identificateur matériel.</p> • Via les sélections suivantes dans la vue du projet : <ul style="list-style-type: none"> – Variables API – Table de variables par défaut – Onglet Constantes système <p>Tous les identificateurs matériels d'appareils configurés sont affichés.</p> <p>Remarque : Dans la version V4.0, vous pouvez déterminer l'ID d'appareil (ID matérielle) pour le module d'interface en ouvrant la table des variables et en lisant le paramètre "Nom de l'appareil [EN-TÊTE]" sous Constantes système.</p>
INDEX	IN	Byte, Word, USInt, UInt, SInt, Int, DInt	Numéro de l'enregistrement
MLEN	IN	Byte, USInt, UInt	Longueur maximale en octets des informations de l'enregistrement à lire (RDREC)
VALID	OUT	Bool	Un nouvel enregistrement valide a été reçu (RDREC). Le bit VALID est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
DONE	OUT	Bool	L'enregistrement a été transféré (WRREC). Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • BUSY = 1 : L'opération de lecture (RDREC) ou d'écriture (WRREC) n'est pas encore terminée. • BUSY = 0 : La transmission de l'enregistrement est achevée.
ERROR	OUT	Bool	ERROR = 1 : Une erreur de lecture (RDREC) ou d'écriture (WRREC) s'est produite. Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS	OUT	DWord	État (Page 413) du bloc ou information d'erreur (Page 571)

Paramètre et type		Type de données	Description
LEN	OUT (RDREC) IN (WRREC)	UInt	<ul style="list-style-type: none"> Longueur des informations lues de l'enregistrement (RDREC) Longueur maximale en octets de l'enregistrement à écrire (WRREC)
RECORD	IN_OUT	Variant	<ul style="list-style-type: none"> Zone cible pour l'enregistrement lu (RDREC) Enregistrement (WRREC)

Les instructions RDREC et WRREC ont un fonctionnement asynchrone, c'est-à-dire que le traitement s'étend sur plusieurs appels d'instruction. Lancez la tâche en appelant RDREC ou WRREC avec REQ égal à 1.

L'état de la tâche est indiqué dans le paramètre de sortie BUSY et dans les deux octets centraux du paramètre de sortie STATUS. Le transfert de l'enregistrement est achevé lorsque le paramètre de sortie BUSY a pris la valeur FALSE.

La valeur TRUE (pendant un cycle seulement) du paramètre de sortie VALID (RDREC) ou DONE (WRREC) signifie que l'enregistrement a été transféré avec succès dans la zone cible RECORD (RDREC) ou dans l'appareil cible (WRREC). Pour RDREC, le paramètre de sortie LEN contient la longueur des données lues en octets.

Le paramètre de sortie ERROR indique (pendant un cycle seulement lorsque ERROR = TRUE) qu'une erreur de transmission d'enregistrement s'est produite. Si c'est le cas, le paramètre de sortie STATUS contient les informations d'erreur correspondantes (pendant un cycle seulement lorsque ERROR = TRUE).

Les enregistrements sont définis par le fabricant de l'appareil. Reportez-vous à la documentation du fabricant concernant l'appareil pour obtenir plus de détails sur un enregistrement.

Vous pouvez utiliser jusqu'à quatre instructions RDREC et quatre instructions WRREC simultanément.

Remarque

Si vous configurez un esclave DPV1 au moyen d'un fichier GSD (GSD version 3 ou plus) et que l'interface DP du maître DP est paramétrée à "S7 compatible", vous ne pourrez pas lire d'enregistrements des modules d'entrées-sorties dans le programme utilisateur avec "RDREC" ou écrire dans les modules d'entrées-sorties avec "WRREC". Dans ce cas, le maître DP accède au mauvais emplacement (emplacement configuré + 3).

Solution : Définissez l'interface du maître DP à "DPV1".

Remarque

Les interfaces des instructions "RDREC" et "WRREC" sont identiques aux FB "RDREC" et "WRREC" définis dans la publication "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

Remarque

Si vous utilisez "RDREC" ou "WRREC" pour lire ou écrire un enregistrement pour PROFINET IO, la CPU interprète des valeurs négatives dans les paramètres INDEX, MLEN et LEN en tant qu'entiers de 16 bits non signés.

9.3.3 GETIO (Lire la mémoire image)

L'instruction "GETIO" permet de lire de manière cohérente les entrées de modules ou sous-modules d'esclaves DP et de périphériques PROFINET IO. L'instruction "GETIO" appelle l'instruction "DPRD_DAT (Page 418)". En l'absence d'erreur pendant la transmission des données, les données lues sont reportées dans la zone de destination indiquée par le paramètre INPUTS.

Tableau 9- 77 Instruction GETIO (Lire la mémoire image)

CONT/LOG	SCL	Description
	<pre>"GETIO_DB" (id:=_uint_in_, status=>_dword_out_, len=>_int_out_, inputs:=_variant_inout_);</pre>	<p>Utilisez l'instruction "GETIO" pour lire de manière cohérente toutes les entrées d'un esclave DP norme ou d'un périphérique PROFINET IO.</p>

- ¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- ² Dans l'exemple SCL, "GETIO_DB" est le nom du DB d'instance.

La zone de destination doit avoir une longueur supérieure ou égale à la longueur du composant sélectionné.

Si la lecture se fait dans un esclave DP norme ayant une configuration modulaire ou plusieurs identifiants DP, vous accédez uniquement aux données d'un composant/identifiant DP à l'adresse de début configurée avec un appel "GETIO".

Paramètres

Le tableau suivant présente les paramètres de l'instruction GETIO :

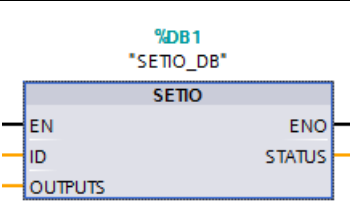
Paramètre	Déclaration	Type de données	Description
ID	IN	HW_SUBMODULE	ID matérielle de l'esclave DP norme/du périphérique PROFINET IO
STATUS ¹	OUT	DWord	Contient l'information d'erreur de "DPRD_DAT (Page 418)" sous la forme DW#16#40xxx00.
LEN	OUT	Int	Volume de données lu en octets
INPUTS	IN_OUT	Variante	<p>Zone de destination pour les données lues. La zone de destination doit avoir une longueur supérieure ou égale à la longueur de l'esclave DP norme ou du périphérique PROFINET IO sélectionné.</p> <p>Vous pouvez utiliser les types de données suivants :</p> <ul style="list-style-type: none"> Types de données système et tableaux de types de données système : BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT, ULINT Types de données utilisateur (UDT) Structures (STRUCT), mais uniquement dans des blocs de données (DB) non optimisés

¹ Utilisez le type de données DWord lors de l'affichage des codes d'erreur de "GETIO".

9.3.4 SETIO (Transférer la mémoire image)

L'instruction "SETIO" permet de transférer de manière cohérente les données de la zone source définie par le paramètre OUTPUTS dans les modules ou sous-modules adressés d'esclaves DP norme et de périphériques PROFINET IO. Si vous avez configuré la zone d'adresses concernée de l'esclave DP norme ou du périphérique PROFINET IO en tant que zone cohérente dans une mémoire image, les données sont transférées dans la mémoire image. "SETIO" appelle l'instruction "DPWR_DAT (Page 418)" pendant le transfert.

Tableau 9- 78 Instruction SETIO (Transférer la mémoire image)

CONT/LOG	SCL	Description
	<pre>"SETIO_DB" (id:=_uint_in_, status=>_dword_out_, outputs:=_variant_inout_);</pre>	<p>Utilisez l'instruction "SETIO" pour transférer de manière cohérente les données de la zone source définie par le paramètre OUTPUTS dans l'esclave DP norme ou le périphérique PROFINET IO adressé.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

² Dans l'exemple SCL, "SETIO_DB" est le nom du DB d'instance.

La zone source doit avoir une longueur supérieure ou égale à la longueur du composant sélectionné.

Dans le cas d'un esclave DP norme ou périphérique PROFINET IO ayant une configuration modulaire ou plusieurs identifiants DP, vous ne pouvez accéder qu'à un identifiant DP ou composant par appel de "SETIO".

Paramètres

Le tableau suivant présente les paramètres de l'instruction SETIO :

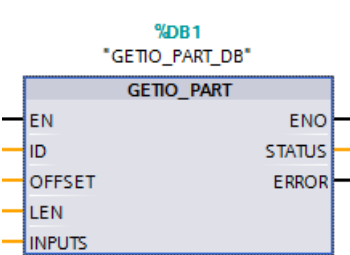
Paramètre	Déclaration	Type de données	Description
ID	IN	HW_SUBMOD ULE	ID matérielle de l'esclave DP norme/du périphérique PROFINET IO
STATUS ¹	OUT	DWord	Contient l'information d'erreur de "DPWR_DAT (Page 418)" sous la forme DW#16#40xxxx00
OUTPUTS	IN_OUT	Variant	<p>Zone source pour les données à écrire. La zone source doit avoir une longueur supérieure ou égale à la longueur de l'esclave DP norme ou du périphérique PROFINET IO sélectionné.</p> <p>Vous pouvez utiliser les types de données suivants :</p> <ul style="list-style-type: none"> Types de données système et tableaux de types de données système : BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT, ULINT Types de données utilisateur (UDT) Structures (STRUCT), mais uniquement dans des blocs de données (DB) non optimisés

¹ Utilisez le type de données DWord lors de l'affichage des codes d'erreur de "SETIO".

9.3.5 GETIO_PART (Lire la zone de mémoire image)

L'instruction "GETIO_PART" permet de lire de manière cohérente une zone liée des entrées de modules ou sous-modules d'esclaves DP et de périphériques PROFINET IO. GETIO_PART appelle l'instruction "DPRD_DAT (Page 418)".

Tableau 9- 79 Instruction GETIO_PART (Lire la zone de mémoire image)

CONT/LOG	SCL	Description
	<pre>"GETIO_PART_DB" (id:=_uint_in_, offset:=_int_in_, len:=_int_in_, status=>_dword_out_, error=>_bool_out_, inputs:=_variant_inout_);</pre>	<p>Utilisez l'instruction GETIO_PART pour lire de manière cohérente une zone liée des entrées d'un module d'E/S.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "GETIO_PART_DB" est le nom du DB d'instance.

Utilisez le paramètre d'entrée ID pour sélectionner le module d'E/S au moyen de l'D matériel.

Utilisez les paramètres OFFSET et LEN pour préciser la zone à lire de la mémoire image. Si la zone d'entrée définie par OFFSET et LEN n'est pas totalement couverte par le module, le bloc renvoie le code d'erreur DW#16#4080B700.

La longueur de la zone de destination doit être supérieure ou égale au nombre d'octets à lire :

- En l'absence d'erreur pendant la transmission des données, ERROR prend la valeur FALSE. Les données lues sont écrites dans la zone de destination définie par le paramètre INPUTS.
- En cas d'erreur pendant la transmission des données, ERROR prend la valeur TRUE. Le paramètre STATUS reçoit les informations d'erreur en provenance de DPRD_DAT.
- Si la zone de destination est supérieure à LEN, l'instruction écrit dans les LEN premiers octets de la zone de destination. ERROR prend la valeur FALSE.

Paramètres

Le tableau suivant présente les paramètres de l'instruction GETIO_PART :

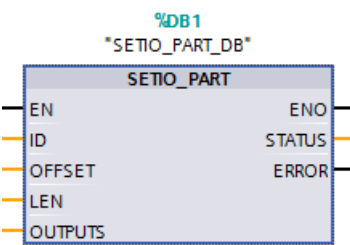
Paramètre	Déclaration	Type de données	Description
ID	IN	HW_SUBMODULE	Identification matérielle du module
OFFSET	IN	Int	Numéro du premier octet à lire dans la mémoire image pour le composant (valeur minimale possible : 0)
LEN	IN	Int	Nombre d'octets à lire
STATUS ¹	OUT	DWord	Contient l'information d'erreur de "DPRD_DAT (Page 418)" sous la forme DW#16#40xxxx00 si ERROR = TRUE.
ERROR	OUT	Bool	Signalisation d'erreur. ERROR = TRUE si une erreur se produit à l'appel de DPRD_DAT (Page 418).
INPUTS	IN_OUT	Variant	<p>Zone de destination pour les données lues. Si la zone de destination est supérieure à LEN, l'instruction écrit dans les LEN premiers octets de la zone de destination.</p> <p>Vous pouvez utiliser les types de données suivants :</p> <ul style="list-style-type: none"> Types de données système et tableaux de types de données système : BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT, ULINT Types de données utilisateur (UDT) Structures (STRUCT), mais uniquement dans des blocs de données (DB) non optimisés

¹ Utilisez le type de données DWord lors de l'affichage des codes d'erreur de "GETIO_PART".

9.3.6 SETIO_PART (Transférer la zone de mémoire image)

L'instruction "SETIO_PART" permet d'écrire de manière cohérente des données de la zone source définie par le paramètre OUTPUTS dans les sorties de modules ou sous-modules d'esclaves DP et de périphériques PROFINET IO. SETIO_PART appelle l'instruction "DPWR_DAT (Page 418)".

Tableau 9- 80 Instruction SETIO_PART (Transférer la zone de mémoire image)

CONT/LOG	SCL	Description
	<pre>"SETIO_PART_DB" (id:=_uint_in_, offset:=_int_in_, len:=_int_in_, status=>_dword_out_, error=>_bool_out_, outputs:=_variant_inout_);</pre>	<p>Utilisez l'instruction SETIO_PART pour écrire de manière cohérente les données de la zone source définie par OUTPUTS dans les sorties d'un module d'E/S.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "SETIO_PART_DB" est le nom du DB d'instance.

Avec le paramètre d'entrée ID, vous sélectionnez le module d'entrées-sorties sur la base de l'identification matérielle.

Quant aux paramètres OFFSET et LEN, ils vous permettent de définir la partie où écrire de la zone de mémoire image du composant adressé au moyen de ID. Si la zone de sortie définie par OFFSET et LEN n'est pas totalement couverte par le module, le bloc renvoie le code d'erreur DW#16#4080B700.

La longueur de la zone de destination doit être supérieure ou égale au nombre d'octets à lire :

- En l'absence d'erreur pendant la transmission des données, ERROR prend la valeur FALSE.
- En cas d'erreur pendant la transmission des données, ERROR prend la valeur TRUE et STATUS reçoit les informations sur l'erreur de DPWR_DAT.
- Si la zone source est supérieure à LEN, l'instruction transfère dans les LEN premiers octets de OUTPUTS. ERROR prend la valeur FALSE.

Paramètres

Le tableau suivant présente les paramètres de l'instruction SETIO_PART :

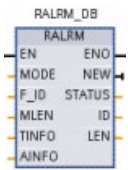
Paramètre	Déclaration	Type de données	Description
ID	IN	HW_SUBMODULE	Identification matérielle du module périphérique
OFFSET	IN	Int	Numéro du premier octet à écrire dans la mémoire image pour le composant (valeur minimale possible : 0)
LEN	IN	Int	Nombre d'octets à écrire
STATUS ¹	OUT	DWord	Contient l'information d'erreur de "DPWR_DAT (Page 418)" sous la forme DW#16#40xxx00 si ERROR = TRUE.
ERROR	OUT	Bool	Signalisation d'erreur. ERROR = TRUE si une erreur se produit à l'appel de DPWR_DAT (Page 418).
OUTPUTS	IN_OUT	Variant	Zone source pour les données à écrire. Si la zone source est supérieure à LEN, les LEN premiers octets de OUTPUTS sont transférés. Vous pouvez utiliser les types de données suivants : <ul style="list-style-type: none"> Types de données système et tableaux de types de données système : BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT, ULINT Types de données utilisateur (UDT) Structures (STRUCT), mais uniquement dans des blocs de données (DB) non optimisés

¹ Utilisez le type de données DWord lors de l'affichage des codes d'erreur de "SETIO_PART".

9.3.7 RALRM (Recevoir l'alarme)

Vous pouvez utiliser l'instruction RALRM (Lecture d'alarme) avec PROFINET et PROFIBUS.

Tableau 9- 81 Instruction RALRM

CONT/LOG	SCL	Description
	<pre>"RALRM_DB" (mode:=_int_in_, f_ID:=_word_in_, mlen:=_uint_in_, new=>_bool_out_, status=>_dword_out_, ID=>_word_out_, len=>_uint_out_, tinfo:=_variant_inout_, ainfo:=_variant_inout_);</pre>	<p>Utilisez l'instruction RALRM (Lecture d'alarme) pour lire des informations d'alarme de diagnostic en provenance de modules ou de périphérique PROFIBUS ou PROFINET IO.</p> <p>Les informations dans les paramètres de sortie contiennent les informations de déclenchement de l'OB appelé, ainsi que des informations de la source de l'alarme.</p> <p>Appelez RALRM dans un OB d'alarme pour renvoyer des informations concernant le ou les événements ayant provoqué l'alarme. Les alarmes suivantes sont prises en charge dans le S7-1200 : état, actualisation, profil, alarme de diagnostic, débrogage ou enfichage de modules, défaillance de châssis ou de station.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

² Dans l'exemple SCL, "RALRM_DB" est le nom du DB d'instance.

Tableau 9- 82 Types de données pour les paramètres

Paramètre et type		Type de données	Description
MODE	IN	Byte, USInt, SInt, Int	Mode de fonctionnement
F_ID	IN	HW_IO (Word)	<p>Adresse de début logique du composant (module) dont il faut recevoir les alarmes</p> <p>Remarque : L'ID d'appareil peut être déterminé de l'une des deux façons suivantes :</p> <ul style="list-style-type: none"> • Via les sélections suivantes dans la vue du réseau : <ul style="list-style-type: none"> – Appareil (boîte grise) – "Propriétés" de l'appareil – "Identificateur matériel" <p>Remarque : Tous les appareils n'affichent pas leur identificateur matériel.</p> • Via les sélections suivantes dans la vue du projet : <ul style="list-style-type: none"> – Variables API – Table de variables par défaut – Onglet Constantes système – Tous les identificateurs matériels d'appareils configurés sont affichés.
MLEN	IN	Byte, USInt, UInt	Longueur maximale en octets des informations d'alarme à recevoir. MLEN égal à 0 permet de recevoir autant d'informations d'alarme qu'il y en a dans la zone cible AINFO.
NEW	OUT	Bool	Une nouvelle alarme a été reçue.
STATUS	OUT	DWord	Etat de l'instruction RALRM. Reportez-vous à "Paramètre STATUS pour RDREC, WRREC et RALRM" (Page 413) pour plus d'informations.
ID	OUT	HW_IO (Word)	<p>Identificateur matériel du module d'E/S ayant provoqué l'alarme de diagnostic</p> <p>Remarque : Pour obtenir une explication sur la manière de déterminer l'ID d'appareil, voir le paramètre F_ID .</p>
LEN	OUT	DWord, UInt, UInt, DInt, Real, LReal	Longueur des informations d'alarme AINFO reçues
TINFO	IN_OUT	Variant	Informations sur la tâche : Plage cible pour les informations de déclenchement et de gestion d'OB. La longueur TINFO est toujours de 32 octets.
AINFO	IN_OUT	Variant	Informations sur l'alarme : Zone cible pour les informations d'en-tête et les informations d'alarme supplémentaires. Pour AINFO, vous devez prévoir une longueur minimale de MLEN octets si MLEN est strictement supérieur à 0. La longueur de AINFO est variable.

Remarque

Si vous appelez "RALRM" dans un OB dont l'événement déclencheur n'est pas une alarme d'E/S, l'instruction fournira des informations réduites dans ses sorties.

Veillez à utiliser des DB d'instance différents lorsque vous appelez "RALRM" dans différents OB. Si vous évaluez les données résultant d'un appel "RALRM" hors de l'OB d'alarme correspondant, utilisez un DB d'instance distinct pour chaque événement déclencheur de l'OB.

Remarque

L'interface de l'instruction "RALRM" est identique au FB "RALRM" défini dans la publication "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

Appel de RALRM

Vous pouvez appeler l'instruction RALRM dans trois modes différents (MODE).

Tableau 9- 83 Modes de fonctionnement de l'instruction RALRM

MODE	Description
0	<ul style="list-style-type: none"> ID contient l'identificateur matériel du module d'E/S ayant déclenché l'alarme. Le paramètre de sortie NEW est mis à TRUE. LEN produit une sortie de 0. AINFO et TINFO ne sont pas actualisés.
1	<ul style="list-style-type: none"> ID contient l'identificateur matériel du module d'E/S ayant déclenché l'alarme. Le paramètre de sortie NEW est mis à TRUE. LEN produit une sortie de la quantité de données AINFO renvoyées en octets. AINFO et TINFO sont actualisés par des informations relatives à l'alarme.
2	<p>Si l'identificateur matériel affecté au paramètre d'entrée F_ID a déclenché l'alarme, alors :</p> <ul style="list-style-type: none"> ID contient l'identificateur matériel du module d'E/S ayant déclenché l'alarme. Il devrait être identique à la valeur indiquée dans F_ID. Le paramètre de sortie NEW est mis à TRUE. LEN produit une sortie de la quantité de données AINFO renvoyées en octets. AINFO et TINFO sont actualisés par des informations relatives à l'alarme.

Remarque

Si vous définissez une zone de destination trop courte pour TINFO ou AINFO, RALRM ne pourra pas renvoyer les informations complètes.

MLEN peut limiter la quantité de données AINFO renvoyées.

Reportez-vous aux paramètres AINFO et aux paramètres TINFO dans le système d'information en ligne de STEP 7 pour obtenir des informations sur la manière d'interpréter les données TINFO et AINFO.

Données de bloc d'organisation TInfo

Le tableau suivant montre comment arranger les données TInfo pour l'instruction RALRM :

Identique pour les OB : Etat, Mise à jour, Profil, Alarme de diagnostic, Débrogage/enfichage de modules, Défaillance du châssis ou de la station	0	SI_Format	OB_Class	OB_Nr	
	4	LADDR			
TI_Submodule - OB : Etat, Mise à jour, Profil	4			Emplacement	
	8	Spécificateur		0	
TI_DiagnosticInterrupt - OB : Alarme de diagnostic	4			IO_State	
	8	Circuit		MultiError	0
TI_PlugPullModule - OB : Débrogage/enfichage de modules	4			Event_Class	Fault_ID
	8	0		0	
TI_StationFailure - OB : Défaillance du châssis ou de la station	4			Event_Class	Fault_ID
	8	0		0	
Identique pour les OB : Etat, Mise à jour, Profil, Alarme de diagnostic, Débrogage/enfichage de modules, Défaillance du châssis ou de la station	12	0			
	16				
	20	adresse		slv_prfl	intr_type
	24	mémertos1	mémertos2	id	
	28 ¹	fabricant		instance	

¹ Les octets 28 à 31 (fabricant et instance) ne sont pas utilisés avec PROFIBUS.

Remarque

Reportez-vous au système d'information en ligne de STEP 7 pour obtenir des informations détaillées sur les données TINFO.

9.3.8 D_ACT_DP (Activer/désactiver les périphériques PROFINET IO)

L'instruction D_ACT_DP permet d'activer et de désactiver de manière ciblée des périphériques PROFINET IO configurés. Vous pouvez également déterminer si chaque périphérique PROFINET IO affecté est actuellement activé ou désactivé.

Remarque

L'instruction D_ACT_DP n'est utilisable qu'avec des périphériques PROFINET IO. Vous ne pouvez pas l'utiliser avec des esclaves PROFIBUS DP.

Tableau 9- 84 Instruction D_ACT_DP

CONT/LOG	SCL	Description
	<pre>"D_ACT_DP_DB" (req:=_bool_in_, mode:=_usint_in_, laddr:=_uint_in_, ret_val=>_int_out_, busy=>_bool_out_);</pre>	<p>Utilisez l'instruction D_ACT_DP pour activer et désactiver des périphériques PROFINET IO configurés et déterminer si chaque périphérique PROFINET IO affecté est actuellement activé ou désactivé.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "D_ACT_DP_SFB_DB" est le nom du DB d'instance.

L'instruction D_ACT_DP ne permet pas d'activer ou désactiver une passerelle de type IE/PB Link PN IO. Toutefois, si vous utilisez D_ACT_DP sur une telle passerelle, la CPU renvoie la valeur W#16#8093 (aucun objet matériel ne peut être activé ou désactivé pour l'adresse indiquée dans LADDR).

Remarque

Une tâche de désactivation ou d'activation nécessite plusieurs passages par le point de contrôle du cycle. C'est pourquoi vous ne pouvez pas attendre la fin d'une telle tâche dans une boucle programmée.

Description fonctionnelle

D_ACT_DP est une instruction asynchrone, ce qui signifie que le traitement de la tâche s'étend sur plusieurs exécutions de l'instruction D_ACT_DP. Vous démarrez la tâche en appelant "D_ACT_DP" avec REQ égal à 1.

Les paramètres de sortie "RET_VAL " et "BUSY " indiquent l'état de la tâche.

Application

Lorsque vous configurez, dans une CPU, des périphériques PROFINET IO qui ne sont pas réellement présents ou qui ne sont pas nécessaires actuellement, la CPU ne cesse d'accéder à ces périphériques PROFINET IO à intervalles réguliers. Une fois ces périphériques désactivés, la CPU cesse d'y accéder. L'événement d'erreur correspondant ne se produit plus.

Exemples

Dans le cadre de la production de machines en série, un constructeur dispose d'un grand nombre d'options en ce qui concerne les dispositifs. Pourtant, chaque machine livrée ne contient qu'une combinaison d'options sélectionnées.

Le constructeur configure chacune des options machines possibles en tant que périphérique PROFINET IO afin de pouvoir écrire et maintenir un programme utilisateur commun comportant toutes les options possibles. Vous utiliserez l'instruction D_ACT_DP pour désactiver tous les périphériques PROFINET IO qui n'existent pas à la mise en route de la machine.

On a une situation similaire pour les machines-outils qui sont équipées de nombreux outils mais qui n'en utilisent qu'un nombre réduit à un moment donné. Ces outils sont réalisés en tant que périphériques PROFINET IO. Avec D_ACT_DP, le programme utilisateur active les outils actuellement requis et désactive les outils qui seront utilisés plus tard.

Identification d'une tâche

Si vous avez démarré une tâche de désactivation ou d'activation et que vous appelez à nouveau l'instruction D_ACT_DP avant que la tâche ne soit achevée, le comportement de l'instruction varie selon que le nouvel appel concerne ou non la même tâche. Si le paramètre d'entrée LADDR est identique, cet appel est considéré comme un appel consécutif.

Désactivation de périphériques PROFINET IO

Si vous désactivez un périphérique PROFINET IO avec D_ACT_DP, ses sorties du processus prennent les valeurs de remplacement configurées ou la valeur 0 (état de sécurité). Le contrôleur PROFINET IO affecté cesse d'accéder à ce composant. Les LED de défaut du contrôleur PROFINET IO ou de la CPU ne signalent pas les périphériques PROFINET IO désactivés comme défaillants ou manquants.

La CPU actualise la mémoire image des entrées des périphériques PROFINET IO désactivés avec la valeur "0". Elle traite donc les périphériques PROFINET IO désactivés tout comme s'il s'agissait de périphériques PROFINET IO défaillants.

Si vous accédez directement dans votre programme aux données utilisateur d'un périphérique PROFINET IO précédemment désactivé, le comportement du système dépend du traitement d'erreurs sélectionné dans le bloc.

- En cas de traitement global des erreurs, le système inscrit un événement déclencheur d'erreur d'accès dans le tampon de diagnostic et reste à l'état MARCHE.
- Si le traitement d'erreur local dans le bloc a été activé, le système inscrit la cause de l'erreur dans la structure d'erreur. Vous pouvez accéder à la cause de l'erreur au moyen de l'instruction GET_ERROR_ID (Page 323).

Une erreur d'accès en lecture renvoie "0". Pour plus d'informations sur le traitement d'erreur, voir "Priorités d'exécution et mise en file d'attente des événements" (Page 106).

Si vous tentez d'accéder à un périphérique PROFINET IO désactivé à l'aide d'une instruction (telle que "RD_REC (Page 395)"), vous recevrez dans RET_VAL la même information d'erreur que pour un périphérique PROFINET IO non disponible.

La défaillance éventuelle d'une station PROFINET IO après sa désactivation avec D_ACT_DP n'est pas détectée par le système d'exploitation.

Activation de périphériques PROFINET IO

Si vous réactivez un périphérique PROFINET IO avec D_ACT_DP, le contrôleur PROFINET IO associé le configure et le paramètre (comme il le ferait au retour d'une station PROFINET IO défaillante). L'activation est achevée lorsque le composant est capable de transmettre des données utilisateur.

Si vous essayez d'activer un périphérique PROFINET IO inaccessible (parce qu'il a, par exemple, été physiquement coupé du bus) avec l'instruction D_ACT_DP, l'instruction renvoie le code d'erreur W#16#80A7 après expiration du temps de paramétrage configuré pour la périphérie décentralisée. Le périphérique PROFINET IO est activé et l'inaccessibilité du périphérique PROFINET IO activé provoque la signalisation correspondante dans le diagnostic système.

Lorsque le périphérique PROFINET IO redeviendra accessible, le système retrouvera son comportement standard.

Remarque

L'activation d'un périphérique PROFINET IO peut durer un certain temps. Si vous souhaitez annuler une tâche d'activation en cours, lancez D_ACT_DP avec la même valeur pour LADDR et MODE = 2. Répétez l'appel de D_ACT_DP avec MODE = 2 jusqu'à ce que l'annulation réussie de la tâche d'activation soit signalée par RET_VAL = 0.

Paramètres

Le tableau suivant présente les paramètres de l'instruction D_ACT_DP :

Paramètre	Déclaration	Type de données	Description
REQ	IN	Bool	Paramètre de commande déclenché sur niveau REQ = 1 : Procéder à l'activation/la désactivation
MODE	IN	USInt	Identificateur de tâche Valeurs possibles : <ul style="list-style-type: none"> • 0 : Demander si le composant adressé est activé ou désactivé (sortie dans le paramètre RET_VAL) • 1 : Activer le périphérique PROFINET IO • 2 : Désactiver le périphérique PROFINET IO
LADDR	IN	HW_DEVICE	Identification matérielle du périphérique PROFINET IO (HW_Device) Ce nombre peut être pris dans les propriétés du périphérique PROFINET IO dans la vue du réseau ou dans l'onglet "Constantes système" de la table de variables par défaut. Si l'identification pour le diagnostic de l'appareil et l'identification pour les transitions d'état de fonctionnement sont toutes deux indiquées, vous devez utiliser le code pour le diagnostic de l'appareil.
RET_VAL	OUT	Int	Si une erreur se produit pendant que le programme exécute l'instruction, la valeur en retour contient un code d'erreur.
BUSY	OUT	Bool	Code d'activité : <ul style="list-style-type: none"> • BUSY = 1 : La tâche est encore active. • BUSY = 0 : La tâche est achevée.

Paramètre RET_VAL

Code d'erreur* (W#16#...)	Explication
0000	Tâche achevée sans erreur.
0001	Le périphérique PROFINET IO est activé (ce code d'erreur n'est possible qu'avec MODE = 0).
0002	Le périphérique PROFINET IO est désactivé (ce code d'erreur n'est possible qu'avec MODE = 0).
7000	Premier appel avec REQ = 0 : La tâche indiquée dans LADDR n'est pas active ; BUSY a la valeur "0".
7001	Premier appel avec REQ = 1. Le programme a déclenché la tâche indiquée dans LADDR. BUSY a la valeur "1".
7002	Appel intermédiaire (REQ sans objet). La tâche activée est toujours active ; BUSY a la valeur "1".
8090	<ul style="list-style-type: none"> Vous n'avez pas configuré de module avec l'adresse indiquée dans LADDR. Vous utilisez votre CPU comme esclave I/périphérique I et vous avez indiqué une adresse de cet esclave I/périphérique I dans LADDR.
8092	La désactivation du périphérique PROFINET IO actuellement adressé (MODE = 2) ne peut pas être annulée par une activation (MODE = 1). Activez le composant ultérieurement.
8093	L'adresse indiquée dans LADDR n'appartient à aucun périphérique PROFINET IO qui puisse être activé ou désactivé ou bien le paramètre MODE est inconnu.
8094	Vous avez essayé d'activer un appareil qui est un partenaire possible pour une station avec ports alternants. Or, un autre appareil est déjà activé sur cette station avec ports alternants. L'appareil activé reste activé.
80A0	Erreur pendant la communication entre la CPU et le contrôleur IO
80A1	Impossible d'affecter des paramètres au composant adressé (ce code d'erreur n'est possible qu'avec MODE = 1) Remarque : L'instruction D_ACT_DP fournit l'information d'erreur si ce composant est de nouveau défaillant lors du paramétrage du périphérique activé. Si le paramétrage d'un seul module échoue, D_ACT_DP renvoie l'information d'erreur W#16#0000.
80A3	Le contrôleur PROFINET IO concerné ne prend pas cette fonction en charge.
80A4	La CPU ne prend pas cette fonction en charge pour un contrôleur PROFINET IO externe.
80A6	Erreur d'emplacement dans le périphérique PROFINET IO ; il n'est pas possible d'accéder à toutes les données utilisateur (ce code d'erreur n'est possible qu'avec MODE = 1). Remarque : D_ACT_DP ne renvoie cette information d'erreur qu'en cas de nouvelle défaillance du composant activé après le paramétrage et avant la fin de l'exécution de l'instruction D_ACT_DP. Si un seul module est indisponible, D_ACT_DP renvoie l'information d'erreur W#16#0000.
80A7	Dépassement du délai pendant l'activation : l'appareil distant est inaccessible ou vous avez donné une valeur trop faible au temps de paramétrage pour la périphérie centralisée et décentralisée. L'état de l'appareil distant est "activé", mais il est inaccessible.
80AA	Activation avec erreurs dans le périphérique PROFINET IO : différences dans la configuration
80AB	Activation avec erreurs dans le périphérique PROFINET IO : erreur de paramétrage

Code d'erreur* (W#16#...)	Explication
80AC	Activation avec erreurs dans le périphérique PROFINET IO : maintenance nécessaire
80C1	D_ACT_DP a démarré et se poursuit avec une autre adresse (ce code d'erreur est possible quand MODE = 1 et MODE = 2).
80C3	<ul style="list-style-type: none"> Erreur temporaire de ressource : la CPU traite actuellement le nombre maximum possible de tâches d'activation et de désactivation, à savoir 8 (ce code d'erreur n'est possible qu'avec MODE = 1 et MODE = 2). La CPU est occupée par la réception d'une configuration modifiée. Vous ne pouvez actuellement pas activer ou désactiver de périphériques PROFINET IO.
80C6	PROFINET : les tâches non récupérées par l'utilisateur sont rejetées au redémarrage.
Information d'erreur générale	Pour plus d'informations sur la manière d'accéder à l'erreur, voir l'instruction GET_ERROR_ID (Page 323).
* Les codes d'erreur dans l'éditeur de programmes peuvent être affichés sous forme de valeurs entières ou hexadécimales.	

9.3.9 Paramètre STATUS pour RDREC, WRREC et RALRM

Le paramètre de sortie STATUS contient des informations d'erreur sous forme d'un tableau ARRAY[1...4] OF BYTE ayant la structure suivante :

Tableau 9- 85 Tableau de sortie STATUS

Élément de tableau	Nom	Description
STATUS[1]	Function_Num	<ul style="list-style-type: none"> B#16#00 en l'absence d'erreur ID de fonction de PDU DPV1 : Si une erreur se produit, une opération logique OU est exécutée avec , B#16#80 (pour la lecture d'enregistrement : B#16#DE ; pour l'écriture d'enregistrement : B#16#DF). Si aucun élément de protocole DPV1 n'est utilisé, B#16#C0 est transmis.
STATUS[2]	Error_Decode	Emplacement du code d'erreur
STATUS[3]	Error_Code_1	Code d'erreur
STATUS[4]	Error_Code_2	Extension de code d'erreur spécifique du fabricant

Tableau 9- 86 Valeurs de STATUS[2]

Error_decode (B#16#....)	Source	Description
00 à 7F	CPU	Pas d'erreur ni d'avertissement
80	DPV1	Erreur selon CEI 61158-6
81 à 8F	CPU	B#16#8x montre une erreur dans le x-ième paramètre d'appel de l'instruction.
FE, FF	Profil DP	Erreur spécifique du profil

Tableau 9- 87 Valeurs de STATUS[3]

Error_decode (B#16#...)	Error_code_1 (B#16#...)	Explication (DVP1)	Description
00	00		Pas d'erreur ni d'avertissement
70	00	Réservé, rejet	Appel initial ; pas de transfert d'enregistrement actif
	01	Réservé, rejet	Appel initial ; un transfert d'enregistrement a commencé
	02	Réservé, rejet	Appel intermédiaire ; un transfert d'enregistrement est encore actif
80	90	Réservé, acceptation	Adresse de début logique invalide
	92	Réservé, acceptation	Type incorrect de pointeur Variant
	93	Réservé, acceptation	Le composant DP adressé via ID ou F_ID n'est pas configuré.
	96		"RALRM (Page 404)" ne peut pas fournir les informations de déclenchement d'OB, les informations de gestion, les informations d'en-tête ou les informations d'alarme additionnelles. Pour les OB suivants, vous pouvez utiliser l'instruction "DPNRM_DG (Page 426)" pour lire de manière asynchrone la trame de message de diagnostic en cours de l'esclave DP concerné (informations d'adresse dans les informations de déclenchement d'OB). <ul style="list-style-type: none"> • Alarme de processus (Page 97) • Etat (Page 103), Mise à jour (Page 104) ou Profil (Page 104) • Alarme de diagnostic (Page 99) • Débrochage/enfichage de modules (Page 101)
	A0	Erreur de lecture	Acquittement négatif pendant la lecture dans le module
	A1	Erreur d'écriture	Acquittement négatif pendant l'écriture dans le module
	A2	Défaillance du module	Erreur de protocole DP, couche 2 (par exemple, défaillance d'esclave ou problèmes de bus)
	A3	Réservé, acceptation	<ul style="list-style-type: none"> • PROFIBUS DP : Erreur de protocole DP avec DDLM (Direct Data Link Mapper) ou interface utilisateur/utilisateur • PROFINET IO : Erreur CM générale
	A4	Réservé, acceptation	Communication interrompue sur le bus de communication
	A5	Réservé, acceptation	-
	A7	Réservé, acceptation	Module ou esclave DP occupé (erreur temporaire)
	A8	Conflit de version	Le module ou l'esclave DP signale des versions non compatibles.
	A9	Fonction non prise en charge	Fonction non prise en charge par le module ou l'esclave DP
AA à AF	Spécifique utilisateur	Le module ou l'esclave DP signale une erreur spécifique du fabricant dans son application. Consultez la documentation du fabricant du module ou de l'esclave DP.	

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Explication (DVP1)	Description
	B0	Indice invalide	L'enregistrement est inconnu dans le module ; numéro d'enregistrement incorrect ≥ 256
	B1	Erreur de longueur d'écriture	L'information de longueur dans le paramètre RECORD est incorrecte. <ul style="list-style-type: none"> Pour "RALRM" : Erreur de longueur dans AINFO <p>Remarque : Reportez-vous au système d'information en ligne de STEP 7 pour obtenir des informations sur l'interprétation des mémoires tampons "AINFO" renvoyées.</p> <ul style="list-style-type: none"> Pour "RDREC (Page 395)" et "WRREC (Page 395)" : Erreur de longueur dans MLEN
	B2	Emplacement incorrect	L'emplacement configuré n'est pas occupé.
	B3	Conflit de type	Le type du module réel ne correspond pas au type de module spécifié.
	B4	Zone invalide	Le module ou l'esclave DP signale un accès à une zone invalide.
	B5	Conflit d'état	Le module ou l'esclave DP n'est pas prêt.
	B6	Accès refusé	Le module ou l'esclave DP refuse l'accès.
	B7	Plage invalide	Le module ou l'esclave DP signale une plage invalide pour un paramètre ou une valeur.
	B8	Paramètre invalide	Le module ou l'esclave DP signale un paramètre invalide.
	B9	Type invalide	Le module ou l'esclave DP signale un type invalide : <ul style="list-style-type: none"> Pour "RDREC (Page 395)" : Mémoire tampon trop petite (impossible de lire les sous-ensembles) Pour "WRREC (Page 395)" : Mémoire tampon trop petite (impossible d'écrire les sous-ensembles)
	BA à BF	Spécifique utilisateur	Le module ou l'esclave DP signale une erreur spécifique du fabricant lors de l'accès. Consultez la documentation du fabricant du module ou de l'esclave DP.
	C0	Conflit de limite de lecture	<ul style="list-style-type: none"> Pour "WRREC (Page 395)" : L'écriture des données n'est possible que lorsque la CPU est à l'état ARRET. Remarque : Cela signifie que les données ne peuvent pas être écrites par le programme utilisateur. Vous pouvez uniquement écrire les données en ligne avec une PG/un PC. Pour "RDREC (Page 395)" : Le module transmet l'enregistrement mais soit il n'y a pas de données, soit les données peuvent uniquement être lues lorsque la CPU est à l'état ARRET. Remarque : Si les données peuvent uniquement être lues lorsque la CPU est à l'état ARRET, aucune évaluation par le programme utilisateur n'est possible. Dans ce cas, vous pouvez uniquement lire les données en ligne avec une PG/un PC.

Error_decode (B#16#...)	Error_code_1 (B#16#...)	Explication (DVP1)	Description
	C1	Conflit de limite d'écriture	Les données de la précédente demande d'écriture dans le module pour le même enregistrement n'ont pas encore été traitées par le module.
	C2	Ressource occupée	Le module traite actuellement le nombre maximum de tâches possibles pour une CPU.
	C3	Ressource non disponible	Les ressources d'exploitation requises sont actuellement occupées.
	C4		Erreur temporaire interne. La tâche n'a pas pu être exécutée. Relancez la tâche. Si cette erreur se produit souvent, vérifiez que votre installation ne présente pas de sources d'interférence électrique.
	C5		Le module ou l'esclave DP n'est pas disponible.
	C6		Le transfert d'enregistrement a été annulé du fait d'une annulation de classe de priorité.
	C7		Tâche interrompue en raison d'un démarrage à chaud ou à froid du maître DP.
	C8 à CF		Le module ou l'esclave DP signale une erreur de ressource spécifique du fabricant. Consultez la documentation du fabricant du module ou de l'esclave DP.
	Dx	Spécifique utilisateur	Spécifique de l'esclave DP. Consultez la description de l'esclave DP.
81	00 à FF		Erreur dans le premier paramètre d'appel (pour "RALRM (Page 404)" : MODE)
	00		Mode de fonctionnement interdit
82	00 à FF		Erreur dans le deuxième paramètre d'appel
88	00 à FF		Erreur dans le huitième paramètre d'appel (pour "RALRM (Page 404)" : TINFO) Remarque : Reportez-vous au système d'information en ligne de STEP 7 pour obtenir des informations sur l'interprétation des mémoires tampons "TINFO" renvoyées.
	01		ID de syntaxe illicite
	23		Structure de quantité dépassée ou zone de destination trop petite
	24		ID de plage illicite
	32		Numéro de DB/DI en dehors de la plage utilisateur
	3A		Numéro de DB/DI nul pour l'ID de zone DB/DI ou DB/DI indiqué inexistant

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Explication (DVP1)	Description
89	00 à FF		Erreur dans le neuvième paramètre d'appel (pour "RALRM (Page 404)" : AINFO) Remarque : Reportez-vous au système d'information en ligne de STEP 7 pour obtenir des informations sur l'interprétation des mémoires tampons "AINFO" renvoyées.
	01		ID de syntaxe illicite
	23		Structure de quantité dépassée ou zone de destination trop petite
	24		ID de plage illicite
	32		Numéro de DB/DI en dehors de la plage utilisateur
	3A		Numéro de DB/DI nul pour l'ID de zone DB/DI ou DB/DI indiqué inexistant
8A	00 à FF		Erreur dans le dixième paramètre d'appel
8F	00 à FF		Erreur dans le quinzième paramètre d'appel
FE, FF	00 à FF		Erreur spécifique du profil

Élément de tableau STATUS[4]

En présence d'erreurs DPV1, le maître DP transmet STATUS[4] à la CPU et à l'instruction. En l'absence d'erreur DPV1, cette valeur est mise à 0, aux exceptions suivantes près pour RDREC :

- STATUS[4] contient la longueur de la zone cible de RECORD si MLEN > longueur de la zone cible de RECORD.
- STATUS[4] est égal à MLEN si longueur réelle de l'enregistrement < MLEN < longueur de la zone cible de RECORD.
- STATUS[4] est égal à 0 si STATUS[4] > 255 ; devrait être défini.

Dans PROFINET IO, STATUS[4] a la valeur 0.

9.3.10 Autres instructions

9.3.10.1 DPRD_DAT et DPWR_DAT (Lire/écrire des données cohérentes)

DPRD_DAT (Lire des données cohérentes) permet de lire un ou plusieurs octets de données de manière cohérente et DPWR_DAT (Écrire des données cohérentes) permet de transférer un ou plusieurs octets de données de manière cohérente. Vous pouvez utiliser les instructions DPRD_DAT et DPWR_DAT avec PROFINET et PROFIBUS.

Tableau 9- 88 Instructions DPRD_DAT et DPWR_DAT

CONT/LOG	SCL	Description
	<pre>ret_val := DPRD_DAT(laddr:=_word_in_, record=>_variant_out_);</pre>	<p>Utilisez l'instruction DPRD_DAT pour lire un ou plusieurs octets de données dans des modules ou sous-modules à l'un des emplacements suivants :</p> <ul style="list-style-type: none"> • E/S de base locales • Esclave DP • Périphérique PROFINET I/O <p>La CPU transfère les données lues de manière cohérente. Si aucune erreur ne survient pendant le transfert de données, la CPU entre les données lues dans la zone cible définie par le paramètre RECORD. La zone cible doit avoir la même longueur que celle que vous avez configurée avec STEP 7 pour le module sélectionné. Lorsque vous exécutez l'instruction DPRD_DAT, vous pouvez accéder aux données d'un seul module ou sous-module. Le transfert démarre à l'adresse de début configurée.</p>
	<pre>ret_val := DPWR_DAT(laddr:=_word_in_, record:=_variant_in_);</pre>	<p>Utiliser l'instruction DPWR_DAT pour transférer les données dans RECORD de manière cohérente vers les emplacements suivants :</p> <ul style="list-style-type: none"> • Module ou sous-module adressé dans la base locale • Esclave DP standard • Périphérique PROFINET I/O <p>La zone source doit avoir la même longueur que celle que vous avez configurée avec STEP 7 pour le module ou le sous-module sélectionné.</p>

- La CPU S7-1200 prend en charge la lecture ou l'écriture cohérente de la périphérie pour 1, 2 ou 4 octets. Utilisez l'instruction DPRD_DAT pour lire et l'instruction DPWR_DAT pour écrire de manière cohérente des données ayant des longueurs autres que 1, 2 ou 4 octets.
- Vous pouvez utiliser ces instructions pour des zones de données de un ou plusieurs octets. Le code d'erreur W#16#8090 est renvoyé si l'accès est refusé.
- PROFINET prend en charge jusqu'à 1 024 octets de données cohérentes. Vous n'avez pas besoin d'utiliser ces instructions pour des transferts cohérents entre le S7-1200 et les périphériques PROFINET.

Remarque

Si vous utilisez les instructions DPRD_DAT et DPWR_DAT avec des données cohérentes, vous devez supprimer ces dernières de l'actualisation automatique de la mémoire image. Reportez-vous à "Concepts API : Exécution du programme utilisateur" (Page 85) pour plus d'informations.

Tableau 9- 89 Paramètres

Paramètre	Déclaration	Type de données	Description
LADDR	IN	HW_IO (Word)	ID matérielle du module dans lequel les données doivent être lues (DPRD_DAT) ID matérielle du module dans lequel les données doivent être écrites (DPWR_DAT) Vous trouverez l'ID matérielle dans les propriétés du module dans la vue des appareils ou dans les constantes système.
RECORD	OUT	Variante	Zone de destination pour les données utilisateur lues (DPRD_DAT) ou zone source pour les données utilisateur à écrire (DPWR_DAT). Cette zone doit être exactement de la taille que vous avez configurée pour le module sélectionné avec STEP 7.
RET_VAL	OUT	Int	Si une erreur se produit pendant que la fonction est active, la valeur en retour contient un code d'erreur.

Fonctionnement de DPRD_DAT

Utilisez le paramètre LADDR pour sélectionner le module de l'esclave DP norme/du périphérique PROFINET IO. Si une erreur d'accès au module adressé se produit, le code d'erreur W#16#8090 est renvoyé.

Le paramètre RECORD permet de définir la zone cible pour les données lues.

- La zone cible doit avoir au moins la longueur des entrées du module sélectionné. Seules les entrées sont transmises ; les autres octets ne sont pas pris en compte. Si la lecture se fait dans un esclave DP norme ayant une configuration modulaire ou plusieurs identificateurs DP, vous pouvez uniquement accéder aux données d'un module de l'identification matérielle configurée pour chaque appel de l'instruction DPRD_DAT. Si la zone cible sélectionnée est trop petite, le code d'erreur W#16#80B1 est fourni dans le paramètre RET_VAL.
- Vous pouvez utiliser les types de données suivants : Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UInt. L'utilisation de ces types de données dans une structure de données UDT (type de données utilisateur) de genre ARRAY ou STRUCT est autorisée.
- Le type de données STRING n'est pas pris en charge.
- En l'absence d'erreur pendant la transmission des données, les données lues sont reportées dans la zone cible définie dans le paramètre RECORD.

Fonctionnement de DPWR_DAT

Utilisez le paramètre LADDR pour sélectionner le module de l'esclave DP norme/du périphérique PROFINET IO. Si une erreur d'accès au module adressé se produit, le code d'erreur W#16#8090 est renvoyé.

Utilisez le paramètre RECORD pour définir la zone source des données à écrire :

- La zone source doit avoir au moins la longueur des sorties du module sélectionné. Seules les sorties sont transmises ; les autres octets ne sont pas pris en compte. Si la zone source dans le paramètre RECORD est plus longue que les sorties du module configuré, les données sont transférées jusqu'à la longueur maximale des sorties uniquement. Si la zone source dans le paramètre RECORD est plus petite que les sorties du module configuré, le code d'erreur W#16#80B1 est transmis au paramètre RET_VAL.
- Vous pouvez utiliser les types de données suivants : Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt. L'utilisation de ces types de données dans une structure de données UDT (type de données utilisateur) de genre ARRAY ou STRUCT est autorisée.
- Le type de données STRING n'est pas pris en charge.
- Les données sont transférées de manière synchrone, c'est-à-dire que l'opération d'écriture est achevée à l'achèvement de l'instruction.

Codes d'erreur

Tableau 9- 90 Codes d'erreur de DPRD_DAT et DPWR_DAT

Code d'erreur ¹	Description
0000	Pas d'erreur
8090	L'un des cas suivants s'est produit : <ul style="list-style-type: none"> • Vous n'avez pas configuré de module pour l'adresse de base logique indiquée. • Vous n'avez pas tenu compte de la restriction concernant la longueur des données cohérentes. • Vous n'avez pas entré l'adresse de début dans le paramètre LADDR en format hexadécimal.
8092	Le paramètre RECORD prend en charge les types de données suivants : Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt, and arrays of these types.
8093	Il n'existe pas de module DP/périphérique PROFINET IO dans lequel vous pouvez lire (DPRD_DAT) ou écrire (DPWR_DAT) des données cohérentes à l'adresse logique indiquée dans LADDR.
80A0	Erreur d'accès détectée pendant l'accès à des périphériques IO (DPRD_DAT).
80B1	La longueur de la zone de destination (DPRD_DAT) ou source (DPWR_DAT) indiquée n'est pas identique à la longueur des données utilisateur configurée avec STEP 7 Basic.
80B2	Erreur système avec le coupleur DP externe (DPRD_DAT et DPWR_DAT)

¹ Utilisez le type de données Word lors de l'affichage des codes d'erreur de DPRD_DAT et DPWR_DAT.

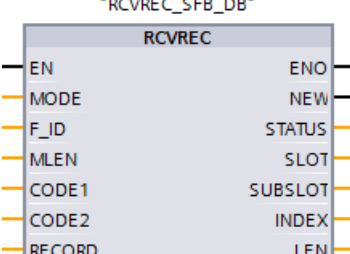
Remarque

Si vous accédez à des esclaves DPV1, les informations d'erreur de ces esclaves peuvent être transmises du maître DP à l'instruction.

9.3.10.2 RCVREC (Recevoir l'enregistrement - périphérique I/esclave I)

Un périphérique I (I-device) peut recevoir un enregistrement provenant d'une commande de niveau supérieur. La réception se fait dans le programme utilisateur avec l'instruction RCVREC (Recevoir l'enregistrement).

Tableau 9- 91 Instruction RCVREC

CONT/LOG	SCL	Description
<p style="text-align: center;">%DB1 "RCVREC_SFB_DB"</p> 	<pre>"RCVREC_SFB_DB" (mode:=_int_in_, F_ID:=_uint_in_, mlen:=_uint_in_, code1:=_byte_in_, code2:=_byte_in_, new=>_bool_out_, status=>_dword_out_, slot=>_uint_out_, subslot=>_uint_out_, index=>_uint_out_, len=>_uint_out_, rec- ord:=_variant_inout_);</pre>	Utilisez l'instruction RCVREC pour recevoir un enregistrement en provenance d'une commande de niveau supérieur.

- ¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- ² Dans l'exemple SCL, "RCVREC_SFB_DB" est le nom du DB d'instance.

L'instruction présente les modes de fonctionnement suivants :

- Vérifier si le périphérique I présente une demande de réception d'enregistrement
- Mettre l'enregistrement à disposition dans les paramètres de sortie
- Envoyer une réponse à la commande de niveau supérieur

Vous déterminez le mode de fonctionnement exécuté par l'instruction à l'aide du paramètre d'entrée MODE (voir ci-après).

Le périphérique I doit être à l'état MARCHE ou MISE EN ROUTE.

Avec MLEN, vous indiquez le nombre maximum d'octets que vous souhaitez recevoir. La longueur sélectionnée de la zone cible RECORD doit donc être d'au moins MLEN octets.

Si un enregistrement a été reçu (MODE = 1 ou MODE = 2), le paramètre de sortie NEW indique que l'enregistrement est stocké dans RECORD. Veillez à ce que RECORD ait une longueur suffisante. Le paramètre de sortie LEN contient la longueur réelle de l'enregistrement reçu, exprimée en octets.

Mettez CODE1 et CODE2 à zéro pour envoyer une réponse positive à la commande de niveau supérieur. Si l'enregistrement reçu doit être refusé, entrez la réponse négative à envoyer à la commande de niveau supérieur dans "Error Code 1" de CODE1 et dans "Error Code 2" de CODE2.

Remarque

Si le périphérique I a reçu une demande pour la réception d'un enregistrement, vous devez détecter l'arrivée de cette demande dans le délai imparti. Une fois la demande détectée, vous devez envoyer une réponse à la commande de niveau supérieur dans ce même délai. Dans le cas contraire, une erreur de dépassement du délai imparti se produit dans le périphérique I, avec pour effet que le système d'exploitation du périphérique I envoie une réponse négative à la commande de niveau supérieur. Pour plus d'informations sur la valeur du délai, voir les caractéristiques de votre CPU.

Le paramètre de sortie STATUS reçoit les informations d'erreur après l'apparition d'une erreur.

Modes de fonctionnement

Vous déterminez le mode de fonctionnement de l'instruction RCVREC à l'aide du paramètre d'entrée MODE. Les modes de fonctionnement possibles sont présentés dans le tableau suivant :

MODE	Signification
0	Vérifier s'il existe une demande de réception d'enregistrement Si un enregistrement provenant d'une commande de niveau supérieur est présent dans le périphérique I, l'instruction écrit uniquement dans les paramètres de sortie NEW, SLOT, SUBSLOT, INDEX et LEN. Si vous appelez l'instruction plusieurs fois avec MODE = 0, le paramètre de sortie fait référence à cette même demande unique.
1	Recevoir un enregistrement pour un sous-emplacement quelconque du périphérique I Si un enregistrement provenant d'une commande de niveau supérieur est présent dans le périphérique I pour un sous-emplacement quelconque du périphérique I, l'instruction écrit dans le paramètre de sortie et transfère l'enregistrement dans le paramètre RECORD.
2	Recevoir un enregistrement pour un sous-emplacement spécifique du périphérique I Si un enregistrement provenant d'une commande de niveau supérieur est présent dans le périphérique I pour un sous-emplacement spécifique du périphérique I, l'instruction écrit dans le paramètre de sortie et transfère l'enregistrement dans le paramètre RECORD.
3	Envoyer une réponse positive à la commande de niveau supérieur L'instruction vérifie la demande de la commande de niveau supérieur pour la réception d'un enregistrement, accepte l'enregistrement existant et envoie un acquittement positif à la commande de niveau supérieur.
4	Envoyer une réponse négative à la commande de niveau supérieur L'instruction vérifie la demande de la commande de niveau supérieur pour la réception d'un enregistrement, rejette l'enregistrement existant et envoie un acquittement négatif à la commande de niveau supérieur. Entrez le motif du rejet dans les paramètres d'entrée CODE1 et CODE2.

Remarque

Après l'arrivée d'un enregistrement (NEW = 1), vous devez appeler l'instruction RCVREC deux fois afin d'assurer son traitement intégral. Vous devez respecter l'ordre suivant :

- Premier appel avec MODE = 1 ou MODE = 2
- Deuxième appel avec MODE = 3 ou MODE = 4

Paramètres

Le tableau suivant présente les paramètres de l'instruction RCVREC :

Paramètre	Déclaration	Type de données	Description
MODE	IN	Int	Mode
F_ID	IN	HW_SUBMODULE	Sous-emplacement dans la zone de transfert du périphérique I pour l'enregistrement à recevoir (uniquement pertinent pour MODE = 2). Le mot de poids fort est toujours à 0.
MLEN	IN	Int	Longueur maximale en octets de l'enregistrement à recevoir
CODE1	IN	Byte	Zéro (pour MODE = 3) et/ou "Error Code 1" (pour MODE = 4)
CODE2	IN	Byte	Zéro (pour MODE = 3) et/ou "Error Code 2" (pour MODE = 4)
NEW	OUT	Bool	<ul style="list-style-type: none"> • MODE = 0 : nouvel enregistrement reçu • MODE = 1 ou 2 : enregistrement transféré dans RECORD
STATUS	OUT	DWord	Information d'erreur. Pour plus d'informations, voir le paramètre STATUS (Page 413).
SLOT	OUT	HW_SUBMODULE	Identique à F_ID
SUBSLOT	OUT	HW_SUBMODULE	Identique à F_ID
INDEX	OUT	UInt	Numéro de l'enregistrement reçu
LEN	OUT	UInt	Longueur de l'enregistrement reçu
RECORD	IN_OUT	Variant	Zone cible pour l'enregistrement reçu

9.3.10.3 PRVREC (Mettre l'enregistrement à disposition - périphérique I/esclave I)

Un périphérique I (I-device) peut recevoir une demande de mise à disposition d'un enregistrement de la part d'une commande de niveau supérieur. La périphérie I met l'enregistrement à disposition dans le programme utilisateur avec l'instruction PRVREC (Mettre l'enregistrement à disposition).

Tableau 9- 92 Instruction PRVREC

CONT/LOG	SCL	Description
<p style="text-align: center;">%DB2 "PRVREC_SFB_DB"</p>	<pre>"PRVREC_SFB_DB" (mode:=_int_in_, F_ID:=_uint_in_, code1:=_byte_in_, code2:=_byte_in_, len:=_uint_in_, new=>_bool_out_, status=>_dword_out_, slot=>_uint_out_, subslot=>_uint_out_, index=>_uint_out_, rlen=>_uint_out_, rec- ord:=_variant_inout_);</pre>	<p>Utilisez l'instruction PRVREC pour mettre à disposition un enregistrement sur demande d'une commande de niveau supérieur.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "PRVREC_SFB_DB" est le nom du DB d'instance.

L'instruction présente les modes de fonctionnement suivants :

- Vérifier si le périphérique I présente une demande de mise à disposition d'enregistrement
- Transmettre l'enregistrement demandé à la commande de niveau supérieur
- Envoyer une réponse à la commande de niveau supérieur

Vous déterminez le mode de fonctionnement exécuté par l'instruction à l'aide du paramètre d'entrée MODE (voir ci-après).

Le périphérique I doit être à l'état MARCHE ou MISE EN ROUTE.

Indiquez dans LEN le nombre maximum d'octets que l'enregistrement à envoyer doit avoir. La longueur sélectionnée de la zone cible RECORD doit donc être d'au moins LEN octets.

S'il existe une demande de mise à disposition d'un enregistrement (MODE = 0), le paramètre de sortie NEW prend la valeur TRUE.

Si la demande de mise à disposition d'un enregistrement est acceptée, inscrivez zéro dans CODE1 et CODE2 pour la réponse positive à la commande de niveau supérieur et fournissez l'enregistrement demandé dans RECORD. Si la demande de mise à disposition d'un enregistrement doit être refusée, entrez la réponse négative à envoyer à la commande de niveau supérieur dans "Error Code 1" de CODE1 et dans "Error Code 2" de CODE2.

Remarque

Si le périphérique I a reçu une demande de mise à disposition d'un enregistrement, vous devez détecter l'arrivée de cette demande dans le délai imparti. Une fois la demande détectée, vous devez envoyer une réponse à la commande de niveau supérieur dans ce même délai. Dans le cas contraire, une erreur de dépassement du délai imparti se produit dans le périphérique I, avec pour effet que le système d'exploitation du périphérique I envoie une réponse négative à la commande de niveau supérieur. Pour plus d'informations sur la valeur du délai, voir les caractéristiques de votre CPU.

Le paramètre de sortie STATUS reçoit les informations d'erreur après l'apparition d'une erreur.

Modes de fonctionnement

Vous déterminez le mode de fonctionnement de l'instruction PRVREC à l'aide du paramètre d'entrée MODE. Les modes de fonctionnement possibles sont présentés dans le tableau suivant :

MODE	Signification
0	Vérifier s'il existe une demande de mise à disposition d'un enregistrement Si une demande de mise à disposition d'un enregistrement de la part d'une commande de niveau supérieur est présente dans le périphérique I, l'instruction écrit uniquement dans les paramètres de sortie NEW, SLOT, SUBSLOT, INDEX et RLEN. Si vous appelez l'instruction plusieurs fois avec MODE = 0, le paramètre de sortie fait référence à cette même demande unique.
1	Recevoir une demande de mise à disposition d'un enregistrement pour un sous-emplacement quelconque du périphérique I Si une telle demande de la part d'une commande de niveau supérieur est présente dans le périphérique I pour un sous-emplacement quelconque du périphérique I, l'instruction écrit dans le paramètre de sortie.
2	Recevoir une demande de mise à disposition d'un enregistrement pour un sous-emplacement spécifique du périphérique I Si une telle demande de la part d'une commande de niveau supérieur est présente dans le périphérique I pour un sous-emplacement spécifique du périphérique I, l'instruction écrit dans le paramètre de sortie.
3	Mettre l'enregistrement à disposition et envoyer une réponse positive à l'automate L'instruction vérifie la demande de la commande de niveau supérieur pour la mise à disposition d'un enregistrement, met l'enregistrement demandé à disposition dans RECORD et envoie un acquittement positif à la commande de niveau supérieur.
4	Envoyer une réponse négative à la commande de niveau supérieur L'instruction vérifie la demande de la commande de niveau supérieur pour la mise à disposition d'un enregistrement, la rejette et envoie un acquittement négatif à l'automate. Entrez le motif du rejet dans les paramètres d'entrée CODE1 et CODE2.

Remarque

Après l'arrivée d'une demande (NEW = 1), vous devez appeler l'instruction PRVREC deux fois afin d'assurer son traitement intégral. Vous devez respecter l'ordre suivant :

- Premier appel avec MODE = 1 ou MODE = 2
- Deuxième appel avec MODE = 3 ou MODE = 4

Paramètres

Le tableau suivant présente les paramètres de l'instruction PRVREC :

Paramètre	Déclaration	Type de données	Description
MODE	IN	Int	Mode
F_ID	IN	HW_SUBMODULE	Sous-emplacement dans la zone de transfert du périphérique I pour l'enregistrement à envoyer (uniquement pertinent pour MODE = 2). Le mot de poids fort est toujours à 0.
CODE1	IN	Byte	Zéro (pour MODE = 3) et/ou "Error Code 1" (pour MODE = 4)
CODE2	IN	Byte	Zéro (pour MODE = 3) et/ou "Error Code 2" (pour MODE = 4)
LEN	IN	UInt	Longueur maximale en octets de l'enregistrement à envoyer
NEW	OUT	Bool	Un nouvel enregistrement a été demandé par la commande de niveau supérieur.
STATUS	OUT	DWord	Information d'erreur. Pour plus d'informations, voir le paramètre STATUS (Page 413).
SLOT	OUT	HW_SUBMODULE	Identique à F_ID
SUBSLOT	OUT	HW_SUBMODULE	Identique à F_ID
INDEX	OUT	UInt	Numéro de l'enregistrement à envoyer
RLEN	OUT	UInt	Longueur de l'enregistrement à envoyer
RECORD	IN_OUT	Variant	Enregistrement mis à disposition

9.3.10.4 DPNRM_DG (Lire des données de diagnostic d'un esclave PROFIBUS DP)

Vous pouvez utiliser l'instruction DPNRM_DG (Lecture de données de diagnostic) avec PROFIBUS.

Tableau 9- 93 Instruction DPNRM_DG

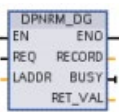
CONT/LOG	SCL	Description
	<pre>ret_val := DPNRM_DG(req:= bool_in_, laddr:= word_in_, record=>_variant_out_, busy=> bool_out);</pre>	Utilisez l'instruction DPNRM_DG pour lire les données de diagnostic en cours d'un esclave DP dans le format spécifié par EN 50 170 Volume 2, PROFIBUS. Les données qui ont été lues sont entrées dans la zone de destination indiquée par RECORD lorsque le transfert de données s'est effectué sans erreur.

Tableau 9- 94 Types de données pour les paramètres de l'instruction DPNRM_DG

Paramètre et type		Type de données	Description
REQ	IN	Bool	REQ=1 : Demande de lecture
LADDR	IN	HW_DPSLAVE	Adresse de diagnostic configurée de l'esclave DP : doit être l'adresse de la station et non celle du périphérique I/O Sélectionnez la station (et non l'image du périphérique) dans la vue de réseau de la configuration de l'appareil afin de déterminer l'adresse de diagnostic. Entrez les adresses au format hexadécimal. L'adresse de diagnostic 1022 signifie par exemple LADDR:=W#16#3FE.
RET_VAL	OUT	Int	Si une erreur se produit pendant que la fonction est active, la valeur en retour contient un code d'erreur. En l'absence d'erreur, la longueur des données effectivement transférées est fournie dans RET_VAL.
RECORD	OUT	Variant	Zone de destination pour les données de diagnostic qui ont été lues. La longueur minimum de l'enregistrement à lire (ou de la zone de destination) est 6 octets. La longueur maximum de l'enregistrement à envoyer est 240 octets. Les esclaves norme peuvent fournir plus de 240 octets de données de diagnostic, avec un maximum de 244 octets. Dans ce cas, les 240 premiers octets sont transférés dans la zone de destination et le bit de débordement est mis à 1 dans les données.
BUSY	OUT	Bool	BUSY=1 : La tâche de lecture n'est pas encore achevée.

Vous démarrez la tâche de lecture en mettant le paramètre d'entrée REQ à 1 dans l'appel de l'instruction DPNRM_DG. La tâche de lecture est exécutée de manière asynchrone ; en d'autres termes, elle nécessite plusieurs appels de l'instruction DPNRM_DG. L'état de la tâche est indiqué par les paramètres de sortie RET_VAL et BUSY.

Tableau 9- 95 Structure des données de diagnostic esclave

Octet	Description
0	Etat de station 1
1	Etat de station 2
2	Etat de station 3
3	Numéro de la station maître
4	ID de fournisseur (octet de poids fort)
5	ID de fournisseur (octet de poids faible)
6 ...	Informations de diagnostic supplémentaires spécifiques de l'esclave

Tableau 9- 96 Codes d'erreur de l'instruction DPNRM_DG

Code d'erreur	Description	Restriction
0000	Pas d'erreur	-
7000	Premier appel avec REQ=0 : Pas de transfert de données actif ; BUSY a la valeur 0.	-
7001	Premier appel avec REQ=1 : Pas de transfert de données actif ; BUSY a la valeur 1.	E/S décentralisées

Code d'erreur	Description	Restriction
7002	Appel intermédiaire (REQ non significatif) : Un transfert de données est déjà actif ; BUSY a la valeur 1.	E/S décentralisées
8090	Adresse de base logique indiquée invalide : Il n'y a pas d'adresse de base.	-
8092	Le paramètre RECORD prend en charge les types de données suivants : Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt, and arrays of these types.	-
8093	<ul style="list-style-type: none"> Cette instruction n'est pas autorisée pour le module indiqué par LADDR (les modules S7 DP pour le S7-1200 sont autorisés). LADDR spécifie le périphérique I/O au lieu de spécifier la station. Sélectionnez la station (et non l'image du périphérique) dans la vue de réseau de la configuration de l'appareil afin de déterminer l'adresse de diagnostic pour LADDR. 	-
80A2	<ul style="list-style-type: none"> Erreur de protocole DP, couche 2 (par exemple, défaillance d'esclave ou problèmes de bus) Pour l'ET200S, l'enregistrement ne peut pas être lu en mode DPV0. 	E/S décentralisées
80A3	Erreur de protocole DP avec l'interface utilisateur/utilisateur	E/S décentralisées
80A4	Problème de communication sur le bus de communication	L'erreur se produit entre la CPU et le coupleur DP externe.
80B0	<ul style="list-style-type: none"> L'instruction n'est pas possible pour le type de module. Le module ne reconnaît pas l'enregistrement. Le numéro d'enregistrement 241 est interdit. 	-
80B1	La longueur indiquée dans le paramètre RECORD est incorrecte.	longueur indiquée > longueur de l'enregistrement
80B2	L'emplacement configuré n'est pas occupé.	-
80B3	Le type du module réel ne correspond pas au type de module demandé.	-
80C0	Il n'y a pas d'informations de diagnostic.	-
80C1	Les données de la précédente tâche d'écriture dans le module pour le même enregistrement n'ont pas encore été traitées par le module.	-
80C2	Le module traite actuellement le nombre maximum de tâches possibles pour une CPU.	-
80C3	Les ressources requises (mémoire, etc.) sont actuellement occupées.	-
80C4	Erreur temporaire interne. La tâche n'a pas pu être traitée. Relancez la tâche. Si cette erreur se produit fréquemment, vérifiez que votre système ne présente pas de sources d'interférence électrique.	-
80C5	E/S décentralisées non disponibles	E/S décentralisées
80C6	Le transfert d'enregistrement a été arrêté en raison d'une annulation de classe de priorité (redémarrage ou arrière-plan).	E/S décentralisées
8xyy ¹	Codes d'erreur généraux	

Reportez-vous à "Instructions avancées, périphérie décentralisée : Informations d'erreur pour RDREC, WRREC et RALRM" (Page 413) pour plus d'informations sur les codes d'erreur généraux.

9.4 PROFlenergy

PROFlenergy est un profil de gestion d'énergie avec PROFINET indépendant des appareils et des constructeurs. PROFlenergy permet d'arrêter les équipements de manière coordonnée et centralisée afin de réduire la consommation d'électricité pendant les arrêts de la production et les interruptions non prévues.

Le contrôleur PROFINET IO met hors tension les périphériques PROFINET/modules d'alimentation au moyen de commandes spéciales dans le programme utilisateur. Aucun matériel supplémentaire n'est requis. Les périphériques PROFINET interprètent les commandes PROFlenergy directement.

La CPU S7-1200 ne prend pas en charge la fonctionnalité de contrôleur PE. La CPU S7-1200 ne peut agir qu'en tant qu'unité PROFlenergy (avec la fonctionnalité I-device).

Contrôleur PROFlenergy (contrôleur PE)

Le contrôleur PE est une CPU de haut niveau (par exemple, une S7-1500) qui active ou désactive l'état de repos sur les appareils subordonnés. Le contrôleur PE désactive et réactive des composants de production spécifiques ou des lignes de production complètes avec le programme utilisateur. Les appareils subordonnés reçoivent des commandes du programme utilisateur par le biais d'instructions correspondantes (blocs fonctionnels).

Le programme utilisateur envoie les commandes via le protocole de communication PROFINET. La commande PE peut être soit un ordre de réglage pour faire passer une entité PE en mode d'économie d'énergie, soit un ordre de lecture de valeur d'état ou de mesure.

Utilisez l'instruction PE_I_DEV pour obtenir des données d'un module. Le programme utilisateur doit déterminer quelles informations sont demandées par le contrôleur PE et les retrouver dans le module d'énergie en utilisant les enregistrements de données. Le module en tant que tel ne prend pas directement en charge les ordres PE. Le module enregistre les informations de mesurage de l'énergie dans un espace partagé et la CPU subordonnée (par exemple, une S7-1200) déclenche l'instruction PE_I_DEV afin de la renvoyer au contrôleur PE.

Entité PROFlenergy (entité PE)

L'entité PE (par exemple, un S7-1200) reçoit les ordres PROFlenergy du contrôleur PE (par exemple, un S7-1500) et les exécute (par exemple, en renvoyant une valeur mesurée ou en activant un mode d'économie d'énergie). La mise en œuvre de l'entité PE dans un appareil compatible PROFlenergy dépend de l'appareil et du constructeur.

Informations de référence

Pour plus d'informations sur PROFlenergy, voir l'aide en ligne STEP 7 de TIA Portal. Vous trouverez des exemples d'utilisation des instructions PROFlenergy dans l'entrée "PROFlenergy - Gestion maîtrisée de l'énergie avec SIMATIC S7 (<http://support.automation.siemens.com/WW/view/fr/41986454>)" sur le site Industry Online Support.

9.5 Alarmes

9.5.1 Instructions ATTACH et DETACH (Affecter/dissocier un OB et un événement déclencheur d'alarme)

Vous pouvez activer et désactiver des sous-programmes d'alarme déclenchés sur événement par le biais des instructions ATTACH et DETACH.

Tableau 9- 97 Instructions ATTACH et DETACH

CONT/LOG	SCL	Description
	<pre>ret_val := ATTACH(ob_nr:=_int_in_, event:=_event_att_in_, add:=_bool_in_);</pre>	ATTACH active l'exécution d'un sous-programme d'OB d'alarme pour un événement d'alarme de processus.
	<pre>ret_val := DETACH(ob_nr:=_int_in_, event:=_event_att_in_);</pre>	DETACH désactive l'exécution d'un sous-programme d'OB d'alarme pour un événement d'alarme de processus.

Tableau 9- 98 Types de données pour les paramètres

Paramètre et type	Type de données	Description
OB_NR	IN	OB_ATT
EVENT	IN	EVENT_ATT
ADD (ATTACH unique- ment)	IN	Bool
RET_VAL	OUT	Int

Événements alarme de processus

La CPU prend en charge les événements alarme de processus suivants :

- Événements front montant : 12 premières entrées TOR intégrées de la CPU (DIa.0 à DIb.3) et toutes les entrées TOR SB
 - Un front montant se produit lorsque l'entrée TOR passe de l'état désactivé à l'état activé en réaction à un changement du signal provenant de l'appareil de terrain connecté à l'entrée.
- Événements front descendant : 12 premières entrées TOR intégrées de la CPU (DIa.0 à DIb.3) et toutes les entrées TOR SB
 - Un front descendant se produit lorsque l'entrée TOR passe de l'état activé à l'état désactivé.
- Événements valeur en cours = valeur de référence (CV = RV) de compteur rapide (compteurs HSC 1 à 6)
 - Une alarme CV = RV pour un compteur rapide est générée lorsque la valeur de comptage en cours passe d'une valeur adjacente à la valeur correspondant exactement à une valeur de référence définie précédemment.
- Événements inversion du sens HSC (compteurs HSC 1 à 6)
 - Un événement inversion du sens se produit lorsque le système détecte que le HSC est passé de l'incréméntation à la décrémentation ou de la décrémentation à l'incréméntation.
- Événements réinitialisation externe HSC (compteurs HSC 1 à 6)
 - Certains modes HSC autorisent la définition d'une entrée TOR en tant qu'entrée de réinitialisation externe qui est utilisée pour remettre à zéro la valeur de comptage du compteur rapide. Un événement réinitialisation externe se produit pour un tel compteur rapide lorsque cette entrée passe de l'état désactivé à l'état activé.

Activation des événements alarme de processus dans la configuration des appareils

Les alarmes de processus doivent être activées pendant la configuration des appareils. Vous devez cocher la case d'activation de l'événement dans la configuration d'appareil pour une voie d'entrée TOR ou un compteur rapide si vous voulez associer cet événement pendant la configuration ou au moment de l'exécution.

Options à cocher dans la configuration d'appareil de l'automate :

- Entrée TOR
 - Activer la détection du front montant
 - Activer la détection du front descendant
- Compteur rapide (HSC)
 - Activer ce compteur rapide
 - Générer une alarme pour un événement de type valeur de comptage égale à valeur de référence
 - Générer une alarme pour un événement de réinitialisation externe
 - Générer une alarme pour un événement d'inversion de sens

Ajout de nouveaux OB d'alarme de processus à votre programme

Par défaut, aucun OB n'est associé à un événement lorsque l'événement est activé pour la première fois. Cela est signalé par la mention "<non connecté>" dans la configuration d'appareil "Alarme de processus". Seuls des OB d'alarme de processus peuvent être associés à un événement d'alarme de processus. Tous les OB d'alarme de processus existants apparaissent dans la liste déroulante "Alarme de processus". Si aucun OB n'est listé, vous devez créer un OB de type "alarme de processus" de la manière suivante. Dans la branche "Blocs de programme" de l'arborescence de projet :

1. Double-cliquez sur "Ajouter nouveau bloc", sélectionnez "Bloc d'organisation (OB)", puis choisissez "Alarme de processus".
2. Vous pouvez optionnellement renommer l'OB, sélectionner le langage de programmation (CONT, LOG ou SCL) et sélectionner le numéro de bloc (commutez en manuel et choisissez un numéro de bloc différent de celui suggéré).
3. Editez l'OB pour y ajouter la réaction programmée que vous voulez exécuter lorsque l'événement se produit. Vous pouvez appeler des FC et des FB à partir de cet OB, jusqu'à la profondeur d'imbrication maximale. La profondeur d'imbrication maximale est de quatre pour les programmes de sécurité. Elle est de six pour les autres programmes.

Paramètre OB_NR

Tous les OB d'alarme de processus existants apparaissent dans la liste déroulante "Alarme de processus" de la configuration d'appareil et dans la liste déroulante du paramètre OB_NR de ATTACH / DETACH.

Paramètre EVENT

Lorsqu'un événement d'alarme de processus est activé, un nom d'événement par défaut unique est affecté à cet événement particulier. Vous pouvez modifier le nom d'événement en l'éditant dans la boîte d'édition "Nom d'événement", mais il doit rester unique. Ces noms d'événement deviennent des noms de variable dans la table de variables "Constantes" et apparaissent dans la liste déroulante du paramètre EVENT des boîtes d'instruction ATTACH et DETACH. La valeur de la variable est un numéro interne utilisé pour identifier l'événement.

Fonctionnement général

Chaque événement d'alarme de processus peut être associé à un OB d'alarme de processus qui sera mis en file d'attente pour exécution lorsque l'événement d'alarme de processus se produit. L'association OB-événement peut se faire au moment de la configuration ou à l'exécution.

Vous pouvez associer un OB à un événement activé ou l'en dissocier au moment de la configuration. Pour associer un OB à un événement lors de la configuration, vous devez utiliser la liste déroulante "Alarme de processus" (cliquez sur la flèche descendante à droite) et sélectionner un OB dans la liste des OB d'alarme de processus disponibles. Sélectionnez le nom de l'OB approprié dans la liste ou sélectionnez "<non connecté>" pour annuler l'association.

Vous pouvez également associer ou dissocier un événement d'alarme de processus activé pendant l'exécution. Utilisez les instructions de programme ATTACH ou DETACH pendant l'exécution – à plusieurs reprises si vous le désirez – pour associer un événement d'alarme activé à l'OB approprié ou pour l'en dissocier. En l'absence d'association à un OB (par sélection de "<non connecté>" dans la configuration d'appareil ou par exécution d'une instruction DETACH), l'événement d'alarme de processus activé n'est pas pris en compte.

Fonctionnement de DETACH

Utilisez l'instruction DETACH pour dissocier un événement particulier ou tous les événements d'un OB particulier. Si un événement est indiqué, seul cet événement est dissocié de l'OB de numéro OB_NR indiqué ; tout autre événement actuellement associé à ce numéro d'OB OB_NR lui reste associé. Si aucun événement EVENT n'est indiqué, tous les événements actuellement associés au numéro d'OB OB_NR en sont dissociés.

Codes d'erreur

Tableau 9- 99Codes d'erreur

RET_VAL (W#16#....)	ENO	Description
0000	1	Pas d'erreur
0001	1	Rien à dissocier (DETACH uniquement)
8090	0	OB inexistant
8091	0	Type d'OB erroné
8093	0	Événement inexistant

9.5.2 Alarmes cycliques

9.5.2.1 SET_CINT (Définir les paramètres de l'alarme cyclique)

Tableau 9- 100 SET_CINT (Définir les paramètres de l'alarme cyclique)

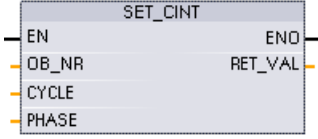
CONT/LOG	SCL	Description
	<pre>ret_val := SET_CINT(ob_nr:=_int_in_, cycle:=_udint_in_, phase:=_udint_in_);</pre>	<p>Permet de définir l'exécution cyclique interrompant le cycle du programme pour l'OB d'alarme indiqué.</p>

Tableau 9- 101 Types de données pour les paramètres

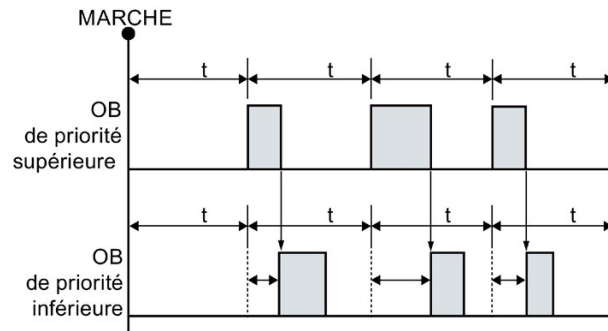
Paramètre et type		Type de données	Description
OB_NR	IN	OB_CYCLIC	Numéro d'OB (noms symboliques acceptés)
CYCLE	IN	UDInt	Intervalle de temps, en microsecondes
PHASE	IN	UDInt	Déphasage, en microsecondes
RET_VAL	OUT	Int	Code d'erreur d'exécution

Exemples : paramètre de temps

- Si le temps CYCLE est égal à 100 µs, l'OB d'alarme référencé par OB_NR interrompt le cycle du programme toutes les 100 µs. L'OB d'alarme s'exécute puis rend le contrôle au cycle du programme au niveau du point d'interruption.
- Si le temps CYCLE est égal à 0, l'événement d'alarme est désactivé et l'OB d'alarme n'est pas exécuté.
- Le déphasage PHASE est un temps de retard défini qui se produit avant que l'intervalle de temps CYCLE ne commence. Le déphasage peut vous servir à gérer l'instant d'exécution des OB de priorité inférieure.

Si des OB de priorité différente sont appelés dans le même intervalle de temps, l'OB de priorité inférieure n'est appelé qu'une fois le traitement de l'OB de priorité supérieure achevé. Le début de l'exécution de l'OB de priorité inférieure peut donc être décalé selon la durée de traitement des OB de priorité supérieure.

Appel d'OB sans déphasage



Si vous voulez démarrer l'exécution d'un OB de faible priorité selon un rythme fixe, définissez un temps de déphasage qui est supérieur à la durée de traitement des OB de priorité supérieure.

Appel d'OB avec déphasage

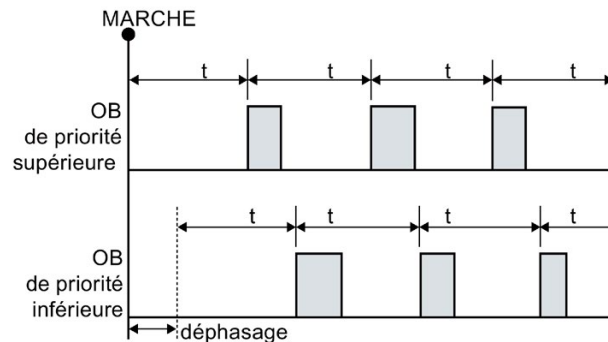


Tableau 9- 102 Codes d'erreur

RET_VAL (W#16#....)	Description
0000	Pas d'erreur
8090	L'OB n'existe pas ou a un type incorrect.
8091	Intervalle de temps invalide
8092	Déphasage invalide
80B2	Aucun événement n'est associé à l'OB.

9.5.2.2 QRY_CINT (Interroger les paramètres de l'alarme cyclique)

Tableau 9- 103 QRY_CINT (Interroger les paramètres de l'alarme cyclique)

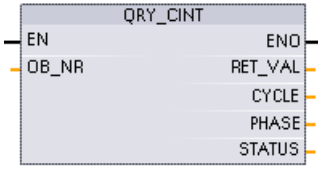
CONT/LOG	SCL	Description
	<pre>ret_val := QRY_CINT (ob_nr:=_int_in_, cycle=>_udint_out_, phase=>_udint_out_, status=>_word_out_);</pre>	Permet d'obtenir les valeurs des paramètres et l'état d'exécution d'un OB d'alarme cyclique. Les valeurs renvoyées sont celles qui étaient en vigueur au moment de l'exécution de QRY_CINT.

Tableau 9- 104 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_CYCLIC	Numéro d'OB (noms symboliques comme OB_MyOBName acceptés)
RET_VAL	OUT	Int	Code d'erreur d'exécution
CYCLE	OUT	UDInt	Intervalle de temps, en microsecondes
PHASE	OUT	UDInt	Déphasage, en microsecondes
STATUS	OUT	Word	Code d'état de l'alarme cyclique : <ul style="list-style-type: none"> • Bits 0 à 4 : voir le tableau pour STATUS ci-dessous • Autres bits : toujours 0

Tableau 9- 105 Paramètre STATUS

Bit	Valeur	Description
0	0	A l'état MARCHE de la CPU
	1	Pendant le démarrage
1	0	L'alarme est validée.
	1	L'alarme est inhibée via l'instruction DIS_IRT.
2	0	L'alarme n'est pas active ou a expiré.
	1	L'alarme est active.
4	0	L'OB identifié par OB_NR n'existe pas.
	1	L'OB identifié par OB_NR existe.
Autres bits		Toujours 0

Si une erreur se produit, RET_VAL contient le code d'erreur approprié et le paramètre STATUS est égal à 0.

Tableau 9- 106 Paramètre RET_VAL

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	L'OB n'existe pas ou a un type incorrect.
80B2	Aucun événement n'est associé à l'OB.

9.5.3 Alarmes horaires

ATTENTION

Accès possible d'un pirate à vos réseaux par la synchronisation NTP

Si un pirate parvient à accéder à vos réseaux par la synchronisation NTP (Network Time Protocol), il aura la possibilité de perturber la commande de votre processus en décalant l'heure système de la CPU. De telles perturbations dans la commande du processus peuvent entraîner la mort, des blessures graves ou des dommages matériels.

La fonction client NTP de la CPU S7-1200 est désactivée par défaut. Lorsqu'elle est activée, elle permet uniquement aux adresses IP configurées d'agir en tant que serveur NTP. Par défaut, la CPU désactive cette fonction et vous devez la configurer pour permettre les corrections à distance de l'heure système de la CPU.

La CPU S7-1200 prend en charge les alarmes horaires et les instructions d'horloge qui sont dépendantes de la précision de l'heure système de la CPU. Si vous configurez NTP et acceptez la synchronisation de l'heure par un serveur, vous devez vous assurer que le serveur est fiable. Sinon, vous risquez une violation de sécurité qui permet à un utilisateur inconnu de contrôler partiellement les process en déphasant l'heure système de la CPU.

Pour obtenir des informations et des recommandations sur la sécurité, veuillez consulter les directives opérationnelles "Operational Guidelines for Industrial Security" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) sur le site Service and Support de Siemens.

9.5.3.1 SET_TINTL (Définir une alarme horaire)

Tableau 9- 107 SET_TINTL (Définir une date et une alarme horaire avec type de données DTL)

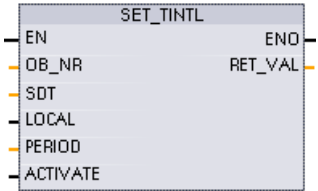
CONT/LOG	SCL	Description
	<pre>ret_val := SET_TINTL(OB_NR:=_int_in_, SDT:=_dtl_in_, LOCAL:=_bool_in_ PERIOD:=_word_in_ ACTIVATE:=_bool_in_);</pre>	Définir une date et une alarme horaire. Cet OB d'alarme programme peut être configuré pour une exécution ou pour une exécution récurrente avec une période de temps définie.

Tableau 9- 108 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_TOD (INT)	Numéro d'OB (noms symboliques acceptés)
SDT	IN	DTL	Date et heure de début : les secondes et millisecondes sont ignorées et peuvent être réglées sur 0.
LOCAL	IN	Bool	0 = Utiliser l'heure système 1 = Utiliser l'heure locale (si la CPU est configurée pour l'heure locale, sinon utiliser l'heure système)
PERIOD	IN	Word	La période à partir de la date et de l'heure de début de l'événement d'alarme récurrent. <ul style="list-style-type: none"> • W#16#0000 = Unique • W#16#0201 = Toutes les minutes • W#16#0401 = Toutes les heures • W#16#1001 = Tous les jours • W#16#1201 = Toutes les semaines • W#16#1401 = Tous les mois • W#16#1801 = Annuel • W#16#2001 = En fin de mois
ACTIVATE	IN	Bool	0 = ACT_TINT doit être exécuté pour activer l'événement d'alarme. 1 = L'événement d'alarme est activé.
RET_VAL	OUT	Int	Code d'erreur d'exécution

Votre programme peut utiliser SET_TINTL pour définir un événement d'alarme horaire qui exécutera l'OB d'alarme associé. La date et l'heure de déclenchement sont définies par le paramètre SDT et la fréquence des alarmes récurrentes (par exemple, tous les jours ou toutes les semaines) est définie par le paramètre PERIOD. Si vous définissez une fréquence mensuelle, vous devez régler la date de déclenchement à un jour compris entre 1 et 28. Les jours de 29 à 31 sont interdits parce qu'ils n'existent normalement pas en février. Si vous désirez un événement d'alarme se produise à la fin de chaque mois, utilisez En fin de mois comme paramètre PERIOD.

La valeur Jour de la semaine de données DTL dans le paramètre SDT est ignorée. Définissez une date et heure courante de la CPU à l'aide de la fonction "Régler l'heure" dans la vue "En ligne & Diagnostic" d'une CPU en ligne. Vous devez définir les valeurs mois, jour du mois et année. STEP 7 calcule la période d'alarme sur la base de l'horloge de la CPU.

Remarque

La première heure du jour n'existe pas lors du passage de l'heure d'été à l'heure d'hiver (heure d'été). Utilisez une heure de démarrage dans l'espace de la deuxième heure ou utilisez une alarme temporisée supplémentaire dans l'espace de la première heure.

Tableau 9- 109 Code d'erreur

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	Paramètre OB_NR invalide
8091	Heure de démarrage SDT invalide : (par exemple, une heure de démarrage dans l'heure sautée au début de l'heure d'été)
8092	Paramètre PERIOD invalide
80A1	L'heure de démarrage est déjà écoulée. (Ce code d'erreur ne se produit qu'avec PERIOD = W #16#0000.)

9.5.3.2 CAN_TINT (Annuler l'alarme horaire)

Tableau 9- 110 CAN_TINT (Annuler l'alarme horaire)

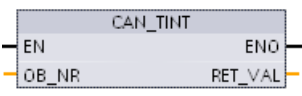
CONT/LOG	SCL	Description
	<pre>ret_val:=CAN_TINT(_int_in);</pre>	Annule l'événement déclencheur d'alarme horaire pour l'OB d'alarme indiqué.

Tableau 9- 111 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_TOD (INT)	Numéro d'OB (noms symboliques acceptés)
RET_VAL	OUT	Int	Code d'erreur d'exécution

Tableau 9- 112 Codes d'erreur

RET_VAL (W#16#....)	Description
0000	Pas d'erreur
8090	Paramètre OB_NR invalide
80A0	Aucune date/heure de début définie pour cet OB d'alarme

9.5.3.3 ACT_TINT (Activer alarme horaire)

Tableau 9- 113 ACT_TINT (Activer alarme horaire)

CONT/LOG	SCL	Description
	<pre>ret_val:=ACT_TINT(_int_in_);</pre>	Active l'événement déclencheur d'alarme horaire pour l'OB d'alarme indiqué.

Tableau 9- 114 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_TOD (INT)	Numéro d'OB (noms symboliques acceptés)
RET_VAL	OUT	Int	Code d'erreur d'exécution

Tableau 9- 115 Codes d'erreur

RET_VAL (W#16#....)	Description
0000	Pas d'erreur
8090	Paramètre OB_NR invalide
80A0	Date et heure de début non définie pour l'OB d'alarme horaire concerné.
80A1	L'heure activée est dans le passé. Cette erreur se produit uniquement lorsque l'OB d'alarme est paramétré pour s'exécuter une seule fois.

9.5.3.4 QRY_TINT (Interroger l'état de l'alarme horaire)

Tableau 9- 116 QRY_TINT (Interroger l'état de l'alarme horaire)

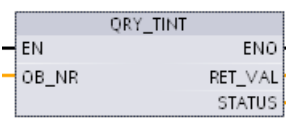
CONT/LOG	SCL	Description
	<pre>ret_val:=QRY_TINT(OB_NR:=_int_in_, STATUS=>_word_out_);</pre>	Interroge l'état de l'alarme horaire pour l'OB d'alarme indiqué.

Tableau 9- 117 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_TOD (INT)	Numéro de l'OB d'alarme à interroger (noms symboliques acceptés)
RET_VAL	OUT	Int	Code d'erreur d'exécution
STATUS	OUT	Word	Etat de l'OB d'alarme indiqué

Tableau 9- 118 Paramètre STATUS

Bit	Valeur	Description
0	0	A l'état MARCHÉ
	1	En démarrage
1	0	L'alarme est validée.
	1	L'alarme est inhibée.
2	0	L'alarme n'est pas active ou a expiré.
	1	L'alarme est active.
4	0	Le numéro OB_NR affecté n'existe pas.
	1	Il existe un OB ayant le numéro OB_NR affecté.
6	1	L'alarme horaire utilise l'heure locale.
	0	L'alarme horaire utilise l'heure système.
Autres		Toujours 0

Tableau 9- 119 Code d'erreur

RET_VAL (W#16#....)	Description
0000	Pas d'erreur
8090	Paramètre OB_NR invalide

9.5.4 Alarmes temporisées

Vous pouvez démarrer et annuler le traitement des alarmes temporisées à l'aide des instructions SRT_DINT et CAN_DINT et interroger l'état des alarmes à l'aide de l'instruction QRY_DINT. Chaque alarme temporisée est un événement unique qui se produit après le temps de retard indiqué. Si l'événement d'alarme temporisée est annulé avant que le temps de retard n'ait expiré, l'interruption du programme ne se produit pas.

Tableau 9- 120 Instructions SRT_DINT, CAN_DINT et QRY_DINT

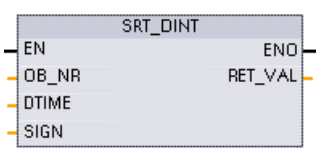
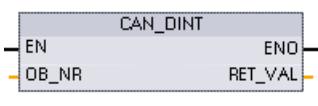
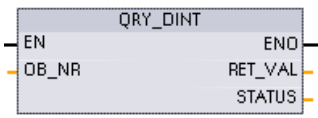
CONT/LOG	SCL	Description
	<pre>ret_val := SRT_DINT(ob_nr:=_int_in_, dtime:=_time_in_, sign:=_word_in_);</pre>	SRT_DINT démarre une alarme temporisée qui exécute un OB lorsque le temps de retard indiqué par le paramètre DTIME s'est écoulé.
	<pre>ret_val := CAN_DINT(ob_nr:=_int_in_);</pre>	CAN_DINT annule une alarme temporisée qui a déjà démarré. L'OB d'alarme temporisée n'est pas exécuté dans ce cas.
	<pre>ret_val := QRY_DINT(ob_nr:=_int_in_, status=>_word_out_);</pre>	QRY_DINT interroge l'état de l'alarme temporisée indiquée par le paramètre OB_NR.

Tableau 9- 121 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_DELAY	Bloc d'organisation (OB) à démarrer après un temps de retard. Faites votre choix parmi les OB d'alarme temporisée disponibles qui ont été créés via la fonction "Ajouter nouveau bloc" de l'arborescence de projet. Double-cliquez sur le champ de paramètre, puis cliquez sur l'icône d'aide pour voir les OB disponibles.
DTIME ¹	IN	Time	Valeur du temps de retard (1 à 60000 ms)
SIGN ¹	IN	Word	Non utilisé par le S7-1200 : Toute valeur est acceptée. Il faut indiquer une valeur pour éviter les erreurs.
RET_VAL	OUT	Int	Code d'erreur d'exécution
STATUS	OUT	Word	Instruction QRY_DINT : Etat de l'OB d'alarme temporisée indiqué. Voir le tableau ci-dessous.

¹ Uniquement pour SRT_DINT

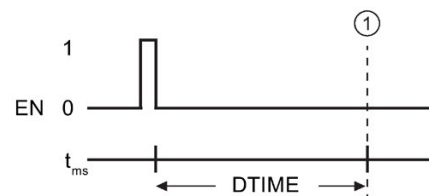
Fonctionnement

Quand EN=1, l'instruction SRT_DINT démarre la temporisation de temps de retard interne (DTIME). Lorsque le temps de retard est écoulé, la CPU génère une interruption du programme qui déclenche l'exécution de l'OB d'alarme temporisée associé. Vous pouvez annuler une alarme temporisée en cours de traitement avant que le temps de retard indiqué n'ait expiré en exécutant l'instruction CAN_DINT. Le nombre total d'événements d'alarme temporisée actifs ne doit pas dépasser quatre.

Remarque

SRT_DINT démarre la temporisation de temps de retard à chaque cycle lorsque EN égale 1. Vous devez donc donner la valeur 1 à EN de manière ponctuelle plutôt que de simplement le mettre à 1 pour démarrer votre temps de retard.

Chronogramme de l'instruction SRT_DINT :



- ① L'alarme temporisée s'exécute

Ajout d'OB d'alarme temporisée à votre projet

Vous ne pouvez affecter des OB d'alarme temporisée qu'aux instructions SRT_DINT et CAN_DINT. Aucun OB d'alarme temporisée n'existe dans un nouveau projet. Vous devez ajouter ces OB d'alarme temporisée à votre projet. Procédez comme suit pour créer un OB d'alarme temporisée :

1. Double-cliquez sur "Ajouter nouveau bloc" dans la branche "Blocs de programme" de l'arborescence de projet, sélectionnez "Bloc d'organisation (OB)", puis choisissez "Alarme temporisée".
2. Vous avez la possibilité de renommer l'OB, de sélectionner le langage de programmation ou de sélectionner le numéro de bloc. Commutez en numérotation manuelle si vous voulez utiliser un numéro de bloc différent de celui affecté automatiquement.
3. Editez le sous-programme de l'OB d'alarme temporisée et créez une réaction programmée que vous voulez exécuter lorsque l'événement d'alarme temporisée se produit. Vous pouvez appeler d'autres blocs de code FB et FC à partir de l'OB d'alarme temporisée. La profondeur d'imbrication maximale est de quatre pour les programmes de sécurité. Elle est de six pour les autres programmes.
4. Les noms des OB d'alarme temporisée nouvellement affectés seront disponibles lorsque vous éditez le paramètre OB_NR des instructions SRT_DINT et CAN_DINT.

Paramètre STATUS de QRY_DINT :

Tableau 9- 122 En présence d'une erreur (REL_VAL <> 0), STATUS = 0.

Bit	Valeur	Description
0	0	A l'état MARCHE
	1	Pendant le démarrage
1	0	L'alarme est validée.
	1	L'alarme est inhibée.
2	0	L'alarme n'est pas active ou a expiré.
	1	L'alarme est active.
4	0	Il n'existe pas d'OB de numéro OB_NR.
	1	Il existe un OB de numéro OB_NR.
Autres bits		Toujours 0

Codes d'erreur

Tableau 9- 123 Codes d'erreur pour SRT_DINT, CAN_DINT et QRY_DINT

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	Paramètre OB_NR incorrect
8091	Paramètre DTIME incorrect
80A0	L'alarme temporisée n'a pas démarré.

9.5.5 Instructions DIS_AIRT et EN_AIRT (Retarder/activer les alarmes de priorité supérieure et les erreurs asynchrones)

Utilisez les instructions DIS_AIRT et EN_AIRT pour inhiber et valider le traitement des alarmes.

Tableau 9- 124 Instructions DIS_AIRT et EN_AIRT

CONT/LOG	SCL	Description
	DIS_AIRT () ;	DIS_AIRT retarde le traitement de nouveaux événements d'alarme. Vous pouvez exécuter DIS_AIRT plus d'une fois dans un OB.
	EN_AIRT () ;	EN_AIRT valide le traitement d'événements d'alarme que vous aviez préalablement inhibé à l'aide de l'instruction DIS_AIRT. Chaque exécution de DIS_AIRT doit être annulée par une exécution de EN_AIRT. Les exécutions de EN_AIRT doivent se produire dans le même OB ou dans toute FC ou tout FB appelé par cet OB pour que les alarmes soient réactivées pour cet OB.

Tableau 9- 125 Types de données pour les paramètres

Paramètre et type		Type de données	Description
RET_VAL	OUT	Int	Nombre de retards = nombre d'exécutions DIS_AIRT dans la file d'attente

Les exécutions de DIS_AIRT sont comptabilisées par le système d'exploitation. Chacune d'elles reste en vigueur jusqu'à ce qu'elle soit expressément annulée par une instruction EN_AIRT ou jusqu'à ce que l'OB en cours ait été intégralement traité. Si, par exemple, vous avez désactivé les alarmes cinq fois via cinq exécutions de DIS_AIRT, vous devez annuler ces dernières avec cinq exécutions de EN_AIRT pour que le traitement des alarmes soit à nouveau activé.

Une fois les événements d'alarme réactivés, les alarmes qui se sont produites alors que DIS_AIRT était en vigueur sont traitées ou les alarmes sont traitées dès que l'OB en cours a été exécuté.

Le paramètre RET_VAL indique le nombre de fois où le traitement des alarmes a été inhibé, ce nombre correspondant au nombre d'exécutions de DIS_AIRT en file d'attente. Le traitement des alarmes n'est réactivé que lorsque le paramètre RET_VAL est égal à 0.

9.6 Messages

9.6.1 Gen_UsrMsg (Créer des alarmes de diagnostic utilisateur)

Tableau 9- 126 Instruction Gen_UsrMsg

CONT/LOG	SCL	Description
<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;">Gen_UsrMsg</p> <p style="margin: 0;">- EN ENO -</p> <p style="margin: 0;">- Mode Ret_Val -</p> <p style="margin: 0;">- TextID</p> <p style="margin: 0;">- TextListID</p> <p style="margin: 0;">- AssocValues</p> </div>	<pre>ret_val :=Gen_UsrMsg(Mode:=_uint_in_, TextID:=_uint_in_, TextListID:=_uint_in_, As- socValues:=_struct_inout_);</pre>	<p>L'instruction "Gen_UsrMsg" permet de générer un message de diagnostic personnalisé, qui peut être soit un message apparaissant soit un message disparaissant. Avec les messages de diagnostic personnalisés, vous pouvez inscrire une entrée dans le tampon de diagnostic et envoyer un message correspondant.</p> <p>La création de l'entrée dans le tampon de diagnostic s'effectue de manière synchrone. La transmission du message est asynchrone.</p> <p>Si une erreur se produit lors de l'exécution de l'instruction, elle est signalée dans le paramètre RET_VAL.</p>

Contenu du message

Vous définissez le contenu du message au moyen d'une liste de textes :

- Définissez avec le paramètre TextListID la liste de textes que vous voulez utiliser. À cet effet, ouvrez la boîte de dialogue "Listes de textes" dans le navigateur du projet. Affichez la colonne "ID" dans la boîte de dialogue "Listes de textes". Reprenez l'ID dans le paramètre TextListID.
- Utilisez le paramètre TextID pour sélectionner l'entrée de liste de textes que vous voulez inscrire dans le tampon de diagnostic. Sélectionnez à cet effet une entrée dans la boîte de dialogue "Entrées de listes de textes" en affectant un nombre des colonnes "Plage de/Plage à" au paramètre TextID. Vous devez utiliser le même nombre pour les colonnes "Plage de" et "Plage à" pour l'entrée de liste de textes.

Pour plus de détails sur les listes de textes, voir le système d'information de STEP 7.

Définition de valeurs associées

L'entrée de liste de textes définit de nouvelles valeurs associées à ajouter au message :

- Ajoutez l'information suivante à l'entrée de liste de textes pour définir des valeurs associées :

@<n° de la valeur associée><type d'élément><format>@

- Utilisez le type de données système AssocValues pour indiquer quelle valeur associée ajouter lors de la création du message.

Pour plus de détails sur la structure des valeurs associées, voir le système d'information de STEP 7.

Paramètres

Le tableau suivant présente les paramètres de l'instruction Gen_UsrMsg :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
Mode	Input	UInt	I, Q, M, D, L ou constante	Paramètre pour la sélection du type de message : <ul style="list-style-type: none"> • 1 : message apparaissant • 2 : message disparaissant
TextID	Input	UInt	I, Q, M, D, L ou constante	ID de l'entrée de liste de textes qui doit être utilisée comme texte du message
TextListID	Input	UInt	I, Q, M, D, L ou constante	ID de la liste de textes qui contient l'entrée de liste de textes
Ret_Val	Return	Int	I, Q, M, D, L	Code d'erreur de l'instruction
AssocValues	InOut	VARIANT	D, L	Pointeur désignant le type de données système AssocValues permettant de définir les valeurs associées

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Paramètre AssocValues

Utilisez le type de données système AssocValues pour définir quelles valeurs associées seront envoyées. Un maximum de 8 valeurs associées est possible. Entrez le type de données "AssocValues" en tant que bloc de données pour créer la structure.

Vous sélectionnez des valeurs associées en entrant leur numéro pour les paramètres Value[x]. Tenez compte ici des points suivants :

- L'instruction Gen_UsrMsg traite les valeurs pour TextID et TextListID en tant que valeurs associées à envoyer. En conséquence, les numéros "1" et "2" sont déjà attribués pour l'adressage de ces valeurs associées. Vous ne devez donc pas utiliser pas les numéros "1" et "2" pour adresser des valeurs associées.
- Adressez la valeur associée au paramètre Value [1] en tant que numéro "3", la valeur associée au paramètre Value [2] en tant que numéro "4", etc.

Octet	Paramètre	Type de données	Valeur initiale	Description	Numéro de la valeur associée
0..1	Value[1]	UINT	0	Première valeur associée du message	3
2..3	Value[2]	UINT	0	Deuxième valeur associée du message	4
4..5	Value[3]	UINT	0	...	5
6..7	Value[4]	UINT	0	...	6
8..9	Value[5]	UINT	0	...	7
10..11	Value[6]	UINT	0	...	8
12..13	Value[7]	UINT	0	...	9
14..15	Value[8]	UINT	0	Huitième valeur associée du message	10

Paramètre RET_VAL

Le tableau suivant définit les valeurs de sortie pour le paramètre RET_VAL . Voir aussi : Codes d'erreur communs pour les instructions avancées (Page 571).

Code d'erreur* (W#16#...)	Explication
0000	Pas d'erreur
8080	La valeur dans le paramètre MODE n'est pas prise en charge.
80C1	Manque de ressources dû à un excès d'appels en parallèle
8528	Le paramètre 5 (AssocValues) n'est pas aligné sur les octets.
853A	Le paramètre 5 (AssocValues) référence un point invalide.
* Vous pouvez afficher le code d'erreur sous forme de valeur entière ou hexadécimale dans l'éditeur de programmes.	

9.7 Diagnostic (PROFINET ou PROFIBUS)

9.7.1 Opérations de diagnostic

Vous pouvez utiliser les instructions de diagnostic suivantes avec PROFINET ou PROFIBUS :

- Instruction RD_SINFO (Page 450) : lit l'information de déclenchement de l'OB actuel.
- Instruction LED (Page 461) : lit l'état des LED pour un périphérique IO décentralisé.
- Instruction Get_IM_Data (Page 462) : lit les données d'identification et de maintenance (I&M) pour le module ou sous-module indiqué.
- Instruction Get_Name (Page 464) : lit le nom d'un périphérique PROFINET IO, d'un esclave PROFIBUS ou d'un esclave AS-i.
- Instruction GetStationInfo (Page 471) : lit l'adresse IP ou MAC d'un périphérique PROFINET IO dans le réseau IO local ou d'un périphérique PROFINET IO situé dans un réseau IO de niveau inférieur (connecté par des modules CP/CM).
- Instruction DeviceStates (Page 479) : lit l'état de fonctionnement d'un périphérique IO décentralisé d'un sous-système IO.
- Instruction ModuleStates (Page 485) : lit l'état de fonctionnement des modules dans un périphérique IO décentralisé.
- Instruction GET_DIAG (Page 491) : lit les informations de diagnostic du périphérique indiqué.


Remarque

L'instruction GetStationInfo n'est utilisable qu'avec des périphériques PROFINET IO. Vous ne pouvez pas l'utiliser avec des esclaves PROFIBUS DP.

9.7.2 RD_SINFO (Lire l'information de déclenchement de l'OB actuel)

Description

Tableau 9- 127 Instruction RD_SINFO

CONT/LOG	SCL	Description
	<pre>ret_val := RD_SINFO(TOP_SI=>_variant_out_, START_UP_SI=>_variant_out_) ;</pre>	<p>L'instruction RD_SINFO permet de lire l'information de déclenchement des blocs d'organisation suivants :</p> <ul style="list-style-type: none"> • dernier OB appelé dont le traitement n'est pas encore entièrement achevé • dernier OB de démarrage déclenché par la CPU <p>Il n'y a d'horodatage dans aucun des cas. Si l'appel a lieu dans l'OB 100, l'OB 101 ou l'OB 102, deux messages d'information de déclenchement identiques sont renvoyés.</p>

Paramètres

Le tableau suivant présente les paramètres de l'instruction RD_SINFO :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
RET_VAL	Return	INT	I, Q, M, D, L	Information d'erreur
TOP_SI	Output	VARIANT	D, L	Information de déclenchement de l'OB actuel
START_UP_SI	Output	VARIANT	D, L	Information de déclenchement du dernier OB de démarrage déclenché

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

SDT du paramètre TOP_SI

Le tableau suivant présente les SDT possibles pour le paramètre TOP_SI :

Blocs d'organisation (OB)	Types de données système (SDT)	Numéros des types de données système
Tous	SI_classic*	592*
	SI_none	593
ProgramCycleOB	SI_ProgramCycle	594
TimeOfDayOB	SI_TimeOfDay	595
TimeDelayOB	SI_Delay	596
CyclicOB	SI_Cyclic	597
ProcessEventOB	SI_HWInterrupt	598
ProfileEventOB StatusEventOB UpdateEventOB	SI_Submodule	601
SynchronousCycleOB	SI_SynchCycle	602
IOredundancyErrorOB	SI_IORedundancyError	604
CPUredundancyErrorOB	SI_CPURedundancyError	605
TimeErrorOB	SI_TimeError	606
DiagnosticErrorOB	SI_DiagnosticInterrupt	607
PullPlugEventOB	SI_PlugPullModule	608
PeripheralAccessErrorOB	SI_AccessError	609
RackStationFailureOB	SI_StationFailure	610
ServoOB	SI_Servo	611
IpoOB	SI_Ipo	612
StartupOB	SI_Startup	613
ProgrammingErrorOB IOaccessErrorOB	SI_ProgIOAccessError	614

* Le SDT SI_classic n'est pas applicable dans le S7-1200. La CPU S7-1200 renvoie la valeur #16#8081 dans RET_VAL si le paramètre TOP_SI est de type SI_classic.

SDT du paramètre START_UP_SI

Le tableau suivant présente les SDT possibles pour le paramètre START_UP_SI :

Types de données système (SDT)	Numéros des types de données système
SI_classic*	592
SI_none	593
SI_Startup	613

* Le SDT SI_classic n'est pas applicable dans le S7-1200. La CPU S7-1200 renvoie la valeur #16#8083 dans RET_VAL si le paramètre START_UP_SI est de type SI_classic.

Structures

Les tableaux suivants définissent les éléments des différentes structures.

Tableau 9- 128 Structure SI_classic

Élément de structure	Type de données	Description
EV_CLASS	BYTE	<ul style="list-style-type: none"> Bits 0 à 3 : ID d'événement Bits 4 à 7 : Classe d'événement
EV_NUM	BYTE	Numéro d'événement
PRIORITY	BYTE	Numéro de classe de priorité (signification de B#16#FE : OB non disponible ou inhibé ou ne pouvant pas être démarré dans l'état de fonctionnement actuel)
NUM	BYTE	Numéro d'OB
TYP2_3	BYTE	ID de données 2_3 : Identifie l'information entrée dans ZI2_3
TYP1	BYTE	ID de données 1 : Identifie l'information entrée dans ZI1
ZI1	WORD	Information supplémentaire 1
ZI2_3	DWORD	Information supplémentaire 2_3

Tableau 9- 129 Structure SI_none

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)

Tableau 9- 130 Structure SI_ProgramCycle

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 1	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
Initial_Call	BOOL	Pour OB_Class = 1, 30, 52, 61, 65
Remanence	BOOL	Pour OB_Class = 1

Tableau 9- 131 Structure SI_TimeOfDay

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 10	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
CaughtUp	BOOL	Pour OB_Class = 10
SecondTime	BOOL	Pour OB_Class = 10

Tableau 9- 132 Structure SI_Delay

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 20	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
Sign	WORD	Pour OB_Class = 20

Tableau 9- 133 Structure SI_Cyclic

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 30	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
Initial_Call	BOOL	Pour OB_Class = 1, 30, 52, 61, 65
Event_Count	INT	Pour OB_Class = 30, 51, 52, 61, 65, 91, 92

Tableau 9- 134 Structure SI_HWInterrupt

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 40	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
LADDR	HW_IO	Pour OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
USI	WORD	Pour OB_Class = 40
IChannel	USINT	Pour OB_Class = 40
EventType	BYTE	Pour OB_Class = 40

Tableau 9- 135 Structure SI_Submodule

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
LADDR	HW_IO	Pour OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Slot	UINT	Pour OB_Class = 55, 56, 57
Specifier	WORD	Pour OB_Class = 55, 56, 57

Tableau 9- 136 Structure SI_SynchCycle

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 61	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
Initial_Call	BOOL	Pour OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Pour OB_Class = 61, 91, 92
PIP_Output	BOOL	Pour OB_Class = 61, 91, 92
IO_System	USINT	Pour OB_Class = 61, 91, 92
Event_Count	INT	Pour OB_Class = 30, 51, 52, 61, 65, 91, 92
SyncCycleTime	LTIME	Temps de cycle calculé

Tableau 9- 137 Structure SI_IORedundancyError

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 70	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
LADDR	HW_ANY	Pour OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Pour OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Pour OB_Class = 70, 80, 83, 85, 86

Tableau 9- 138 Structure SI_CPURedundancyError

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 72	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
Switch_Over	BOOL	Pour OB_Class = 72

Tableau 9- 139 Structure SI_TimeError

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 80	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
Fault_ID	BYTE	Pour OB_Class = 70, 80, 83, 85, 86
Csg_OBnr	OB_ANY	Pour OB_Class = 80
Csg_Prio	UINT	Pour OB_Class = 80

9.7 Diagnostic (PROFINET ou PROFIBUS)

Tableau 9- 140 Structure SI_DiagnosticInterrupt

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 82	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
IO_State	WORD	Pour OB_Class = 82
LADDR	HW_ANY	Pour OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Channel	UINT	Pour OB_Class = 82
MultiError	BOOL	Pour OB_Class = 82

Tableau 9- 141 Structure SI_PlugPullModule

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 83	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
LADDR	HW_IO	Pour OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Pour OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Pour OB_Class = 70, 80, 83, 85, 86

Tableau 9- 142 Structure SI_AccessError

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 85	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
LADDR	HW_IO	Pour OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Pour OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Pour OB_Class = 70, 80, 83, 85, 86
IO_Addr	UINT	Pour OB_Class = 85
IO_LEN	UINT	Pour OB_Class = 85

Tableau 9- 143 Structure SI_StationFailure

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 86	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
LADDR	HW_IO	Pour OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Pour OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Pour OB_Class = 70, 80, 83, 85, 86

Tableau 9- 144 Structure SI_Servo

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 91	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
Initial_Call	BOOL	Pour OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Pour OB_Class = 61, 91, 92
PIP_Output	BOOL	Pour OB_Class = 61, 91, 92
IO_System	USINT	Pour OB_Class = 61, 91, 92
Event_Count	INT	Pour OB_Class = 30, 51, 52, 61, 65, 91, 92
Synchronous	BOOL	

Tableau 9- 145 Structure SI_Ipo

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> • 16#FF = Pas d'information • 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 92	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
Initial_Call	BOOL	Pour OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Pour OB_Class = 61, 91, 92
PIP_Output	BOOL	Pour OB_Class = 61, 91, 92
IO_System	USINT	Pour OB_Class = 61, 91, 92
Event_Count	INT	Pour OB_Class = 30, 51, 52, 61, 65, 91, 92
Reduction	UINT	Pour OB_Class = 92

Tableau 9- 146 Structure SI_Startup

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> • 16#FF = Pas d'information • 16#FE = Information de déclenchement optimisée
OB_Class	USINT := 100	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
LostRetentive	BOOL	Pour OB_Class = 100
LostRTC	BOOL	Pour OB_Class = 100

Tableau 9- 147 Structure SI_ProgIOAccessError

Élément de structure	Type de données	Description
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Pas d'information 16#FE = Information de déclenchement optimisée
OB_Class	USINT	Classe d'OB pour "Pas d'information" ou "Information de déclenchement optimisée"
OB_Nr	UINT	Numéro d'OB (1 ... 32767)
BlockNr	UINT	Pour OB_Class = 121, 122
Reaction	USINT	Pour OB_Class = 121, 122
Fault_ID	BYTE	Pour OB_Class = 121, 122
BlockType	USINT	Pour OB_Class = 121, 122
Area	USINT	Pour OB_Class = 121, 122
DBNr	DB_ANY	Pour OB_Class = 121, 122
Csg_OBNr	OB_ANY	Pour OB_Class = 121, 122
Csg_Prio	USINT	Pour OB_Class = 121, 122
Width	USINT	Pour OB_Class = 121, 122

Remarque

Les éléments de structure indiqués pour la structure "SI_classic" sont équivalents du point de vue du contenu aux variables temporaires d'un OB lorsque celui-ci a été créé avec la propriété de bloc "Standard".

Notez toutefois que les variables temporaires des différents OB peuvent porter des noms différents et avoir un type de données différent. Notez également que l'interface d'appel de chaque OB comprend des informations supplémentaires concernant la date et l'heure auxquelles l'OB a été appelé.

Les bits 4 à 7 de l'élément de structure EV_CLASS contiennent la classe d'événement. Les valeurs suivantes sont possibles :

- 1 : Événements de déclenchement d'OB standard
- 2 : Événements de déclenchement d'OB d'erreur synchrone
- 3 : Événements de déclenchement d'OB d'erreur asynchrone

L'élément de structure PRIORITY fournit la classe de priorité correspondant à l'OB actuel.

Outre ces deux éléments, NUM est également pertinent : il contient le numéro de l'OB actuel ou du dernier OB de démarrage déclenché.

Paramètre RET_VAL

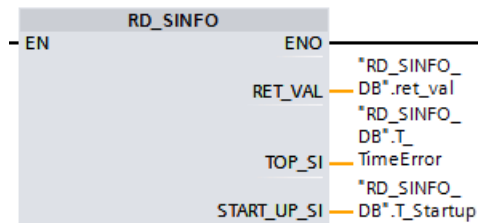
Le tableau suivant donne la signification des valeurs du paramètre RET_VAL :

Code d'erreur* (W#16#...)	Explication
8081	L'information de déclenchement de l'OB actuel ne correspond pas au type de données système indiqué.
8083	L'information de déclenchement du dernier OB de démarrage déclenché ne correspond pas au type de données système indiqué.

* Vous pouvez afficher le code d'erreur sous forme de valeur entière ou hexadécimale dans l'éditeur de programmes.

Exemple

Un OB d'erreur de temps (OB 80) est l'OB qui a été appelé en dernier et dont le traitement n'est pas encore achevé. L'OB 100 est l'OB de démarrage qui a été déclenché en dernier. L'appel de l'instruction pour lire les informations de déclenchement se présente comme suit, RD_SINFO_DB étant le bloc de données qui contient les variables des SDT pour les types d'OB :



Le tableau suivant indique l'affectation entre les éléments de structure du paramètre TOP_SI de l'instruction "RD_SINFO" et les variables locales correspondantes de l'OB 80.

Élément de structure TOP_SI	Type de données	Variable locale OB 80 correspondante	Type de données
EV_CLASS	BYTE	OB80_EV_CLASS	BYTE
EV_NUM	BYTE	OB80_FLT_ID	BYTE
PRIORITY	BYTE	OB80_PRIORITY	BYTE
NUM	BYTE	OB80_OB_NUMBR	BYTE
TYP2_3	BYTE	OB80_RESERVED_1	BYTE
TYP1	BYTE	OB80_RESERVED_2	BYTE
Z11	WORD	OB80_ERROR_INFO	WORD
Z12_3	DWORD	OB80_ERR_EV_CLASS	BYTE
		OB80_ERR_EV_NUM	BYTE
		OB80_OB_PRIORITY	BYTE
		OB80_OB_NUM	BYTE

Le tableau suivant indique l'affectation entre les éléments de structure du paramètre START_UP_SI de l'instruction "RD_SINFO" et les variables locales correspondantes de l'OB 100.

Élément de structure START_UP_SI	Type de données	Variables locales de l'OB 100	Type de données
EV_CLASS	BYTE	OB100_EV_CLASS	BYTE
EV_NUM	BYTE	OB100_STRTUP	BYTE
PRIORITY	BYTE	OB100_PRIORITY	BYTE
NUM	BYTE	OB100_OB_NUMBR	BYTE
TYP2_3	BYTE	OB100_RESERVED_1	BYTE
TYP1	BYTE	OB100_RESERVED_2	BYTE
ZI1	WORD	OB100_STOP	WORD
ZI2_3	DWORD	OB100_STRT_INFO	DWORD

9.7.3 LED (Lire l'état de la LED)

Tableau 9- 148 Instruction LED

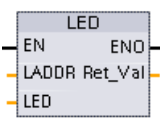
CONT/LOG	SCL	Description
	<pre>ret_val := LED(laddr:=_word_in_, LED:=_uint_in_);</pre>	Utilisez l'instruction LED pour lire l'état des DEL sur la CPU (Page 1379). L'état de la DEL indiquée est renvoyé dans la sortie RET_VAL.

Tableau 9- 149 Types de données pour les paramètres

Paramètre et type		Type de données	Description		
LADDR	IN	HW_IO	Identification de la CPU ¹		
LED	IN	UInt	Numéro identificateur de la DEL		
			1	RUN/STOP	Couleur 1 = vert, couleur 2 = jaune
			2	Défaut	Couleur 1 = rouge
			3	Maintenance	Couleur 1 = jaune
RET_VAL	OUT	Int	Etat de la DEL		

¹ Pour l'identification de la CPU connectée, sélectionnez Local-Common dans la liste déroulante du paramètre.

Tableau 9- 150 Etat de RET_VAL

RET_VAL (W#16#...)	Description	
Etat de DEL 0 à 9	0	La DEL n'existe pas.
	1	La DEL est éteinte.
	2	La DEL est allumée en couleur 1 (feu fixe).
	3	La DEL est allumée en couleur 2 (feu fixe).
	4	La DEL clignote en couleur 1 à 2 Hz.
	5	La DEL clignote en couleur 2 à 2 Hz.
	6	La DEL clignote alternativement en couleur 1 et 2 à 2 Hz.
	9	L'état de la DEL n'est pas disponible.
8091	L'appareil identifié par LADDR n'existe pas.	
8092	L'appareil identifié par LADDR ne comporte pas de DEL.	
8093	L'identificateur de DEL n'est pas défini.	
80Bx	La CPU identifiée par LADDR ne prend pas en charge l'instruction LED.	

9.7.4 Get_IM_Data (Lire les données d'identification et de maintenance)

Vous utilisez l'instruction Get_IM_Data pour vérifier les données d'identification et de maintenance (I&M) pour le module ou le sous-module spécifié.

Tableau 9- 151 Instruction Get_IM_Data

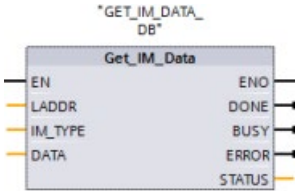
CONT/LOG	SCL	Description
 <p>The diagram shows a rectangular block for the instruction 'Get_IM_Data'. On the left side, there are four input terminals: 'EN' (top), 'LADDR' (middle), 'IM_TYPE' (bottom), and 'DATA' (bottom). On the right side, there are five output terminals: 'ENO' (top), 'DONE' (middle), 'BUSY' (middle), 'ERROR' (bottom), and 'STATUS' (bottom). The block is titled 'Get_IM_Data' and is part of a larger context labeled '*GET_IM_DATA_DB'.</p>	<pre>"GET_IM_DATA_DB" (LADDR:=16#0 , IM_TYPE:=0, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA:= variant inout);</pre>	<p>Utilisez l'instruction Get_IM_Data pour vérifier les données d'identification et de maintenance (I&M) pour le module ou le sous-module spécifié.</p>

Tableau 9- 152 Types de données pour les paramètres

Paramètre et type		Type de données	Description
LADDR	Input	HW_IO	Identificateur du module
IM_TYPE	Input	UInt	Numéro des données d'identification et de maintenance (I&M) : <ul style="list-style-type: none"> • 0 : I&M0 (n° de référence, numéro de série, version, etc.) • 1 : I&M1 (Identificateurs) • 2 : I&M2 (Date d'installation) • 3 : I&M3 (Description) • 4 : I&M4 (Signature)
RET_VAL	Output	Int	Etat (code d'erreur)
DATA	InOut	Variante	Données I&M (STRING ou un tableau de BYTE) ; utilisation recommandée du SDT "IM0_Data" pour IM_TYPE = 0.

Les données d'identification et de maintenance peuvent vous aider à vérifier la configuration du système, à détecter les modifications matérielles ou à visualiser les données de maintenance. Les données d'identification du module (données I) sont en lecture seule. Les données de maintenance du module (données M) dépendent des informations système, telles que la date d'installation. Les données M sont créées lors de la planification de la maintenance et sont inscrites dans le module :

- Si le type de données utilisé dans le paramètre DATA est une chaîne de caractères, la longueur actuelle de la chaîne est définie selon la longueur des données I&M.
- Si le type de données utilisé dans le paramètre DATA est un tableau de Byte ou Char, les données I&M sont copiées sous forme d'une séquence d'octets.
- Si le type de données utilisé dans le paramètre DATA est une structure, les données I&M sont copiées sous forme d'une séquence d'octets.
- Si le tableau d'octets/de char donné dans DATA est plus long que les données I&M demandées, la valeur d'octet 16#00 est ajoutée à la fin.
- Les autres types de données ne sont pas pris en charge et l'erreur 8093 est renvoyée.

Tableau 9- 153 Codes d'erreur

RET_VAL (W#16#...)	Description
0	Pas d'erreur
8091	LADDR n'existe pas.
8092	LADDR n'accède pas à un objet matériel qui prend en charge les données I&M.
8093	Le type de données dans le paramètre DATA n'est pas pris en charge.
80B1	L'instruction DATA n'est pas prise en charge par la CPU pour cette adresse LADDR.
80B2	IM_TYPE n'est pas prise en charge par la CPU.
8452	La longueur de l'ensemble des données I&M est incompatible avec la variable donnée dans le paramètre DATA. Un résultat partiel pouvant aller jusqu'à la longueur en octets de la variable est renvoyé.

9.7.5 Get_Name (Lire le nom d'un périphérique PROFINET IO)

L'instruction Get_Name permet de lire le nom d'un périphérique PROFINET IO, d'un esclave PROFIBUS ou d'un esclave AS-i. Le nom est affiché dans la vue du réseau et dans les propriétés du périphérique IO.

Tableau 9- 154 Instruction Get_Name

CONT/LOG	SCL	Description
	<pre>"Get_Name_DB" (LADDR:=_uint_in_, STATION_NR:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, LEN=>_dint_out_, STATUS=>_word_out_, DATA:=_variant_inout_);</pre>	<p>Utilisez l'instruction Get_Name pour lire le nom d'un périphérique PROFINET IO ou d'un esclave PROFIBUS.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "Get_Name_DB" est le nom du DB d'instance.

Vous sélectionnez le périphérique IO au moyen de l'identification matérielle du réseau IO décentralisé (dans le paramètre LADDR) et du numéro du périphérique PROFINET IO ou de l'adresse PROFIBUS de l'esclave PROFIBUS (dans le paramètre STATION_NR).

Une fois l'instruction exécutée, le programme écrit le nom du périphérique IO dans la zone désignée par le paramètre DATA.

Le nom lu dépend du type de périphérique IO :

- Esclave DP ou périphérique IO : nom du module de tête
- Esclave I ou périphérique I : nom du module d'interface
- Pupitre IHM : nom de l'interface
- Station PC : nom du coupleur
- Appareils GSD : affiche le nom du point d'accès d'appareil (DAP ; nom du module d'interface ou de tête)

L'instruction écrit la longueur du nom dans le paramètre LEN. Si le nom est plus long que la zone indiquée dans le paramètre DATA, le programme en écrit uniquement la partie correspondant à la longueur maximale de la zone désignée.

La longueur maximale pour un nom est de 128 caractères.

Remarque

Lecture du nom de la CPU (version 1.1)

Si vous affectez la valeur zéro aux paramètres LADDR et STATION_NR,, l'instruction écrit le nom de la CPU.

Paramètres

Le tableau suivant présente les paramètres de l'instruction Get_Name :

Paramètre	Déclaration	Type de données	Description
LADDR	IN	HW_IOSYSTEM	Identification matérielle (HW_IOSYSTEM) du réseau IO décentralisé. Ce nombre est pris dans les constantes système ou les propriétés du réseau IO.
STATION_NR	IN	UInt	<ul style="list-style-type: none"> Périphérique PROFINET IO : Le numéro d'appareil est appliqué dans la vue du réseau à partir des propriétés du périphérique IO sous "Adresses Ethernet". Esclave PROFIBUS : l'adresse PROFIBUS est appliquée dans la vue du réseau à partir des propriétés de l'esclave PROFIBUS sous "Adresse PROFIBUS".
DATA	IN_OUT	Variant	Pointeur désignant la zone dans laquelle le nom est écrit
DONE	OUT	Bool	L'instruction s'exécute avec succès. Le nom du module est transféré dans la zone désignée par le paramètre DATA.
BUSY	OUT	Bool	Paramètre d'état : <ul style="list-style-type: none"> 0 : L'exécution de l'instruction est achevée. 1 : L'exécution de l'instruction n'est pas encore achevée.
ERROR	OUT	Bool	Paramètre d'état : <ul style="list-style-type: none"> 0 : Pas d'erreur 1 : Une erreur s'est produite pendant l'exécution de l'instruction. Le paramètre STATUS contient des informations détaillées.
LEN	OUT	DInt	Longueur du nom du périphérique IO (nombre de caractères)
STATUS	OUT	Word	Paramètre d'état : ce paramètre est mis à 1 pour la durée d'un appel uniquement. Pour afficher l'état, il faut donc copier STATUS dans une zone de données libre.

Paramètre STATUS

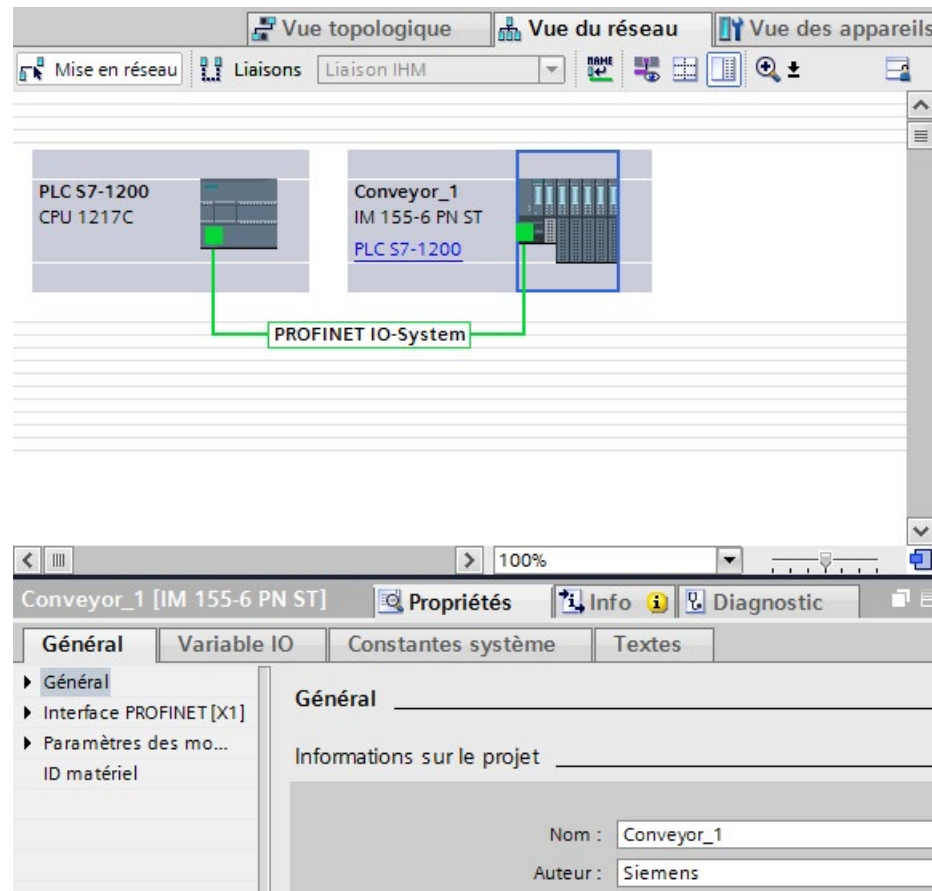
Code d'erreur* (W#16#...)	Explication
0	Pas d'erreur
7000	Il n'y a pas de tâche en cours.
7001	Premier appel de l'instruction asynchrone Get_Name. L'exécution de l'instruction n'est pas encore achevée (BUSY = 1, DONE = 0).
7002	Autre appel de l'instruction asynchrone Get_Name. L'exécution de l'instruction n'est pas encore achevée (BUSY = 1, DONE = 0).
8090	L'identification matérielle indiquée dans le paramètre LADDR n'existe pas dans le projet.
8092	La valeur dans le paramètre LADDR ne désigne pas un réseau PROFINET IO.
8093	L'instruction ne prend pas en charge le type de données dans le paramètre DATA.
8095	Le numéro d'appareil (paramètre STATION_NR) n'existe pas dans le réseau PROFINET IO sélectionné ou ne désigne pas un périphérique IO.
80B1	La CPU utilisée ne prend pas l'instruction en charge.
80C3	Erreur temporaire de ressource : la CPU traite actuellement le nombre maximum possible d'appels de bloc simultanés. Get_Name ne pourra être exécuté que lorsque au moins un des appels de bloc sera achevé.
8852	La zone indiquée dans le paramètre DATA est trop courte pour le nom entier du périphérique IO. Le nom peut être écrit jusqu'à la longueur maximale possible. Pour lire le nom en entier, utilisez une zone de données plus longue dans le paramètre DATA. La zone doit avoir au moins autant de caractères qu'il y en a dans le paramètre LEN.
* Les codes d'erreur dans l'éditeur de programmes peuvent être affichés sous forme de valeurs entières ou hexadécimales.	

Exemple

L'exemple suivant montre comment lire le nom de station d'un périphérique PROFINET IO ET 200SP :

1. Configurer l'ET 200SP :

- Créez l'ET 200SP avec le nom de station "Conveyor_1" dans la vue du réseau et affectez-le au même réseau PROFINET IO que la CPU.
- Définissez la CPU comme contrôleur IO pour l'ET 200SP.
- Utilisez le numéro de périphérique par défaut "1" situé dans les propriétés sous "Adresses Ethernet".



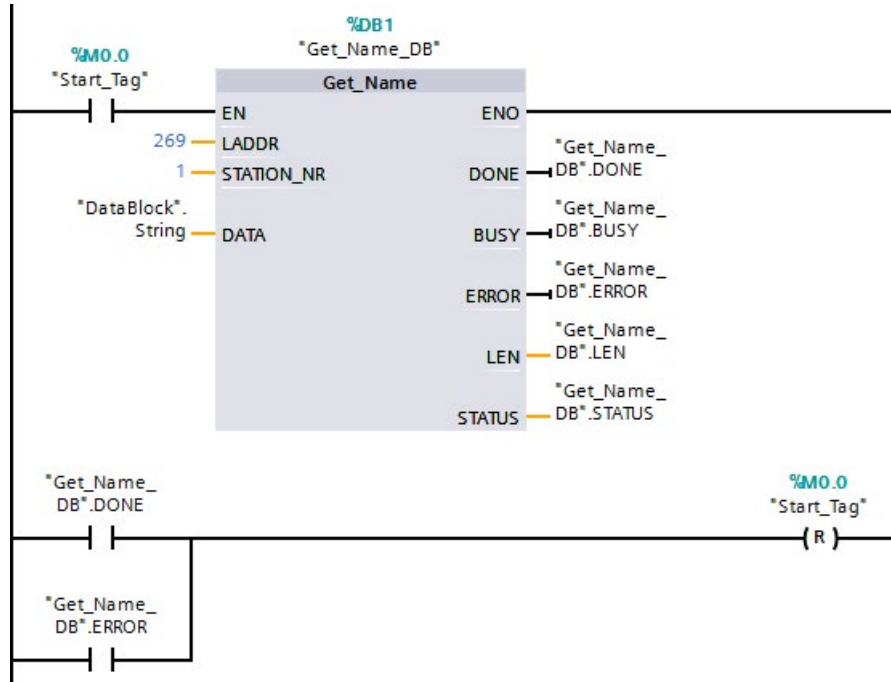
2. Affecter les paramètres pour l'instruction Get_Name :

- Saisissez l'ID matérielle du réseau IO dans le paramètre LADDR. Dans cet exemple, l'ID matérielle est "269". Vous trouverez l'ID matérielle à l'emplacement suivant : Variables API > Afficher toutes les variables > onglet Constantes système > Réseau PROFINET IO local
- Entrez le numéro d'appareil de l'ET 200SP dans le paramètre STATION_NR. Dans cet exemple, le numéro d'appareil est "1".
- Connectez une variable d'un bloc de données avec le type de données STRING dans le paramètre DATA.

Remarque

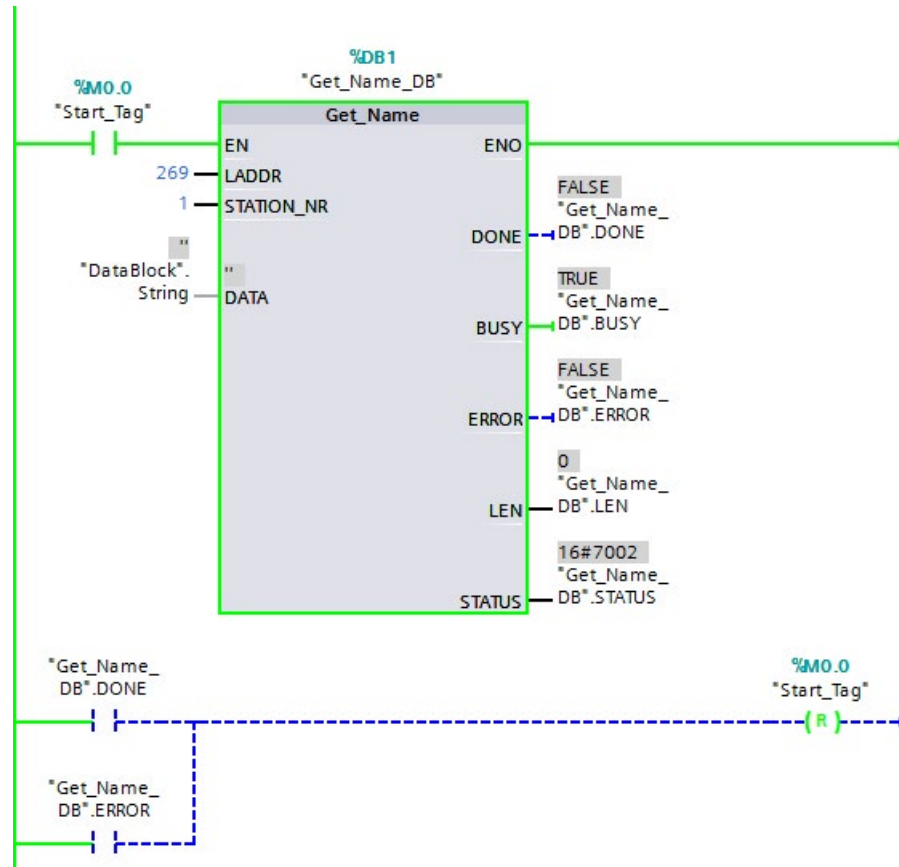
Lorsque vous configurez la variable pour le paramètre DATA à l'aide de la liste déroulante, sélectionnez le DB ("Datablock" dans l'exemple) et la variable ("String[]" dans l'exemple). Pour lire le type de données String entier, vous devez effacer les crochets afin d'obtenir le résultat final suivant : "Datablock".String

- Définissez les variables API (zone de mémoire, mémentos) pour les paramètres de sortie de l'instruction.

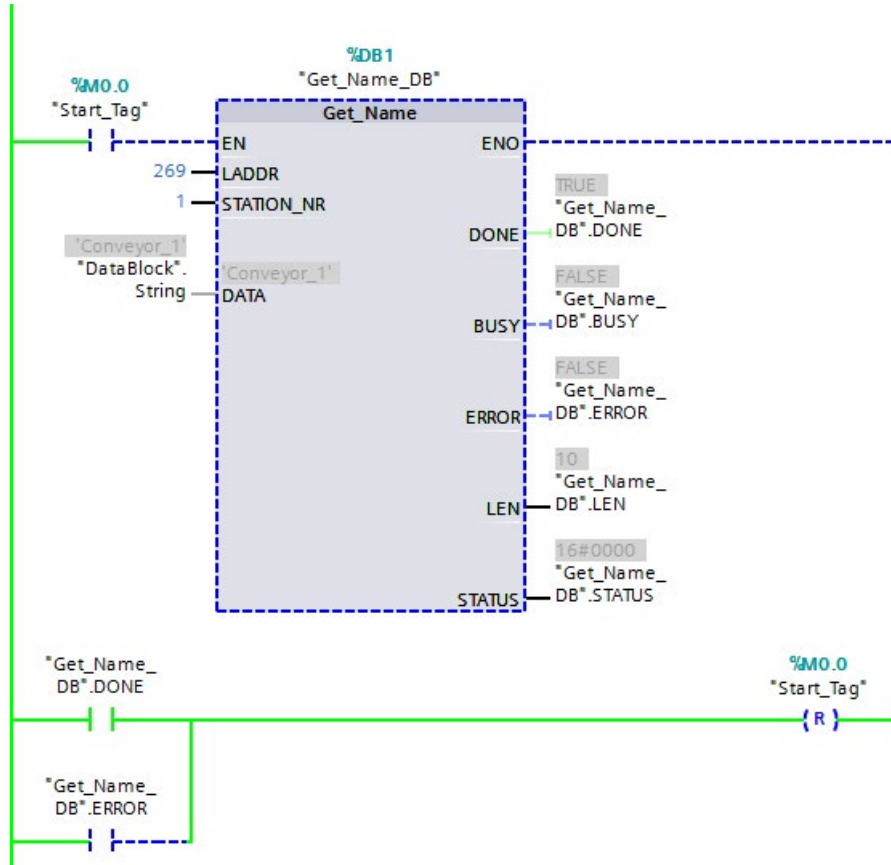


3. Exécuter l'instruction Get_Name :

- Pendant l'exécution de l'instruction, il est possible que le paramètre de sortie BUSY soit mis à 1 ; le paramètre DONE est alors mis à 0.
- Les informations d'erreur s'affichent dans le paramètre de sortie STATUS.



- 4. Achever l'exécution de l'instruction Get_Name :
 - Après l'exécution de l'instruction, le programme écrit "Conveyor_1", le nom de station de l'ET 200SP, dans le bloc de données au paramètre DATA.
 - Le programme écrit "10", le nombre de caractères dans le nom de station, dans le paramètre LEN.



9.7.6 GetStationInfo (Lire l'adresse IP ou MAC d'un périphérique PROFINET IO)

L'instruction "GetStationInfo" lit l'adresse IP ou MAC d'un périphérique PROFINET IO dans le réseau IO local ou d'un périphérique PROFINET IO situé dans un réseau IO de niveau inférieur (connecté par des modules CP/CM).

Remarque

L'instruction GetStationInfo n'est utilisable qu'avec des périphériques PROFINET IO. Vous ne pouvez pas l'utiliser avec des esclaves PROFIBUS DP.

Tableau 9- 155 Instruction GetStationInfo

CONT/LOG	SCL	Description
	<pre>"GetStationInfo_SFB_DB" (REQ:=_bool_in_, LADDR:=_uint_in_, DETAIL:=_uint_in_, MODE:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA:=_variant_inout_);</pre>	<p>Utilisez l'instruction GetStationInfo pour lire l'adresse IP ou MAC d'un périphérique PROFINET IO. Cette instruction permet également de lire l'adresse IP ou MAC d'un périphérique IO situé dans un réseau IO de niveau inférieur (connecté par des modules CP/CM).</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "GetStationInfo_SFB_DB" est le nom du DB d'instance.

Vous désignez le périphérique IO à l'aide de l'identification matérielle de la station dans le paramètre LADDR. Vous trouverez l'ID matérielle à l'emplacement suivant : Variables API > Afficher toutes les variables > onglet Constantes système. Recherchez le périphérique IO dans la colonne Nom et "Hw_Device" dans la colonne Type de données.

Le paramètre MODE permet de sélectionner les informations à lire.

Dans le paramètre DATA, affectez la zone de données dans laquelle l'instruction écrira les données de l'adresse lue. Utilisez la structure "IF_CONF_v4" pour stocker l'adresse IP. Utilisez la structure "IF_CONF_MAC" pour stocker l'adresse MAC.

Validez la lecture des données d'adresse à l'aide du paramètre de commande REQ. Cela nécessite que le périphérique IO soit accessible.

L'instruction signale l'état d'exécution de la tâche de lecture au moyen des paramètres de sortie BUSY, DONE, ERROR et STATUS.

Remarque

Ne désignez le périphérique IO que par l'identification matérielle de la station

La station, le périphérique IO et l'interface PROFINET ont leur propre identification matérielle. Utilisez uniquement l'identification matérielle de la station pour l'instruction GetStationInfo.

Si, par exemple, une interface PROFINET est adressée au moyen du paramètre LADDR, les données d'adresse ne sont pas lues et la CPU génère un code d'erreur "8092".

Utilisez l'instruction "RDREC" pour lire les données d'adresse d'une interface PROFINET intégrée ou d'un module CM/CP dans la configuration centrale.

Paramètres

Le tableau suivant présente les paramètres de l'instruction GetStationInfo :

Paramètre	Déclaration	Type de données	Description
REQ	IN	Bool	Paramètre de commande pour la demande REQ = "1" active la lecture des informations.
LADDR	IN	HW_DEVICE	Identification matérielle de la station du périphérique IO Ce nombre est pris dans les propriétés de la station dans la vue du réseau ou dans l'onglet "Constantes système" de la table de variables par défaut.
DETAIL	IN	HW_SUBMODULE	Le paramètre DETAIL n'est pas utilisé. Ne le connectez pas.
MODE	IN	UNIT	Sélection des données d'adresse à lire : <ul style="list-style-type: none"> • MODE = 1 : paramètre d'adresse selon IPv4 (CPU S7-1200 à partir de la version de firmware V4.2) • MODE = 2 : adresse MAC (CPU S7-1200 à partir de la version de firmware V4.2)
DATA	IN_OUT	Variant	Pointeur désignant la zone dans laquelle le programme écrit les données d'adresse du périphérique IO. Utilisez la structure "IF_CONF_v4" pour MODE = 1 et la structure "IF_CONF_MAC" pour MODE = 2.
DONE	OUT	Bool	Le programme utilisateur a exécuté l'instruction avec succès. Le programme a transféré les données d'adresse dans le paramètre DATA.
BUSY	OUT	Bool	Paramètre d'état : <ul style="list-style-type: none"> • 0 : L'exécution de l'instruction est achevée. • 1 : L'exécution de l'instruction n'est pas encore achevée.

Paramètre	Déclaration	Type de données	Description
ERROR	OUT	Bool	Paramètre d'état : <ul style="list-style-type: none"> • 0 : Pas d'erreur • 1 : Une erreur s'est produite pendant l'exécution de l'instruction. Des informations détaillées sont transmises dans le paramètre STATUS.
STATUS	OUT	Word	Paramètre d'état : ce paramètre est mis à 1 pour la durée d'un appel uniquement. Pour afficher l'état, il faut donc copier STATUS dans une zone de données libre.

Paramètre DATA

- Utilisez la structure "IF_CONF_v4" dans le paramètre DATA pour stocker le paramètre d'adresse conformément à IPv4 :

Octet	Paramètre	Type de données	Valeur initiale	Description
0 ... 1	Id	UINT	30	ID de la structure "IF_CONF_v4"
2 ... 3	Length	UNIT	18	Longueur des données lues en octets (BYTE)
4 ... 5	Mode	UNIT	0	Sans objet pour l'instruction "GetStationInfo" (à 0)
6 ... 9	InterfaceAddress	ARRAY [1..4] of BYTE	-	Adresse IP du périphérique IO au format IP_V4 (par exemple, 192.168.3.10) : <ul style="list-style-type: none"> • addr[1] = 192 • addr[2] = 168 • addr[3] = 3 • addr[4] = 10
10 ... 13	SubnetMask	ARRAY [1..4] of BYTE	-	Masque de sous-réseau du périphérique IO au format IP_V4 (par exemple, 255.255.255.0) : <ul style="list-style-type: none"> • addr[1] = 255 • addr[2] = 255 • addr[3] = 255 • addr[4] = 0
14 ... 17	DefaultRouter	ARRAY [1..4] of BYTE	-	Adresse IP du routeur au format IP_V4 (par exemple, 192.168.3.1) : <ul style="list-style-type: none"> • addr[1] = 192 • addr[2] = 168 • addr[3] = 3 • addr[4] = 1

- Utilisez la structure "IF_CONF_MAC" dans le paramètre DATA pour stocker l'adresse MAC :

Octet	Paramètre	Type de données	Valeur initiale	Description
0 ... 1	Id	UINT	3	ID de la structure "IF_CONF_MAC"
2 ... 3	Length	UNIT	12	Longueur des données lues en octets (BYTE)
4 ... 5	Mode	UNIT	0	Sans objet pour l'instruction "GetStationInfo" (à 0)
6 ... 11	MACAddress	ARRAY [1..6] of BYTE	-	Adresse MAC du périphérique IO (par exemple, 08-00-06-12-34-56) : <ul style="list-style-type: none"> • Mac[1] = 8 • Mac[2] = 0 • Mac[3] = 6 • Mac[4] = 12 • Mac[5] = 34 • Mac[6] = 56

Paramètre STATUS

Code d'erreur* (W#16#...)	Explication
0	Pas d'erreur
7000	Il n'y a pas de tâche en cours.
7001	Premier appel de l'instruction asynchrone GetStationInfo. L'exécution de l'instruction n'est pas encore achevée (BUSY = 1, DONE = 0).
7002	Autre appel de l'instruction asynchrone GetStationInfo. L'exécution de l'instruction n'est pas encore achevée (BUSY = 1, DONE = 0).
8080	La valeur dans le paramètre MODE n'est pas prise en charge.
8090	L'identification matérielle indiquée dans le paramètre LADDR n'est pas configurée.
8092	Le paramètre LADDR ne désigne pas un périphérique PROFINET IO.
8093	Type de données invalide dans le paramètre DATA.
80A0	Informations demandées non lues.
80C0	Le périphérique IO adressé n'est pas accessible.
80C3	Le nombre maximum d'appels simultanés de l'instruction GetStationInfo (10 instances) a été atteint.
* Les codes d'erreur dans l'éditeur de programmes sont affichés sous forme de valeurs entières ou hexadécimales.	

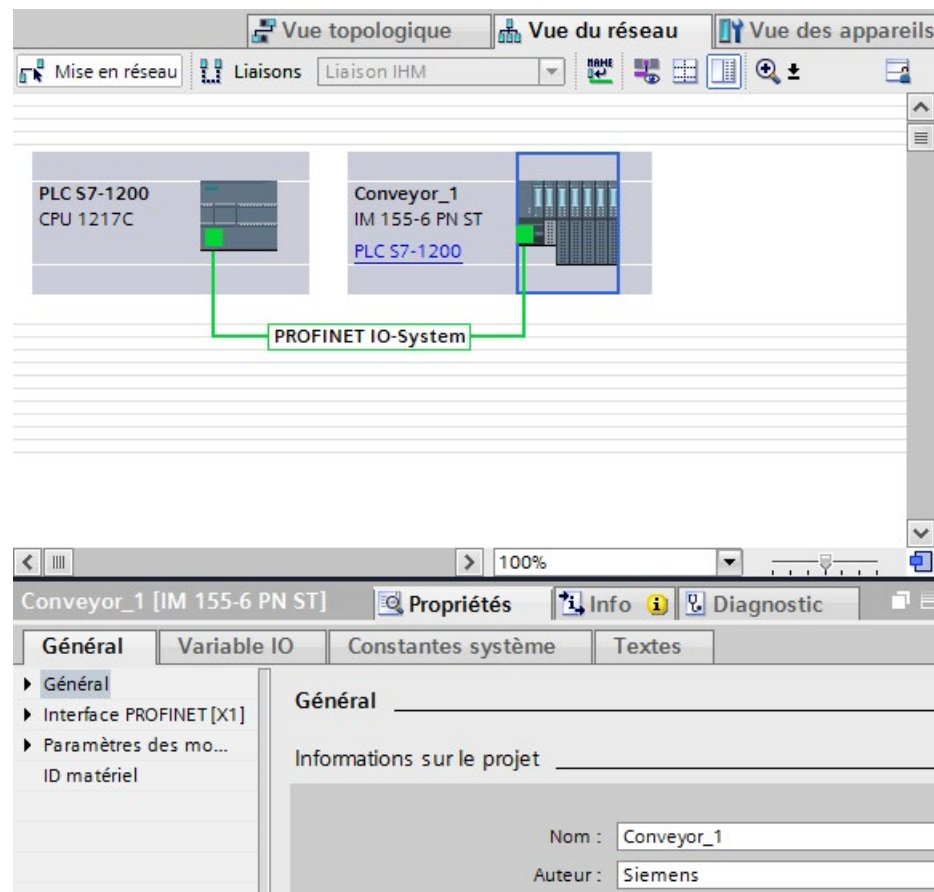
Exemple

Dans l'exemple ci-dessous, vous utilisez l'instruction GetStationInfo pour lire les données d'adresse IP d'un périphérique IO et pour écrire ces informations dans un bloc de données. Les données d'adresse IP comprennent l'adresse IP, le masque de sous-réseau et, le cas échéant, les données d'adresse du routeur.

Le contrôleur IO exécute l'instruction GetStationInfo qui lit les informations d'adresse IP d'un périphérique IO de niveau inférieur (un ET 200SP dans cet exemple) :

1. Configurer l'ET 200SP :

- Créez l'ET 200SP avec le nom de station "Conveyor_1" dans la vue du réseau et affectez-le au même réseau PROFINET IO que la CPU.
- Définissez la CPU comme contrôleur IO pour l'ET 200SP.



2. Affecter les paramètres pour l'instruction GetStationInfo :

- Créez cinq variables et une structure de type de données IF_CONF_v4 dans un bloc de données global afin de stocker les données d'adresse IP. Affectez un nom de votre choix à la structure (dans l'exemple, le nom de la structure est "IP_Address").

GetStationInfo_Global_DB			
	Nom	Type de données	Valeur de départ
1	Static		
2	Execute	Bool	false
3	IP_address	IF_CONF_v4	
4	Id	UInt	30
5	Length	UInt	18
6	Mode	UInt	0
7	InterfaceAddress	IP_V4	
8	ADDR	Array[1..4] of Byte	
9	ADDR[1]	Byte	16#0
10	ADDR[2]	Byte	16#0
11	ADDR[3]	Byte	16#0
12	ADDR[4]	Byte	16#0
13	SubnetMask	IP_V4	
14	DefaultRouter	IP_V4	
15	Done	Bool	false
16	Busy	Bool	false
17	Error	Bool	false
18	Status	Word	16#0

3. Affecter les paramètres pour l'instruction GetStationInfo :

- Saisissez l'ID matérielle du périphérique IO dans le paramètre LADDR. L'identification matérielle identifie le produit de manière unique ; dans cet exemple, l'ID matérielle est "270". Vous trouverez l'ID matérielle à l'emplacement suivant : Variables API > Afficher toutes les variables > onglet Constantes système.

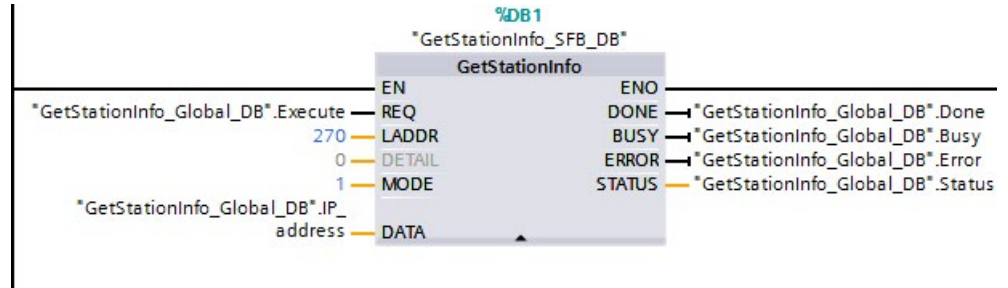
Recherchez le périphérique IO dans la colonne Nom et "Hw_Device" dans la colonne Type de données. La valeur associée est l'ID matérielle que vous avez saisie dans le paramètre LADDR.

- Sélectionnez "1" (lire les paramètres d'adresse conformément à IPv4) pour le paramètre MODE.
- Connectez la structure IF_CONF_v4 dans le paramètre DATA.

Remarque

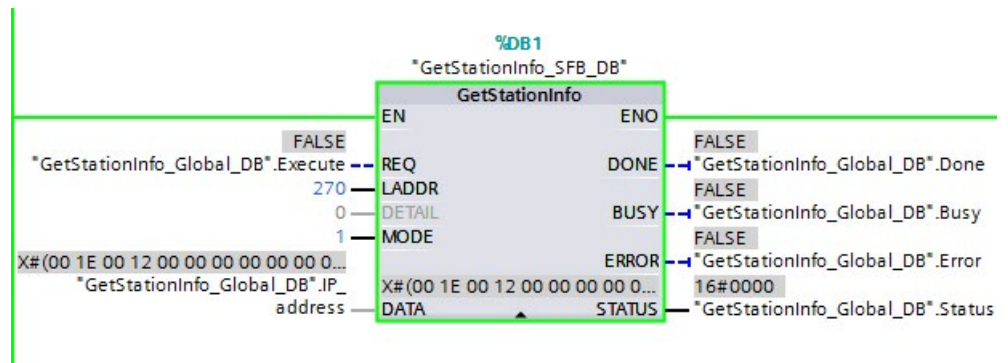
Lorsque vous configurez la variable pour le paramètre DATA à l'aide de la liste déroulante, sélectionnez le DB ("GetStationInfo_Global_DB" dans l'exemple) et la variable ("IP address" dans l'exemple). Pour lire le type de données IF_CONF_v4 entier, vous devez effacer le point qui suit "IP address" afin d'obtenir le résultat final suivant : "GetStationInfo_Global_DB".IP address

- Définissez les variables API de votre DB global pour les paramètres de sortie de l'instruction.



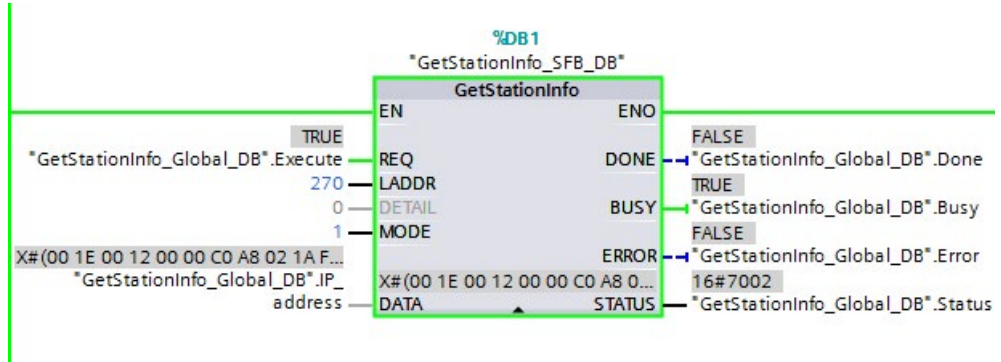
4. Exécuter l'instruction GetStationInfo :

- Lorsque l'entrée REQ est à 0 (FALSE), l'instruction n'affiche pas d'informations d'adresse IP dans le paramètre d'entrée/sortie DATA ni d'informations d'erreur dans le paramètre de sortie STATUS.



5. Achever l'exécution de l'instruction GetStationInfo :

- Lorsque l'entrée REQ est à 1 (TRUE), le programme exécute l'instruction et écrit l'adresse IP dans le bloc de données. Le programme écrit l'adresse IP, "C0 A8 02 1A" (valeur décimale de "192.168.2.26"), dans le paramètre d'entrée/sortie DATA.

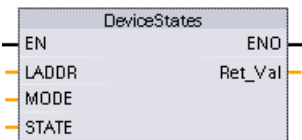


GetStationInfo_Global_DB				
	Nom	Type de données	Valeur de départ	Valeur de visualisati.
1	Static			
2	Execute	Bool	false	TRUE
3	IP_address	IF_CONF_v4		
4	Id	UInt	30	30
5	Length	UInt	18	18
6	Mode	UInt	0	0
7	InterfaceAddress	IP_V4		
8	ADDR	Array[1..4] of Byte		
9	ADDR[1]	Byte	16#0	16#C0
10	ADDR[2]	Byte	16#0	16#A8
11	ADDR[3]	Byte	16#0	16#02
12	ADDR[4]	Byte	16#0	16#1A
13	SubnetMask	IP_V4		
14	DefaultRouter	IP_V4		
15	Done	Bool	false	TRUE
16	Busy	Bool	false	FALSE
17	Error	Bool	false	FALSE
18	Status	Word	16#0	16#0000

9.7.7 Instruction DeviceStates

Vous pouvez utiliser l'instruction DeviceStates pour renvoyer les états de toutes les périphéries décentralisées IO esclaves connectées à un maître IO décentralisé défini.

Tableau 9- 156 Instruction DeviceStates

CONT/LOG	SCL	Description
	<pre>ret_val := DeviceStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	<p>DeviceStates permet d'extraire les états de fonctionnement des périphériques IO d'un sous-système IO. Après l'exécution, le paramètre STATE contient l'état d'erreur de chaque périphérique IO dans une liste de bits (pour les paramètres LADDR et MODE affectés). Cette information correspond à l'état des appareils présenté dans la vue de diagnostic de STEP 7.</p> <p>L'entrée LADDR de DeviceStates utilise l'identificateur matériel d'une interface IO décentralisée. Dans TIA Portal, les identifications matérielles pour un API figurent dans les types de données "HW_IOSYSTEM" dans l'onglet Constantes système de la table des variables API.</p>

9.7 Diagnostic (PROFINET ou PROFIBUS)

Tableau 9- 157 Types de données pour les paramètres

Paramètre et type		Type de données	Description
LADDR	IN	HW_IOSYSTEM	Adresse logique (identificateur pour le système IO)
MODE	IN	UInt	Prend en charge cinq modes de fonctionnement. L'entrée de MODE détermine quelles données seront renvoyées à l'adresse indiquée pour l'information STATE. Les modes sont les suivants : <ul style="list-style-type: none"> • 1 : Configuration de l'appareil active • 2 : Appareil défectueux • 3 : L'appareil est désactivé. • 4 : Appareil disponible • 5 : Problème dans l'appareil
RET_VAL	OUT	Int	Code d'erreur d'exécution
STATE ¹	InOut	Variant	Mémoire tampon qui reçoit l'état d'erreur de chaque périphérique : Le type de données choisi pour le paramètre STATE peut être tout type de données binaire (Bool, Byte, Word ou DWord) ou un tableau de type binaire. <ul style="list-style-type: none"> • Le bit 0 du premier octet du paramètre STATE renvoyé est un bit récapitulatif. Lorsqu'il est mis à VRAI, il signale que d'autres données sont disponibles. • Les données renvoyées par le paramètre STATE montrent une corrélation biunivoque entre une adresse de bit et une adresse de périphérie décentralisée. Cette adresse de périphérie est VRAI pour PROFIBUS et PROFINET. Par exemple, le bit 4 dans le premier octet est en corrélation avec l'adresse PROFIBUS 4 ou l'appareil PROFINET numéro 4.

¹ Pour PROFIBUS DP, la longueur des informations d'état est de 128 bits. Pour PROFINET IO, cette longueur est de 1024 bits.

Après l'exécution, le paramètre STATE contient l'état d'erreur de chaque périphérique IO sous forme de liste de bits (pour les paramètres LADDR et MODE affectés).

Tableau 9- 158 Codes d'erreur

RET_VAL (W#16#...)	Description
0	Pas d'erreur
8091	LADDR n'existe pas.
8092	LADDR ne correspond pas à un système IO.
8093	Type de données invalide affecté au paramètre STATE : Les types de données valides sont Bool, Byte, Word ou Dword ou un tableau de Bool, Byte, Word ou Dword.
80Bx	L'instruction DeviceStates n'est pas prise en charge par la CPU pour cette adresse LADDR.
8452	Les données d'état complètes sont trop grandes pour le paramètre STATE affecté. La mémoire tampon STATE contient un résultat partiel.

9.7.7.1 Exemples de configuration de DeviceStates

Exemple PROFIBUS

L'exemple PROFIBUS comporte les éléments suivants :

- 16 périphériques PROFIBUS nommés "DPSlave_10" à "DPSlave_25"
- Les 16 périphériques PROFIBUS utilisent les adresses PROFIBUS 10 à 25, respectivement.
- Chaque esclave est configuré avec des modules d'E/S multiples.
- Les quatre premiers octets d'information du paramètre STATE renvoyé sont affichés.

MODE	Exemple 1 : fonctionnement normal sans erreurs	Exemple 2 : esclave PROFIBUS DPSlave_12 avec un module débroché	Exemple 3 : esclave PROFIBUS DPSlave_12 déconnecté
1: Configuration de l'appareil active	0x01FC_FF03	0x01FC_FF03	0x01FC_FF03
2: Appareil défectueux	0x0000_0000	0x0110_0000	0x0110_0000
3: L'appareil est désactivé.	0x0000_0000	0x0000_0000	0x0000_0000
4: Appareil disponible	0x01FC_FF03	0x01FC_FF03	0x01EC_FF03
5: Problème dans l'appareil	0x0000_0000	0x0110_0000	0x0110_0000

Les quatre tableaux suivants montrent une distribution binaire des quatre octets de données analysés :

Tableau 9- 159 Exemple 1 : pas d'erreur : 0x01FC_FF03 est renvoyé pour le MODE 1 (Configuration de l'appareil active).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0xFC	Bit 15 1111-1100 Bit 8	
Octet 3 0xFF	Bit 23 1111-1111 Bit 16	
Octet 4 0x03	Bit 31 0000-0011 Bit 24	

Les périphériques sont configurés dans les adresses 10 (Bit 10) à 25 (Bit 25).

Aucun périphérique n'est configuré dans les adresses 1 à 9.

Données MODE 4 (Appareil disponible) correspondent à MODE 1 (Configuration de l'appareil active) de telle sorte que les périphériques configurés correspondent aux périphériques existants.

9.7 Diagnostic (PROFINET ou PROFIBUS)

Tableau 9- 160 Exemple 2 : un module a été débrogé de l'esclave PROFIBUS "DPSlave_12". 0x0110_0000 est renvoyé pour le MODE 2 (Appareil défectueux).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0x10	Bit 15 0001-0000 Bit 8	
Octet 3 0x00	Bit 23 0000-0000 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

L'appareil 12 (Bit 12) est défini comme défectueux.

MODE 5 (Problème dans l'appareil) renvoie la même information que MODE 2 (Appareil défectueux).

Tableau 9- 161 Exemple 2 (suite) : un module a été débrogé de l'esclave PROFIBUS "DPSlave_12". 0x01FC_FF03 est renvoyé pour le MODE 4 (Appareil disponible).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0xFC	Bit 15 1111-1100 Bit 8	
Octet 3 0xFF	Bit 23 1111-1111 Bit 16	
Octet 4 0x03	Bit 31 0000-0011 Bit 24	

Quand bien même l'appareil 12 (Bit 12) présenterait une erreur comme indiqué dans MODE 2 (Appareil défectueux) ci-dessus, l'appareil continue de fonctionner dans le réseau, faisant que MODE 4 (Appareil disponible) affiche l'appareil comme un "appareil disponible".

Tableau 9- 162 Exemple 3 : l'esclave PROFIBUS "DPSlave_12" est déconnecté (câble déconnecté ou coupure de courant) du réseau PROFIBUS. "DPSlave_12" est toujours détecté comme un appareil défectueux ainsi qu'une erreur dans l'appareil. Différence : "DPSlave_12" n'est plus détecté comme un appareil existant. 0x01EC_FF03 est renvoyé pour le MODE 4 (Appareil disponible).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0xEC	Bit 15 1110-1100 Bit 8	
Octet 3 0xFF	Bit 23 1111-1111 Bit 16	
Octet 4 0x03	Bit 31 0000-0011 Bit 24	

L'appareil 12 (Bit 12) est défini comme inexistant. Grâce à cette exception, les appareils 10 à 25 continuent d'être signalés comme existants.

Exemple PROFINET

L'exemple PROFINET comporte les éléments suivants :

- 16 esclaves PROFINET nommés "et200s_1" à "et200s_16"
- Les 16 périphériques PROFINET utilisent les numéros d'appareil PROFINET 1 à 16, respectivement.
- Chaque esclave est configuré avec des modules d'E/S multiples.
- Les quatre premiers octets d'information du paramètre STATE renvoyé sont affichés.

MODE	Exemple 1 : fonctionnement normal sans erreurs	Exemple 2 : esclave PROFINET et200s_1 avec module débrouché	Exemple 3 : esclave PROFINET et200s_1 déconnecté
1: Configuration de l'appareil active	0xFFFF_0100	0xFFFF_0100	0xFFFF_0100
2 - L'appareil est défectueux.	0x0000_0000	0x0300_0000	0x0300_0000
3 - L'appareil est désactivé.	0x0000_0000	0x0000_0000	0x0000_0000
4 - L'appareil est disponible.	0xFFFF_0100	0xFFFF_0100	0xFDFF_0100
5 - Problème dans l'appareil	0x0000_0000	0x0300_0000	0x0300_0000

Les quatre tableaux suivants montrent une distribution binaire des quatre octets de données analysés :

Tableau 9- 163 Exemple 1 : pas d'erreur : 0xFFFF_0100 est renvoyé pour le MODE 1 (Configuration de l'appareil active).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0xFF	Bit 15 1111-1111 Bit 8	
Octet 3 0x01	Bit 23 0000-0001 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Les périphériques sont configurés dans les adresses 1 (Bit 1) à 16 (Bit 16).

Aucun périphérique n'est configuré dans les adresses 1 à 9.

Données MODE 4 (Appareil disponible) correspondent à MODE 1 (Configuration de l'appareil active) de telle sorte que les périphériques configurés correspondent aux périphériques existants.

9.7 Diagnostic (PROFINET ou PROFIBUS)

Tableau 9- 164 Exemple 2 : un module a été débrouché de l'esclave PROFINET "et200s_1".
0x0300_0000 est renvoyé pour le MODE 2 (Appareil défectueux).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x03	Bit 7 0000-0011 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0x00	Bit 15 0000-0000 Bit 8	
Octet 3 0x00	Bit 23 0000-0000 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

L'appareil 1 (Bit 1) est défini comme défectueux. Comme l'appareil reste disponible, MODE 4 (Appareil disponible) affiche les mêmes données que lors d'un fonctionnement normal.
MODE 5 (Problème dans l'appareil) renvoie la même information que MODE 2 (Appareil défectueux).

Tableau 9- 165 Exemple 2 (suite) : un module a été débrouché de l'esclave PROFIBUS "et200s_1".
0xFFFF_0100 est renvoyé pour le MODE 4 (Appareil disponible).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0xFF	Bit 15 1111-1111 Bit 8	
Octet 3 0x01	Bit 23 0000-0001 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Quand bien même l'appareil 1 (Bit 1) présenterait une erreur comme indiqué dans MODE 2 (Appareil défectueux) ci-dessus, l'appareil continue de fonctionner dans le réseau, faisant que MODE 4 (Appareil disponible) affiche l'appareil comme un "appareil disponible".

Tableau 9- 166 Exemple 3 : l'esclave PROFINET "et200s_1" est déconnecté (câble déconnecté ou coupure de courant) du réseau PROFINET. 0xFDFF_0100 est renvoyé pour le MODE 4 (Appareil disponible).

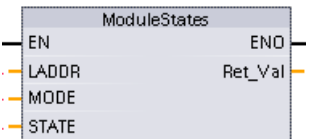
Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0xFD	Bit 7 1111-1101 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0xFF	Bit 15 1111-1111 Bit 8	
Octet 3 0x01	Bit 23 0000-0001 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

L'Appareil 1 (Bit 1) n'existe pas. Les Appareils 2 (Bit 2) à 16 (Bit 16) existent.

9.7.8 Instruction ModuleStates

Vous pouvez utiliser l'instruction ModuleStates pour renvoyer l'état de tous les modules dans une station PROFIBUS ou PROFINET.

Tableau 9- 167 Instruction ModuleStates

CONT/LOG	SCL	Description
	<pre>ret_val := ModuleStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout);</pre>	<p>ModuleStates permet d'extraire les états de fonctionnement des modules d'E/S. Après l'exécution, le paramètre STATE contient l'état d'erreur de chaque module IO dans une liste de bits (pour les paramètres LADDR et MODE affectés). Cette information correspond à l'état des modules présenté dans la vue de diagnostic de STEP 7.</p> <p>L'entrée LADDR de ModuleStates utilise un identificateur matériel d'une station IO décentralisée, pas celui du module de tête même. Vous pouvez trouver l'identificateur matériel en sélectionnant toute la station dans la vue du réseau où il figure dans la section correspondante sous les propriétés. Il figure également dans les types de données "Hw_Device" et "Hw_DpSlave" dans l'onglet Constantes système de la Table des variables API.</p>

9.7 Diagnostic (PROFINET ou PROFIBUS)

Tableau 9- 168 Types de données pour les paramètres

Paramètre et type		Type de données	Description
LADDR	IN	HW_DEVICE	Adresse logique (identificateur pour les modules d'E/S)
MODE	IN	UInt	Prend en charge cinq modes de fonctionnement. L'entrée de MODE détermine quelles données seront renvoyées à l'adresse indiquée pour l'information STATE. Les modes sont les suivants : <ul style="list-style-type: none"> • 1 : Configuration du module active • 2 : Module défaillant • 3 : Module désactivé • 4 : Module disponible • 5 : Problème dans le module
RET_VAL	OUT	Int	Etat (code d'erreur)
STATE ¹	InOut	Variant	Mémoire tampon qui reçoit l'état d'erreur de chaque module : Le type de données utilisé pour le paramètre STATE peut être tout type de données binaire (Bool, Byte, Word ou DWord) ou un tableau de type binaire. <ul style="list-style-type: none"> • Le bit 0 du premier octet du paramètre STATE renvoyé est un bit récapitulatif. Lorsqu'il est mis à VRAI, il signale que d'autres données sont disponibles. • Les données renvoyées par le paramètre STATE montrent une corrélation biunivoque entre une adresse de bit et une adresse de module. Cette adresse d'emplacement est VRAI pour PROFIBUS et PROFINET. Exemple : pour un ET 200SP ayant un module de tête, un module d'alimentation et deux modules d'E/S, le Bit 1 du premier octet est en corrélation avec le module de tête, Bit 2 avec le module d'alimentation et Bits 3 et 4 avec les modules d'E/S, respectivement.

¹ 128 bits au maximum peuvent être affectés. Le nombre de bits requis dépend de votre utilisation des modules d'E/S.

Tableau 9- 169 Codes d'erreur

RET_VAL (W#16#...)	Description
0	Pas d'erreur
8091	Le module identifié par LADDR n'existe pas.
8092	Le module identifié par LADDR ne correspond pas à un périphérique IO.
8093	Type de données pour paramètre STATE invalide : Les types de données valides sont Bool, Byte, Word ou Dword ou un tableau de Bool, Byte, Word ou Dword.
80Bx	L'instruction ModuleStates n'est pas prise en charge par cette CPU pour cette adresse LADDR.
8452	Les données d'état complètes sont trop grandes pour le paramètre STATE affecté. La mémoire tampon STATE contient un résultat partiel.

9.7.8.1 Exemples de configuration de ModuleStates

Exemple PROFIBUS

L'exemple PROFIBUS comporte les éléments suivants :

- 16 périphériques PROFIBUS nommés "DPSlave_10" à "DPSlave_25"
- Les 16 périphériques PROFIBUS utilisent les adresses PROFIBUS 10 à 25, respectivement.
- Chaque esclave est configuré avec des modules d'E/S multiples.
- Cet exemple utilise le paramètre LADDR de l'esclave PROFIBUS "DPSlave_12" qui contient un module de tête, un module d'alimentation et deux modules d'E/S.
- Les quatre premiers octets d'information du paramètre STATE renvoyé sont affichés.

MODE	Exemple 1 : fonctionnement normal sans erreurs	Exemple 2 : esclave PROFIBUS DPSlave_12 avec module débouché	Exemple 3 : esclave PROFIBUS DPSlave_12 déconnecté
1: Configuration du module active	0x1F00_0000	0x1F00_0000	0x1F00_0000
2: Module défaillant	0x0000_0000	0x0900_0000	0x1F00_0000
3: Module désactivé	0x0000_0000	0x0000_0000	0x0000_0000
4: Module disponible	0x1F00_0000	0x1700_0000	0x0000_0000
5: Problème dans le module	0x0000_0000	0x0900_0000	0x1F00_0000

Les quatre tableaux suivants montrent une distribution binaire des quatre octets de données analysés :

Tableau 9- 170 Exemple 1 : pas d'erreur : 0x1F00_0000 est renvoyé pour le MODE 1 (Configuration du module active).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x1F	Bit 7 0001-1111 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0x00	Bit 15 0000-0000 Bit 8	
Octet 3 0x00	Bit 23 0000-0000 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Les Emplacements 1 (Bit 1) à 4 (Bit 4) contiennent des modules. Les Emplacements 5 (Bit 5) et au-delà ne contiennent pas de module.

MODE 4 (Module disponible) correspondent à MODE 1 (Configuration du module active) de telle sorte que les modules configurés correspondent aux modules existants.

9.7 Diagnostic (PROFINET ou PROFIBUS)

Tableau 9- 171 Exemple 2 : un module a été débroché de l'esclave PROFIBUS "DPSlave_12". 0x0900_0000 est renvoyé pour le MODE 2 (Module défaillant).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x09	Bit 7 0000-1001 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0x00	Bit 15 0000-0000 Bit 8	
Octet 3 0x00	Bit 23 0000-0000 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Seul le module 3 (Bit 3) est défini comme défaillant. Tous les autres modules fonctionnent.

Tableau 9- 172 Exemple 2 (suite) : un module a été débroché de l'esclave PROFIBUS "DPSlave_12". 0x1700_0000 est renvoyé pour le MODE 4 (Module disponible).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x17	Bit 7 0001-0111 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0x00	Bit 15 0000-0000 Bit 8	
Octet 3 0x00	Bit 23 0000-0000 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Le module 3 (Bit 3) est signalé comme étant manquant. Les modules 1, 2 et 4 (Bits 1, 2 et 4) sont signalés comme étant disponibles.

Tableau 9- 173 Exemple 3 : l'esclave PROFIBUS "DPSlave_12" est déconnecté (câble déconnecté ou coupure de courant) du réseau PROFIBUS. 0x1F00_0000 est renvoyé pour le MODE 2 (Module défaillant).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x1F	Bit 7 0001-1111 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0x00	Bit 15 0000-0000 Bit 8	
Octet 3 0x00	Bit 23 0000-0000 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Les modules dans les emplacements 1 à 4 (Bits 1 à 4) sont tous définis comme défaillants étant donné que le périphérique manque.

MODE 5 (Problème dans le module) affiche la même information que MODE 2 (Module défaillant).

Exemple PROFINET

L'exemple PROFINET comporte les éléments suivants :

- 16 esclaves PROFINET nommés "et200s_1" à "et200s_16"
- Les 16 périphériques PROFINET utilisent les numéros d'appareil PROFINET 1 à 16, respectivement.
- Chaque esclave est configuré avec des modules d'E/S multiples.
- Cet exemple utilise l'esclave PROFINET "et200s_1" qui contient un module de tête, un module d'alimentation et 18 modules d'E/S.
- Les quatre premiers octets d'information du paramètre STATE renvoyé sont affichés.

MODE	Exemple 1 : fonctionnement normal sans erreurs	Exemple 2 : esclave PROFINET et200s_1 avec module débouché	Exemple 3 : esclave PROFINET et200s_1 déconnecté
1: Configuration du module active	0xFFFF_1F00	0xFFFF_1F00	0xFFFF_1F00
2: Module défaillant	0x0000_0000	0x0180_0000	0xFFFF_1F00
3: Module désactivé	0x0000_0000	0x0000_0000	0x0000_0000
4: Module disponible	0xFFFF_1F00	0xFF7F_1F00	0x0000_0000
5: Problème dans le module	0x0000_0000	0x0180_0000	0xFFFF_1F00

Les quatre tableaux suivants montrent une distribution binaire des quatre octets de données analysés :

Tableau 9- 174 Exemple 1 : pas d'erreur : 0xFFFF_1F00 est renvoyé pour le MODE 1 (Configuration du module active).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0xFF	Bit 15 1111-1111 Bit 8	
Octet 3 0x1F	Bit 23 0001-1111 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Les Emplacements 1 (Bit 1) à 20 (Bit 20) contiennent des modules. Les Emplacements 21 (Bit 21) et au-delà ne contiennent pas de module.

MODE 4 (Module disponible) correspondent à MODE 1 (Configuration du module active) de telle sorte que les modules configurés correspondent aux modules existants.

9.7 Diagnostic (PROFINET ou PROFIBUS)

Tableau 9- 175 Exemple 2 : un module a été débrogé de l'esclave PROFINET "et200s_1".
0x0180_0000 est renvoyé pour le MODE 2 (Module défaillant).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0x80	Bit 15 1000-0000 Bit 8	
Octet 3 0x00	Bit 23 0000-0000 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Seul le module 15 (Bit 15) est défini comme défaillant. Tous les autres modules fonctionnent.

Tableau 9- 176 Exemple 2 (suite) : un module a été débrogé de l'esclave PROFIBUS "et200s_1".
0xFF7F_1F00 est renvoyé pour le MODE 4 (Module disponible).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0x7F	Bit 15 0111-1111 Bit 8	
Octet 3 0x1F	Bit 23 0001-1111 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Le module 15 (Bit 15) est signalé comme étant manquant. Les modules 1 à 14 (Bits 1 à 14) et 16 à 20 (Bits 16 à 20) sont signalés comme étant disponibles.

Tableau 9- 177 Exemple 3 : l'esclave PROFINET "et200s_1" est déconnecté (câble déconnecté ou coupure de courant) du réseau PROFINET. 0xFFFF_1F00 est renvoyé pour le MODE 2 (Module défaillant).

Octet et valeur	Configuration binaire et valeur	Remarques
Octet 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 est vrai ; les données sont disponibles.
Octet 2 0xFF	Bit 15 1111-1111 Bit 8	
Octet 3 0x1F	Bit 23 0001-1111 Bit 16	
Octet 4 0x00	Bit 31 0000-0000 Bit 24	

Les modules dans les emplacements 1 à 20 (Bits 1 à 20) sont tous définis comme défaillants étant donné que le périphérique manque.

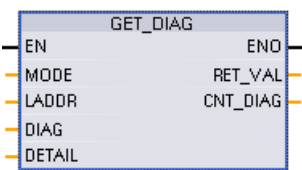
MODE 5 (Problème dans le module) affiche la même information que MODE 2 (Module défaillant).

9.7.9 GET_DIAG (Lire l'information de diagnostic)

Description

L'instruction "GET_DIAG" permet de lire les informations de diagnostic d'un périphérique. Vous sélectionnez le périphérique à l'aide du paramètre LADDR. Le paramètre MODE vous permet de sélectionner les informations de diagnostic à lire.

Tableau 9- 178 Instruction GET_DIAG

CONT/LOG	SCL	Description
 <p>Le diagramme illustre l'instruction GET_DIAG avec les paramètres suivants :</p> <ul style="list-style-type: none"> Entrées (à gauche) : EN, MODE, LADDR, DIAG, DETAIL Sorties (à droite) : END, RET_VAL, CNT_DIAG 	<pre>ret_val := GET_DIAG(mode:=_uint_in_, laddr:=_word_in_, cnt_diag=>_uint_out_, diag:=_variant_inout_, detail:=_variant_inout_);</pre>	Permet de lire les informations de diagnostic du périphérique affecté.

Paramètres

Le tableau suivant présente les paramètres de l'instruction GET_DIAG :

Tableau 9- 179 Types de données pour les paramètres

Paramètre et type		Type de données	Description
MODE	IN	UInt	Le paramètre MODE permet de sélectionner les données de diagnostic à transmettre.
LADDR	IN	HW_ANY (Word)	ID matériel de l'appareil
RET_VAL	OUT	Int	Etat de l'instruction
CNT_DIAG	OUT	UInt	Nombre de détails de diagnostic transmis
DIAG	InOut	Variant	Pointeur désignant la zone de données pour le stockage des informations de diagnostic du mode sélectionné
DETAILS	InOut	Variant	Pointeur désignant la zone de données pour le stockage des détails de diagnostic conformément au mode sélectionné

Paramètre MODE

Selon la valeur du paramètre MODE, des données de diagnostic différentes sont transmises aux paramètres de sortie DIAG, CNT_DIAG et DETAILS :

Tableau 9- 180 Paramètre MODE

MODE	Description	DIAG	CNT_DIAG	DETAILS
0	Transmission de toutes les informations de diagnostic prises en charge pour un module sous forme de double mot (DWord), bit X=1 indiquant que le mode X est pris en charge.	Chaîne de bits des modes pris en charge sous forme de double mot (DWord), bit X=1 indiquant que le mode X est pris en charge. La CPU S7-1200 ne prend pas en compte le paramètre LADDR lorsque le paramètre MODE est égal à 0.	0	-
1	Transmission de l'état inhérent de l'objet matériel adressé.	Etat de diagnostic : Transmission conformément à la structure DIS. (Remarque : voir les informations sur la structure DIS ci-après et l'exemple d'instruction GET_DIAG à la fin du paragraphe)	0	-
2	Transmission de l'état de tous les modules subordonnés de l'objet matériel adressé.	Transmission des données de diagnostic conformément à la structure DNN. (Remarque : voir les informations sur la structure DNN ci-après et l'exemple d'instruction GET_DIAG à la fin du paragraphe)	0	-

Structure DIS

Lorsque le paramètre MODE a la valeur 1, les informations de diagnostic sont transmises conformément à la structure DIS. Le tableau suivant donne la signification des différents paramètres :

Tableau 9- 181 Structure DIS (source d'informations de diagnostic)

Paramètre	Type de données	Valeur	Description
MaintenanceState	DWord	Enum	
		0	Pas de maintenance nécessaire
		1	Le module ou l'appareil est désactivé.
		2	-
		3	-
		4	-
		5	Maintenance nécessaire
		6	Maintenance requise
		7	Erreur
		8	Etat inconnu / erreur dans le module subordonné
		9	-
Componentstate Detail	DWord	Tableau de bits	Etat des sous-modules du module : <ul style="list-style-type: none"> • Bits 0 à 15 : Message d'état du module • Bits 16 à 31 : Message d'état de la CPU
		0 à 2 (enum)	Information supplémentaire : <ul style="list-style-type: none"> • Bit 0 : Pas d'information supplémentaire • Bit 1 : Transfert interdit
		3	Bit 3 = 1 : Au moins une voie prend en charge des qualifieurs pour le diagnostic.
		4	Bit 4 = 1 : Maintenance nécessaire pour au moins une voie ou un composant
		5	Bit 5 = 1 : Maintenance requise pour au moins une voie ou un composant
		6	Bit 6 = 1 : Erreur dans au moins une voie ou un composant
		7 à 10	Réservés (toujours = 0)
		11 à 14	Bit 11 = 1 : PNIO, sous-module correct Bit 12 = 1 : PNIO, module de remplacement Bit 13 = 1 : PNIO, module incorrect Bit 14 = 1 : PNIO, module déconnecté
		15	Réservés (toujours = 0)

Paramètre	Type de données	Valeur	Description
		16 à 31	Information d'état pour modules générée par la CPU : Bit 16 = 1 : Module désactivé Bit 17 = 1 : Opération CiR active Bit 18 = 1 : Entrée non disponible Bit 19 = 1 : Sortie non disponible Bit 20 = 1 : Débordement de la mémoire tampon de diagnostic Bit 21 = 1 : Diagnostic non disponible Bit 22 à 31: Réservés (toujours = 0)
OwnState	Uint16	Enum	La valeur du paramètre OwnState décrit l'état de maintenance du module.
		0	Pas d'erreur
		1	Le module ou l'appareil est désactivé.
		2	Maintenance nécessaire
		3	Maintenance requise
		4	Erreur
		5	Le module ou l'appareil n'est pas accessible depuis la CPU (vaut pour les modules et appareils en dessous d'une CPU).
		6	Les entrées/sorties ne sont pas disponibles.
IO State	Uint16	Tableau de bits	Etat d'E/S du module
		0	Bit 0 = 1 : Pas de maintenance nécessaire
		1	Bit 1 = 1 : Le module ou l'appareil est désactivé.
		2	Bit 2 = 1 : Maintenance nécessaire
		3	Bit 3 = 1 : Maintenance requise
		4	Bit 4 = 1 : Erreur
		5	Bit 5 = 1 : Le module ou l'appareil n'est pas accessible depuis la CPU (vaut pour les modules et appareils en dessous d'une CPU).
		6	Qualificateur ; bit 7 = 1 si les bits 0, 2 ou 3 sont à 1
		7	Les entrées/sorties ne sont pas disponibles.
8 à 15	Réservés (toujours = 0)		
OperatingState	UInt16	Enum	
		0	-
		1	A l'état ARRET / mise à jour du firmware
		2	À l'état ARRÊT / effacement général de la mémoire
		3	À l'état ARRÊT / auto-démarrage
		4	A l'état ARRET
		5	Effacement général de la mémoire
		6	A l'état MISE EN ROUTE
		7	A l'état MARCHE
		8	-
9	A l'état ATTENTE		

Paramètre	Type de données	Valeur	Description
		10	-
		11	-
		12	Module défaillant
		13	-
		14	Absence d'alimentation
		15	CiR
		16	À l'état ARRÊT / sans DIS
		17	A l'état
		18	
		19	
		20	

Structure DNN

Lorsque le paramètre MODE a la valeur 2, les détails de diagnostic sont transmis conformément à la structure DNN. Le tableau suivant donne la signification des différents paramètres :

Tableau 9- 182 Structure DNN (noeud de navigation de diagnostic)

Paramètre	Type de données	Valeur	Description
SubordinateState	UINT	Enum	Etat du module subordonné (voir paramètre OwnState de la structure DIS)
SubordinateIOState	WORD	Bitarray	Etat des entrées et sorties du module subordonné (voir paramètre IO State de la structure DIS)
DNNmode	WORD	Bitarray	<ul style="list-style-type: none"> • Bit 0 = 0 : Diagnostic activé • Bit 0 = 1 : Diagnostic désactivé • Bits 1 à 15 : Réservés

Paramètre RET_VAL

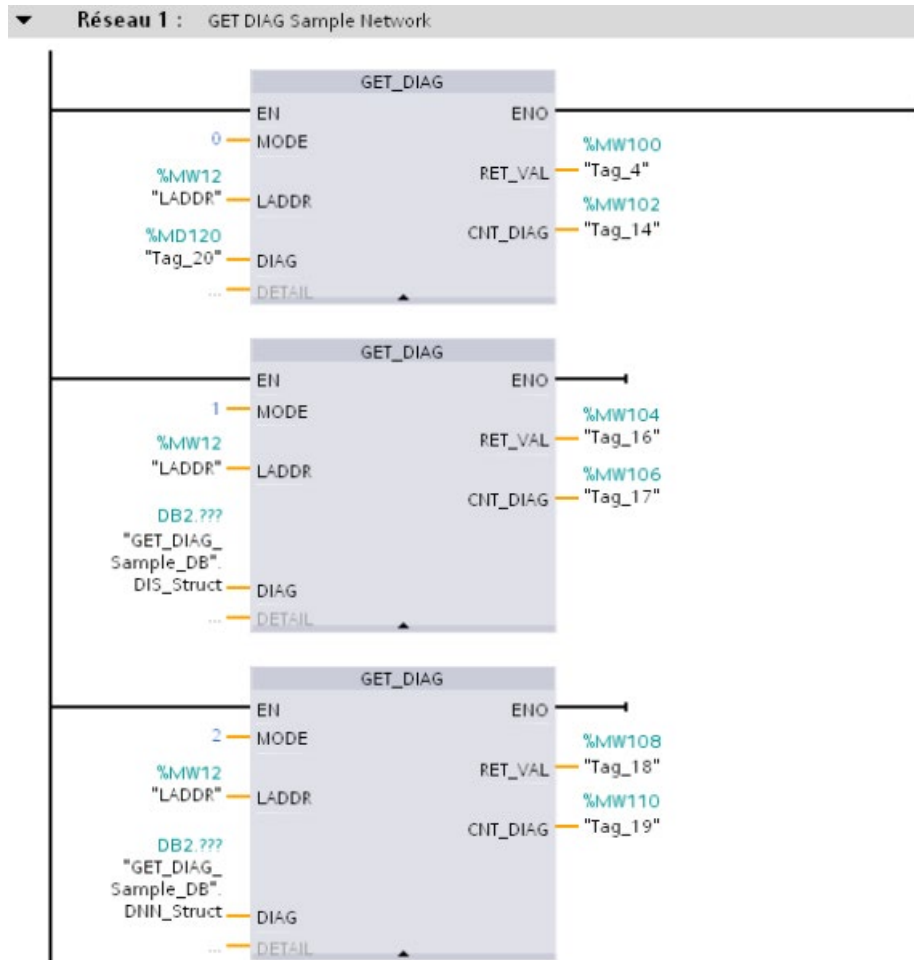
Tableau 9- 183 Codes d'erreur du paramètre RET_VAL

Code d'erreur (W#16#...)	Description
0	Pas d'erreur
8080	La valeur dans le paramètre MODE n'est pas prise en charge.
8081	Le type dans le paramètre DIAG n'est pas pris en charge pour le mode sélectionné (paramètre MODE).
8082	Le type dans le paramètre DETAILS n'est pas pris en charge pour le mode sélectionné (paramètre MODE).
8090	LADDR n'existe pas.
8091	La voie sélectionnée dans le paramètre CHANNEL n'existe pas.
80C1	Ressources insuffisantes pour une exécution en parallèle

Exemple

Le réseau CONT et le DB ci-après montrent comment utiliser les trois modes avec les trois structures :

- DIS
- DNN



GET_DIAG_Sample_DB							
	Nom	Type de données	Déc...	Vale... de...	Rém...	Visible dans...	Commentaire
1	Static						
2	DNN_Struct	DNN	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	SubordinateState	UInt	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	SubordinateIOState	Word	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	DNNmode	Word	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	DIS_Struct	DIS	6.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7	MaintenanceState	DWord	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
8	ComponentStateDetail	DWord	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
9	OwnState	UInt	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
10	IOState	Word	10.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
11	OperatingState	UInt	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

① DNN

② DIS

Remarque

Dans le DB, vous devez saisir manuellement le type de données pour accéder à chacune des trois structures ; il n'y a pas de sélection par liste déroulante. Saisissez les types de données exactement comme indiqué ci-dessous :

- DNN
- DIS

9.7.10 Evénements de diagnostic d'une périphérie décentralisée

Remarque

Dans un réseau PROFIBUS IO, la CPU passe à l'état MARCHE après un chargement ou une mise hors tension puis sous tension à moins que la compatibilité matérielle ne soit configurée pour autoriser des modules de remplacement compatibles (Page 177) et qu'un ou plusieurs modules manquent ou ne soient pas compatibles avec le module configuré.

Comme le montre le tableau suivant, la CPU prend en charge le diagnostic pouvant être configuré pour les composants du système de périphérie décentralisé. Chacune de ces erreurs génère une entrée dans le tampon de diagnostic.

Tableau 9- 184 Gestion des événements de diagnostic pour PROFINET et PROFIBUS

Type d'erreur	Information de diagnostic pour la station ?	Entrée dans le tampon de diagnostic ?	Etat de fonctionnement de la CPU
Erreur de diagnostic	Oui	Oui	reste à l'état Marche
Défaillance du châssis ou de la station	Oui	Oui	reste à l'état Marche
Erreur d'accès aux E/S ¹	Non	Oui	reste à l'état Marche
Erreur d'accès à la périphérie ²	Non	Oui	reste à l'état Marche
Evénement débrogage/enfichage	Oui	Oui	reste à l'état Marche

¹ Exemple d'erreur d'accès aux E/S : un module a été débrogé.

² Exemple d'erreur d'accès à la périphérie : communication acyclique avec un sous-module qui ne transmet pas.


Utilisez l'instruction GET_DIAG (Page 491) pour chaque station afin d'obtenir les informations de diagnostic. Ceci vous permet de traiter les erreurs/défauts constatés sur l'appareil via le programme et de commuter, si vous le souhaitez, la CPU sur Arrêt. Cette méthode requiert la spécification de l'appareil à partir duquel les informations d'état doivent être lues.

L'instruction GET_DIAG utilise l'adresse "L " (LADDR) de la station pour lire le diagnostic de toute la station. Vous trouvez cette adresse L dans la vue de configuration de réseau après avoir sélectionné le châssis entier de la station (zone grise dans sa totalité), l'adresse L est affichée dans l'onglet Propriétés de la station. Vous trouvez le paramètre LADDR pour chaque module individuel soit dans les propriétés du module (dans la configuration de l'appareil), soit dans la table de variables par défaut de la CPU.

9.8 Impulsion

9.8.1 CTRL_PWM (Modulation de largeur d'impulsion)

Tableau 9- 185 Instruction CTRL_PWM (Modulation de largeur d'impulsion)

CONT/LOG	SCL	Description
	<pre>"CTRL_PWM_DB" (PWM:=_uint_in_, ENABLE:=_bool_in_, BUSY=>_bool_out_, STATUS=>_word_out_);</pre>	<p>Fournit une sortie à période fixe avec un rapport cyclique variable. La sortie PWM s'exécute en continu une fois qu'elle a été lancée à la fréquence indiquée (période). On fait varier la largeur d'impulsion de la manière nécessaire pour influencer sur la commande souhaitée.</p>

- 1 Lorsque vous insérez l'instruction, STEP 7 affiche la boîte de dialogue "Options d'appel" pour créer le DB associé.
- 2 Dans l'exemple SCL, "CTRL_PWM_DB" est le nom du DB d'instance.

Tableau 9- 186 Types de données pour les paramètres

Paramètre et type		Type de données	Description
PWM	IN	HW_PWM (Word)	Identificateur PWM : les noms des générateurs d'impulsions activés deviennent des variables dans la table de variables "Constantes" et sont disponibles pour utilisation comme paramètre PWM. (valeur par défaut : 0)
ENABLE	IN	Bool	1=démarrer le générateur d'impulsions 0 = arrêter le générateur d'impulsions
BUSY	OUT	Bool	Fonction occupée (valeur par défaut : 0) 0)
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

L'instruction CTRL_PWM stocke les informations de paramètres dans le DB. L'utilisateur ne peut pas modifier séparément les paramètres du bloc de données qui sont gérés par l'instruction CTRL_PWM.

Indiquez le générateur d'impulsions activé à utiliser en spécifiant son nom de variable comme paramètre PWM.

Lorsque l'entrée EN est à VRAI, l'instruction PWM_CTRL démarre ou arrête le générateur d'impulsions PWM identifié conformément à la valeur de l'entrée ENABLE. La durée de l'impulsion est précisée par la valeur dans l'adresse de mot de sortie Q associée.

Etant donné que la CPU traite la requête une fois que l'instruction CTRL_PWM est exécutée, le paramètre BUSY signalera toujours FAUX. Si une erreur est détectée, ENO est mis à FAUX et le paramètre STATUS contient un code d'erreur.

La largeur d'impulsion est définie à la valeur initiale paramétrée dans la configuration des appareils lorsque la CPU passe à l'état MARCHE. Vous écrivez si nécessaire des valeurs à l'adresse de mot Q indiquée dans la configuration d'appareil ("adresses de sortie"/"adresse de début :") pour modifier la largeur d'impulsion. Vous utilisez une instruction de transfert, de conversion, mathématique ou une boîte PID pour écrire la durée d'impulsion désirée dans le mot Q approprié. Vous devez utiliser la plage valide de la valeur de mot Q (pourcentage, millièmes, dix millièmes ou format analogique S7).

Remarque

Les E/S TOR affectées à PWM et PTO ne peuvent pas être forcées

Les E/S TOR utilisées par la modulation de largeur d'impulsion (PWM) et la sortie de trains d'impulsions (PTO) sont affectées pendant la configuration des appareils. Lorsque des adresses d'E/S TOR sont affectées à ces appareils, les valeurs dans les adresses d'E/S affectées ne peuvent pas être forcées par la fonction de forçage permanent de la table de visualisation.

Tableau 9- 187 Valeurs du paramètre STATUS

STATUS	Description
0	Pas d'erreur
80A1	L'identificateur PWM n'accède pas à un nom de générateur d'impulsions valide.


9.8.2 CTRL_PTO (Émettre un train d'impulsions à fréquence prédéfinie)

L'instruction CTRL_PTO fournit une sortie en onde carrée avec un rapport cyclique de 50 % à une fréquence définie. Vous pouvez utiliser l'instruction CTRL_PTO pour affecter la fréquence sans objet technologique Bloc de données d'axe.

Cette instruction nécessite un générateur d'impulsions. Vous devez activer le générateur d'impulsions et sélectionner un type de signal dans la configuration matérielle. Pour plus d'informations, voir "Configuration d'une voie d'impulsions pour PWM ou PTO" (Page 507).

Vous accédez à l'instruction CTRL_PTO dans les Task Cards, Instructions avancées.

Tableau 9- 188 Instruction CTRL_PTO (Émettre un train d'impulsions à fréquence prédéfinie)

CONT/LOG ¹	SCL ²	Description
	<pre>"CTRL_PTO_DB" (REQ:= _bool_in_, PTO:= _uint_in_, FREQUENCY:= _udint_in_, DONE=> _bool_out_, BUSY=> _bool_out_, ERROR=> _bool_out_, STATUS=> word_out);</pre>	<p>L'instruction CTRL_PTO permet à l'utilisateur de gérer la fréquence pour une sortie en onde carrée (rapport cyclique de 50 %).</p>

¹ Lorsque vous insérez l'instruction, STEP 7 affiche la boîte de dialogue "Options d'appel" pour créer le DB associé.

² Dans l'exemple SCL, "CTRL_PTO_DB" est le nom du DB d'instance.

Tableau 9- 189 Types de données pour les paramètres

Paramètre et type		Type de données	Description
EN	IN	Bool	1 = Instruction activée 0 = Instruction désactivée
REQ	IN	Bool	1 = Régler la fréquence de sortie PTO à la valeur figurant dans l'entrée FREQUENCY 0 = Ne pas modifier PTO
PTO	IN	HW_PTO (Word)	Identificateur PTO : ID matérielle du générateur d'impulsions : <ul style="list-style-type: none"> Les noms des générateurs d'impulsions activés deviennent des variables dans la table de variables "Constantes" et sont disponibles pour utilisation comme paramètre PTO. (valeur par défaut = 0) Vous trouverez l'ID matérielle dans les propriétés du générateur d'impulsions dans la vue des appareils. Les constantes système listent également les ID matérielles des générateurs d'impulsions. (valeur par défaut = 0)
FREQUENCY	IN	UDInt	Fréquence désirée (en Hz) de la sortie PTO. Cette valeur n'est appliquée que si REQ = 1 (la valeur par défaut est 0 Hz).
DONE	OUT	Bool	Fonction achevée sans erreur (valeur par défaut : 0)
BUSY	OUT	Bool	Fonction occupée (valeur par défaut : 0) 0)

9.8 Impulsion

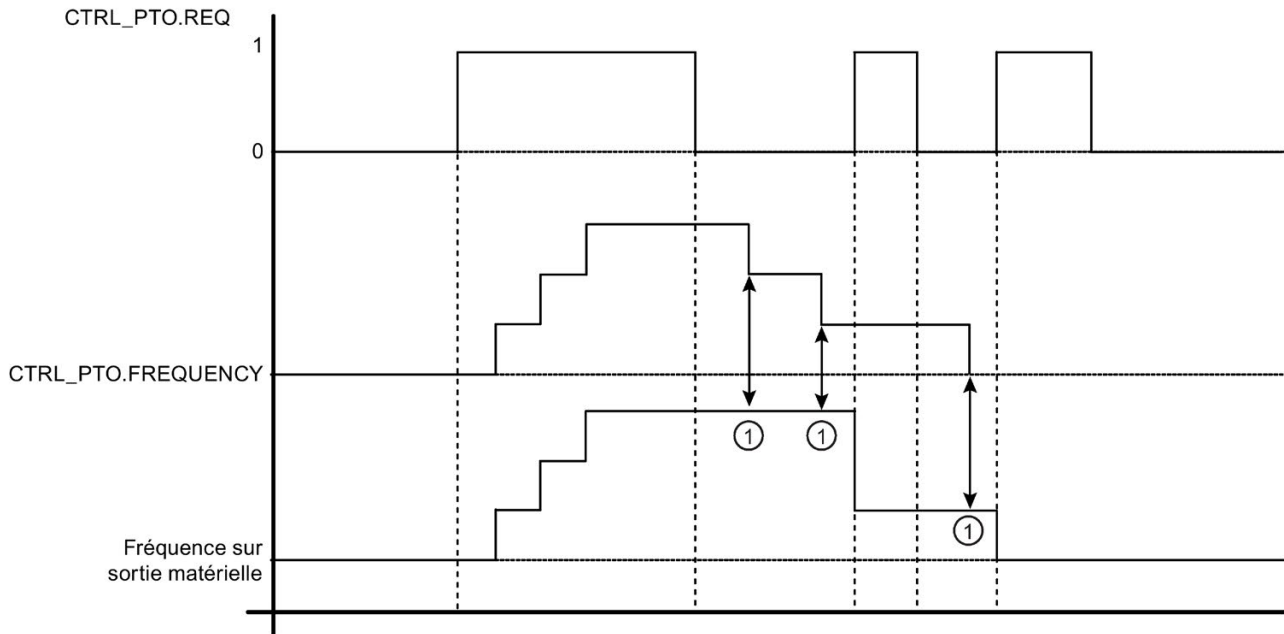
Paramètre et type		Type de données	Description
ERROR	OUT	Word	Erreur détectée (valeur par défaut : 0)
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

L'instruction CTRL_PTO stocke les informations de paramètres dans le DB. L'utilisateur ne peut pas modifier séparément les paramètres du bloc de données qui sont gérés par l'instruction CTRL_PTO.

Indiquez le générateur d'impulsions activé à utiliser en précisant son nom de variable ou son identification matérielle comme paramètre PTO.

Lorsque l'entrée EN est à VRAI, l'instruction CTRL_PTO démarre ou arrête le générateur d'impulsions PTO identifié. Lorsque l'entrée EN est à FAUX, l'instruction CTRL_PTO ne s'exécute pas et le générateur d'impulsions PTO reste dans son état actuel.

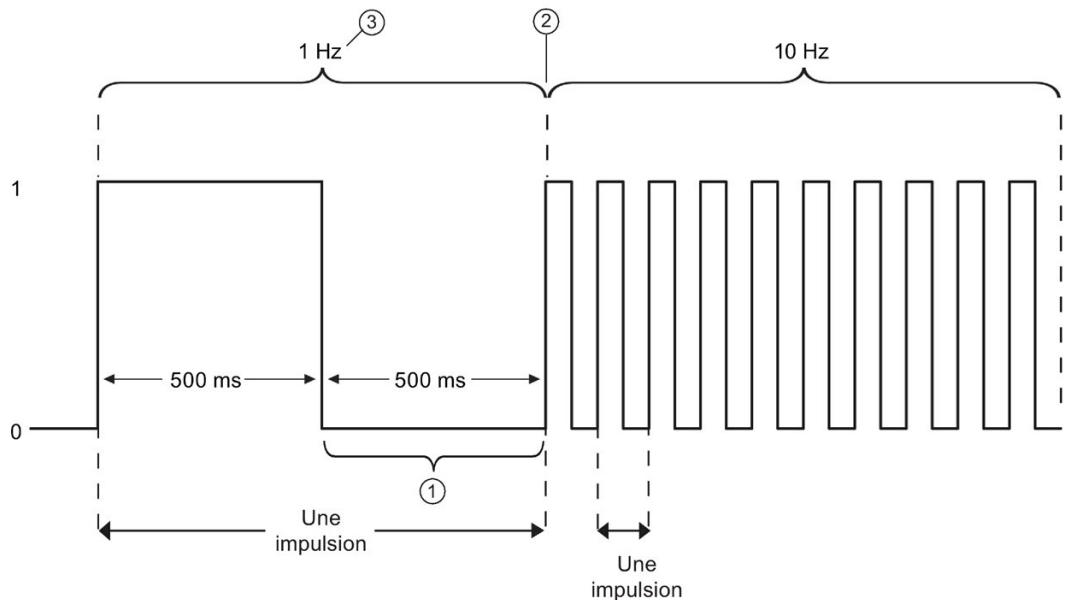
Lorsque l'entrée REQ prend la valeur VRAI, la valeur FREQUENCY est prise en compte. Si REQ est à FAUX, la fréquence de sortie du générateur d'impulsions PTO ne peut pas être modifiée et le générateur d'impulsions continue à émettre des impulsions.



① Pas de changement dans la fréquence de sortie tant que REQ = 0

Comme l'instruction CTRL_PTO ne fait que démarrer le générateur d'impulsions PTO, elle s'achève immédiatement. En conséquence, la sortie BUSY n'est jamais à 1. La sortie DONE est activée aussi longtemps qu'aucune erreur ne se produit. Si une erreur est détectée, le paramètre ERROR est mis à VRAI et le paramètre STATUS contient un code d'erreur.

Lorsque l'utilisateur active l'instruction CTRL_PTO avec une fréquence donnée, le S7-1200 émet un train d'impulsions à cette fréquence. L'utilisateur peut modifier la fréquence désirée à tout moment. En cas de modification de la fréquence, le S7-1200 achève l'impulsion en cours avant de changer la fréquence à la nouvelle fréquence désirée. Ainsi, par exemple, si la fréquence désirée est 1 Hz (ce qui prend 1 000 ms) et que l'utilisateur change la fréquence à 10 Hz après 500 ms, la fréquence est modifiée à la fin de la période des 1 000 ms.



- ① L'utilisateur change la fréquence à 10 Hz après 500 ms.
- ② L'impulsion de 1 Hz doit s'achever avant que la fréquence ne puisse être modifiée à la nouvelle fréquence de 10 Hz.
- ③ 1 Hz correspond à 1 000 ms.

L'objet matériel Générateur d'impulsions présente la restriction suivante : une seule instruction peut utiliser le générateur d'impulsions comme sortie de train d'impulsions et l'éditeur de la configuration matérielle gère l'utilisation du générateur d'impulsions. Les autres instructions qui tentent d'accéder à cette sortie PTO renvoient une erreur "0x8090" (le générateur d'impulsions avec l'ID matérielle indiquée est en cours d'utilisation).

Remarque

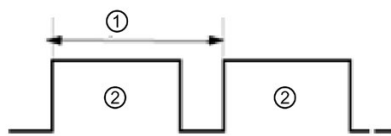
Les E/S TOR affectées à PWM et PTO ne peuvent pas être forcées

Les E/S TOR utilisées par la modulation de largeur d'impulsion (PWM) et la sortie de train d'impulsions (PTO) sont affectées pendant la configuration des appareils. Lorsque des adresses d'E/S TOR sont affectées à ces appareils, les valeurs dans les adresses d'E/S affectées ne peuvent pas être forcées par la fonction de forçage permanent de la table de visualisation.

Tableau 9- 190 Valeurs des codes d'erreur dans le paramètre STATUS

Code d'erreur (W#16#...)	Description
0	Pas d'erreur
0x8090	Le générateur d'impulsions avec l'ID matérielle indiquée est en cours d'utilisation.
0x8091	Fréquence hors plage. La fréquence désirée dépasse la fréquence maximale de la sortie d'impulsions sélectionnée.
0x80A1	L'identification PTO (ID matérielle) ne désigne pas une sortie PTO valide.
0x80D0	Le générateur d'impulsions avec l'ID matérielle indiquée n'est pas activé. Activez le générateur d'impulsions sous "Générateurs d'impulsions (PTO/PWM)" dans les propriétés de la CPU.
0x80D1	Le générateur d'impulsions avec l'ID matérielle indiquée n'a pas de sélection PTO. Sélectionnez PTO dans la configuration matérielle.

9.8.3 Fonctionnement des sorties d'impulsions



- ① Temps de cycle
- ② Largeur d'impulsion

La largeur d'impulsion peut être exprimée sous forme de centièmes de la période (0 à 100), de millièmes (0 à 1000), de dix millièmes (0 à 10000) ou de format analogique S7.

La largeur d'impulsion peut varier de 0 (pas d'impulsion, toujours désactivée) à la pleine échelle (pas d'impulsion, toujours activée).

Comme la sortie PWM peut varier de 0 à la pleine échelle, elle fournit une sortie TOR qui est, en de nombreuses façons, identique à une sortie analogique. Vous pouvez, par exemple, utiliser cette sortie PWM pour commander la vitesse d'un moteur de l'arrêt à la pleine vitesse ou pour commander la position d'une soupape de "fermée" à "complètement ouverte".

Vous configurez la fréquence dans la configuration matérielle (Page 507). Vous gérez la largeur d'impulsion dans le programme utilisateur.

Quatre générateurs d'impulsions sont disponibles pour commander les fonctions de sortie d'impulsions rapides : PWM et PTO (Sortie de trains d'impulsions). PTO est utilisé par les instructions Motion Control. Vous pouvez affecter chaque générateur d'impulsions à PWM ou à PTO, mais pas aux deux en même temps.

Vous pouvez utiliser les sorties CPU intégrées ou les sorties du Signal Board optionnel. Les numéros des sorties sont indiqués dans le tableau suivant (sur la base de la configuration par défaut des sorties). Si vous avez modifié la numérotation des sorties, les numéros des sorties seront ceux que vous avez définis. Notez que PWM ne nécessite qu'une sortie alors que PTO peut optionnellement utiliser deux sorties par voie. Si une sortie n'est pas utilisée pour une fonction d'impulsion, elle est disponible pour d'autres usages. Voir le tableau ci-dessous pour l'affectation des E/S.

Le tableau ci-dessous montre les affectations des E/S par défaut ; toutefois, les quatre générateurs d'impulsions peuvent être affectés à toute sortie TOR intégrée à la CPU ou du SB. Lors de l'affectation des sorties aux PTO/PWM, tenir compte du fait que les sorties prennent en charge des tensions et des vitesses différentes.

Remarque

Les sorties de trains d'impulsions ne peuvent pas être utilisées par d'autres instructions dans le programme utilisateur.

En cas de configuration des sorties de la CPU ou du Signal Board comme générateurs d'impulsions (pour les instructions PWM, PTO ou Motion Control), les adresses des sorties correspondantes sont supprimées de la mémoire Q et ne peuvent pas être utilisées à d'autres fins dans le programme utilisateur. Si votre programme utilisateur écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.

Remarque

Les sorties indiquant le sens des PTO peuvent être libérées pour être utilisées ailleurs dans votre programme.

Chaque PTO requiert l'affectation de deux sorties : l'une comme sortie d'impulsions et l'autre comme sortie de sens. Vous pouvez utiliser juste la sortie d'impulsions, sans la sortie de sens. Ainsi, vous pouvez libérer la sortie de sens et l'utiliser à d'autres fins dans votre programme utilisateur.

Tableau 9- 191 Affectations par défaut des sorties aux générateurs d'impulsions³

Description	Impulsion	Sens
PTO1		
E/S intégrées	Q0.0	Q0.1
E/S du SB	Q4.0	Q4.1
PWM1		
Sorties intégrées	Q0.0	-
Sorties du SB	Q4.0	-
PTO2		
E/S intégrées	Q0.2	Q0.3
E/S du SB	Q4.2	Q4.3
PWM2		
Sorties intégrées	Q0.2	-
Sorties du SB	Q4.2	-
PTO3		
E/S intégrées	Q0.4 ¹	Q0.5 ¹
E/S du SB	Q4.0	Q4.1
PWM3		
Sorties intégrées	Q0.4 ¹	-
Sorties du SB	Q4.1	-
PTO4		
E/S intégrées	Q0.6 ²	Q0.7 ²
E/S du SB	Q4.2	Q4.3
PWM4		
Sorties intégrées	Q0.6 ²	-
Sorties du SB	Q4.3	-

¹ La CPU 1211C ne comporte pas de sorties Q0.4, Q0.5, Q0.6 et Q0.7. Ces sorties ne peuvent donc pas être utilisées dans la CPU 1211C.

² La CPU 1212C ne comporte pas de sorties Q0.6 et Q0.7. Ces sorties ne peuvent donc pas être utilisées dans la CPU 1212C.

³ Ce tableau s'applique aux fonctions PTO/PWM des CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C et CPU 1217C.

9.8.4 Configuration d'une voie d'impulsions pour PWM ou PTO

Pour préparer le fonctionnement PWM ou PTO, configurez d'abord une voie d'impulsions dans la configuration d'appareil en sélectionnant la CPU puis Générateur d'impulsions (PTO/PWM) et en choisissant PWM1/PTO1 à PWM4/PTO4. Activez le générateur d'impulsions (case à cocher). Si un générateur d'impulsions est activé, un nom par défaut unique est affecté à ce générateur d'impulsions particulier. Vous pouvez modifier ce nom en l'éditant dans le champ d'édition "Nom", mais il doit rester unique. Les noms des générateurs d'impulsions activés deviennent des variables dans la table de variables "Constantes" et sont disponibles pour être utilisés comme :

- paramètre PWM de l'instruction CTRL_PWM
- paramètre PTO de l'instruction CTRL_PTO

Vous pouvez également écrire un commentaire sur ce générateur d'impulsions spécifique dans la boîte d'édition "Commentaire".

Tableau 9- 192 Sortie de la CPU : fréquence maximale (PTO) et période minimale (PWM)

CPU	Voie de sortie CPU	Fréquence PTO maximale	Période PWM minimale
1211C	Qa.0 à Qa.3	100 kHz	10 µs
1212C	Qa.0 à Qa.3	100 kHz	10 µs
	Qa.4, Qa.5	20 kHz	50 µs
1214C et 1215C	Qa.0 à Qa.3	100 kHz	10 µs
	Qa.4 à Qb.1	20 kHz	50 µs
1217C	DQa.0 à DQa.3 (.0+, .0- à .3+, .3-)	1 MHz	1 µs
	DQa.4 à DQb.1	100 kHz	10 µs

Tableau 9- 193 Sortie de Signal Board (SB) : fréquence maximale (PTO) et période minimale (PWM)

Signal Board (SB)	Voie de sortie SB	Fréquence PTO maximale	Période PWM minimale
SB 1222, 200 kHz	DQe.0 à DQe.3	200 kHz	5 µs
SB 1223, 200 kHz	DQe.0, DQe.1	200 kHz	5 µs
SB 1223	DQe.0, DQe.1	20 kHz	50 µs

Remarque

La période minimale de chacune des sorties de la CPU et du Signal Board est indiquée dans les tableaux ci-dessus. Toutefois, TIA Portal ne vous avertit pas si vous configurez un générateur d'impulsions PWM avec une période inférieure à la période minimale du matériel. Afin d'éviter tout problème dans votre application, veillez toujours à ce que la période se situe dans les limites du matériel.

Remarque

Lorsque vous définissez la durée d'impulsion d'un signal PWM, le temps d'impulsion réel (temps pendant lequel l'impulsion est haute) doit être supérieur ou égal à 1 milliseconde si la base de temps est en millisecondes. Si la base de temps est en microsecondes, le temps d'impulsion réel doit être supérieur ou égal à 1 microseconde. La sortie est désactivée si le temps d'impulsion est inférieur à 1 "base de temps".

Par exemple, une période de 10 millisecondes et une durée d'impulsion de 5 centièmes donnent un temps d'impulsion de 0,5 microseconde. Cette valeur étant inférieure à 1 microseconde, le signal PWM est désactivé.

Paramétrage

La section d'affectation des paramètres permet à l'utilisateur de configurer les paramètres de l'impulsion de sortie. Les options suivantes sont disponibles selon qu'on a sélectionné PWM ou PTO :

- Type de signal : Configurez la sortie d'impulsions en tant que PWM ou PTO. Pour plus d'informations sur les choix PTO, référez-vous à "Phasage" (Page 680) :
 - PWM
 - PTO (impulsion A et sens B)
 - PTO (incrémentatation impulsion A et décrémentation impulsion B)
 - PTO (déphasage A/B)
 - PTO (déphasage A/B - quadruple)
- Base de temps (s'applique uniquement à PWM) : Sélectionnez les unités de temps à utiliser :
 - Millisecondes
 - Microsecondes
- Format de la durée d'impulsion (s'applique uniquement à PWM) : Définissez la résolution de la durée d'impulsion (largeur) :
 - Centièmes (0 à 100)
 - Millièmes (0 à 1000)
 - Dix millièmes (0 à 10000)
 - Format analogique S7 (0 à 27648)
- Période (s'applique uniquement à PWM) : Affectez la durée nécessaire pour achever une impulsion (temps impulsion haute + temps impulsion basse = période). Vous pouvez modifier le temps de cycle lors de l'exécution en sélectionnant la case à cocher "Permettre la modification du temps de cycle lors de l'exécution". Pour plus d'informations, voir la section "Adresses E/S" ci-dessous. La plage va de 1 à 16 777 215 unités de temps.

- Durée d'impulsion initiale (s'applique uniquement à PWM) : Affectez la durée de la première impulsion. Vous pouvez modifier cette valeur lors de l'exécution à l'aide de l'adresse de mot Q configurée dans les adresses d'E/S. La plage se base sur le format de la durée d'impulsion.
- Permettre la modification du temps de cycle lors de l'exécution (s'applique uniquement à PWM) : La sélection de cette option permet à votre programme de modifier le temps de cycle du signal PWM alors que le programme est en cours d'exécution. Pour plus d'informations, voir la section "Adresses E/S" ci-dessous.

Remarque

Lorsque vous définissez la durée d'impulsion d'un signal PWM, tenez compte du retard de commutation de la voie de sortie qui est indiquée à l'annexe A. En effet, la durée d'impulsion effective mesurée au niveau de la sortie peut s'avérer supérieure à la durée d'impulsion sélectionnée. L'augmentation de la durée d'impulsion est plus marquée pour les durées d'impulsion courtes et les fréquences élevées. Pensez à vérifier que la durée d'impulsion mesurée à la sortie est conforme à vos exigences.

Détermination de la valeur de durée d'impulsion

On obtient la durée d'impulsion en multipliant la durée d'impulsion initiale par la période. Lorsque vous sélectionnez la base de temps, le format de durée d'impulsion, la période et la durée d'impulsion initiale, gardez à l'esprit que la durée d'impulsion globale ne peut pas être un nombre fractionnaire. Si la durée d'impulsion résultante est un nombre fractionnaire, vous devez ajuster la durée d'impulsion initiale ou modifier la base de temps afin de générer une valeur entière.

Voici deux exemples :

- Exemple 1 : si vous sélectionnez les valeurs suivantes :
 - Base de temps = millisecondes (ms)
 - Format de la durée d'impulsion = centièmes (0 à 100)
 - Période = 3 ms
 - Durée d'impulsion initiale = 75

La durée d'impulsion résultante est égale à : $0,75 \times 3 \text{ ms} = 2,25 \text{ ms}$

Cette valeur de durée d'impulsion est fractionnaire et provoque une erreur lorsque vous utilisez l'instruction CTRL_PWM. La durée d'impulsion doit être modifiée en une valeur entière.

- Exemple 2 : si vous sélectionnez les valeurs suivantes :
 - Base de temps = microsecondes (μs)
 - Format de la durée d'impulsion = centièmes (0 à 100)
 - Période = 3000 μs
 - Durée d'impulsion initiale = 75

La durée d'impulsion résultante est égale à : $0,75 \times 3000 \mu\text{s} = 2250 \mu\text{s}$

Cette valeur de durée d'impulsion est entière et permet un fonctionnement correct de l'instruction CTRL_PWM.

Sorties matérielles

Dans la section des sorties matérielles, sélectionnez la voie de sortie dans le menu déroulant. Selon la configuration, il peut y avoir une ou deux sorties à sélectionner. Si vous affectez une voie de sortie à un générateur d'impulsions, cette voie de sortie ne peut pas être utilisée par un autre générateur d'impulsions, compteur rapide HSC ou mémoire image du processus.

Remarque

Les sorties de générateur d'impulsions ne peuvent pas être utilisées par d'autres instructions dans le programme utilisateur

En cas de configuration des sorties de la CPU ou du Signal Board comme générateurs d'impulsions (pour les instructions PWM, PTO ou Motion Control), les adresses des sorties correspondantes sont supprimées de la mémoire Q et ne peuvent pas être utilisées à d'autres fins dans votre programme. Si votre programme écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.

Adresses E/S

Le PWM possède deux octets de mémoire Q conçue pour la "Durée d'impulsion". Tandis que PWM est en cours d'exécution, vous pouvez modifier la valeur affectée dans la mémoire Q et modifier la Durée d'impulsion.

Dans la section des adresses d'E/S, saisissez l'adresse de mot Q où vous voulez stocker la valeur de durée d'impulsion.

Les adresses par défaut pour les valeurs de durée d'impulsion PWM sont les suivantes :

- PWM1 : QW1000
- PWM2 : QW1002
- PWM3 : QW1004
- PWM4 : QW1006

Pour PWM, la valeur à cette adresse gère la durée de l'impulsion et est initialisée à la valeur "durée d'impulsion initiale" (définie ci-avant) à chaque fois que la CPU passe de l'état ARRÊT à l'état MARCHE. Vous modifiez cette valeur de mot Q pendant l'exécution pour changer la durée d'impulsion. La plage de cette valeur dépend du format de durée d'impulsion configuré dans l'affectation des paramètres.

Vous pouvez également assigner quatre octets supplémentaires de mémoire Q pour le "Temps de cycle" du signal PWM. Référez-vous à "Utilisation des sorties d'impulsion" (Page 504) pour un diagramme du signal PWM. Lorsque vous sélectionnez la case à cocher "Permettre la modification du temps de cycle pendant l'exécution", les deux premiers octets contiennent la valeur de durée d'une impulsion et les quatre derniers octets détiennent la valeur de temps de cycle.

Tandis que PWM est en cours d'exécution, vous pouvez modifier la valeur du mot double à la fin de la mémoire Q qui est assignée à ce PWM. Ceci modifie le temps de cycle du signal PWM. Par exemple, vous activez cette option de sorte que la CPU assigne six octets pour PWM1 et vous décidez d'utiliser QB1008 à QB1013. Une fois que vous avez téléchargé le programme et démarré le PWM, vous pouvez modifier la durée d'impulsion en utilisant QW1008 et le temps de cycle en utilisant QD1010.

Chaque fois que la CPU effectue une transition entre le mode ARRÊT et le mode MARCHE, la CPU initialise la valeur de temps de cycle dans la mémoire Q à la valeur de "Temps de cycle" assignée ci-dessus dans la section "Paramétrage". Les unités et la plage des valeurs pour la valeur de temps de cycle dans la mémoire Q sont les mêmes que la configuration dans la section "Paramétrage".

Lorsque vous sélectionnez la case à cocher "Permettre la modification du temps de cycle pendant l'exécution", TIA Portal sélectionne automatiquement une nouvelle adresse pour l'adresse de sortie. La nouvelle adresse de sortie ne peut pas être la même que l'adresse par défaut pour ce générateur d'impulsions. Le TIA Portal utilise le prochain bloc disponible constitué de six octets consécutifs. Si la recherche ne permet pas de trouver un bloc disponible dans la mémoire Q avant d'en atteindre la fin, la recherche recommence à l'adresse "0" de la mémoire Q et continue la recherche d'un bloc disponible.

Un générateur d'impulsions configuré pour PTO n'utilise pas l'adresse de mot Q.

9.9 Recettes et journaux

9.9.1 Recettes

9.9.1.1 Liste des recettes

Stockage des données de recette

- Tout bloc de données de recette que vous créez dans votre projet doit être enregistré dans la mémoire de **chargement** de la CPU. Vous pouvez utiliser la mémoire CPU interne ou une carte "programme" externe.
- Un autre DB que vous devez créer est le bloc de données de recette actif. Ce DB doit se trouver en mémoire de **travail** où est effectuée la lecture ou l'écriture d'un enregistrement de recette actif par la logique de votre programme.

Gestion des données de recette

Le DB de recette utilise un tableau avec des enregistrements de recette de produits. Chaque élément du tableau de recette représente une saveur de recette différente basée sur un ensemble d'éléments communs.

- Vous créez un type de données API ou Struct qui définit tous les éléments dans un enregistrement de recette. Ce modèle de type de données est réutilisé pour tous les enregistrements de recette. Les recettes de produit varient selon les valeurs initiales affectées aux ingrédients de la recette.
- L'une des recettes peut être transférée à tout moment du DB de recette (toutes les recettes dans la mémoire de chargement) dans le DB de recette actif (une recette dans la mémoire de travail) à l'aide de l'instruction READ_DBL. Une fois un enregistrement de recette transféré dans la mémoire de travail, la logique de votre programme peut lire les valeurs des éléments et commencer un lot de fabrication. Le transfert réduit la taille de mémoire de travail de la CPU nécessaire pour les données de recette.
- Si les valeurs des éléments de recette actifs sont modifiées par un appareil IHM pendant un lot, vous pouvez réécrire les valeurs modifiées dans le DB de recette à l'aide de l'instruction WRIT_DBL.

Exporter la recette (du DB de recette dans un fichier CSV)

L'instruction RecipeExport permet de générer un fichier CSV à partir de l'ensemble des enregistrements de recette. Les enregistrements de recette non utilisés sont également exportés.

Importer la recette (d'un fichier CSV dans le DB de recette)

Une fois l'opération d'exportation de recette achevée, vous pouvez utiliser le fichier CSV généré comme modèle de structure de données.

1. Utilisez la page File Browser du serveur Web de la CPU pour charger un fichier CSV de recette existant de la CPU dans un PC.
2. Modifiez le fichier CSV à l'aide d'un éditeur de texte ASCII. Vous pouvez modifier les valeurs initiales attribuées aux éléments, mais pas les types de données ou leur structure.
3. Rechargez le fichier CSV modifié du PC dans la CPU. Cependant, l'ancien fichier CSV (du même nom) qui se trouve dans la mémoire de chargement de la CPU doit être supprimé ou renommé avant que le serveur Web de la CPU autorise le chargement.
4. Après avoir chargé le fichier CSV modifié dans la CPU, vous pouvez utiliser l'instruction RecipelImport pour transférer les nouvelles valeurs initiales du fichier CSV modifié (en mémoire de chargement de la CPU) dans le DB de recette (en mémoire de chargement de la CPU).

9.9.1.2 Exemple de recette

Exemples de recette

Le tableau suivant montre comment préparer des données de la recette pour les utiliser dans un DB de recette. L'exemple de DB de recette est constitué de cinq enregistrements dont trois sont utilisés. Le quatrième et le cinquième enregistrements sont laissés libres pour des extensions ultérieures. Chaque colonne du tableau représente un enregistrement composé du nom de la recette, des types de données et des valeurs des ingrédients.

productname	water	barley	wheat	hops	yeast	waterTmp	mashTmp	mashTime	QTest
Pils	10	9	3	280	39	40	30	100	0
Lager	10	9	3	150	33	50	30	120	0
BlackBeer	10	9	3	410	47	60	30	90	1
Not_used	0	0	0	0	0	0	0	0	0
Not_used	0	0	0	0	0	0	0	0	0

Créer un bloc de données de recette

Remarque

Règles pour les blocs de données de recette

- Le DB de recette doit contenir un tableau à une dimension de type de données API ou Struct. L'exemple de recette montre comment créer un bloc de données de recette de type API.
- Dans cet exemple, tous les ingrédients des éléments sont de type UInt. Les types de données des éléments peuvent également être un mélange de type de données, excepté STRUCT. Dans un élément de tableau du DB de recette, un STRUCT dans des données de type API ou un STRUCT dans un STRUCT n'est pas autorisé.

D'abord, créez un nouveau type de données API

Ajoutez un nouveau type de données API dont le nom est le type de recette. Dans l'image suivante, "Beer_Recipe" est le nouveau type de données API complexe qui contient une séquence de types de données simples. Le type de données API "Beer_Recipe" est un modèle de données réutilisé dans chaque enregistrement de DB de recette ainsi que dans le DB de recette actif. Entrez les noms et types de données des éléments communs à tous les exemples de recette. Les valeurs des différents éléments sont ajoutées ultérieurement dans le bloc de données de recette.

Beer_Recipe			
	Name	Data type	Default value
1	productname	String[20]	'Beer_Recipe'
2	water	UInt	0
3	barley	UInt	0
4	wheat	UInt	0
5	hops	UInt	0
6	yeast	UInt	0
7	waterTmp	UInt	0
8	mashTmp	UInt	0
9	mashTime	UInt	0
10	QTest	UInt	0

Ensuite, créez un bloc de données de recette

- Créez votre DB de recette comme un bloc de données global avec la propriété "Sauvegarder uniquement dans la mémoire de chargement" activée.
- Le nom d'un bloc de données de recette est utilisé comme nom de fichier pour le fichier CSV correspondant. Vous devez respecter les restrictions concernant les noms de DB dans le système de fichiers Windows. Les caractères \ / : * ? " < > | et l'espace ne sont pas autorisés.
- L'affectation du tableau de recette est "Products" comme Array [1.. 5] of "Beer_Recipe". Un tableau de 5 colonnes est le nombre maximum de saveurs de recette possibles.
- Les valeurs des éléments de la recette sont ajoutées comme valeurs initiales de DB.

Dans l'image suivante, le détail de la recette "BlackBeer" est affiché pour montrer tous les éléments d'un enregistrement de recette.

Recipe_DB				
	Name	Data type	Offset	Start value
1	▼ Static			
2	▣ ▾ Products	Array [1 .. 5] of "Beer_Recipe"	...	
3	▣ ▸ Products[1]	"Beer_Recipe"	...	
4	▣ ▸ Products[2]	"Beer_Recipe"	...	
5	▣ ▾ Products[3]	"Beer_Recipe"	...	
6	▣ ▸ productname	String[20]	...	'BlackBeer'
7	▣ ▸ water	UInt	...	10
8	▣ ▸ barley	UInt	...	9
9	▣ ▸ wheat	UInt	...	3
10	▣ ▸ hops	UInt	...	410
11	▣ ▸ yeast	UInt	...	47
12	▣ ▸ waterTmp	UInt	...	60
13	▣ ▸ mashTmp	UInt	...	30
14	▣ ▸ mashTime	UInt	...	90
15	▣ ▸ QTest	UInt	...	1
16	▣ ▸ Products[4]	"Beer_Recipe"	...	
17	▣ ▸ Products[5]	"Beer_Recipe"	...	

Exporter la recette (du DB de recette dans un fichier CSV)

"RecipeExport (Page 517)" reporte les données de recettes du DB dans un fichier texte CSV avec la structure suivante.

```
Recipe_DB.csv
index,productname,water,barley,wheat,hops,yeast,waterTmp,
mashTmp,mashTime,QTest
1,"Pils",10,9,3,280,39,40,30,100,0
2,"Lager",10,9,3,150,33,50,30,120,0
3,"BlackBeer",10,9,3,410,47,60,30,90,1
4 "Not_used",0,0,0,0,0,0,0,0,0,0
5 "Not_used",0,0,0,0,0,0,0,0,0,0
```

Importer la recette (depuis un fichier CSV vers le DB de recette)

1. Utilisez la page Navigateur de fichiers (Page 1075) du serveur Web pour charger un fichier CSV de recettes existant de la mémoire de chargement de la CPU dans un PC.
2. Modifiez le fichier CSV à l'aide d'un éditeur de texte ASCII. Vous pouvez modifier les valeurs initiales attribuées aux éléments, mais pas les types de données ou leur structure.
3. Rechargez le fichier CSV modifié du PC dans la CPU. Vous devez toutefois supprimer ou renommer l'ancien fichier CSV (de même nom) dans la mémoire de chargement de la CPU avant que le serveur Web n'autorise le chargement.
4. Après avoir chargé le fichier CSV modifié dans la CPU, vous pouvez utiliser l'instruction RecipeImport pour transférer les nouvelles valeurs initiales du fichier CSV modifié (en mémoire de chargement de la CPU) dans le DB de recettes (en mémoire de chargement de la CPU).

Les fichiers CSV et la structure de DB de recette doivent exactement se correspondre

- Si les valeurs du fichier CSV peuvent être modifiées, modifier la structure n'est pas autorisé. L'instruction RecipelImport exige que le nombre exact d'enregistrements et d'éléments soit le même dans la structure du bloc de données de recette cible. Sinon, l'exécution de RecipelImport échoue. Par exemple, si 10 recettes sont définies dans le DB de recette, mais 6 seulement sont utilisées, les lignes 7 à 10 du fichier CSV sont également transférées dans le DB. Vous devez indiquer si ces données sont valides ou pas. Vous pouvez, par exemple, affecter une variable "Not_used" au nom de produit dans des enregistrements de recette inutilisés.
- Si vous ajoutez des enregistrements au fichier texte et que vous importez le fichier modifié, assurez-vous que la limite d'éléments dans le tableau du DB de recette définie est suffisamment grande pour tous les enregistrements de recette.
- Un numéro d'indice est généré automatiquement lors de l'exportation dans le fichier CSV. Si vous créez des enregistrements supplémentaires, ajoutez des numéros d'indice consécutifs en conséquence.
- L'exécution de RecipelImport vérifie que la structure des données du fichier CSV est correcte et si les valeurs correspondent aux types de données affectés dans le DB de recette associé. Par exemple, un type de données Bool ne peut pas contenir une valeur entière ; sinon, l'exécution de RecipelImport échoue.

Représentation de données de recette CSV dans Excel

Vous pouvez ouvrir le fichier CSV dans Excel pour le lire et l'éditer facilement. Si les virgules ne sont pas reconnues comme séparateurs décimaux, utilisez la fonction d'importation d'Excel pour afficher les données sous forme structurée.

	A	B	C	D	E	F	G	H	I	J	K
1	index	product	water	barley	wheat	hops	yeast	waterTmp	mashTmp	mashTime	QTest
2	1	"Pils"	10	9	3	280	39	40	30	100	0
3	2	"Lager"	10	9	3	150	33	50	30	120	0
4	3	"BlackBeer"	10	9	3	410	47	60	30	90	1
5	4	"Not_used"	0	0	0	0	0	0	0	0	0
6	5	"Not_used"	0	0	0	0	0	0	0	0	0

Remarque

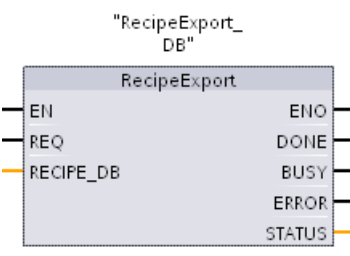
Virgules dans le champ de nom de l'élément de type de données API

Ne mettez pas de virgules dans le champ de nom de l'élément ou des éléments de type de données API utilisés dans une recette. Dans ce cas en effet, Excel insérerait des colonnes supplémentaires dans le fichier .csv affiché. Ces colonnes supplémentaires pourraient introduire des erreurs lorsque vous modifiez les valeurs initiales du fichier d'enregistrement de recette.

9.9.1.3 Instructions de transfert des données de recette

RecipeExport (Exporter la recette)

Tableau 9- 194 Instruction RecipeExport

CONT/LOG	SCL	Description
	<pre>"RecipeExport_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, Reci- pe_DB:=_variant_inout_);</pre>	<p>L'instruction "RecipeExport" exporte tous les enregistrements de la recette depuis un bloc de données dans un fichier CSV. Le fichier CSV contient les noms des produits, des éléments et les valeurs initiales. Le fichier CSV est stocké dans la mémoire de chargement interne ou dans la mémoire de chargement externe, si une carte "programme" externe est installée en option.</p> <p>L'exportation est lancée par le paramètre "REQ". Le paramètre BUSY est mis à "1" lors de l'exportation. Une fois l'exécution de RecipeExport achevée, BUSY est remis à "0" et l'achèvement de l'exécution s'affiche avec "1" dans le paramètre DONE. Si une erreur s'est produite lors de l'exécution, celle-ci est signalée via les paramètres ERROR et STATUS.</p>

Un DB de recette doit être créé avant toute possibilité d'exportation de recette. Le nom d'un bloc de données de recette est utilisé comme nom de fichier pour le nouveau fichier CSV. Si un fichier CSV avec le même nom existe déjà, celui-ci est écrasé lors de l'exportation.

Vous pouvez utiliser la page File Browser (Page 1075) du serveur Web intégré de la CPU pour accéder au fichier CSV avec la recette. Le fichier est créé dans le dossier "Recipes" dans le répertoire racine de la mémoire de chargement de la CPU.

Tableau 9- 195 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Paramètre de commande REQUEST : son front positif active l'exportation.
RECIPE_DB	In/Out	Variant	Pointeur sur le bloc de données de recette. Voir "Exemple de DB de recette (Page 513)" pour plus de détails. Vous devez respecter les restrictions concernant les noms de DB dans le système de fichiers Windows. Les caractères \ / : * ? " < > et l'espace ne sont pas autorisés.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur. (valeur par défaut : faux)
BUSY	OUT	Bool	Exécution de RecipeExport <ul style="list-style-type: none"> • 0 : Pas d'opération en cours • 1 : Opération en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE. <ul style="list-style-type: none"> • 0 : Ni avertissement ni erreur • 1 : Une erreur s'est produite. STATUS fournit des informations sur le type d'erreur.
STATUS	OUT	Word	Code d'erreur d'exécution

Tableau 9- 196 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#....)	Description
0	0000	Pas d'erreur
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	Nième appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance est utilisée.
1	8090	Le nom du fichier contient des caractères non valides
1	8091	La structure de données spécifiée à RECIPE_DB ne peut pas être traitée.
1	8092	La structure de données définie dans RECIPE_DB dépasse 5 000 octets
1	80B3	Espace mémoire insuffisante sur la cartouche mémoire (MC) ou dans la mémoire de chargement interne
1	80B4	La cartouche mémoire est protégée en écriture.
1	80B6	L'attribut "Sauvegarder uniquement dans la mémoire de chargement" du DB de recette n'est pas activé.
1	80C0	Fichier CSV momentanément verrouillé
1	80C1	DB momentanément verrouillé

RecipeImport (Importer la recette)

Tableau 9- 197 Instruction RecipeImport

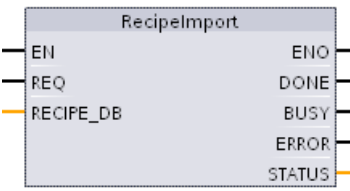
CONT/LOG	SCL	Description
<p>"RecipeImport_DB"</p> 	<pre>"RecipeImport_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, Reci- pe_DB:=_variant_inout_);</pre>	<p>L'instruction "RecipeImport" importe les données de la recette depuis un fichier CSV sur la mémoire de chargement de la CPU dans un bloc de données spécifié au paramètre RECIPE_DB. Les valeurs initiales dans le bloc de données de recette sont écrasées lors de l'importation. L'importation est lancée par le paramètre "REQ". Le paramètre BUSY est mis à "1" lors de l'importation. Une fois l'exécution de RecipeImport achevée, BUSY est remis à "0" et l'achèvement de l'exécution s'affiche avec "1" dans le paramètre DONE. Si une erreur s'est produite lors de l'exécution, celle-ci est signalée via les paramètres ERROR et STATUS.</p>

Tableau 9- 198 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Paramètre de commande REQUEST : son front positif active l'importation.
RECIPE_DB	In/Out	Variant	Pointeur sur le bloc de données de recette. Voir "Exemple de DB de recette (Page 513)" pour plus de détails. Vous devez respecter les restrictions concernant les noms de DB dans le système de fichiers Windows. Les caractères \ / : * ? " < > et l'espace ne sont pas autorisés.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur. (valeur par défaut : faux)
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 : Pas d'opération en cours 1 : Opération en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Avant toute possibilité d'importation de recette, il doit exister un bloc de données de recette qui contient une structure cohérente avec la structure de données du fichier CSV.

Règles pour fichier CSV :

- Le fichier CSV doit se trouver dans le répertoire racine "Recipes" de la mémoire de chargement interne ou de la mémoire de chargement externe, si une carte "programme" externe est installée en option.
- Le nom du fichier CSV doit être le même que celui du bloc de données dans le paramètre RECIPE_DB.
- La première ligne (ligne d'en-tête) du fichier CSV contient les noms des éléments de la recette. La première ligne est ignorée lors de l'importation. Un mappage des noms des éléments de la recette dans le fichier CSV et dans le bloc de données n'est pas effectué lors de l'importation.
- Dans chaque cas, la première valeur de chaque ligne du fichier CSV constitue le numéro d'index de la recette. Les différentes recettes sont importées dans l'ordre de l'index. L'index dans le fichier CSV doit pour cela être croissant et ininterrompu (si tel n'est pas le cas, le code d'erreur 80B0 est émis dans le paramètre STATUS.)
- Le fichier CSV ne doit pas contenir plus d'enregistrements de recette que prévu dans le bloc de données. Le nombre maximum d'enregistrements est prédéfini par les limites du tableau dans le bloc de données.

Tableau 9- 199 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#...)	Description
0	0000	Pas d'erreur
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	N ^{ième} appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance est utilisée.
1	8090	Le nom du fichier contient des caractères non valides.
1	8092	Aucun fichier CSV approprié trouvé pour l'importation. Cause possible : Le nom du fichier CSV ne correspond pas à celui du DB recette
1	80C0	Fichier CSV momentanément verrouillé.
1	80C1	Bloc de données momentanément verrouillé.
1	80B0	La numérotation dans l'index du fichier CSV est non continue, non croissante ou dépasse le nombre maximum (limite du tableau) dans le bloc de données.
1	80B1	La structure du bloc de données de recette ne concorde pas avec celle du fichier CSV : Le fichier CSV contient trop de champs.
1	80B2	La structure du bloc de données de recette ne concorde pas avec celle du fichier CSV : Le fichier CSV contient trop peu de champs.
1	80B6	L'attribut "Sauvegarder uniquement dans la mémoire de chargement" du DB de recette n'est pas activé.
1	80D0 +n	La structure du bloc de données de recette ne concorde pas avec celle du fichier CSV : Le type de données dans le champ n ne concorde pas (n<=46).
1	80FF	La structure du bloc de données de recette ne concorde pas avec celle du fichier CSV : Le type de données dans le champ n ne concorde pas (n>46).

9.9.1.4 Exemple de programme de recette

Conditions requises pour l'exemple de programme de recette

Les conditions requises de l'exemple de programme de recette sont les suivantes :

- Un bloc de données de recette constitué de tous les enregistrements. Le bloc de données de recette est stocké en mémoire de chargement.
- Un bloc de données de recette actif contenant la copie d'une recette en mémoire de travail.

Voir "Exemple de DB de recette (Page 513)" pour plus de détails sur le bloc de données de recette et le fichier CSV correspondant.

Créer le DB de recette actif

Dans la boîte de dialogue "Ajouter nouveau bloc" :

- Sélectionnez le bouton "Bloc de données" dans la boîte de dialogue "Ajouter nouveau bloc".
- Dans le menu déroulant "Type", sélectionnez le type de données API "Beer_recipe" que vous avez créé précédemment.

Les valeurs initiales ne sont pas requises. Les valeurs du bloc de données sont définies lorsqu'une recette est transférée du DB de recette dans le DB de recette actif. Dans l'exemple, le bloc de données de recette actif est la cible des données READ_DBL et il fournit les données source pour WRITE_DBL. L'image suivante montre le bloc de données Active_Recipe.

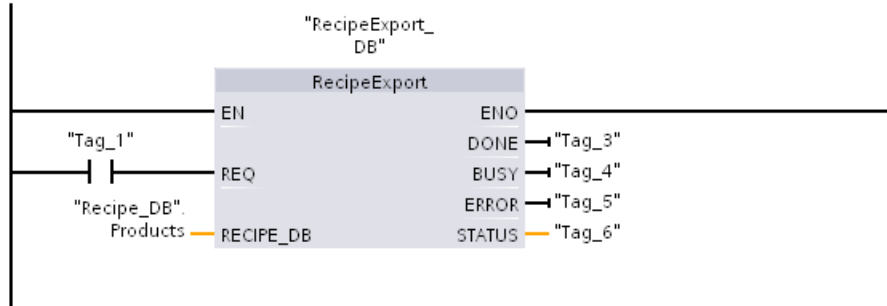
Active_Recipe			
	Name	Data type	Start value
1	Static		
2	productname	String[20]	'Beer_Recipe'
3	water	UInt	0
4	barley	UInt	0
5	wheat	UInt	0
6	hops	UInt	0
7	yeast	UInt	0
8	waterTmp	UInt	0
9	mashTmp	UInt	0
10	mashTime	UInt	0
11	QTest	UInt	0

DB d'instance

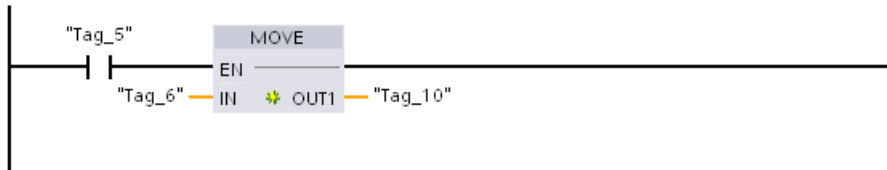
Les blocs de données d'instance utilisés par les instructions RecipeExport ("RecipeExport_DB") et RecipeImport ("RecipeImport_DB") sont créés automatiquement lorsque vous placez les instructions dans votre programme. Les blocs de données d'instance sont utilisés pour commander l'exécution des instructions. Ils ne sont pas référencés dans la logique du programme.

Exemple de programme de recette

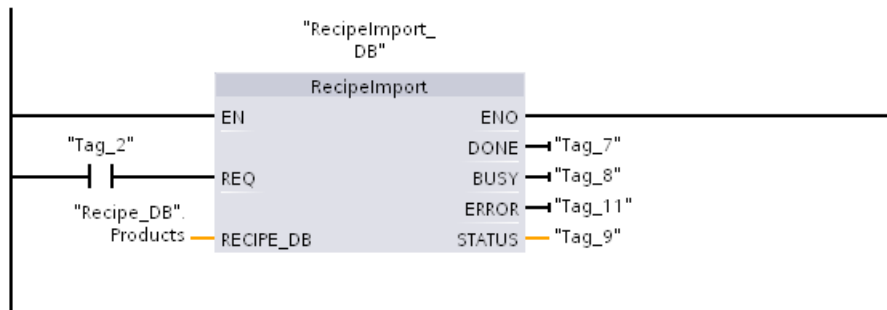
Réseau 1 Un front montant de REQ démarre l'exportation. Un fichier CSV est généré à partir des données du bloc et placé dans le répertoire Recettes de la mémoire de la CPU.



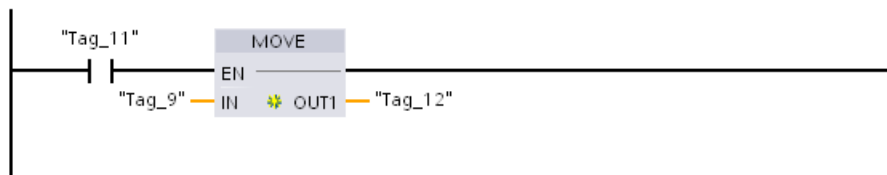
Réseau 2 Capturer la sortie STATUS de l'exécution de RecipeExport car elle n'est valable que pendant un cycle.



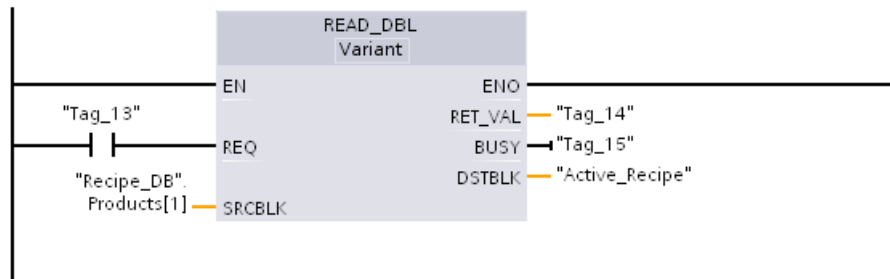
Réseau 3 Un front montant de REQ démarre l'importation. Le bloc de données de recette existant est chargé avec toutes les données lues du fichier CSV correspondant qui se trouve dans le répertoire Recettes de la mémoire de la CPU.



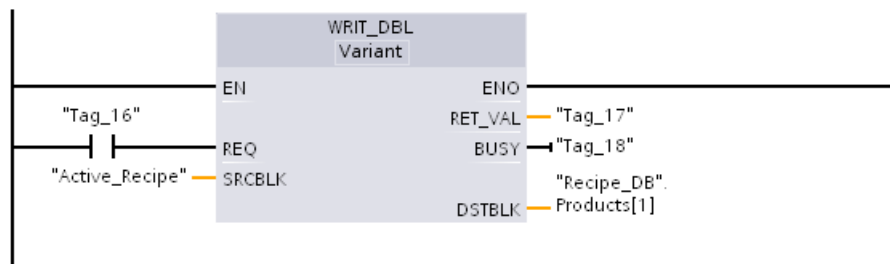
Réseau 4 Capturer la sortie STATUS de l'exécution de RecipeImport car elle n'est valable que pendant un cycle.



Réseau 5 READ_DBL copie les valeurs initiales d'une recette "Recipe_DB". Products[1] (en mémoire de chargement de la CPU) dans les valeurs du bloc de données Active_Recipe en cours (en mémoire de travail de la CPU). Après l'exécution de l'opération READ_DBL, la logique du programme peut accéder aux valeurs des éléments de la recette en utilisant les adresses dans le bloc de données Active_Recipe. Par exemple, les adresses symboliques ("Active_Recipe".productname) et ("Active_Recipe.water) fournissent à la logique du programme le nom de la recette en cours et la quantité d'eau.



Réseau 6 Pendant l'exécution, un appareil IHM peut avoir modifié la valeur d'un élément stocké dans le bloc de données Active_Recipe. Les données de recette améliorées peuvent être enregistrées en exécutant WRIT_DBL. Dans l'exemple, toutes les valeurs initiales de Recipe_DB pour la recette simple "Recipe_DB". Products[1] sont écrasées par les valeurs actuelles du bloc de données "Active_Recipe".



9.9.2 Journaux de données

Votre programme de commande peut utiliser les instructions Data log pour stocker des valeurs de données d'exécution dans des fichiers journaux permanents. La CPU stocke les fichiers journaux en mémoire flash (CPU ou carte mémoire) en format CSV standard (valeurs séparées par une virgule). La CPU organise les enregistrements de données sous forme d'un fichier d'enregistrement circulaire d'une dimension prédéterminée.

Vous utilisez les instructions Data log dans votre programme pour créer, ouvrir, écrire un enregistrement et fermer les fichiers journaux. Vous décidez des valeurs du programme qui seront consignées en créant une mémoire tampon qui définit un enregistrement de journal unique. La CPU se sert de votre mémoire tampon de données comme stockage temporaire pour un nouvel enregistrement de journal. Votre programme de commande copie les nouvelles valeurs en cours dans la mémoire tampon pendant l'exécution. Une fois toutes les valeurs de données en cours actualisées, le programme peut exécuter l'instruction DataLogWrite pour transférer les données de la mémoire tampon dans un enregistrement de journal.

Vous pouvez ouvrir, éditer, enregistrer, renommer et supprimer des fichiers journaux de la page Navigateur de Fichiers du Serveur Web. Vous devez avoir lu les droits pour visualiser le navigateur de fichiers et vous devez avoir modifié les droits pour éditer, supprimer ou renommer des fichiers journaux.

9.9.2.1 Structure des enregistrements de journaux

Les paramètres DATA et HEADER de l'instruction DataLogCreate définissent le type de données et la description d'en-tête de colonne de tous les éléments de données dans un enregistrement de journal.

Paramètre DATA pour l'instruction DataLogCreate

Le paramètre DATA désigne de la mémoire utilisée comme tampon temporaire pour un nouvel enregistrement de journal et doit correspondre à une adresse M ou DB.

Vous pouvez affecter un DB entier (dérivé d'un type de données API que vous affectez à la création du DB) ou une partie d'un DB (l'élément de DB indiqué peut être tout type de données, structure de données, type de données API ou tableau de données).

Les structures sont limitées à un niveau d'imbrication unique. Le nombre total d'éléments de données déclarés doit correspondre au nombre de colonnes indiqué dans le paramètre HEADER. Le nombre maximum d'éléments de données que vous pouvez affecter est 253 (avec horodatage) ou 255 (sans horodatage). Cette limitation maintient votre enregistrement en deçà de la limite de colonne de 256 d'une feuille Excel.

Le paramètre DATA peut indiquer des éléments de données rémanents ou non rémanents dans un DB de type "standard" (compatible avec S7-300/400) ou "optimisé".

Pour écrire un enregistrement de journal DATA, vous devez d'abord charger de nouvelles valeurs du processus dans l'enregistrement DATA temporaire, puis exécuter l'instruction DataLogWrite qui enregistre les nouvelles valeurs de l'enregistrement dans le fichier journal.

Paramètre HEADER pour l'instruction DataLogCreate

Le paramètre HEADER désigne les noms d'en-tête de colonne dans la ligne supérieure de la matrice de données codée dans le fichier CSV. Les données HEADER doivent se situer en mémoire DB ou M et les caractères doivent respecter les règles du format CSV standard avec des virgules séparant chaque nom de colonne. Les types de données possibles sont la chaîne, le tableau d'octets ou le tableau de caractères. Les tableaux de caractères ou d'octets permettent d'avoir une taille plus importante, les chaînes étant limitées à 255 octets au maximum. Le paramètre HEADER est facultatif. Si HEADER n'est pas défini, aucune ligne d'en-tête n'est créée dans le fichier journal.

9.9.2.2 Instructions de gestion des journaux de données

DataLogCreate (Créer Data Log)

Tableau 9- 200 Instruction DataLogCreate

CONT/LOG	SCL	Description
	<pre>"DataLogCreate_DB" (req:=_bool_in_, records:=_uint_in_, format:=_uint_in_, timestamp:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_string_inout_, ID:=_dword_inout_, header:=_variant_inout_, data:=_variant_inout_);</pre>	<p>Crée et initialise un fichier journal. La CPU crée le fichier dans le répertoire \DataLogs en se basant sur le paramètre NAME et ouvre implicitement le fichier pour des opérations d'écriture. Vous pouvez utiliser les instructions Data log pour stocker les données de process d'exécution par programme dans la mémoire flash de la CPU ou sur la carte mémoire.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

¹ Dans l'exemple SCL, "DataLogCreate_DB" est le nom du DB d'instance.

Tableau 9- 201 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
RECORDS	IN	UDint	Nombre maximum d'enregistrements que le journal circulaire peut contenir avant que l'entrée la plus ancienne ne soit écrasée. L'enregistrement d'en-tête n'est pas inclus. L'API doit disposer d'une mémoire de chargement suffisante pour que le journal soit créé avec succès. (valeur par défaut - 1)
FORMAT	IN	UInt	Format du journal de données : <ul style="list-style-type: none"> • 0 : Format interne (non pris en charge) • 1 : Valeurs séparées par des virgules "csv-eng" (valeur par défaut)
TIMESTAMP	IN	UInt	Format d'horodatage des données. Les en-têtes de colonne sont facultatifs pour les champs date et heure. L'horodatage peut utiliser soit l'heure système (UTC : temps universel coordonné), soit l'heure locale. <ul style="list-style-type: none"> • 0 : Pas d'horodatage • 1 : Horodatage, heure système (valeur par défaut) • 2 : Horodatage, heure locale
NAME	IN	Variant	Nom du journal de données. Vous indiquez ce nom. Cette variante accepte seulement le type de données String et doit se situer en mémoire DB ou locale. (valeur par défaut : '') Cette référence de type chaîne est le nom du fichier journal. Utilisez des caractères du jeu ASCII à l'exception des caractères \ / : * ? " < > et de l'espace.
ID	In/Out	DWord	Identificateur numérique du journal de données. Vous mémorisez cette valeur générée dont vous aurez besoin avec les autres instructions de journaux de données. Le paramètre ID est uniquement utilisé comme sortie dans l'instruction DataLogCreate. (valeur par défaut : 0) L'accès symbolique à ce paramètre n'est pas autorisé.
HEADER	In/Out	Variant	Pointeur désignant les noms d'en-tête de colonne du journal dans la ligne supérieure de la matrice de données codée dans le fichier CSV. (Valeur par défaut : nul). Les données HEADER doivent se situer en mémoire DB ou M. Les caractères doivent respecter les règles du format CSV standard avec des virgules séparant chaque nom de colonne. Les types de données possibles sont la chaîne, le tableau d'octets ou le tableau de caractères. Les tableaux de caractères ou d'octets permettent d'avoir une taille plus importante, les chaînes étant limitées à 255 octets au maximum. Le paramètre HEADER est facultatif. Si HEADER n'est pas paramétré, aucune ligne d'en-tête n'est créée dans le fichier journal.

Paramètre et type		Type de données	Description
DATA	In/Out	Variant	Pointeur désignant une structure, un type de données utilisateur (UDT) ou un tableau pour les données des enregistrements. Les données des enregistrements doivent se situer en mémoire DB ou M. Le paramètre DATA indique les éléments de données individuels (colonnes) d'un enregistrement de journal et leur type de données. Les structures sont limitées à un niveau d'imbrication unique. Le nombre d'éléments de données déclarés doit correspondre au nombre de colonnes indiqué dans le paramètre HEADER. Le nombre maximum d'éléments de données que vous pouvez affecter est 253 (avec horodatage) ou 255 (sans horodatage). Cette limitation maintient votre enregistrement en deçà de la limite de colonne de 256 d'une feuille Excel.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur. (valeur par défaut : faux)
BUSY	OUT	Bool	<ul style="list-style-type: none"> • 0 : Pas d'opération en cours • 1 : Opération en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

La CPU crée un fichier journal avec une taille fixe prédéterminée basée sur les paramètres RECORDS et DATA et organise les enregistrements de données sous forme d'un fichier journal circulaire. L'instruction DataLogCreate alloue de la mémoire CPU rémanente pour le journal entier lorsque DONE est renvoyé avec la valeur VRAI. La mémoire requise dans la CPU est plus importante que la taille du fichier en raison des valeurs de gestion du système de fichiers et apparentées. La mémoire rémanente pour le journal reste allouée jusqu'à ce que la CPU annule l'allocation de l'une des manières suivantes :

- Le programme utilisateur appelle l'instruction DataLogDelete.
- Un utilisateur du serveur Web efface le journal dans le serveur Web.
- Un utilisateur de l'outil SIMATIC Automation Tool efface le journal dans SIMATIC Automation Tool.

La suppression du fichier journal par d'autres moyens, avec un lecteur de carte par exemple, n'annule pas l'allocation de mémoire CPU rémanente pour le journal.

L'instruction DataLogWrite ajoute de nouveaux enregistrements au fichier journal jusqu'à ce que le nombre maximum d'enregistrements indiqué dans le paramètre RECORDS soit sauvegardé. L'enregistrement suivant se substituera alors à l'enregistrement le plus ancien. Une nouvelle opération DataLogWrite écrasera l'enregistrement le plus ancien suivant et ainsi de suite.

Utilisation de la mémoire :

- Les journaux ne consomment que de la mémoire de chargement.
- La taille de tous les journaux combinés est limitée par les ressources de mémoire de chargement disponibles. Huit fichiers journaux au maximum peuvent être ouverts simultanément. Vous pouvez gérer vos journaux depuis la page Web standard Navigateur de fichiers (Page 1075). Voir la description de la page Web standard pour obtenir des conseils sur le nombre de journaux à gérer simultanément.
- Le nombre maximum possible pour le paramètre RECORDS est la limite des nombres UDint (4 294 967 295). La limite effective pour le paramètre RECORD dépend de la taille d'un enregistrement individuel, de la taille des autres journaux et des ressources de mémoire de chargement disponibles. En outre, Excel limite le nombre de lignes autorisées dans une feuille Excel.

Remarque

L'exécution de la création du journal doit être achevée avant de commencer une opération d'écriture dans le journal

- Les opérations de création de journal DataLogCreate et DataLogNewFile s'étendent sur de nombreux cycles du programme. La durée réelle nécessaire à la création d'un fichier journal dépend de la structure d'un enregistrement et du nombre d'enregistrements. La logique de votre programme doit surveiller et capturer la transition à l'état VRAI du bit DONE qui indique l'achèvement de l'opération de création d'un fichier journal. Si le programme utilisateur exécute une instruction DataLogWrite avant l'achèvement d'une opération de création d'un fichier journal, l'opération ne parviendra pas à écrire un nouvel enregistrement de journal comme prévu.
- Dans certains cas, lorsqu'un balayage programme très rapide est en cours, la création du journal peut être longue. Si la création est trop lente, assurez-vous que la case à cocher Activer un temps de cycle minimum pour les OB de cycle est active et que le temps de cycle minimum est supérieur ou égal à 1 ms. Reportez-vous à Configuration du temps de cycle et de la charge de communication (Page 111) pour plus d'informations.

Remarque

L'instruction DataLogNewFile copie la structure d'un enregistrement de journal existant

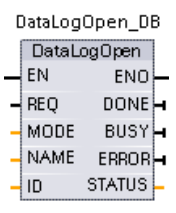
Si vous voulez éviter l'écrasement d'enregistrements, vous pouvez créer un nouveau journal sur la base du journal en cours avec l'instruction DataLogNewFile lorsque le journal en cours a atteint le nombre maximum d'enregistrements. Les nouveaux enregistrements seront alors sauvegardés dans le nouveau fichier journal. L'ancien fichier journal et ses données enregistrées sont conservés dans la mémoire flash.

Tableau 9- 202 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#....)	Description
0	0000	Pas d'erreur
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	N ^{ième} appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance interne est utilisée.
1	807F	Erreur interne
1	8090	Nom de fichier invalide
1	8091	Le paramètre NAME ne fait pas référence à une chaîne de caractères.
1	8093	Un journal de même nom existe déjà. Utilisez un autre nom, vérifiez que le fichier csv du journal n'est pas ouvert, puis utilisez la page Navigateur de fichiers (Page 1075) du Serveur Web pour supprimer le journal existant.
1	8097	La longueur de fichier demandée dépasse le maximum du système de fichiers.
1	80B2	ID de manque de ressources Remarque : Supprimez certains journaux existants ou réduisez le nombre de colonnes dans la structure de l'enregistrement pour éviter cette erreur.
1	80B3	Mémoire de chargement disponible insuffisante
1	80B4	La carte mémoire est protégée en écriture.
1	80C0	Fichier archive verrouillé
1	80C1	Trop de fichiers ouverts : huit fichiers journaux ouverts sont autorisés au maximum.
1	8253	Nombre d'enregistrements invalide
1	8353	Format sélectionné invalide
1	8453	Horodatage sélectionné invalide
1	8B24	Allocation de zone HEADER invalide : Pointe par exemple sur la mémoire locale
1	8B51	Type de données du paramètre HEADER invalide
1	8B52	Trop d'éléments de données dans le paramètre HEADER
1	8C24	Affectation de zone DATA invalide : Pointe par exemple sur la mémoire locale
1	8C51	Type de données du paramètre DATA invalide
1	8C52	Trop d'éléments de données dans le paramètre DATA

DataLogOpen (Ouvrir Data Log)

Tableau 9- 203 Instruction DataLogOpen

CONT/LOG	SCL	Description
	<pre>"DataLogOpen_DB" (req:=_bool_in_, mode:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_string_inout_, ID:=_dword_inout_);</pre>	<p>Ouvre un fichier journal pré-existant. Un journal doit être ouvert pour que vous puissiez y écrire (Page 532) de nouveaux enregistrements. Vous pouvez ouvrir et fermer les journaux de façon individuelle. Huit journaux au maximum peuvent être ouverts en même temps.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

² Dans l'exemple SCL, "DataLogOpen_DB" est le nom du DB d'instance.

Tableau 9- 204 Types de données pour les paramètres

Paramètre et type	Type de données	Description
REQ IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
MODE IN	UInt	Mode de fonctionnement : <ul style="list-style-type: none"> 0 : Ajouter aux données existantes (valeur par défaut) 1 : Effacer tous les enregistrements existants
NAME IN	Variant	Nom d'un journal existant. Cette variante accepte seulement le type de données String et peut uniquement se situer en mémoire locale, DB ou M. (valeur par défaut : '')
ID IN_OUT	DWord	Identificateur numérique d'un journal de données (valeur par défaut : 0) Remarque : L'accès symbolique à ce paramètre n'est pas autorisé.
DONE OUT	Bool	Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur. (valeur par défaut : faux)
BUSY OUT	Bool	<ul style="list-style-type: none"> 0 : Pas d'opération en cours 1 : Opération en cours
ERROR OUT	Bool	Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Vous pouvez indiquer soit le nom NAME, soit l'identificateur numérique ID (ID en tant que paramètre d'entrée) d'un journal pré-existant. Si vous indiquez les deux paramètres et que l'ID valide indiqué correspond bien au journal désigné par NAME, l'ID est utilisé et le nom NAME n'est pas pris en compte.

Le nom indiqué dans NAME doit être le nom d'un journal créé avec l'instruction DataLogCreate. Si vous indiquez uniquement le paramètre NAME et que NAME désigne un journal valide, l'ID correspondant est renvoyé (ID en tant que paramètre de sortie).

Remarque

Utilisation générale des fichiers journaux

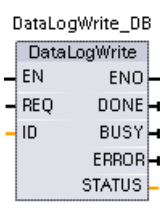
- Les fichiers journaux sont automatiquement ouverts après les instructions DataLogCreate et DataLogNewFile.
 - Les fichiers journaux sont automatiquement fermés lors d'un passage de l'API de l'état MARCHE à l'état ARRET ou lors d'une mise hors tension puis sous tension de l'API.
 - Un fichier journal doit être ouvert pour qu'une nouvelle opération DataLogWrite soit possible.
 - Huit fichiers journaux au plus peuvent être ouverts à un moment donné. Il peut exister plus de huit fichiers journaux, mais certains d'entre eux doivent être fermés pour qu'il n'y ait pas plus de huit ouverts.
-

Tableau 9- 205 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#)	Description
0	0000	Pas d'erreur
0	0002	Avertissement : Fichier journal déjà ouvert par ce programme d'application
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	N ^{ième} appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance interne est utilisée.
1	8090	La définition de journal est incompatible avec le fichier journal existant.
1	8091	Le paramètre NAME ne fait pas référence à une chaîne de caractères.
1	8092	Le journal n'existe pas.
1	80C0	Le fichier journal est verrouillé.
1	80C1	Trop de fichiers ouverts : huit fichiers journaux ouverts sont autorisés au maximum.

DataLogWrite (Écrire Data Log)

Tableau 9- 206 Instruction DataLogWrite

CONT/LOG	SCL	Description
	<pre>"DataLogWrite_DB" (req=>_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>Écrit un enregistrement dans le journal spécifié. L'enregistrement de données cible existant doit être ouvert (Page 530) pour que vous puissiez y écrire avec une instruction DataLogWrite.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

² Dans l'exemple SCL, "DataLogWrite_DB" est le nom du DB d'instance.

Tableau 9- 207 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
ID	In/Out	DWord	Identificateur numérique du journal de données. Utilisé uniquement comme entrée pour l'instruction DataLogWrite (valeur par défaut : 0) Remarque : L'accès symbolique à ce paramètre n'est pas autorisé.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 : Pas d'opération en cours 1 : Opération en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

L'adresse et la structure de données de la mémoire tampon d'enregistrements sont définies par le paramètre DATA d'une instruction DataLogCreate. Le programme utilisateur doit charger les valeurs du processus en cours d'exécution dans la mémoire tampon, puis exécuter l'instruction DataLogWrite pour copier les nouvelles données de la mémoire tampon dans le journal.

Le paramètre ID identifie un journal et une configuration d'enregistrement. L'instruction DataLogCreate génère le nombre ID ID.

S'il y a des enregistrements vides dans le fichier journal circulaire, l'instruction DataLogWrite écrit l'enregistrement vide disponible suivant. Si tous les enregistrements sont pleins, l'instruction DataLogWrite écrase l'enregistrement le plus ancien.

IMPORTANT

Les opérations de création du journal doivent être achevées avant de commencer une opération d'écriture dans le journal
--

Les opérations de création de journal DataLogCreate et DataLogNewFile s'étendent sur de nombreux cycles du programme. La durée réelle nécessaire à la création d'un fichier journal dépend de la structure d'un enregistrement et du nombre d'enregistrements. La logique de votre programme doit surveiller et capturer la transition à l'état VRAI du bit DONE qui indique l'achèvement de l'opération de création d'un fichier journal. Si une instruction DataLogWrite est exécutée avant l'achèvement d'une opération de création d'un fichier journal, l'opération ne parviendra pas à écrire un nouvel enregistrement de journal.
--

Remarque

Répercussions des journaux sur la mémoire interne de la CPU

Chaque écriture de journal prend au minimum 2 Ko de mémoire. Si votre programme écrit fréquemment de petites quantités de données, il prend au moins 2 Ko de mémoire sur chaque écriture. Une meilleure implémentation consisterait à accumuler les petits éléments de données dans un bloc de données (DB), et à écrire le bloc de données dans le journal à des intervalles moins fréquents.

Si votre programme écrit de nombreuses entrées de journal à une fréquence élevée, envisagez d'utiliser une carte mémoire SD remplaçable.

IMPORTANT

Perte potentielle de données de journaux pendant une coupure de courant affectant la CPU

Si une coupure de courant se produit pendant une opération DataLogWrite qui n'est pas encore achevée, l'enregistrement en cours de transfert dans le journal peut être perdu.

Tableau 9- 208 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#)	Description
0	0000	Pas d'erreur
0	0001	Indique que le journal est plein. Chaque journal est créé avec un nombre maximum défini d'enregistrements. Le dernier enregistrement du nombre maximum a été écrit. La prochaine opération d'écriture écrasera donc l'enregistrement le plus ancien.
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	N ^{ième} appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance interne est utilisée.
1	8092	Le journal n'existe pas.
1	80B0	Le fichier journal n'est pas ouvert (uniquement pour le mode d'ouverture explicite).

DataLogClear (Vider Data Log)

Description

Tableau 9- 209 Instruction DataLogClear

CONT/LOG	SCL	Description
<pre> *DataLogClear_ DB* DataLogClear - EN ENO - - REQ DONE - - ID BUSY - ERROR - STATUS - </pre>	<pre> "DataLogClear_DB" (REQ:= _bool_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, ID:= _dword_inout_); </pre>	<p>L'instruction "DataLogClear" efface tous les enregistrements dans un journal existant. Cette instruction ne supprime pas l'en-tête facultatif du fichier CSV (voir la description du paramètre HEADER de l'instruction "DataLogCreate (Page 525)").</p> <p>Le paramètre "ID" permet de sélectionner le journal dont les enregistrements doivent être effacés.</p>

"DataLogClear_DB" est le nom du DB d'instance.

Condition

Le journal doit être ouvert pour que vous puissiez en effacer des enregistrements (voir l'instruction "DataLogOpen (Page 530)").

Paramètres

Le tableau suivant présente les paramètres de l'instruction DataLogClear :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, L, D, T, C ou constante (T et C sont uniquement disponibles en CONT et LOG avec le S7-1500)	Exécution de l'instruction lors d'un front montant
ID	InOut	DWORD	I, Q, M, D, L	Identificateur numérique du journal
DONE	Output	BOOL	I, Q, M, D, L	L'instruction a été exécutée avec succès.
BUSY	Output	BOOL	I, Q, M, D, L	L'exécution de l'instruction n'est pas encore achevée.
ERROR	Output	BOOL	I, Q, M, D, L	<ul style="list-style-type: none"> • 0 : Pas d'erreur • 1 : Une erreur s'est produite pendant l'exécution de l'instruction. Des informations détaillées sont transmises dans le paramètre STATUS.
STATUS	Output	WORD	I, Q, M, D, L	Paramètre d'état Ce paramètre est mis à 1 pour la durée d'un appel uniquement. Pour afficher l'état, il faut donc copier le paramètre STATUS dans une zone de données libre.

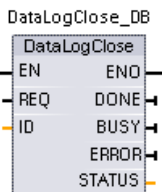
Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Paramètre STATUS

Code d'erreur* (W#16#...)	Explication
0000	Pas d'erreur
7000	Pas de traitement de tâche actif
7001	Démarrage du traitement de la tâche. Paramètres BUSY = 1, DONE = 0
7002	Appel intermédiaire (REQ sans objet). L'instruction est déjà active ; BUSY a la valeur "1".
8080	Le fichier journal sélectionné avec le paramètre ID ne peut pas être traité avec l'instruction "DataLog-Clear".
8092	Le journal n'existe pas.
80A2	Erreur d'écriture signalée par le système de fichiers
80B0	Le journal n'est pas ouvert.
80B4	La carte mémoire est protégée en écriture.
* Les codes d'erreur peuvent être affichés sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes. Pour plus d'informations sur le changement de format d'affichage, voir "Voir aussi".	

DataLogClose (Fermer Data Log)

Tableau 9- 210 Instruction DataLogClose

CONT/LOG	SCL	Description
	<pre>"DataLogClose_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>Ferme un fichier journal ouvert. Les opérations DataLogWrite sur un fichier journal fermé provoquent une erreur. Aucune opération d'écriture dans ce journal n'est autorisée tant qu'une nouvelle opération DataLogOpen n'est pas exécutée.</p> <p>Un passage à l'état ARRET ferme tous les fichiers journaux ouverts.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

² Dans l'exemple SCL, "DataLogClose_DB" est le nom du DB d'instance.

Tableau 9- 211 Types de données pour les paramètres

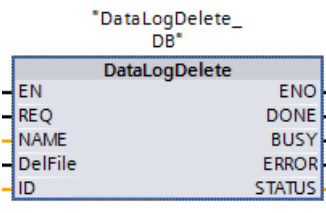
Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
ID	IN_OUT	DWord	Identificateur numérique d'un journal de données. Utilisé uniquement comme entrée pour l'instruction DataLogClose (valeur par défaut : 0) Remarque : L'accès symbolique à ce paramètre n'est pas autorisé.
DONE	OUT	Bool	Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 : Pas d'opération en cours 1 : Opération en cours
ERROR	OUT	Bool	Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Tableau 9- 212 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#)	Description
0	0000	Pas d'erreur
0	0001	Le journal n'est pas ouvert.
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	N ^{ième} appel (fonctionne) : BUSY = 1, DONE = 0
1	8092	Le journal n'existe pas.

DataLogDelete (Supprimer Data Log)

Tableau 9- 213 Instruction DataLogDelete

CONT/LOG	SCL	Description
	<pre>"DataLogDelete_DB" (REQ:=_bool_in_, NAME:=_variant_in_, DelFile:=_bool_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, ID:=_dword_inout_);</pre>	<p>Utilisez l'instruction "DataLogDelete" pour supprimer un fichier journal. Le journal et les enregistrements qu'il contient peuvent uniquement être effacés si le journal a été créé avec l'instruction "DataLogCreate" ou "DataLogNewFile".</p>

"DataLogDelete_DB" est le nom du DB d'instance.

Paramètres

Le tableau suivant présente les paramètres de l'instruction DataLogDelete :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, L, D, T, C ou constante (T et C sont uniquement disponibles en CONT et LOG avec le S7-1500)	Exécution de l'instruction lors d'un front montant
NAME	Input	VARIANT	L, D	Nom de fichier du journal
DELFILE	Input	BOOL	I, Q, M, D, L ou constante	<ul style="list-style-type: none"> 0 : Le journal est conservé. 1 : Le journal est supprimé.
ID	InOut	DWORD	I, Q, M, D, L	Identificateur numérique du journal
DONE	Output	BOOL	I, Q, M, D, L	L'instruction a été exécutée avec succès.
BUSY	Output	BOOL	I, Q, M, D, L	La suppression du journal n'est pas encore achevée.
ERROR	Output	BOOL	I, Q, M, D, L	<ul style="list-style-type: none"> 0 : Pas d'erreur 1 : Une erreur s'est produite pendant l'exécution de l'instruction. <p>Des informations détaillées sont transmises dans le paramètre STATUS.</p>
STATUS	Output	WORD	I, Q, M, D, L	<p>Paramètre d'état</p> <p>Ce paramètre est mis à 1 pour la durée d'un appel uniquement. Pour afficher l'état, il faut donc copier le paramètre STATUS dans une zone de données libre.</p>

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Paramètres NAME et ID

Sélectionnez le journal à supprimer au moyen des paramètres NAME et ID. Le paramètre ID est évalué en premier. S'il existe un journal avec l'ID indiqué, le paramètre NAME n'est pas évalué. Si le paramètre ID a la valeur "0", le paramètre NAME doit contenir une valeur de type de données STRING.

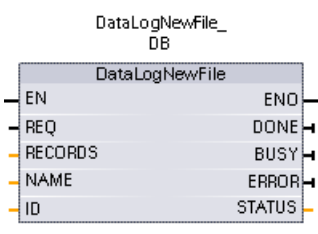
Paramètre RET_VAL

Code d'erreur* (W#16#...)	Explication
0	Pas d'erreur
7000	Pas de traitement de tâche actif
7001	Démarrage du traitement de la tâche. Paramètres BUSY = 1, DONE = 0
7002	Appel intermédiaire (REQ sans objet). L'instruction est déjà active ; BUSY a la valeur "1".
8091	Un type de données autre que STRING est utilisé dans le paramètre NAME.
8092	Le journal n'existe pas.
80A2	Erreur d'écriture signalée par le système de fichiers
80B4	La carte mémoire est protégée en écriture.

* Les codes d'erreur peuvent être affichés sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes. Pour plus d'informations sur le changement de format d'affichage, voir "Voir aussi".

DataLogNewFile (Data Log dans un nouveau fichier)

Tableau 9- 214 Instruction DataLogNewFile

CONT/LOG	SCL	Description
 <p>The diagram shows the DataLogNewFile instruction with the following connections:</p> <ul style="list-style-type: none"> EN (Enable) is connected to ENO (Enable Out). REQ (Request) is connected to DONE (Done). RECORDS (Records) is connected to BUSY (Busy). NAME (Name) is connected to ERROR (Error). ID (ID) is connected to STATUS (Status). 	<pre>"DataLogNewFile_DB" (req:=_bool_in_, records:=_udint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_DataLog_out_, ID:=_dword_inout_);</pre>	<p>Permet à votre programme de créer un nouveau fichier journal sur la base d'un fichier journal existant.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

² Dans l'exemple SCL, "DataLogNewFile_DB" est le nom du DB d'instance.

Tableau 9- 215 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
RECORDS	IN	UDInt	Nombre maximum d'enregistrements que le journal circulaire peut contenir avant que l'entrée la plus ancienne ne soit écrasée. (valeur par défaut : 1) L'enregistrement d'en-tête n'est pas inclus. La CPU doit disposer d'une mémoire de chargement suffisante pour que le journal soit créé avec succès.
NAME	IN	Variant	Nom du journal de données. Vous indiquez ce nom. Cette variante accepte seulement le type de données String et peut uniquement se situer en mémoire locale, DB ou M. (valeur par défaut : '') Cette référence chaîne est également utilisée comme nom du fichier journal. Vous devez respecter les restrictions concernant les noms dans le système de fichiers Windows. Les caractères \ / : * ? " < > et l'espace ne sont pas autorisés.
ID	In/Out	DWord	Identificateur numérique du journal de données (valeur par défaut : 0) : <ul style="list-style-type: none"> • Au moment de l'exécution, l'entrée ID identifie un journal de données valide. La configuration du nouveau journal est reprise de ce journal. • Après l'exécution, le paramètre ID devient une sortie qui renvoie l'ID du fichier journal nouvellement créé. Remarque : L'accès symbolique à ce paramètre n'est pas autorisé.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • 0 : Pas d'opération en cours • 1 : Opération en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Vous pouvez exécuter l'instruction DataLogNewFile lorsqu'un journal est rempli ou est considéré comme achevé et que vous ne voulez perdre aucune des données qui y sont contenues. Un nouveau fichier journal vide ayant la structure du fichier journal plein peut être créé. L'enregistrement d'en-tête est repris dans le journal d'origine, ainsi que les propriétés du journal d'origine (mémoire tampon d'enregistrements DATA, format des données et horodatage). Le fichier journal d'origine est implicitement fermé et le nouveau fichier journal est implicitement ouvert.

Déclenchement via les paramètres de DataLogWrite : Votre programme doit surveiller les paramètres ERROR et STATUS de chaque opération DataLogWrite. Lorsque le dernier enregistrement est écrit et qu'un journal est plein, le bit ERROR et le mot STATUS de DataLogWrite prennent la valeur 1. Ces valeurs ERROR et STATUS ne sont valables que pour un cycle. Votre logique de surveillance doit donc utiliser ERROR = 1 comme validation temporelle pour capturer la valeur STATUS, puis vérifier que STATUS est égal à 1 (le journal est plein).

Fonctionnement de DataLogNewFile : Lorsque la logique de votre programme reçoit le signal "journal plein", elle active une opération DataLogNewFile. Vous devez exécuter DataLogNewFile avec l'ID d'un journal existant et ouvert (déjà plein) mais avec un nouveau paramètre NAME unique. Une fois l'opération DataLogNewFile exécutée, une nouvelle valeur ID correspondant au nom du nouveau journal est renvoyée (en tant que paramètre de sortie). Le nouveau fichier journal est implicitement ouvert et est prêt à recevoir de nouveaux enregistrements. Les nouvelles opérations DataLogWrite destinées au nouveau fichier journal doivent utiliser la valeur ID renvoyée par l'opération DataLogNewFile.

IMPORTANT

Les opérations de création du journal doivent être achevées avant de commencer une opération d'écriture dans le journal

Les opérations de création de journal DataLogCreate et DataLogNewFile s'étendent sur de nombreux cycles du programme. La durée réelle nécessaire à la création d'un fichier journal dépend de la structure d'un enregistrement et du nombre d'enregistrements. La logique de votre programme doit surveiller et capturer la transition à l'état VRAI du bit DONE qui indique l'achèvement de l'opération de création d'un fichier journal. Si une instruction DataLogWrite est exécutée avant l'achèvement d'une opération de création d'un fichier journal, l'opération ne parviendra pas à écrire un nouvel enregistrement de journal comme prévu.

Tableau 9- 216 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#)	Description
0	0000	Pas d'erreur
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	N ^{ième} appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance interne est utilisée.
1	8090	Nom de fichier invalide
1	8091	Le paramètre NAME ne fait pas référence à une chaîne de caractères.
1	8092	Le journal n'existe pas.
1	8093	Le journal de données existe déjà.
1	8097	La longueur de fichier demandée dépasse le maximum du système de fichiers.
1	80B2	ID en manque de ressources Remarque : supprimez quelques journaux existants pour créer des ressources pour un nouveau journal.
1	80B3	Mémoire de chargement disponible insuffisante
1	80B4	La cartouche mémoire est protégée en écriture.
1	80C1	Trop de fichiers ouverts

9.9.2.3 Utilisation des journaux de données

Les fichiers journaux sont stockés en format "valeurs séparées par des virgules" (*.csv) en mémoire flash permanente. Vous pouvez visualiser les journaux de données à l'aide du serveur Web de l'API ou en retirant la carte mémoire de l'API et en l'insérant dans un lecteur de carte PC standard.

Visualisation de journaux de données avec le serveur Web API

Si le port PROFINET de l'API et un PC sont connectés à un réseau, vous pouvez utiliser un navigateur Web PC tel que Microsoft Internet Explorer ou Mozilla Firefox pour accéder au serveur Web API intégré. L'API peut être à l'état MARCHE ou ARRET lorsque vous utilisez le serveur Web API. Si l'API est à l'état MARCHE, votre programme de commande continue à s'exécuter pendant que le serveur Web API transfère les données de journaux via le réseau.

Accès par le serveur Web :

1. Activez le serveur Web dans la configuration d'appareil pour la CPU cible (Page 1040).
2. Connectez votre PC à l'API via le réseau PROFINET (Page 1044).
3. Accédez à la CPU par l'intermédiaire du serveur Web intégré (Page 1048).
4. Téléchargez, éditez et supprimez les fichiers journaux à l'aide de la page Web "Navigateur de fichiers" standard (Page 1075).
5. Ouvrez le fichier .csv à l'aide d'un tableur tel que Microsoft Excel.

Remarque

Gestion des journaux de données

Ne conservez pas plus de 1 000 journaux de données dans un système de fichiers. Dépasser ce nombre peut empêcher le serveur web d'avoir suffisamment de ressources CPU pour afficher les journaux de données.

Si vous constatez que la page web du navigateur de fichiers n'est pas en mesure d'afficher les journaux de données, alors vous devez mettre la CPU à l'état ARRET afin d'afficher et de supprimer des journaux de données.

Gérez vos journaux de données pour vous assurer que vous en conservez uniquement le nombre que vous devez respecter, et ne pas dépasser 1 000 journaux de données.

Visualisation de journaux de données sur une carte mémoire API

Si une carte mémoire S7-1200 de type "programme" est insérée dans la CPU S7-1200, vous pouvez enlever la carte mémoire et l'insérer dans une fente de carte SD (Secure Digital) ou MMC (MultiMediaCard) standard sur un PC ou une PG. L'API est à l'état ARRET lorsque la carte mémoire est retirée et le programme de commande ne s'exécute pas.

Servez-vous de l'explorateur de fichiers Windows pour naviguer jusqu'au répertoire \DataLog sur la carte mémoire. Tous vos fichiers journaux *.csv se trouvent dans ce répertoire.

Faites une copie des fichiers journaux que vous placerez sur un lecteur local de votre PC. Puis, utilisez Excel pour ouvrir une copie locale d'un fichier *.csv et non le fichier original stocké sur la carte mémoire.

IMPORTANT

Vous pouvez copier des fichiers journaux sur une carte mémoire S7-1200 à l'aide d'un lecteur PC, mais ne modifiez pas ni ne supprimez ces fichiers.

La page standard Navigateur de fichiers du serveur Web est l'outil recommandé pour la visualisation, le téléchargement (copie) et la suppression des fichiers journaux.

La navigation directe dans le système de fichiers de la carte mémoire via l'explorateur Windows présente un risque car vous pouvez supprimer ou modifier accidentellement les fichiers journaux ou d'autres fichiers système, ce qui pourrait altérer des fichiers ou rendre la carte mémoire inutilisable.

IMPORTANT

Répercussions des fichiers journaux sur les cartes mémoire

Pour garantir la performance et la fiabilité globales de votre système, limitez la fréquence des fichiers journaux à un maximum de 200 ms.

9.9.2.4 Limite de taille des fichiers journaux

Les fichiers journaux partagent la mémoire de chargement de l'API avec le programme, les données de programme, les données de configuration, les pages Web personnalisées et les données système API. Un grand programme utilisant la mémoire de chargement interne requiert une part importante de la mémoire de chargement. L'espace libre peut alors s'avérer insuffisant pour les fichiers journaux. Dans ce cas, vous pouvez utiliser une "Carte programme" (Page 150) pour augmenter la taille de la mémoire de chargement. Les CPU S7-1200 peuvent utiliser soit la mémoire de chargement interne, soit la mémoire de chargement externe, mais pas les deux simultanément.

Règle de la taille maximale pour les fichiers journaux

La taille maximale d'un fichier journal ne peut pas dépasser la taille de la mémoire de chargement libre ou 500 mégaoctets selon ce qui est le plus bas. Dans ce cas, la taille de 500 mégaoctets se rapporte à la définition décimale du mégaoctet, de sorte que la taille maximale du fichier journal est de 500 000 000 octets ou $500 \times 1\,000^2$ octets.

Tableau 9- 217 Taille de mémoire de chargement

Zone de données	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C, CPU 1217C	Stockage des données
Mémoire de chargement interne mémoire flash	1 Mo	2 Mo	4 Mo	4 Mo	Programme utilisateur et données de programme, données de configuration, journaux (Data logs), pages Web personnalisées plus données système API
Mémoire de chargement externe cartes mémoire flash "programme" optionnelles	4 Mo, 12 Mo, 24 Mo, 256 Mo, 2 Go ou 32 Go selon la taille de la carte SD				

Détermination de l'espace libre en mémoire de chargement

La quantité d'espace libre en mémoire de chargement varie pendant le fonctionnement normal au fur et à mesure que le système d'exploitation utilise et libère de la mémoire. Procédez comme suit pour consulter la taille de la mémoire de chargement.

1. Établissez une liaison en ligne entre STEP 7 et l'API S7-1200 cible.
2. Chargez le programme qui dirige les opérations du fichier journal.
3. Créez les pages Web personnalisées optionnelles dont vous avez besoin. Les pages Web standard qui accèdent aux journaux de données sont stockées dans le firmware de l'API et n'utilisent pas la mémoire de chargement.
4. Utilisez les outils En ligne et diagnostic (Page 1379) ou la page Diagnostic du serveur Web (Page 1056) pour afficher la taille de mémoire de chargement totale et l'espace libre.

Calcul de la taille d'un fichier journal (tous les enregistrements)

Lors de la création du fichier journal, la CPU alloue la taille de mémoire maximale. En plus de l'espace requis pour la totalité des enregistrements, vous devez ajouter l'espace pour enregistrer l'en-tête du fichier journal (si utilisé), de l'horodatage (si utilisé), de l'indice de l'enregistrement et la taille minimum de bloc pour l'affectation de mémoire.

Utilisez la formule suivante pour calculer la taille de vos fichiers journaux et vous assurer que vous n'enfreignez pas la règle de taille maximale.

Octets de données de fichier journal = ((octets de données dans un enregistrement + octets horodatage + 12 octets) * nombre d'enregistrements)

En-tête

Octets d'en-tête du fichier journal = octets caractères d'en-tête + 2 octets

Octets caractères d'en-tête

- Sans données d'en-tête et sans horodatages = 7 octets
- Sans données d'en-tête ni horodatages (avec un en-tête horodatage) = 21 octets
- Données d'en-tête et sans horodatages = nombre d'octets caractères dans tout le texte d'en-tête de colonne, y compris virgules de séparation
- Données d'en-tête et horodatages (avec un en-tête horodatage) = nombre d'octets caractères dans tout le texte d'en-tête de colonne, y compris virgules de séparation + 21 octets

Données

Octets de données de fichier journal = ((octets de données dans un enregistrement + octets horodatage + 12 octets) * nombre d'enregistrements)

Octets de données dans un enregistrement

Le paramètre DATA de DataLogCreate désigne une structure qui définit le nombre de champs de données et le type de données de chaque champ pour un enregistrement de données.

Multipliez le nombre d'occurrences d'un type de données précis par le nombre d'octets requis. Répétez cette opération pour chaque type de données dans un enregistrement et additionnez les nombres d'octets obtenus pour avoir tous les éléments de données de l'enregistrement.

Taille des éléments individuels

Les données des journaux sont stockées sous forme d'octets caractères dans le format de fichier CSV (valeurs séparées par des virgules). Le tableau suivant montre le nombre d'octets nécessaires pour le stockage de chaque élément de données.

Type de données	Nombre d'octets (y compris données plus un octet de virgule)
Bool	2
Byte	5
Word	7
DWord	12
Char	4

String **Exemple 1** : MaChaîne[10]
La taille de chaîne maximale est définie à 10 caractères.
caractères de texte + remplissage automatique avec des espaces = 10 octets
guillemets ouvrant et fermant + virgule = 3 octets
10 + 3 = 13 octets au total

Exemple 2 : MaChaîne2
Si aucune taille n'est indiquée entre crochets, 254 octets est pris par défaut.
caractères de texte + remplissage automatique avec des espaces = 254 octets
guillemets ouvrant et fermant + virgule = 3 octets
254 + 3 = 257 octets au total

USInt	5
UInt	7
UDInt	12
SInt	5
Int	7
DInt	12
Real	16
LReal	25
Time	15
DTL	24

Nombre d'enregistrements dans un fichier journal

Le paramètre RECORDS de l'instruction DataLogCreate définit le nombre maximum d'enregistrements dans un fichier journal.

Octets d'horodatage dans un enregistrement

- Pas d'horodatage = 0 octets
- Horodatage = 20 octets

9.9.2.5 Exemple de programme pour les journaux de données

Cet exemple de programme pour les journaux de données ne montre pas toute la logique de programme nécessaire pour récupérer des valeurs échantillons d'un processus dynamique, mais présente les opérations clés des instructions de journaux de données. La structure et le nombre de fichiers journaux que vous utilisez dépendent des exigences de commande de votre processus.

Remarque

Utilisation générale des fichiers journaux

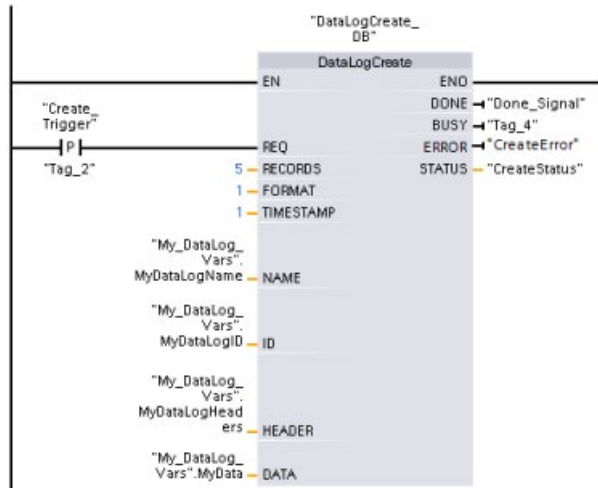
- Les fichiers journaux sont automatiquement ouverts après les instructions DataLogCreate et DataLogNew File.
 - Les fichiers journaux sont automatiquement fermés lors d'un passage de l'API de l'état MARCHE à l'état ARRET ou lors d'une mise hors tension puis sous tension de l'API.
 - Un fichier journal doit être ouvert pour qu'une opération DataLogWrite soit possible.
 - Huit fichiers journaux au plus peuvent être ouverts à un moment donné. Il peut exister plus de huit fichiers journaux, mais certains d'entre eux doivent être fermés pour qu'il n'y en ait pas plus de huit ouverts.
-

Exemple de programme de journal de données

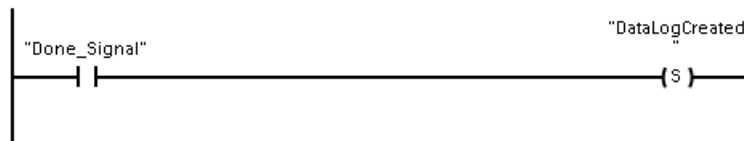
Des noms de journaux, un texte d'en-tête et la structure MyData sont créés à titre d'exemple dans un bloc de données. Les trois variables MyData stockent temporairement de nouvelles valeurs échantillons. On transfère les valeurs échantillons du processus situées à ces adresses de DB dans un fichier journal en exécutant l'instruction DataLogWrite.

My_Datalog_Vars			
	Nom	Type de données	Valeur de départ
1	Static		
2	MyNewDataLogName	String	'MyNEWDDataLog'
3	MyDataLogName	String	'MyDataLog'
4	MyDataLogID	DWord	0
5	MyDataLogHeaders	String	'Count,Temperature,Pressure'
6	MyData	Struct	
7	MyCount	Int	0
8	MyTemperature	Real	0.0
9	MyPressure	Real	0.0

Réseau 1 Un front montant de REQ démarre l'opération de création du journal.



Réseau 2 Capturer la sortie DONE de DataLogCreate car elle n'est valable que pendant un cycle.



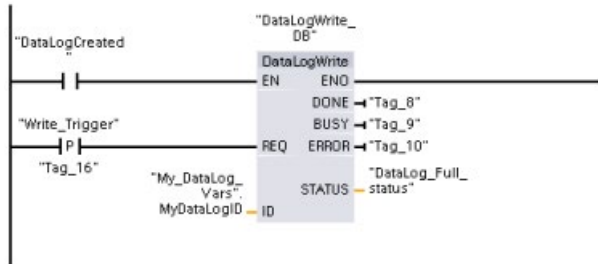
Réseau 3 En cas d'erreur, enregistrer la sortie STATUS



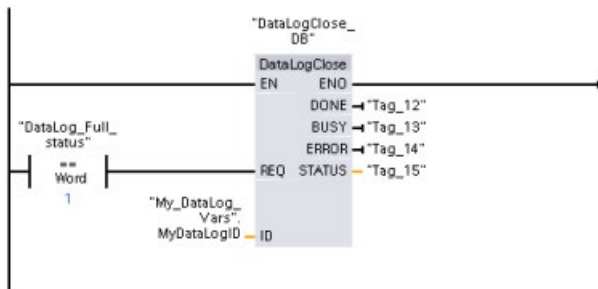
Réseau 4 Un front montant déclenche le stockage de nouvelles valeurs du processus dans la structure MyData.



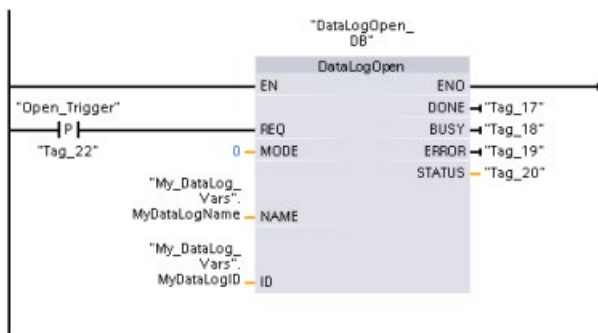
Réseau 5 L'état de l'entrée EN indique lorsque l'opération DataLogCreate est achevée. Une opération de création s'étend sur plusieurs cycles et doit être achevée avant l'exécution d'une opération d'écriture. Un front montant à l'entrée REQ est l'événement qui déclenche une opération d'écriture validée.



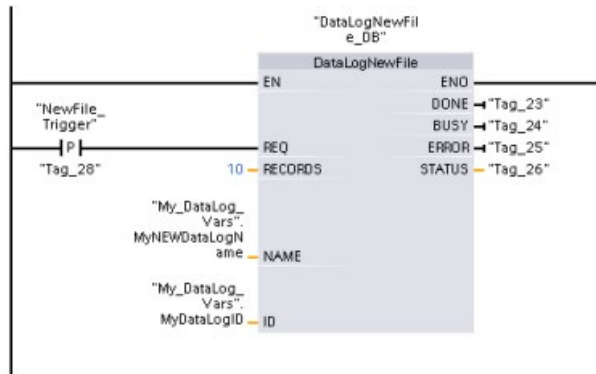
Réseau 6 Fermer le journal de données une fois le dernier enregistrement écrit. Après l'exécution de l'opération DataLogWrite qui écrit le dernier enregistrement, l'état plein du fichier journal est signalé par la mise à 1 de la sortie STATUS de DataLogWrite.



Réseau 7 Un front montant à l'entrée REQ de DataLogOpen simule l'enfoncement par un utilisateur d'un bouton sur une IHM qui ouvre un fichier journal. Si vous ouvrez un fichier journal dont tous les enregistrements sont pleins, l'opération DataLogWrite suivante écrasera l'enregistrement le plus ancien. Vous voudrez peut-être conserver l'ancien journal et, pour ce faire, créer un nouveau journal comme illustré au réseau 7.



Réseau 8 Le paramètre ID est un paramètre d'entrée/sortie (IN/OUT). Vous fournissez d'abord la valeur ID du journal existant dont vous voulez copier la structure. Une fois l'opération DataLogNewFile achevée, une nouvelle valeur ID unique pour le nouveau journal est renvoyée dans l'adresse de référence ID. La capture nécessaire du bit DONE = VRAI n'est pas montrée (voir les réseaux 1, 2 et 4 pour un exemple de logique pour le bit DONE).



Fichiers journaux créés par l'exemple de programme visualisés via le serveur Web de la CPU S7-1200

SIEMENS S7-1200 station_1 / PLC_1

21:55:10 18.12.2015 UTC Français

Utilisateur : admin [Déconnexion](#) [Désactivé](#)

Navigateur de fichiers

S7-1200 station_1 / DataLogs

Nom	Taille	Modifié le	Supprimer	Renommer
_				
DataLog001.csv	312	21:51:28 18.12.2015		<input type="text"/>
DataLog002.csv	312	21:52:18 18.12.2015		<input type="text"/>

Opérations sur les répertoires :

No file chosen

- ① L'option "Delete" (supprimer) n'est pas disponible si vous n'avez pas ouvert de session avec des privilèges de modification.
- ② L'option "Rename" (renommer) n'est pas disponible si vous n'avez pas ouvert de session avec des privilèges de modification.

Tableau 9- 218 Exemples de fichiers .csv téléchargés visualisés dans Excel

Deux enregistrements écrits dans un fichier de cinq enregistrements au maximum		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00
	3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00
	4						
	5						
Cinq enregistrements dans un fichier journal de cinq enregistrements au maximum		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	1	9/30/2010	20:26:58	1	9.86E+01	3.52E+01
	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
	7						
Une sixième opération d'écriture pour l'écriture d'un enregistrement supplémentaire dans le fichier ci-dessus qui est plein provoque l'écrasement de l'enregistrement 1 le plus ancien par l'enregistrement 6. Une nouvelle opération d'écriture provoquera l'écrasement de l'enregistrement 2 par l'enregistrement 7, et ainsi de suite.		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01
	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
	7						

Remarque

Les fichiers journaux n'utilisent plus un marqueur //END pour marquer la fin d'un fichier journal qui n'est pas plein. Avant la version V4.1 de la CPU S7-1200, les fichiers journaux qui n'étaient pas pleins comprenaient un marqueur //END.

9.10 Gestion des blocs de données

9.10.1 CREATE_DB (Créer un bloc de données)

Tableau 9- 219 Instruction CREATE_DB

CONT/LOG	SCL	Description
<div style="border: 1px solid black; padding: 5px; background-color: #e0e0e0;"> <p style="text-align: center; margin: 0;">CREATE_DB</p> <p style="margin: 0;">- EN ENO -</p> <p style="margin: 0;">- REQ RET_VAL -</p> <p style="margin: 0;">- LOW_LIMIT BUSY -</p> <p style="margin: 0;">- UP_LIMIT DB_NUM -</p> <p style="margin: 0;">- COUNT</p> <p style="margin: 0;">- ATTRIB</p> <p style="margin: 0;">- SRCBLK</p> </div>	<pre>ret_val := CREATE_DB(REQ:=_bool_in_, LOW_LIMIT:=_uint_in_, UP_LIMIT:=_uint_in_, COUNT:=_udint_in_, ATTRIB:=_byte_in_, BUSY=>_bool_out_, DB_NUM=>_uint_out_);</pre>	<p>Utilisez l'instruction "CREATE_DB" pour créer un nouveau bloc de données dans la mémoire de chargement et/ou de travail.</p> <p>L'instruction "CREATE_DB" ne modifie pas le total de contrôle du programme utilisateur.</p> <p>Un bloc de données que vous créez uniquement en mémoire de travail présente les propriétés suivantes :</p> <ul style="list-style-type: none"> • Ce bloc n'existe plus après un effacement général ou une mise hors tension puis sous tension. • Son contenu demeure inchangé lors du chargement ou lors d'un passage de l'état ARRÊT à l'état MARCHÉ.

Numéro du bloc de données

Le bloc de données créé reçoit un numéro compris dans la plage définie par les paramètres LOW_LIMIT (limite inférieure) et UP_LIMIT (limite supérieure). "CREATE_DB" affecte au DB le plus petit numéro possible dans la plage définie. Vous ne pouvez pas utiliser les numéros des blocs de données déjà contenus dans le programme utilisateur.

Pour créer un DB de numéro spécifique, entrez le même nombre pour la limite inférieure et la limite supérieure de la plage à indiquer. S'il existe déjà un DB de même numéro dans la mémoire de travail et/ou la mémoire de chargement ou si le DB existe en tant que version copiée, l'instruction prend fin et un message d'erreur est généré et renvoyé dans le paramètre RET_VAL.

Valeurs initiales du bloc de données

Le paramètre SRCBLK sert à définir les valeurs initiales pour le bloc de données à créer. Le paramètre SRCBLK est un pointeur désignant un DB ou une zone de DB où vous reprenez les valeurs initiales. Le DB désigné par le paramètre SRCBLK doit avoir été généré avec accès standard (attribut "Accès au bloc optimisé" désactivé).

- Si la zone indiquée dans le paramètre SRCBLK est plus grande que le DB généré, les valeurs jusqu'à la longueur du DB généré seront appliquées comme valeurs initiales.
- Si la zone indiquée dans le paramètre SRCBLK est plus petite que le DB généré, les valeurs restantes seront remplies par des "0".

Pour garantir la cohérence des données, vous ne devez pas modifier la zone de données pendant l'exécution de "CREATE_DB" (c'est-à-dire tant que le paramètre BUSY a la valeur TRUE).

Description fonctionnelle

L'instruction "CREATE_DB" fonctionne de manière asynchrone, ce qui signifie que son traitement s'étend sur plusieurs appels. Vous démarrez la tâche en appelant "CREATE_DB" avec REQ égal à 1.

Les paramètres de sortie "RET_VAL" et "BUSY" indiquent l'état de la tâche.

Voir aussi : DELETE_DB (Supprimer le bloc de données) (Page 561)

Paramètres

Le tableau suivant présente les paramètres de l'instruction CREATE_DB :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L ou constante	Paramètre de commande "request to activate" déclenché sur niveau REQ = 1 : demande de génération du bloc de données
LOW_LIMIT	Input	UINT	I, Q, M, D, L ou constante	Limite inférieure de la plage pour l'affectation d'un numéro de bloc de données. Le plus petit numéro possible est 60000.
UP_LIMIT	Input	UINT	I, Q, M, D, L ou constante	Limite supérieure de la plage utilisée par "CREATE_DB" pour affecter un numéro à votre bloc de données (plus grand numéro possible : 60999)
COUNT	Input	UDINT	I, Q, M, D, L ou constante	La valeur COUNT précise le nombre d'octets que vous voulez réserver pour le DB généré. Ce doit être un nombre pair. La longueur maximale est 65 534 octets.
ATTRIB	Input	BYTE	I, Q, M, D, L ou constante	<p>Vous utilisez les quatre premiers bits de l'octet dans le paramètre ATTRIB pour définir les propriétés du bloc de données* :</p> <ul style="list-style-type: none"> • Bit 0 = 0 : L'attribut "Sauvegarder uniquement dans la mémoire de chargement" n'est pas activé. • Bit 0 = 1 : L'attribut "Sauvegarder uniquement dans la mémoire de chargement" est activé. Avec ce paramétrage, le bloc de données ne prend pas d'espace dans la mémoire de travail et n'est pas inclus dans le programme. Le DB n'est pas accessible par des commandes sur bits. Lorsque le bit 0 est égal à 1, la sélection effectuée pour le bit 2 n'est pas pertinente. <p>Pour garantir la compatibilité avec STEP 7 V5.x, les bits 0 et 3 doivent être envisagés ensemble (voir ci-après).</p> <ul style="list-style-type: none"> • Bit 1 = 0 : L'attribut "Bloc de données protégé en écriture dans l'appareil" n'est pas activé. • Bit 1 = 1 : L'attribut "Bloc de données protégé en écriture dans l'appareil" est activé. <ul style="list-style-type: none"> • Bit 2 = 0 : Le bloc de données est rémanent (uniquement pour les DB générés dans la mémoire de chargement et la mémoire de travail). Le bloc de données est considéré comme rémanent si au moins une de ses valeurs a été définie comme rémanente. • Bit 2 = 1 : Le bloc de données n'est pas rémanent. <p>La rémanence n'est pas prise en charge pour les DB qui sont sauvegardés uniquement dans la mémoire de chargement ou uniquement dans la mémoire de travail. Si vous appelez l'instruction "CREATE_DB" avec l'une des deux combinaisons "rémanent et mémoire de chargement uniquement" ou "rémanent et mémoire de travail uniquement", le DB à générer ne sera pas signalé comme rémanent.</p>

9.10 Gestion des blocs de données

Paramètre	Déclaration	Type de données	Zone de mémoire	Description												
				<ul style="list-style-type: none"> Bit 3 = 0 : Création du bloc de données soit dans la mémoire de chargement, soit dans la mémoire de travail (sélection au moyen du bit 0, voir ci-dessus) Bit 3 = 1 : Création du bloc de données à la fois dans la mémoire de chargement et la mémoire de travail (le bit 0 n'est pas pertinent) <p>Pour garantir la compatibilité avec STEP 7 V5.x, les bits 0 et 3 doivent être utilisés conjointement. Lorsque le bit 3 est égal à 1, le bit 0 n'est pas pertinent.</p> <table border="1"> <thead> <tr> <th>Bit 0</th> <th>Bit 3</th> <th>Génération du bloc de données</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Dans la mémoire de travail uniquement</td> </tr> <tr> <td>1</td> <td>0</td> <td>Dans la mémoire de chargement uniquement</td> </tr> <tr> <td>Non pertinent</td> <td>1</td> <td>Dans la mémoire de travail et la mémoire de chargement</td> </tr> </tbody> </table> <ul style="list-style-type: none"> Bit 4 = 0 : Aucune valeur initiale n'est indiquée (les valeurs d'entrée dans le paramètre SRCBLK ne seront pas prises en compte) Bit 4 = 1 : Indiquer des valeurs initiales (valeurs correspondant au DB désigné par le paramètre SRCBLK) 	Bit 0	Bit 3	Génération du bloc de données	0	0	Dans la mémoire de travail uniquement	1	0	Dans la mémoire de chargement uniquement	Non pertinent	1	Dans la mémoire de travail et la mémoire de chargement
Bit 0	Bit 3	Génération du bloc de données														
0	0	Dans la mémoire de travail uniquement														
1	0	Dans la mémoire de chargement uniquement														
Non pertinent	1	Dans la mémoire de travail et la mémoire de chargement														
SRCBLK	Input	VARIANT	D	Pointeur désignant le bloc de données dont les valeurs seront utilisées pour initialiser le bloc de données à générer												
RET_VAL	Return	INT	I, Q, M, D, L	Information d'erreur												
BUSY	Output	BOOL	I, Q, M, D, L	BUSY = 1 : Le processus n'est pas encore achevé.												
DB_NUM	Output	DB_DYN (UINT)	I, Q, M, D, L	Numéro du bloc de données créé												
* Les propriétés sélectionnées ici correspondent aux attributs dans les propriétés d'un bloc de données.																

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Paramètre RET_VAL

Code d'erreur* (W#16#...)	Description
0000	Pas d'erreur
0081	La zone de destination est plus grande que la zone source. La zone source est entièrement écrite dans la zone de destination. Les octets restants de la zone de destination restent inchangés.
7000	Premier appel avec REQ = 0 : aucun transfert de données actif ; BUSY a la valeur "0".
7001	Premier appel avec REQ = 1 : Transfert de données déclenché ; BUSY a la valeur "1".
7002	Appel intermédiaire (REQ sans objet) : transfert de données déjà actif ; BUSY a la valeur "1".
8081	La zone source est plus grande que la zone de destination. La zone de destination entière est écrite. Les octets restants de la zone source ne sont pas pris en compte.
8092	La fonction "Créer un bloc de données" est actuellement indisponible parce que <ul style="list-style-type: none"> la fonction "Comprimer la mémoire utilisateur" est actuellement active ; le nombre maximum de blocs a déjà été atteint dans votre CPU.
8093	Aucun bloc de données ou un bloc de données qui n'est pas dans la mémoire de travail est indiqué comme paramètre SRCBLK.
8094	Une valeur invalide a été indiquée dans le paramètre ATTRIB.
80A1	Numéro de bloc de données incorrect : <ul style="list-style-type: none"> Le numéro est "0". Le numéro dépasse la limite supérieure de la CPU pour les numéros de DB. La limite inférieure est supérieure à la limite supérieure.
80A2	Longueur de bloc de données incorrecte : <ul style="list-style-type: none"> La longueur est "0". La longueur est un nombre impair. La longueur est supérieure à la longueur autorisée par la CPU.
80A3	Le bloc de données dans le paramètre SRCBLK n'a pas été créé avec accès standard.
80B1	Il n'y a pas de numéro de DB libre.
80B2	Mémoire de travail insuffisante
80B4	La carte mémoire est protégée en écriture.
80BB	Mémoire de chargement insuffisante
80C3	Le nombre maximum d'instructions "CREATE_DB" simultanément actives a déjà été atteint.
Information d'erreur générale	Voir aussi : Codes d'erreur communs pour les instructions avancées (Page 571)
* Vous pouvez afficher le code d'erreur sous forme de valeur entière ou hexadécimale dans l'éditeur de programmes.	

9.10.2 Instructions READ_DBL et WRIT_DBL (Lire/écrire un bloc de données en mémoire de chargement)

Tableau 9- 220 Instructions READ_DBL et WRIT_DBL

CONT/LOG	SCL	Description
<p>Diagram of READ_DBL instruction logic: EN (input) and SRCBLK (input) are connected to the instruction block. The block outputs END, RET_VAL, BUSY, and DSTBLK.</p>	<pre> READ_DBL(req:=_bool_in_, srcblk:=_variant_in_, busy=>_bool_out_, dstblk=>_variant_out_); </pre>	<p>Permet de copier les valeurs initiales d'un DB ou une partie de ces valeurs de la mémoire de chargement dans un DB cible en mémoire de travail.</p> <p>Le contenu de la mémoire de chargement n'est pas modifié pendant l'opération de copie.</p>
<p>Diagram of WRIT_DBL instruction logic: EN (input) and SRCBLK (input) are connected to the instruction block. The block outputs END, RET_VAL, BUSY, and DSTBLK.</p>	<pre> WRIT_DBL(req:=_bool_in_, srcblk:=_variant_in_, busy=>_bool_out_, dstblk=>_variant_out_); </pre>	<p>Permet de copier les valeurs en cours d'un DB ou une partie de ces valeurs de la mémoire de travail dans un DB cible en mémoire de chargement.</p> <p>Le contenu de la mémoire de travail n'est pas modifié pendant l'opération de copie.</p>

Tableau 9- 221 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	BOOL	Un signal de niveau haut démarre l'opération si BUSY = 0.
SRCBLK	IN	VARIANT	READ_DBL : Pointeur désignant le bloc de données source en mémoire de chargement WRIT_DBL : Pointeur désignant le bloc de données source en mémoire de travail
RET_VAL	OUT	INT	Code d'erreur d'exécution
BUSY	OUT	BOOL	BUSY = 1 signale que l'opération de lecture/écriture n'est pas achevée.
DSTBLK	OUT	VARIANT	READ_DBL : Pointeur désignant le bloc de données de destination en mémoire de travail WRIT_DBL : Pointeur désignant le bloc de données de destination en mémoire de chargement

Typiquement, un DB est stocké à la fois dans la mémoire de chargement (mémoire flash) et dans la mémoire de travail (mémoire vive). Les valeurs initiales sont toujours stockées en mémoire de chargement et les valeurs en cours sont toujours stockées en mémoire de travail. On peut utiliser READ_DBL pour copier un ensemble de valeurs initiales de la mémoire de chargement dans les valeurs en cours d'un DB en mémoire de travail auquel votre programme fait référence. Vous pouvez utiliser WRIT_DBL pour actualiser les valeurs initiales stockées en mémoire de chargement interne ou dans une carte mémoire avec les valeurs en cours dans la mémoire de travail.

Remarque**Répercussion de l'instruction WRIT_DBL et READ_DBL sur la mémoire flash**

L'instruction WRIT_DBL effectue des opérations d'écriture en mémoire flash (mémoire de chargement interne ou carte mémoire). Pour éviter de réduire la durée de vie de la mémoire flash, utilisez l'instruction WRIT_DBL pour des mises à jour exceptionnelles, telles que l'enregistrement de modifications apportées à un processus de production. Pour des raisons similaires, évitez les appels fréquents de READ_DBL pour les opérations de lecture.

Vous devez créer les blocs de données pour READ_DBL et WRIT_DBL avant d'appeler ces instructions dans le programme STEP 7. Si vous avez créé le DB source avec le type "standard", alors le DB de destination doit également être de type "standard". Si vous avez créé le bloc de données source avec le type "optimisé", alors le bloc de données de destination doit également être de type "optimisé".

S'il s'agit de DB standard, vous pouvez indiquer un nom de variable ou une valeur P#. La valeur P# vous permet d'indiquer et de copier un nombre quelconque d'éléments de la taille précisée (octet, mot ou double mot). Vous pouvez ainsi copier tout ou partie d'un DB. S'il s'agit de DB optimisés, vous pouvez uniquement indiquer un nom de variable ; vous ne pouvez pas utiliser l'opérateur P#. Si vous indiquez un nom de variable pour des DB standard ou optimisés (ou pour d'autres types de mémoire de travail), alors l'instruction copie les données désignées par cette variable. Il peut s'agir d'un type utilisateur, d'un tableau ou d'un élément de base. Vous pouvez uniquement utiliser le type Struct avec ces instructions si le DB est standard, pas s'il est optimisé. Vous devez utiliser un type de données utilisateur (UDT) pour une structure en mémoire optimisée. Seul un type de données utilisateur garantit que les "types de données" sont exactement identiques à la fois pour la structure source et la structure cible.

Remarque**Utilisation d'une structure (type de données Struct) dans un DB "optimisé"**

Lorsque vous utilisez un type de données Struct avec des DB "optimisés", vous devez d'abord créer un type de données utilisateur (UDT) pour la structure Struct. Vous configurez alors les deux DB - source et destination - avec le type UDT. Le type UDT garantit que les types de données à l'intérieur de la structure Struct restent cohérents pour les deux DB.

Pour les DB "standard", vous utilisez la structure Struct sans créer d'UDT.

READ_DBL et WRIT_DBL s'exécutent de manière asynchrone par rapport au cycle du programme. Le traitement s'étend sur plusieurs appels de READ_DBL et WRIT_DBL. Vous démarrez la tâche de transfert de DB en appelant l'instruction avec REQ = 1, puis vous surveillez les sorties BUSY et RET_VAL pour déterminer si le transfert de données est achevé et correct.

Remarque

Répercussion de l'instruction WRIT_DBL et READ_DBL sur la charge due à la communication

Lorsque l'instruction WRIT_DBL ou READ_DBL est continuellement active, elle peut consommer des ressources de communication à tel point que STEP 7 puisse perdre la communication avec la CPU. Pour cette raison, utilisez une entrée à front montant (Page 243) pour le paramètre REQ plutôt qu'une entrée normalement ouverte ou fermée (Page 237) qui resterait active (niveau de signal élevé) pour plusieurs cycles.

Pour garantir la cohérence des données, ne modifiez pas la zone de destination pendant le traitement de READ_DBL ou la zone source pendant le traitement de WRIT_DBL (c'est-à-dire tant que le paramètre BUSY est VRAI).

Restrictions concernant les paramètres SRCBLK et DSTBLK :

- Un bloc de données doit avoir été créé avant de pouvoir être référencé.
- La longueur d'un pointeur VARIANT de type BOOL doit être divisible par 8.
- La longueur d'un pointeur VARIANT de type STRING doit être identique dans les pointeurs source et destination.

Recettes et informations de configuration de machine

Vous pouvez utiliser les instructions READ_DBL et WRIT_DBL pour gérer les recettes et les informations de configuration de machine. Vous disposez ainsi d'une autre méthode pour assurer la rémanence de valeurs qui ne changent pas souvent, bien que vous deviez veiller à limiter le nombre d'écritures pour éviter l'usure prématurée de la mémoire flash. Dans les faits, cela permet d'augmenter la quantité de mémoire rémanente au-delà de la quantité normale prise en charge pour les mises hors tension, au moins pour les valeurs qui ne changent pas souvent. Vous pouvez, par exemple, sauvegarder des informations de recette ou de configuration de machine de la mémoire de travail dans la mémoire de chargement à l'aide de l'instruction WRIT_DBL et transférer ce type d'informations de la mémoire de chargement dans la mémoire de travail à l'aide de l'instruction READ_DBL.

Tableau 9- 222 Codes d'erreur

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
0081	Avertissement : La zone source est plus petite que la zone de destination. Les données source sont copiées dans leur intégralité, les octets surnuméraires dans la zone de destination restant inchangés.
7000	Appel avec REQ = 0 : BUSY = 0
7001	Premier appel avec REQ = 1 (fonctionne) : BUSY = 1
7002	N ^{ième} appel (fonctionne) : BUSY = 1
8051	Erreur de type de bloc de données
8081	La zone source est plus grande que la zone de destination. La zone de destination est complètement remplie et les octets restants de la source ne sont pas pris en compte.
8251	Erreur de type de bloc de données source
82B1	Bloc de données source manquant
82C0	Le DB source est en cours d'édition par une autre instruction ou une fonction de communication.
8551	Erreur de type de bloc de données de destination
85B1	Bloc de données de destination manquant
85C0	Le DB de destination est en cours d'édition par une autre instruction ou une fonction de communication.
80C3	Plus de 50 instructions READ_DBL ou 50 instructions WRIT_DBL sont actuellement en file d'attente pour exécution.

Voir aussi Recettes (Page 512)

9.10.3 ATTR_DB (Lire les attributs d'un bloc de données)

Tableau 9- 223 Instruction ATTR_DB

CONT/LOG	SCL	Description
<pre> ATTR_DB - EN ENO - REQ RET_VAL - DB_NUMBER DB_LENGTH ATTRIB </pre>	<pre> ret_val := ATTR_DB(REQ:=_bool_in_, DB_NUMBER:=_uint_in_, DB_LENGTH=>_udint_out_, ATTRIB=>_byte_out_); </pre>	<p>Utilisez l'instruction "ATTR_DB" pour obtenir des informations sur un bloc de données (DB) situé dans la mémoire de travail de la CPU. L'instruction indique les attributs activés dans le paramètre ATTRIB pour le DB sélectionné.</p> <p>Il n'est pas possible de lire la longueur pour les blocs de données à accès optimisé et pour les blocs de données présents uniquement dans la mémoire de chargement. Dans ces cas, le paramètre DB_LENGTH a la valeur "0".</p> <p>N'utilisez pas ATTR_DB pour les blocs de données à accès optimisé et réserve de mémoire activée.</p> <p>Ne lisez pas les blocs de données Motion Control avec l'instruction "ATTR_DB". Le code d'erreur 80B2 serait renvoyé.</p>

Paramètres

Le tableau suivant présente les paramètres de l'instruction ATTR_DB :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L ou constante	REQ = 1 : Demande de lecture des attributs du bloc
DB_NUMBER	Input	DB_ANY	I, Q, M, D, L ou constante	Numéro du bloc de données à interroger
RET_VAL	Output	INT	I, Q, M, D, L	Information d'erreur
DB_LENGTH	Output	UDINT	I, Q, M, D, L	<ul style="list-style-type: none"> Nombre d'octets de données que le bloc de données sélectionné contient. "0" pour les blocs de données à accès optimisé et pour les blocs de données présents uniquement dans la mémoire de chargement
ATTRIB	Output	BYTE	I, Q, M, D, L	Propriétés du bloc de données :
				<ul style="list-style-type: none"> Bit 0* = 0 : L'attribut "Sauvegarder uniquement dans la mémoire de chargement" n'est pas activé. Bit 0* = 1 : L'attribut "Sauvegarder uniquement dans la mémoire de chargement" est activé.
				<ul style="list-style-type: none"> Bit 1 = 0 : L'attribut "Bloc de données protégé en écriture dans l'appareil" n'est pas activé. Bit 1 = 1 : L'attribut "Bloc de données protégé en écriture dans l'appareil" est activé.
				<p>Si le bit 0 est égal à 1, le bit 2 n'est pas pertinent et reçoit la valeur 1.</p> <ul style="list-style-type: none"> Bit 2 = 0 : Rémanent. Le bloc de données est considéré comme rémanent si au moins une de ses valeurs a été définie comme rémanente. Bit 2 = 1 : Non rémanent. Le bloc de données entier n'est pas rémanent.
				<ul style="list-style-type: none"> Bit 3* = 0 : Le bloc de données est soit dans la mémoire de chargement (bit 0 = 1), soit dans la mémoire de travail (bit 0 = 0). Bit 3* = 1 : Le bloc de données est généré à la fois dans la mémoire de chargement et la mémoire de travail.
<p>* La relation entre le bit 0 et le bit 3 est expliquée dans les paramètres de l'instruction "CREATE_DB (Créer un bloc de données) (Page 551)".</p>				

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Paramètre RET_VAL

Code d'erreur* (W#16#...)	Explication
0000	Pas d'erreur
80A1	Erreur dans le paramètre d'entrée DB_NUMBER : le paramètre effectif sélectionné <ul style="list-style-type: none"> est "0" ; est supérieur au numéro de DB maximum autorisé par la CPU utilisée.
80B1	Le bloc de données avec le numéro indiqué n'existe pas dans la CPU.
80B2	Il n'est pas possible de lire les blocs de données des objets technologiques de Motion Control avec l'instruction "ATTR_DB".
Information d'erreur générale	Voir aussi : Codes d'erreur communs pour les instructions avancées (Page 571)
* Vous pouvez afficher le code d'erreur sous forme de valeur entière ou hexadécimale dans l'éditeur de programmes.	

9.10.4 DELETE_DB (Supprimer le bloc de données)

Tableau 9- 224 Instruction DELETE_DB

CONT/LOG	SCL	Description
<pre> DELETE_DB - EN ENO - - REQ Ret_Val - - DB_NUMBER BUSY - </pre>	<pre> ret_val := DELETE_DB(REQ := _bool_in_, DB_NUMBER := _uint_in_, BUSY => _bool_out_); </pre>	<p>Utilisez l'instruction "DELETE_DB" pour supprimer un bloc de données (DB) que le programme utilisateur a créé par appel de l'instruction "CREATE_DB (Page 551)".</p> <p>Si le bloc de données n'a pas été créé avec "CREATE_DB", "DELETE_DB " renvoie le code d'erreur W#16#80B5 dans le paramètre RET_VAL.</p> <p>L'appel de DELETE_DB ne supprime pas le bloc de données sélectionné immédiatement, mais au point de contrôle du cycle après l'exécution de l'OB de cycle.</p>

Description fonctionnelle

L'instruction "DELETE_DB" fonctionne de manière asynchrone, ce qui signifie que son exécution s'étend sur plusieurs appels. Vous démarrez l'opération de suppression en appelant l'instruction avec REQ égal à 1.

Le paramètre de sortie BUSY et les octets 2 et 3 du paramètre de sortie RET_VAL indiquent l'état de la tâche.

La suppression du bloc de données est achevée lorsque le paramètre de sortie BUSY a la valeur FALSE.

Paramètres

Le tableau suivant présente les paramètres de l'instruction DELETE_DB :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L ou constante	REQ = 1 : Demande de suppression du bloc de données dont le numéro figure dans le paramètre DB_NUMBER
DB_NUMBER	Input	UINT	I, Q, M, D, L ou constante	Numéro du bloc de données à supprimer
RET_VAL	Output	INT	I, Q, M, D, L	Information d'erreur (voir "Paramètre RET_VAL")
BUSY	Output	BOOL	I, Q, M, D, L	BUSY = 1 : Le processus n'est pas encore achevé.

Pour plus d'informations sur les types de données valides, voir "Types de données (Page 129)".

Paramètre RET_VAL

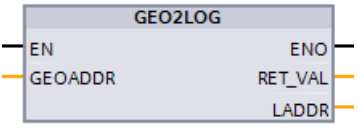
Code d'erreur* (W#16#...)	Explication
0000	Pas d'erreur
7000	Premier appel avec REQ = 0 : aucun transfert de données actif ; BUSY a la valeur "0".
7001	Premier appel avec REQ = 1 : Transfert de données déclenché ; BUSY a la valeur "1".
7002	Appel intermédiaire (REQ sans objet) : transfert de données déjà actif ; BUSY a la valeur "1".
80A1	Erreur dans le paramètre d'entrée DB_NUMBER : <ul style="list-style-type: none"> • La valeur dans le paramètre est "0". • La valeur dans le paramètre est supérieure au numéro de DB maximum autorisé par la CPU utilisée.
80B1	Le bloc de données avec le numéro indiqué n'existe pas dans la CPU.
80B4	Le bloc de données ne peut pas être supprimé, car la carte mémoire de la CPU est protégée en écriture.
80B5	Le bloc de données n'a pas été créé avec l'instruction "CREATE_DB".
80BB	Mémoire de chargement insuffisante
80C3	La fonction "Supprimer le bloc de données" ne peut pas être exécutée en ce moment en raison d'un manque de ressources temporaire.
Information d'erreur générale	Voir aussi : Codes d'erreur communs pour les instructions avancées (Page 571)
* Vous pouvez afficher le code d'erreur sous forme de valeur entière ou hexadécimale dans l'éditeur de programmes.	

9.11 Gestion des adresses

9.11.1 GEO2LOG (Déterminer l'identification matérielle à partir de l'emplacement)

Vous pouvez utiliser l'instruction GEO2LOG pour déterminer l'identificateur matériel sur la base des informations relatives à l'emplacement.

Tableau 9- 225 Instruction GEO2LOG

CONT / LOG	SCL	Description
	<pre>ret_val := GEO2LOG(GEOADDR:=_variant_in_out_, laddr:=_word_out_);</pre>	<p>Vous pouvez utiliser l'instruction GEO2LOG pour déterminer l'identificateur matériel sur la base des informations relatives à l'emplacement.</p>

L'instruction GEO2LOG détermine l'identificateur matériel sur la base des informations relatives à l'emplacement que vous définissez à l'aide du type de données système GEOADDR :

En fonction du type de matériel que vous définissez dans le paramètre HWTYPE, les informations suivantes sont évaluées à partir des autres paramètres GEOADDR :

- Avec HWTYPE = 1 (système PROFINET IO) :
 - Seul IOSYSTEM est évalué. Les autres paramètres de GEOADDR ne sont pas pris en compte.
 - L'identificateur matériel du système PROFINET IO est transmis.
- Avec HWTYPE = 2 (périphérique PROFINET IO) :
 - IOSYSTEM et STATION sont évalués. Les autres paramètres de GEOADDR ne sont pas pris en compte.
 - L'identificateur matériel du périphérique PROFINET IO est transmis.
- Avec HWTYPE = 3 (châssis) :
 - Seuls IOSYSTEM et STATION sont évalués. Les autres paramètres de GEOADDR ne sont pas pris en compte.
 - L'identificateur matériel du châssis est transmis.
- Avec HWTYPE = 4 (module) :
 - IOSYSTEM, STATION, et SLOT sont évalués. Le paramètre SUBSLOT de GEOADDR n'est pas pris en compte.
 - L'identificateur matériel du module est transmis.
- Avec HWTYPE = 5 (sous-module) :
 - Tous les paramètres de GEOADDR sont évalués.
 - L'identificateur matériel du sous-module est transmis.

Le paramètre AREA du type de données système GEOADDR n'est pas évalué.

Tableau 9- 226 Types de données pour les paramètres

Paramètre et type		Type de données	Description
GEOADDR	IN/OUT or IN ?	Variant	Pointeur désignant la structure du type de données système GEOADDR. Le type de données système GEOADDR contient les informations relatives à l'emplacement à partir desquelles l'ID matériel est déterminé. Voir le "type de données système GEOADDR (Page 569)" pour plus d'informations.
RET_VAL	OUT or RETURN ?	Int	Transmission d'informations relatives aux erreurs.
LADDR	OUT	HW_ANY	Identificateur matériel de l'ensemble ou du module. Le numéro est automatiquement affecté et est sauvegardé dans les propriétés dans la configuration matérielle.

Pour plus d'informations sur les types de données valides, voir "Présentation des types de données valides" dans l'aide en ligne STEP 7.


Tableau 9- 227 Codes d'erreur

RET_VAL* (W#16#...)	Explication
0	Pas d'erreur.
8091	Valeur invalide dans GEOADDR pour HWTYPE.
8094	Valeur invalide dans GEOADDR pour IOSYSTEM.
8095	Valeur invalide dans GEOADDR pour STATION.
8096	Valeur invalide dans GEOADDR pour SLOT.
8097	Valeur invalide dans GEOADDR pour SUBSLOT.
* Les codes d'erreur peuvent être affichés sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.	

9.11.2 LOG2GEO (Déterminer l'emplacement à partir de l'identification matérielle)

L'instruction LOG2GEO vous permet de déduire l'adresse géographique (emplacement du module) de l'adresse logique correspondant à un identificateur matériel.

Tableau 9- 228 Instruction LOG2GEO

CONT/LOG	SCL	Description
	<pre>ret_val := LOG2GEO(laddr:=_word_in_, GEOADDR:=_variant_in_out_);</pre>	<p>L'instruction LOG2GEO vous permet de déterminer l'emplacement du module correspondant à un identificateur matériel.</p>

L'instruction LOG2GEO détermine l'adresse géographique correspondant à une adresse logique à l'aide de l'identificateur matériel :

- Utilisez le paramètre LADDR pour sélectionner l'adresse logique à l'aide de l'identificateur matériel.
- GEOADDR contient l'adresse géographique correspondant à l'adresse logique définie dans l'entrée LADDR.

Remarque

Dans les cas où le type HW ne prend pas en charge un composant, un numéro de sous-emplacement pour un module 0 est renvoyé.

Un erreur est émise si l'entrée LADDR n'accède pas à un objet HW.

Tableau 9- 229 Types de données pour les paramètres

Paramètre et type		Type de données	Description
LADDR	IN	HW_ANY	Identificateur matériel du réseau IO ou du module. Le numéro est attribué automatiquement et est enregistré dans les propriétés de la CPU ou de l'interface dans la configuration matérielle.
RET_VAL	OUT	Int	Code d'erreur de l'instruction
GEOADDR	IN_OUT	Variant	Pointeur désignant le type de données système GEOADDR. Le type de données système GEOADDR contient les informations relatives à l'emplacement. Voir le "type de données système GEOADDR (Page 569)" pour plus d'informations.

Pour plus d'informations sur les types de données valides, voir "Présentation des types de données valides" dans l'aide en ligne STEP 7.

Tableau 9- 230 Codes d'erreur

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	L'adresse spécifiée dans le paramètre LADDR n'est pas valide.
* Les codes d'erreur peuvent être affichés sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.	

9.11.3 IO2MOD (Déterminer l'identification matérielle à partir d'une adresse E/S)

Vous pouvez utiliser l'instruction IO2MOD pour déterminer l'identificateur matériel du module depuis une adresse d'entrée/sortie d'un (sous-)module.

Tableau 9- 231 Instruction IO2MOD

CONT / LOG	SCL	Description
	<pre>ret_val := IO2MOD(ADDR:=_word_in_, LADDR:=_word_out_);</pre>	L'instruction IO2MOD vous permet de déterminer l'emplacement du module correspondant à un identificateur matériel.

L'instruction IO2MOD détermine l'identificateur matériel du module depuis une adresse d'entrée/sortie (I, Q, PI, PQ) d'un (sous-)module.

Entrez l'adresse d'entrée/sortie dans le paramètre ADDR. Si une série d'adresses d'entrée/sortie est utilisée dans ce paramètre, seule la première adresse est évaluée pour déterminer l'identificateur matériel. Si la première adresse est spécifiée correctement, la longueur de spécification d'adresse dans l'ADDR est sans importance. Si une zone d'adresses englobant plusieurs modules ou adresses inutilisées est utilisée, l'identificateur matériel du premier module peut également être déterminé.

Si aucune adresse d'entrée/sortie d'un (sous-)module n'est spécifiée dans le paramètre ADDR, le code d'erreur "8090" est transmis dans le paramètre RET_VAL.

Remarque

Saisie d'une adresse d'entrée/sortie dans SCL

Vous ne pouvez pas effectuer une programmation à l'aide de l'identification d'accès d'entrée/sortie "%QWx:P" dans SCL. Dans ce cas, utilisez le nom de variable symbolique ou l'adresse absolue dans la mémoire image.

Tableau 9- 232 Types de données pour les paramètres

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
ADDR	IN or IN/OUT ?	Variant	I, Q, M, D, L	Adresse d'entrée/sortie (I, Q, PI, PQ) dans un (sous-)module. Assurez-vous que l'accès par tranche n'est pas utilisé pour le paramètre ADDR. Si c'est le cas, des valeurs incorrectes sont transmises dans le paramètre LADDR .
RET_VAL	OUT or RETURN ?	Int	I, Q, M, D, L	Code d'erreur de l'instruction.
LADDR	OUT	HW_IO	I, Q, M, D, L	Identificateur matériel déterminé (adresse logique) du (sous-)module d'entrée/sortie.

Pour plus d'informations sur les types de données valides, voir "Présentation des types de données valides" dans l'aide en ligne STEP 7.

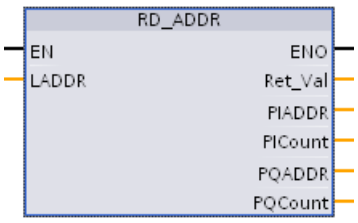
Tableau 9- 233 Codes d'erreur

RET_VAL* (W#16#...)	Explication
0	Pas d'erreur.
8090	L'adresse d'entrée/sortie spécifiée dans le paramètre ADDR n'est utilisée par aucun composant matériel.
* Les codes d'erreur peuvent être affichés sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.	

9.11.4 RD_ADDR (Déterminer les adresses E/S à partir de l'identification matérielle)

L'instruction RD_ADDR vous permet de déterminer les adresses d'E/S d'un sous-module.

Tableau 9- 234 Instruction RD_ADDR

CONT/LOG	SCL	Description
	<pre>ret_val := RD_ADDR(laddr:=_word_in_, PIADDR=>_uint_out_, PICount=>_uint_out_, PQADDR=>_uint_out_, PQCount=>_uint_out_,);</pre>	<p>L'instruction RD_ADDR vous permet de déterminer les adresses d'E/S d'un sous-module.</p>

L'instruction RD_ADDR détermine la longueur et l'adresse de début des entrées ou sorties à l'aide de l'identificateur matériel d'un sous-module :

- Utilisez le paramètre LADDR pour sélectionner le module d'entrées ou de sorties à l'aide de l'identificateur matériel.
- Les paramètres de sortie suivants sont utilisés selon qu'il s'agisse d'un module d'entrées ou de sorties :
 - Dans le cas d'un module d'entrées, les valeurs déterminées s'affichent dans les paramètres PIADDR et PICOUNT.
 - Dans le cas d'un module de sorties, les valeurs déterminées s'affichent dans les paramètres PQADDR et PQCOUNT.
- Les paramètres PIADDR et PQADDR contiennent tous les deux l'adresse de début des adresses d'E/S du module.
- Les paramètres PICOUNT et PQCOUNT contiennent tous les deux le nombre d'octets des entrées ou sorties (1 octet pour 8 entrées/sorties, 2 octets pour 16 entrées/sorties).

Tableau 9- 235 Types de données pour les paramètres

Paramètre et type		Type de données	Description
LADDR	IN	HW_IO	Identificateur matériel du (sous-)module
RET_VAL	OUT	Int	Code d'erreur de l'instruction
PIADDR	OUT	UDInt	Adresse de début du module d'entrées
PICOUNT	OUT	UInt	Nombre d'octets des entrées
PQADDR	OUT	UDInt	Adresse de début du module de sorties
PQCOUNT	OUT	UInt	Nombre d'octets des sorties

Pour plus d'informations sur les types de données valides, voir "Présentation des types de données valides" dans l'aide en ligne STEP 7.

Tableau 9- 236 Codes d'erreur

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	L'identificateur matériel du module dans le paramètre LADDR n'est pas valide.
* Les codes d'erreur peuvent être affichés sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.	

9.11.5 Type de données système GEOADDR

Adresse géographique

Le type de données système GEOADDR contient l'adresse géographique d'un module (ou les informations relatives à l'emplacement).

- Adresse géographique de PROFINET IO :
Pour PROFINET IO, l'adresse géographique se compose de l'identifiant du système PROFINET IO, du numéro d'appareil, du numéro d'emplacement et du sous-module (si un sous-module est utilisé).
- Adresse géographique de PROFIBUS DP :
Pour PROFIBUS DP, l'adresse géographique se compose de l'identifiant du système maître DP, du numéro de poste et du numéro d'emplacement.

Vous trouverez les informations relatives à l'emplacement des modules dans la configuration matérielle de chaque module.

Structure du type de données système GEOADDR

La structure GEOADDR est automatiquement créée si vous entrez "GEOADDR" comme type de données dans un bloc de données.

Nom du paramètre	Type de données	Description
GEOADDR	STRUCT	
HWTYPE	UINT	Type de matériel : <ul style="list-style-type: none"> • 1 : Réseau IO (PROFINET/PROFIBUS) • 2 : Périphérique IO/esclave DP • 3 : Châssis • 4 : Module • 5 : Sous-module Si le type de matériel n'est pas pris en charge par l'instruction, HWTYPE affiche "0".
AREA	UINT	Identificateur de zone : <ul style="list-style-type: none"> • 0 = CPU • 1 = PROFINET IO • 2 = PROFIBUS DP • 3 = AS-i
IOSYSTEM	UINT	Système PROFINET IO (0=unité centrale dans le châssis)
STATION	UINT	<ul style="list-style-type: none"> • N° du châssis si l'identificateur de zone AREA = 0 (module central). • Numéro de station si l'identificateur de zone AREA > 0.
SLOT	UINT	Numéro d'emplacement
SUBSLOT	UINT	Numéro du sous-module. Ce paramètre a pour valeur "0" si aucun sous-module n'est disponible ou ne peut être connecté.

9.12 Codes d'erreur communs pour les instructions avancées

Tableau 9- 237 Codes d'erreur communs pour les instructions avancées

Code d'erreur (W#16#....) ¹	Description
8x22 ²	Zone trop petite pour l'entrée
8x23	Zone trop petite pour la sortie
8x24	Zone d'entrée illicite
8x25	Zone de sortie illicite
8x28	Affectation illicite de bit d'entrée
8x29	Affectation illicite de bit de sortie
8x30	La zone de sortie est un DB en lecture seule.
8x3A	DB inexistant

¹ Si l'une de ces erreurs se produit pendant l'exécution d'un bloc de code, la CPU reste à l'état MARCHE (par défaut) ou peut être configurée pour passer à l'état ARRET. Vous pouvez optionnellement utiliser l'instruction GetError ou GetErrorID dans ce bloc de code pour gérer l'erreur localement (la CPU reste à l'état MARCHE) et créer une réaction programmée à l'erreur.

² Le "x" représente le numéro de paramètre avec l'erreur. Les numéros de paramètre commencent par 1.

Instructions technologiques

10.1 Comptage (compteurs rapides)

Les instructions de comptage de base, décrites sous "Compteurs" (Page 255), sont limitées au comptage d'événements qui se produisent à une fréquence inférieure au cycle de la CPU S7-1200. La fonction Compteur rapide (HSC) permet de compter des impulsions apparaissant à une fréquence supérieure au cycle de l'automate programmable. Il est en outre possible de configurer le compteur rapide afin qu'il mesure la fréquence et la période des impulsions générées ou de manière telle qu'il puisse être utilisé par Motion Control pour lire un signal de codeur moteur.

Pour utiliser la fonction HSC, vous devez d'abord valider et configurer le compteur rapide à l'aide de l'onglet Propriétés de la CPU dans l'écran Configuration des appareils. Pour vous initier à la configuration des compteurs rapides, reportez-vous à "Configuration d'un compteur rapide" (Page 590).

Une fois que vous avez chargé la configuration matérielle, le compteur rapide peut compter les impulsions ou mesurer la fréquence sans qu'il soit nécessaire d'appeler des instructions. Lorsque le compteur est en mode Comptage ou Période, la valeur de comptage est automatiquement capturée et actualisée dans la mémoire image (mémoire I) à chaque cycle. Si le compteur rapide est en mode Fréquence, la valeur de la mémoire image est la fréquence en Hz.

En plus de compter et mesurer, le compteur rapide peut générer des événements d'alarmes de processus, fonctionner selon l'état d'entrées physiques ou produire une impulsion de sortie conformément à un événement de compteur indiqué (CPU de version V4.2 ou plus). L'instruction technologique CTRL_HSC_EXT permet la commande du compteur rapide par le programme utilisateur. CTRL_HSC_EXT actualise les paramètres du HSC et renvoie les valeurs les plus récentes lorsqu'il est exécuté. L'instruction CTRL_HSC_EXT est utilisable lorsque le compteur rapide est en mode Comptage, Période ou Fréquence.

Remarque

L'instruction CTRL_HSC_EXT remplace l'ancienne instruction CTRL_HSC pour les projets destinés à des CPU de version V4.2 ou plus. L'instruction CTRL_HSC_EXT fournit toutes les fonctionnalités de l'instruction CTRL_HSC, ainsi que certaines nouvelles fonctions. L'ancienne instruction CTRL_HSC est mise à disposition uniquement pour assurer la compatibilité avec les vieux projets S7-1200 et ne doit pas être utilisée dans les nouveaux projets.

10.1.1 Instruction CTRL_HSC_EXT (Commande de compteurs rapides)

10.1.1.1 Présentation de l'instruction

Tableau 10- 1 Instruction CTRL_HSC_EXT

CONT / LOG	SCL	Description
	<pre>"CTRL_HSC_1_DB" (hsc:=_hw_hsc_in_, done:=_done_out_, busy:=_busy_out_, error:=_error_out_, status:=_status_out_, ctrl:=_variant_in_);</pre>	<p>Chaque instruction CTRL_HSC_EXT (Commander le compteur rapide (avancé)) utilise une structure de données définie par le système et sauvegardée dans un DB global défini par l'utilisateur pour ranger les données du compteur. Vous affectez les types de données HSC_Count, HSC_Period ou HSC_Frequency en tant que paramètre d'entrée à l'instruction CTRL_HSC_EXT.</p>

- STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- Dans l'exemple SCL, "CTRL_HSC_1_DB" est le nom du DB d'instance.

Tableau 10- 2 Types de données pour les paramètres

Paramètre	Déclaration	Type de données	Description
HSC	IN	HW_HSC	Identificateur de compteur rapide
CTRL	IN_OUT	Variant	Entrée SFB et données de renvoi. Remarque : Pour plus d'informations, voir "Types de données système (SDT) de l'instruction CTRL_HSC_EXT (Page 578)".
DONE	OUT	Bool	1= Indique que SFB est terminé. Toujours 1 car SFB est synchrone
BUSY	OUT	Bool	Toujours 0, la fonction n'est jamais occupée
ERROR	OUT	Bool	1 = Indique une erreur
STATUS	OUT	Word	Code d'erreur d'exécution Remarque : Voir le tableau "Codes d'erreur d'exécution" ci-dessous pour plus d'informations.

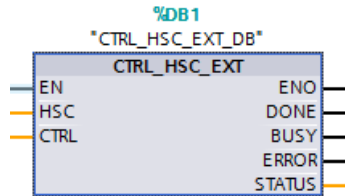
Tableau 10- 3 Codes d'erreur d'exécution

STATUS (W#16#)	Description
0	Pas d'erreur
80A1	L'identificateur HSC n'accède pas à un compteur rapide
80B1	Valeur illicite dans NewDirection
80B4	Valeur illicite dans NewPeriod
80B5	Valeur illicite dans NewOpModeBehavior
80B6	Valeur illicite dans NewLimitBehavior
80D0	SFB 124 non disponible

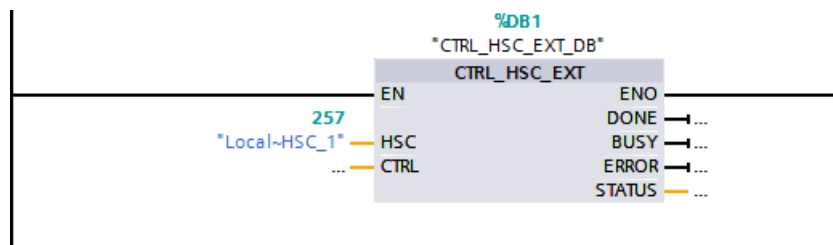
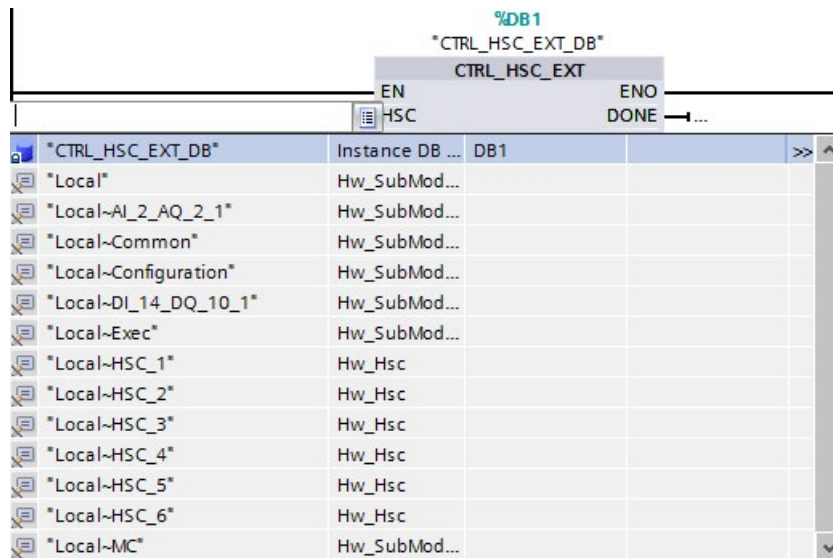
10.1.1.2 Exemple

Suivez les étapes ci-dessous pour utiliser l'instruction CTRL_HSC_EXT :

1. Placez l'instruction CTRL_HSC_EXT dans le réseau CONT, ce qui crée également le bloc de données d'instance "CTRL_HSC_EXT_DB" suivant :



2. Reliez l'identification matérielle du compteur rapide, figurant dans les propriétés du compteur rapide, à la broche "HSC" de l'instruction CONT. Vous pouvez également sélectionner l'un des six objets "Hw_Hsc" dans la liste déroulante qui s'affiche pour cette entrée. Le nom de variable par défaut pour HSC1 est "Local~HSC_1" :



10.1 Comptage (compteurs rapides)

3. Créez un bloc de données global de nom "Data_Block_1" (vous pouvez également utiliser un bloc de données global existant) :
 - Dans "Data_Block_1", repérez une ligne vide et ajoutez la variable "MyHSC".
 - Dans la colonne "Type de données", ajoutez l'un des types de données système (SDT) suivants. Sélectionnez le SDT correspondant au type de comptage configuré pour le HSC. Vous trouverez plus d'informations sur les SDT pour compteurs rapides plus loin dans cette section. La liste déroulante ne mentionne pas ces types de données ; vous devez donc veiller à saisir le nom du SDT exactement comme indiqué ici : HSC_Count, HSC_Period ou HSC_Frequency.
 - Une fois que vous avez saisi le type de données, vous pouvez développer la variable "MyHSC" pour voir tous les champs contenus dans la structure de données. Les types de données s'affichent pour chacun des champs et vous pouvez modifier les valeurs initiales prises par défaut :

Data_block_1			
	Nom	Type de données	Valeur de départ
1	Static		
2	MyHSC	HSC_Count	
3	CurrentCount	DInt	0
4	CapturedCount	DInt	0
5	SyncActive	Bool	false
6	DirChange	Bool	false
7	CmpResult_1	Bool	false
8	CmpResult_2	Bool	false
9	OverflowNeg	Bool	false
10	OverflowPos	Bool	false
11	EnHSC	Bool	false
12	EnCapture	Bool	false
13	EnSync	Bool	false
14	EnDir	Bool	false
15	EnCV	Bool	false
16	EnSV	Bool	false
17	EnReference1	Bool	false
18	EnReference2	Bool	false
19	EnUpperLmt	Bool	false
20	EnLowerLmt	Bool	false
21	EnOpMode	Bool	false
22	EnLmtBehavior	Bool	false
23	EnSyncBehavior	Bool	false
24	NewDirection	Int	0
25	NewOpModeBeha...	Int	0
26	NewLimitBehavior	Int	0
27	NewSyncBehavior	Int	0
28	NewCurrentCount	DInt	0
29	NewStartValue	DInt	0
30	NewReference1	DInt	0
31	NewReference2	DInt	0
32	NewUpperLimit	DInt	0
33	New_Lower_Limit	DInt	0

4. Affectez la variable "Data_block_1'. MyHSC" à la broche d'entrée CTRL de l'instruction CTRL_HSC_EXT :

– Sélectionnez "Data_Block_1".

Variable	Type	DB
#Initial_Call	Bool	
#Remanence	Bool	
~-Port_1	Hw_Interface	
~-Port_2	Hw_Interface	
Automatic update	Pip	
CTRL_HSC_EXT_DB	Instance DB of HSC [SF...	DB1
Data_block_1	Global DB	DB2
Local	Hw_SubModule	
Local~AI_2_AQ_2_1	Hw_SubModule	
Local~Common	Hw_SubModule	
Local~Configuration	Hw_SubModule	
Local~Device	Hw_Device	
Local~DI_14_DQ_10_1	Hw_SubModule	
Local~Exec	Hw_SubModule	
Local~HSC_1	Hw_Hsc	
Local~HSC_2	Hw_Hsc	
Local~HSC_3	Hw_Hsc	
Local~HSC_4	Hw_Hsc	
Local~HSC_5	Hw_Hsc	
Local~HSC_6	Hw_Hsc	
Local~MC	Hw_SubModule	

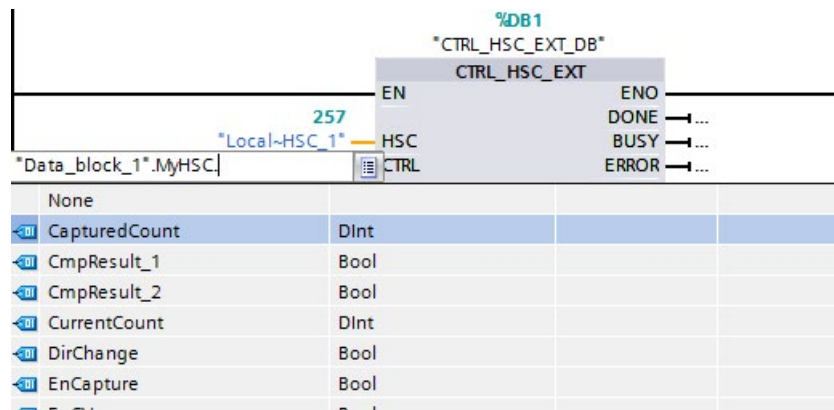
– Sélectionnez "MyHSC".

Variable	Type	DB
None		
MyHSC	HSC_Count	

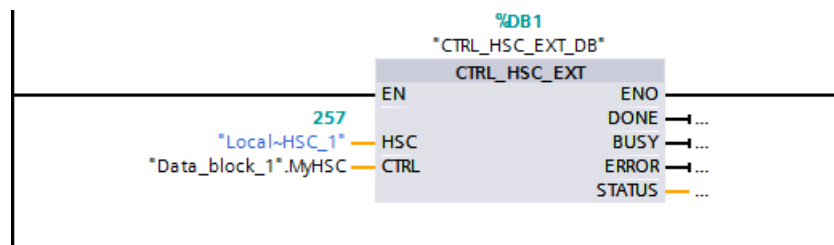
– Supprimez le point (.) après "Data_Block_1'.MyHSC". Puis, cliquez en dehors de la boîte ou bien appuyez une fois sur la touche Échap puis sur la touche Entrée.

Remarque

Il ne faut pas se contenter d'appuyer sur la touche Entrée une fois le point supprimé après "Data_Block_1'.MyHSC", car le point est alors automatiquement réintroduit dans la boîte.



– L'entrée CTRL achevée se présente comme suit :



Une fois le compteur rapide configuré dans l'automate programmable, vous pouvez exécuter l'instruction CTRL_HSC_EXT. Si une erreur se produit, ENO est mis à 0 et la sortie STATUS indique le code d'erreur.

Voir aussi

Types de données système (SDT) de l'instruction CTRL_HSC_EXT (Page 578)

10.1.1.3 Types de données système (SDT) de l'instruction CTRL_HSC_EXT

Les types de données système (SDT) suivants s'utilisent uniquement avec le paramètre CTRL de l'instruction CTRL_HSC_EXT. Pour les utiliser, créez un bloc de données utilisateur et ajoutez un objet ayant le type de données du SDT correspondant au mode configuré pour le HSC (type de comptage). STEP 7 n'affiche pas ces types de données dans la liste déroulante. Saisissez le nom du SDT exactement tel qu'il est montré.

Les entrées des SDT du compteur rapide sont caractérisées par le préfixe "En" ou "New". Les entrées commençant par "En" valident une fonction HSC ou actualisent le paramètre correspondant. Le préfixe "New" identifie la valeur d'actualisation. Pour que la nouvelle valeur prenne effet, il faut que le bit "En" correspondant soit vrai et que la valeur "New" soit valide. Lors de l'exécution de l'instruction CTRL_HSC_EXT, le programme applique les modifications aux entrées et actualise les sorties avec la référence SDT appropriée.

SDT HSC_Count

Le type de données "HSC_Count" correspond à un compteur rapide configuré pour le mode "Comptage". Le mode Comptage offre les possibilités suivantes :

- Accès au décompte d'impulsions en cours
- Verrouillage du décompte d'impulsions en cours sur un événement d'entrée
- Réinitialisation du décompte d'impulsions en cours à la valeur de départ sur un événement d'entrée
- Accès aux bits d'état signalant que certains événements HSC se sont produits
- Désactivation du compteur rapide au moyen d'une entrée logicielle ou matérielle
- Modification du sens de comptage au moyen d'une entrée logicielle ou matérielle
- Modification du décompte d'impulsions en cours
- Modification de la valeur de départ (utilisé lorsque la CPU passe à l'état MARCHE ou lorsque la fonction Synchronisation est déclenchée)
- Modification de deux valeurs de référence (ou prédéfinies) indépendantes utilisées aux fins de comparaison
- Modification des limites de comptage inférieure et supérieure
- Modification de la manière dont le compteur rapide réagit lorsque le décompte d'impulsions atteint ces limites
- Génération d'un événement d'alarme de processus lorsque le décompte d'impulsions en cours atteint une valeur de référence (prédéfinie)
- Génération d'un événement d'alarme de processus lorsque l'entrée de synchronisation (réinitialisation) devient active
- Génération d'un événement d'alarme de processus lorsque le sens de comptage change sur la base d'une entrée externe
- Génération d'une impulsion de sortie unique à l'apparition d'un événement de comptage indiqué

Lorsqu'un événement se produit et que l'instruction CTRL_HSC_EXT s'exécute, cette dernière met un bit d'état à 1. À l'exécution suivante de l'instruction CTRL_HSC_EXT, l'instruction efface le bit d'état à moins que l'événement ne se soit reproduit avant l'exécution de l'instruction.

Tableau 10- 4 Structure HSC_Count

Élément de structure	Déclaration	Type de données	Description
CurrentCount	OUT	Dint	Renvoie la valeur de comptage en cours du HSC
CapturedCount	OUT	Dint	Renvoie la valeur du compteur capturée lorsque l'événement d'entrée indiqué s'est produit
SyncActive	OUT	Bool	Bit d'état : l'entrée Sync était activée
DirChange	OUT	Bool	Bit d'état : le sens de comptage a changé
CmpResult1	OUT	Bool	Bit d'état : l'événement CurrentCount égale Reference1 s'est produit
CmpResult2	OUT	Bool	Bit d'état : l'événement CurrentCount égale Reference2 s'est produit
OverflowNeg	OUT	Bool	Bit d'état : CurrentCount a atteint la limite LowerLimit
OverflowPos	OUT	Bool	Bit d'état : CurrentCount a atteint la limite UpperLimit
EnHSC	IN	Bool	Si vrai, active le compteur rapide pour le comptage d'impulsions ; si faux, désactive le comptage
EnCapture	IN	Bool	Valide l'entrée Capture si vrai ; l'entrée Capture n'a pas d'effet si faux
EnSync	IN	Bool	Valide l'entrée Sync si vrai ; l'entrée Sync n'a pas d'effet si faux
EnDir	IN	Bool	Valide l'entrée en vigueur de la valeur NewDirection
EnCV	IN	Bool	Valide l'entrée en vigueur de la valeur NewCurrentCount
EnSV	IN	Bool	Valide l'entrée en vigueur de la valeur NewStartValue
EnReference1	IN	Bool	Valide l'entrée en vigueur de la valeur NewReference1
EnReference2	IN	Bool	Valide l'entrée en vigueur de la valeur NewReference2
EnUpperLmt	IN	Bool	Valide l'entrée en vigueur de la valeur NewUpperLimit
EnLowerLmt	IN	Bool	Valide l'entrée en vigueur de la valeur New_Lower_Limit
EnOpMode	IN	Bool	Valide l'entrée en vigueur de la valeur NewOpModeBehavior
EnLmtBehavior	IN	Bool	Valide l'entrée en vigueur de la valeur NewLimitBehavior
EnSyncBehavior	IN	Bool	Cette valeur n'est pas utilisée.
NewDirection	IN	Int	Sens de comptage : 1 = incrémentation ; -1 = décrémentation ; toutes les autres valeurs sont réservées.
NewOpModeBehavior	IN	Int	Réaction du compteur rapide en cas de débordement : 1 = le compteur rapide arrête de compter (il doit être désactivé puis réactivé pour poursuivre le comptage) ; 2 = le compteur rapide continue de fonctionner ; toutes les autres valeurs sont réservées.
NewLimitBehavior	IN	Int	Valeur CurrentCount résultante en cas de débordement : 1 = régler CurrentCount à la limite opposée ; 2 = régler CurrentCount à la valeur de départ StartValue ; toutes les autres valeurs sont réservées.
NewSyncBehavior	IN	Int	Cette valeur n'est pas utilisée.
NewCurrentCount	IN	Dint	Valeur de comptage en cours
NewStartValue	IN	Dint	Valeur initiale du compteur rapide
NewReference1	IN	Dint	Valeur de référence 1
NewReference2	IN	Dint	Valeur de référence 2

Élément de structure	Déclaration	Type de données	Description
NewUpperLimit	IN	Dint	Valeur limite de comptage supérieure
New_Lower_Limit	IN	Dint	Valeur limite de comptage inférieure

SDT HSC_Period

Le type de données "HSC_Period" correspond à un compteur rapide configuré pour le mode "Période". L'instruction CTRL_HSC_EXT fournit l'accès par programme au nombre d'impulsions d'entrée pendant un intervalle de mesure indiqué. Cette instruction permet de calculer la période entre impulsions d'entrée avec une résolution haute de l'ordre des nanosecondes.

Tableau 10- 5 Structure HSC_Period

Élément de structure	Déclaration	Type de données	Description
ElapsedTime	OUT	UDInt	Voir la description ci-après
EdgeCount	OUT	UDInt	Voir la description ci-après
EnHSC	IN	Bool	Si vrai, active le compteur rapide pour la mesure de période ; si faux, désactive la mesure de période
EnPeriod	IN	Bool	Valide l'entrée en vigueur de la valeur NewPeriod
NewPeriod	IN	Int	Indique la durée de l'intervalle de mesure en millisecondes. Les seules valeurs autorisées sont 10, 100 ou 1000 ms.

ElapsedTime renvoie le temps, en nanosecondes, entre les derniers événements de comptage d'intervalles de mesure séquentiels. Si aucun événement de comptage ne s'est produit pendant un intervalle de mesure, ElapsedTime renvoie le temps cumulé depuis le dernier événement de comptage. ElapsedTime a une plage comprise entre 0 et 4 294 967 280 nanosecondes (0x0000 0000 à 0xFFFF FFF0). La valeur en retour 4 294 967 295 (0xFFFF FFFF) signale qu'un débordement de période s'est produit. Un débordement signale que le temps entre des fronts d'impulsion est supérieur à 4,295 secondes et que la période ne peut pas être calculée avec cette instruction. Les valeurs comprises entre 0xFFFF FFF1 et 0xFFFF FFFE sont réservées.

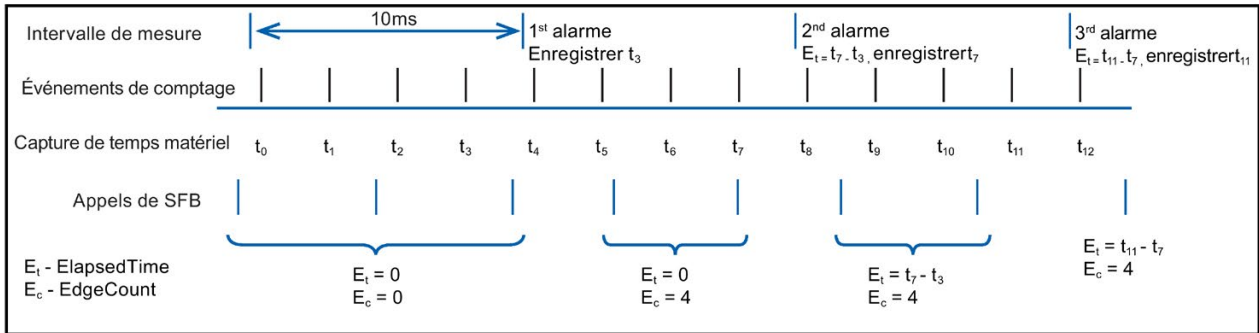
EdgeCount renvoie le nombre d'événements de comptage reçus pendant l'intervalle de mesure. La période ne peut être calculée que si la valeur de EdgeCount est supérieure à zéro. Si ElapsedTime est soit égal à 0 (aucune impulsion d'entrée reçue), soit égal à 0xFFFF FFFF (débordement de période), EdgeCount n'est pas valide.

Si EdgeCount est valide, utilisez la formule suivante pour calculer la période en nanosecondes : période = ElapsedTime/EdgeCount

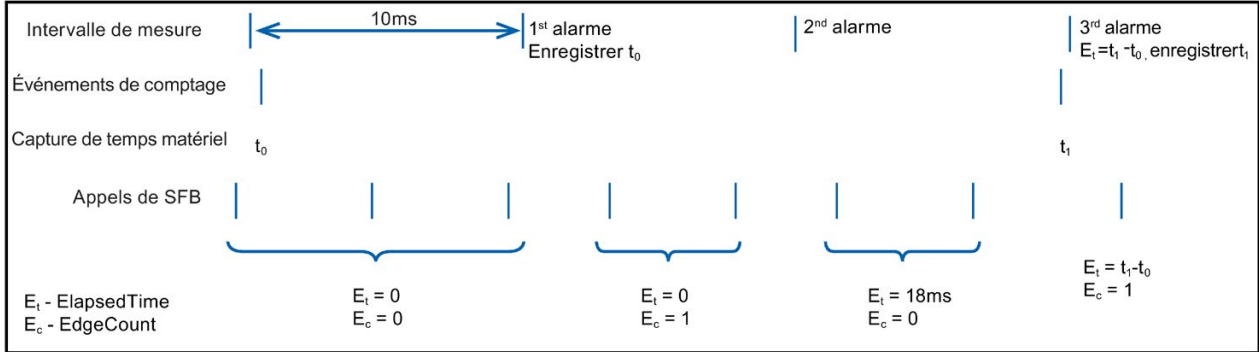
La valeur de période calculée est une moyenne des périodes de toutes les impulsions qui se produisent pendant l'intervalle de mesure. Si la période d'une impulsion entrante est supérieure à l'intervalle de mesure (10, 100 ou 1 000 ms), le calcul de la période nécessite plusieurs intervalles de mesure.

Les exemples suivants montrent comment l'instruction effectue des mesures de période :

Exemple 1 : Plusieurs évènements de comptage dans un intervalle de mesure



Exemple 2 : Zéro et un évènement de comptage dans plusieurs intervalles de mesure



Règles :

1. Si E_i = 0, l'intervalle de temps est non invalide
2. Else, intervalle de temps = E_i / E_c

SDT HSC_Frequency

Le type de données "HSC_Frequency" correspond à un compteur rapide configuré pour le mode "Fréquence". L'instruction CTRL_HSC_EXT fournit l'accès par programme à la fréquence des impulsions d'entrée, mesurée pendant une période précisée.

L'utilisation de l'instruction CTRL_HSC_EXT en mode Fréquence offre les possibilités suivantes :

Tableau 10- 6 Structure HSC_Frequency

Élément de structure	Déclaration	Type de données	Description
Frequency	OUT	DInt	Renvoie une fréquence en Hz, mesurée pendant la durée de l'intervalle de mesure. Lorsque le compteur rapide décroît, l'instruction renvoie une fréquence négative.
EnHSC	IN	Bool	Si vrai, active le compteur rapide pour la mesure de fréquence ; si faux, désactive la mesure de fréquence
EnPeriod	IN	Bool	Valide l'entrée en vigueur de la valeur NewPeriod
NewPeriod	IN	Int	Indique la durée de l'intervalle de mesure en millisecondes. Les seules valeurs autorisées sont 10, 100 ou 1 000 millisecondes.

L'instruction CTRL_HSC_EXT mesure la fréquence avec la même technique de mesure que le mode Période afin de trouver les valeurs ElapsedTime et EdgeCount. L'instruction calcule la fréquence en Hz sous forme de valeur entière signée au moyen de la formule : fréquence = EdgeCount/ElapsedTime

Si vous avez besoin d'une valeur en virgule flottante pour la fréquence, vous pouvez utiliser la formule ci-dessus pour la fréquence lorsque le compteur rapide est en mode Période. Notez que, en mode Période, la valeur ElapsedTime est renvoyée en nanosecondes et peut nécessiter une mise à l'échelle.

10.1.2 Utilisation du compteur rapide

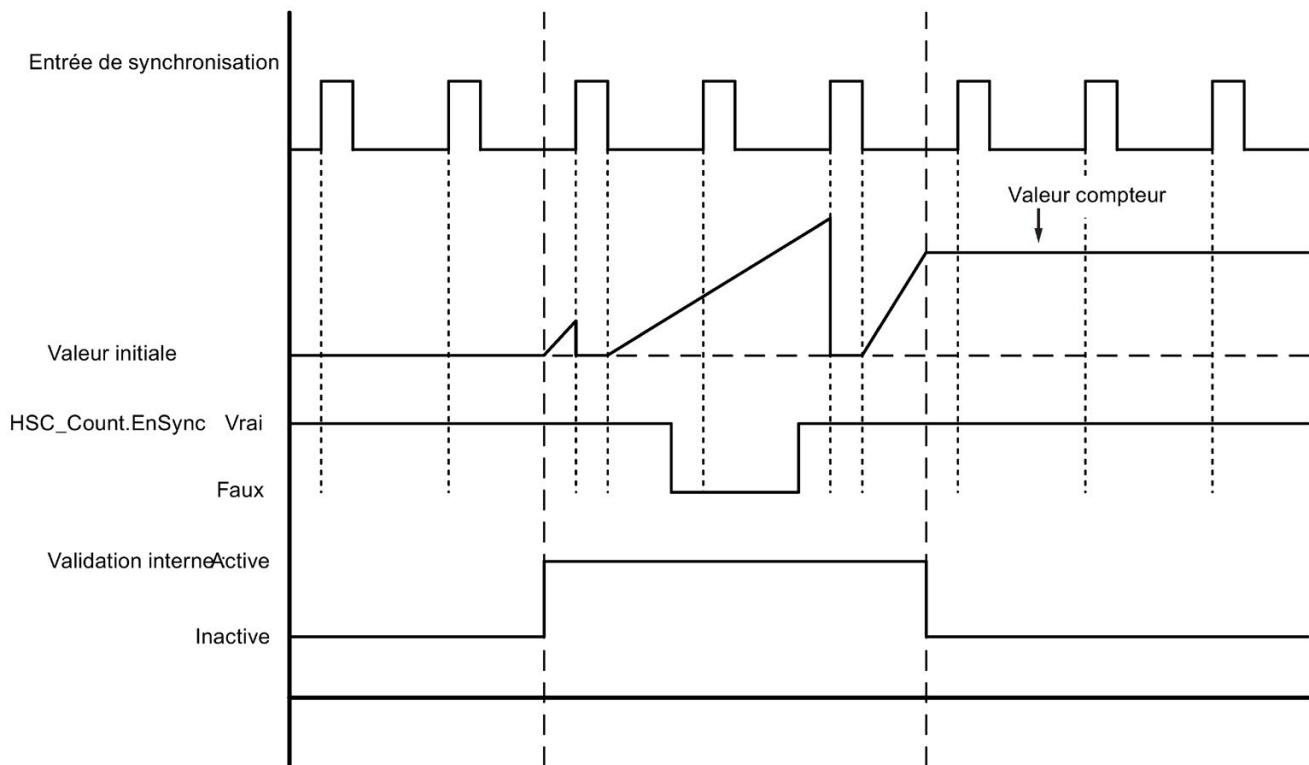
10.1.2.1 Fonction Synchronisation

Vous utilisez la fonction Synchronisation (Sync) pour régler le compteur à la valeur initiale avec un signal d'entrée externe. Vous pouvez modifier la valeur initiale en exécutant l'instruction CTRL_HSC_EXT. Cela permet à l'utilisateur de synchroniser la valeur de comptage en cours avec une valeur désirée à l'apparition d'un signal d'entrée externe.

La synchronisation se produit toujours à l'apparition du signal d'entrée et est effective quel que soit l'état de la validation interne. Vous devez mettre le bit "HSC_Count.EnSync" à 1 pour activer la fonction Sync.

L'instruction CTRL_HSC_EXT met le bit d'état HSC_Count.SyncActive à 1 une fois la synchronisation achevée. Elle met le bit d'état HSC_Count.SyncActive à 0 s'il n'y a pas eu de synchronisation depuis la dernière exécution de l'instruction.

La figure ci-dessous montre un exemple de synchronisation lorsque le signal d'entrée est configuré pour un niveau haut actif :



Remarque

Les filtres d'entrée configurés retardent le signal de commande de l'entrée TOR.

Cette fonction d'entrée n'est disponible que lorsque le compteur rapide est configuré pour le mode Comptage.

Pour plus d'informations sur la manière de configurer la fonction Synchronisation, voir Fonctions d'entrée (Page 597).

10.1.2.2 Fonction Validation

De nombreuses applications requièrent que les opérations de comptage commencent ou s'arrêtent en lien avec d'autres événements. Dans de tels cas, le comptage est démarré et arrêté au moyen de la fonction de validation interne. Chaque voie HSC comporte deux validations : une validation logicielle et une validation matérielle. L'état de ces validations détermine l'état de la validation interne (voir le tableau ci-dessous).

La validation interne est ouverte si la validation logicielle est ouverte et que la validation matérielle est ouverte ou n'a pas été configurée. Si la validation interne est ouverte, le comptage démarre. Si la validation interne est fermée, aucune autre impulsion de comptage n'est prise en compte et le comptage s'arrête.

Tableau 10- 7 États de la fonction Validation

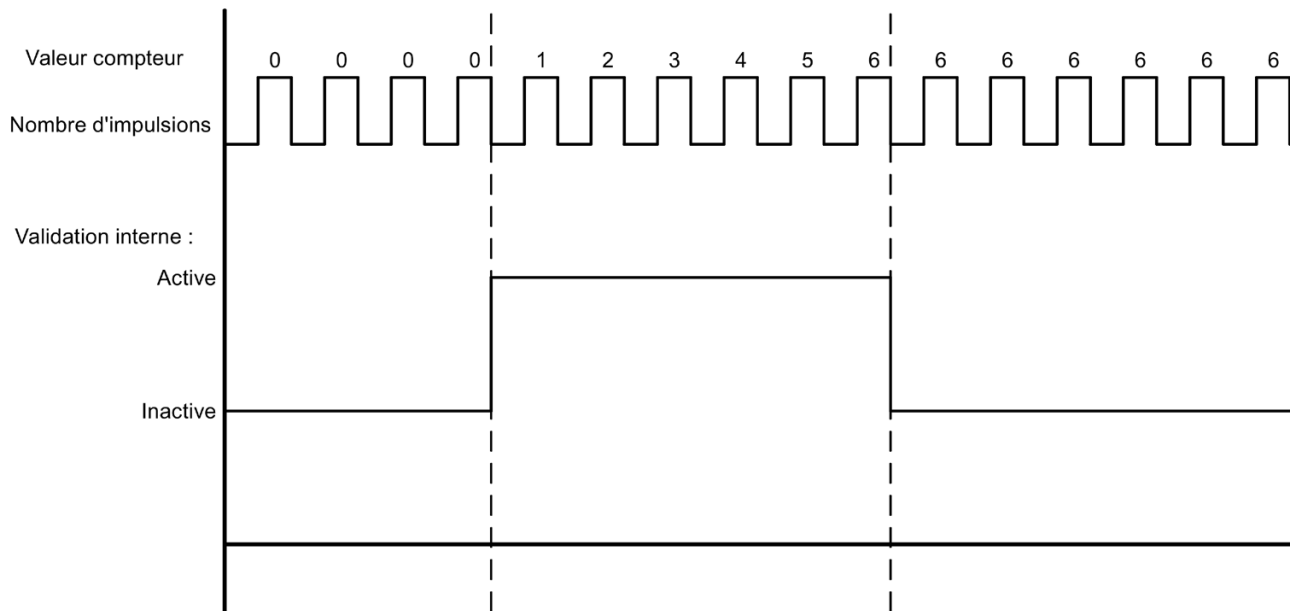
Validation matérielle	Validation logicielle	Validation interne
Ouverte/non configurée	Ouverte	Ouverte
Ouverte/non configurée	Fermée	Fermée
Fermée	Ouverte	Fermée
Fermée	Fermée	Fermée

Le terme "ouvert" est défini comme état l'état actif de la validation. De manière similaire, le terme "fermé" est défini comme état l'état inactif de la validation.

Vous commandez la validation logicielle avec le bit de validation "HSC_Count.EnHSC" dans le SDT relié à l'instruction CTRL_HSC_EXT. Pour ouvrir la validation logicielle, mettez le bit "HSC_Count.EnHSC" à vrai. Pour fermer la validation logicielle, mettez le bit "HSC_Count.EnHSC" à faux. Exécutez l'instruction CTRL_HSC_EXT pour actualiser l'état de la validation logicielle.

La validation matérielle est facultative et vous pouvez l'activer ou la désactiver dans la section des propriétés du compteur rapide. Pour commander un processus de comptage avec la validation matérielle uniquement, il faut que la validation logicielle reste ouverte. Si vous ne configurez pas de validation matérielle, la validation matérielle est considérée comme étant toujours ouverte et l'état de la validation interne est identique à celui de la validation logicielle.

La figure ci-dessous montre un exemple de validation matérielle s'ouvrant et se fermant avec une entrée TOR. L'entrée TOR est configurée pour un niveau actif haut :



Remarque

Les filtres d'entrée configurés retardent le signal de commande de l'entrée TOR.

La fonction de validation matérielle n'est disponible que lorsque le compteur rapide est configuré pour le mode Comptage. En modes Période et Fréquence, l'état de la validation interne est identique à celui de la validation logicielle.

En mode Période, la validation logicielle est commandée par "HSC_Period.EnHSC".

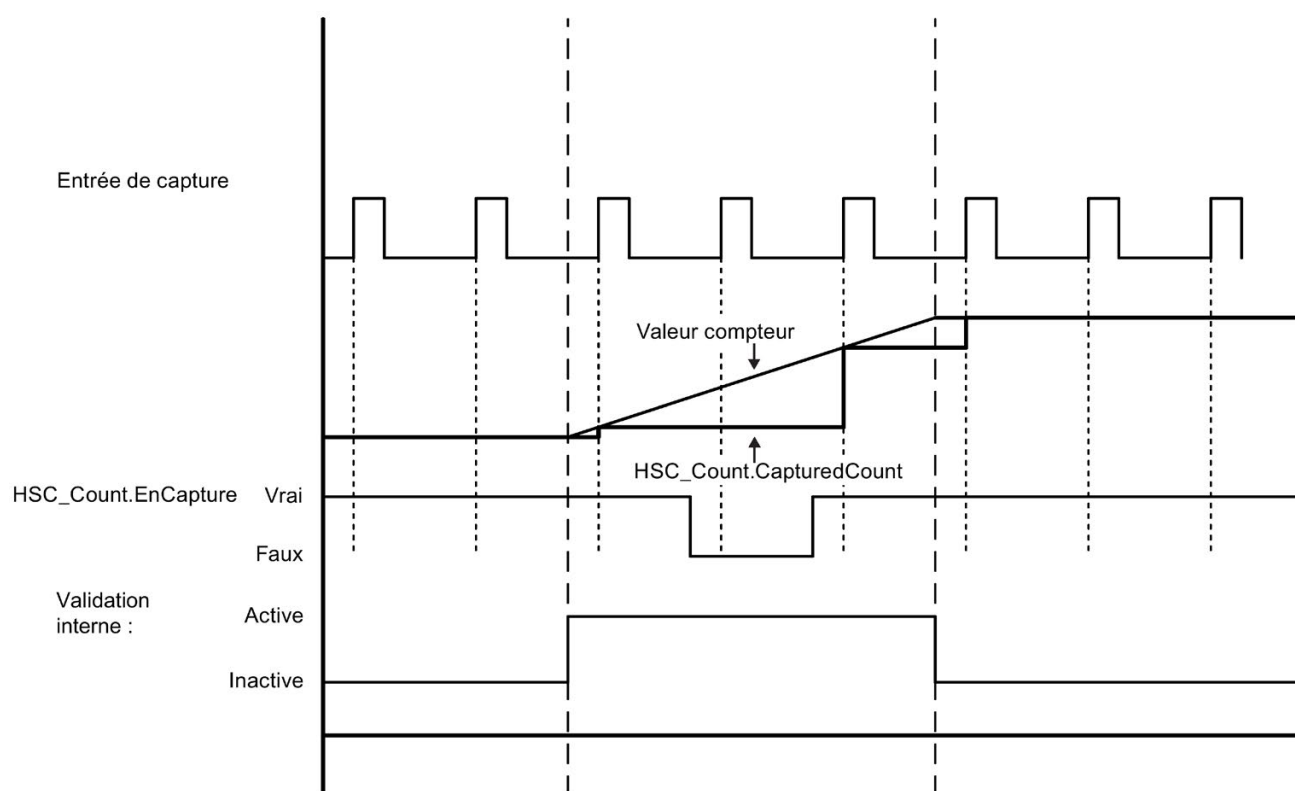
En mode Fréquence, la validation logicielle est commandée par "HSC_Frequency.EnHSC".

Pour plus d'informations sur la manière de configurer la fonction Validation, voir Fonctions d'entrée (Page 597).

10.1.2.3 Fonction Capture

Vous utilisez la fonction Capture pour sauvegarder la valeur en cours du compteur avec un signal de référence externe. Lorsqu'elle est configurée et validée par le bit "HSC_Count.EnCapture", la fonction Capture provoque l'acquisition de la valeur de comptage en cours à l'apparition d'un front d'entrée externe. La fonction Capture est effective quel que soit l'état de la validation interne. Le programme sauvegarde la valeur inchangée du compteur lorsque la validation se ferme. Après l'exécution de l'instruction CTRL_HSC_EXT, le programme mémorise la valeur capturée dans "HSC_Count.CapturedCount".

La figure ci-dessous montre un exemple de la fonction Capture configurée pour déclencher la capture sur un front montant. L'entrée de capture ne déclenche pas de capture de la valeur de comptage en cours lorsque le bit "HSC_Count.EnCapture" est mis à faux par le biais de l'instruction CTRL_HSC_EXT.



Remarque

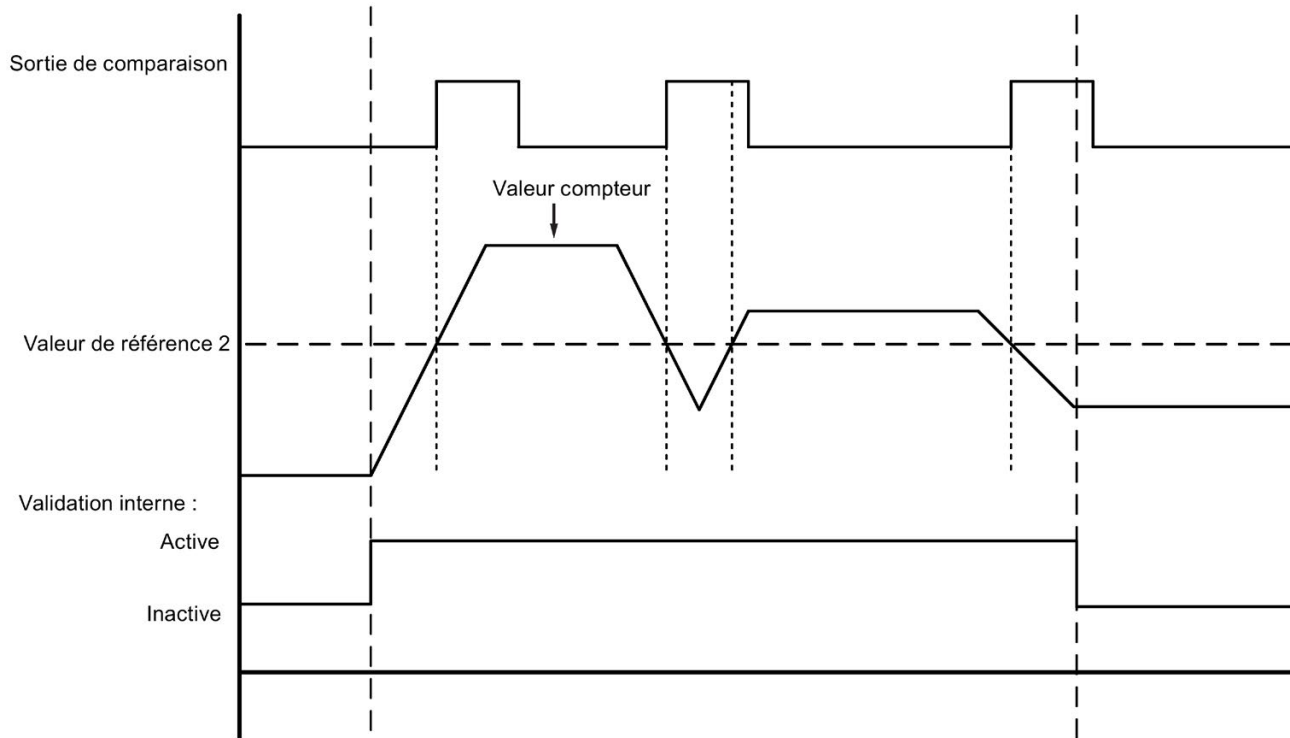
Les filtres d'entrée configurés retardent le signal de commande de l'entrée TOR.

Cette fonction d'entrée ne peut être utilisée que lorsque le compteur rapide est configuré pour le mode Comptage.

Pour plus d'informations sur la manière de configurer la fonction Capture, voir Fonctions d'entrée (Page 597).

10.1.2.4 Fonction Comparaison

Lorsqu'elle est activée, la fonction de sortie Comparaison génère une impulsion unique configurable à chaque fois que l'événement configuré se produit. On dispose des événements "valeur de comptage égale à l'une des valeurs de référence" et "débordement du compteur". Si une impulsion est en cours et que l'événement se produit à nouveau, aucune impulsion n'est générée pour cet événement.



Remarque

Cette fonction de sortie ne peut être utilisée que lorsque le compteur rapide est configuré pour le mode Comptage.

Pour plus d'informations sur la manière de configurer la fonction Comparaison, voir Fonctions de sortie (Page 598).

10.1.2.5 Applications

Une application typique du compteur rapide est son utilisation pour surveiller la réponse d'un codeur angulaire incrémental. Le codeur angulaire fournit un nombre défini de valeurs de comptage par tour que vous pouvez utiliser comme entrée de générateur d'horloge pour le compteur rapide. Il fournit également une impulsion de réinitialisation qui se produit une fois par tour et que vous pouvez utiliser comme entrée de synchronisation pour le compteur rapide.

Pour commencer, le programme charge la première valeur de référence dans le compteur rapide et active l'état initial des sorties. Les sorties restent dans cet état pour la durée où la valeur de comptage en cours est inférieure à la valeur de référence. Le compteur rapide fournit une alarme lorsque la valeur en cours est égale à la valeur de référence, lorsque l'événement de synchronisation (réinitialisation) se produit et lorsque le sens de comptage change.

À chaque fois que la valeur du compteur égale la valeur de référence, un événement d'alarme se produit et le programme charge une nouvelle valeur de référence dans le compteur rapide et active l'état suivant des sorties. Lorsque l'événement d'alarme Synchronisation se produit, le programme active la première valeur de référence ainsi que le premier état des sorties et le cycle se répète.

Comme les alarmes se produisent à une fréquence bien inférieure à la vitesse de comptage du compteur rapide, vous pouvez réaliser une commande précise des opérations rapides qui a un impact relativement faible sur le cycle de la CPU. La méthode d'association des alarmes à des programmes d'alarme permet d'effectuer chaque chargement d'une nouvelle valeur prédéfinie dans un sous-programme distinct, ce qui facilite la commande de l'état. Mais vous pouvez aussi traiter tous les événements d'alarme dans un sous-programme unique.

La fonction Validation, déclenchée soit par le programme utilisateur soit par un signal d'entrée externe, peut désactiver le comptage des impulsions du codeur. Vous pouvez ne pas tenir compte des mouvements de l'arbre en désactivant la validation. Cela signifie que, alors que le codeur continue à envoyer des impulsions au compteur rapide, la valeur de comptage est maintenue à la dernière valeur en vigueur avant la désactivation de la validation. Lorsque la validation redevient active, le comptage reprend à partir de la dernière valeur en vigueur avant la désactivation de la validation.

Lorsqu'elle est activée, la fonction Capture provoque l'acquisition de la valeur de comptage en cours à l'apparition d'une entrée externe. Un processus, un programme de calibration par exemple, peut utiliser cette fonction pour déterminer combien d'impulsions se produisent entre des événements.

Lorsqu'elle est activée, la fonction de sortie Comparaison génère une impulsion unique configurable à chaque fois que la valeur de comptage en cours atteint l'une des valeurs de référence ou qu'un débordement se produit (dépassement des limites de comptage). Vous pouvez utiliser cette impulsion comme signal pour déclencher un autre processus à chaque fois qu'un événement HSC particulier se produit.

Le sens de comptage est commandé par le programme utilisateur ou par un signal d'entrée externe.

Pour obtenir la vitesse de l'arbre rotatif, vous pouvez configurer le compteur rapide en mode Fréquence. Cette fonction fournit une valeur entière signée en Hz. Comme le signal de réinitialisation se produit une fois par tour, mesurer la fréquence du signal de réinitialisation donne une indication rapide de la vitesse de l'arbre, en nombre de tours par seconde.

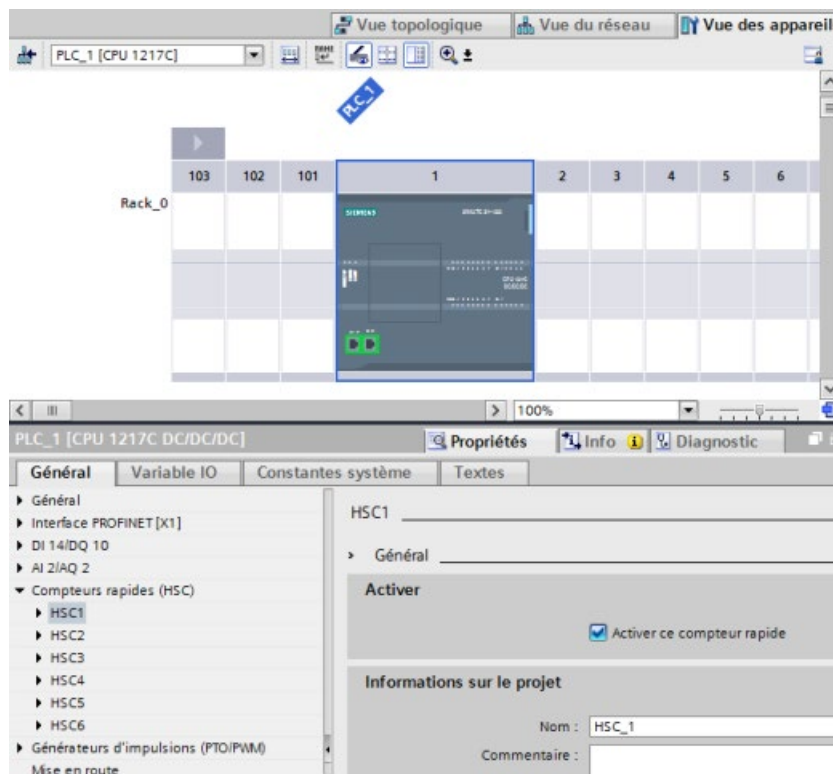
Si vous désirez une valeur en virgule flottante de la fréquence, configurez le compteur rapide en mode Période. Vous pouvez utiliser les valeurs ElapsedTime et EdgeCount renvoyées en mode Période pour calculer la fréquence.

10.1.3 Configuration d'un compteur rapide

Pour configurer le compteur rapide (HSC) :

- Sélectionnez la configuration d'appareil dans le navigateur du projet.
- Sélectionnez la CPU que vous voulez configurer.
- Cliquez sur l'onglet Propriétés de la fenêtre d'inspection (voir la figure ci-dessous).
- Sélectionnez le compteur rapide à activer dans la liste affichée dans l'onglet Général (voir la figure ci-dessous).

Vous pouvez configurer jusqu'à six compteurs rapides (HSC1 à HSC6). Pour activer un HSC, sélectionnez l'option "Activer ce compteur rapide". Lorsqu'un compteur rapide est activé, STEP 7 lui affecte un nom par défaut unique. Vous pouvez modifier ce nom en l'éditant dans le champ d'édition "Nom", mais il doit rester unique. Les noms des compteurs rapides activés deviennent des variables de type de données "HW_Hsc" dans la table de variables "Constantes système" et sont disponibles pour utilisation comme paramètre "HSC" des instructions CTRL_HSC_EXT. Pour plus d'informations, voir "Configuration du fonctionnement de la CPU (Page 177)".



Après l'activation du compteur rapide, STEP 7 définit le comptage monophasé comme configuration par défaut. Une fois que vous avez défini le filtre d'entrée TOR pour l'entrée générateur d'horloge du compteur rapide, vous pouvez charger le programme dans l'API et la CPU est prête à compter. Pour modifier la configuration du compteur rapide, passez à la section suivante, "Type de comptage".

Le tableau suivant donne une vue d'ensemble des entrées et sorties disponibles pour chaque configuration :

Tableau 10- 8 Modes de comptage des compteurs rapides

Type	Entrée 1	Entrée 2	Entrée 3	Entrée 4	Entrée 5	Sortie 1	Fonction
Monophasé avec gestion interne du sens de comptage	Horloge	-	-	-	-	-	Comptage, fréquence ou période
			Synchronisation	Validation	Capture	Comparaison	Comptage
Monophasé avec gestion externe du sens de comptage	Horloge	Sens	-	-	-	-	Comptage, fréquence ou période
			Synchronisation	Validation	Capture	Comparaison	Comptage
Biphase	Horloge, incrémentation	Horloge, décrémentation	-	-	-	-	Comptage, fréquence ou période
			Synchronisation	Validation	Capture	Comparaison	Comptage
Compteur A/B	Phase A	Phase B	-	-	-	-	Comptage, fréquence ou période
			Synchronisation ¹	Validation	Capture	Comparaison	Comptage
Compteur A/B quadruple	Phase A	Phase B	-	-	-	-	Comptage, fréquence ou période
			Synchronisation ¹	Validation	Capture	Comparaison	Comptage

¹ Pour un codeur : phase Z, référence

10.1.3.1 Type de comptage

Il existe quatre types ou modes de comptage. Lorsque vous changez le mode d'un compteur rapide, les options de configuration disponibles changent également :

- **Comptage** : Compte le nombre d'impulsions et incrémente ou décrémente la valeur de comptage en fonction de l'état de la gestion du sens. Une E/S externe peut réinitialiser la valeur de comptage, désactiver le comptage, déclencher une capture de la valeur de comptage en cours et générer une impulsion unique à l'apparition d'un événement défini. Les valeurs de sortie sont la valeur de comptage en cours et la valeur de comptage au moment où l'événement de capture s'est produit.
- **Période** : Compte le nombre d'impulsions d'entrée pendant un intervalle de temps indiqué. Renvoie le nombre d'impulsions et la durée en nanosecondes (ns). Les valeurs sont capturées et calculées à la fin de l'intervalle de temps indiqué dans "Période de mesure de fréquence". Le mode Période est disponible pour l'instruction CTRL_HSC_EXT mais pas pour l'instruction CTRL_HSC.
- **Fréquence** : Mesure les impulsions d'entrée et la durée et calcule la fréquence des impulsions. Le programme renvoie la fréquence en Hz sous forme de double entier signé. La valeur est négative si le sens de comptage est décrémental. Les valeurs sont capturées et calculées à la fin de l'intervalle de temps indiqué dans "Période de mesure de fréquence".
- **Motion control** : Utilisé par l'objet technologique Motion Control et pas disponible pour les instructions HSC. Pour plus d'informations, voir "Motion Control (Page 674)".

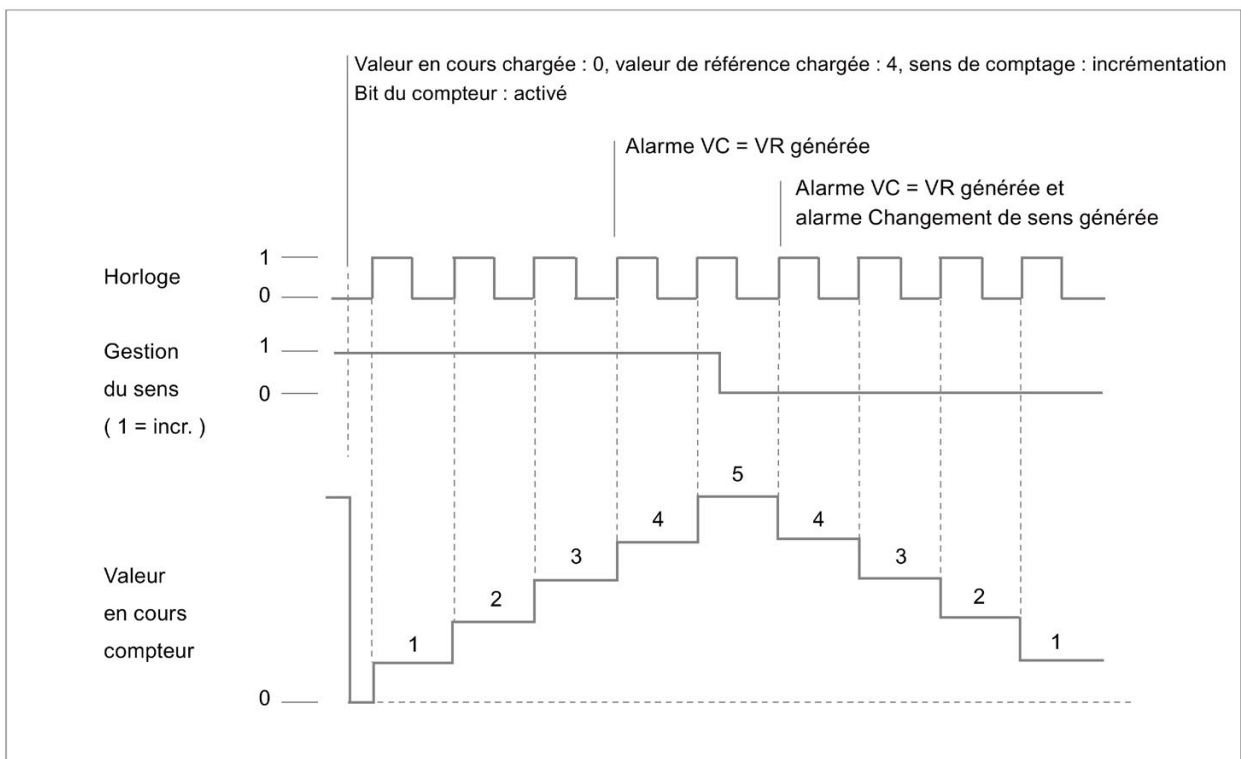
10.1.3.2 Phase de fonctionnement

Sélectionnez la phase de fonctionnement souhaitée pour le compteur rapide. Les quatre figures ci-après illustrent des changements de la valeur de comptage, l'événement Valeur en cours (VC) égale à la valeur de référence (VR) et l'événement Changement du sens de comptage.

Monophasé

Monophasé (pas disponible avec Motion Control) compte les impulsions :

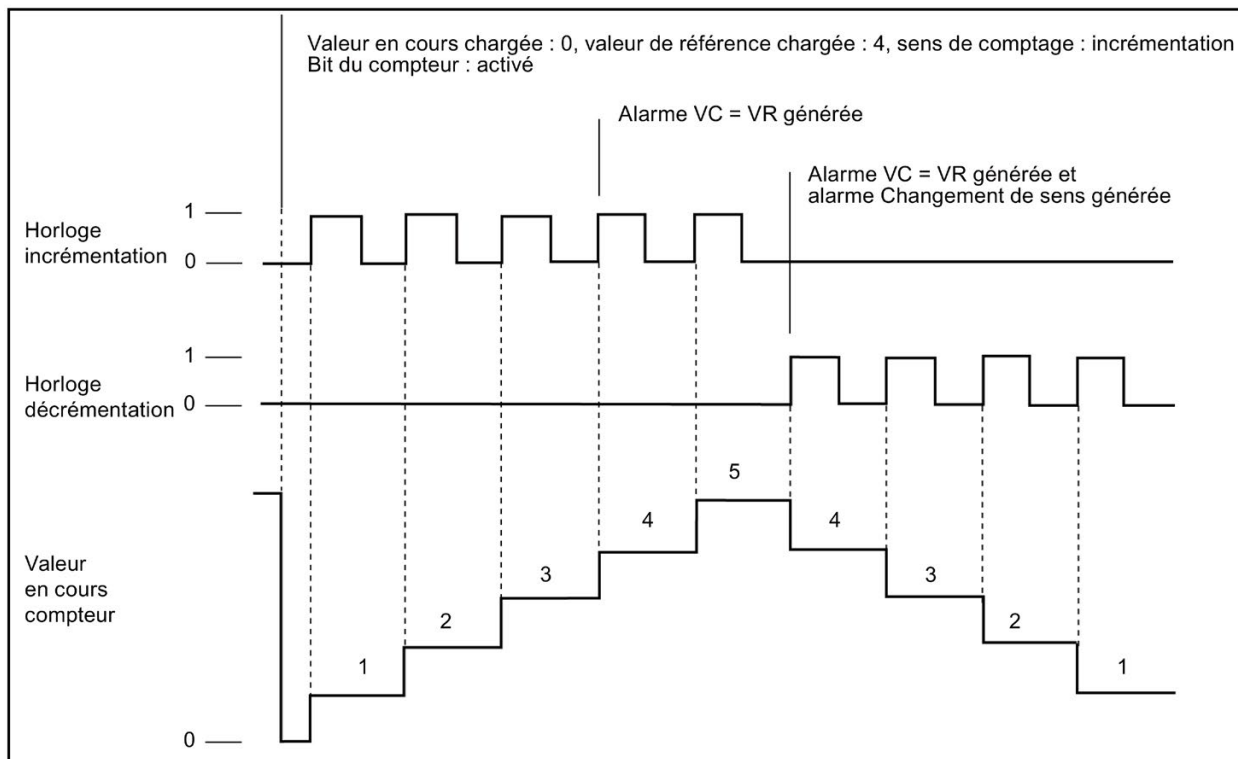
- Programme utilisateur (gestion interne du sens de comptage) :
 - 1 = incrémentation
 - -1 = décrémentation
- Entrée matérielle (gestion externe du sens de comptage) :
 - Niveau haut = incrémentation
 - Niveau bas = décrémentation



Biphase

Biphase :

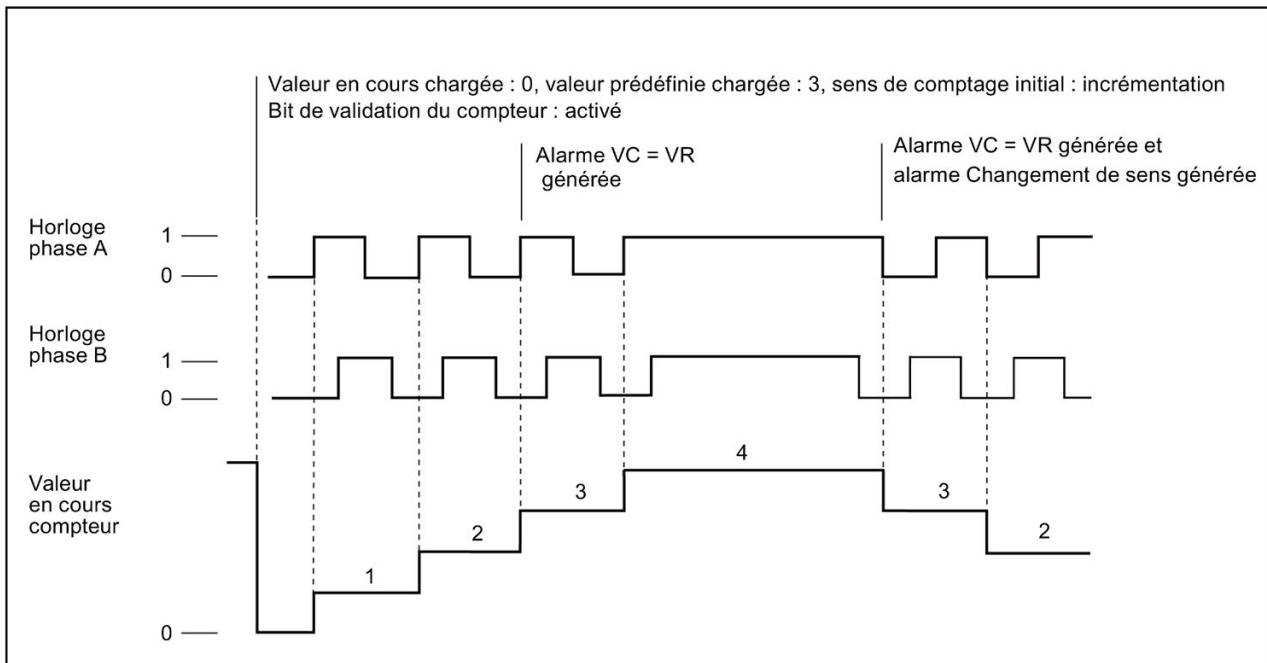
- incrémente sur l'entrée d'incrémentation de l'horloge
- décrémente sur l'entrée de décrémentation de l'horloge



Compteur A/B

Quadrature de phase A/B :

- incrémente en cas de front montant de l'entrée d'horloge A lorsque l'entrée d'horloge B est au niveau bas
- décrémente en cas de front descendant de l'entrée d'horloge A lorsque l'entrée d'horloge B est au niveau bas



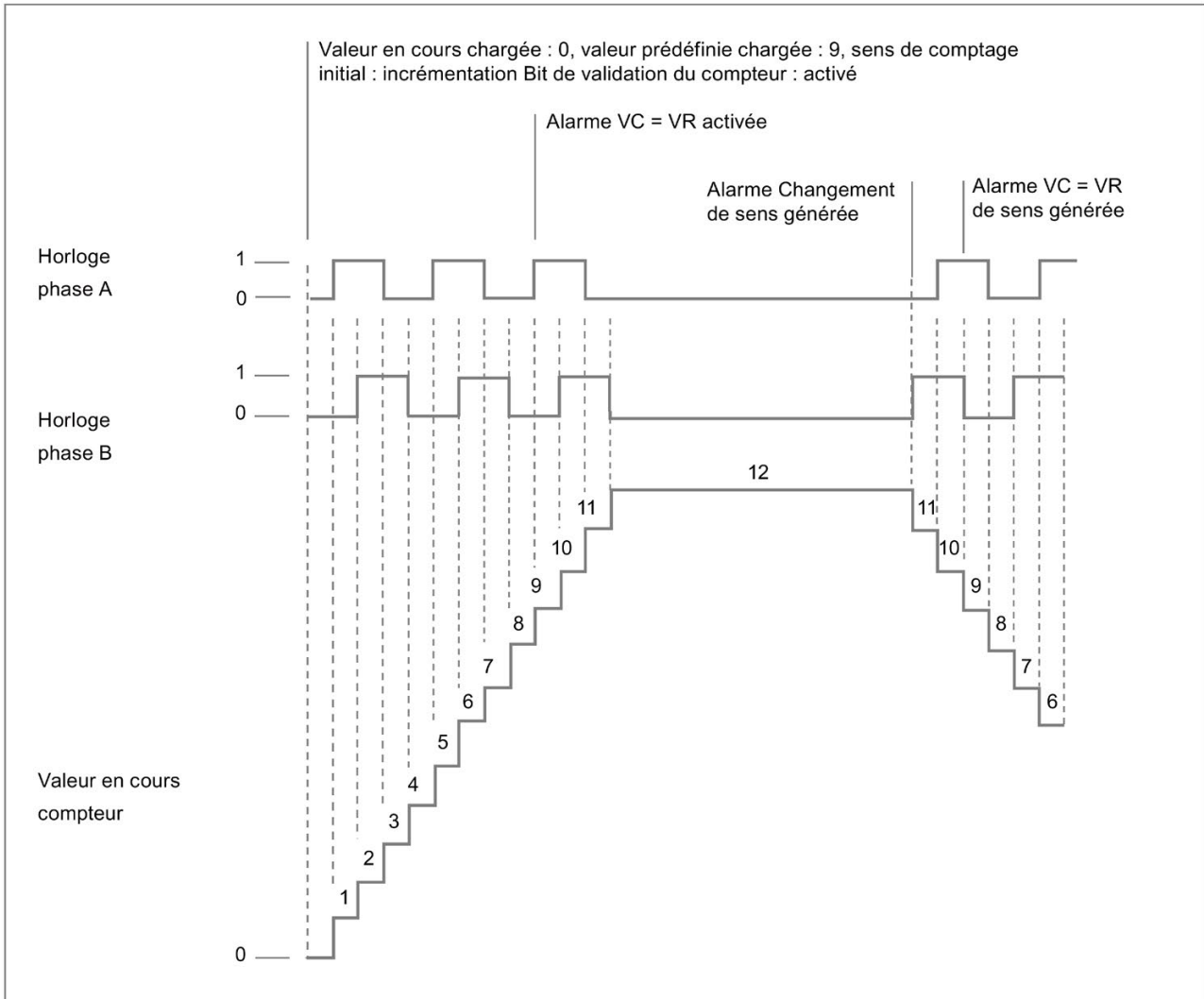
Compteur A/B quadruple

Quadrature de phase quadruple A/B :

- incrémente en cas de front montant de l'entrée d'horloge A lorsque l'entrée d'horloge B est au niveau bas
- incrémente en cas de front descendant de l'entrée d'horloge A lorsque l'entrée d'horloge B est au niveau haut
- incrémente en cas de front montant de l'entrée d'horloge B lorsque l'entrée d'horloge A est au niveau haut
- incrémente en cas de front descendant de l'entrée d'horloge B lorsque l'entrée d'horloge A est au niveau bas
- décrémente en cas de front montant de l'entrée d'horloge B lorsque l'entrée d'horloge A est au niveau bas
- décrémente en cas de front descendant de l'entrée d'horloge B lorsque l'entrée d'horloge A est au niveau haut

10.1 Comptage (compteurs rapides)

- décrémente en cas de front montant de l'entrée d'horloge A lorsque l'entrée d'horloge B est au niveau haut
- décrémente en cas de front descendant de l'entrée d'horloge A lorsque l'entrée d'horloge B est au niveau bas



10.1.3.3 Valeurs initiales

À chaque fois que la CPU passe à l'état MARCHE, elle charge les valeurs initiales. Les valeurs initiales sont uniquement utilisées en mode Comptage :

- Valeur de compteur initiale : Le programme donne à la valeur de comptage en cours la valeur de compteur initiale lorsque la CPU passe de l'état ARRÊT à l'état MARCHE ou que le programme déclenche l'entrée de synchronisation.
- Valeur de référence initiale : Lorsque la valeur de comptage en cours atteint la valeur de référence, et si les fonctions correspondantes sont configurées, le programme génère une alarme et/ou une impulsion de sortie.
- Valeur de référence initiale 2 : Lorsque la valeur de comptage en cours atteint la valeur de référence 2, et si la fonction est configurée, le programme génère une impulsion de sortie.
- Limite supérieure initiale : valeur de comptage maximale. La valeur par défaut est la plus grande valeur possible : +2,147,483,647 impulsions.
- Limite inférieure initiale : valeur de comptage minimale. La valeur par défaut est la plus petite valeur possible : -2,147,483,648 impulsions.

Les valeurs ci-dessus et le comportement du compteur quand il atteint une limite sont uniquement disponibles en mode Comptage. Vous pouvez ajuster ces valeurs et le comportement du compteur avec le SDT HSC_Count dans l'instruction CTRL_HSC_EXT.

10.1.3.4 Fonctions d'entrée

Les entrées Horloge et Sens déterminent les événements de comptage et le sens sur la base de la phase de fonctionnement. Vous pouvez utiliser les entrées de synchronisation, de capture et de validation uniquement en mode Comptage et vous pouvez activer et configurer ces entrées individuellement pour différents types de déclenchement.

Entrée de synchronisation

L'entrée de synchronisation (Sync) règle la valeur de comptage en cours à la valeur (de compteur) initiale. Vous utiliserez l'entrée de synchronisation typiquement pour remettre le compteur à 0. Vous pouvez déclencher la synchronisation lorsque la broche d'entrée est dans l'un des états suivants :

- Haut
- Bas
- Transition de bas à haut
- Transition de haut à bas
- Transition de haut à bas ou de bas à haut

Entrée de capture

L'entrée de capture donne à la valeur de comptage capturée la valeur de comptage sauvegardée au moment où vous avez déclenché l'entrée de capture. Vous pouvez déclencher la capture lorsque la broche d'entrée est dans l'un des états suivants :

- Transition de bas à haut
- Transition de haut à bas
- Transition de haut à bas ou de bas à haut

Entrée de validation

L'entrée de validation arrête le comptage rapide. Vous pouvez déclencher la validation lorsque la broche d'entrée est dans l'un des états suivants :

- Haut
- Bas

10.1.3.5 Fonction de sortie

La fonction de sortie Comparaison est l'unique sortie pour le compteur rapide et est uniquement disponible en mode Comptage.

Sortie de comparaison

Vous pouvez configurer la sortie de comparaison afin qu'elle génère une impulsion unique à l'apparition de l'un des événements suivants :

- Compteur égal à valeur de référence (sens de comptage incrémental)
- Compteur égal à valeur de référence (sens de comptage décrémental)
- Compteur égal à valeur de référence (sens de comptage incrémental ou décrémental)
- Compteur égal à valeur de référence 2 (sens de comptage incrémental)
- Compteur égal à valeur de référence 2 (sens de comptage décrémental)
- Compteur égal à valeur de référence 2 (sens de comptage incrémental ou décrémental)
- Débordement positif
- Débordement négatif

Vous pouvez configurer l'impulsion de sortie avec une période comprise entre 1 et 500 ms. La période par défaut est 10 ms. Vous pouvez définir la largeur d'impulsion (ou rapport cyclique) à une valeur quelconque comprise entre 1 % et 100 %. La largeur d'impulsion par défaut est 50 %.

Si plusieurs événements de sortie de comparaison se produisent pendant la période indiquée, les impulsions de sortie de ces événements sont perdues du fait que l'impulsion en cours n'avait pas encore achevé son cycle. Une fois l'impulsion achevée (la période configurée s'est écoulée), le générateur d'impulsions est disponible pour produire une nouvelle impulsion.

10.1.3.6 Événements d'alarme

Dans la section de configuration Événement, vous pouvez sélectionner un OB d'alarme de processus dans la liste déroulante (ou créer un nouvel OB) que vous associez à un événement HSC. La priorité de cette alarme est comprise entre 2 et 26, 2 étant la priorité la plus faible et 26 la priorité la plus élevée. Selon la configuration du compteur rapide, les événements suivants sont disponibles :

- Événement Valeur du compteur égale à valeur de référence : Cet événement se produit lorsque la valeur de comptage du compteur rapide atteint la valeur de référence. Vous pouvez définir la valeur de référence dans la section Valeur de référence initiale pendant la configuration ou au moyen de l'instruction CTRL_HSC_EXT en actualisant "NewReference1". Pour plus d'informations, voir le paragraphe "Phase de fonctionnement (Page 593)".
- Événement Synchronisation : Cet événement se produit à chaque fois que vous activez et déclenchez l'entrée de synchronisation.
- Événement Changement de sens : Cet événement se produit lorsque le sens de comptage change. Pour plus d'informations, voir le paragraphe "Phase de fonctionnement (Page 593)".

10.1.3.7 Affectation d'entrée matérielle


Pour chaque entrée de compteur rapide que vous activez, sélectionnez l'entrée individuelle désirée soit dans la CPU, soit dans le Signal Board optionnel (les modules de communication et d'entrées-sorties n'acceptent pas les entrées HSC). Lorsque vous sélectionnez une entrée, STEP 7 affiche la valeur de fréquence maximale à côté de l'entrée sélectionnée. Il peut s'avérer nécessaire d'ajuster les paramètres de filtre d'entrée TOR de manière à ce que toutes les fréquences de signal valides puissent passer le filtre. La configuration des filtres d'entrée HSC est décrite au paragraphe "Configuration des temps de filtre d'entrée TOR (Page 179)".

Remarque

Les voies d'entrée des CPU et des SB (firmware V4 ou ultérieur) ont des valeurs de temps de filtre d'entrée configurables.

Les versions plus anciennes du firmware avaient des voies d'entrée HSC fixes et des temps de filtre fixes qui ne pouvaient pas être modifiés.

Depuis la version V4, il est possible de définir les voies d'entrée et les temps de filtre. La valeur par défaut pour le filtre d'entrée est de 6,4 ms, ce qui limite la fréquence de comptage maximale à 78 Hz. Vous pouvez modifier la valeur du filtre pour compter à des fréquences supérieures ou inférieures en fonction de la conception de votre système.

 ATTENTION
<p>Risques liés à la modification du temps de filtre pour des voies d'entrée TOR</p> <p>Si le temps de filtre pour une voie d'entrée TOR est modifié par rapport au paramétrage précédent, une nouvelle valeur d'entrée de niveau "0" peut devoir être présente pendant une durée cumulée allant jusqu'à 20,0 ms pour que le filtre réagisse pleinement aux nouvelles entrées. Pendant ce temps, les événements d'impulsion "0" courts de moins de 20,0 ms peuvent ne pas être détectés ni comptés.</p> <p>La modification du temps de filtre peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.</p> <p>Pour garantir la prise en compte immédiate d'un nouveau temps de filtre, mettez la CPU hors tension puis sous tension.</p>

Utilisez le tableau suivant pour vous assurer que les voies d'entrée de la CPU et du SB que vous connectez peuvent prendre en charge les vitesses d'impulsions maximales dans vos signaux du processus :

Tableau 10- 9 Entrée de CPU : fréquence maximale

CPU	Voix d'entrée de CPU	Phase de fonctionnement : monophasé ou biphasé	Phase de fonctionnement : compteur A/B ou compteur A/B quadruple
1211C	la.0 à la.5	100 kHz	80 kHz
1212C	la.0 à la.5	100 kHz	80 kHz
	la.6, la.7	30 kHz	20 kHz
1214C et 1215C	la.0 à la.5	100 kHz	80 kHz
	la.6 à lb.5	30 kHz	20 kHz
1217C	la.0 à la.5	100 kHz	80 kHz
	la.6 à lb.1	30 kHz	20 kHz
	lb.2 à lb.5 (.2+, .2- à .5+, .5-)	1 MHz	1 MHz

Tableau 10- 10 Entrée de Signal Board : fréquence maximale (SB optionnel)

Signal Board (SB)	Voix d'entrée de SB	Phase de fonctionnement : monophasé ou biphasé	Phase de fonctionnement : compteur A/B ou compteur A/B quadruple
SB 1221, 200 kHz	le.0 à le.3	200 kHz	160 kHz
SB 1223, 200 kHz	le.0, le.1	200 kHz	160 kHz
SB 1223	le.0, le.1	30 kHz	20 kHz

Lors de l'affectation d'entrées aux fonctions HSC, vous pouvez affecter la même entrée à plusieurs fonctions HSC. Ainsi, affecter I0.3 à l'entrée de synchronisation de HSC1 et à l'entrée de synchronisation de HSC2 pour synchroniser la valeur de comptage de ces deux compteurs rapides au même moment constitue une configuration valide. La compilation génère toutefois un avertissement.

Lorsque c'est possible, évitez d'affecter plusieurs fonctions d'entrée du même compteur rapide à la même entrée. L'affectation de I0.3 à l'entrée de synchronisation et à l'entrée de validation de HSC1 pour synchroniser la valeur de comptage et désactiver le comptage en même temps, par exemple, constitue également une configuration valide. Vous pouvez procéder à cette configuration, mais elle pourrait produire des résultats inattendus.

ATTENTION

Risques liés à l'affectation de plusieurs fonctions à une voie d'entrée TOR unique

Affecter plusieurs fonctions d'entrée du même compteur rapide à une entrée commune peut produire des résultats imprévisibles. Lorsqu'une entrée à laquelle plusieurs fonctions sont affectées est déclenchée, il est impossible de savoir dans quel ordre les fonctions seront exécutées par l'API. C'est ce que l'on appelle une situation de concurrence, qui constitue souvent une situation indésirable.

Cette situation de concurrence peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Pour éviter une situation de concurrence, n'affectez pas plus de deux fonctions d'entrée du même compteur rapide à la même entrée. Si deux fonctions d'entrée d'un compteur rapide sont affectées à la même entrée, définissez les déclenchements de telle manière qu'ils ne puissent jamais se produire en même temps. Souvenez-vous qu'un front descendant se produit alors qu'un niveau bas commence et qu'un front montant se produit alors qu'un niveau haut commence.

Remarque

Vous affectez les entrées et sorties TOR utilisées par les compteurs rapides pendant la configuration d'appareil de la CPU. Lorsque vous affectez des entrées et des sorties à des compteurs rapides, vous ne pouvez pas en modifier la valeur au moyen de la fonction de forçage permanent dans une table de visualisation. C'est le compteur rapide qui a le contrôle absolu sur ces entrées et sorties.

10.1.3.8 Affectation d'une sortie matérielle

Si vous activez la sortie de comparaison, sélectionnez une sortie disponible. Une fois que vous avez configuré une sortie en vue de son utilisation par un compteur rapide (ou un autre objet technologique tel qu'un générateur d'impulsions), cette sortie est la propriété exclusive de cet objet. Aucun autre composant ne peut utiliser cette sortie et cette sortie ne peut pas être forcée à une certaine valeur. Si vous configurez une voie de sortie unique pour plusieurs compteurs rapides ou pour une utilisation par un compteur rapide et par une sortie d'impulsions, le programme génère une erreur de compilation.

10.1.3.9 Adresses de mémoire d'entrée des compteurs rapides

Chaque compteur rapide utilise un double mot en mémoire I pour stocker la valeur de comptage en cours. Si vous configurez le compteur rapide en mode fréquence, la fréquence est stockée dans cette adresse de mémoire d'entrée. La plage d'adresse d'entrée disponible va de I0.0 à I1023.7 (l'adresse de début maximum est I1020.0). Le compteur rapide ne peut pas utiliser une adresse d'entrée qui chevauche une adresse d'entrée allouée à un autre composant. Pour plus d'informations sur la mémoire image, voir "Exécution du programme utilisateur (Page 85)".

Le tableau suivant montre les adresses par défaut affectées à chaque HSC :

Tableau 10- 11 Adresses par défaut des compteurs rapides

Compteur rapide (HSC)	Type de données de valeur en cours	Adresse par défaut de la valeur en cours
HSC1	DInt	ID 1000
HSC2	DInt	ID 1004
HSC3	DInt	ID 1008
HSC4	DInt	ID 1012
HSC5	DInt	ID 1016
HSC6	DInt	ID 1020

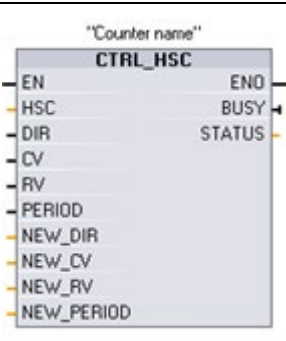
10.1.3.10 Identification matérielle

Chaque compteur rapide a une identification matérielle unique qui est utilisée par les instructions HSC_CTRL et HSC_CTRL_EXT. Vous trouverez la variable API pour l'identification matérielle sous "Constantes système". Un compteur rapide de nom "HSC_1" a la variable "Local-HSC_1" et le type de données "Hw_Hsc". Cette variable s'affiche également dans la liste déroulante pour la sélection de l'entrée HSC des instructions CTRL_HSC_EXT.

10.1.4 Instruction CTRL_HSC (Commander le compteur rapide) héritée

10.1.4.1 Présentation de l'instruction

Tableau 10- 12 Instruction CTRL_HSC (à des fins générales de comptage)

CONT / LOG	SCL	Description
	<pre>"CTRL_HSC_1_DB" (hsc:=W#16#0, dir:=False, cv:=False, rv:=False, period:=False, new_dir:=0, new_cv:=L#0, new_rv:=L#0, new_period:=0, busy=>_bool_out_, status=>_word_out_);</pre>	<p>Chaque instruction CTRL_HSC (Commande de compteurs rapides) utilise une structure sauvegardée dans un DB afin de conserver les données de compteur. Vous affectez le DB lors du placement de l'instruction CTRL_HSC dans l'éditeur.</p>

- 1 Lorsque vous insérez l'instruction, STEP 7 affiche la boîte de dialogue "Options d'appel" pour créer le DB associé.
- 2 Dans l'exemple SCL, "CTRL_HSC_1_DB" est le nom du DB d'instance.

Tableau 10- 13 Types de données pour les paramètres

Paramètre	Déclaration	Type de données	Description
HSC	IN	HW_HSC	Identificateur de compteur rapide
DIR ^{1, 2}	IN	Bool	1 = Demande de nouveau sens de comptage
CV ¹	IN	Bool	1 = Demande de paramétrage d'une nouvelle valeur de compteur
RV ¹	IN	Bool	1= Demande de paramétrage d'une nouvelle valeur de référence
PERIOD ¹	IN	Bool	1 = Demande de paramétrage d'une nouvelle valeur de période (uniquement pour le mode de mesure de fréquence)
NEW_DIR	IN	Int	Nouveau sens de comptage : 1= incrémentation, -1= décrémentation
NEW_CV	IN	DInt	Nouvelle valeur de compteur
NEW_RV	IN	DInt	Nouvelle valeur de référence
NEW_PERIOD	IN	Int	Nouvelle valeur de période en millisecondes (uniquement pour le mode de mesure de fréquence). Les seules valeurs autorisées sont 10, 100 ou 1 000 millisecondes : 1 000 = 1 seconde 100 = 0,1 seconde 10 = 0,01 seconde
BUSY ³	OUT	Bool	Fonction occupée
STATUS	OUT	Word	Code d'erreur d'exécution

- ¹ Si vous ne demandez pas l'actualisation d'une valeur de paramètre, les valeurs d'entrée correspondantes ne sont pas prises en compte.
- ² Le paramètre DIR n'est valide que si le sens de comptage configuré a la valeur "Programme utilisat. (commande de sens interne)". Vous déterminez comment utiliser ce paramètre dans la configuration d'appareil HSC.
- ³ Pour un compteur rapide sur la CPU ou le Signal Board, le paramètre BUSY a toujours la valeur 0.

Vous configurez les paramètres pour chaque HSC dans la configuration d'appareil de la CPU pour la fonction de comptage/fréquence, les options de réinitialisation, la configuration d'événements d'alarme, les E/S matérielles et l'adresse des valeurs de comptage.

Certains des paramètres du HSC peuvent être modifiés par votre programme utilisateur pour fournir une commande par programme du processus de comptage :

- Définir le sens de comptage à une valeur NEW_DIR
- Définir la valeur de comptage en cours à une valeur NEW_CV
- Définir la valeur de référence à une valeur NEW_RV
- Définir la valeur de la période (pour le mode de mesure de fréquence) à une valeur NEW_PERIOD

Si les valeurs des mementos booléens suivants sont mises à 1 lors de l'exécution de l'instruction CTRL_HSC, la valeur NEW_xxx correspondante est chargée dans le compteur. Plusieurs demandes (plus d'un memento est mis à 1 au même moment) sont traitées lors d'une exécution unique de l'instruction CTRL_HSC.

- DIR = 1 constitue une demande de chargement d'une valeur NEW_DIR ; 0 = pas de modification
- CV = 1 constitue une demande de chargement d'une valeur NEW_CV ; 0 = pas de modification
- RV = 1 constitue une demande de chargement d'une valeur NEW_RV ; 0 = pas de modification
- PERIOD = 1 constitue une demande de chargement d'une valeur NEW_PERIOD ; 0 = pas de modification

Si une erreur se produit, ENO est mis à 0 et la sortie STATUS indique un code d'erreur :

Tableau 10- 14 Execution condition codes

STATUS (W#16#)	Description
0	Pas d'erreur
80A1	L'identificateur HSC n'accède pas à un compteur rapide
80B1	Valeur illicite dans NEW_DIR
80B2	Valeur illicite dans NEW_CV
80B3	Valeur illicite dans NEW_RV
80B4	Valeur illicite dans NEW_PERIOD
80C0	Plusieurs accès au compteur rapide Cette erreur peut se produire si le paramétrage du type de comptage (Page 592) est "Période" ou "Motion control". Ces types sont invalides pour l'instruction CTRL_HSC et ne sont pris en charge que par l'instruction CTRL_HSC_EXT.
80D0	Compteur rapide (HSC) non activé dans la configuration matérielle de la CPU

10.1.4.2 Utilisation de CTRL_HSC

L'instruction CTRL_HSC se place typiquement dans un OB d'alarme de processus qui est exécuté lorsque l'événement d'alarme de processus de compteur se produit. Par exemple, si un événement CV=RV déclenche l'alarme de compteur, un OB d'alarme de processus exécute l'instruction CTRL_HSC et peut modifier la valeur de référence en chargeant une valeur NEW_RV.

La valeur de comptage en cours n'est pas disponible dans les paramètres CTRL_HSC. L'adresse de mémoire image qui contient la valeur de comptage en cours est définie pendant la configuration matérielle du compteur rapide. Vous pouvez vous servir d'une logique de programme pour lire directement la valeur de comptage. La valeur renvoyée à votre programme correspondra au comptage correct à l'instant où le compteur a été lu. Le compteur continuera à compter les événements rapides. Ainsi, la valeur de comptage en cours pourrait changer avant que votre programme n'ait terminé un processus basé sur une ancienne valeur en cours.

10.1.4.3 Valeur de comptage en cours du compteur

La CPU sauvegarde la valeur en cours de chaque compteur rapide dans une adresse d'entrée (I). Le tableau suivant montre les adresses par défaut affectées à la valeur en cours pour chaque HSC. Vous pouvez modifier l'adresse I pour la valeur en cours en modifiant les propriétés de la CPU dans la configuration des appareils.

Les compteurs rapides utilisent une valeur DInt pour sauvegarder la valeur en cours. Une valeur de comptage DInt est comprise entre -2147483648 et +2147483647. Vous pouvez configurer les limites de plage à partir des CPU de firmware V4.2. Pour plus d'informations, voir "Valeurs initiales (Page 597)".

Le compteur saute de la valeur positive maximum à la valeur négative minimum lorsqu'il incrémente et de la valeur négative minimum à la valeur positive maximum lorsqu'il décrémente. La fréquence est renvoyée en hertz (par exemple, 123,4 Hz est renvoyé sous la forme 123).

Tableau 10- 15 Adresses par défaut des compteurs rapides

HSC	Type de données de valeur en cours	Adresse par défaut de la valeur en cours
HSC1	DInt	ID1000
HSC2	DInt	ID1004
HSC3	DInt	ID1008
HSC4	DInt	ID1012
HSC5	DInt	ID1016
HSC6	DInt	ID1020

10.2 Régulation PID

STEP 7 fournit les instructions PID suivantes pour la CPU S7-1200 :

- L'instruction PID_Compact sert à la régulation de processus industriels à grandeurs d'entrée et de sortie continues.
- L'instruction PID_3Step sert à la régulation de dispositifs actionnés par moteur, tels que des vannes, qui nécessitent des signaux discrets pour la commande d'ouverture et de fermeture.
- L'instruction PID_Temp offre un régulateur PID universel qui permet de gérer des exigences spécifiques en matière de régulation de la température.

Remarque

Les modifications que vous apportez à la configuration PID et que vous chargez à l'état MARCHE ne prennent effet qu'après un passage de la CPU de l'état ARRÊT à l'état MARCHE. Les modifications que vous apportez dans la boîte de dialogue des "paramètres PID" à l'aide de la "Commande de la valeur de début" prennent effet immédiatement.

Les trois instructions PID (PID_Compact, PID_3Step, et PID_Temp) peuvent calculer les actions P, I et D lors de la mise en route (si "l'optimisation préalable" est configurée). Vous pouvez également configurer l'instruction pour une "optimisation fine" qui vous permet d'optimiser les paramètres. Vous n'avez pas besoin de déterminer les paramètres manuellement.

Remarque

Exécutez l'instruction PID à intervalles constants de la période d'échantillonnage (de préférence dans un OB d'alarme cyclique).

Comme la boucle PID a besoin d'un certain temps pour réagir aux variations de la valeur de réglage, ne calculez pas la valeur de réglage à chaque cycle. N'exécutez pas l'instruction PID dans l'OB de cycle de programme principal (tel que l'OB 1).

La période d'échantillonnage de l'algorithme PID représente le temps entre deux calculs de la valeur de réglage. La valeur de réglage est calculée pendant l'autoréglage et arrondie à un multiple de la période. Toutes les autres fonctions de l'instruction PID sont exécutées à chaque appel.

Algorithme PID

Le régulateur PID (actions proportionnelle/intégrale/dérivée) mesure l'intervalle de temps entre deux appels, puis évalue les résultats pour surveiller la période d'échantillonnage. Une valeur moyenne de la période d'échantillonnage est générée à chaque changement de mode ainsi qu'à la mise en route initiale. Cette valeur sert de référence à la fonction de surveillance et est utilisée pour les calculs. La surveillance porte sur le temps de mesure actuel entre deux appels et sur la valeur moyenne de la période d'échantillonnage définie du régulateur.

La valeur de réglage pour le régulateur PID est constituée de trois actions :

- Action proportionnelle P : La valeur de réglage calculée par l'action P est proportionnelle à l'écart entre la consigne et la mesure.
- Action intégrale I : La valeur de réglage calculée par l'action I augmente proportionnellement à la durée de l'écart entre la consigne et la mesure pour finalement compenser l'écart.
- Action dérivée D : La valeur de réglage calculée par l'action D augmente avec l'accroissement de la vitesse de modification de l'écart entre la consigne et la mesure. La valeur de réglage est synchronisée sur la consigne le plus rapidement possible.

Le régulateur PID utilise la formule suivante pour calculer la valeur de réglage pour l'instruction PID_Compact.

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Valeur de réglage	x	Mesure
w	Consigne	s	Opérateur de Laplace
K _p	Gain proportionnel (action P)	a	Coefficient de retard de l'action dérivée (action D)
T _i	Temps d'intégration (action I)	b	Pondération de l'action proportionnelle (action P)
T _D	Temps de dérivation (action D)	c	Pondération de l'action dérivée (action D)

Le régulateur PID utilise la formule suivante pour calculer la valeur de réglage pour l'instruction PID_3Step.

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Valeur de réglage	x	Mesure
w	Consigne	s	Opérateur de Laplace
K _p	Gain proportionnel (action P)	a	Coefficient de retard de l'action dérivée (action D)
T _i	Temps d'intégration (action I)	b	Pondération de l'action proportionnelle (action P)
T _D	Temps de dérivation (action D)	c	Pondération de l'action dérivée (action D)

10.2.1 Insertion de l'instruction PID et de l'objet technologique

STEP 7 fournit deux instructions pour la régulation PID :

- L'instruction PID_Compact et son objet technologique associé fournissent un régulateur PID universel optimisable. L'objet technologique contient tous les paramètres pour la boucle de régulation.
- L'instruction PID_3Step et son objet technologique associé fournissent un régulateur PID avec des paramètres spécifiques pour les vannes commandées par moteur. L'objet technologique contient tous les paramètres pour la boucle de régulation. Le régulateur PID_3Step comporte deux sorties booléennes supplémentaires.

Une fois l'objet technologique créé, vous devez configurer les paramètres (Page 643). Vous ajustez également les paramètres d'autooptimisation ("optimisation préalable" à la mise en route ou "optimisation fine" manuelle) pour mettre en service le régulateur PID (Page 661).

Tableau 10- 16 Insertion de l'instruction PID et de l'objet technologique

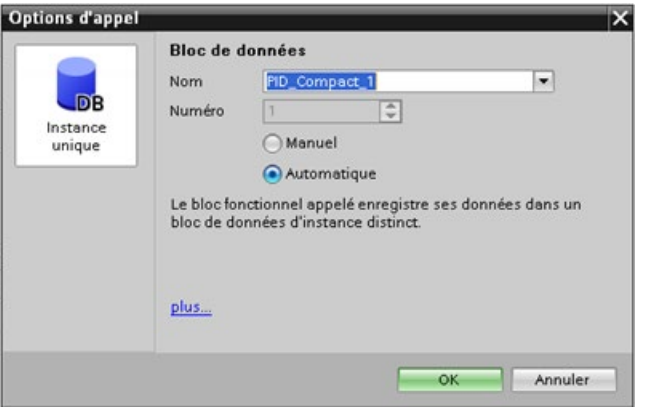
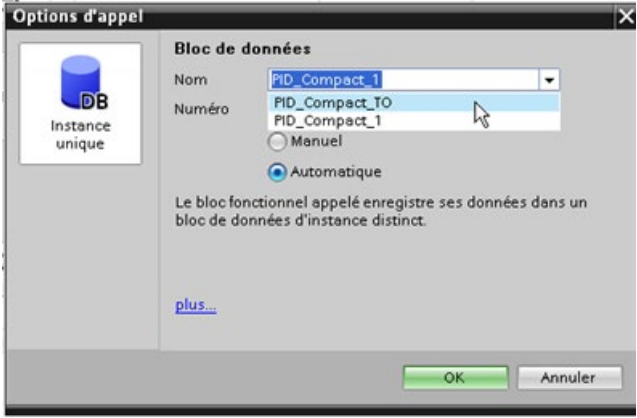
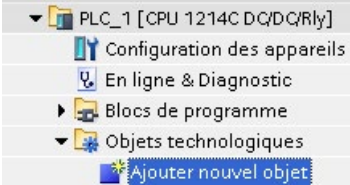
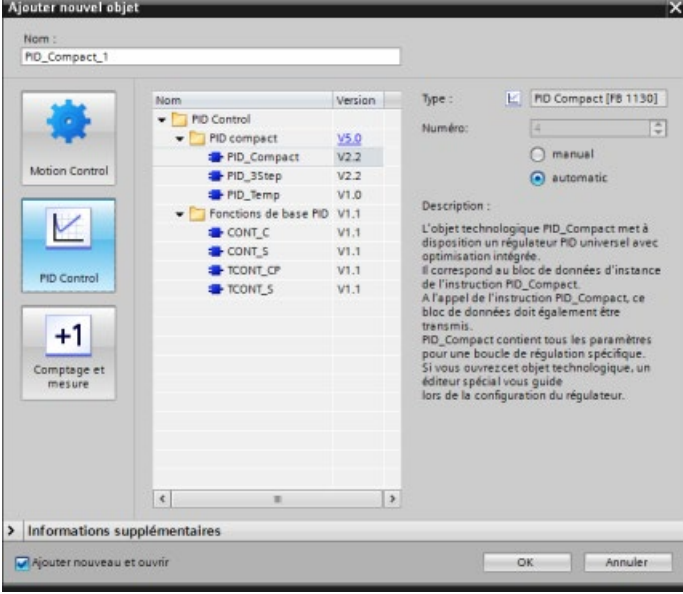
<p>Lorsque vous insérez une instruction PID dans votre programme utilisateur, STEP 7 crée automatiquement un objet technologique et un DB d'instance pour l'instruction. Le DB d'instance contient tous les paramètres utilisés par l'instruction PID. Chaque instruction PID doit avoir son propre DB d'instance unique pour fonctionner correctement.</p> <p>Après l'insertion de l'instruction PID et la création de l'objet technologique et du DB d'instance, vous configurez les paramètres pour l'objet technologique (Page 643).</p>	
--	---

Tableau 10- 17 (Facultatif) Création d'un objet technologique dans le navigateur du projet

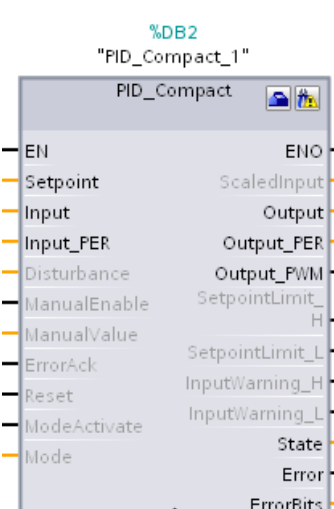
<p>Vous pouvez également créer des objets technologiques pour votre projet avant d'insérer l'instruction PID. Si vous avez créé l'objet technologique avant d'insérer une instruction PID dans votre programme utilisateur, vous pourrez sélectionner cet objet technologique à l'insertion de l'instruction PID.</p>	
<p>Pour créer un objet technologique, double-cliquez sur l'icône "Ajouter nouvel objet" dans le navigateur du projet.</p>	
<p>Cliquez sur l'icône "Régulation" et sélectionnez l'objet technologique correspondant au type de régulateur PID (PID_Compact ou PID_3Step). Vous pouvez créer un nom facultatif pour l'objet technologique. Cliquez sur "OK" pour créer l'objet technologique.</p>	

10.2.2 PID_Compact

10.2.2.1 Instruction PID_Compact

L'instruction PID_Compact fournit un régulateur PID universel avec fonction d'autooptimisation intégrée pour les modes automatique et manuel.

Tableau 10- 18 Instruction PID_Compact

CONT/LOG	SCL	Description
	<pre>"PID_Compact_1" (Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_word_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, ScaledInput=>_real_out_, Output=>_real_out_, Output_PER=>_word_out_, Output_PWM=>_bool_out_, SetpointLimit_H=>_bool_out_, SetpointLimit_L=>_bool_out_, InputWarning_H=>_bool_out_, InputWarning_L=>_bool_out_, State=>_int_out_, Error=>_bool_out_, ErrorBits=>_dword_out_);</pre>	<p>PID_Compact fournit un régulateur PID à fonction d'autooptimisation pour les modes automatique et manuel. PID_Compact est un régulateur PID T1 avec anti-emballement (anti-windup) et pondération des actions P et D.</p>

- STEP 7 crée automatiquement l'objet technologique et le DB d'instance lorsque vous insérez l'instruction. Le DB d'instance contient les paramètres de l'objet technologique.
- Dans l'exemple SCL, "PID_Compact_1" est le nom du DB d'instance.

Tableau 10- 19 Types de données pour les paramètres

Paramètre et type		Type de données	Description
Setpoint	IN	Real	Consigne du régulateur PID en mode automatique. (valeur par défaut : 0,0)
Input	IN	Real	Une variable du programme utilisateur est utilisée comme source pour la mesure. (valeur par défaut : 0,0) Si vous utilisez le paramètre Input alors il faut définir Config.InputPerOn = FALSE.
Input_PER	IN	Word	Une entrée analogique est utilisée comme source pour la mesure. (valeur par défaut : W#16#0) Si vous utilisez le paramètre Input_PER alors il faut définir Config.InputPerOn = TRUE.
Disturbance	IN	Real	Variable de perturbation ou valeur de pré-régulation
ManualEnable	IN	Bool	Active ou désactive le mode de fonctionnement manuel. (valeur par défaut : FALSE) : <ul style="list-style-type: none"> Un front de FALSE à TRUE active le mode manuel", tant que Etat = 4, le mode reste inchangé. Tant que ManualEnable est VRAI, vous ne pouvez pas modifier l'état de fonctionnement à l'aide d'un front montant pour ModeActivate ou utiliser la boîte de dialogue de mise en service. <ul style="list-style-type: none"> Un front TRUE à FALSE active l'état de fonctionnement qui est affecté par Mode. Note : Nous vous recommandons de modifier l'état de fonctionnement à l'aide de ModeActivate uniquement.
ManualValue	IN	Real	Valeur de sortie pour le mode manuel. (valeur par défaut : 0,0) Vous pouvez utiliser des valeurs à partir Config.OutputLowerLimit à Config.OutputUpperLimit.
ErrorAck	IN	Bool	Réinitialise le ErrorBits et les messages d'alerte. Front de FALSE à TRUE
Reset	IN	Bool	Redémarre le régulateur. (valeur par défaut : FALSE) : <ul style="list-style-type: none"> Front de FALSE à TRUE : <ul style="list-style-type: none"> Passe en mode "inactif" Réinitialise les ErrorBits et les sorties d'alerte Efface l'action intégrale Maintient les paramètres PID Tant que Reset à la valeur TRUE, PID_Compact reste en mode "Inactif" (Etat = 0). Front de TRUE à FALSE : <ul style="list-style-type: none"> PID_Compact passe dans l'état de fonctionnement qui est enregistré dans le paramètre Mode.
ModeActivate	IN	Bool	Le PID_Compact passe dans l'état de fonctionnement qui est enregistré dans le paramètreMode. Front de FALSE à TRUE :
Mode	IN	Int	Le mode PID souhaité ; Activé sur le front montant de l'entrée Mode Activate .
ScaledInput	OUT	Real	Mesure mise à l'échelle. (valeur par défaut : 0,0)

Paramètre et type		Type de données	Description
Output ¹	OUT	Real	Valeur de sortie pour le mode REEL. (valeur par défaut : 0,0)
Output_PER ¹	OUT	Word	Valeur de réglage analogique. (valeur par défaut : W#16#0)
Output_PWM ¹	OUT	Bool	Valeur de réglage pour la modulation de largeur d'impulsion. (valeur par défaut : FALSE) Les durées Marche et Arrêt forment la valeur de réglage.
SetpointLimit_H	OUT	Bool	Limite supérieure de la consigne. (valeur par défaut : FALSE) Si SetpointLimit_H a la valeur TRUE, la limite supérieure absolue de la consigne est atteinte. (Setpoint \geq Config.SetpointUpperLimit). La consigne est limitée à Config.SetpointUpperLimit.
SetpointLimit_L	OUT	Bool	Limite inférieure de la consigne. (valeur par défaut : FALSE) Si SetpointLimit_L a la valeur TRUE, la limite inférieure absolue de la consigne est atteinte. (Setpoint \leq Config.SetpointLowerLimit). La consigne est limitée à Config.SetpointLowerLimit.
InputWarning_H	OUT	Bool	Si InputWarning_H a la valeur TRUE, la mesure a atteint ou dépassé la limite d'alerte supérieure. (valeur par défaut : FALSE)
InputWarning_L	OUT	Bool	Si InputWarning_L a la valeur TRUE, la mesure de processus a atteint ou est tombée en dessous de la limite d'alerte inférieure. (valeur par défaut : FALSE)
State	OUT	Int	Mode de fonctionnement en cours du régulateur PID. (valeur par défaut : 0) Vous pouvez modifier l'état de fonctionnement à l'aide du paramètre d'entrée Mode et un front montant pour ModeActivate : <ul style="list-style-type: none"> • State = 0 : Inactif • State = 1 : Optimisation préalable • State = 2 : Optimisation fine manuelle • State = 3 : Mode automatique • State = 4 : Mode manuel • State = 5 : Valeur de réglage de remplacement avec surveillance d'erreur
Error	OUT	Bool	Si Error a la valeur TRUE, au moins un message d'erreur est en attente dans ce cycle. (valeur par défaut : FALSE) Note : Le paramètre Error dans V1.x PID était le champ ErrorBits qui contenait les codes d'erreurs. C'est maintenant une signalisation booléenne qui indique qu'une erreur est survenue.
ErrorBits	OUT	DWord	Le tableau des PID_Compact paramètres de l'instruction ErrorBits présente les messages d'erreur (Page 616) qui sont en attente. (valeur par défaut : DW#16#0000 (aucune erreur)). ErrorBits est rémanent et réinitialisé sur un front montant pour Reset ou ErrorAck. Note : Dans V1.x, le paramètre ErrorBits a été défini comme le paramètre Error et n'existait pas.

¹ Vous pouvez utiliser les sorties de la Output, Output_PER et Output_PWM des paramètres en parallèle.

Fonctionnement du régulateur PID_Compact

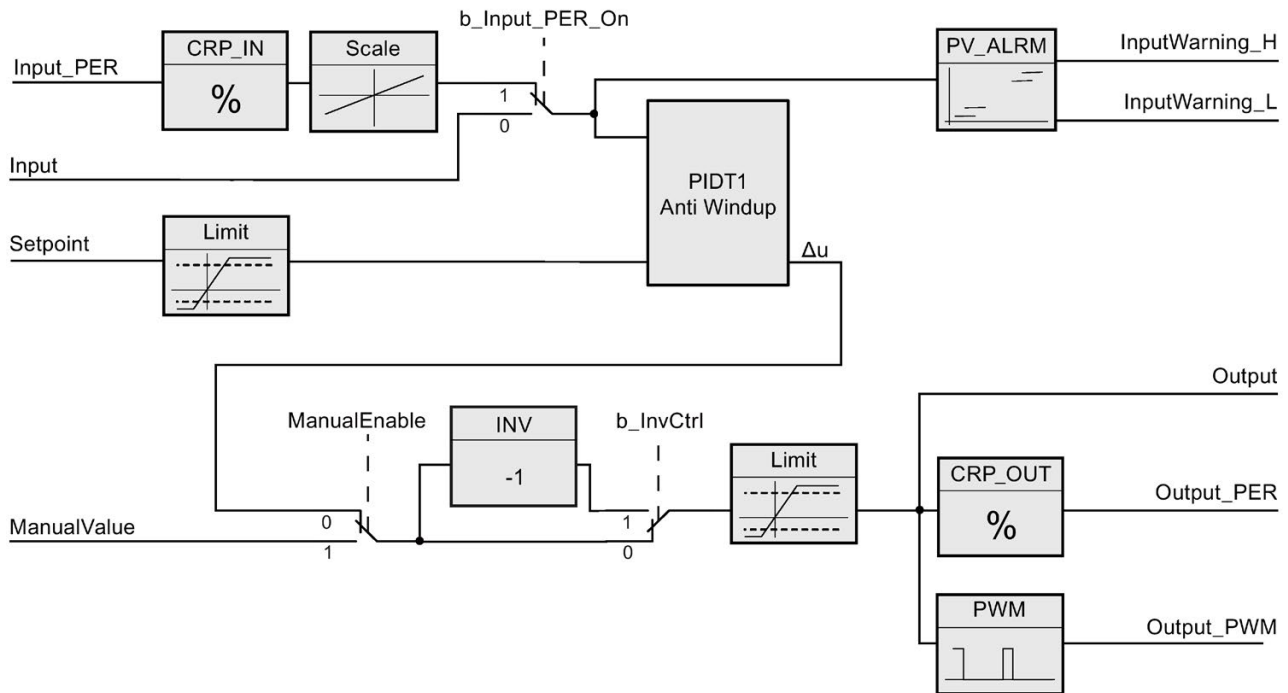


Figure 10-1 Fonctionnement du régulateur PID_Compact

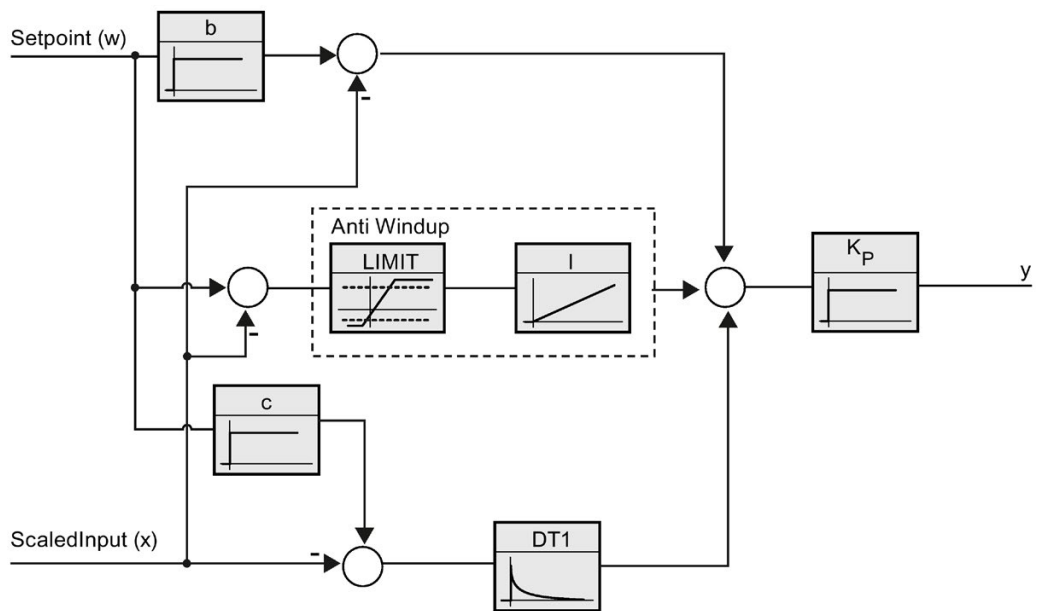


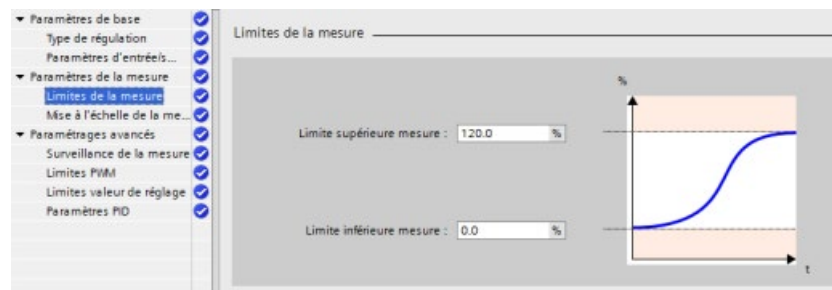
Figure 10-2 Fonctionnement du régulateur PID_Compact en tant que régulateur PID1 avec anti-emballement

10.2.2.2 Limites de la mesure dans l'instruction PID_Compact

Les limites de la mesure sont normalement utilisées en lien avec l'entrée analogique. Il est toutefois possible de les utiliser à d'autres fins.

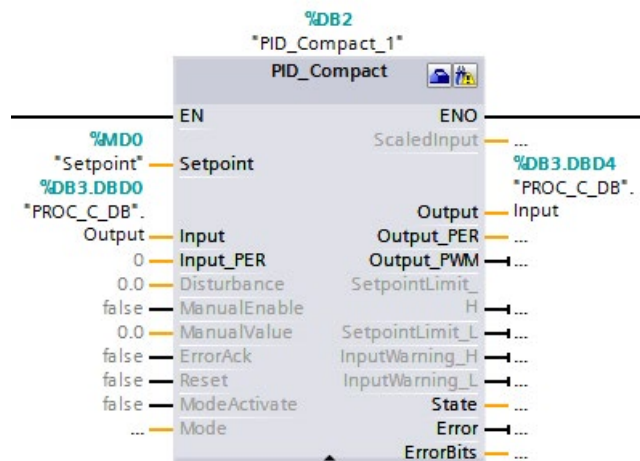
La configuration des limites de la mesure permet de :

- définir les limites supérieure et inférieure pour les sorties d'alarme supérieure/inférieure de la consigne dans le bloc PID ;
- définir une limite de sorte que la variable de processus ne puisse pas dépasser ces limites par le haut ou le bas quelle que soit la consigne. Cette configuration définit des limites fixes pour le processus.



Dans le bloc PID, vous pouvez utiliser quatre sorties d'alarme :

- SetpointLimit_H : la consigne est supérieure à la limite supérieure.
- SetpointLimit_L : la consigne est inférieure à la limite inférieure.
- InputWarning_H : la variable de processus est supérieure à la limite supérieure d'entrée.
- InputWarning_L : la variable de processus est inférieure à la limite inférieure d'entrée.



Lorsque vous définissez les limites de la mesure, les valeurs qui déclenchent le passage à vrai des sorties d'alarme supérieure et inférieure de la consigne sont les mêmes. Avec les paramétrages ci-dessus par exemple, si vous saisissez une consigne supérieure à 120%, la sortie "SetpointLimit_H" prendra la valeur "vrai". Il en est de même pour la limite inférieure : si vous saisissez une consigne inférieure à 0%, la sortie "SetpointLimit_L" prendra la valeur "vrai". Cela indique à votre programme que la consigne saisie est hors plage. Cette alarme peut déclencher l'affichage d'un message vous demandant de ressaisir la consigne.

Si vous saisissez une consigne hors plage, Compact_PID limite automatiquement la variable de processus à la plage configurée. Par exemple, si la limite supérieure de la mesure est définie à 120% (comme illustré dans la figure ci-avant), vous pouvez toujours saisir une consigne supérieure à 120%. Au fur et à mesure que la variable du processus s'approche des 120%, l'instruction PID réduit la sortie et pilote le processus à la limite supérieure de 120%. L'action est similaire si la consigne saisie est inférieure à la limite inférieure de la mesure : l'instruction PID ne laisse pas la variable du processus passer en dessous de la limite inférieure. Cette fonction vous permet de définir un fonctionnement du processus acceptable pendant la régulation PID automatique normale. Elle ne s'applique toutefois pas au démarrage et à l'arrêt à moins que le régulateur PID ne soit en mode automatique. Si le régulateur PID est en mode automatique et que la consigne et la variable du processus sont inférieures à la limite inférieure, le régulateur PID tente de piloter le processus à la limite inférieure configurée.

10.2.2.3 Paramètre ErrorBits de l'instruction PID_Compact

Si plusieurs erreurs sont en attente, les valeurs des codes d'erreur sont données au moyen d'une addition binaire. Ainsi, l'affichage du code d'erreur 0003 signifie que les erreurs 0001 et 0002 sont également en attente.

Tableau 10- 20 Paramètre ErrorBits de l'instruction PID_Compact

ErrorBit (DW#16#...)	Description
0000	Pas d'erreur
0001 ^{1, 2}	Le paramètre Input se situe en dehors des limites de la mesure. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Valeur invalide pour le paramètre Input_PER. Vérifiez s'il y a une erreur en attente au niveau de l'entrée analogique.
0004 ⁴	Erreur pendant l'optimisation fine. L'oscillation de la mesure n'a pas pu être conservée.
0008 ⁴	Erreur au démarrage de l'optimisation préalable. La mesure est trop proche de la consigne. Démarrez l'optimisation fine.
0010 ⁴	La consigne a été modifiée pendant l'optimisation. Note : Vous pouvez définir la fluctuation permise sur la consigne à la variable CancelTuningLevel.
0020	L'optimisation préalable n'est pas permise pendant l'optimisation fine. Note : Si ActivateRecoverMode à la valeur TRUE avant que l'erreur ne survienne, PID_Compact reste en mode optimisation fine.
0080 ⁴	Erreur pendant l'optimisation préalable. Configuration incorrecte des limites de la valeur de réglage. Vérifiez si les limites de la valeur de réglage sont configurées correctement et correspondent à la logique du régulateur.
0100 ⁴	Une erreur pendant l'optimisation fine a entraîné des paramètres invalides.
0200 ^{2, 3}	Valeur invalide pour le paramètre Input. La valeur a un format numérique invalide.
0400 ^{2, 3}	Le calcul de la valeur de réglage a échoué. Vérifiez les paramètres PID.
0800 ^{1, 2}	Erreur de période d'échantillonnage : PID_Compact n'est pas appelé dans la période d'échantillonnage de l'OB d'alarme cyclique.

ErrorBit (DW#16#...)	Description
1000 ^{2, 3}	Valeur invalide pour le paramètre Setpoint. La valeur a un format numérique invalide.
10000	Valeur invalide pour le paramètre ManualValue. La valeur a un format numérique invalide. Note : Si ActivateRecoverMode à la valeur TRUE avant que l'erreur ne survienne, PID_Compact utilise SubstituteOutput comme valeur de réglage. Dès que vous affectez une valeur invalide dans le paramètre ManualValue, PID_Compact l'utilise comme valeur de réglage.
20000	Valeur invalide pour la variable SubstituteValue : La valeur a un format numérique invalide. PID_Compact utilise la limite inférieure de la valeur de réglage comme valeur de réglage. Note : Si le mode automatique était actif avant l'apparition de l'erreur, ActivateRecoverMode a la valeur TRUE et que l'erreur n'est plus en attente, PID_Compact repasse en mode automatique.
40000	Valeur invalide pour le paramètre Disturbance. La valeur a un format numérique invalide. Note : Si le mode automatique était actif et si ActivateRecoverMode a la valeur FALSE avant l'apparition de l'erreur, Disturbance est défini sur zéro. PID_Compact reste en mode automatique. Note : Si l'optimisation préalable ou l'optimisation fine était active et ActivateRecoverMode a la valeur TRUE avant que l'erreur ne se produise, PID_Compact passe dans l'état de fonctionnement qui est enregistré pour le paramètre Mode. Si une perturbation dans la phase actuelle n'a pas d'effet sur la valeur de réglage, l'optimisation n'est pas annulée.

- ¹ Note : Si l'état automatique était actif avant que l'erreur ne se produise et ActivateRecoverMode a la valeur TRUE, PID_Compact reste en mode automatique.
- ² Note : Si l'optimisation préalable ou l'optimisation fine était active avant que l'erreur ne se produise et si ActivateRecoverMode a la valeur TRUE, PID_Compact passe dans l'état de fonctionnement qui est enregistré pour le paramètre Mode.
- ³ Note : Si le mode automatique était actif avant que l'erreur ne se produise et si ActivateRecoverMode a la valeur TRUE, PID_Compact délivre la valeur de réglage de remplacement configurée. Dès que l'erreur n'est plus en attente, PID_Compact repasse en mode automatique.
- ⁴ Note : Si ActivateRecoverMode a la valeur TRUE avant que l'erreur ne se produise, PID_Compact annule l'optimisation et passe dans l'état de fonctionnement qui est enregistré pour le paramètre Mode.

10.2.2.4 Paramètres d'avertissement de l'instruction PID_Compact

Si le contrôleur PID a plusieurs avertissements en attente, il affiche les valeurs des codes d'erreur au moyen d'une addition binaire. L'affichage du code d'erreur 0003, par exemple, indique que les erreurs 0001 et 0002 sont en attente.

Tableau 10- 21 Instruction PID_Compact, paramètres Warning

Avertissement (DW#16#...)	Description
0000	Aucun avertissement en attente.
0001 ¹	Le point d'inflexion n'a pas été trouvé pendant le préréglage.
0002	L'oscillation a été appliquée pendant le "réglage en cours". (Le paramètre "Warning" supprime cet avertissement et est uniquement visible dans le paramètre "WarningInternal" à des fins de diagnostic.)
0004 ¹	Le point de consigne a été limité selon les limites configurées.
0008 ¹	Toutes les propriétés de système contrôlées nécessaires n'ont pas été définies pour la méthode de calcul sélectionnée. À l'inverse, les paramètres PID ont été calculés en utilisant la méthode TIR.TuneRuleHeat / TIR TuneRuleCool = 3.
0010	Le mode de fonctionnement ne pourrait pas être changé car Reset = VRAI ou ManualEnabled = VRAI.
0020	Le temps de cycle de l'OB appelant limite le temps d'échantillonnage de l'algorithme PID. Améliorez les résultats en utilisant des temps de cycle OB plus courts.
0040 ¹	La valeur du processus a dépassé l'une de ses limites d'avertissement.
0080	Valeur invalide au niveau du mode. Le mode de fonctionnement n'est pas activé.
0100 ¹	La valeur manuelle a été limitée selon les limites de la sortie du contrôleur.
0200	La règle spécifiée pour le réglage n'est pas supportée. Aucun paramètre PID n'est calculé.
1000	La valeur de sortie de remplacement ne peut pas être atteinte car elle se situe en dehors des limites de la valeur de sortie.

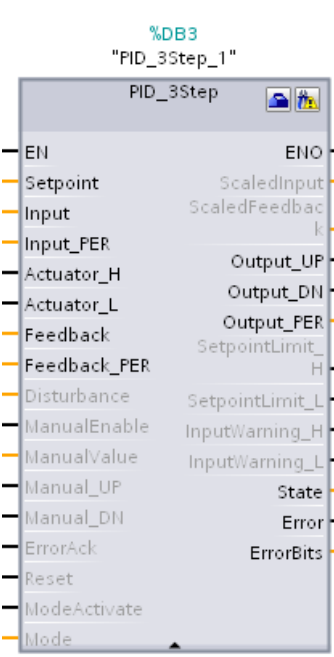
¹ Remarque : Le contrôleur PID a automatiquement supprimé les avertissements suivants dès que la cause a été éliminée ou que l'action de l'utilisateur a été répétée avec les paramètres valides : 0001, 0004, 0008, 0040 et 0100.

10.2.3 PID_3Step

10.2.3.1 Instruction PID_3Step

L'instruction PID_3Step configure un régulateur PID à fonction d'autooptimisation qui a été optimisé pour des vannes et actionneurs commandés par moteur.

Tableau 10- 22 Instruction PID_3Step

CONT/LOG	SCL	Description
	<pre>"PID_3Step_1" (SetpoInt:=_real_in_, Input:=_real_in_, ManualValue:=_real_in_, Feedback:=_real_in_, InputPer:=_word_in_, FeedbackPer:=_word_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualUP:=_bool_in_, ManualDN:=_bool_in_, ActuatorH:=_bool_in_, ActuatorL:=_bool_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, ScaledInput=>_real_out_, ScaledFeedback=>_real_out_, ErrorBits=>_dword_out_, OutputPer=>_word_out_, State=>_int_out_, OutputUP=>_bool_out_, OutputDN=>_bool_out_, SetpoIntLimitH=>_bool_out_, SetpoIntLimitL=>_bool_out_, InputWarningH=>_bool_out_, InputWarningL=>_bool_out_, Error=>_bool_out_, ErrorBits=>_dword_out_);</pre>	<p>PID_3Step configure un régulateur PID à fonction d'autooptimisation qui a été optimisé pour des vannes et actionneurs commandés par moteur. Cette instruction fournit deux sorties booléennes.</p> <p>PID_3Step est un régulateur PID T1 avec anti-emballement et pondération des actions P et D.</p>

- 1 STEP 7 crée automatiquement l'objet technologique et le DB d'instance lorsque vous insérez l'instruction. Le DB d'instance contient les paramètres de l'objet technologique.
- 2 Dans l'exemple SCL, "PID_3Step_1" est le nom du DB d'instance.

Tableau 10- 23 Types de données pour les paramètres

Paramètre et type		Type de données	Description
Setpoint	IN	Real	Consigne du régulateur PID en mode automatique. (valeur par défaut : 0.0)
Input	IN	Real	Une variable du programme utilisateur est utilisée comme source pour la mesure. (valeur par défaut : 0.0) Si vous utilisez le paramètre Input alors il faut définir Config.InputPerOn = FALSE.
Input_PER	IN	Word	Une entrée analogique est utilisée comme source pour la mesure. (valeur par défaut : W#16#0) Si vous utilisez le paramètre Input_PER alors il faut définir Config.InputPerOn = TRUE.
Actuator_H	IN	Bool	Signalisation de position numérique de la vanne pour la butée supérieure Si Actuator_H = TRUE, la vanne est à la butée supérieure et n'est plus déplacée dans ce sens. (valeur par défaut : FALSE)
Actuator_L	IN	Bool	Signalisation de position numérique de la vanne pour la butée inférieure Si Actuator_L = TRUE, la vanne est à la butée inférieure et n'est plus déplacée dans ce sens. (valeur par défaut : FALSE)
Feedback	IN	Real	Signalisation de position de la vanne. (valeur par défaut : 0.0) Si vous utilisez le paramètre Feedback alors il faut définir Config.FeedbackPerOn = FALSE.
Feedback_PER	IN	Int	Signalisation analogique de la position de la vanne. (valeur par défaut : W#16#0) Si vous utilisez le paramètre Feedback_PER, vous devez définir Config.FeedbackPerOn = TRUE. Feedback_PER est mise à l'échelle, en fonction des variables suivantes : <ul style="list-style-type: none"> • Config.FeedbackScaling.LowerPointIn • Config.FeedbackScaling.UpperPointIn • Config.FeedbackScaling.LowerPointOut • Config.FeedbackScaling.UpperPointOut
Disturbance	IN	Real	Variable de perturbation ou valeur de pré-régulation
ManualEnable	IN	Bool	Active ou désactive le mode de fonctionnement manuel. (valeur par défaut : FALSE) : <ul style="list-style-type: none"> • Un front de FALSE à TRUE active le mode manuel", tant que Etat = 4, le mode reste inchangé. <p>Tant que ManualEnable est VRAI, vous ne pouvez pas modifier l'état de fonctionnement à l'aide d'un front montant pour ModeActivate ou utiliser la boîte de dialogue de mise en service.</p> <ul style="list-style-type: none"> • Un front TRUE à FALSE active l'état de fonctionnement qui est affecté par Mode. <p>Note : Nous vous recommandons de modifier l'état de fonctionnement à l'aide de ModeActivate uniquement.</p>

Paramètre et type		Type de données	Description
ManualValue	IN	Real	Valeur de réglage pour le mode manuel. (valeur par défaut : 0.0) En mode manuel, vous indiquez la position absolue de la vanne. ManualValue est évalué uniquement si vous utilisez OutputPer ou si la signalisation de position est disponible.
ManualUP	IN	Bool	<ul style="list-style-type: none"> Manual_UP = TRUE : <ul style="list-style-type: none"> La vanne est ouverte même si vous utilisez Output_PER ou une signalisation de position. La vanne n'est plus déplacée si la butée supérieure a été atteinte. Voir aussi Config.VirtualActuatorLimit Manual_UP = FALSE : <ul style="list-style-type: none"> Si vous utilisez Output_PER ou une signalisation de position, la vanne est déplacée sur ManualValue. Sinon, la vanne n'est plus déplacée. <p>Note : Si Manual_UP et Manual_DN ne sont pas paramétrés sur TRUE simultanément, la vanne n'est pas déplacée.</p>
ManualDN	IN	Bool	<ul style="list-style-type: none"> Manual_DN = TRUE : <ul style="list-style-type: none"> La vanne est ouverte même si vous utilisez Output_PER ou une signalisation de position. La vanne n'est plus déplacée si la butée supérieure a été atteinte. Voir aussi Config.VirtualActuatorLimit Manual_DN = FALSE : <ul style="list-style-type: none"> Si vous utilisez Output_PER ou une signalisation de position, la vanne est déplacée sur ManualValue. Sinon, la vanne n'est plus déplacée.
ErrorAck	IN	Bool	Réinitialise le ErrorBits et les messages d'alerte. Front de FALSE à TRUE
Reset	IN	Bool	Redémarre le régulateur. (valeur par défaut : FALSE) : <ul style="list-style-type: none"> Front de FALSE à TRUE : <ul style="list-style-type: none"> Passé en mode "inactif" Réinitialise les ErrorBits et les sorties d'alerte Efface l'action intégrale Maintient les paramètres PID Tant que Reset à la valeur TRUE, PID_3Step reste en mode "Inactif" (Etat = 0). Front de TRUE à FALSE : <ul style="list-style-type: none"> PID_3Step passe dans l'état de fonctionnement qui est enregistré dans le paramètre Mode.
ModeActivate	IN	Bool	Le PID_3Step passe dans l'état qui est enregistré pour le paramètre Mode. Front de FALSE à TRUE :
Mode	IN	Int	Le mode PID souhaité ; Activé sur le front montant de l'entrée Mode Activate .
ScaledInput	OUT	Real	Mesure mise à l'échelle

Paramètre et type		Type de données	Description
ScaledFeedback	OUT	Real	Signalisation de position de vanne mise à l'échelle Note : Pour un actionneur sans signalisation de position, la position de l'actionneur indiquée par ScaledFeedback est très imprécise. ScaledFeedback peut seulement être utilisée que pour une estimation approximative de la position actuelle dans ce cas.
Output_UP	OUT	Bool	Valeur de réglage TOR pour l'ouverture de la vanne. (valeur par défaut : FALSE) Si Config.OutputPerOn a la valeur FALSE, le paramètre Output_UP est utilisé.
Output_DN	OUT	Bool	Valeur de réglage TOR pour la fermeture de la vanne. (valeur par défaut : FALSE) Si Config.OutputPerOn a la valeur FALSE, le paramètre Output_DN est utilisé.
Output_PER	OUT	Word	Valeur de réglage analogique. Si Config.OutputPerOn a la valeur TRUE, le paramètre Output_PER est utilisé.
SetpointLimitH	OUT	Bool	Limite supérieure de la consigne. (valeur par défaut : FALSE) Si SetpointLimitH a la valeur TRUE, la limite supérieure absolue de la consigne est atteinte (Setpoint \geq Config.SetpointUpperLimit). Note : La consigne est limitée à (Setpoint \geq Config.SetpointUpperLimit).
SetpointLimitL	OUT	Bool	Limite inférieure de la consigne. (valeur par défaut : FALSE) Si SetpointLimitL a la valeur TRUE, la limite inférieure absolue de la consigne est atteinte (Setpoint \geq Config.SetpointLowerLimit). Note : La consigne est limitée à (Setpoint \geq Config.SetpointLowerLimit).
InputWarningH	OUT	Bool	Si InputWarningH a la valeur TRUE, la grandeur d'entrée a atteint ou dépassé la limite d'alerte supérieure. (valeur par défaut : FALSE)
InputWarningL	OUT	Bool	Si InputWarningL a la valeur TRUE, la grandeur d'entrée a atteint ou dépassé la limite d'alerte inférieure. (valeur par défaut : FALSE)
State	OUT	Int	Mode de fonctionnement en cours du régulateur PID. (valeur par défaut : 0) Vous pouvez modifier l'état de fonctionnement à l'aide du paramètre d'entrée Mode et un front montant pour ModeActivate: <ul style="list-style-type: none"> • State = 0 : Inactif • State = 1 : Optimisation préalable • State = 2 : Optimisation fine manuelle • State = 3 : Mode automatique • State = 4 : Mode manuel • State = 5 : Accostage de la valeur de réglage de remplacement • State = 6 : Mesure du temps de transition • State = 7 : Surveillance d'erreur • State = 8 : Accostage de la valeur de réglage de remplacement avec surveillance d'erreur • State = 10 : Mode manuel sans signaux de butée d'extrémité

Paramètre et type		Type de données	Description
Error	OUT	Bool	Si Error = TRUE, au moins un message d'erreur est en attente. (valeur par défaut : FALSE) Note : Le paramètre Error dans V1.x PID était le champ ErrorBits qui contenait les codes d'erreurs. C'est maintenant une signalisation booléenne qui indique qu'une erreur est survenue.
ErrorBits	OUT	DWord	L'instruction PID_3StepErrorBits table de paramètres (Page 626) définit les messages d'erreurs qui sont en attente. (valeur par défaut : DW#16#0000 (aucune erreur)). ErrorBits est rémanent et réinitialisé sur un front montant pour Reset ou ErrorAck. Note : Dans V1.x, le paramètre ErrorBits a été défini comme le paramètre Error et n'existait pas.

Fonctionnement du régulateur PID_3Step

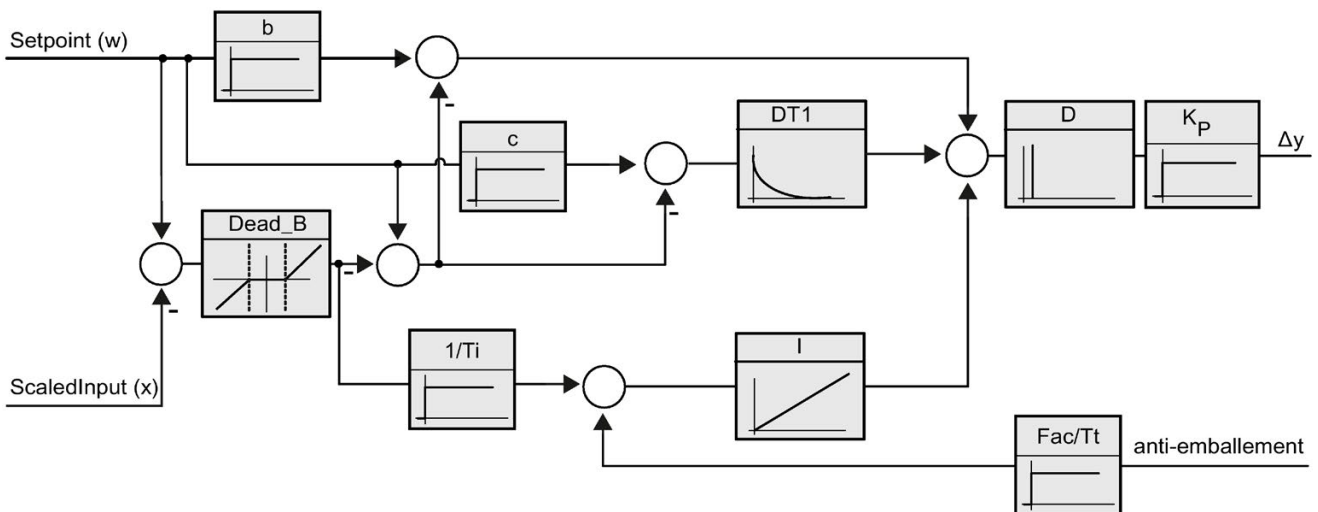


Figure 10-3 Fonctionnement du régulateur PID_3Step en tant que régulateur PID T1 avec anti-emballement

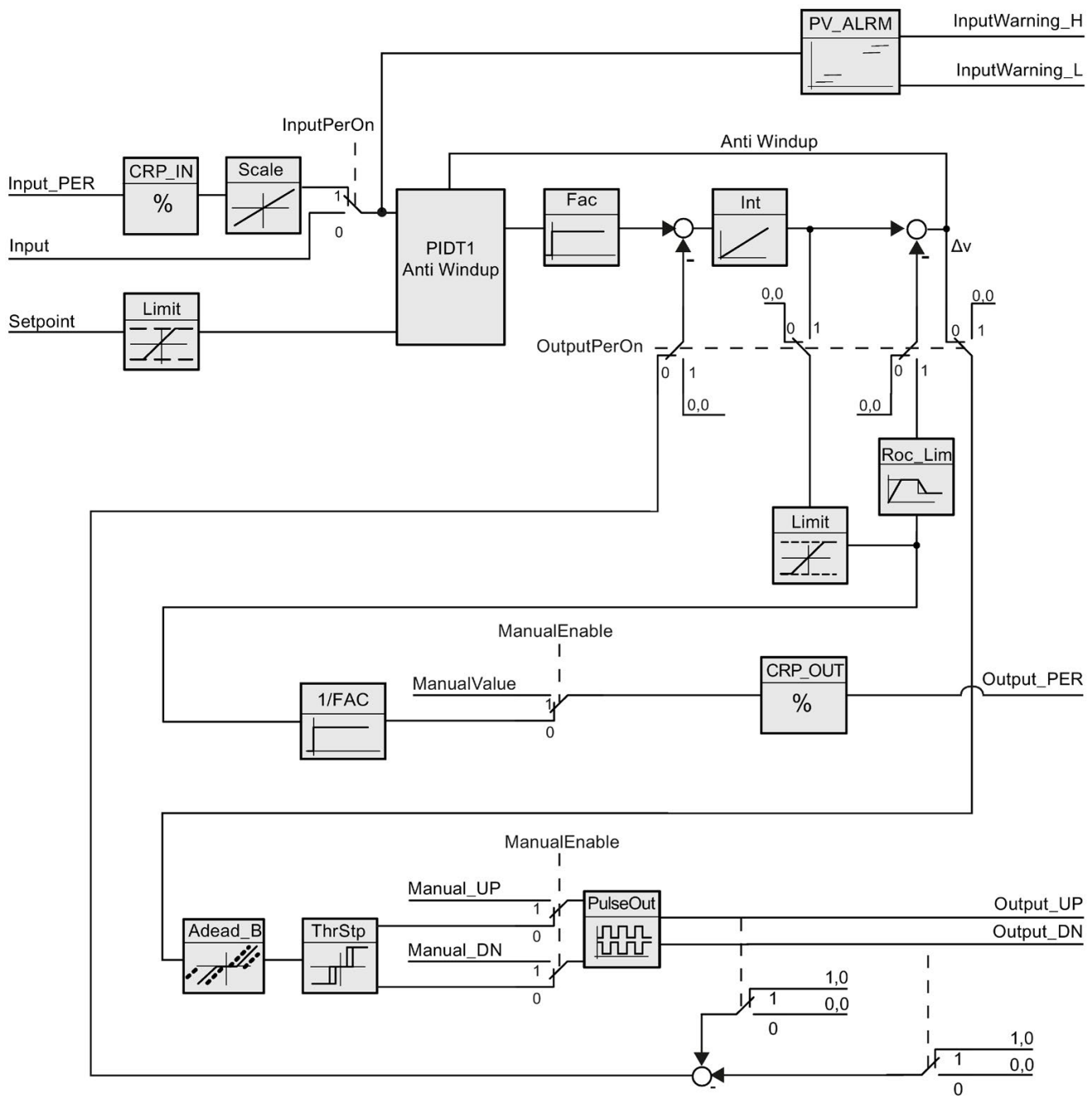


Figure 10-4 Fonctionnement du régulateur PID_3Step sans signalisation de position

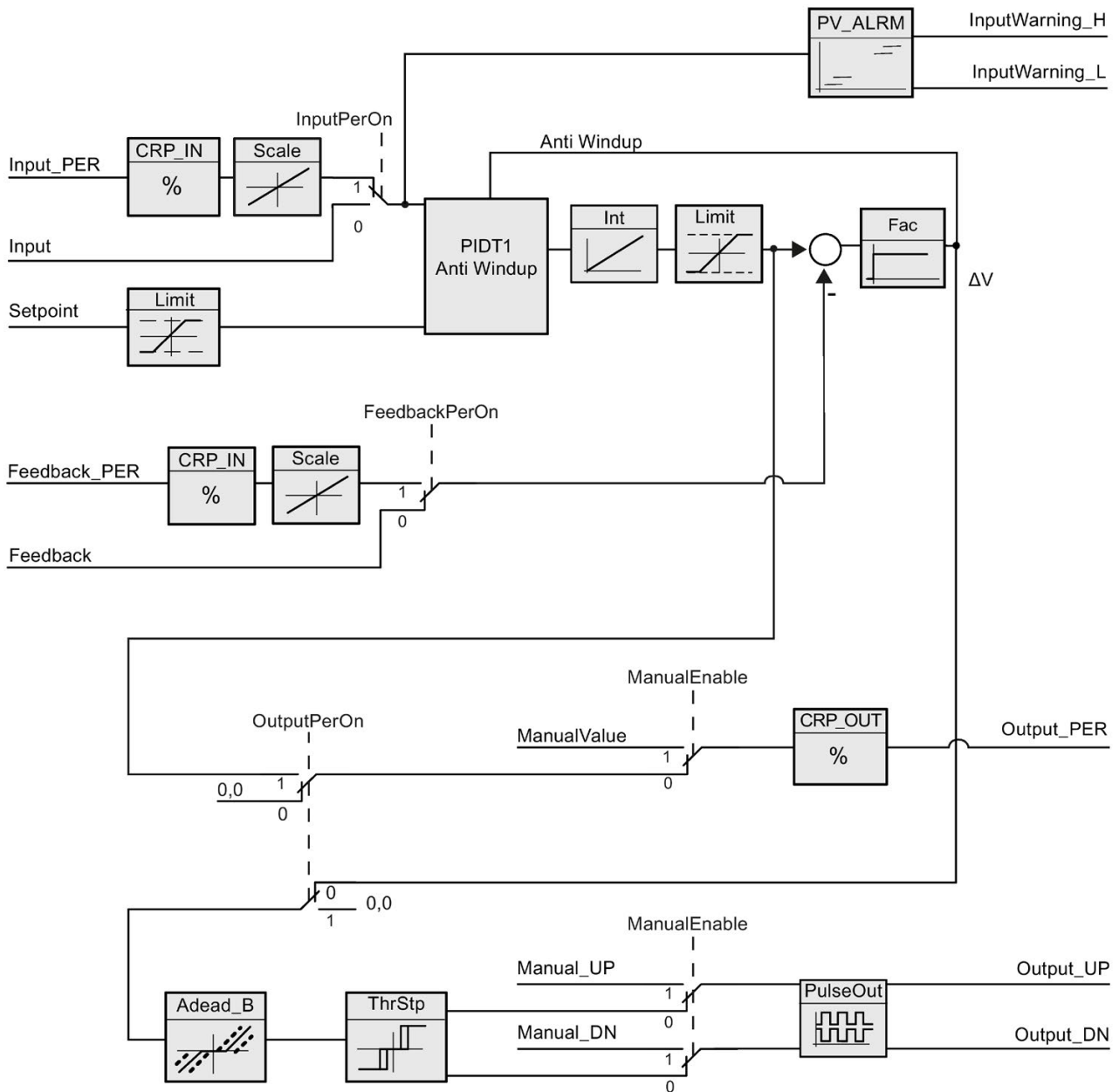


Figure 10-5 Fonctionnement du régulateur PID_3Step avec signalisation de position activée

10.2.3.2 Paramètres ErrorBit de l'instruction PID_3Step

Si plusieurs erreurs sont en attente, les valeurs des codes d'erreur sont données au moyen d'une addition binaire. Ainsi, l'affichage du code d'erreur 0003 signifie que les erreurs 0001 et 0002 sont également en attente.

Tableau 10- 24 Paramètres ErrorBit de l'instruction PID_3STEP

ErrorBit (DW#16#...)	Description
0000	Pas d'erreur
0001 ^{1, 2}	Le paramètre Input se situe en dehors des limites de la mesure. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Valeur invalide pour le paramètre Input_PER. Vérifiez s'il y a une erreur en attente au niveau de l'entrée analogique.
0004 ⁴	Erreur pendant l'optimisation fine. L'oscillation de la mesure n'a pas pu être conservée.
0010 ⁴	La consigne a été modifiée pendant l'optimisation. Note : Vous pouvez définir la fluctuation permise sur la consigne à la variable CancelTuningLevel.
0020	L'optimisation préalable n'est pas permise pendant l'optimisation fine. Note : Si ActivateRecoverMode à la valeur TRUE avant que l'erreur ne survienne, PID_3Step reste en mode optimisation fine.
0080 ⁴	Erreur pendant l'optimisation préalable. Configuration incorrecte des limites de la valeur de réglage. Vérifiez si les limites de la valeur de réglage sont configurées correctement et correspondent à la logique du régulateur.
0100 ⁴	Une erreur pendant l'optimisation fine a entraîné des paramètres invalides.
0200 ^{2, 3}	Valeur invalide pour le paramètre Input. La valeur a un format numérique invalide.
0400 ^{2, 3}	Le calcul de la valeur de réglage a échoué. Vérifiez les paramètres PID.
0800 ^{1, 2}	Erreur de période d'échantillonnage : PID_3Step n'est pas appelé dans la période d'échantillonnage de l'OB d'alarme cyclique.
1000 ^{2, 3}	Valeur invalide pour le paramètre Setpoint. La valeur a un format numérique invalide.
2000 ^{1, 2, 5}	Valeur invalide pour le paramètre Feedback_PER. Vérifiez s'il y a une erreur en attente au niveau de l'entrée analogique.
4000 ^{1, 2, 5}	Valeur invalide pour le paramètre Feedback. La valeur a un format numérique invalide.
8000 ^{1, 2}	Erreur pendant la signalisation de position numérique. Actuator_H = TRUE et Actuator_L = TRUE. L'actionneur ne peut pas être déplacé vers la valeur de réglage de remplacement et reste dans la position actuelle. Le mode manuel n'est pas possible dans cet état. Afin de déplacer l'actionneur de cet état, vous devez désactiver la "butée d'extrémité de l'actionneur" (Config.ActuatorEndStopOn a la valeur FALSE) ou passer sur le mode manuel sans signaux de butée d'extrémité (Mode = 10).

ErrorBit (DW#16#...)	Description
10000	Valeur invalide pour le paramètre ManualValue. La valeur a un format numérique invalide. L'actionneur ne peut pas être déplacé vers la valeur manuelle et reste dans la position en cours. Affecte une valeur valide dans ManualValue ou déplace l'actionneur en mode manuel avec Manual_UP et Manual_DN.
20000	Valeur invalide pour la variable SavePosition : La valeur a un format numérique invalide. L'actionneur ne peut pas être déplacé vers la valeur de réglage de remplacement et reste dans la position actuelle.
40000	Valeur invalide pour le paramètre Disturbance. La valeur a un format numérique invalide. Note : Si le mode automatique était actif et si ActivateRecoverMode a la valeur FALSE avant l'apparition de l'erreur, Disturbance est défini sur zéro. PID_3Step reste en mode automatique. Note : Si l'optimisation préalable ou l'optimisation fine était active et ActivateRecoverMode a la valeur TRUE avant que l'erreur ne se produise, PID_3Step passe dans l'état de fonctionnement qui est enregistré pour le paramètre Mode. Si une perturbation dans la phase actuelle n'a pas d'effet sur la valeur de réglage, l'optimisation n'est pas annulée. L'erreur n'a pas d'effet pendant la mesure du temps de transition.

- ¹ Note : Si l'état automatique était actif avant que l'erreur ne se produise et ActivateRecoverMode a la valeur TRUE, PID_3Step reste en mode automatique.
- ² Note : Si le mode optimisation préalable, optimisation fine ou mesure du temps de transition était actif et si ActivateRecoverMode = TRUE avant que l'erreur ne survienne, PID_3Step passe sur l'état de fonctionnement qui a été enregistré pour le paramètre Mode.
- ³ Note : Si le mode automatique était actif avant que l'erreur ne survienne et si ActivateRecoverMode = TRUE, PID_3Step passe dans l'état "Accostage de la valeur de réglage de remplacement avec une surveillance d'erreur" ou "Surveillance d'erreur". Dès que l'erreur n'est plus en attente, PID_3Step repasse en mode automatique.
- ⁴ Note : Si ActivateRecoverMode a la valeur TRUE avant que l'erreur ne se produise, PID_3Step annule l'optimisation et passe dans l'état de fonctionnement qui est enregistré pour le paramètre Mode.
- ⁵ L'actionneur ne peut pas être déplacé vers la valeur de réglage de remplacement et reste dans la position actuelle. En mode manuel, vous pouvez modifier la position de l'actionneur uniquement avec Manual_UP et Manual_DN, et pas avec ManualValue.

10.2.3.3 Paramètres d'avertissement de l'instruction PID_3Step

Si le contrôleur PID a plusieurs avertissements en attente, il affiche les valeurs des codes d'erreur au moyen d'une addition binaire. L'affichage du code d'erreur 0003, par exemple, indique que les erreurs 0001 et 0002 sont en attente.

Tableau 10- 25 Instruction PID_Compact, paramètres Warning

Avertissement (DW#16#...)	Description
0000	Aucun avertissement en attente.
0001 ¹	Le point d'inflexion n'a pas été trouvé pendant le préréglage.
0002	L'oscillation a été appliquée pendant le "réglage en cours". (Le paramètre "Warning" supprime cet avertissement et est uniquement visible dans le paramètre "WarningInternal" à des fins de diagnostic.)
0004 ¹	Le point de consigne a été limité selon les limites configurées.
0008 ¹	Toutes les propriétés de système contrôlées nécessaires n'ont pas été définies pour la méthode de calcul sélectionnée. À l'inverse, les paramètres PID ont été calculés en utilisant la méthode TIR.TuneRuleHeat / TIR TuneRuleCool = 3.
0010	Le mode de fonctionnement ne pourrait pas être changé car Reset = VRAI ou ManualEnable = VRAI.
0020	Le temps de cycle de l'OB appelant limite le temps d'échantillonnage de l'algorithme PID. Améliorez les résultats en utilisant des temps de cycle OB plus courts.
0040 ¹	La valeur du processus a dépassé l'une de ses limites d'avertissement.
0080	Valeur invalide au niveau du mode. Le mode de fonctionnement n'est pas activé.
0100 ¹	La valeur manuelle a été limitée selon les limites de la sortie du contrôleur.
0200	La règle spécifiée pour le réglage n'est pas supportée. Aucun paramètre PID n'est calculé.
1000	La valeur de sortie de remplacement ne peut pas être atteinte car elle se situe en dehors des limites de la valeur de sortie.

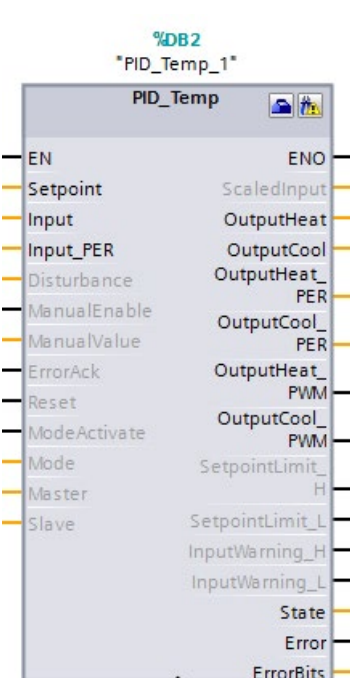
¹ Remarque : Le contrôleur PID a automatiquement supprimé les avertissements suivants dès que la cause a été éliminée ou que l'action de l'utilisateur a été répétée avec les paramètres valides : 0001, 0004, 0008, 0040 et 0100.

10.2.4 PID_Temp

10.2.4.1 Instruction PID_Temp

L'instruction PID_Temp offre un régulateur PID universel qui permet de gérer des exigences spécifiques en matière de régulation de la température.

Tableau 10- 26 Instruction PID_Temp

CONT/LOG	SCL	Description
	<pre>"PID_Temp_1" (Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_int_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, Master:=_dword_in Save:=_dword_in ScaledInput=>_real_out_, OutputHeat=>_real_out_, OutputCool=>_real_out_, OutputHeat_PER=>_int_out_, OutputCool_PER=>_int_out_, Out- putHeat_PWM=>_bool_out_, Out- putCool_PWM=>_bool_out_, SetpointLimit_H=>_bool_out_, SetpointLimit_L=>_bool_out_, InputWarn- ing_H=>_bool_out_, InputWarn- ing_L=>_bool_out_, State=>_int_out_, Error=>_bool_out_, ErrorBits=>_dword_out_);</pre>	<p>PID_Temp offre les possibilités suivantes :</p> <ul style="list-style-type: none"> • Chauffage et refroidissement du processus grâce à différents actionneurs • Réglage automatique intégré pour gérer les processus de température • Effet de cascade pour traiter plusieurs températures qui dépendent du même actionneur

1 STEP 7 crée automatiquement l'objet technologique et le DB d'instance lorsque vous insérez l'instruction. Le DB d'instance contient les paramètres de l'objet technologique.

2 Dans l'exemple SCL, "PID_Temp_1" est le nom du DB d'instance.

Tableau 10- 27 Types de données pour les paramètres

Paramètre et type		Type de données	Description
Setpoint	IN	Real	Consigne du régulateur PID en mode automatique. (valeur par défaut : 0.0)
Input	IN	Real	Une variable du programme utilisateur est utilisée comme source pour la mesure. (valeur par défaut : 0.0) Si vous utilisez le paramètre Input alors il faut définir Config.InputPerOn = FALSE.
Input_PER	IN	Int	Une entrée analogique est utilisée comme source pour la mesure. (valeur par défaut : 0) Si vous utilisez le paramètre Input_PER alors il faut définir Config.InputPerOn = TRUE.
Disturbance	IN	Real	Variable de perturbation ou valeur de pré-régulation
ManualEnable	IN	Bool	Active ou désactive le mode de fonctionnement manuel. (valeur par défaut : FALSE) : <ul style="list-style-type: none"> Un front de FALSE à TRUE active le mode manuel, tant que Etat = 4, le mode reste inchangé. Tant que ManualEnable est VRAI, vous ne pouvez pas modifier l'état de fonctionnement à l'aide d'un front montant pour ModeActivate ou utiliser la boîte de dialogue de mise en service. <ul style="list-style-type: none"> Un front TRUE à FALSE active l'état de fonctionnement qui est affecté par Mode. Note : Nous vous recommandons de modifier l'état de fonctionnement à l'aide de ModeActivate uniquement.
ManualValue	IN	Real	Valeur de sortie pour le mode manuel. (valeur par défaut : 0.0) Vous pouvez utiliser des valeurs à partir Config.OutputLowerLimit à Config.OutputUpperLimit.
ErrorAck	IN	Bool	Réinitialise les ErrorBits et les messages d'alerte avec un front de FALSE à TRUE. (valeur par défaut : FALSE)
Reset	IN	Bool	Redémarre le régulateur. (valeur par défaut : FALSE) : <ul style="list-style-type: none"> Front de FALSE à TRUE : <ul style="list-style-type: none"> – Passe en mode "inactif" – Réinitialise les ErrorBits et les sorties d'alerte – Efface l'action intégrale – Maintient les paramètres PID Tant que Reset à la valeur TRUE, PID_Temp reste en mode Inactif (Etat = 0). Front de TRUE à FALSE : <ul style="list-style-type: none"> – PID_Temp passe à l'état de fonctionnement qui est enregistré dans le paramètre Mode.
ModeActivate	IN	Bool	Le PID_Temp passe à l'état de fonctionnement qui est enregistré dans le paramètre Mode avec un front de FALSE à TRUE. (valeur par défaut : FALSE)

Paramètre et type		Type de données	Description
Mode	IN/OUT	Int	Activé sur le front montant de l'entrée Mode Active. Sélection du mode de fonctionnement (valeur par défaut : 0.0) : <ul style="list-style-type: none"> • Mode = 0 : Inactif • Mode = 1 : Optimisation préalable • Mode = 2 : Optimisation fine • Mode = 3 : Mode automatique • Mode = 4 : Mode manuel "Valeur de réglage de remplacement avec surveillance d'erreur" (State = 5). Celui-ci ne peut pas être activé par l'utilisateur ; il ne s'agit que d'une réaction automatique à une erreur.
Master	IN/OUT	DWord	Connexion en cascade au maître (anti-emballement et conditions d'optimisation). (valeur par défaut : DW#16#0000) <ul style="list-style-type: none"> • Bits 0 - 15 : Pas utilisés dans l'instruction PID_Temp • Bits 16 - 23 : Compteur de limite : Un esclave incrémente cette valeur si elle atteint sa limite. Le nombre d'esclaves en limitation est traité pour la fonctionnalité anti-emballement (Voir le paramètre Config.Cascade.AntiWindUpMode. • Bit 24 : IsAutomatic : Ce bit est défini sur "1" si tous les esclaves de ce contrôleur sont en mode automatique et sont traités pour vérifier les conditions d'optimisation en cascade. Ce bit est identique au paramètre AllSlaveAutomaticState. • Bit 25 : "IsReplacement-Setpoint" : Ce bit est défini sur "1" si un esclave de ce contrôleur a la "Consigne de remplacement" activée et est traité pour vérifier les conditions d'optimisation en cascade. La valeur inversée est sauvegardée dans le paramètre NoSlaveReplacementSetpoint.
Slave	IN/OUT	DWord	
ScaledInput	OUT	Real	Mesure mise à l'échelle. (valeur par défaut : 0.0)
OutputHeat ¹	OUT	Real	Valeur de sortie pour le chauffage en mode REEL. (valeur par défaut : 0.0) Cette valeur de sortie est calculée, indépendamment de la sélection de sortie, à l'aide du paramètre Config.Output.Heat.Select.
OutputCool ¹	OUT	Real	Valeur de sortie pour le refroidissement en mode REEL. (valeur par défaut : 0.0) Cette valeur de sortie est calculée, indépendamment de la sélection de sortie, à l'aide du paramètre Config.Output.Cool.Select.
OutputHeat_PER ¹	OUT	Int	Valeur de sortie pour le chauffage au format périphérique (valeur par défaut : 0) Cette valeur de sortie n'est calculée que si elle est sélectionnée à l'aide du paramètre Config.Output.Heat.Select = 2. Si elle n'est pas sélectionnée, cette sortie est toujours égale à "0".
OutputCool_PER ¹	OUT	Int	Valeur de sortie pour le refroidissement au format périphérique (valeur par défaut : 0) Cette valeur de sortie n'est calculée que si elle est sélectionnée à l'aide du paramètre Config.Output.Cool.Select = 2. Si elle n'est pas sélectionnée, cette sortie est toujours égale à "0".

Paramètre et type		Type de données	Description
OutputHeat_PWM ¹	OUT	Bool	Valeur de sortie à modulation de largeur des impulsions pour le chauffage. (valeur par défaut : FALSE) Cette valeur de sortie n'est calculée que si elle est sélectionnée à l'aide du paramètre Config.Output.Heat.Select = 1 (valeur par défaut). Si elle n'est pas sélectionnée, cette sortie est toujours FALSE.
OutputCool_PWM ¹	OUT	Bool	Valeur de sortie à modulation de largeur des impulsions pour le refroidissement. (valeur par défaut : FALSE) Cette valeur de sortie n'est calculée que si elle est sélectionnée à l'aide du paramètre Config.Output.Cool.Select = 1 (valeur par défaut). Si elle n'est pas sélectionnée, cette sortie est toujours FALSE.
SetpointLimit_H	OUT	Bool	Limite supérieure de la consigne. (valeur par défaut : FALSE) Si SetpointLimit_H a la valeur TRUE, la limite supérieure absolue de la consigne est atteinte.(Setpoint ≥ Config.SetpointUpperLimit). La consigne est limitée à Config.SetpointUpperLimit.
SetpointLimit_L	OUT	Bool	Limite inférieure de la consigne. (valeur par défaut : FALSE) Si SetpointLimit_L a la valeur TRUE, la limite inférieure absolue de la consigne est atteinte.(Setpoint ≤ Config.SetpointLowerLimit). La consigne est limitée à Config.SetpointLowerLimit.
InputWarning_H	OUT	Bool	Si InputWarning_H a la valeur TRUE, la mesure a atteint ou dépassé la limite d'alerte supérieure. (valeur par défaut : FALSE)
InputWarning_L	OUT	Bool	Si InputWarning_L a la valeur TRUE, la mesure de processus a atteint ou est tombée en dessous de la limite d'alerte inférieure. (valeur par défaut : FALSE)
State	OUT	Int	Mode de fonctionnement en cours du régulateur PID. (valeur par défaut : 0) Vous pouvez modifier l'état de fonctionnement à l'aide du paramètre d'entrée Mode et un front montant pour ModeActivate : <ul style="list-style-type: none"> • State = 0 : Inactif • State = 1 : Optimisation préalable • State = 2 : Optimisation fine • State = 3 : Mode automatique • State = 4 : Mode manuel • State = 5 : Valeur de réglage de remplacement avec surveillance d'erreur
Error	OUT	Bool	Si Error a la valeur TRUE, au moins un message d'erreur est en attente dans ce cycle. (valeur par défaut : FALSE) Note : Le paramètre Error dans V1.x PID était le champ ErrorBits qui contenait les codes d'erreurs. C'est maintenant une signalisation booléenne qui indique qu'une erreur est survenue.
ErrorBits	OUT	DWord	Le tableau des paramètres ErrorBits (Page 640)de l'instruction PID_Temp présente les messages d'erreur qui sont en attente. (valeur par défaut : DW#16#0000 (aucune erreur)). ErrorBits est rémanent et réinitialisé sur un front montant pour Reset ou ErrorAck. Note : Dans V1.x, le paramètre ErrorBits a été défini comme le paramètre Error et n'existait pas.

Paramètre et type		Type de données	Description
Warning	OUT	DWord	Le tableau des paramètres Warning (Page 642) de l'instruction PID_Temp présente les messages d'alerte importants pour l'utilisateur, qui sont en attente. (valeur par défaut : DW#16#0000 (pas d'alerte)).
WarningInternal	OUT	DWord	Le tableau des paramètres WarningInternal de l'instruction PID_Temp présente les messages d'alerte internes qui sont en attente (comprend toutes les alertes). (valeur par défaut : DW#16#0000 (pas d'alerte interne)).

¹ Vous pouvez utiliser les sorties de la Output, Output_PER et Output_PWM des paramètres en parallèle.

Fonctionnement du régulateur PID_Temp

Sélection de la commande de chauffage et/ou de refroidissement

Vous devez d'abord choisir au paramètre "ActivateCooling" si vous avez besoin d'un dispositif de refroidissement en plus de la sortie de chauffage. Ensuite, vous devez indiquer au paramètre "AdvancedCooling" si vous voulez utiliser deux jeux de paramètres PID (mode avancé) ou un seul jeu de paramètres PID avec un facteur de chauffage/refroidissement supplémentaire.

Utilisation du facteur de refroidissement (CoolFactor)

Si vous voulez appliquer un facteur de chauffage/refroidissement, vous devez définir la valeur manuellement. Vous devez identifier cette valeur grâce aux données techniques de votre application (rapport entre le gain proportionnel des actionneurs (par exemple, le rapport entre la puissance de chauffage et la puissance de refroidissement maximum des actionneurs)) et l'affecter au paramètre "CoolFactor". Un facteur de chauffage/refroidissement de 2,0 signifie que le dispositif de chauffage est deux fois plus puissant que le dispositif de refroidissement. Si vous utilisez un facteur de refroidissement, PID_Temp calcule le signal de sortie, et selon son signe, il multiplie le signal de sortie par le facteur de chauffage/refroidissement (lorsque le signe est négatif), ou non (lorsque le signe est positif).

Utilisation de deux jeux de paramètres PID

Différents jeux de paramètres PID pour le chauffage et le refroidissement peuvent être automatiquement détectés pendant la mise en service. Vous pouvez escompter une meilleure performance de la commande, en comparaison du facteur de chauffage/refroidissement car, en plus de différents gains proportionnels, vous pouvez envisager différentes temporisations avec deux jeux de paramètres. Toutefois, l'inconvénient de cette méthode est que le processus d'optimisation peut durer plus longtemps. Si la commutation des paramètres PID est activée (Config.AdvancedCooling = TRUE), le régulateur PID_Temp détecte en "mode automatique" (le contrôle est actif) s'il est nécessaire de chauffer ou de refroidir à ce moment et il utilise les jeux de paramètres PID pour la commande.

ControlZone

Le régulateur PID_Temp vous permet de définir une zone de régulation pour chaque jeu de paramètres au paramètre "ControlZone". Si l'écart de régulation (consigne - mesure) se trouve dans la zone de régulation, PID_Temp utilise l'algorithme PID pour calculer les signaux de sortie. Cependant, si l'écart de régulation dépasse la plage définie, la sortie est mise à la valeur de réglage de chauffage ou de refroidissement maximum (si la sortie de refroidissement est activée) / à la valeur de réglage de chauffage minimum (si la sortie de refroidissement est désactivée). Vous pouvez utiliser cette fonction pour atteindre la consigne souhaitée plus rapidement, notamment pour le chauffage initial dans le cadre de processus thermiques lents.

Zone morte (DeadZone)

Dans le paramètre "DeadZone", vous pouvez définir un intervalle d'écart de régulation pour le chauffage et le refroidissement négligé par l'algorithme PID. Cela signifie que l'écart de régulation est supprimé dans cette plage et que le régulateur PID_Temp réagit comme si la consigne et les valeurs de process étaient identiques. Ainsi, vous pouvez éviter les interventions inutiles du régulateur autour de la consigne et ménager l'actionneur. Si vous voulez appliquer une zone morte, vous devez définir la valeur manuellement. L'auto-optimisation ne fixe pas automatiquement de valeur de zone morte. La zone morte est symétrique (entre -Retain.CtrlParams.Heat.DeadZone et +Retain.CtrlParams.Heat.DeadZone) pour les régulateurs de chauffage sans refroidissement ou les régulateurs de chauffage/refroidissement employant un facteur de refroidissement. La zone morte peut être asymétrique (entre -Retain.CtrlParams.Cool.DeadZone et +Retain.CtrlParams.Heat.DeadZone) pour les régulateurs de chauffage/refroidissement employant deux jeux de paramètres PID.

Fonctionnement du régulateur PID_Temp

Les schémas fonctionnels suivants illustrent le fonctionnement standard et en cascade de l'instruction PID_Temp :

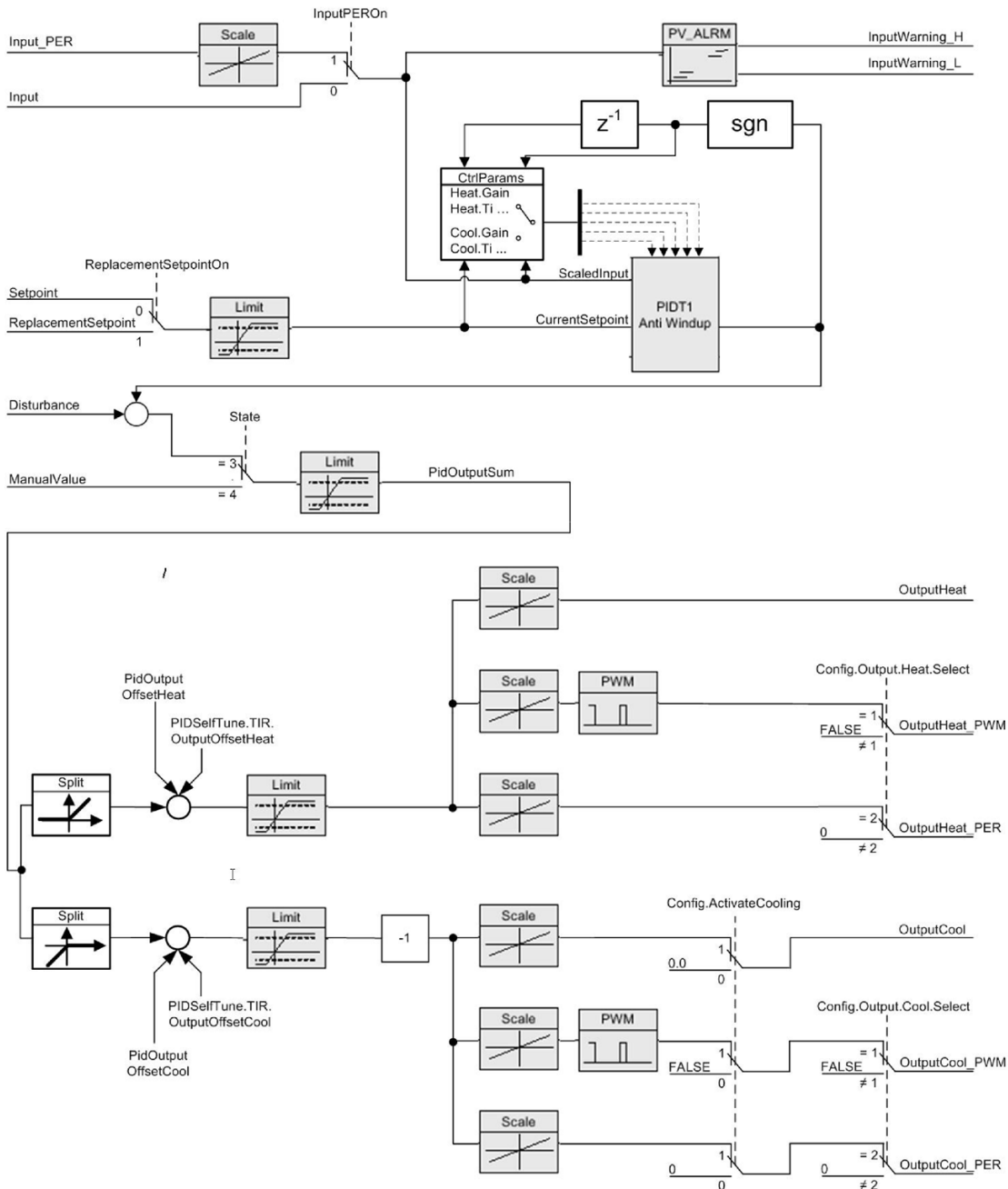


Figure 10-6 PID_Temp_Operation_Block_Diagram

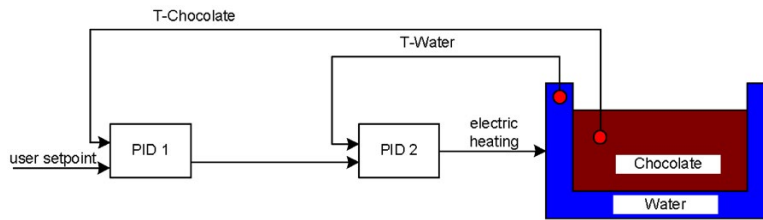


Figure 10-7 PID_Temp_Cascade_Operation_Block_Diagram

Régulateurs en cascade

Vous pouvez mettre en cascade des régulateurs thermiques PID pour traiter plusieurs températures qui dépendent du même actionneur.

Ordre d'appel

Vous devez appeler des régulateurs PID mis en cascade dans le même cycle d'OB. D'abord, il vous faut appeler le maître, puis le ou les esclave(s) suivant(s) dans le flux des signaux de commande pour finir au dernier esclave de la cascade. L'instruction PID_Temp n'effectue pas de vérification automatique de l'ordre d'appel.

Liaisons de communication

Lorsque vous mettez en cascade des régulateurs, vous devez connecter le maître à l'esclave de telle sorte qu'ils puissent partager des informations. Vous devez connecter le paramètre IN/OUT "maître" d'un esclave au paramètre IN/OUT "esclave" de son maître dans le sens du flux des signaux.

La figure ci-dessous représente le montage de régulateurs PID_Temp en cascade avec deux sous-cascades : "PID_Temp1" fournit la consigne. La configuration connecte les sorties de "PID_Temp2", "PID_Temp3", "PID_Temp5", "PID_Temp6", et "PID_Temp8" au processus :

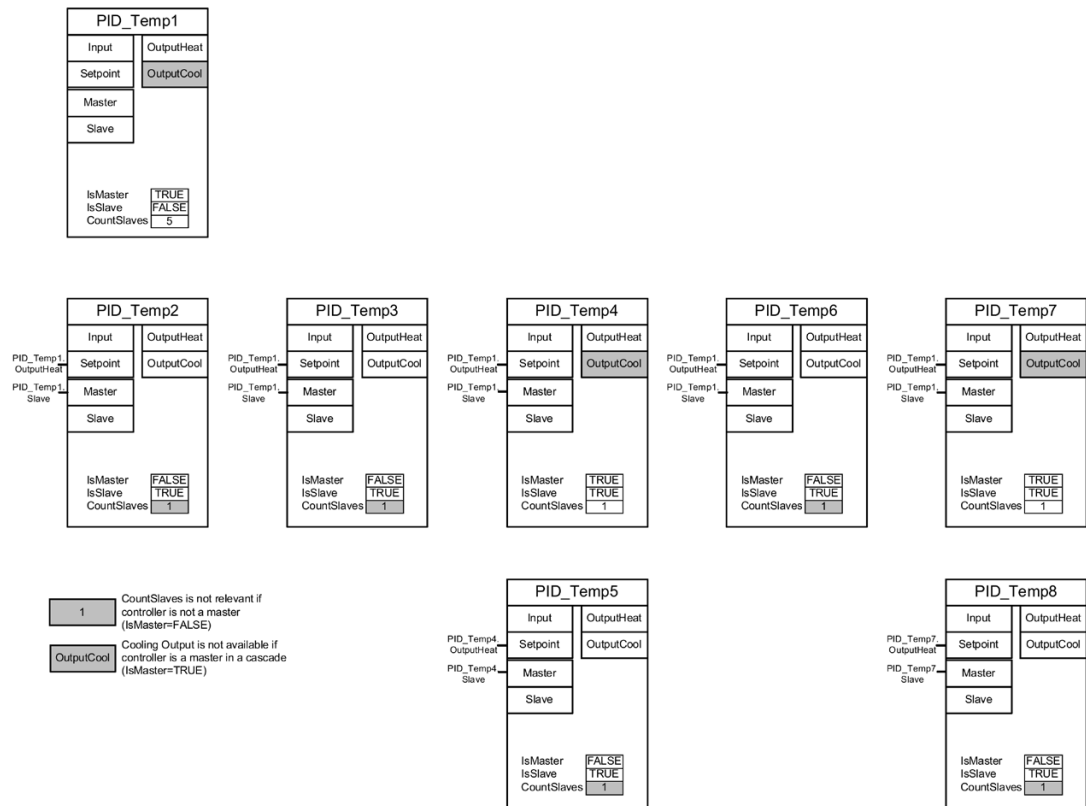


Figure 10-8 PID_Temp_Cascading_communication_connection

Consigne de remplacement

L'instruction PID_Temp fournit une deuxième entrée de consigne au paramètre "ReplacementSetpoint" que vous pouvez activer en réglant le paramètre "ReplacementSetpointOn" = TRUE. Vous pouvez utiliser "ReplacementSetpoint" comme votre entrée de consigne pendant la mise en service ou l'optimisation d'un régulateur esclave sans être obligé de couper la connexion mesure-consigne entre le maître et l'esclave. Cette connexion est nécessaire au fonctionnement normal de la cascade.

Par conséquent, vous n'êtes pas obligé de modifier votre programme et de le télécharger si vous voulez séparer temporairement un esclave de son maître. Vous devez seulement activer "ReplacementSetpoint" et la désactiver ensuite lorsque vous avez terminé. La consigne s'applique à l'algorithme PID lorsque vous pouvez voir sa valeur au paramètre "CurrentSetpoint".

Auto-optimisation

L'auto-optimisation d'un régulateur maître mis en cascade doit remplir les conditions suivantes :

- La mise en service doit être effectuée à partir de l'esclave interne vers le premier maître.
- Tous les esclaves du maître doivent être en "mode automatique".
- La sortie du maître doit servir de consigne aux esclaves.

L'instruction PID_Temp apportera l'aide suivante pour l'auto-optimisation dans la cascade :

- Si vous démarrez l'auto-optimisation pour un régulateur maître, celui-ci vérifie que tous les esclaves sont en "mode automatique" et vérifie la désactivation de la fonction consigne de remplacement pour tous les esclaves ("ReplacementSetpointOn" = FALSE). Si ces conditions ne sont pas remplies, vous ne pouvez pas auto-optimiser le maître. Le maître annule l'optimisation, passe en "mode inactif" (si ActivateRecoverMode" = FALSE) ou revient dans le mode enregistré dans le paramètre "Mode" (si "ActivateRecoverMode" = TRUE). Le maître affiche le message d'erreur 200000hex ("Erreur dans un maître de la cascade. Les esclaves ne sont pas en mode automatique ou ont une consigne de remplacement activée, et empêchent l'optimisation du maître").
- Lorsque tous les esclaves sont en "mode automatique", le système règle le paramètre "AllSlaveAutomaticState" sur TRUE. Vous pouvez appliquer ce paramètre dans vos programmes ou localiser la cause de l'erreur 200000hex.
- Lorsque "ReplacementSetpoint" est désactivé pour tous les esclaves, le système règle le paramètre "NoSlaveReplacementSetpoint" sur TRUE. Vous pouvez appliquer ce paramètre dans leurs programmes ou localiser la cause de l'erreur 200000hex.

Utilisez la boîte de dialogue de mise en service de l'instruction PID_Temp pour bénéficier d'une assistance supplémentaire pour l'optimisation de la cascade (Page 663).

Etats de fonctionnement et gestion des erreurs

Le régulateur PID_Temp n'autorise pas son maître ou ses esclaves à commuter l'état de fonctionnement. Cela signifie qu'un maître d'une cascade reste dans son état actuel lorsqu'un esclave signale une erreur. C'est un avantage si deux esclaves parallèles ou plus fonctionnent avec ce régulateur maître ; une erreur dans un segment n'entraîne pas l'arrêt du segment parallèle.

De même, un esclave d'une cascade reste dans son état de fonctionnement actuel si son maître a une erreur. Toutefois, le fonctionnement ultérieur de l'esclave dépend de la configuration du maître, car la consigne de l'esclave correspond à la sortie du maître. Par conséquent, si vous avez configuré "ActivateRecoverMode" = TRUE sur le maître et qu'une erreur survient, le maître émet la dernière valeur de réglage valide ou une valeur de réglage de remplacement comme consigne pour l'esclave. Si vous avez configuré "ActivateRecoverMode" = FALSE sur le maître, le maître passe en "mode inactif" et met toutes les sorties à "0,0" de telle sorte que l'esclave utilise "0,0" comme consigne.

C'est parce que seuls les régulateurs esclaves ont un accès direct aux actionneurs et que ces derniers restent dans leur état de fonctionnement en cas d'erreur d'un maître, que vous pouvez éviter des dommages au processus. Par exemple, pour les appareils de traitement du plastique, il est catastrophique que les esclaves cessent de travailler, arrêtent les actionneurs et permettent au plastique de durcir dans l'appareil uniquement parce que le régulateur maître a subi une erreur.

Anti-emballement

Un esclave d'une cascade reçoit sa consigne de la sortie de son maître. Si l'esclave atteint ses propres limites de réglage alors que le maître détecte toujours un écart de régulation (consigne - mesure), le maître bloque ou réduit sa contribution d'intégration afin d'éviter le phénomène dit d'emballement. En cas d'emballement, le maître augmente considérablement sa contribution d'intégration et il doit commencer par la réduire avant que le régulateur puisse retrouver une réaction normale. Un "emballement" de ce type nuit à la dynamique de la commande. PID_Temp fournit des moyens d'éviter cet effet dans une cascade par la configuration du paramètre Config.Cascade.AntiWindUpMode" du régulateur maître :

Valeur	Description
0	Désactive la fonction anti-emballement.
1	Réduit la contribution d'intégration du régulateur maître au rapport entre "esclaves en limitation" et "esclaves existants" (paramètre "CountSlaves").
2	Bloque la contribution d'intégration du maître dès qu'un esclave atteint sa limite. N'est pertinent que si "Config.Cascade.IsMaster" = TRUE.

10.2.4.2 Paramètres ErrorBit de l'instruction PID_Temp

Si le régulateur PID a plusieurs alertes en attente, il affiche les valeurs des codes d'erreur au moyen d'une addition binaire. Ainsi, l'affichage du code d'erreur 0003 signifie, par exemple, que les erreurs 0001 et 0002 sont en attente.

Tableau 10- 28 Paramètres ErrorBit de l'instruction PID_Temp

ErrorBit (DW#16#...)	Description
0000	Pas d'erreur
0001 ^{1, 2}	Le paramètre Input se situe en dehors des limites de la mesure. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Valeur invalide pour le paramètre Input_PER. Vérifiez s'il y a une erreur en attente au niveau de l'entrée analogique.
0004 ⁴	Erreur pendant l'optimisation fine. L'oscillation de la mesure n'a pas pu être conservée.
0008 ⁴	Erreur au démarrage de l'optimisation préalable. La mesure est trop proche de la consigne. Démarrez l'optimisation fine.
0010 ⁴	La consigne a été modifiée pendant l'optimisation. Note : Vous pouvez définir la fluctuation permise sur la consigne à la variable CancelTuningLevel.
0020	L'optimisation préalable n'est pas permise pendant l'optimisation fine. Note : Si ActivateRecoverMode à la valeur TRUE avant que l'erreur ne survienne, PID_Temp reste en mode optimisation fine.
0040 ⁴	Erreur pendant l'optimisation préalable. Le refroidissement n'a pas pu réduire la valeur de process.
0080 ⁴	Erreur pendant l'optimisation préalable. Configuration incorrecte des limites de la valeur de réglage. Vérifiez si les limites de la valeur de réglage sont configurées correctement et correspondent à la logique du régulateur.
0100 ⁴	Une erreur pendant l'optimisation fine a entraîné des paramètres invalides.
0200 ^{2, 3}	Valeur invalide pour le paramètre Input. La valeur a un format numérique invalide.
0400 ^{2, 3}	Le calcul de la valeur de réglage a échoué. Vérifiez les paramètres PID.
0800 ^{1, 2}	Erreur de période d'échantillonnage : PID_Temp n'est pas appelé dans la période d'échantillonnage de l'OB d'alarme cyclique.
1000 ^{2, 3}	Valeur invalide pour le paramètre Setpoint. La valeur a un format numérique invalide.
10000	Valeur invalide pour le paramètre ManualValue. La valeur a un format numérique invalide. Note : Si ActivateRecoverMode à la valeur TRUE avant que l'erreur ne survienne, PID_Temp utilise SubstituteOutput comme valeur de réglage. Dès que vous affectez une valeur invalide dans le paramètre ManualValue, PID_Temp l'utilise comme valeur de réglage.

ErrorBit (DW#16#...)	Description
20000	Valeur invalide pour la variable SubstituteValue : La valeur a un format numérique invalide. PID_Temp utilise la limite inférieure de la valeur de réglage comme valeur de réglage. Remarque : Si le mode automatique était actif avant l'apparition de l'erreur, ActivateRecoverMode a la valeur TRUE et l'erreur n'est plus en attente, PID_Temp repasse en mode automatique.
40000	Valeur invalide pour le paramètre Disturbance. La valeur a un format numérique invalide. Note : Si le mode automatique était actif et si ActivateRecoverMode a la valeur FALSE avant l'apparition de l'erreur, Disturbance est défini sur zéro. PID_Temp reste en mode automatique. Remarque : Si l'optimisation préalable ou l'optimisation fine était active et ActivateRecoverMode a la valeur TRUE avant que l'erreur ne se produise, PID_Temp passe à l'état de fonctionnement qui est enregistré pour le paramètre Mode. Si une perturbation dans la phase actuelle n'a pas d'effet sur la valeur de réglage, l'optimisation n'est pas annulée.
200000	Erreur dans un maître de la cascade. Les esclaves ne sont pas en mode automatique ou ont une consigne de remplacement activée, ce qui empêche l'optimisation du maître.
400000	Le régulateur PID ne permet pas l'optimisation préalable pour le chauffage lorsque le refroidissement est actif.
800000	La valeur de process doit être proche de la consigne pour démarrer l'optimisation préalable pour le refroidissement.
1000000	Erreur au démarrage de l'optimisation. "Heat.EnableTuning" et "Cool.EnableTuning" ne sont pas à 1 ou ne correspondent pas à la configuration.
2000000	L'optimisation préalable pour le refroidissement requiert la réussite de l'optimisation préalable pour le chauffage.
4000000	Erreur au démarrage de l'optimisation fine. "Heat.EnableTuning" et "Cool.EnableTuning" ne peuvent pas être à 1 en même temps.
8000000	Une erreur pendant le calcul des paramètres PID a entraîné des paramètres invalides (par exemple un gain négatif, les paramètres PID actuels ne sont pas modifiés et l'optimisation n'a pas d'effet).

- ¹ Note : Si l'état automatique était actif avant que l'erreur ne se produise et ActivateRecoverMode a la valeur TRUE, PID_Temp reste en mode automatique.
- ² Note : Si l'optimisation préalable ou l'optimisation fine était active avant que l'erreur ne se produise et si ActivateRecoverMode a la valeur TRUE, PID_Temp passe à l'état de fonctionnement qui est enregistré pour le paramètre Mode.
- ³ Note : Si le mode automatique était actif avant que l'erreur ne se produise et si ActivateRecoverMode a la valeur TRUE, PID_Compact délivre la valeur de réglage de remplacement configurée. Dès que l'erreur n'est plus en attente, PID_Temp repasse en mode automatique.
- ⁴ Note : Si ActivateRecoverMode a la valeur TRUE avant que l'erreur ne se produise, PID_Temp annule l'optimisation et passe à l'état de fonctionnement qui est enregistré pour le paramètre Mode.

10.2.4.3 Paramètres Warning de l'instruction PID_Temp

Si le contrôleur PID a plusieurs avertissements en attente, il affiche les valeurs des codes d'erreur au moyen d'une addition binaire. L'affichage du code d'erreur 0003, par exemple, indique que les erreurs 0001 et 0002 sont en attente.

Tableau 10- 29 Instruction PID_Temp, paramètres Warning

Avertissement (DW#16#...)	Description
0000	Aucun avertissement en attente.
0001 ¹	Le point d'inflexion n'a pas été trouvé pendant le pré-réglage.
0002	L'oscillation a été appliquée pendant le "réglage en cours". (Le paramètre "Warning" supprime cet avertissement et est uniquement visible dans le paramètre "WarningInternal" à des fins de diagnostic.)
0004 ¹	Le point de consigne a été limité selon les limites configurées.
0008 ¹	Toutes les propriétés de système contrôlées nécessaires n'ont pas été définies pour la méthode de calcul sélectionnée. À l'inverse, les paramètres PID ont été calculés en utilisant la méthode TIR.TuneRuleHeat / TIR Tune-RuleCool = 3.
0010	Le mode de fonctionnement ne pourrait pas être changé car Reset = VRAI ou ManualEnable = VRAI.
0020	Le temps de cycle de l'OB appelant limite le temps d'échantillonnage de l'algorithme PID. Améliorez les résultats en utilisant des temps de cycle OB plus courts.
0040 ¹	La valeur du processus a dépassé l'une de ses limites d'avertissement.
0080	Valeur invalide au niveau du mode. Le mode de fonctionnement n'est pas activé.
0100 ¹	La valeur manuelle a été limitée selon les limites de la sortie du contrôleur.
0200	La règle spécifiée pour le réglage n'est pas supportée. Aucun paramètre PID n'est calculé.
1000	La valeur de sortie de remplacement ne peut pas être atteinte car elle se situe en dehors des limites de la valeur de sortie.
4000	La sélection de sortie spécifiée pour le chauffage et/ou le refroidissement n'est pas supportée. Uniquement OutputHeat et OutputCool sont actifs.
8000	La valeur spécifiée pour le paramètre PIDSelfTune.SUT.AdaptDelayTime n'est pas supportée, la valeur par défaut "0" est donc utilisée.
10000	La valeur spécifiée pour le paramètre PIDSelfTune.SUT.CoolingMode n'est pas supportée, la valeur par défaut "0" est donc utilisée.

¹ Remarque : Le contrôleur PID a automatiquement supprimé les avertissements suivants dès que la cause a été éliminée ou que l'action de l'utilisateur a été répétée avec les paramètres valides : 0001, 0004, 0008, 0040 et 0100.

10.2.5 Configuration des contrôleurs PID_Compact et PID_3Step

Les paramètres de l'objet technologique déterminent le fonctionnement du régulateur PID. Utilisez l'icône pour ouvrir l'éditeur de configuration.

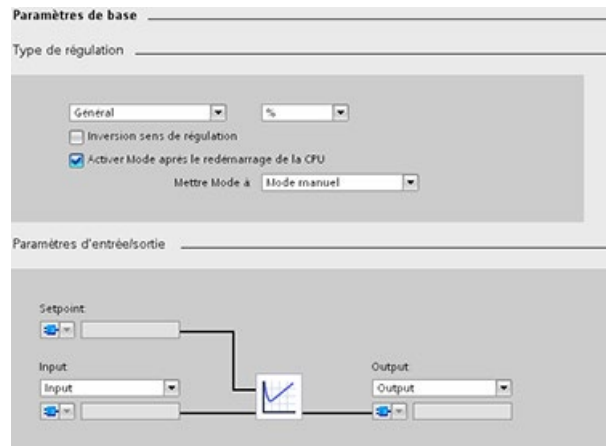


Tableau 10- 30 Exemple de paramètres de configuration pour l'instruction PID_Compact

Paramètres	Description	
Paramètres de base	Type de régulation	Sélectionne les unités physiques.
	Inversion du sens de régulation	Permet la sélection d'une boucle PID à action inverse. <ul style="list-style-type: none"> Si la case n'est pas cochée, la boucle PID est à action directe et la valeur de réglage de la boucle PID augmente si la mesure est inférieure à la consigne. Si cette case est cochée, la valeur de réglage de la boucle PID augmente si la mesure est supérieure à la consigne.
	Activer le dernier mode après redémarrage de la CPU	Redémarre la boucle PID après sa réinitialisation ou si une limite de mesure a été dépassée et est revenue dans la plage valide.
	Mesure	Sélectionne soit le paramètre Input, soit le paramètre Input_PER (analogique) pour la mesure. Input_PER peut provenir directement d'un module d'entrées analogiques.
	Valeur de réglage	Sélectionne soit le paramètre Output, soit le paramètre Output_PER (analogique) pour la valeur de réglage. Output_PER peut aller directement à un module de sorties analogiques.
Mesure	Met à l'échelle la plage ainsi que les limites pour la mesure. Si la mesure devient inférieure à la limite inférieure ou supérieure à la limite supérieure, la boucle PID passe en mode inactif et met la valeur de réglage à 0. Pour utiliser Input_PER, vous devez mettre à l'échelle la mesure analogique.	

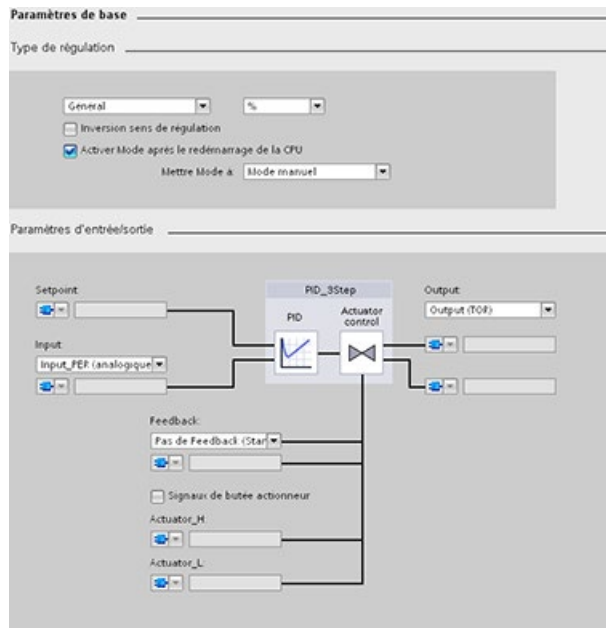


Tableau 10- 31 Exemple de paramètres de configuration pour l'instruction PID_3Step

Paramètres		Description
basiques	Type de régulateur	Sélectionne les unités physiques.
	Inverse le sens de régulation	Permet la sélection d'une boucle PID à action inverse. <ul style="list-style-type: none"> • Si la case n'est pas cochée, la boucle PID est à action directe et la valeur de réglage de la boucle PID augmente si la mesure est inférieure à la consigne. • Si cette case est cochée, la valeur de réglage de la boucle PID augmente si la mesure est supérieure à la consigne.
	Activer le mode après le démarrage de la CPU	Redémarre la boucle PID après sa réinitialisation ou si une limite de mesure a été dépassée et est revenue dans la plage valide. Paramétrer le mode sur : Définit le mode dans lequel l'utilisateur veut que le PID aille après le redémarrage.
	Entrée	Sélectionne soit le paramètre Input, soit le paramètre Input_PER (analogique) pour la mesure. Input_PER peut provenir directement d'un module d'entrées analogiques.
	Sortie	Sélectionne soit l'utilisation de sorties TOR (Output_UP et Output_DN), soit l'utilisation d'une sortie analogique (Output_PER) pour la valeur de réglage.
	Rétroaction	Sélectionne le type d'état renvoyé par la boucle PID : <ul style="list-style-type: none"> • Pas de rétroaction (valeur par défaut) • Rétroaction • Feedback_PER
Mesure	Met à l'échelle la plage ainsi que les limites pour la mesure. Si la mesure devient inférieure à la limite inférieure ou supérieure à la limite supérieure, la boucle PID passe en mode inactif et met la valeur de réglage à 0. Pour utiliser Input_PER, vous devez mettre à l'échelle la mesure analogique (valeur d'entrée).	

Paramètres		Description
Actionneur	Temps de transition du moteur	Définit le temps nécessaire pour passer de l'état ouvert à l'état fermé de la vanne (vous trouverez cette valeur sur la fiche technique ou la plaque signalétique de la vanne).
	Temps d'activation minimum	Définit le temps de mouvement minimum de la vanne (vous trouverez cette valeur sur la fiche technique ou la plaque signalétique de la vanne).
	Temps de désactivation minimum	Définit le temps de pause minimum de la vanne (vous trouverez cette valeur sur la fiche technique ou la plaque signalétique de la vanne).
	Réaction à l'erreur	Définit le comportement de la vanne lorsqu'une erreur est détectée ou que la boucle PID est réinitialisée. Si vous optez pour l'utilisation d'une position de substitution, entrez la "Position de sécurité". Pour une rétroaction ou une sortie analogique, sélectionnez une valeur entre la limite supérieure et la limite inférieure pour la valeur de réglage. Pour des sorties TOR, vous pouvez choisir 0% (désactivé) ou 100% (activé).
	Mise à l'échelle de la rétroaction de position ¹	<ul style="list-style-type: none"> • "Butée supérieure" et "Butée inférieure" définissent respectivement la position positive maximale (entièrement ouvert) et la position négative maximale (entièrement fermé). "Butée supérieure" doit être supérieure à "Butée inférieure". • "Limite supérieure de la mesure" et "Limite inférieure de la mesure" définissent les positions supérieure et inférieure de la vanne pendant le réglage et le mode automatique. • "FeedbackPER" ("Low" et "High") définit la rétroaction analogique de la position de la vanne. "FeedbackPER High" doit être supérieur à "FeedbackPER Low".
Paramètres avancés	Surveillance de la mesure	Définit les limites d'avertissement inférieure et supérieure pour la mesure.
	Paramètres PID	Si l'utilisateur le souhaite, il peut entrer ses propres paramètres d'optimisation PID dans cette fenêtre. La case "Activer l'entrée manuelle" doit être cochée pour permettre cela.

¹ "Mise à l'échelle de la rétroaction de position" ne peut être modifié que si vous avez activé "Rétroaction" dans les paramètres de base.

10.2.6 Configuration du régulateur PID_Temp

Les paramètres de l'objet technologique déterminent le fonctionnement du régulateur PID. Utilisez l'icône pour ouvrir l'éditeur de configuration.

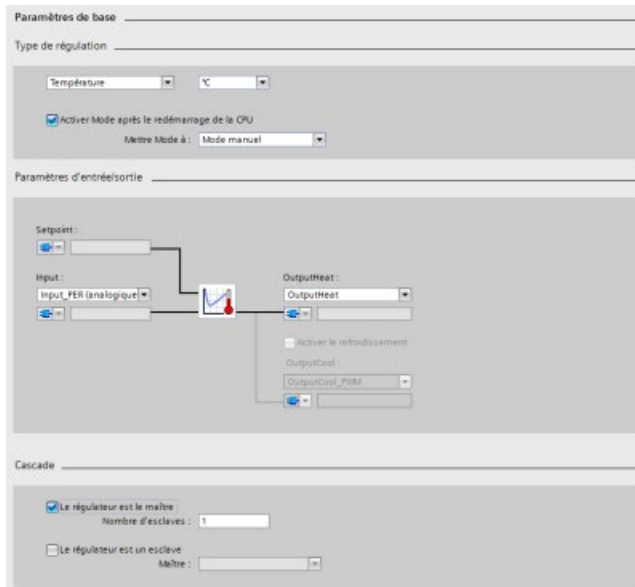


Tableau 10- 32 Exemple de paramètres de configuration pour l'instruction PID_Temp

Paramètres		Description
basiques	Type de régulateur	Sélectionne les unités physiques.
	Activer le mode après le démarrage de la CPU	Redémarre la boucle PID après sa réinitialisation ou si une limite de mesure a été dépassée et est revenue dans la plage valide. Paramétrer le mode sur : Définit le mode dans lequel l'utilisateur veut que le PID aille après le redémarrage.
	Entrée	Sélectionne soit le paramètre Input, soit le paramètre Input_PER (analogique) pour la mesure. Input_PER peut provenir directement d'un module d'entrées analogiques.
	Sortie chauffage	Sélectionne soit l'utilisation de sorties TOR (OutputHeat et OutputHeat_PWM), soit l'utilisation d'une sortie analogique (OutputHeat_PER (analog)) pour la valeur de réglage.
	Sortie Refroidissement	Sélectionne soit l'utilisation de sorties TOR (OutputCool et OutputCool_PWM), soit l'utilisation d'une sortie analogique (OutputCool_PER (analog)) pour la valeur de réglage.
Mesure	Met à l'échelle la plage ainsi que les limites pour la mesure. Si la mesure devient inférieure à la limite inférieure ou supérieure à la limite supérieure, la boucle PID passe en mode inactif et met la valeur de réglage à 0. Pour utiliser Input_PER, vous devez mettre à l'échelle la mesure analogique (valeur d'entrée).	
Cascade	Le régulateur est un maître	Définit le régulateur comme maître et sélectionne le nombre d'esclaves.
	Le régulateur est un esclave.	Définit le régulateur comme esclave et sélectionne le nombre de maîtres.

Type de régulateur

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Quantité physique	"PhysicalQuantity"	Entier (énum)	<ul style="list-style-type: none"> Général Température (=par défaut) 	Pré-sélection de la valeur de l'unité physique Pas de commande multi-valeur et pas de possibilité d'édition dans le mode en ligne de la vue fonctionnelle.
Unité de mesure	"PhysicalUnit"	Entier (énum)	<ul style="list-style-type: none"> Général : Unités = % Température : Unités (sélections possibles) = <ul style="list-style-type: none"> °C (= par défaut) °F K 	La sélection personnalisée de l'unité est remise à "0" si vous modifiez la quantité physique.
Activer le mode après le démarrage de la CPU	"RunModeByStartup"	Booleen	Case à cocher	Si TRUE est activé (= réglage par défaut), le régulateur passe à l'état enregistré dans la variable "Mode" après un cycle d'alimentation (alimentation allumée, éteinte puis rallumée) ou une commutation ARRÊT-DEMARRAGE de l'API. En cas contraire, PID_Temp reste en mode "inactif".
Paramétrer le mode sur	"Mode"	Entier (énum)	Modes (sélections possibles) : <ul style="list-style-type: none"> 0: Inactif 1: Optimisation préalable 2: Optimisation fine 3: Mode automatique 4: Mode manuel (=valeur par défaut) 	La station d'ingénierie (ES) définit la valeur de démarrage de la variable "Mode" selon la sélection de l'utilisateur. La valeur par défaut de "Mode" (enregistrée dans TO-DB) est Mode manuel.

Paramètres d'entrée/de sortie

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Consigne	Consigne	(Nombre réel)	Nombre réel	Accessible uniquement sur la page Propriétés. Pas de commande multi-valeur en mode en ligne de la vue fonctionnelle.
Sélection de l'entrée	"Config.InputPerOn"	Booléen (énum)	Booléen	Sélectionne le type d'entrée à utiliser. Sélections possibles : <ul style="list-style-type: none"> FALSE : "Entrée" (nombre réel) TRUE : "Input_PER (analogique)"
Entrée	Input ou Input_PER	Réel ou entier	Réel ou entier	Accessible uniquement sur la page Propriétés. Pas de commande multi-valeur en mode en ligne de la vue fonctionnelle.
Sélection de la sortie (chauffage)	"Config.Output.Heat.Select"	Entier (énum)	2 >= Config.Output.Heat.Select >= 0	Sélectionne le type de sortie à utiliser pour le chauffage. Sélections possibles : <ul style="list-style-type: none"> "OutputHeat" (nombre réel) "OutputHeat_PWM" (Booléen) (=par défaut) "OutputHeat_PER (analogique)" (mot) Est réglé une fois sur "OutputHeat" si la case "Ce régulateur est un maître" est cochée par l'utilisateur dans la partie "Cascade".
Sortie (chauffage)	OutputHeat, OutputHeat_PER, ou OutputHeat_PWM	Réel ou entier ou booléen	Réel, entier ou booléen	Accessible uniquement sur la page Propriétés. Pas de commande multi-valeur en mode en ligne de la vue fonctionnelle.

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Activer sortie (refroidissement)	"Config.ActivateCooling"	Booléen	Booléen	<p>Cocher cette case :</p> <ul style="list-style-type: none"> • règle "Config.Output.Heat.PidLowerLimit" sur 0,0 une fois. • règle le paramètre "Config.ActivateCooling" sur TRUE, au lieu de FALSE si la case n'est pas cochée (=valeur par défaut). • Active toutes les autres commandes "Sortie (refroidissement)" (dans "Paramètres de base" et d'autres vues). • Fait passer la ligne allant du symbole PID aux commandes de gris à noir. • La case "Ce régulateur est un maître" est décochée dans la partie Cascade. <p>Remarque : Disponible seulement si vous ne configurez pas le régulateur comme un maître d'une cascade (la case "Ce régulateur est un maître" est décochée dans la partie "Cascade" ; "Config.Cascade.IsMaster" = FALSE).</p>

Para- mètre	Paramètre TO-DB	Type de don- nées	Plage de valeurs	Description
Sélection de la sortie (refroidissement)	"Config.Output.Cool.Select"	Entier (énum)	2 >= Config.Output.Heat.Select >= 0	Sélectionne le type de sortie à utiliser pour le refroidissement. Sélections possibles : <ul style="list-style-type: none"> • "OutputCool" (nombre réel) • "OutputCool_PWM" (Booléen) (=par défaut) • "OutputCool_PER (analogique)" (mot) Disponible seulement si vous cochez "Activer sortie (refroidissement)" ; (Config.ActivateCooling = TRUE).
Sortie (refroidissement)	OutputCool, OutputCool_PER, ou OutputCool_PWM	Réel ou entier ou booléen	Réel, entier ou booléen	Accessible uniquement sur la page Propriétés. Pas de commande multi-valeur en mode en ligne de la vue fonctionnelle.

Paramètres de la cascade

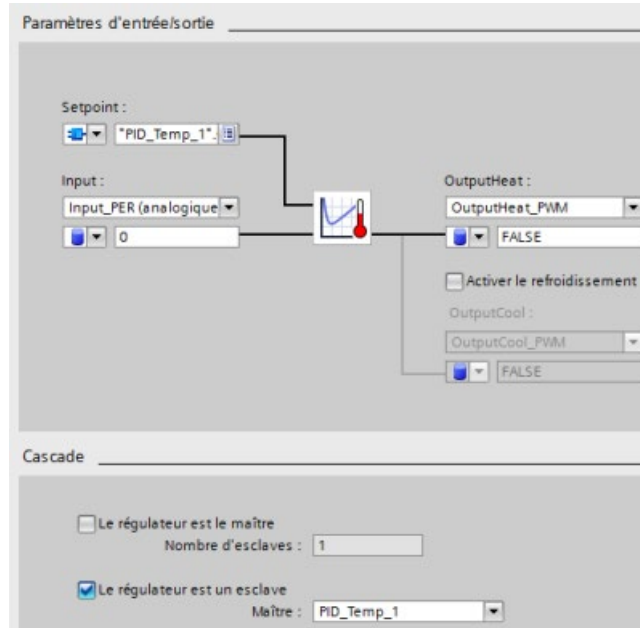
Les paramètres suivants vous permettent de sélectionner des régulateurs comme maîtres ou esclaves et de déterminer le nombre de régulateurs esclaves recevant leur consigne directement du régulateur maître :

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Ce régulateur est un maître	"Config.Cascade.IsMaster"	Booléen	Booléen	<p>Montre si ce régulateur est un maître dans une cascade. Lorsque vous cochez cette case, vous effectuez ce qui suit :</p> <ul style="list-style-type: none"> régler le paramètre "Config.Cascade.IsMaster" sur TRUE, au lieu de FALSE si la case n'est pas cochée (=valeur par défaut). régler une fois "Sélection de la sortie (chauffage)" sur "OutputHeat" dans la partie "Paramètres d'entrée/sortie" (Config.Output.Heat.Select = 0). activer le champ de saisie "Nombre d'esclaves". décocher la case "Activer sortie (refroidissement)" dans la partie "Paramètres d'entrée/sortie". <p>Remarque : Disponible seulement si la sortie de refroidissement de ce régulateur est désactivée (la case "Activer sortie (refroidissement)" dans la partie "Paramètres d'entrée/sortie" est décochée (Config.ActivateCooling = FALSE)).</p>

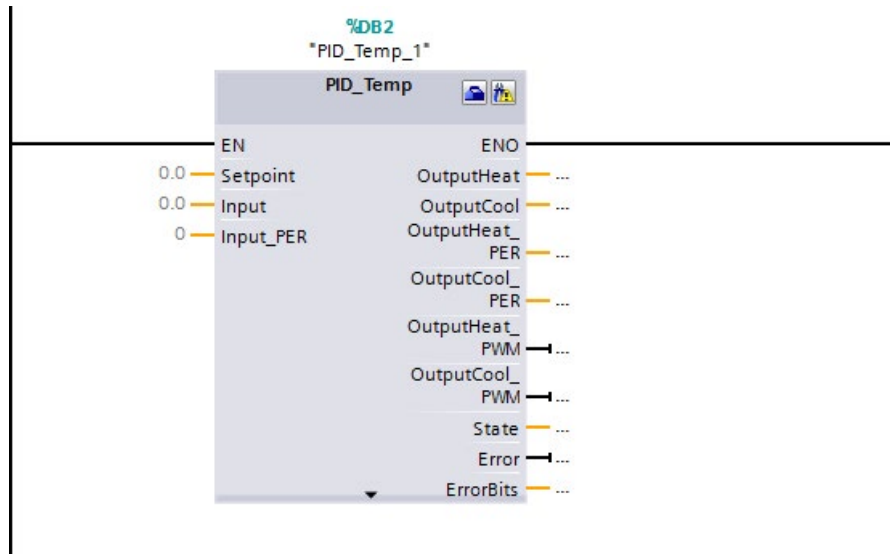
Para- mètre	Paramètre TO-DB	Type de don- nées	Plage de valeurs	Description
Nombre d'es- claves	"Config.Cascade.CountSlaves"	Ent	255 >= Con- fig.Cascade. CountSlaves >= 1	Nombre de régulateurs esclaves recevant leur consigne directement de ce régulateur maître. L'instruction PID_Temp traite cette valeur, ainsi que d'autres, pour la gestion de l'emballement. "Nombre d'esclaves" n'est disponible que si la case "Ce régulateur est un maître" est activée (Config.Cascade.IsMaster = TRUE).
Ce régu- lateur est un es- clave	"Config.Cascade.IsSlave"	Boo- léen	Booléen	Montre si ce régulateur est un esclave dans une cascade. Lorsque vous cochez cette case, vous réglez le paramètre "Config.Cascade.IsSlave" sur TRUE, au lieu de FALSE si la case n'est pas cochée (=valeur par défaut). Vous devez cocher cette case dans la page des propriétés pour activer la liste déroulante "SelectionMaster".

Exemple : Régulateurs en cascade

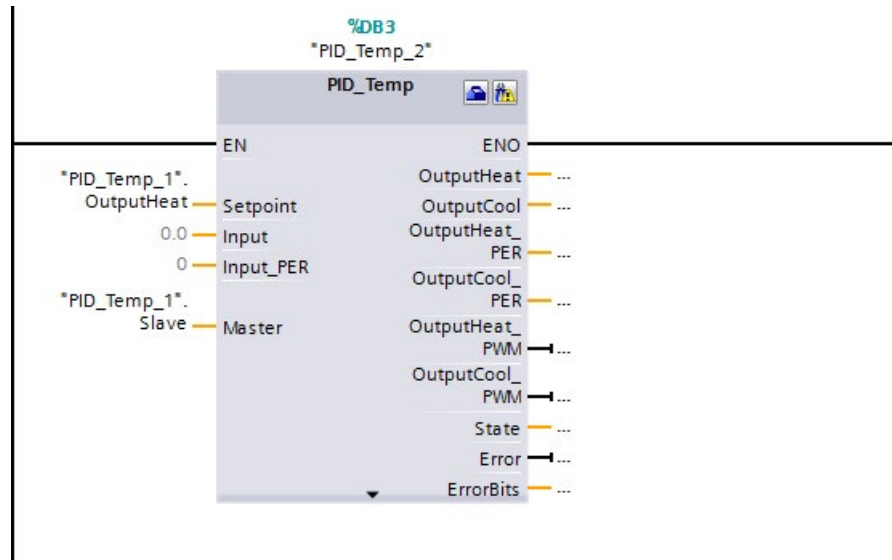
Dans la boîte de dialogue "Paramètres de base" ci-dessous, vous voyez la partie "Paramètres d'entrée/sortie" et la partie "Cascade" pour le régulateur esclave "PID_Temp_2" après avoir sélectionné "PID_Temp_1" comme maître. Vous établissez la connexion entre le régulateur maître et le régulateur esclave :



Réseau 1 : Dans ces réseaux, vous établissez la connexion entre le maître "PID_Temp_1" et l'esclave "PID_Temp_2" dans l'éditeur de programmation :



Réseau 2 : Vous établissez la connexion entre les paramètres "OutputHeat" et "Slave" du maître "PID_Temp_1" et les paramètres "Setpoint" et "Master" de l'esclave "PID_Temp_2", respectivement :



Auto-optimisation de processus thermiques

L'instruction PID_Temp propose deux modes pour l'auto-optimisation :

- "Optimisation préalable" (paramètre "Mode" = 1)
- "Optimisation fine" (paramètre "Mode" = 2)

Selon la configuration du régulateur, différentes variantes de ces méthodes d'optimisation sont disponibles :

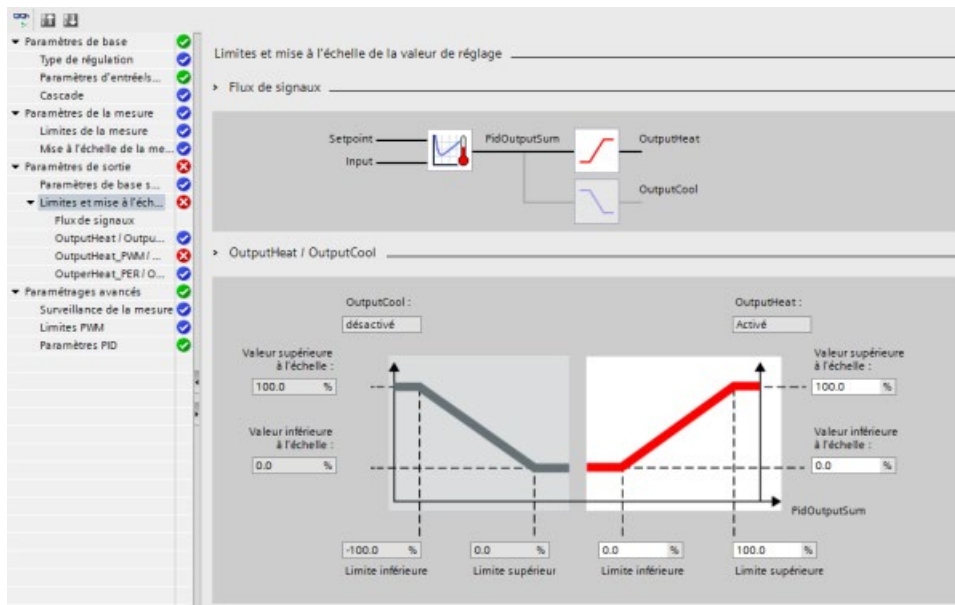
Configuration	Régulateur avec sortie de chauffage	Régulateur avec sortie de chauffage et de refroidissement utilisant un facteur de refroidissement	Régulateur avec sortie de chauffage et de refroidissement utilisant deux jeux de paramètres PID
Valeurs TO-DB associées	<ul style="list-style-type: none"> Config.ActivateCooling = FALSE Config.AdvancedCooling = non pertinent 	<ul style="list-style-type: none"> Config.ActivateCooling = TRUE Config.AdvancedCooling = FALSE 	<ul style="list-style-type: none"> Config.ActivateCooling = TRUE Config.AdvancedCooling = TRUE
Méthodes d'optimisation disponibles	<ul style="list-style-type: none"> "Optimisation préalable du chauffage" "Optimisation fine du chauffage" (le décalage de refroidissement ne peut pas être utilisé) 	<ul style="list-style-type: none"> "Optimisation préalable du chauffage" "Optimisation fine du chauffage" (le décalage de refroidissement peut être utilisé) 	<ul style="list-style-type: none"> "Optimisation préalable du chauffage et du refroidissement" "Optimisation préalable du chauffage" "Optimisation préalable du refroidissement" "Optimisation fine du chauffage" (le décalage de refroidissement peut être utilisé) "Optimisation fine du refroidissement" (le décalage de chauffage peut être utilisé)

Limites et mise à l'échelle des valeurs de réglage

Activation du refroidissement désactivée

Si vous configurez l'instruction PID_Temp comme maître pour une cascade, la case "Activer sortie (refroidissement)" est décochée et désactivée dans la vue "Paramètres de base" et tous les paramètres dans la vue "Paramètres de sortie" dépendant de l'activation du refroidissement sont également désactivés.

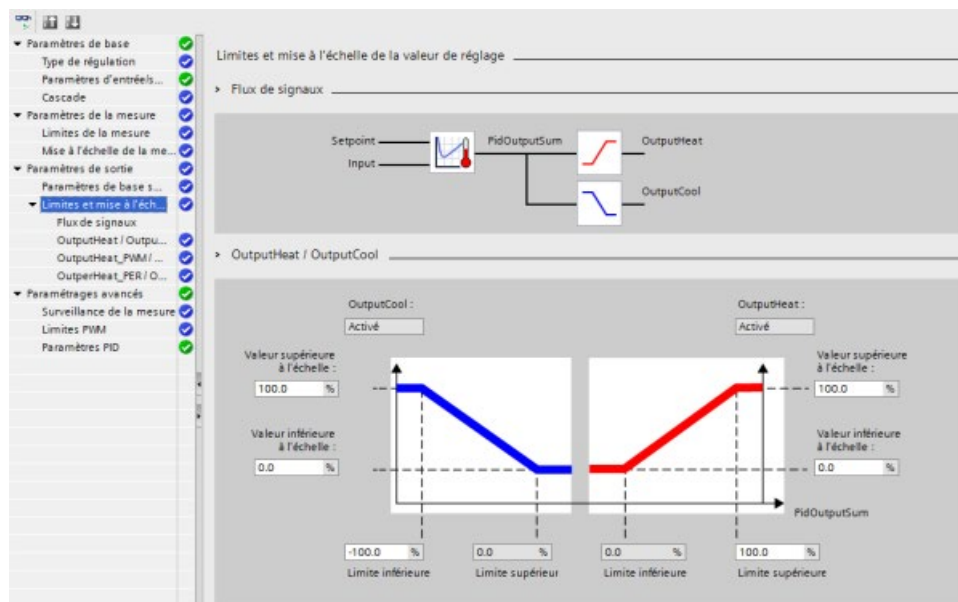
La figure ci-dessous représente la partie "Limites et mise à l'échelle de la valeur de réglage" dans la vue "Paramètres de sortie" avec refroidissement désactivé (OutputHeat_PWM sélectionné dans la vue "Paramètres d'entrée/sortie" et OutputHeat toujours activé) :

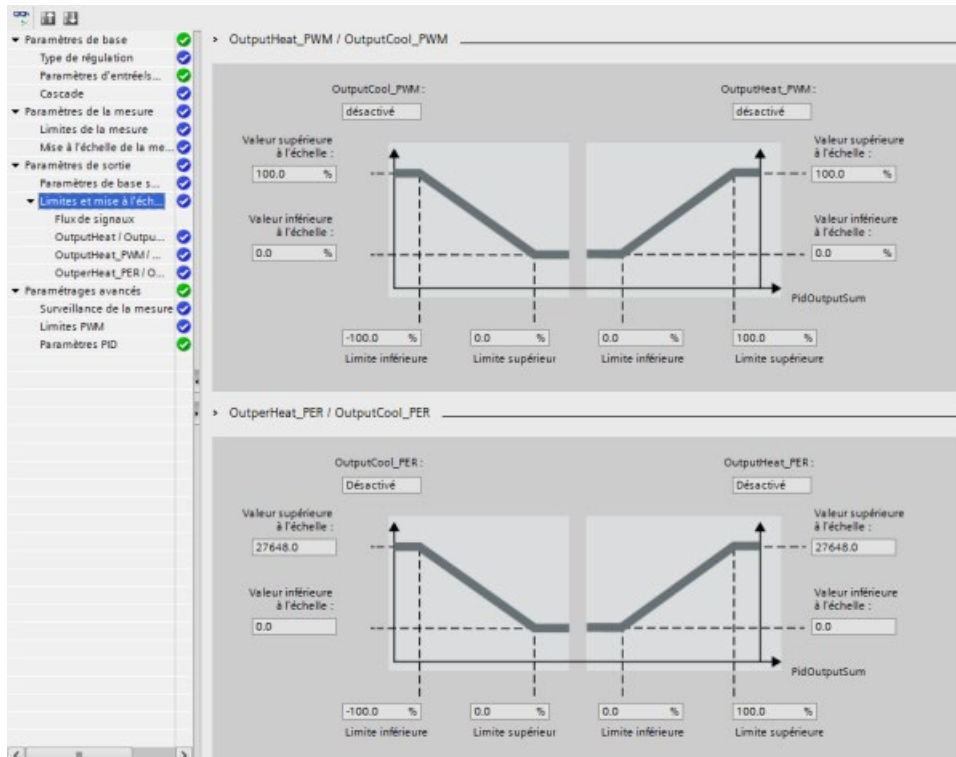




Activation du refroidissement activée

La figure ci-dessous représente la partie "Limites et mise à l'échelle de la valeur de réglage" dans la vue "Paramètres de sortie" avec refroidissement activé (OutputCool_PER et OutputHeat_PWM sélectionné dans la vue "Paramètres d'entrée/sortie" ; OutputCool et OutputHeat toujours activés) :





Etats de fonctionnement

Pour modifier l'état de fonctionnement manuellement, l'utilisateur doit régler le paramètre d'entrée-sortie "Mode" du régulateur et l'activer en faisant passer "ModeActivate" de FALSE à TRUE (front montant déclenché). Vous devez réinitialiser "ModeActivate" avant le prochain changement de mode ; il n'est pas automatiquement réinitialisé.

Le paramètre de sortie "State" montre l'état de fonctionnement actuel et est mis sur le "Mode" requis, si possible. Le paramètre "State" ne peut pas être modifié directement ; il ne peut être modifié que par l'intermédiaire du paramètre "Mode" ou par des changements du mode automatique par le régulateur.

"Mode" / "State"	Nom	Description
0	Inactif	<p>L'instruction PID_Temp :</p> <ul style="list-style-type: none"> désactive l'algorithme PID et la modulation de largeur d'impulsion Met à "0" (FALSE) toutes les sorties du régulateur (OutputHeat, OutputCool, OutputHeat_PWM, OutputCool_PWM, OutputHeat_PER, OutputCool_PER), quels que soient les limites de sortie ou les décalages configurés. Vous pouvez atteindre ce mode en paramétrant "Mode" = 0, "Reset" = TRUE, ou par erreur.
1	Optimisation préalable (optimisation au démarrage / SUT)	<p>Ce mode détermine les paramètres pendant le premier démarrage du régulateur. Contrairement à PID_Compact, pour PID_Temp, vous devez indiquer si vous avez besoin d'une optimisation de chauffage, d'une optimisation de refroidissement ou des deux à l'aide des paramètres "Heat.EnableTuning" et "Cool.EnableTuning".</p> <p>Vous pouvez activer "Pretuning" (optimisation préalable) à partir du Mode inactif, Mode automatique ou Mode manuel.</p> <p>Si l'optimisation est réussie, PID_Temp passe en mode automatique. Si l'optimisation n'a pas réussi, la commutation de l'état de fonctionnement dépend de "ActivateRecoverMode".</p>
2	Optimisation fine (optimisation en cours de fonctionnement / TIR)	<p>Ce mode détermine le paramétrage optimal du régulateur PID à la consigne. Contrairement à PID_Compact, pour PID_Temp, vous devez indiquer si vous avez besoin d'une optimisation de chauffage ou d'une optimisation de refroidissement à l'aide des paramètres "Heat.EnableTuning" et "Cool.EnableTuning".</p> <p>Vous pouvez activer "Finetuning" (optimisation fine) à partir du Mode inactif, Mode automatique ou Mode manuel.</p> <p>Si l'optimisation est réussie, PID_Temp passe en mode automatique. Si l'optimisation n'a pas réussi, la commutation de l'état de fonctionnement dépend de "ActivateRecoverMode".</p>
3	Mode automatique	<p>En mode automatique (le mode de régulation PID standard), le résultat de l'algorithme PID détermine les valeurs de réglage.</p> <p>PID_Temp bascule sur Inactif si une erreur survient et que "ActivateRecoverMode" = FALSE. Si une erreur survient et que "ActivateRecoverMode" = TRUE, la commutation de l'état de fonctionnement dépend de l'erreur. Pour plus d'informations, veuillez vous référer à l'instruction PID_Temp ErrorBit parameters (Page 640).</p>

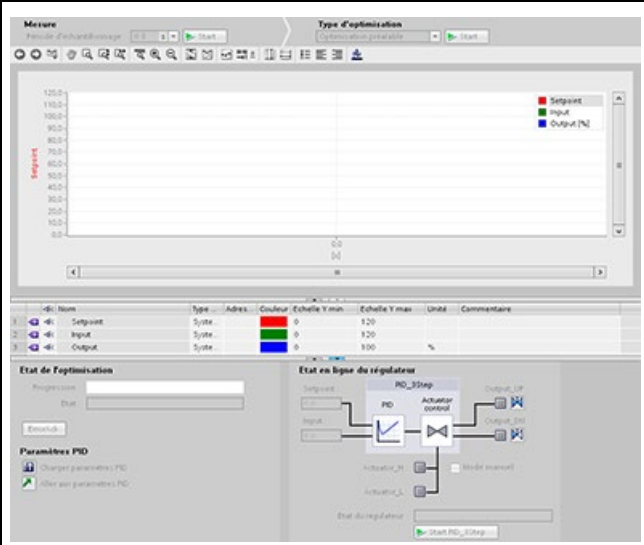
"Mode" / "State"	Nom	Description
4	Mode manuel	<p>Dans ce mode, le régulateur PID met à l'échelle, limite et transmet la valeur du paramètre "ManualValue" aux sorties.</p> <p>Le régulateur PID affecte "ManualValue" lors de la mise à l'échelle de l'algorithme PID (comme par exemple "PidOutputSum"), sa valeur décide donc si elle s'applique aux sorties de chauffage ou de refroidissement.</p> <p>Vous pouvez atteindre ce mode en paramétrant "Mode" = 4 ou ManualEnable"= TRUE.</p>
5	Valeur de réglage de remplacement avec surveillance d'erreur (mode Recover)	<p>Vous pouvez activer ce mode en réglant "Mode" sur 5. Ce mode est une réaction automatique du régulateur aux erreurs si le mode automatique est activé au moment où l'erreur se produit :</p> <ul style="list-style-type: none"> • SetSubstituteOutput = FALSE (dernière valeur de réglage valide) • SetSubstituteOutput = TRUE (valeur enregistrée dans le paramètre "SubstituteOutput") <p>Lorsque PID_Temp est en "Mode automatique" et que le paramètre "ActivateRecoverMode" = TRUE, PID_Temp passe dans ce mode si les erreurs suivantes surviennent :</p> <ul style="list-style-type: none"> • "Valeur invalide au paramètre "Input_PER". Vérifier la présence d'une erreur à l'entrée analogique (par exemple, un fil coupé)". (ErrorBits = DW#16#0002) • "Valeur invalide au paramètre "Input". La valeur n'est pas un nombre." (ErrorBits = DW#16#0200) • "Echec du calcul de la valeur de réglage. Vérifiez les paramètres PID." (ErrorBits = DW#16#0400) • "Valeur invalide au paramètre "Setpoint". La valeur n'est pas un nombre." (ErrorBits = DW#16#1000) <p>Dès que l'erreur n'est plus en attente, PID_Temp repasse automatiquement en mode automatique.</p>

10.2.7 Mise en service des régulateurs PID_Compact et PID_3Step

Utilisez l'éditeur de mise en service pour configurer le régulateur PID afin qu'une optimisation préalable soit effectuée à la mise en route et en fonctionnement. Pour ouvrir l'éditeur de mise en service, cliquez sur l'icône correspondante dans l'instruction ou dans le navigateur du projet.



Tableau 10- 33 Exemple d'écran de mise en service (PID_3Step)



- **Mesure** : Pour afficher la consigne, la mesure et la valeur de réglage dans un affichage de courbes en temps réel, entrez la période d'échantillonnage et cliquez sur le bouton "Démarrer".
- **Optimisation** : Pour optimiser la boucle PID, sélectionnez soit "Optimisation préalable" soit "Optimisation fine" (manuelle) et cliquez sur le bouton "Démarrer". Le régulateur PID exécute plusieurs phases pour calculer les temps de réponse et d'actualisation du système. Les paramètres optimisés appropriés sont calculés à partir de ces valeurs.

Une fois le processus d'optimisation achevé, vous pouvez stocker les nouveaux paramètres en cliquant sur le bouton "Charger les paramètres PID dans le projet" dans la partie "Paramètres PID" de l'éditeur de mise en service.

Si une erreur se produit pendant l'optimisation, la valeur de réglage de la boucle PID est mise à 0. Le régulateur PID passe alors en mode "inactif". L'état signale l'erreur.

Régulation de la valeur de démarrage PID

Vous pouvez éditer les valeurs réelles des paramètres de configuration PID afin que le comportement du régulateur PID puisse être optimisé en mode en ligne.

Ouvrez les "Objets technologiques" pour votre régulateur PID et son objet "Configuration". Pour accéder à la commande de valeur de début, cliquez sur l'icône lunettes" dans le coin supérieur gauche de la boîte de dialogue :



Vous pouvez maintenant modifier la valeur des paramètres de configuration de votre régulateur PID tel qu'indiqué dans la figure ci-dessous.

Vous pouvez comparer la valeur réelle à la valeur de démarrage (hors ligne) du projet et à la valeur de démarrage (en ligne) API de chaque paramètre. Ceci est nécessaire pour comparer les différences en ligne/hors ligne du bloc de données Objet technologique (TO-DB) et pour être informé des valeurs qui seront utilisées comme valeurs actuelles à la prochaine commutation Arrêt-Démarrage de l'API. De plus, une icône de comparaison donne une indication visuelle pour vous aider à identifier facilement les différences en ligne/hors ligne :



La figure ci-dessus montre l'écran de paramètre PID avec des icônes de comparaison montrant quelles valeurs sont différentes entre les projets en ligne et hors ligne. Une icône verte indique que les valeurs sont les mêmes ; une icône bleue/orange indique que les valeurs sont différentes.

En outre, cliquez sur le bouton paramètre avec la flèche vers le bas pour ouvrir une petite fenêtre qui montre la valeur de démarrage (hors ligne) du projet et sur la valeur de démarrage (en ligne) API de chaque paramètre :



10.2.8 Mise en service du régulateur PID_Temp

Utilisez l'éditeur de mise en service pour configurer le régulateur PID afin qu'une optimisation préalable soit effectuée à la mise en route et en fonctionnement. Pour ouvrir l'éditeur de mise en service, cliquez sur l'icône correspondante dans l'instruction ou dans le navigateur du projet.



Tableau 10- 34 Exemple d'écran de mise en service (PID_Temp)

The screenshot displays the 'Mise en service' (Service) editor for a PID controller. At the top, there are tabs for 'Mesure', 'Mode d'optimisation', 'Déchargement d'optimisation', and 'Optimisation'. Below the tabs is a graph showing the PID_Temp_1 (Environment de données) with various data series. A table below the graph lists variables: Setpoint, Input, OutputMax, and OutputMin, with their respective types and scales. At the bottom, there are sections for 'Etat de l'optimisation' and 'Etat en ligne du régulateur', along with a 'Paramètres PID' section containing fields for Kp, Ki, and Kd, and buttons for 'Charger les paramètres PID dans le projet' and 'Sauvegarder'.

Mesure : Pour afficher la consigne, la mesure et la valeur de réglage dans un affichage de courbes en temps réel, entrez la période d'échantillonnage et cliquez sur le bouton "Démarrer".

Optimisation : Pour optimiser la boucle PID_Temp, sélectionnez soit "Optimisation préalable" soit "Optimisation fine" (manuelle) et cliquez sur le bouton "Démarrer". Le régulateur PID exécute plusieurs phases pour calculer les temps de réponse et d'actualisation du système. Les paramètres optimisés appropriés sont calculés à partir de ces valeurs.

Une fois le processus d'optimisation achevé, vous pouvez enregistrer les nouveaux paramètres en cliquant sur le bouton "Charger les paramètres PID dans le projet" dans la partie "Paramètres PID" de l'éditeur de mise en service.

Si une erreur se produit pendant l'optimisation, la valeur de réglage de PID se met à "0". Le mode de PID est ensuite réglé sur le mode "inactif". L'état signale l'erreur.

Limites PWM

Les actionneurs commandés par la fonction logicielle PWM de PID_Temp peuvent nécessiter une protection contre les durées d'impulsion trop courtes (par exemple un relais à thyristor doit être activé pendant plus de 20 ms avant de pouvoir même réagir) ; vous affectez un temps d'activation minimum. L'actionneur peut également négliger les impulsions courtes et donc corrompre la qualité de la régulation. Un temps de désactivation minimum peut s'avérer nécessaire (par exemple pour éviter la surchauffe).

Pour afficher la vue des limites PWM, vous devez ouvrir la vue fonctionnelle dans la configuration des objets technologiques (TO) et sélectionner "Limites PWM" dans le nœud "Paramètres avancées" dans l'arborescence de navigation.

Si vous ouvrez la vue "Limites PWM" dans la vue fonctionnelle et que vous activez la surveillance (bouton "lunettes"), tous les contrôles indiquent la valeur de surveillance en ligne de TO-DB avec un arrière-plan de couleur orange et le contrôle multi-valeur, et vous pouvez éditer les valeurs (si les conditions de configuration sont remplies ; voir le tableau ci-dessous).

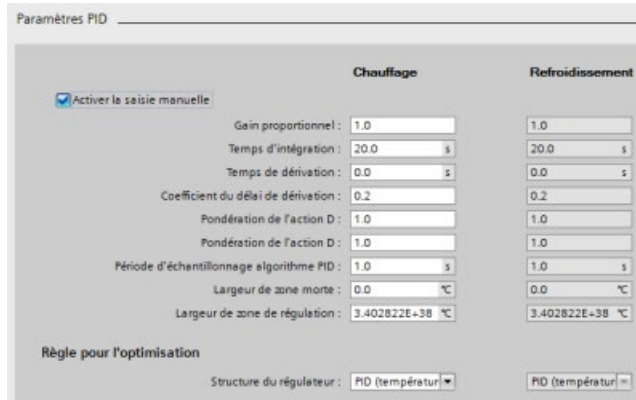
	Chauffage	Refroidissement
Plus petit temps ON :	0.0 s	0.0 s
Plus petit temps OFF :	0.0 s	0.0 s

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Temps d'activation minimum temps (chauffage) 1,2	"Config.Output.Heat.MinimumOnTime"	Nombre réel	100000.0 >= "Config.Output.Heat.MinimumOnTime >= 0.0	Une impulsion à OutputHeat_PWM n'est jamais plus courte que cette valeur.
Temps de désactivation minimum temps (chauffage) 1,2	"Config.Output.Heat.MinimumOffTime"	Nombre réel	100000.0 >= "Config.Output.Heat.MinimumOffTime >= 0.0	Une rupture à OutputHeat_PWM n'est jamais plus courte que cette valeur.
Temps d'activation minimum temps (refroidissement) 1,3,4	"Config.Output.Cool.MinimumOnTime"	Nombre réel	100000.0 >= Config.Output.Cool.MinimumOnTime >= 0.0	Une impulsion à OutputCool_PWM n'est jamais plus courte que cette valeur.
Temps de désactivation minimum temps (refroidissement) 1,3,4	"Config.Output.Cool.MinimumOffTime"	Nombre réel	100000.0 >= Config.Output.Cool.MinimumOffTime >= 0.0	Une rupture à OutputCool_PWM n'est jamais plus courte que cette valeur.

- ¹ Ce champ affiche "s" (secondes) comme unité de temps.
- ² Si la sélection de la Sortie (chauffage) dans la vue "Paramètres de base" n'est pas "OutputHeat_PWM" (Config.Output.Heat.Select = TRUE), vous devez régler cette valeur sur "0,0".
- ³ Si la sélection de la Sortie (refroidissement) dans la vue "Paramètres de base" n'est pas "OutputCool_PWM" (Config.Output.Cool.Select = TRUE), vous devez régler cette valeur sur "0,0".
- ⁴ Disponible seulement si vous cochez "Activer sortie (refroidissement)" dans "Paramètres de base (Config.ActivateCooling = TRUE).

Paramètres PID

La partie "Paramètres PID" de la vue "Paramètres avancés" est affichée en-dessous avec la fonction refroidissement et/ou "Commutation des paramètres PID" désactivées.



Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Activer l'entrée manuelle	"Retain.CtrlParams.SetByUser"	Booléen	Booléen	Vous devez cocher cette case pour saisir les paramètres PID manuellement.
Gain proportionnel (chauffage) ²	"Retain.CtrlParams.Heat.Gain"	Nombre réel	Gain >= 0,0	Gain proportionnel PID pour le chauffage
Temps d'intégration (chauffage) ^{1,2}	"Retain.CtrlParams.Heat.Ti"	Nombre réel	100000.0 >= Ti >= 0.0	Action intégrale PID pour le chauffage.
Temps de dérivation (chauffage) ^{1,2}	"Retain.CtrlParams.Heat.Td"	Nombre réel	100000.0 >= Td >= 0.0	Temps de dérivation PID pour le chauffage.
Coefficient de retard de l'action dérivée (chauffage) ²	"Retain.CtrlParams.Heat.TdFiltRatio"	Nombre réel	TdFiltRatio >= 0,0	Coefficient de retard de l'action dérivée PID pour le chauffage qui définit le temps de retard de l'action dérivée sous forme de coefficient du temps de dérivation PID.
Pondération de l'action proportionnelle (chauffage) ²	"Retain.CtrlParams.Heat.PWeighting"	Nombre réel	1,0 >= PWeighting >= 0,0	Pondération du gain proportionnel PID pour le chauffage soit dans le circuit de régulation directe soit dans la boucle de régulation.

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Pondération de l'action dérivée (chauffage) ²	"Retain.CtrlParams.Heat.DWeighting"	Nombre réel	1,0 >=DWeighting >= 0,0	Pondération de la partie de dérivation PID pour le chauffage soit dans le circuit de régulation directe soit dans la boucle de régulation.
Période d'échantillonnage de l'algorithme PID (chauffage) ^{1,2}	"Retain.CtrlParams.Heat.Cycle"	Nombre réel	100000,0 >=Cycle > 0,0	Cycle d'appel interne du régulateur PID pour le chauffage. Arrondi à un multiple entier du temps de cycle d'appel du FB.
Largeur de bande morte(chauffage) ^{2,3}	"Retain.CtrlParams.Heat.DeadZone"	Nombre réel	Zone morte >= 0,0	Largeur de la bande morte pour l'écart de régulation de chauffage.
Zone de régulation (chauffage) ^{2,3}	"Retain.CtrlParams.Heat.ControlZone"	Nombre réel	Zone de régulation > 0,0	Largeur de la zone d'écart de régulation pour le chauffage lorsque la régulation PID est active. Si l'écart de régulation quitte cette plage, la sortie est commutée sur des valeurs de réglage maximum. La valeur par défaut est "MaxReal" ainsi la zone de régulation est désactivée tant que l'auto-optimisation n'est pas exécutée. La valeur "0,0" est interdite pour la Zone de régulation ; avec la valeur "0,0", PID_Temp se comporte comme un régulateur tout ou rien qui chauffe ou refroidit toujours à pleine puissance.

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Structure du régulateur (chauffage)	"PIDSelfTune.SUT.TuneRuleHeat", "PIDSelfTune.TIR.TuneRuleHeat"	Ent	"PID-SelfTune.SUT.TuneRuleHeat" = 0..2, "PID-SelfTune.TIR.TuneRuleHeat" = 0..5	Vous pouvez sélectionner l'algorithme d'optimisation pour le chauffage. Sélections possibles : <ul style="list-style-type: none"> • PID (Température) (=par défaut) ("PIDSelfTune.SUT.TuneRuleHeat" = 2) ("PIDSelfTune.TIR.TuneRuleHeat" = 0) • PID ("PIDSelfTune.SUT.TuneRuleHeat" = 0) ("PIDSelfTune.TIR.TuneRuleHeat" = 0) • PI ("PIDSelfTune.SUT.TuneRuleHeat" = 1) ("PIDSelfTune.TIR.TuneRuleHeat" = 4) Toute autre combinaison affiche "Personnalisé" mais "Personnalisé" n'est pas réglé par défaut. "PID (Température)" est nouveau pour PID_Temp, avec une méthode de pré-optimisation (SUT) spécifique pour les processus de température.
Gain proportionnel (refroidissement) ⁴	"Retain.CtrlParams.Cool.Gain"	Nombre réel	Gain >= 0,0	Gain proportionnel PID pour le refroidissement
Temps d'intégration (refroidissement) ^{1,4}	"Retain.CtrlParams.Cool.Ti"	Nombre réel	100000,0 >=Ti >= 0,0	Action intégrale PID pour le refroidissement

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Temps de dérivation (refroidissement) ^{1,4}	"Retain.CtrlParams.Cool.Td"	Nombre réel	100000,0 >=Td >= 0,0	Temps de dérivation PID pour le refroidissement.
Coefficient de retard de l'action dérivée (refroidissement) ⁴	Retain.CtrlParams.Cool.TdFiltRatio"	Nombre réel	TdFiltRatio >= 0,0	Coefficient de retard de l'action dérivée PID pour le refroidissement qui définit le temps de retard de l'action dérivée sous forme de coefficient du temps de dérivation PID.
Pondération de l'action proportionnelle (refroidissement) ⁴	"Retain.CtrlParams.Cool.PWeighting"	Nombre réel	1,0 >=PWeighting >= 0,0	Pondération du gain proportionnel PID pour le refroidissement soit dans le circuit de régulation directe soit dans la boucle de régulation.
Pondération de l'action dérivée (refroidissement) ⁴	Retain.CtrlParams.Cool.DWeighting"	Nombre réel	1,0 >=DWeighting >= 0,0	Pondération de la partie de dérivation PID pour le refroidissement soit dans le circuit de régulation directe soit dans la boucle de régulation.
Période d'échantillonnage de l'algorithme PID (refroidissement) ^{1,4}	"Retain.CtrlParams.Cool.Cycle"	Nombre réel	100000,0 >=Cycle > 0,0	Cycle d'appel interne du régulateur PID pour le refroidissement. Arrondi à un multiple entier du temps de cycle d'appel du FB.
Largeur de bande morte (refroidissement) ^{3,4}	"Retain.CtrlParams.Cool.DeadZone"	Nombre réel	Zone morte >= 0,0	Largeur de la bande morte pour l'écart de régulation de refroidissement.

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Zone de régulation (refroidissement) ^{3,4}	"Retain.CtrlParams.Cool.ControlZone"	Nombre réel	Zone de régulation > 0,0	<p>Largeur de la zone d'écart de régulation pour le refroidissement lorsque la régulation PID est active. Si l'écart de régulation quitte cette plage, la sortie est commutée sur des valeurs de réglage maximum.</p> <p>La valeur par défaut est "MaxReal" ainsi la zone de régulation est désactivée tant que l'auto-optimisation n'est pas exécutée.</p> <p>La valeur "0,0" est interdite pour la Zone de régulation ; avec la valeur "0,0", PID_Temp se comporte comme un régulateur tout ou rien qui chauffe ou refroidit toujours à pleine puissance.</p>

Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
Structure du régulateur (refroidissement)	"PIDSelfTune.SUT.TuneRuleCool", "PIDSelfTune.TIR.TuneRuleCool"	Ent	"PID-SelfTune.SUT.TuneRuleHeat" = 0..2, "PID-SelfTune.TIR.TuneRuleHeat" = 0..5	<p>Vous pouvez sélectionner l'algorithme d'optimisation pour le refroidissement.</p> <p>Sélections possibles :</p> <ul style="list-style-type: none"> • PID (Température) (=par défaut) ("PIDSelfTune.SUT.TuneRuleCool" = 2) ("PIDSelfTune.TIR.TuneRuleCool" = 0) • PID ("PIDSelfTune.SUT.TuneRuleCool" = 0) ("PIDSelfTune.TIR.TuneRuleCool" = 0) • PI ("PIDSelfTune.SUT.TuneRuleCool" = 1) ("PIDSelfTune.TIR.TuneRuleCool" = 4) <p>Toute autre combinaison affiche "Personnalisé" mais "Personnalisé" n'est pas réglé par défaut.</p> <p>"PID (Température)" est nouveau pour PID_Temp, avec une méthode de pré-optimisation (SUT) spécifique pour les processus de température.</p> <p>Disponible seulement si vous co-chez/sélectionnez les éléments suivants : "Activer sortie (refroidissement)" dans la vue "Paramètres de base" ("Config.ActivateCooling" = TRUE), et "Commutation des paramètres PID" dans la vue "Paramètres de sortie" ("Config.AdvancedCooling" = TRUE).</p>

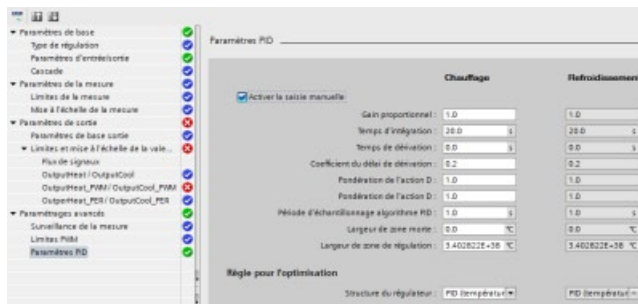
Paramètre	Paramètre TO-DB	Type de données	Plage de valeurs	Description
-----------	-----------------	-----------------	------------------	-------------

- 1 Ce champ affiche "s" (secondes) comme unité de temps.
- 2 Disponible seulement si vous cochez "Activer l'entrée manuelle" dans les paramètres PID ("Retain.CtrlParams.SetByUser" = TRUE).
- 3 L'unité de mesure est affichée à la fin du champ comme sélectionné dans la vue "Paramètres de base".
- 4 Disponible seulement si vous cochez/sélectionnez les éléments suivants : "Activer l'entrée manuelle" dans les paramètres PID ("Retain.CtrlParams.SetByUser" = TRUE), "Activer sortie (refroidissement) dans la vue "Paramètres de base" ("Config.ActivateCooling" = TRUE), et "Commutation des paramètres PID" dans la vue "Paramètres de sortie" (Config.AdvancedCooling = TRUE).

Régulation de la valeur de démarrage PID

Vous pouvez éditer les valeurs réelles des paramètres de configuration PID afin que le comportement du régulateur PID puisse être optimisé en mode en ligne.

Ouvrez les "Objets technologiques" pour votre régulateur PID et son objet "Configuration". Pour accéder à la commande de valeur de début, cliquez sur l'icône lunettes dans le coin supérieur gauche de la boîte de dialogue :



Vous pouvez maintenant modifier la valeur des paramètres de configuration de votre régulateur PID tel qu'indiqué dans la figure ci-dessous.

Vous pouvez comparer la valeur réelle à la valeur de démarrage (hors ligne) du projet et à la valeur de démarrage (en ligne) API de chaque paramètre. Ceci est nécessaire pour comparer les différences en ligne/hors ligne du bloc de données Objet technologique (TO-DB) et pour être informé des valeurs qui seront utilisées comme valeurs actuelles à la prochaine commutation Arrêt-Démarrage de l'API. De plus, une icône de comparaison donne une indication visuelle pour vous aider à identifier facilement les différences en ligne/hors ligne :

The screenshot shows the 'Paramètres PID' interface with two columns: 'Chauffage' and 'Refroidissement'. A checkbox 'Activer la saisie manuelle' is checked. The parameters and their values are as follows:

Paramètre	Chauffage	Refroidissement
Gain proportionnel	1.0	1.0
Temps d'intégration	20.0 s	20.0 s
Temps de dérivation	0.0 s	0.0 s
Coefficient du délai de dérivation	0.2	0.2
Pondération de l'action D	1.0	1.0
Pondération de l'action D	1.0	1.0
Période d'échantillonnage algorithme PID	1.0 s	1.0 s
Largeur de zone morte	0.0 °C	0.0 °C
Largeur de zone de régulation	3.402822E+38 °C	3.402822E+38 °C

At the bottom, the 'Règle pour l'optimisation' section shows 'Structure du régulateur' set to 'PID (température)' for both modes.

La figure ci-dessus montre l'écran de paramètre PID avec des icônes de comparaison montrant quelles valeurs sont différentes entre les projets en ligne et hors ligne. Une icône verte indique que les valeurs sont les mêmes ; une icône bleue/orange indique que les valeurs sont différentes.

En outre, cliquez sur le bouton paramètre avec la flèche vers le bas pour ouvrir une petite fenêtre qui montre la valeur de démarrage (hors ligne) du projet et sur la valeur de démarrage (en ligne) API de chaque paramètre :

This screenshot is identical to the one above, showing the 'Paramètres PID' interface with the same parameters and values. The comparison icons are not clearly visible in this version of the image.

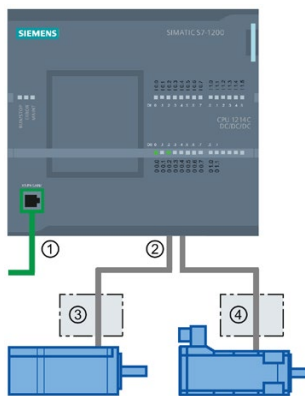
10.3 Motion control

La CPU offre une fonctionnalité Motion control (commande de mouvement) permettant l'utilisation de moteurs pas à pas et de servomoteurs à interface d'impulsion. La fonctionnalité Motion control se charge de la commande et de la surveillance des entraînements.

- L'objet technologique "Axe" configure les données mécaniques de l'entraînement, l'interface de l'entraînement, les paramètres dynamiques et d'autres propriétés de l'entraînement.
- Vous configurez les sorties d'impulsions et de sens de la CPU pour commander l'entraînement.
- Votre programme utilisateur utilise les instructions Motion Control pour commander l'axe et déclencher des tâches de déplacement.
- Utilisez l'interface PROFINET pour établir la liaison en ligne entre la CPU et la console de programmation. En plus des fonctions en ligne de la CPU, des fonctions supplémentaires de mise en service et de diagnostic sont disponibles pour Motion Control.

Remarque

Les modifications que vous apportez à la configuration de commande de mouvement et que vous chargez à l'état MARCHE ne prennent effet qu'après passage de la CPU de l'état ARRET à l'état MARCHE.



- ① PROFINET
- ② Sorties d'impulsions et de sens
- ③ Partie puissance du moteur pas à pas
- ④ Partie puissance du servomoteur

Les variantes DC/DC/DC de la CPU S7-1200 ont des sorties intégrées pour la commande directe d'entraînements. Les variantes à relais de la CPU ont besoin du Signal Board à sorties DC pour commander des entraînements.

Un Signal Board (SB) augmente les E/S intégrées en ajoutant quelques E/S supplémentaires. Un SB à deux sorties TOR peut être utilisé comme sorties d'impulsions et de sens pour commander un moteur. Un SB à quatre sorties TOR peut être utilisé comme sorties d'impulsions et de sens pour commander deux moteurs. Les sorties relais intégrées ne peuvent pas être utilisées comme sorties d'impulsions pour commander des moteurs. Que vous utilisiez les E/S intégrées, les E/S d'un SB ou une combinaison des deux, vous avez au maximum quatre générateurs d'impulsions.

Les entrées et sorties sont affectées par défaut aux quatre générateurs d'impulsions ; toutefois, ils peuvent être affectés à toute sortie TOR sur la CPU ou le SB. Les générateurs d'impulsions sur la CPU ne peuvent pas être affectés à des SM ou des périphéries décentralisées.

Remarque

Les sorties de trains d'impulsions ne peuvent pas être utilisées par d'autres instructions dans le programme utilisateur

En cas de configuration des sorties de la CPU ou du Signal Board comme générateurs d'impulsions (pour les instructions PWM ou Motion Control), les adresses des sorties correspondantes ne pilotent plus les sorties. Si votre programme utilisateur écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.

Tableau 10- 35 Nombre maximum d'entraînements pouvant être commandés

Type de CPU		E/S intégrées ; pas de SB installé		Avec un SB (2 sorties DC)		Avec un SB (4 sorties DC)	
		Avec sens	Sans sens	Avec sens	Sans sens	Avec sens	Sans sens
CPU 1211C	DC/DC/DC	2	4	3	4	4	4
	AC/DC/Relais	0	0	1	2	2	4
	DC/DC/Relais	0	0	1	2	2	4
CPU 1212C	DC/DC/DC	3	4	3	4	4	4
	AC/DC/Relais	0	0	1	2	2	4
	DC/DC/Relais	0	0	1	2	2	4
CPU 1214C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/Relais	0	0	1	2	2	4
	DC/DC/Relais	0	0	1	2	2	4
CPU 1215C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/Relais	0	0	1	2	2	4
	DC/DC/Relais	0	0	1	2	2	4
CPU 1217C	DC/DC/DC	4	4	4	4	4	4

Remarque

Le nombre maximum de générateurs d'impulsions est de quatre.

Que vous utilisiez les E/S intégrées, les E/S d'un SB ou une combinaison des deux, vous avez au maximum quatre générateurs d'impulsions.

Tableau 10- 36 Sortie de la CPU : fréquence maximale

CPU	Sortie voie de la CPU	Sortie d'impulsions et de sens	A/B, quadrature, incrément/décément et impulsion/sens
1211C	Qa.0 à Qa.3	100 kHz	100 kHz
1212C	Qa.0 à Qa.3	100 kHz	100 kHz
	Qa.4, Qa.5	20 kHz	20 kHz
1214C et 1215C	Qa.0 à Qa.3	100 kHz	100 kHz
	Qa.4 à Qb.1	20 kHz	20 kHz
1217C	DQa.0 à DQa.3 (.0+, .0- à .3+, .3-)	1 MHz	1 MHz
	DQa.4 à DQb.1	100 kHz	100 kHz

Tableau 10- 37 Sortie de Signal Board (SB) : fréquence maximale (SB optionnel)

Signal Board (SB)	Sortie voie de SB	Sortie d'impulsions et de sens	A/B, quadrature, incrément/décément et impulsion/sens
SB 1222, 200 kHz	DQe.0 à DQe.3	200 kHz	200 kHz
SB 1223, 200 kHz	DQe.0, DQe.1	200 kHz	200 kHz
SB 1223	DQe.0, DQe.1	20 kHz	20 kHz

Tableau 10- 38 Fréquences limites des sorties d'impulsions

Sortie d'impulsions	Fréquence
Intégrée	4 PTO : $2 \text{ Hz} \leq f \leq 1 \text{ MHz}$, 4 PTO : $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$, ou la combinaison de ces valeurs pour 4 PTO. ^{1,2}
SB standard	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
SB rapides	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$

¹ Voir le tableau ci-dessous pour quatre combinaisons de fréquence de sorties d'impulsions possibles pour la CPU 1217C.

² Voir le tableau ci-dessous pour quatre combinaisons de fréquence de sorties d'impulsions possibles pour les CPU 1211C, CPU 1212C, CPU 1214C ou CPU 1215C.

Exemple : Configurations de fréquence de sorties d'impulsions pour la CPU 1217C

Remarque

La CPU 1217C peut générer des sorties d'impulsions à des fréquences pouvant atteindre 1 MHz à l'aide des sorties différentielles intégrées.

Les exemples ci-dessous montrent quatre combinaisons de fréquence de sorties d'impulsions possibles :

- Exemple 1 : 4 PTO - 1 MHz, pas de sortie de sens
- Exemple 2 : 1 PTO - 1 MHz, 2 PTO - 100 kHz et 1 PTO - 20 kHz, tous avec sortie de sens
- Exemple 3 : 4 PTO - 200 kHz, pas de sortie de sens
- Exemple 4 : 2 PTO - 100 kHz et 2 PTO - 200 kHz, tous avec sorties de sens

P = Impulsion D = Sens		Sorties intégrées de la CPU										Sorties SB rapides				Sorties SB standard	
		Sorties 1 MHz (Q)				Sorties 100 kHz (Q)						Sorties 200 kHz (Q)				Sorties 20 kHz (Q)	
		0.0+	0.1+	0.2+	0.3+	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		0.0-	0.1-	0.2-	0.3-												
Ex. 1 : 4 - 1 MHz (pas de sortie de sens)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
Ex. 2 : 1 PTO - 1 MHz, 2 PTO - 100 et 1 PTO - 20 kHz (tous avec sortie de sens)	PTO1	P	D														
	PTO2					P	D										
	PTO3							P	D								
	PTO4															P	D
Ex. 3 : 4 - 200 kHz (pas de sortie de sens)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
Ex. 4 : 2 - 100 kHz ;	PTO1					P	D										
	PTO2							P	D								
	PTO3											P	D				

10.3 Motion control

P = Impulsion D = Sens		Sorties intégrées de la CPU										Sorties SB rapides				Sorties SB standard	
2 - 200 kHz (tous avec sortie de sens)	PTO4													P	D		

Exemple : Configurations de fréquence de sorties d'impulsions pour les CPU 1211C, CPU 1212C, CPU 1214C et CPU 1215C

Les exemples ci-dessous montrent quatre combinaisons de fréquence de sorties d'impulsions possibles :

- Exemple 1 : 4 PTO - 100 kHz, pas de sortie de sens
- Exemple 2 : 2 PTO - 100 kHz et 2 PTO - 20 kHz, tous avec sorties de sens
- Exemple 3 : 4 PTO - 200 kHz, pas de sortie de sens
- Exemple 4 : 2 PTO - 100 kHz et 2 PTO - 200 kHz, tous avec sorties de sens

P = Impulsion D = Sens		Sorties intégrées de la CPU										Sorties SB rapides				Sorties SB lentes		
		Sorties 100 kHz (Q)				Sorties 20 kHz (Q)						Sorties 200 kHz (Q)				Sorties 20 kHz (Q)		
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1	
		CPU 1211C																
		CPU 1212C				CPU 1212C												
		CPU 1214C				CPU 1214C		CPU 1214C										
		CPU 1215C				CPU 1215C		CPU 1215C										
Ex. 1 : 4 PTO - 100 kHz (pas de sortie de sens)	PTO1	P																
	PTO2		P															
	PTO3			P														
	PTO4				P													
Ex. 2 : 2 PTO - 100 kHz ; 2 PTO - 20 kHz (tous avec sortie de sens)	PTO1	P	D															
	PTO2			P	D													
	PTO3					P	D											
	PTO4							P	D									

P = Impulsion D = Sens		Sorties intégrées de la CPU										Sorties SB rapides			Sorties SB lentes			
Ex. 3 : 4 - 200 kHz (pas de sortie de sens)	PTO1												P					
	PTO2													P				
	PTO3														P			
	PTO4															P		
Ex. 4 : 2 - 100 kHz ; 2 - 200 kHz (tous avec sor- tie de sens)	PTO1	P	D															
	PTO2			P	D													
	PTO3											P	D					
	PTO4													P	D			

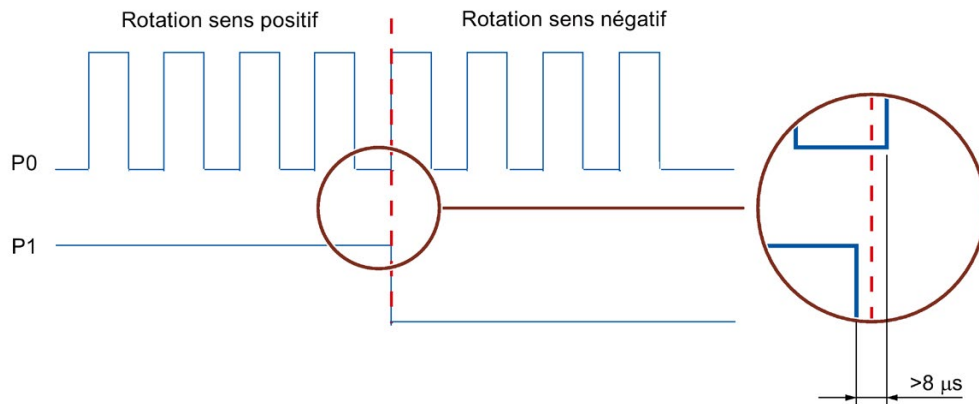
Remarque

Pour plus d'informations, voir la description fonctionnelle *SIMATIC STEP 7 S7-1200 Motion Control V14*.

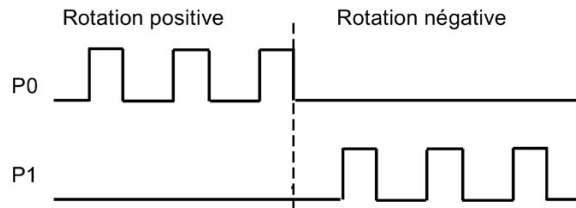
10.3.1 Phasage

Vous avez quatre options pour l'interface de "phasage" pour l'entraînement par moteur pas-à-pas/servomécanisme. Ces options sont les suivantes :

- PTO (impulsion A et sens B) : Si vous sélectionnez une option de PTO (impulsion A et sens B), alors une sortie (P0) commande la génération d'impulsions et une sortie (P1) commande le sens. P1 est élevé (actif) si la génération d'impulsions est dans le sens positif. P1 est faible (inactif) si la génération d'impulsions est dans le sens négatif :



- PTO (incrémentation impulsion A et décrémentation impulsion B) : Si vous sélectionnez une option PTO (incrémentation impulsion A et décrémentation impulsion B), alors une sortie (P0) envoie des impulsions pour des sens positifs et une sortie différente (P1) envoie des impulsions pour des sens négatifs :



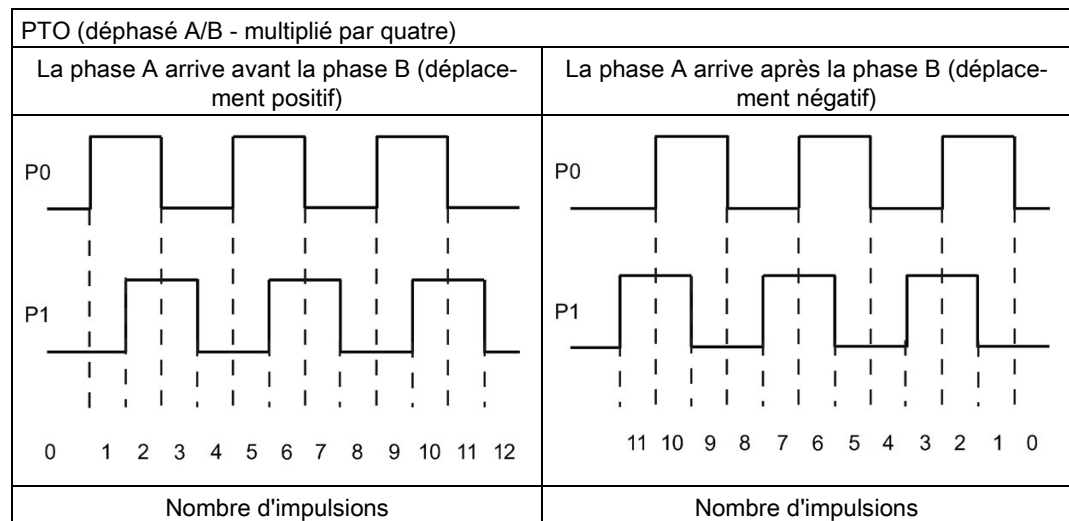
- PTO (déphasé A/B) : Si vous sélectionnez une option PTO (déphasé A/B), alors les deux sorties envoient des impulsions à la vitesse spécifiée, mais avec un déphasage de 90 degrés. C'est une configuration 1X, ce qui signifie qu'une impulsion est la durée entre des commutations positives de P0. Dans ce cas, le sens est déterminé par la sortie qui effectue une commutation élevée la première. P0 arrive avant P1 pour le sens positif. P1 arrive avant P0 pour le sens négatif.

Le nombre d'impulsions générées est fonction du nombre de commutations de 0 à 1 de la phase A. Le rapport de phase détermine le sens de déplacement :

PTO (déphasé A/B)	
La phase A arrive avant la phase B (déplacement positif)	La phase A arrive après la phase B (déplacement négatif)
<p>The diagram shows two square wave pulses, P0 and P1. P0 pulses occur first, followed by P1 pulses. Vertical dashed lines indicate the timing of each P0 pulse. Below the x-axis, the numbers 0, 1, 2, and 3 are marked, corresponding to the four P0 pulses.</p>	<p>The diagram shows two square wave pulses, P0 and P1. P1 pulses occur first, followed by P0 pulses. Vertical dashed lines indicate the timing of each P1 pulse. Below the x-axis, the numbers 2, 1, and 0 are marked, corresponding to the three P1 pulses.</p>
Nombre d'impulsions	Nombre d'impulsions

- PTO (déphasé A/B - multiplié par quatre) : Si vous sélectionnez une option PTO (déphasé A/B - multiplié par quatre), alors les deux sorties envoient des impulsions à la vitesse spécifiée, mais avec un déphasage de 90 degrés. La configuration multipliée par quatre est une configuration 4X, ce qui signifie qu'une impulsion est la transition de chaque sortie (positive et négative). Dans ce cas, le sens est déterminé par la sortie qui effectue une commutation élevée la première. P0 arrive avant P1 pour le sens positif. P1 arrive avant P0 pour le sens négatif.

La multiplication par quatre est basée sur des transitions positives et négatives de la phase A et de la phase B. Vous configurez le nombre de transitions. Le rapport de phase (A arrive avant B ou B arrive avant A) détermine le sens de déplacement.



- PTO (impulsion et sens (sens désélectionné)) : Si vous désélectionnez la sortie de sens dans un PTO (impulsion et sens (sens désélectionné)), alors la sortie (P0) commande la génération d'impulsions. La sortie P1 n'est pas utilisée et est disponible pour des utilisations d'autres programmes. Seules des instructions de déplacement positif sont acceptées par la CPU dans ce mode. Motion control vous empêche de faire des configurations négatives illicites quand vous sélectionnez ce mode. Vous pouvez enregistrer une sortie si votre application de déplacement est dans un seul sens. La phase unique (une sortie) est représentée dans la figure ci-dessous (en supposant une polarité positive) :



10.3.2 Configuration d'un générateur d'impulsions

1. Ajoutez un objet technologique :

- Dans l'arborescence du projet, affichez le détail du nœud "Objets technologiques" et sélectionnez "Ajouter nouvel objet".
- Sélectionnez l'icône "Axe" (renommez-la si nécessaire) et cliquez sur "OK" pour ouvrir l'éditeur de configuration pour l'objet axe.

Remarque

Pour garantir la cohérence de votre projet après que vous avez renommé les objets technologiques, chargez le projet dans la CPU lorsque celle-ci est à l'ARRÊT. Il y a changement de nom lorsque vous supprimez un objet technologique et que vous créez un nouvel objet technologique avec un nouveau nom et numéro de bloc de données.

- Affichez les propriétés "Sélection de PTO pour la commande d'axe" sous les "Paramètres de base" et sélectionnez l'impulsion désirée.

Remarque

Si PTO n'a pas été configuré auparavant dans les propriétés CPU, il est configuré pour utiliser l'une des sorties intégrées.

Si vous utilisez une sortie de Signal Board, sélectionnez le bouton "Configuration d'appareil" pour afficher les propriétés CPU. Dans les "Options d'impulsions" sous "Affectation de paramètres", configurez la source de la sortie sur une sortie de Signal Board.

- Configurez les autres paramètres de base et avancés.

2. Programmez votre application : Insérez l'instruction MC_Power dans un bloc de code.

- Pour l'entrée Axis, sélectionnez l'objet technologique axe que vous avez créé et configuré.
- Lorsque l'entrée Enable à la valeur VRAI, les autres instructions de mouvement sont autorisées à fonctionner.
- Lorsque l'entrée Enable a la valeur FAUX, les autres instructions de mouvement sont annulées.

Remarque

N'utilisez qu'une instruction MC_Power par axe.

3. Insérez les autres instructions de mouvement pour produire le mouvement requis.

Remarque

Configuration d'un générateur d'impulsions sur des sorties de Signal Board : Sélectionnez les propriétés "Générateurs d'impulsions (PTO/PWM)" pour une CPU (dans la configuration d'appareil) et activez un générateur d'impulsions. Deux générateurs d'impulsions sont disponibles pour chaque CPU S7-1200 de versions V1.0, V2.0, V2.1 et V2.2. Les CPU S7-1200 de versions V3.0 et V4.0 disposent de quatre générateurs d'impulsions chacune. Sous "Options d'impulsions" dans cette même zone de configuration, sélectionnez "Générateur d'impulsions sous forme de : PTO".

Remarque

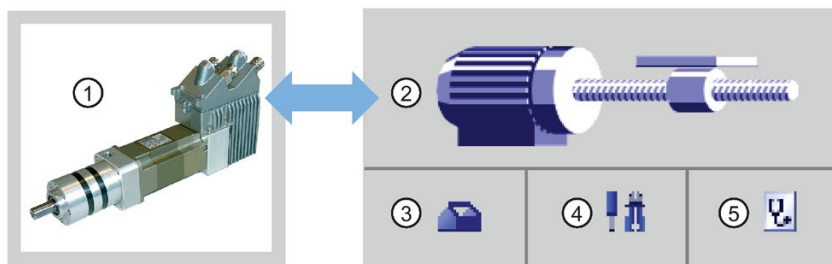
La CPU calcule les tâches de mouvement en "tranches" ou segments de 10 ms. Lorsqu'une tranche est en cours d'exécution, la prochaine attend son exécution dans la file d'attente. Si vous interrompez la tâche de mouvement sur un axe (en exécutant une autre nouvelle tâche de mouvement pour cet axe), la nouvelle tâche de mouvement ne peut pas être exécutée durant un maximum de 20 ms (le reste de la tranche actuelle plus la tranche en file d'attente).

10.3.3 Ouvrir la boucle de commande de mouvement

10.3.3.1 Configuration de l'axe

Vous connectez l'axe de la boucle ouverte sur l'API et l'entraînement par le biais d'une PTO (sortie de train d'impulsions). Pour des applications de déplacement utilisant PTO, la CPU exige des E/S TOR intégrées ou de signal board SB. Cela limite le nombre d'axes disponibles sur les plus petits API.

STEP 7 fournit les outils de configuration, les outils de mise en service et les outils de diagnostic pour l'objet technologique "Axe".



- ① Entraînement
- ② Objet technologique
- ③ Configuration

- ④ Mise en service
- ⑤ Diagnostic

Remarque

Pour les versions de firmware CPU V2.2 ou antérieures, PTO exige la fonctionnalité interne d'un compteur rapide (HSC). Cela signifie que le compteur rapide HSC correspondant ne peut pas être utilisé ailleurs.

L'affectation entre PTO et HSC est fixe. Lorsque PTO1 est activé, il est associé à HSC1. Lorsque PTO2 est activé, il est associé à HSC2. Vous ne pouvez pas visualiser la valeur en cours (par exemple, dans ID1000) lorsque des impulsions surviennent.

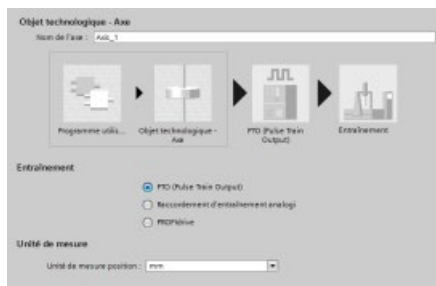
Les CPU S7-1200 V3.0 ou plus ne présentent pas cette restriction ; tous les HSC restent disponibles pour le programme lorsque des sorties d'impulsions sont configurées dans les CPU concernées.

Tableau 10- 39 Outils STEP 7 pour Motion Control

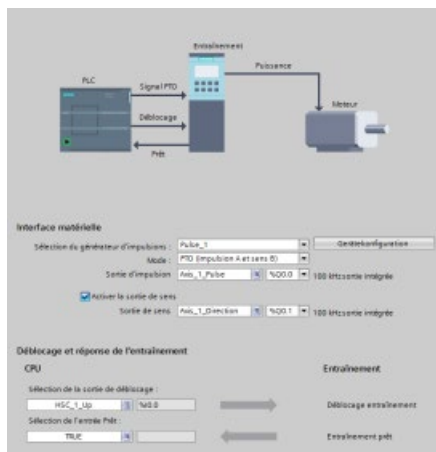
Outil	Description
Configuration	<p>Configure les propriétés suivantes de l'objet technologique "Axe" :</p> <ul style="list-style-type: none"> • Sélection de la sortie PTO à utiliser et configuration de l'interface d'entraînement • Propriétés de la mécanique et du rapport de transmission de l'entraînement (ou machine ou système) • Propriétés pour les limites de positionnement, la dynamique et le référencement <p>Sauvegardez la configuration dans le bloc de données de l'objet technologique.</p>
Mise en service	<p>Teste la fonction de votre axe sans avoir à créer un programme utilisateur. Lorsque l'outil est lancé, le panneau de commande s'affiche. Les commandes suivantes sont disponibles sur le panneau de commande :</p> <ul style="list-style-type: none"> • Libérer et bloquer l'axe • Déplacer l'axe en mode Marche par à-coups • Positionnement absolu et relatif de l'axe • Référencer l'axe • Acquitter les erreurs <p>La vitesse et l'accélération/la décélération peuvent être spécifiées pour les commandes de mouvement. Le panneau de commande affiche également l'état en cours de l'axe.</p>
Diagnostic	Surveille l'état en cours et les informations d'erreur concernant l'axe et l'entraînement.



Le sélecteur d'arborescence pour l'axe PTO ne comprend pas les menus de configuration Codeur, Modulo, Surveillance de position et Boucle de commande.



Une fois que vous avez créé l'objet technologique pour l'axe, vous configurez l'axe en définissant les paramètres de base, tels que la sortie PTO et la configuration de l'interface d'entraînement. Vous configurez également d'autres propriétés de l'axe, telles que les limites de positionnement, la dynamique et le référencement.

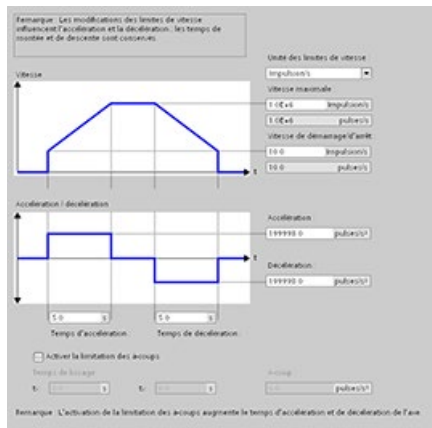


Remarque

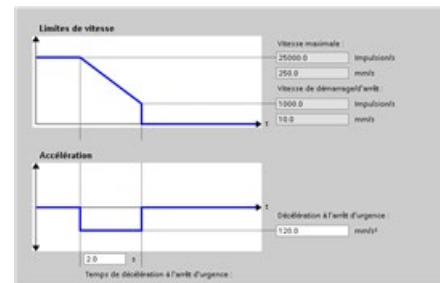
Vous devrez peut-être adapter les valeurs des paramètres d'entrée des instructions Motion Control à la nouvelle unité de mesure dans le programme utilisateur.



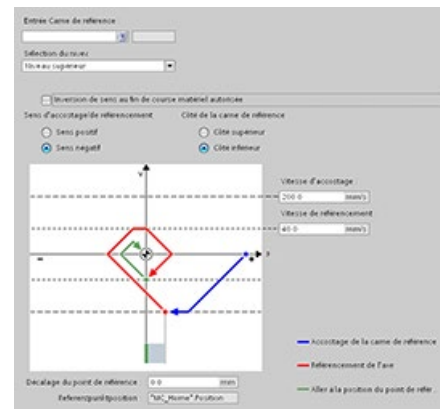
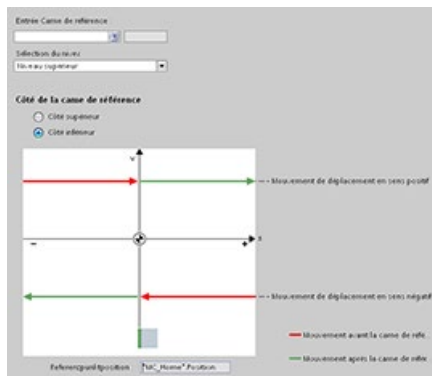
Configurez les propriétés pour les signaux de l'entraînement, la mécanique de l'entraînement et la surveillance de position (fins de course matériels et logiciels).



Vous configurez la dynamique du mouvement et le comportement de la commande d'arrêt d'urgence.



Vous configurez en outre le comportement de référencement (passif et actif).



Utilisez le panneau de commande "Mise en service" pour tester la fonctionnalité indépendamment de votre programme utilisateur.



Cliquez sur l'icône "Démarrer" pour mettre l'axe en service.

Le panneau de commande montre l'état en cours de l'axe. Vous pouvez non seulement libérer et bloquer l'axe, mais également tester le positionnement de l'axe (en termes absolus et relatifs) et vous pouvez indiquer la vitesse, l'accélération et la décélération. Vous pouvez également tester les tâches de référencement et le mode Marche par à-coups. Le panneau de commande vous permet en outre d'acquiescer les erreurs.

10.3.3.2 Mise en service

Fonction de diagnostic "Bits d'état et d'erreur"

La fonction de diagnostic "Bits d'état et d'erreur" vous permet de surveiller les principaux messages d'état et d'erreur de l'axe. L'affichage de la fonction de diagnostic est disponible en mode en ligne dans les modes de fonctionnement "Commande manuelle" et "Commande automatique" lorsque l'axe est actif.

Tableau 10- 40 Etat de l'axe

Etat	Description
Libéré	L'axe est débloqué et prêt à être piloté par des tâches Motion Control. (variable de l'objet technologique : <nom de l'axe>.StatusBits.Enable)
Référencé	L'axe est référencé et capable d'exécuter des tâches de positionnement absolu de l'instruction Motion Control "MC_MoveAbsolute". Il n'est pas nécessaire que l'axe soit référencé pour le positionnement relatif. Situations spéciales : <ul style="list-style-type: none"> • Pendant le référencement actif, l'état est FAUX. • Si un axe référencé est soumis à un référencement passif, l'état est mis à VRAI pendant le référencement passif. (variable de l'objet technologique : <nom de l'axe>.StatusBits.HomingDone)
Erreur	Une erreur s'est produite dans l'objet technologique Axe. En mode automatique, les paramètres ErrorID et ErrorInfo des instructions Motion Control fournissent plus d'informations sur l'erreur. En mode manuel, le champ "Dernière erreur" du panneau de commande affiche des informations détaillées sur la cause de l'erreur. (variable de l'objet technologique : <nom de l'axe>.StatusBits.Error)
Panneau de commande actif	Le mode "Commande manuelle" a été activé dans le panneau de commande. Le panneau de commande est en charge de la commande de l'objet technologique Axe. L'axe ne peut pas être piloté à partir du programme utilisateur. (variable de l'objet technologique : <nom de l'axe>.StatusBits.ControlPanelActive)

Tableau 10- 41 Etat de l'entraînement

Etat	Description
Entraînement prêt	L'entraînement est prêt à fonctionner. (variable de l'objet technologique : <nom de l'axe>.StatusBits.DriveReady)
Erreur	L'entraînement a signalé une erreur après défaillance de son signal Prêt. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.DriveFault)

Tableau 10- 42 Etat du déplacement de l'axe

Etat	Description
Immobilisation	L'axe est immobilisé. (variable de l'objet technologique : <nom de l'axe>.StatusBits.StandStill)
Accélération	L'axe accélère. (variable de l'objet technologique : <nom de l'axe>.StatusBits.Acceleration)
Vitesse constante	L'axe se déplace à vitesse constante. (variable de l'objet technologique : <nom de l'axe>.StatusBits.ConstantVelocity)
Décélération	L'axe décélère (freine). (variable de l'objet technologique : <nom de l'axe>.StatusBits.Deceleration)

Tableau 10- 43 Etat du mode de déplacement

Etat	Description
Positionnement	Cet axe exécute une tâche positionnement de l'instruction Motion Control "MC_MoveAbsolute" ou "MC_MoveRelative" ou du panneau de commande. (variable de l'objet technologique : <nom de l'axe>.StatusBits.PositioningCommand)
Prescription de vitesse	Cet axe exécute une tâche à vitesse prédéfinie de l'instruction Motion Control "MC_MoveVelocity" ou "MC_MoveJog" ou du panneau de commande. (variable de l'objet technologique : <nom de l'axe>.StatusBits.SpeedCommand)
Référencement	Cet axe exécute une tâche de référencement de l'instruction Motion Control "MC_Home" ou du panneau de commande. (variable de l'objet technologique : <nom de l'axe>.StatusBits.Homing)

Tableau 10- 44 Bits d'erreur

Erreur	Description
Fin de course logiciel min. atteint	Le fin de course logiciel inférieur a été atteint. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.SwLimitMinReached)
Fin de course logiciel min. dépassé	Le fin de course logiciel inférieur a été dépassé. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.SwLimitMinExceeded)
Fin de course logiciel max. atteint	Le fin de course logiciel supérieur a été atteint. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.SwLimitMaxReached)
Fin de course logiciel max. dépassé	Le fin de course logiciel supérieur a été dépassé. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.SwLimitMaxExceeded)
Fin de course matériel négatif	Le fin de course matériel inférieur a été accosté. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.HwLimitMin)
Fin de course matériel positif	Le fin de course matériel supérieur a été accosté. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.HwLimitMax)
PTO déjà utilisé	Un deuxième axe utilise le même PTO et est activé avec "MC_Power". (variable de l'objet technologique : <nom de l'axe>.ErrorBits.HwUsed)
Erreur de configuration	L'objet technologique Axe a été mal configuré ou des données de configuration éditables ont été modifiées de manière incorrecte pendant l'exécution du programme utilisateur. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.ConfigFault)
Erreur générale	Une erreur interne s'est produite. (variable de l'objet technologique : <nom de l'axe>.ErrorBits.SystemFault)

Fonction de diagnostic "Etat du déplacement"

La fonction de diagnostic "Etat du déplacement" vous permet de surveiller l'état de déplacement de l'axe. L'affichage de la fonction de diagnostic est disponible en mode en ligne dans les modes de fonctionnement "Commande manuelle" et "Commande automatique" lorsque l'axe est actif.

Tableau 10- 45 Etat du déplacement

Etat	Description
Position cible	Le champ "Position cible" indique la position cible actuelle d'une tâche positionnement active de l'instruction Motion Control "MC_MoveAbsolute" ou "MC_MoveRelative" ou du panneau de commande. La valeur de la position cible n'est valable que pendant l'exécution d'une tâche de positionnement. (variable de l'objet technologique : <nom de l'axe>.MotionStatus.TargetPosition)
Position actuelle	Le champ "Position actuelle" affiche la position actuelle de l'axe. Si l'axe n'est pas référencé, la valeur indique la position par rapport à la position de validation de l'axe. (variable de l'objet technologique : <nom de l'axe>.MotionStatus.Position)
Vitesse actuelle	Le champ "Vitesse actuelle" affiche la vitesse actuelle de l'axe. (variable de l'objet technologique : <nom de l'axe>.MotionStatus.Velocity)

Tableau 10- 46 Limites dynamiques

Limite dynamique	Description
Vitesse	Le champ "Vitesse" affiche la vitesse maximale configurée pour l'axe. (variable de l'objet technologique : <nom de l'axe>.Config.DynamicLimits.MaxVelocity)
Accélération	Le champ "Accélération" affiche l'accélération maximale configurée pour l'axe. (variable de l'objet technologique : <nom de l'axe>.Config.DynamicDefaults.Acceleration)
Décélération	Le champ "Décélération" affiche la décélération maximale configurée pour l'axe. (variable de l'objet technologique : <nom de l'axe>.Config.DynamicDefaults.Deceleration)

Commande de la valeur de début du déplacement

Vous pouvez éditer les valeurs réelles des paramètres de configuration Déplacement afin que le comportement du processus puisse être optimisé en mode en ligne.

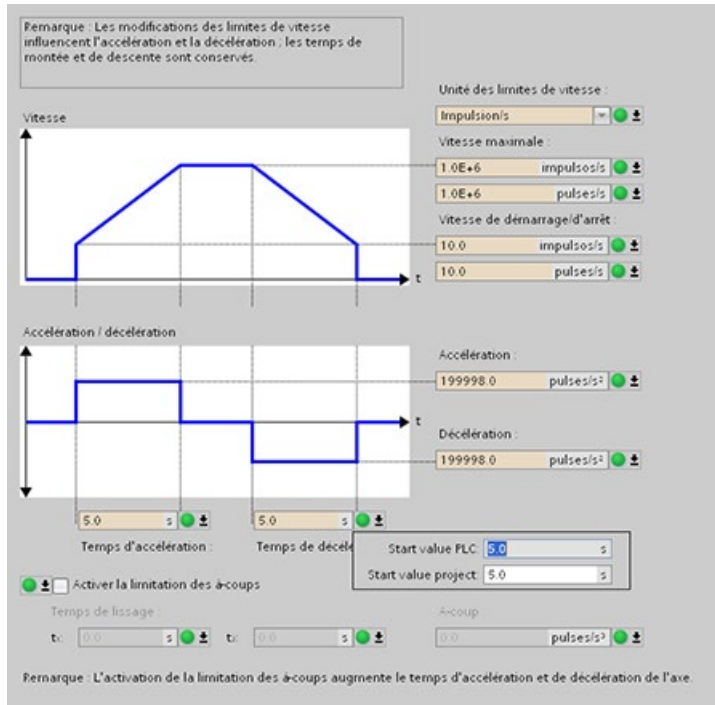
Ouvrez les "Objets technologiques" pour votre Motion Control et son objet "Configuration". Pour accéder à la commande de valeur de début, cliquez sur l'icône lunettes" dans le coin supérieur gauche de la boîte de dialogue :

The screenshot displays the configuration window for dynamic limits. On the left, a tree view shows the hierarchy: Paramètres de base > Général > Paramètres avancés > Dynamique > Général. The main area is titled 'Général' and contains a note: 'Remarque: Les modifications des limites de vitesse influencent l'accélération et la décélération: les temps de montée et de descente sont conservés.' Below the note are two graphs: 'Vitesse' (Velocity) and 'Accélération / décélération' (Acceleration / deceleration). The Velocity graph shows a trapezoidal profile with a constant maximum velocity. The Acceleration/Deceleration graph shows a step function with positive acceleration and negative deceleration. To the right of the graphs are input fields for: 'Unité des limites de vitesse' (Impulsion/s), 'Vitesse maximale' (1.0E+6 Impulsion/s), 'Vitesse de démarrage/arrêt' (10.0 Impulsion/s), 'Accélération' (199998.0 Impulsion/s²), and 'Décélération' (199998.0 Impulsion/s²). Below these are fields for 'Temps d'accélération' (5.0 s) and 'Temps de décélération' (5.0 s). At the bottom, there is a checkbox for 'Activer la limitation des à-coups' (checked), a 'Temps de blocage' field (0.0 s), and an 'À-coups' field (0.0 Impulsion/s). A final note states: 'Remarque: L'activation de la limitation des à-coups augmente le temps d'accélération et de décélération de l'axe.'

10.3 Motion control

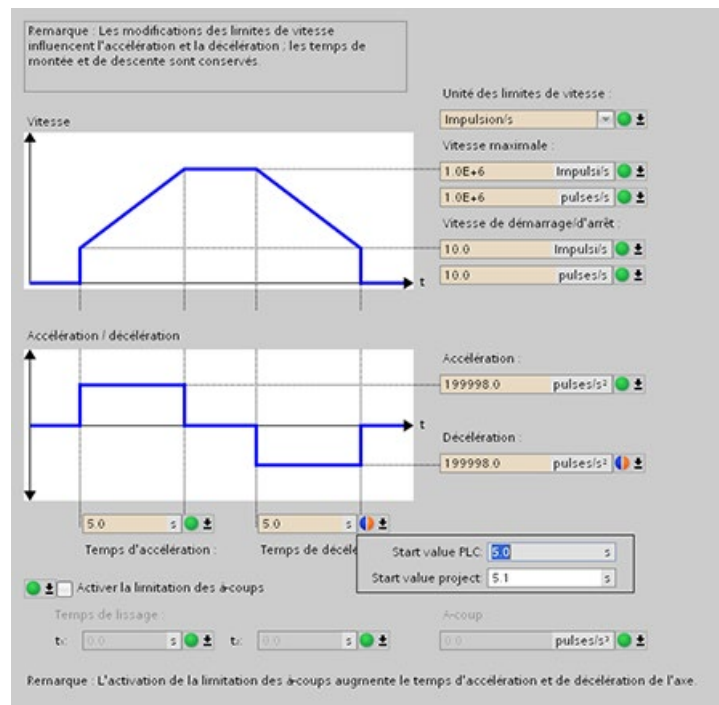
Vous pouvez maintenant modifier la valeur de vos paramètres de configuration Motion Control comme illustré dans la figure ci-dessous.

Vous pouvez comparer la valeur réelle à la valeur de démarrage (hors ligne) du projet et à la valeur de démarrage (en ligne) API de chaque paramètre. Ceci est nécessaire pour comparer les différences en ligne/hors ligne du bloc de données Objet technologique (TO-DB) et pour être informé des valeurs qui seront utilisées comme valeurs actuelles à la prochaine commutation Arrêt-Démarrage de l'API. De plus, une icône de comparaison donne une indication visuelle pour vous aider à identifier facilement les différences en ligne/hors ligne.



La figure ci-dessus montre l'écran de paramètre Déplacement avec des icônes de comparaison montrant quelles valeurs sont différentes entre les projets en ligne et hors ligne. Une icône verte indique que les valeurs sont les mêmes ; une icône bleue/orange indique que les valeurs sont différentes.

En outre, cliquez sur le bouton paramètre avec la flèche vers le bas pour ouvrir une petite fenêtre qui montre la valeur de démarrage (hors ligne) du projet et sur la valeur de démarrage (en ligne) API de chaque paramètre.



10.3.4 Motion control en boucle fermée

10.3.4.1 Configuration de l'axe

Vous connectez l'axe de la boucle fermée sur l'API et l'entraînement par le biais de l'une des deux connexions :

- Entraînement analogique : Cette connexion peut utiliser l'E/S analogique intégrée, SB ou module d'entrées-sorties (SM) ; elle n'utilise pas les PTO. Vous disposez de la résolution E/S analogique suivante pour votre application :
 - Périphérie intégrée : 10 bits (plus faible résolution E/S)
 - E/S Signal Board (SB) : 12 bits
 - Module d'entrées-sorties (SM) : 14 bits (plus haute résolution E/S)
- PROFIdrive : Cette connexion est une solution réseau et elle n'utilise pas de PTO.

L'axe de la boucle fermée requiert également un codeur. Vous pouvez raccorder des codeurs aux dispositifs suivants :

- Interface du codeur sur l'entraînement
- HSC (compteur rapide)
- Modules technologiques (TM)
- Codeur PROFIdrive sur PROFINET / PROFIBUS

Vous ne pouvez pas dépasser un nombre maximal de huit entraînements (ou axes) pour la connexion par PROFIdrive ou par entraînement analogique.

STEP 7 fournit les outils de configuration, les outils de mise en service et les outils de diagnostic pour l'objet technologique "Axe".

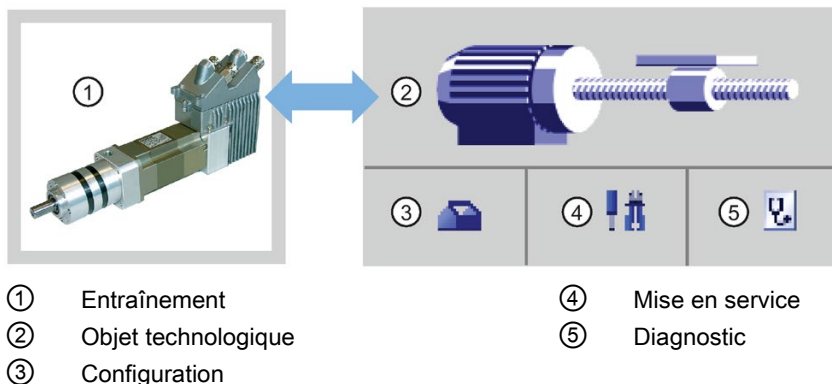


Tableau 10- 47 Outils STEP 7 pour Motion Control en boucle fermée

Outil	Description
Configuration	<p>Configure les propriétés suivantes de l'objet technologique "Axe" :</p> <ul style="list-style-type: none"> • Sélection de la connexion de l'entraînement analogique ou la connexion PROFIdrive à utiliser et configuration de l'interface de l'entraînement et du codeur • Propriétés de la mécanique et du rapport de transmission de l'entraînement et du codeur (ou machine ou système) • Propriétés pour les limites de positionnement, la dynamique et le référencement <p>Sauvegardez la configuration dans le bloc de données de l'objet technologique.</p>
Mise en service	<p>Teste la fonction de votre axe sans avoir à créer un programme utilisateur. Lorsque l'outil est lancé, le panneau de commande s'affiche. Les commandes suivantes sont disponibles sur le panneau de commande :</p> <ul style="list-style-type: none"> • Libérer et bloquer l'axe • Déplacer l'axe en mode Marche par à-coups • Positionnement absolu et relatif de l'axe • Référencer l'axe • Acquitter les erreurs <p>La vitesse et l'accélération/la décélération peuvent être spécifiées pour les commandes de mouvement. Le panneau de commande affiche également l'état en cours de l'axe.</p>
Diagnostic	Surveille l'état en cours et les informations d'erreur concernant l'axe et l'entraînement.

Remarque

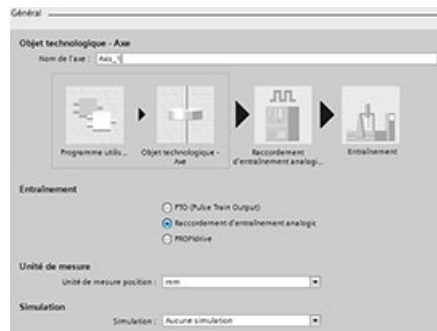
Vous devrez peut-être adapter les valeurs des paramètres d'entrée des instructions Motion Control à la nouvelle unité de mesure dans le programme utilisateur.

Une fois que vous avez créé l'objet technologique pour l'axe, vous configurez l'axe en définissant les paramètres de base, la connexion de l'entraînement analogique ou la connexion PROFIdrive et la configuration de l'interface de l'entraînement et du codeur.

▼ Paramètres de base	✓
Général	✓
Entraînement	✓
Codeur	✓
▼ Paramètres avancés	✓
Mécanique	✓
Modulo	✓
Limites de position	✓
▼ Dynamique	✓
Général	✓
Arrêt d'urgence	✓
▼ Référencement	✓
Activé	✓
Passif	✓
▼ Surveillance de positionnement	✓
Surveillance de positionnement	✓
Écart de traînage	✓
Signal d'immobilisation	✓
Boucle de régulation	✓

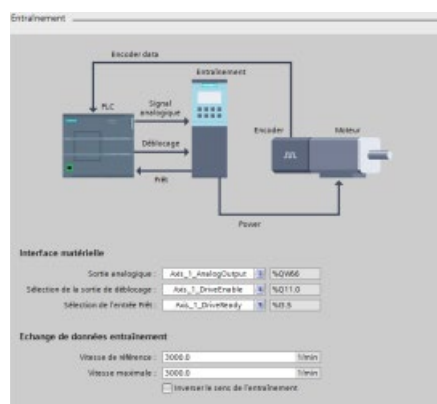
Le sélecteur d'arborescence pour la connexion de l'entraînement analogique ou la connexion PROFIdrive comprend les menus de configuration Codeur, Modulo, Surveillance de position et Boucle de commande.

Configuration de la connexion de l'entraînement analogique



Vous sélectionnez les paramètres suivants dans la boîte de dialogue de configuration générale :

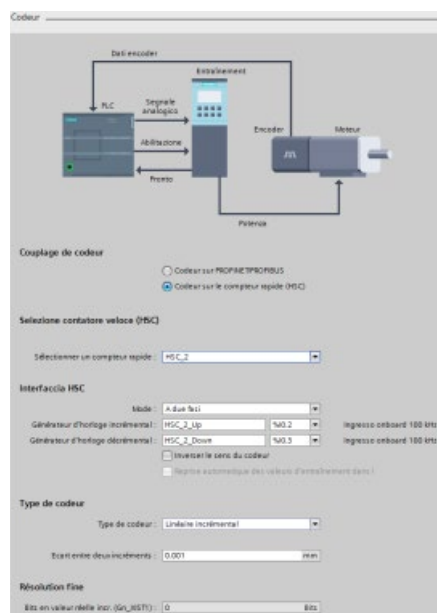
- Bouton radio "Connexion de l'entraînement analogique"
- Unité de mesure



Vous sélectionnez les paramètres suivants dans la boîte de dialogue de configuration de l'entraînement :

- Sorties matérielles de l'entraînement analogique
- Vitesses d'échange de données de l'entraînement

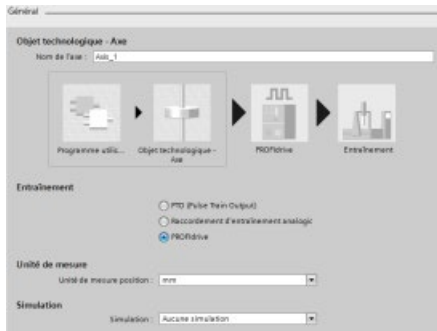
Remarque : La vitesse maximale doit être supérieure ou égale à la vitesse de référence (nominale).



Vous sélectionnez les paramètres suivants dans la boîte de dialogue de configuration du codeur :

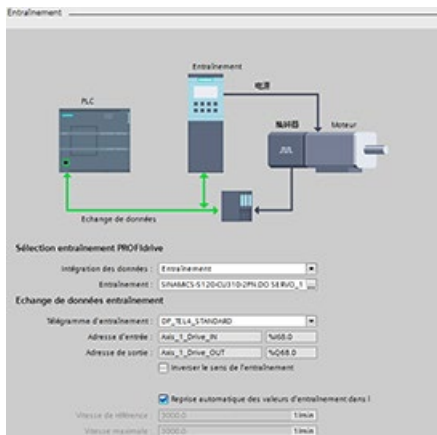
- couplage du codeur de l'entraînement analogique (par exemple un compteur rapide (HSC))
- Interface HSC
- Type de codeur
- Résolution fine

Configuration de PROFdrive



Vous sélectionnez les paramètres suivants dans la boîte de dialogue de configuration générale :

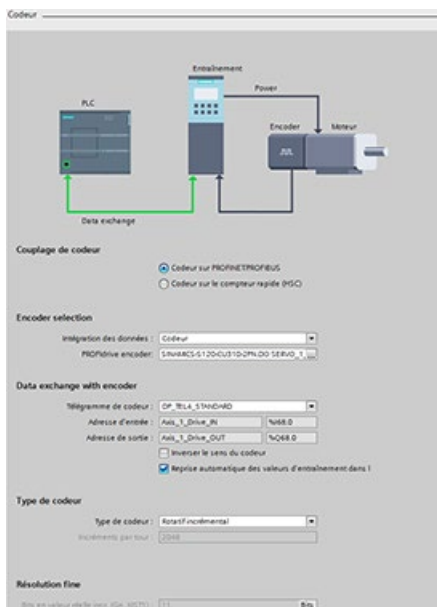
- Bouton radio "PROFdrive"
- Unité de mesure



Vous sélectionnez les paramètres suivants dans la boîte de dialogue de configuration de l'entraînement :

- Entraînement PROFdrive
- Echange de données avec l'entraînement

Remarque : La vitesse maximale doit être supérieure ou égale à la vitesse de référence (nominale).



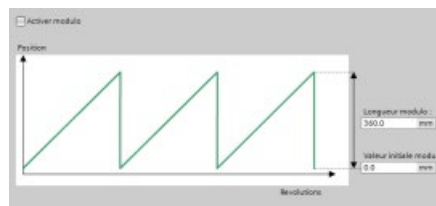
Vous sélectionnez les paramètres suivants dans la boîte de dialogue de configuration du codeur :

- couplage du codeur PROFdrive (par exemple un codeur PROFdrive sur PROFINET)
- Codeur PROFdrive
- Echange de données avec le codeur
- Type de codeur
- Résolution fine

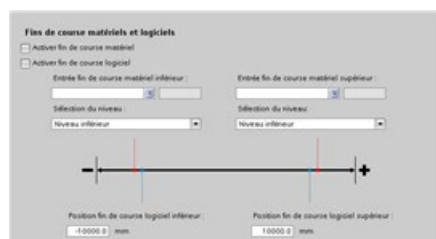
Paramètres avancés

Vous pouvez également configurer les propriétés suivantes de l'axe de la boucle fermée :

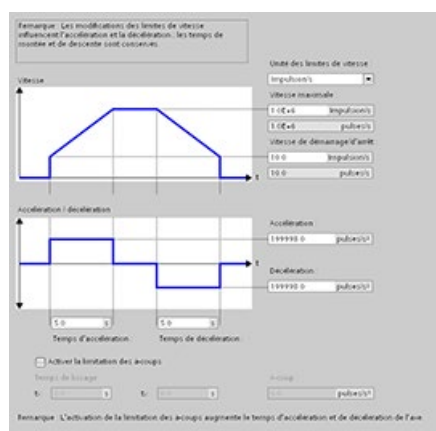
- Modulo
- Limites de position
- Dynamique
- Référencement
- Surveillance de position
- Ecart de traînage
- Signal d'arrêt
- Boucle de régulation



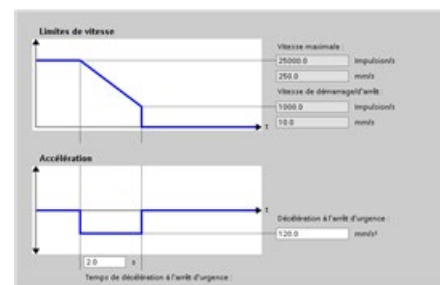
Modulo : Vous pouvez configurer un axe "Modulo" pour déplacer la charge dans une zone cyclique ayant une valeur/position de départ et une longueur définie. Si la position de la charge atteint la fin de cette zone, elle est automatiquement remise à la valeur de départ. Vous activez les champs "Longueur" et "Valeur de départ modulo" lorsque vous cochez la case "Activer modulo".



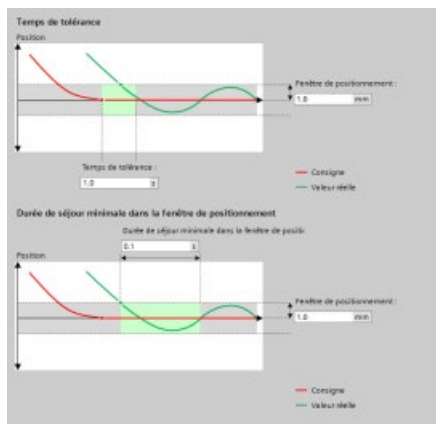
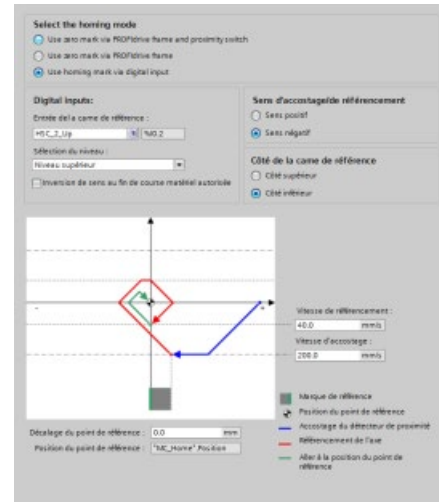
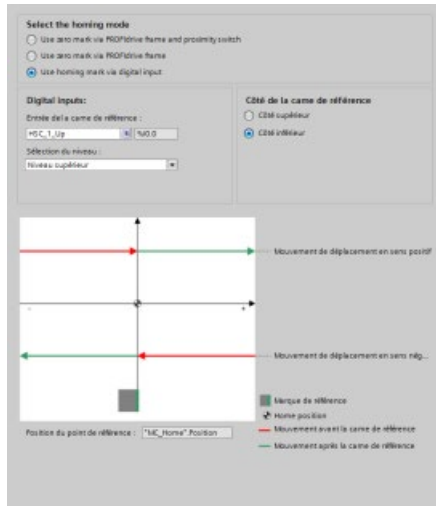
Limites de position : Vous pouvez configurer les propriétés pour les signaux de l'entraînement, la mécanique de l'entraînement et la surveillance de position (fins de course matériels et logiciels).



Dynamique : Vous pouvez configurer la dynamique du mouvement et le comportement de la commande d'arrêt d'urgence.



Référencement : Vous pouvez configurer le comportement de référencement (passif et actif).

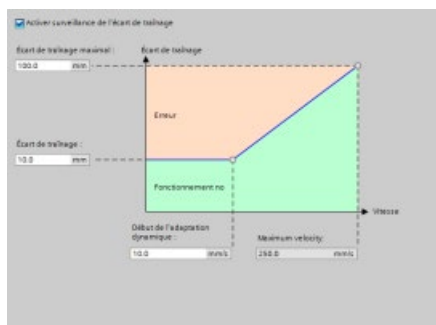


"Surveillance de position" : Vous pouvez configurer le temps de tolérance de même que le temps de maintien minimum pour la fenêtre de positionnement.

Le système connecte les trois paramètres suivants directement à l'axe TO-DB :

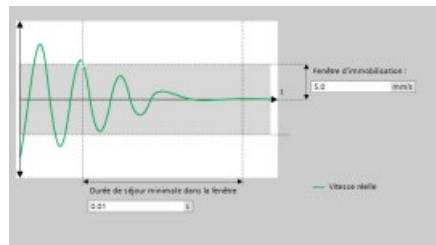
- Fenêtre de positionnement
- Temps de tolérance
- Temps de maintien minimum dans la fenêtre de positionnement

Remarque : Le champ "Fenêtre de positionnement" a pour valeur minimale "0.001" et pour valeur maximale "1 E+12".



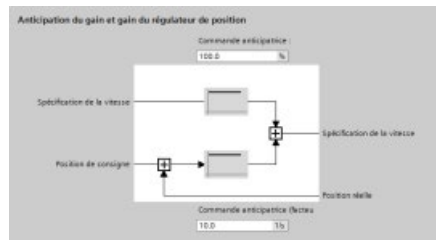
"Ecart de traînage" : Vous pouvez configurer la différence de la distance d'erreur autorisée sur une plage de vitesse. Vous cochez la case "Activer surveillance de l'écart de traînage" pour activer l'écart de traînage. Vous pouvez configurer les paramètres suivants :

- Ecart de traînage maximum
- Ecart de traînage
- Début de l'adaptation dynamique
- Vitesse maximum



"Signal d'arrêt" : Vous pouvez configurer les paramètres suivants :

- Temps de maintien minimum dans la fenêtre d'arrêt
- Fenêtre d'arrêt.



"Boucle de régulation" : Vous pouvez configurer le gain de vitesse dénommé de "Commande anticipatrice (facteur Kv)".

Utilisez le panneau de commande "Mise en service" pour tester la fonctionnalité indépendamment de votre programme utilisateur.



Cliquez sur l'icône "Démarrer" pour mettre l'axe en service.

Le panneau de commande montre l'état en cours de l'axe. Vous pouvez non seulement libérer et bloquer l'axe, mais également tester le positionnement de l'axe (en termes absolus et relatifs) et vous pouvez indiquer la vitesse, l'accélération et la décélération. Vous pouvez également tester les tâches de référencement et le mode Marche par à-coups. Le panneau de commande vous permet en outre d'acquiescer les erreurs.

10.3.4.2 ServoOB

Lorsque vous créez un objet technologique Motion Control (commande de mouvement) pour S7-1200, la CPU crée automatiquement le bloc d'organisation "MC-Servo (OB 91)" pour le traitement des objets technologiques. La fonctionnalité Motion Control des objets technologiques crée sa propre classe de priorité et le système exécutif du SIMATIC S7-1200 appelle le bloc d'organisation conformément au cycle d'application.

L'OB MC-Servo est protégé en écriture. Vous ne pouvez donc pas modifier son contenu.

Les algorithmes de régulation de position de tous les objets technologiques configurés pour Motion Control sur la CPU sont calculés dans l'OB MC-Servo.

Vous pouvez définir le cycle d'application et la priorité du bloc d'organisation conformément à vos exigences en matière de qualité de la commande et de charge du système. Plusieurs blocs d'organisation ServoOB peuvent être associés à un événement "CyclicServoEvent". La page des propriétés de l'OB MC-Servo montre les deux valeurs paramétrables de l'événement "CyclicServoEvent" :

- Numéro de priorité
- Période

L'OB MC-PreServo et l'OB MC-PostServo sont des instances de ServoOB et sont de type ServoOB. Ces blocs d'organisation sont facultatifs ; ils existent uniquement si le bloc d'organisation MC-Servo est présent et ils contiennent du code utilisateur. Les trois OB MC-PreServo, MC-Servo et MC-PostServo doivent s'exécuter dans le même niveau Runtime et le même événement "CyclicServoEvent" les déclenche. Le firmware de l'API les exécute séquentiellement, conformément à leur numéro de bloc :

Instance d'OB	Numéro d'OB	Événement ASOM	Événement ESOM
MC-PreServo	67	ServoOB	-
MC-Servo	91	ServoOB	ServoOB
MC-PostServo	95	ServoOB	-

Contrairement au modèle d'objet du système d'automatisation (ASOM), le modèle d'objet du système d'ingénierie (ESOM) relie l'événement uniquement au bloc d'organisation MC-Servo. Il ne le relie pas aux blocs d'organisation MC-PreServo et MC-PostServo.

Configuration de l'axe avec un cycle d'application MC-Servo (OB 91)

Cycle d'application de l'OB 91 MC-Servo

Vous pouvez définir le cycle d'application auquel l'OB MC-Servo est appelé dans les propriétés du bloc d'organisation :

- Synchrones au bus : vous appelez l'OB MC-Servo de manière synchrone au système de bus. Vous réglez la cadence d'émission dans les propriétés du système de bus sélectionné.
- Cyclique : vous appelez l'OB MC-Servo cycliquement avec le cycle d'application affecté.

Pour éviter toute perturbation lors de l'exécution du programme sur la CPU, définissez le cycle d'application en fonction du nombre d'axes utilisés comme suit :

cycle d'application = nombre d'axes x 2 ms

Nombre d'axes	Cycle d'application
1	2 ms
2	4 ms
4	8 ms
8	16 ms

L'entraînement SINAMICS G120 met à jour la mémoire image du processus de l'entraînement toutes les 4 ms. Pour améliorer la commande, réglez le cycle d'application de l'OB 91 MC-Servo à 4 ms ou à un multiple de 4 ms.

Débordement de l'OB 91 MC-Servo

Le cycle d'application sélectionné doit être suffisamment long pour permettre le traitement de tous les objets technologiques de Motion Control dans un cycle. Si le cycle d'application n'est pas respecté, un débordement se produit.

La CPU ne passe pas à l'état ARRÊT en cas de débordement de l'OB 91 MC-Servo (l'affirmation de l'aide en ligne de TIA Portal concernant le passage à l'ARRÊT de la CPU en cas de débordement de l'OB MC-Servo (OB 91) est incorrecte).

Si nécessaire, vous pouvez mettre la CPU à l'arrêt lors du débordement de l'OB MC-Servo (OB 91) à l'aide d'un OB d'erreur de temps (OB 80).

Mémoire image partielle MIP OB Servo

Pour une commande optimale, affectez tous les modules d'E/S utilisés par Motion Control (les fins de course matériels, par exemple) à la mémoire image partielle "MIP OB Servo". Cette affectation assure un traitement des modules d'E/S en simultanéité avec l'objet technologique.

Si Motion Control utilise un compteur rapide (HSC), Motion Control affecte automatiquement celui-ci à la mémoire image partielle "MIP OB Servo".

10.3.4.3 Fonctionnement en régulation de vitesse

Avec le fonctionnement en régulation de vitesse, vous pouvez déplacer un axe de positionnement à vitesse régulée. Vous pouvez débloquent l'axe avec l'instruction MC_Power. Avec les instructions MC_MoveVelocity et MC_MoveJog, vous pouvez déplacer l'axe avec des consignes de vitesse. L'axe peut être déplacé malgré une erreur de capteur, et ce même si aucune valeur réelle valide n'est disponible. Le mode "Régulation de vitesse" agit comme suit :

- Il désactive le régulateur de position de l'axe.
- Il transmet des consignes de vitesse directement à l'entraînement.
- Il met la consigne de position de l'axe à zéro.
- Il actualise la position en cours de l'axe en présence de valeurs de capteur valides.
- Il ne définit pas l'écart de traînage et l'écart de régulation et les met à zéro.

Vous pouvez activer et désactiver le mode "Régulation de vitesse" avec les trois instructions et paramètres suivants :

- MC_Power.StartMode (Int)
- MC_MoveVelocity.PositionControlled (Bool)
- MC_MoveJog.PositionControlled (Bool)

MC_Power

Avec l'instruction MC_Power, vous pouvez débloquer l'axe en mode "Régulation de vitesse". Cela est possible si aucune valeur de capteur valide n'est disponible ou si l'axe ne peut pas passer en mode "Régulation de position".

Vous pouvez utiliser uniquement les modes de démarrage StartMode "0" et "1". Les autres valeurs signalent une erreur :

StartMode	Axe PTO	Axe servo
0	Non pris en compte	Régulation de vitesse
1	Non pris en compte	Régulation de position
Autre	Mode invalide MC_Power.ErrorID = 0x8412 MC_Power.ErrorID = 0x0011	Mode invalide MC_Power.ErrorID = 0x8412 MC_Power.ErrorID = 0x0011

Pendant le blocage de l'objet technologique Axe avec "MC_Power.Enable = FALSE", vous obtenez le mode de fonctionnement. L'axe réagit différemment selon le mode de fonctionnement :

StartMode	Mode "Régulation de position"	Mode "Régulation de vitesse"
0 : arrêt d'urgence	L'axe décélère de manière régulée en position avec "DynamicDefaults. EmergencyDeceleration" sur la base de ActualPosition et ActualVelocity.	L'axe décélère de manière régulée en vitesse avec "DynamicDefaults. EmergencyDeceleration" sur la base de ActualPosition et ActualVelocity.
1 : arrêt immédiat	L'entraînement s'arrête avec la rampe "AUS3" dans l'entraînement connecté.	L'entraînement s'arrête avec la rampe "AUS3" dans l'entraînement connecté.
2 : arrêt d'urgence avec commande d'à-coup	Les freins de l'axe sont régulés en position à la décélération d'urgence configurée sur la base de la position de consigne. Si la commande d'à-coup est activée, l'à-coup configuré est pris en compte.	Les freins de l'axe sont régulés en vitesse à la décélération d'urgence configurée sur la base de la vitesse de consigne. Si la commande d'à-coup est activée, l'à-coup configuré est pris en compte.

Le mode de fonctionnement ne peut pas être modifié pendant la désactivation de l'entraînement, car aucune nouvelle commande de mouvement n'est acceptée jusqu'à l'arrêt de l'entraînement. Une fois l'entraînement immobilisé, l'axe peut à nouveau être débloqué.

MC_MoveVelocity/MC_MoveJog

Indépendamment du mode de fonctionnement (régulation de vitesse / régulation de position), les blocs font passer l'axe au mode configuré. Cela peut se produire pendant un mouvement en boucle fermée ou à l'arrêt.

MC_Halt

L'instruction MC_Halt ne modifie pas le mode de fonctionnement. En mode "Régulation de vitesse", la rampe de décélération calculée dépend de la vitesse de consigne avec la décélération configurée.

Si la fenêtre d'arrêt est atteinte, la commande est achevée et le paramètre Done prend la valeur TRUE.

Autres commandes de mouvement

Le mode "Régulation de vitesse" reste actif jusqu'à ce que l'une des commandes suivantes devienne active :

- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity (PositionControlled = TRUE)
- MC_MoveJog (PositionControlled = TRUE)
- MC_Home :
 - Référencement actif (mode 3)
 - Les autres modes (passif, direct, référencement du codeur absolu) sont rejetés avec les codes d'erreur suivants :
 - ErrorId 8207 : Commande rejetée
 - ErrorInfo 006B : Appel interdit en mode "Régulation de vitesse"

Bloc de données d'objet technologique Axe

Le DB d'objet technologique Axe affiche le mode de fonctionnement si l'une des trois instructions de mouvement MC_Power, MC_MoveVelocity ou MC_MoveJog est exécutée en mode "Régulation de vitesse" :

- Axis.Statusbit.NonPositionControlled = TRUE
- Axis.Position = 0.0

Calcul de la nouvelle position de consigne

La position de consigne doit être recalculée après un passage du mode Régulation de vitesse au mode Boucle fermée pendant le mouvement :

- Avec commande anticipatrice ($k_{pc} > 0$) : $Position = ActualPosition + ActualVelocity * v_{tc}$
- Sans commande anticipatrice ($k_{pc} = 0$) : $Position = ActualPosition + ActualVelocity / kv$
(v_{tc} = constante de temps de remplacement de la commande anticipatrice ; kv = gain de la commande de mouvement)

Fins de course logiciels

Les fins de course logiciels ne sont pas actifs en mode Régulation de vitesse.

Fins de course matériels

Les fins de course matériels sont pris en charge en mode Régulation de vitesse.

10.3.4.4 Prise en charge du télégramme 4

Le télégramme 4 PROFIdrive contient des valeurs d'actionneur et deux valeurs provenant de différents codeurs. La première valeur de capteur provient du codeur situé sur le moteur. La deuxième valeur de capteur est fournie par un codeur supplémentaire sur la machine.

Le codeur de la machine est raccordé directement à une Control Unit SINAMICS qui transmet les deux valeurs de capteur dans le télégramme 4.

Boîte de dialogue de configuration Entraînement de l'axe

Vous configurez le télégramme 4 dans la configuration matérielle et vous pouvez le sélectionner dans la boîte de dialogue de configuration Entraînement de l'axe.

Boîte de dialogue de configuration Codeur de l'axe

Vous disposez de deux options dans la boîte de dialogue de configuration Codeur de l'axe :

- Codeur sur PROFINET/PROFIBUS
- Codeur sur le compteur rapide (HSC)

"Codeur sur PROFINET/PROFIBUS" est l'option prise par défaut, mais les deux options sont disponibles.

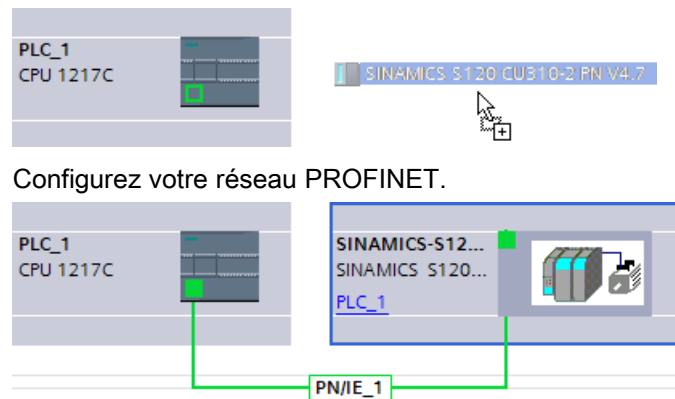
Exemple : configuration du SINAMICS S120 et du codeur avec le télégramme 4

1. Sélection de l'entraînement SINAMICS :

Utilisez le catalogue du matériel pour ajouter un entraînement SINAMICS S120 CU310-2 PN V4.7 : Pour ce faire, affichez le détail des dossiers suivants :

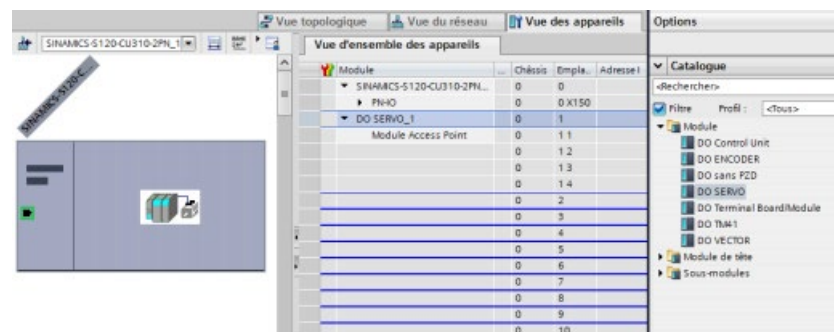
- Autres appareils de terrain
- PROFINET IO
- Entraînements
- SIEMENS AG
- SINAMICS

Insérez l'entraînement comme illustré dans les figures ci-dessous :



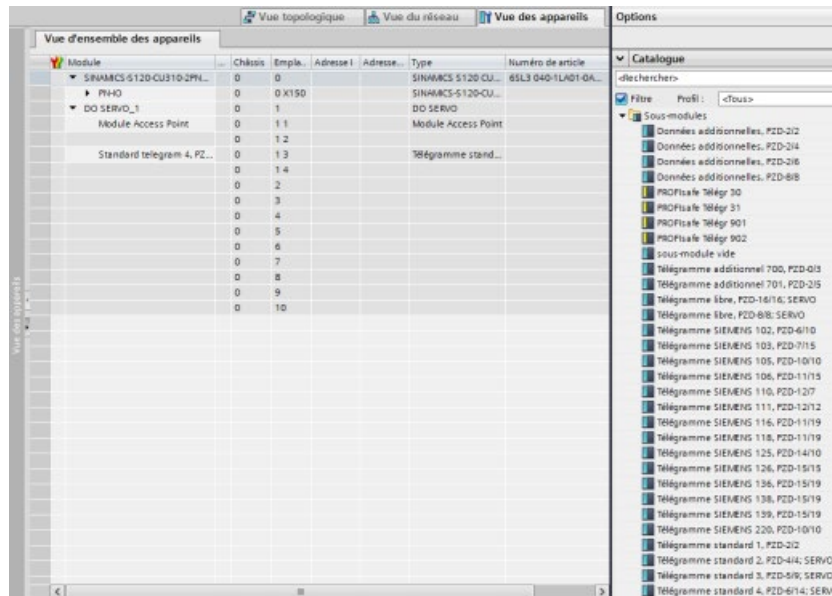
2. Sélection de l'objet entraînement DO SERVO :

- Dans la vue du réseau, double-cliquez sur l'entraînement SINAMICS S120 CU310-2 PN V4.7.
- Ouvrez la vue d'ensemble des appareils.
- Dans le catalogue du matériel, affichez le détail du dossier Module.
- Double-cliquez sur l'objet entraînement DO SERVO ou faites-le glisser pour l'insérer dans la première ligne vierge :



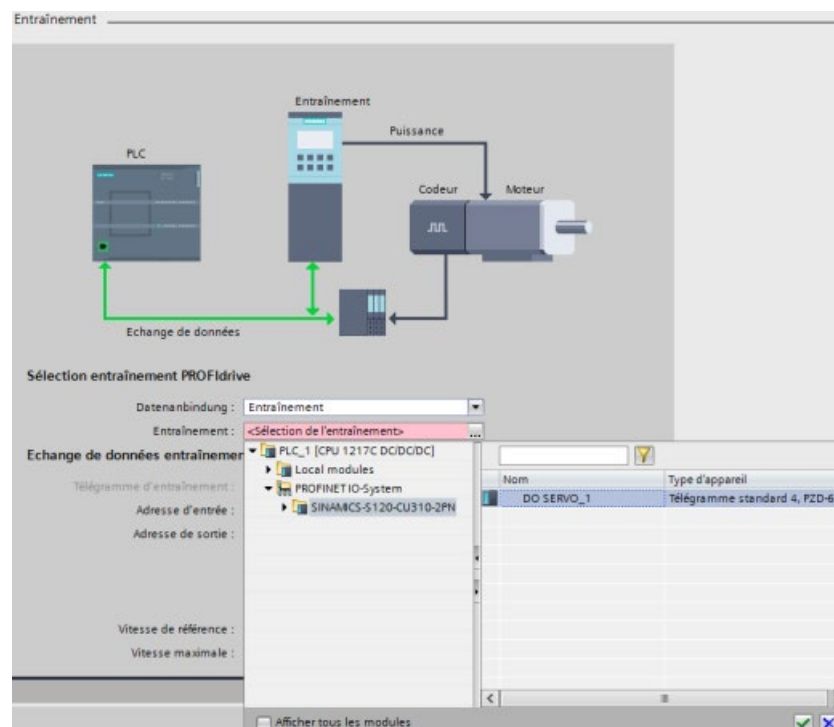
3. Sélection du télégramme 4 :

- Dans le catalogue du matériel, affichez le détail du dossier Sous-modules.
- Double-cliquez sur "Télégramme standard 4, PZD-6/15 ; SERVO" ou faites-le glisser pour l'insérer dans la deuxième ligne vierge.
- Vous devez sauter une ligne pour insérer le télégramme 4 comme illustré dans la figure ci-dessous :



4. Sélection du codeur dans la boîte de dialogue de configuration Entraînement :

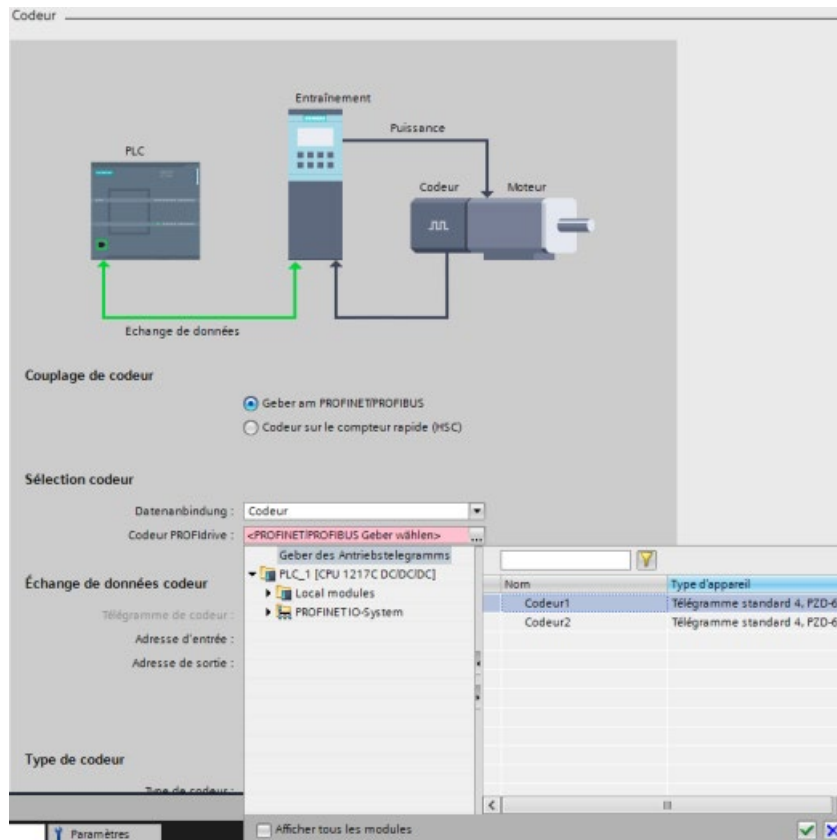
- Naviguez jusqu'à la boîte de dialogue de configuration de l'axe, Paramètres de base, boîte de dialogue de configuration Entraînement.
- Allez dans le champ "Entraînement" sous Sélection entraînement PROFIdrive.
- Cliquez sur les points de suspension.
- Double-cliquez sur "Réseau PROFINET IO" pour ouvrir son dossier.
- Cliquez sur "SINAMICS-S120-CU310-2PN".
- "DO SERVO_1 : Télégramme standard 4, PZD-6/14 ; SERVO" s'affiche dans le volet de droite.
- Cliquez sur la coche verte pour entrer la configuration.



5. Configuration du codeur :

- Comme vous avez sélectionné le télégramme 4 dans la boîte de dialogue de configuration Entraînement de l'axe, la boîte de dialogue de configuration Codeur fournit une nouvelle entrée "Codeur du télégramme d'entraînement" dans le navigateur du projet. Lorsque vous sélectionnez l'entrée "Codeur du télégramme d'entraînement", le panneau de droite affiche deux entrées Codeur1 et Codeur2 avec des valeurs de codeur.
- Motion control définit l'un des codeurs comme codeur de l'entraînement et l'autre comme codeur de la machine. Lorsque vous sélectionnez Codeur1 ou Codeur2, vous déterminez quel codeur est défini comme codeur de l'entraînement. Le codeur que vous ne sélectionnez pas est défini comme codeur de la machine.

- Une fois le codeur d'entraînement sélectionné, cliquez sur la coche verte pour entrer la configuration :



10.3.4.5 Axe de simulation

Vous utilisez le mode Simulation lorsque vous voulez travailler avec un axe à entraînement PROFIdrive ou analogique dans un API sans qu'un entraînement soit raccordé.

Le mode Simulation est requis pour exécuter l'une des tâches suivantes :

- Mise en service d'une séquence de programme en l'absence d'entraînement
- Test du programme utilisateur sans déplacement d'axe
- Simulation du comportement de l'axe à l'aide d'un modèle de processus sans mouvement réel
- Ni entraînement ni codeur ne doivent être raccordés. Le test est également possible sans entraînement configuré ; l'entraînement sera ajouté et configuré a posteriori.

Configuration du mode Simulation

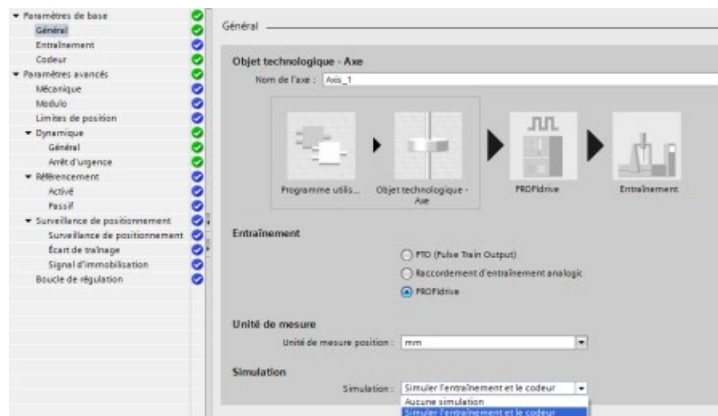
Les options suivantes sont disponibles :

Valeur DB TO	Choix dans la boîte de dialogue	Commentaire
0	Aucune simulation	Aucune simulation
1	Simuler l'entraînement et le codeur	Simulation avec et sans adresses configurées correctes pour l'axe et E/S configurées dans la configuration matérielle

Vous pouvez utiliser le mode Simulation uniquement avec l'axe Servo et donc uniquement avec l'entraînement PROFIdrive et analogique. Pour un axe PTO, le mode de simulation est "0".

TIA Portal doit implémenter le reste du mode Simulation en cas de commutation du type d'axe à PTO. Vérifiez en outre, avec "Adaptation des données", que le mode de simulation est "0" pour PTO ; si ce n'est pas le cas, cela doit être défini automatiquement.

L'utilisateur sélectionne les différents modes de simulation dans la boîte de dialogue Paramètres de base > section Général > champ Simulation, comme illustré dans la figure suivante :



Mode 1 : "Simuler l'entraînement et le codeur"

- Dans ce mode, l'axe dans votre programme est mis en œuvre et fonctionne dans un API sans qu'un codeur et un entraînement PROFIdrive ne soient matériellement raccordés.
- Les adresses d'E/S logiques ne sont pas requises dans le bloc de données d'objet technologique. Cela signifie que vous n'avez besoin de configurer ni capteur, ni entraînement, ni signaux TOR supplémentaires (fins de course matériels et cames de référence) dans l'objet technologique et dans la configuration matérielle. L'axe ne transmet pas de données aux adresses logiques du capteur et de l'entraînement.
- Ce mode simule la vitesse ActualSpeed du DB d'objet technologique.
- Il n'utilise pas le télégramme PROFIdrive si PROFIdrive a été sélectionné.
- Dans le cas de PROFIdrive, l'automate émet une erreur de diagnostic signalant que le matériel configuré (périphériques ou entraînements) manque, mais cela n'a pas d'effet sur les possibilités d'application de l'axe.

- Des rapports de référencement sont créés directement dans tous les modes de référencement. Les positions et l'état sont définis en conséquence. Les entrées matérielles ne sont pas détectées.
- Toutes les connexions peuvent être vides.
- Le panneau de commande / d'optimisation de l'axe fonctionne comme à l'accoutumée.
- Ce mode prend en charge l'axe en régulation de vitesse.
- Ce mode n'accepte pas d'axe virtuel (comme défini par SMC). Vous pouvez toutefois utiliser un axe simulé sans connexions matérielles comme un axe virtuel.

Chargement :

Vous pouvez charger l'axe avec le mode de simulation modifié à l'état MARCHE de l'API. Dans ce cas, l'API met le bit d'état RestartRequired à 1. Après le redémarrage de l'axe, l'API transfère les modifications dans la mémoire de travail. Le tableau suivant montre tous les paramètres de l'axe qui sont connectés à la configuration matérielle (actionneur, capteur et limites de positionnement) et indique s'ils sont obligatoires ou facultatifs pour la simulation :

Paramètre DB TO	Mode 1 : Simuler PROFIdrive
Actor.Interface.AddressIn	Facultatif
Actor.Interface.AddressOut	Facultatif
Actor.Interface.EnableDriveOutput	Facultatif
Actor.Interface.DriveReadyInput	Facultatif
Sensor.Interface.AddressIn	Facultatif
Sensor.Interface.AddressOut	Facultatif
Sensor.ActiveHoming.DigitalInputAddress	Facultatif
Sensor.PassiveHoming.DigitalInputAddress	Facultatif
PositionLimits_HW.MinSwitchAddress	Facultatif
PositionLimits_HW.MaxSwitchAddress	Facultatif

10.3.4.6 Adaptation des données

Vue d'ensemble

Vous pouvez adapter les données d'actionneur et de capteur qui peuvent être lues dans le module entraînement ou capteur et qui ont été configurées de manière identique à la fois dans l'automate et dans l'entraînement/le capteur.

Configuration de l'adaptation RT

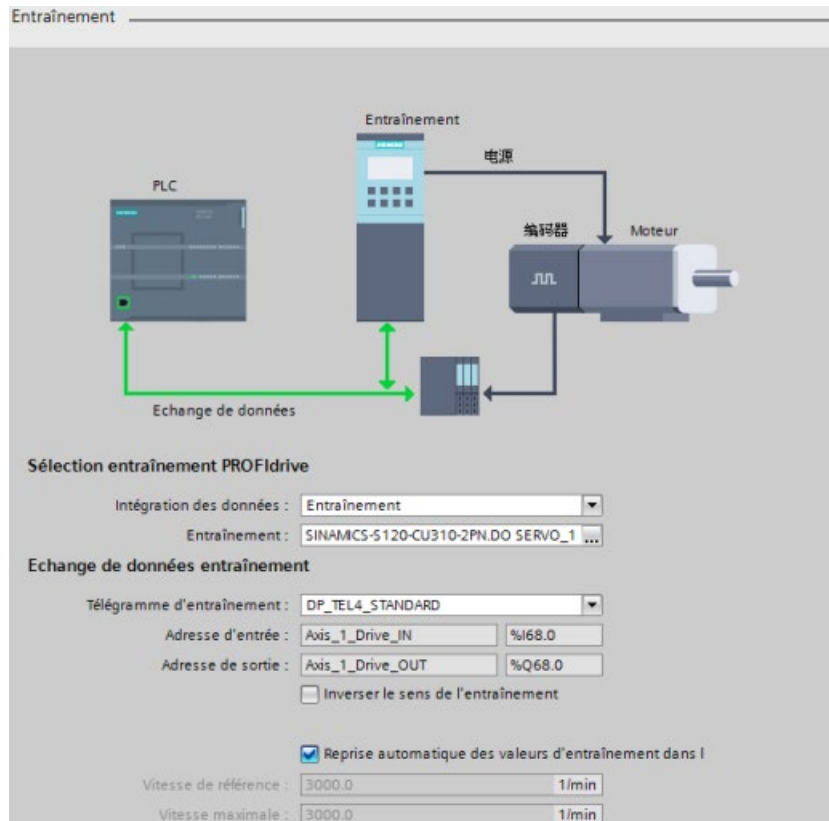
La CPU configure l'adaptation RT dans le DB d'objet technologique pour l'actionneur et le capteur concernés :

- <axe>.Sensor[i].DataAdaptation : DINT [0 : non | 1 : oui]
- <axe>.Actor.DataAdaptation : DINT [0 : non | 1 : oui]

Vous pouvez modifier la configuration de l'adaptation RT à l'exécution dans la boîte de dialogue de configuration de l'axe, Paramètres de base, boîtes de dialogue de configuration Entraînement et Codeur. L'adaptation prend effet au démarrage ou redémarrage de l'objet technologique ou en cas de perte de communication avec l'entraînement.

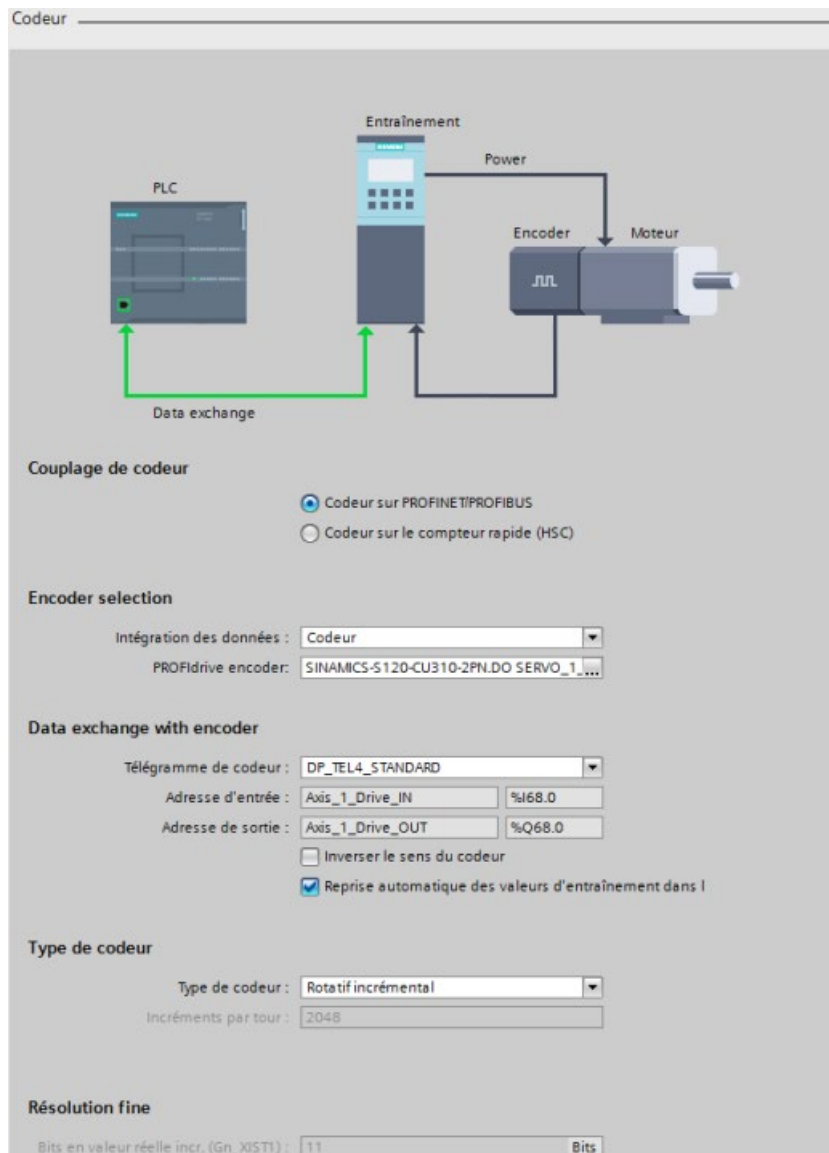
Dans TIA Portal, vous sélectionnez l'adaptation des données en cochant la case "Reprise automatique des valeurs d'entraînement dans l'appareil" présente dans les deux boîtes de dialogue Entraînement et Codeur sous Paramètres de base dans la boîte de dialogue de configuration de l'axe.

La figure suivante montre la boîte de dialogue Entraînement dans la configuration de l'axe :



- Si un entraînement SINAMICS est raccordé :
 - La case est cochée par défaut.
 - Les champs "Vitesse de référence" et "Vitesse maximale" sous la case à cocher sont grisés.
 - L'adaptation RT pour les données d'actionneur est activée.
 - Si vous décochez la case manuellement, vous pouvez modifier les valeurs dans les deux champs sous la case à cocher.
- Si aucun entraînement, SINAMICS ou autre, n'est raccordé :
 - La case n'est pas cochée et l'option est grisée par défaut.
 - Les champs "Vitesse de référence" et "Vitesse maximale" sous la case à cocher sont actifs et leur valeur est modifiable.
 - L'adaptation RT pour les données d'actionneur n'est pas activée.

La figure suivante montre la boîte de dialogue Codeur dans la configuration de l'axe :



- Si un codeur PROFIdrive est raccordé :
 - La case est cochée par défaut.
 - Les champs "Type de codeur" (uniquement "Incréments par tour") et "Résolution fine" sous la case à cocher sont grisés.
 - L'adaptation RT pour les données de codeur est activée.
 - Si vous décochez la case manuellement, vous pouvez modifier les valeurs dans les deux champs sous la case à cocher.
- Si aucun codeur PROFIdrive n'est raccordé (HSC ou modules TM) :
 - La case n'est pas cochée et l'option est grisée par défaut.
 - Les champs "Type de codeur" et "Résolution fine" sous la case à cocher sont actifs et leur valeur est modifiable.
 - L'adaptation RT pour les données d'actionneur n'est pas activée.

Exécution de l'adaptation RT

L'adaptation s'exécute lorsque l'adaptation RT est sélectionnée :

- au démarrage de l'objet technologique (ou au démarrage de l'automate) ou à la création de l'objet technologique (lors du chargement du DB d'objet technologique),
- après une interruption de l'entraînement ou du codeur (ou une coupure de la communication) suivie d'un redémarrage,
- au redémarrage de l'objet technologique (les données sont adaptées après chaque redémarrage).

Quand la communication pour l'adaptation RT est active, Motion Control refuse toutes les commandes de communication du programme utilisateur en lien avec cet entraînement et signale une erreur de ressource dans le programme utilisateur.

La perte de l'entraînement pendant la communication d'adaptation RT provoque l'interruption de l'adaptation. L'adaptation redémarre si l'entraînement répond à nouveau.

Si l'entraînement émet une confirmation négative, Motion Control n'écrase pas la configuration de l'objet technologique Axe. Si l'adaptation RT est sélectionnée mais que le système ne parvient pas à mener l'adaptation à bien, Motion Control signale une erreur et l'appareil ne peut pas être activé.

Affichage de l'état et des erreurs d'adaptation

Motion control signale l'état et les erreurs de l'adaptation RT par des paramètres et des codes et informations d'erreur (ErrorID/ErrorInfo) :

- Motion control signale l'état de l'adaptation sur l'axe : Si l'adaptation ne fonctionne pas ou entraîne des erreurs, Motion control signale une erreur sur l'axe ainsi que dans les paramètres ErrorID / ErrorInfo de l'instruction MC_Power.
- Paramètre d'adaptation général : le fabricant, l'appareil, la version et la date sont lus dans le paramètre P964[7]. La paramètre P964 est un paramètre PROFIdrive qui est disponible au niveau de l'actionneur et du codeur.
- Performances : Motion control exécute l'adaptation RT après le démarrage ou le redémarrage. Si la configuration n'est pas modifiée ensuite, il est recommandé de désactiver l'adaptation des données pour des raisons de performances. Il existe deux solutions :
 - Enregistrer les valeurs par application dans RT après l'adaptation et désactiver l'adaptation
 - Charger les valeurs après l'adaptation et les transférer dans le projet et désactiver l'adaptation
- Comportement de l'adaptation :
 - Au démarrage et au redémarrage, tous les appareils configurés (actionneur et capteur) qui doivent être adaptés le sont.
 - Si un appareil ne peut pas être adapté à ce stade, Motion control signale une erreur et l'état passe à "ADAPTATION_ERROR".
 - Des erreurs au niveau du codeur n'empêchent pas le déblocage de l'axe en mode Régulation de vitesse, car ce mode opère généralement sans codeur.

Une liste des codes et informations d'erreur pour l'adaptation des données figure sous "Valeurs ErrorId et ErrorInfo pour Motion Control".

Adaptation des données d'actionneur

L'adaptation des données d'actionneur est propre à l'appareil et prise en charge uniquement pour les entraînements SINAMICS. Ceux-ci prennent en charge l'adaptation des données et signalent une erreur si l'adaptation RT est activée.

Unités

L'adaptation des données d'actionneur tient compte des unités configurées dans l'entraînement. Les valeurs et les unités dépendent du type DO, du module de fonction DO et du système d'unités.

L'adaptation des données d'actionneur ne prend en charge que les entraînements rotatifs (pas les entraînements linéaires) et les unités 1/min.

Les paramètres suivants sont lus et évalués :

- p107 (type DO)
- p108 (module de fonction DO)
- p505 (unités SI ou unités US)

Sélection de l'enregistrement en cours de l'entraînement

Un entraînement SINAMICS prend en charge différents jeux de données pour le codeur et l'actionneur. Il adapte l'enregistrement en cours au moment de l'adaptation. Ainsi, "p51" (enregistrement en cours de l'entraînement) est lu :

- La vitesse de référence est égale à tous les enregistrements : "p2000" (vitesse de référence) est indépendant de l'enregistrement.
- La vitesse maximale est indépendante de l'enregistrement : "p1082"
- L'entraînement SINAMICS ne vérifie pas les différentes valeurs maximales des différents enregistrements si elles sont cohérentes eu égard à leur valeur de réglage dans l'enregistrement en cours. Cela se produit, par exemple, après une adaptation des données à un autre enregistrement.

Type de moteur

SINAMICS prend en charge deux types de moteur :

- Moteur linéaire
- Moteur par défaut (moteur rotatif)

Avec Basic Motion Control (BMC), SINAMICS accepte uniquement les moteurs rotatifs.

Si un moteur linéaire est configuré pour l'entraînement SINAMICS (paramètre "r108, bit 12"), Motion control signale une erreur après un contrôle de cohérence ou une annulation de l'adaptation des données.

Paramètres

Motion Control prend en charge les objets entraînement "DO-Servo" et "DO-Vector". Les paramètres suivants de l'entraînement sont adaptés :

Paramètre d'actionneur DB TO	Paramètre SINAMICS
Actor.DriveParameter.ReferenceSpeed	p2000
Actor.DriveParameter.MaxSpeed	p1082

Trois paramètres du DB d'objet technologique de l'actionneur ne sont pas adaptés, mais leur vraisemblance est vérifiée :

Paramètre de capteur DB TO	Paramètre SINAMICS/PROFIdrive
Actor.type	r108, bit 12
Actor.Interface.AddressIn.RID	p922 ou p2079
Actor.Interface.AddressOut.RID	p922 ou p2079

Les valeurs de chaque paramètre sont présentées dans les tableaux suivants :

Paramètre d'actionneur DB TO	Valeur
Actor.type	<ul style="list-style-type: none"> • 0 = Analogique • 1 = PROFIdrive • 2 = PTO Remarque : TIA Portal prend en charge uniquement l'entraînement rotatif, ce qui signifie que seule la valeur 1 est valide.
Actor.Interface.AddressIn.RID	<ul style="list-style-type: none"> • 0208_0708 = Télégramme 1 • 0208_070A = Télégramme 2 • 0208_070C = Télégramme 3 • 0208_0720 = Télégramme 4 • 0208_070E = Télégramme 81 • 0208_0710 = Télégramme 83
Actor.Interface.AddressOut.RID	<ul style="list-style-type: none"> • 0208_0709 = Télégramme 1 • 0208_070B = Télégramme 2 • 0208_070D = Télégramme 3 • 0208_0721 = Télégramme 4 • 0208_070F = Télégramme 81 • 0208_0711 = Télégramme 83

Paramètre SINAMICS/PROFIdrive	Valeur
r108, bit 12	<ul style="list-style-type: none"> • 0 = Entraînement linéaire • 1 = Entraînement rotatif Remarque : TIA Portal prend en charge uniquement l'entraînement rotatif.
p922 ou p2079	<ul style="list-style-type: none"> • 1 = Télégramme 1 • 2 = Télégramme 2 • 3 = Télégramme 3 • 4 = Télégramme 4 • 81 = Télégramme 81 • 83 = Télégramme 83

Contrôle de la vitesse maximale

TIA Portal vérifie la validité du paramètre "DynamicLimits.MaxVelocity" du DB d'objet technologique. Toutefois, avec l'adaptation des données, le contrôle n'est possible que lorsque le système est à l'état MARCHE et n'exécute pas la vérification TIA Portal.

Contrôle de cohérence

Motion Control exécute un contrôle de cohérence au premier démarrage ou au redémarrage de l'objet technologique Axe. Motion Control signale également une erreur si l'adaptation des données de l'actionneur est activée. Le contrôle de cohérence concerne le télégramme, le type de moteur et la vitesse maximale :

- Télégramme dans "p922" ou "p2079" : Motion Control signale une erreur en cas d'incohérence entre l'objet technologique et la configuration de l'entraînement.
- Type de moteur dans "r108, bit 12" de DO Servo : Motion Control signale une erreur en cas d'incohérence entre l'objet technologique et la configuration de l'entraînement.
- La vitesse maximale ne peut pas être atteinte avec les paramètres d'axe configurés et Motion Control signale une erreur. Remarque : Après confirmation, Motion Control ne signale plus cette erreur si la configuration de l'axe n'est pas modifiée.

Cas spécial :

- vitesse maximale (p1082) > 2 x vitesse de référence (p2000)
- Motion Control réduit en interne la vitesse maximale à deux fois la vitesse de référence. La saisie dans la configuration de l'axe n'est pas limitée. Toutefois, Motion Control adapte la valeur de "p1082" et signale une erreur.

Adaptation des données de capteur

Seule l'adaptation des données avec un codeur actif est prise en charge (p0979).

Volume des données adaptées

Motion Control adapte la description de la valeur actuelle de "p0979" (configuration de codeur rotatif ou linéaire comprise) et vérifie le paramètre "type de codeur" dans TIA Portal (incrémental ou absolu). Motion Control n'adapte pas et n'évalue pas la référence NIST dans le télégramme 83.

Corrélation entre le codeur dans le télégramme et le codeur dans l'automate et l'entraînement

- Motion Control configure la correspondance entre un codeur dans l'automate et la valeur actuelle dans le télégramme (valeur actuelle 1 ou valeur actuelle 2) dans le VREF.RID des deux paramètres suivants du DB d'axe technologique :
 - <axe>.Sensor[i].Interface.AddressIn
 - <axe>.Sensor[i].Interface.AddressOut
- La correspondance dans l'entraînement se fait avec la valeur d'indice dans "p979" (modélisation du codeur dans SINAMICS).

Paramètres

Motion Control adapte les paramètres suivants :

Paramètre de capteur DB TO	Paramètre SINAMICS/PROFIdrive
Sensor[i].System	<ul style="list-style-type: none"> • P979.[1] ou P979.[11] • Bit 0 = 0 : codeur rotatif • Bit 0 = 1 : codeur linéaire
Codeur rotatif incrémental	
Sensor[i].Parameter.StepsPerRevolution	P979.[2] ou P979.[12]
Sensor[i].Parameter.FineResolutionXist1	P979.[3] ou P979.[13]
Codeur linéaire incrémental	
Sensor[i].Parameter.Resolution	P979.[2] ou P979.[12]
Sensor[i].Parameter.FineResolutionXist1	P979.[3] ou P979.[13]
Codeur rotatif absolu	
Sensor[i].Parameter.StepsPerRevolution	P979.[2] ou P979.[12]
Sensor[i].Parameter.FineResolutionXist1	P979.[3] ou P979.[13]
Sensor[i].Parameter.DeterminableRevolutions	P979.[5] ou P979.[15]
Sensor[i].Parameter.FineResolutionXist2	P979.[4] ou P979.[14]
Codeur linéaire absolu	
Sensor[i].Parameter.Resolution	P979.[2] ou P979.[12]
Sensor[i].Parameter.FineResolutionXist1	P979.[3] ou P979.[13]
Sensor[i].Parameter.FineResolutionXist2	P979.[4] ou P979.[14]

Motion Control n'adapte pas un paramètre de capteur du DB d'objet technologique, mais elle en vérifie la cohérence :

Paramètre de capteur DB TO	Paramètre SINAMICS/PROFIdrive
Sensor[i].Type	<ul style="list-style-type: none"> • P979.[5] ou P979.[15] • Bit 0 = 0 : codeur incrémental • Bit 0 > 1 : codeur absolu
Sensor[i].Interface.AddressIn	p922 ou p2079
Sensor[i].Interface.AddressOut	

Vous pouvez utiliser un codeur absolu comme codeur incrémental, mais pas un codeur incrémental comme codeur absolu. Motion Control signale une incompatibilité entre le type de codeur dans l'objet technologique et sa valeur actuelle correspondante dans le télégramme PROFIdrive par une erreur. Remarque : avec SINAMICS FW V2.6, l'interface PROFIdrive comporte un top zéro de codeur.

Contrôle de cohérence

Motion Control exécute un contrôle de cohérence au premier démarrage ou au redémarrage de l'objet technologique Axe. Motion Control signale également une erreur si l'adaptation des données du capteur est activée. Le contrôle de cohérence concerne le télégramme et le type de capteur :

- Télégramme dans "p922" ou "p2079" : Motion Control signale une erreur en cas d'incohérence entre l'objet technologique et la configuration du capteur.
- Type de capteur dans "P979.[5]" ou "P979.[15]" : Vous pouvez utiliser un codeur absolu comme codeur incrémental, mais pas un codeur incrémental comme codeur absolu. Motion Control signale une erreur en cas d'incohérence.

Paramètres devant être chargés

La liste suivante présente tous les paramètres que vous devez charger de l'entraînement dans TIA Portal. Pour gérer cette adaptation des données, vous devez étendre le DB d'objet technologique. Utilisez les structures suivantes du DB d'objet technologique pour réaliser cette extension :

Actor	Type	Valeur par défaut	Modifiable	Commentaire
Ac-tor.DataAdaptation	DINT	0	R (avec redémarrage)	Activation de l'adaptation : <ul style="list-style-type: none"> • 0 : NON • 1 : OUI

StatusDrive	Type	Commentaire
StatusDrive.AdaptationState	DINT	État de l'adaptation : <ul style="list-style-type: none"> • 0 : NOT_ADAPTED : ne peut pas accepter de données. • 1 : IN_ADAPTATION : l'adaptation des données vient de commencer. • 2 : ADAPTED : les données sont adaptées. • 3 : NOT_APPLICABLE : l'adaptation n'est pas sélectionnée ou n'est pas disponible pour cet entraînement. • 4 : ADAPTATION_ERROR : erreur pendant l'adaptation : <ul style="list-style-type: none"> – Impossible de débloquer l'axe – Motion Control signale une erreur de configuration.

Sensor	Type	Valeur par défaut	Modifiable	Commentaire
Sensor.DataAdaptation	DINT	0	R (avec redémarrage)	Activation de l'adaptation : <ul style="list-style-type: none"> • 0 : NON • 1 : OUI

StatusSensor	Type	Commentaire
StatusSensor.AdaptationState	DINT	État de l'adaptation : <ul style="list-style-type: none"> • 0 : NOT_ADAPTED : ne peut pas accepter de données. • 1 : IN_ADAPTATION : l'adaptation des données vient de commencer. • 2 : ADAPTED : Les données sont adaptées. • 3 : NOT_APPLICABLE : L'adaptation n'est pas sélectionnée ou n'est pas disponible pour cet entraînement. • 4 : ADAPTATION_ERROR : erreur pendant l'adaptation : <ul style="list-style-type: none"> – Impossible de débloquer l'axe – Motion Control signale une erreur de configuration.

ErrorWord	Type	Commentaire
...	Bool	
Bit 15 : Erreur d'adaptation	Bool	ErrorID
...	Bool	

10.3.4.7 Commande d'axe utilisant le module TM Pulse

Le module TM Pulse est un module de sortie d'impulsion ET 200SP à deux voies à utiliser avec des vannes et des moteurs. Le module peut prendre en charge soit deux voies 24 V CC à 2 A ou une voie 24 V CC à 4 A.

En utilisant le mode de fonctionnement "moteur à courant continu" du module de sortie TM Pulse 2x24V, vous pouvez entraîner un moteur dans les deux directions avec une sortie PWM bipolaire. Vous pouvez affecter une entrée TOR en tant que signal "Arrêt externe" pour le moteur.

Le module TM Pulse peut exécuter les fonctions suivantes prises en charge par Motion Control :

- Réponse de sortie programmable à une condition ARRET de CPU/Maître
- Détection des erreurs et diagnostic :
 - Tension d'alimentation L+ manquante ou sous-tension
 - Court-circuit/surcharge d'une sortie TOR
 - Court-circuit/sous-tension d'un capteur d'énergie d'alimentation
 - Erreur de surchauffe
 - Défaut de paramétrage
 - Erreur de module/firmware

Exemple

Certaines applications requièrent la capacité de se déplacer facilement vers un emplacement spécifique, mais ne requièrent pas une précision absolue et le respect d'un profil spécifique. L'utilisation d'un moteur à courant continu commun en association avec la commande Closed Loop Servo est possible en utilisant le module TM Pulse ET 200SP pour commander le moteur. Cet exemple d'application couvre un mode du module TM Pulse ET200SP : "PWM avec moteur à courant continu". Pour obtenir plus d'informations sur les autres modes TM Pulse ET200SP, référez-vous au *Manuel TM Pulse 2x24V Module technologique ET200SP*.

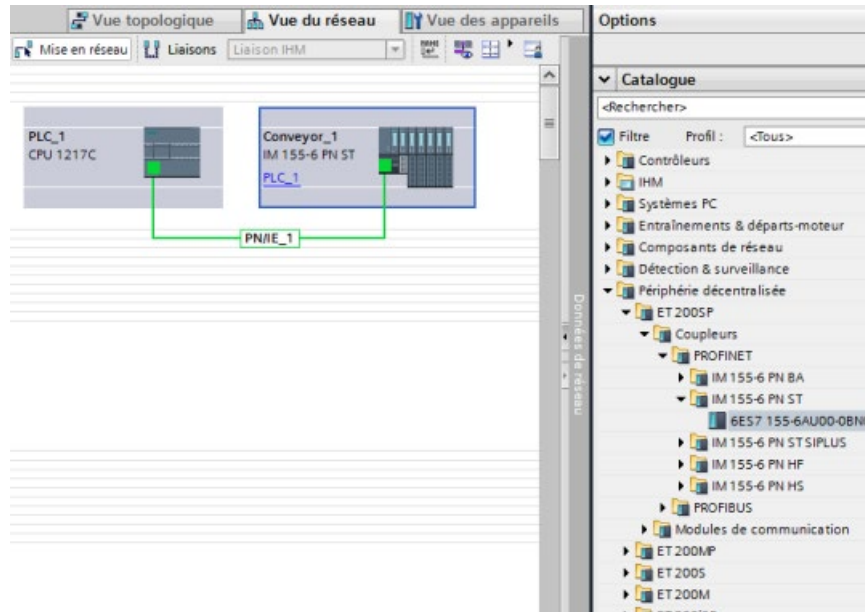
Afin d'utiliser une régulation authentique, vous aurez besoin d'un procédé de signalisation en retour de position. Vous devez connecter un encodeur au moteur afin de fournir une réponse au système de commande. Cet exemple présente trois méthodes pour y parvenir :

- High Speed Counter (HSC) intégré au S7-1200
- Module de comptage ET 200SP TM
- Module TM PosInput ET 200SP

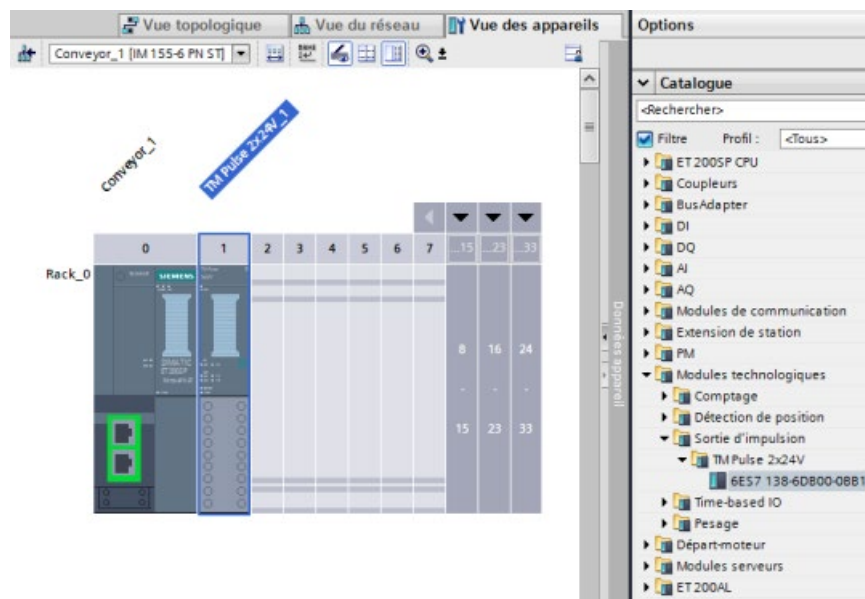
Dans certains cas, lorsque le déplacement à des vitesses différentes est un prérequis, la régulation peut fonctionner en mode de régulation de vitesse. Si vous utilisez ce mode, vous n'avez pas besoin de signalisation en retour de position.

Procédez comme suit pour configurer le module TM Pulse :

1. Configurez votre CPU S7-1200.
2. Sélectionnez le module Interface ET 200SP et placez-le dans la vue de l'appareil :



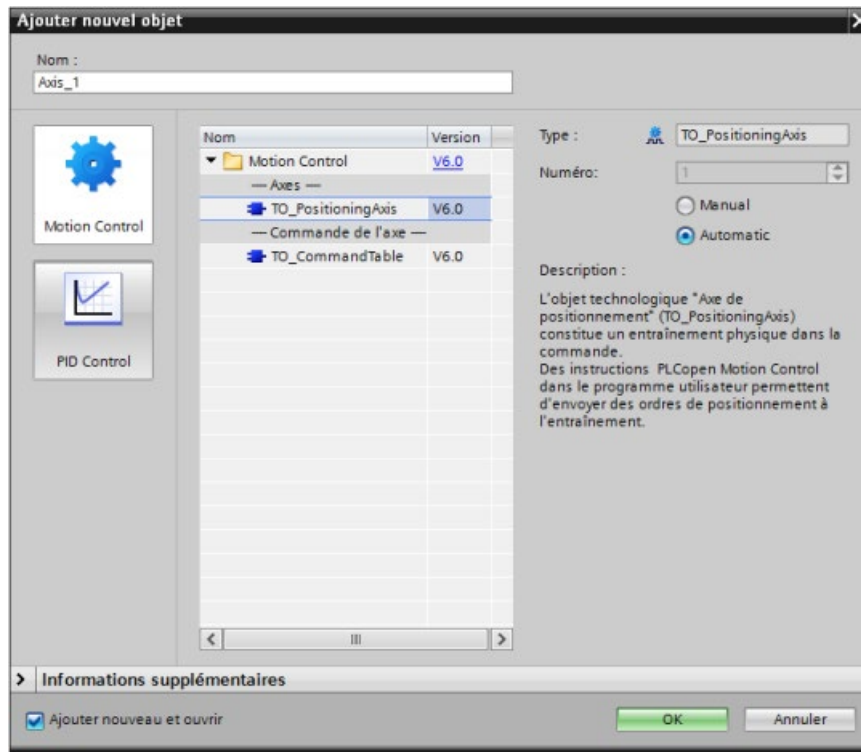
3. Ajoutez le module TM Pulse ET200SP :



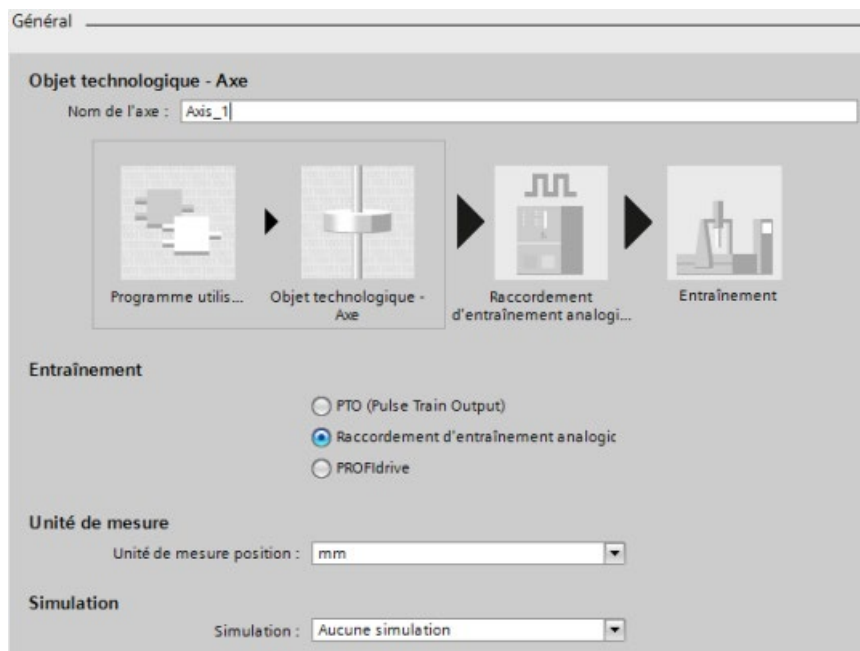
4. Dans la configuration de la voie, sélectionnez "2 voies (2A)".
5. Définissez le mode de fonctionnement "PWM avec moteur à courant continu".
6. Définissez les diagnostics et les paramètres de voie comme requis.

Procédez comme suit pour configurer l'axe de déplacement avec une signalisation en retour de position :

1. Pour la configuration de votre interface avec un système Motion Control en boucle fermée, vous utilisez la commande analogique et non PROFIdrive ou PTO. Configurez la commande de l'axe de la même manière que la configuration d'un axe qui utilise une sortie analogique comme l'entrée vers un servomécanisme. Ajoutez un "TO PositioningAxis" comme indiqué dans la figure ci-dessous :



2. Sélectionnez "Connexion d'entraînement analogique" dans la boîte de dialogue de configuration générale pour la configuration de l'axe :



3. Vous avez maintenant besoin de sorties analogiques et d'une validation d'entraînement pour votre configuration d'entraînement. Passez en vue de l'appareil du module d'interface ET 200SP et à l'entrée de la vue d'ensemble de l'appareil pour le module TM Pulse. En utilisant l'adresse de démarrage Q pour le module TM Pulse et les tableaux de l'interface de commande et de réponse dans le *Manuel TM Pulse 2x24V (6ES7138-6DB00-0BB1) Module technologique ET200SP SIMATIC*, vous pouvez déterminer les sorties analogiques suivantes et la validation d'entraînement requise pour votre entraînement :

- Sorties analogiques
- Sélectionnez une sortie de validation
- Sélectionner une entrée prête à être mise en service

Sélectionnez votre interface matériel E/S et les valeurs d'échange de données dans la boîte de dialogue de configuration d'entraînement pour la configuration de l'axe :

Entraînement

PLC

Entraînement

Codeur

Moteur

Signal analogique

Déblocage

Prêt

Puissance

Données codeur

TM

Echange de données

Interface matérielle

Sortie analogique : Axis_1_AnalogOutput %QW4

Sélection de la sortie de déblocage : Axis_1_DriveEnable %Q11.0

Sélection de l'entrée Prêt : Axis_1_DriveReady %I3.5

Echange de données entraînement

Vitesse de référence : 3000.0 1/min

Vitesse maximale : 3000.0 1/min

Inverser le sens de l'entraînement

4. Dans la boîte de dialogue de configuration de l'encodeur, pour la configuration de l'axe, effectuez la configuration en sélectionnant l'un des encodeurs suivants :

- Module TM Count
- Module TM PosInput
- Compteur rapide (HSC)

Codeur

Couplage de codeur

Codeur sur PROFINET/PROFIBUS
 Codeur sur le compteur rapide (HSC)

Encoder selection

Intégration des données : Codeur
 PROFIdrive encoder: Conveyor_1.TM Count 1x24V_1_Encoder1...

Data exchange with encoder

Télégramme de codeur : DP_TEL83_STANDARD
 Adresse d'entrée : Axis_1_Encoder_IN %I18.0
 Adresse de sortie : Axis_1_Encoder_OUT %Q26.0
 Inverser le sens du codeur
 Reprise automatique des valeurs d'entraînement dans l

Type de codeur

Type de codeur : Rotatif incrémental
 Incréments par tour : 2050

Résolution fine

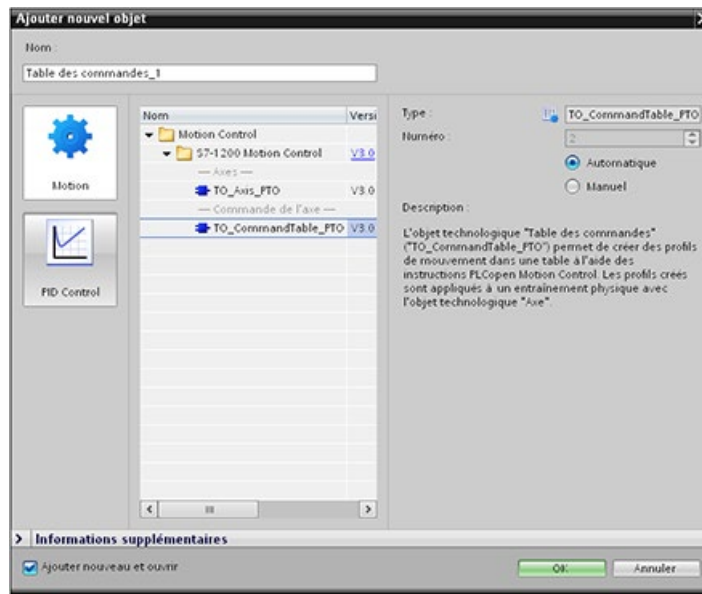
Bits en valeur réelle incr. (Gn_XIST1) : 0 Bits

10.3.5 Configuration de la table de commande TO_CommandTable_PTO

Vous pouvez configurer une instruction MC_CommandTable à l'aide des objets technologiques. L'exemple suivant illustre comme cela est possible.

Ajout d'un objet technologique

1. Dans l'arborescence du projet, affichez le détail du nœud "Objets technologiques" et sélectionnez "Ajouter nouvel objet".
2. Sélectionnez l'icône "CommandTable" (renommez-la si nécessaire) et cliquez sur "OK" pour ouvrir l'éditeur de configuration pour l'objet CommandTable.



Planification des étapes pour votre application

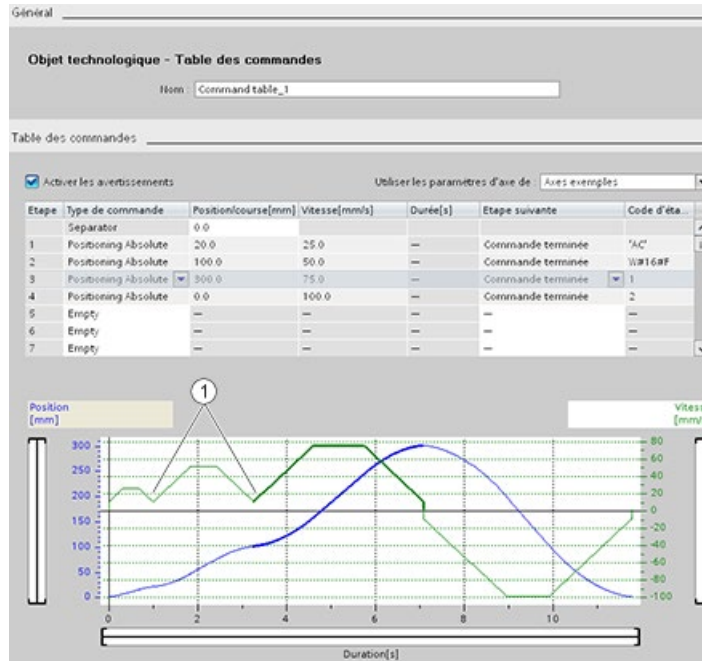
Vous pouvez créer la séquence de mouvement désirée dans la fenêtre de configuration "Table de commande" et vérifier le résultat dans la vue graphique du graphique de tendance.

Vous pouvez sélectionner les types de commandes qui doivent être utilisés pour le traitement de la table de commande. Il est possible d'entrer jusqu'à 32 étapes. Les commandes sont traitées l'une après l'autre, produisant aisément un profil de mouvement complexe.

Tableau 10- 48 Types de commandes MC_CommandTable

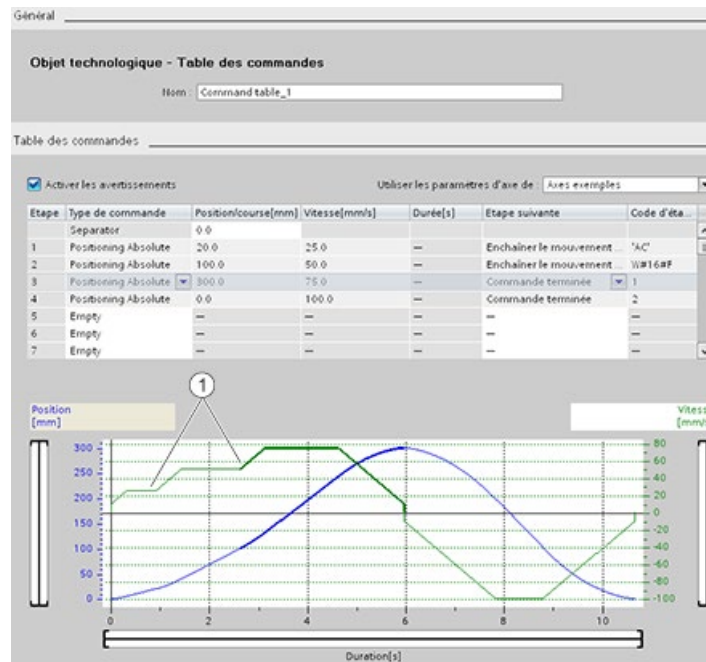
Type de commande	Description
Empty	Marque de réservation pour toute commande à ajouter. L'entrée Empty n'est pas prise en compte lors du traitement de la table de commande.
Halt	Pause de l'axe. Remarque : cette commande n'est exécutée qu'après une commande de consigne de vitesse.
Positioning Relative	Positionnement de l'axe en fonction de la distance. La commande déplace l'axe de la distance indiquée à la vitesse donnée.
Positioning Absolute	Positionnement de l'axe en fonction de l'emplacement. La commande déplace l'axe à l'emplacement indiqué, en utilisant la vitesse donnée.
Velocity setpoint	Déplacement de l'axe à la vitesse donnée.
Wait	Attente de l'expiration de la durée indiquée. "Wait" n'interrompt pas un accostage actif.
Separator	Ajout d'une ligne séparatrice au-dessus de la ligne sélectionnée. La ligne séparatrice permet de définir plus d'un profil dans une seule table de commande.

Dans la figure ci-dessous, "Commande achevée" est utilisé comme transition à l'étape suivante. Ce type de transition permet à votre appareil de décélérer jusqu'à la vitesse de démarrage/d'arrêt, puis de réaccélérer au début de l'étape suivante.



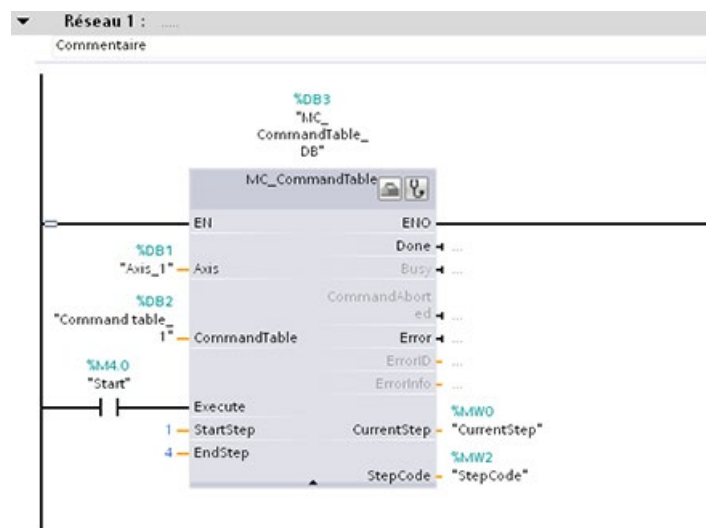
① L'axe décélère jusqu'à la vitesse de démarrage/d'arrêt entre les étapes.

Dans la figure ci-dessous, "Lissage mouvement" est utilisé comme transition à l'étape suivante. Ce type de transition permet à l'appareil de conserver sa vitesse lors de l'entrée dans l'étape suivante, lissant ainsi la transition d'une étape à la suivante pour l'appareil. Le lissage peut réduire le temps total nécessaire à l'exécution d'un profil complet. Sans lissage, l'exécution de cet exemple prend sept secondes. Avec le lissage, le temps d'exécution est réduit d'une seconde pour atteindre un total de six secondes.



① L'axe continue à se déplacer et accélère ou décélère selon la vitesse de l'étape suivante, permettant ainsi un gain de temps et une réduction de l'usure mécanique.

Le fonctionnement de votre table CommandTable est géré par une instruction MC_CommandTable, comme illustré ci-dessous :



10.3.6 Fonctionnement de Motion Control pour S7-1200

10.3.6.1 Sorties de la CPU utilisées pour Motion Control

La CPU fournit quatre générateurs d'impulsions. Chaque générateur d'impulsions fournit une sortie d'impulsions et une sortie de sens pour la commande d'un entraînement de moteur pas à pas ou d'un entraînement de servomoteur à interface d'impulsion. La sortie d'impulsions fournit à l'entraînement les impulsions nécessaires au mouvement du moteur. La sortie de sens commande le sens de déplacement de l'entraînement.

La sortie PTO génère un signal carré de fréquence variable. La génération d'impulsions est gérée par les informations de configuration et d'exécution fournies via la configuration matérielle et/ou des SFC/SFB.

Selon la sélection de l'utilisateur, lorsque la CPU est à l'état MARCHE, ce sont soit les valeurs stockées dans la mémoire image, soit les sorties de génération d'impulsions qui pilotent les sorties TOR. A l'état ARRÊT, le générateur PTO ne pilote pas les sorties.

Les sorties CPU intégrées et les sorties d'un Signal Board peuvent être utilisées comme sorties d'impulsions et de sens. Vous choisissez entre sorties CPU intégrées et sorties de Signal Board pendant la configuration d'appareil, sous "Générateurs d'impulsions (PTO/PWM)" dans l'onglet "Propriétés". Seul PTO (Pulse Train Output) s'applique à Motion Control.

Le tableau ci-dessous montre les affectations par défaut des E/S. Toutefois, les quatre générateurs d'impulsions peuvent être configurés pour utiliser n'importe quelles sortie TOR.

Remarque

Les sorties de trains d'impulsions ne peuvent pas être utilisées par d'autres instructions dans le programme utilisateur.

En cas de configuration des sorties de la CPU ou du Signal Board comme générateurs d'impulsions (pour les instructions PWM ou Motion Control), les adresses des sorties correspondantes ne pilotent plus les sorties. Si votre programme utilisateur écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.

Remarque

Les sorties de sens PTO peuvent être libérées afin d'être utilisées à un autre endroit dans le programme.

Chaque PTO requiert l'affectation de deux sorties : l'une comme sortie d'impulsions et l'autre comme sortie de sens. Il se peut que vous utilisiez uniquement la sortie d'impulsions et pas la sortie de sens. Dans ce cas, vous pouvez libérer la sortie de sens pour une autre utilisation dans votre programme utilisateur. La sortie ne peut pas être utilisée simultanément comme sortie de sens PTO et dans le programme utilisateur.

Tableau 10- 49 Affectations par défaut des sorties d'impulsions et de sens

Utilisation des sorties pour Motion Control		
	Impulsion	Sens
PTO1		
E/S intégrées	Q0.0	Q0.1
E/S du SB	Q4.0	Q4.1
PTO2		
E/S intégrées	Q0.2	Q0.3
E/S du SB	Q4.2 ¹	Q4.3 ¹
PTO3		
E/S intégrées	Q0.4 ²	Q0.5 ²
E/S du SB	Q4.0	Q4.1
PTO4		
E/S intégrées	Q0.6 ³	Q0.7 ³
E/S du SB	Q4.2	Q4.3

¹ Les sorties Q4.2 et Q4.3 sont disponibles uniquement sur le SB 1222 DQ4.

² La CPU 1211C ne comporte pas de sorties Q0.4, Q0.5, Q0.6 et Q0.7. Ces sorties ne peuvent donc pas être utilisées dans la CPU 1211C.

³ La CPU 1212C ne comporte pas de sorties Q0.6 et Q0.7. Ces sorties ne peuvent donc pas être utilisées dans la CPU 1212C.

⁴ Ce tableau s'applique aux fonctions PTO des CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C et CPU 1217C.

Interface d'entraînement

Pour Motion Control, vous pouvez configurer en option une interface d'entraînement pour "Entraînement débloqué" et "Entraînement prêt". Lors de l'utilisation de l'interface d'entraînement, la sortie TOR pour la validation de l'entraînement et l'entrée TOR pour "Entraînement prêt" peuvent être sélectionnées librement.

Remarque

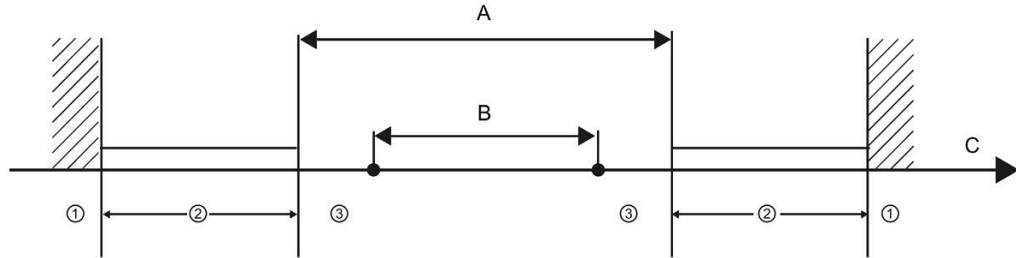
Le firmware prendra le contrôle par le biais des sorties d'impulsions et de sens correspondantes si la sortie PTO (Pulse Train Output) a été sélectionnée et affectée à un axe.

Avec cette mainmise de la fonction de commande, le lien entre la mémoire image et la sortie physique est également rompu. Alors que l'utilisateur peut écrire dans la mémoire image des sorties d'impulsions et de sens via le programme utilisateur ou la table de visualisation, ces valeurs ne sont jamais transférées à la sortie physique. En conséquence, il n'est pas non plus possible de surveiller la sortie physique via le programme utilisateur ou la table de visualisation. Les informations lues ne font que refléter la valeur de la mémoire image et ne correspondent en rien à l'état actuel de la sortie physique.

Pour toutes les autres sorties CPU qui ne sont pas utilisées en permanence par le firmware CPU, l'état de la sortie physique peut être commandé ou surveillé comme d'habitude via la mémoire image.

10.3.6.2 Fins de course matériels et logiciels pour Motion Control

Utilisez les fins de course matériels et logiciels pour limiter la plage de déplacement autorisée et la plage de travail de votre axe.



- ① Butée mécanique
- ② Fins de course matériels inférieur et supérieur
- ③ Fins de course logiciels inférieur et supérieur
- A Plage de déplacement autorisée pour l'axe
- B Plage de travail de l'axe
- C Distance

Il faut activer les fins de course matériels et logiciels avant de les utiliser dans la configuration ou dans le programme utilisateur. Les fins de course logiciels sont actifs uniquement après le référencement de l'axe.

Fins de course matériels

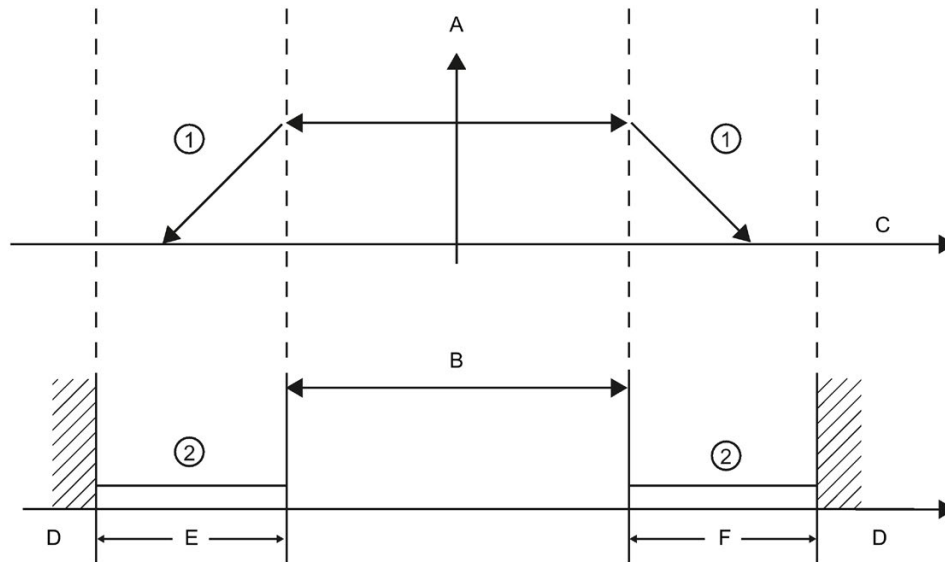
Les fins de course matériels définissent la plage de déplacement maximale de l'axe. Les fins de course matériels sont des commutateurs physiques qui doivent être raccordés à des entrées aptes aux alarmes de la CPU. Utilisez uniquement des fins de course matériels qui restent commutés en permanence après avoir été accostés. Cet état de commutation ne peut être annulé qu'après un retour dans la plage de déplacement autorisée.

Tableau 10- 50 Entrées disponibles pour les limites matérielles

Description	CPU	RPS, LIM- et LIM+ ¹
E/S intégrées	CPU 1211C	I0.0 - I0.5
	CPU 1212C	I0.0 - I0.7
	CPU 1214C, CPU 1215C et CPU 1217C	I0.0 - I01.3
E/S du SB	Toutes les CPU S7-1200	I4.0 - I4.3

¹ Entrée Came de référence (RPS), entrée Limite de déplacement négatif (LIM-) et entrée Limite de déplacement positif (LIM+)

Lors de l'accostage des fins de course matériels, l'axe freine jusqu'à s'immobiliser avec la décélération d'urgence configurée. La décélération d'urgence configurée doit être suffisante pour arrêter l'axe de manière fiable avant la butée mécanique. Le schéma suivant montre le comportement de l'axe après qu'il a accosté les fins de course matériels.



- ① L'axe freine jusqu'à s'immobiliser avec la décélération d'urgence configurée.
- ② Plage dans laquelle les fins de course matériels signalent l'état "accosté".
- A [vitesse]
- B Plage de déplacement autorisée
- C Distance
- D Butée mécanique
- E Fin de course matériel inférieur
- F Fin de course matériel supérieur

ATTENTION

Risques liés à la modification du temps de filtre pour une voie d'entrée TOR

Si le temps de filtre pour une voie d'entrée TOR est modifié par rapport à un paramétrage précédent, une nouvelle valeur d'entrée de niveau "0" doit être présente pendant une durée cumulée allant jusqu'à 20,0 ms pour que le filtre réagisse pleinement aux nouvelles entrées. Pendant ce temps, les événements d'impulsion "0" courts de moins de 20,0 ms peuvent ne pas être détectés ni comptés.

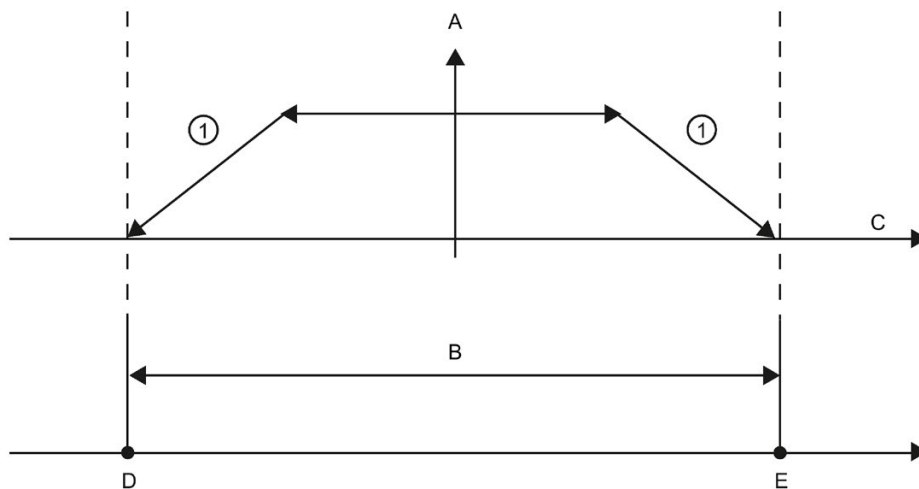
Cette modification des temps de filtre peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Pour garantir la prise en compte immédiate d'un nouveau temps de filtre, mettez la CPU hors tension puis sous tension.

Fins de course logiciels

Les fins de course logiciels limitent la plage de travail de l'axe. Ils doivent se trouver en deçà des fins de course matériels par rapport à la plage de déplacement. Comme les positions des fins de course logiciels sont flexibles, la plage de travail de l'axe peut être restreinte sur une base individuelle en fonction du profil de déplacement en cours. Contrairement aux fins de course matériels, les fins de course logiciels sont réalisés exclusivement au moyen du logiciel et ne nécessitent pas leurs propres commutateurs.

Si des fins de course logiciels sont activés, un déplacement actif est arrêté à la position du fin de course logiciel. L'axe freine avec la décélération configurée. Le schéma suivant montre le comportement de l'axe lorsqu'il atteint les fins de course logiciels.



- ① L'axe freine jusqu'à s'immobiliser avec la décélération configurée.
- A [vitesse]
- B Plage de travail
- C Distance
- D Fin de course logiciel inférieur
- E Fin de course logiciel supérieur

Utilisez des fins de course matériels supplémentaires si une butée mécanique est placée après les fins de course logiciels et qu'il y a risque de dommages mécaniques.

Configuration de détection de front en cas de changement d'adresse

Si vous configurez une limite de position ou une came de référence d'entrée à une adresse d'entrée dans un objet technologique Axe de positionnement, Motion Control la configure automatiquement avec des alarmes sur front. Si vous changez alors la limite de position ou la came de référence d'entrée à une autre adresse, la configuration avec détection de front pour l'ancienne adresse reste active.

Vous pouvez ajouter des fins de course matériels et logiciels dans la boîte de dialogue Limites de position sous Paramètres avancés de la configuration d'axe. Une détection de front est automatiquement activée lorsque vous ajoutez une entrée de fin de course matériel. Si vous décidez ensuite de modifier l'adresse d'entrée, une boîte de dialogue "Détection de front" vous offre les choix suivants :

- Oui : modifie l'adresse, active la détection de front à la nouvelle adresse et désactive la détection de front à l'ancienne adresse (option par défaut).
- Non : modifie l'adresse, active la détection de front à la nouvelle adresse et conserve la détection de front à l'ancienne adresse.
- Annuler : annule la modification de l'adresse et conserve l'état de détection de front actuel.

Choix dans la boîte de dialogue "Détection de front"	Modifie l'adresse	Active la détection de front à la nouvelle adresse	Désactive la détection de front à l'ancienne adresse
Oui (option par défaut)	Oui	Oui	Oui
Non	Oui	Oui	Non
Annuler	Non	Sans objet	Non

Remarque

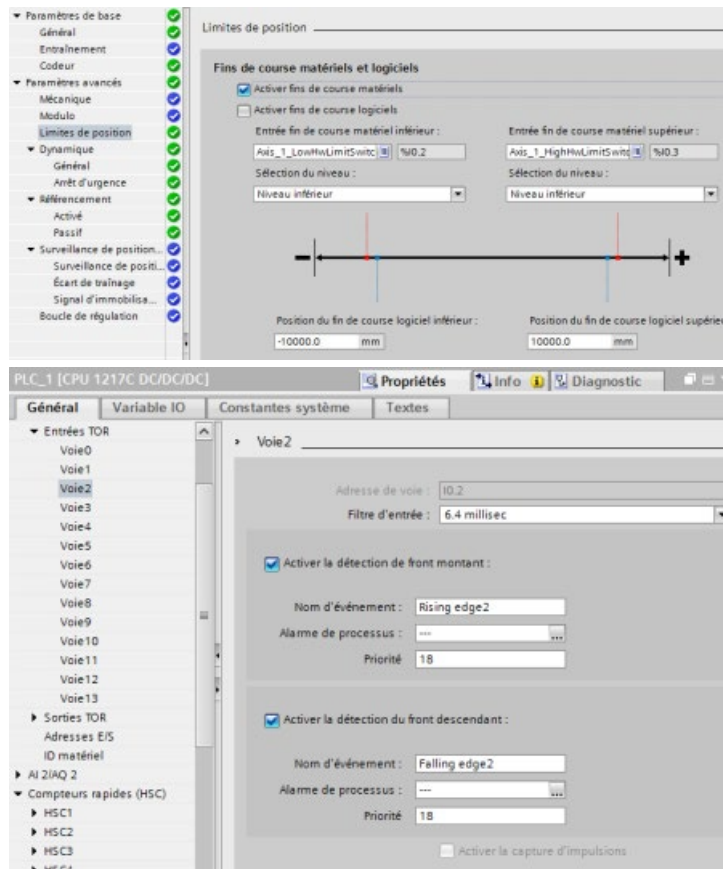
Aucune boîte de dialogue ne s'affiche pour les configurations servo qui ne requièrent pas de détection de front.

Remarque

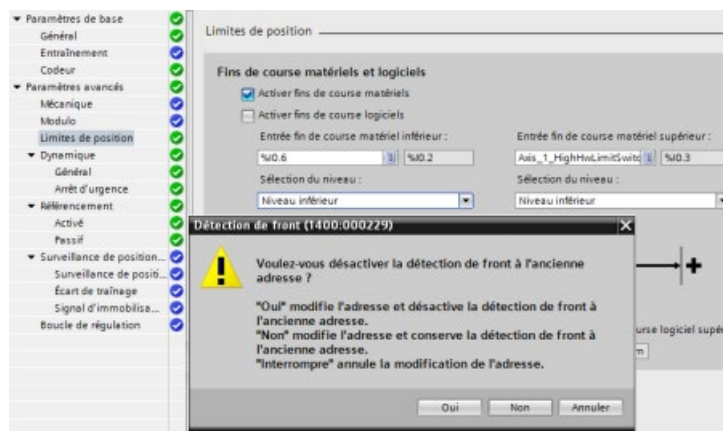
Si un OB est relié à l'entrée, Motion Control n'exécute pas les opérations avec détection de front de la manière décrite ci-dessus.

Exemple : activation d'une nouvelle adresse de fin de course matériel avec détection de front et désactivation de la détection de front à l'ancienne adresse

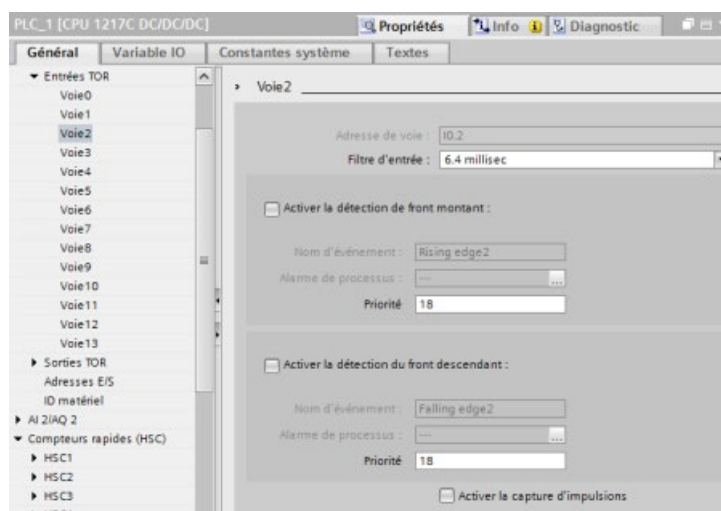
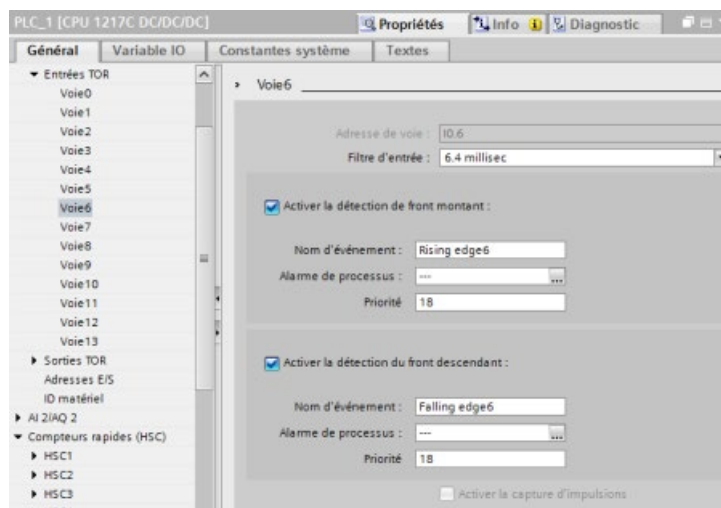
1. État actuel : vous avez associé un fin de course matériel inférieur d'entrée à I0.2. Cette configuration active automatiquement la détection de front en I0.2. Cette configuration apparaît également sous Entrées TOR dans les propriétés de la CPU :



2. Vous changez le fin de course matériel inférieur d'entrée à I0.6 et confirmez. La boîte de dialogue "Détection de front" s'affiche. Vous sélectionnez "Oui : modifie l'adresse et désactive la détection de front à l'ancienne adresse" :

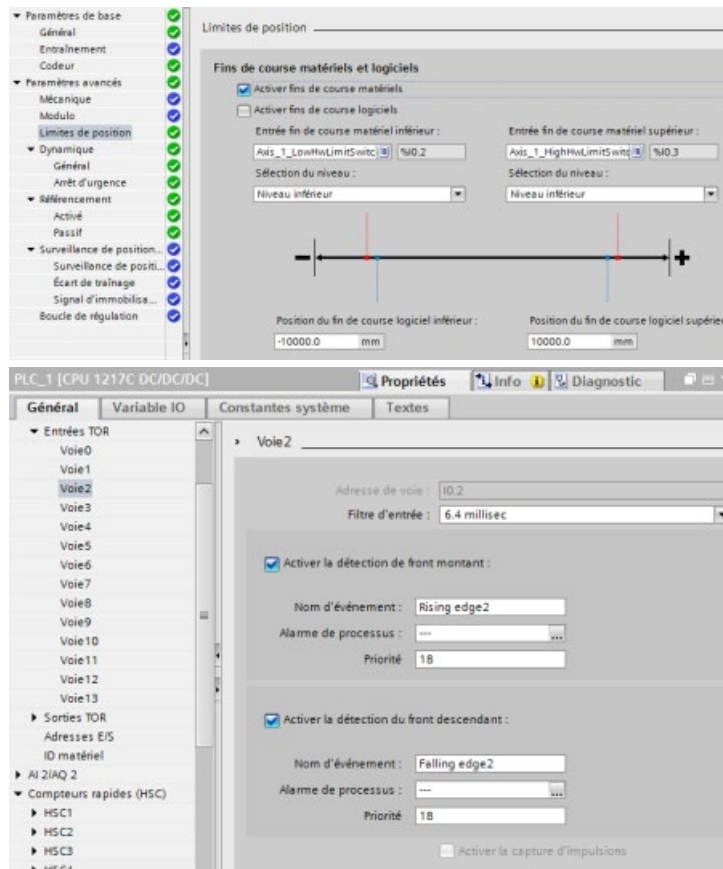


3. La CPU accepte la nouvelle adresse I0.6 et active la détection de front en I0.6. La détection de front en I0.2 est automatiquement désactivée sous Entrées TOR dans les propriétés de la CPU :

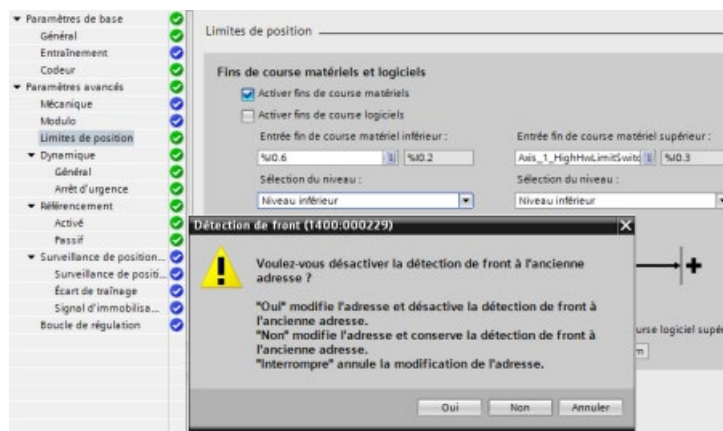


Exemple : activation d'une nouvelle adresse de fin de course matériel avec détection de front et conservation de la détection de front à l'ancienne adresse

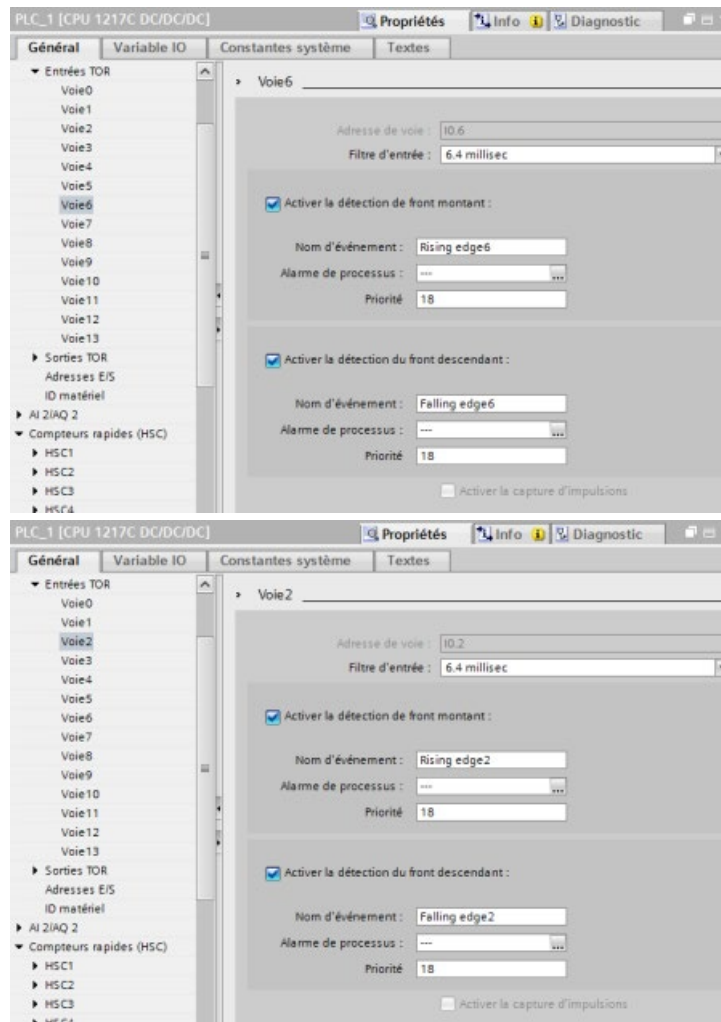
1. État actuel : vous avez associé un fin de course matériel inférieur d'entrée à I0.2. Cette configuration active automatiquement la détection de front en I0.2. Cette configuration apparaît également sous Entrées TOR dans les propriétés de la CPU :



2. Vous changez le fin de course matériel inférieur d'entrée à I0.6 et confirmez. La boîte de dialogue "Détection de front" s'affiche. Vous sélectionnez "Non : modifie l'adresse et conserve la détection de front à l'ancienne adresse" :

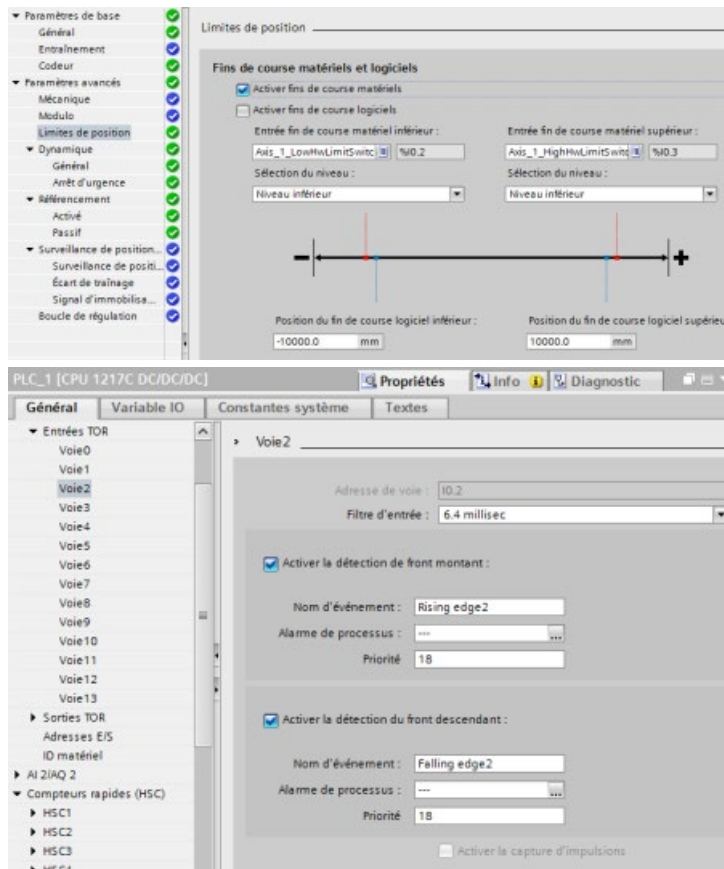


3. La CPU accepte la nouvelle adresse I0.6 et active la détection de front en I0.6. La détection de front en I0.2 reste active sous Entrées TOR dans les propriétés de la CPU :

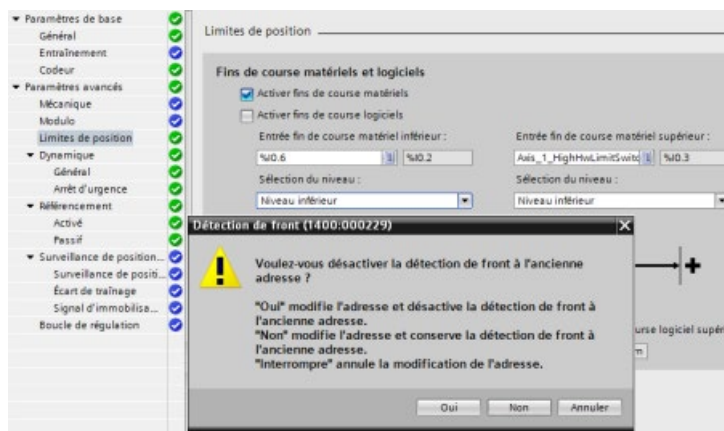


Exemple : annulation de l'activation d'une nouvelle adresse de fin de course matériel avec détection de front

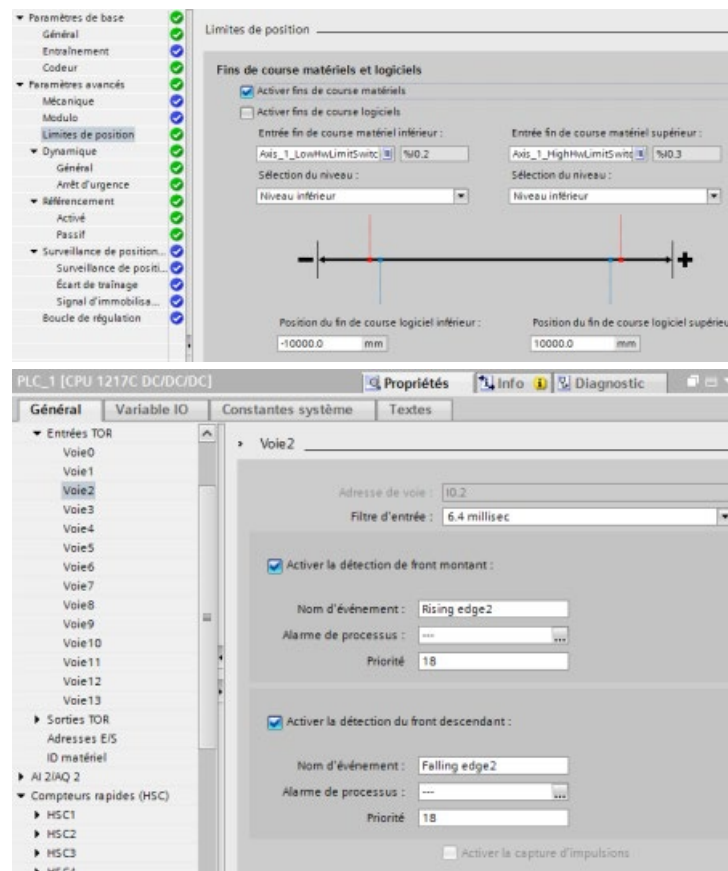
1. État actuel : vous avez associé un fin de course matériel inférieur d'entrée à I0.2. Cette configuration active automatiquement la détection de front en I0.2. Cette configuration apparaît également sous Entrées TOR dans les propriétés de la CPU :



2. Vous changez le fin de course matériel inférieur d'entrée à I0.6 et confirmez. La boîte de dialogue "Détection de front" s'affiche. Vous sélectionnez "Annuler".



3. Le fin de course matériel inférieur d'entrée reste configuré à I0.2 et la détection de front en I0.2 reste active :



Informations supplémentaires

Votre programme utilisateur peut annuler les limites de positionnement matérielles ou logicielles en activant ou en désactivant à la fois la fonctionnalité de limite matérielle et logicielle. La sélection se fait dans le DB de l'axe.

- Pour activer ou désactiver la fonctionnalité de limite matérielle, accédez à la variable "Active" (Bool) dans le chemin de DB "<nom d'axe>/Config/PositonLimits_HW". L'état de la variable "Active" active ou désactive l'utilisation de limites de positionnement matérielles.
- Pour activer ou désactiver la fonctionnalité de limite de positionnement logicielle, accédez à la variable "Active" (Bool) dans le chemin de DB "<nom d'axe>/Config/Position Limits_SW". L'état de la variable "Active" active ou désactive les limites de positionnement logicielles.

Vous pouvez également modifier les limites de positionnement logicielles avec votre programme utilisateur (par exemple, pour offrir une plus grande souplesse pour la configuration d'une machine ou réduire la durée de remplacement d'une machine). Votre programme utilisateur peut écrire de nouvelles valeurs dans les variables "MinPosition" et "MaxPosition" (unités physiques en format Real) dans le DB <nom d'axe>/Config/PositionLimits_SW".

10.3.6.3 Référencement

Le référencement consiste à établir une correspondance entre les coordonnées de l'axe et la position physique réelle de l'entraînement (si l'entraînement est actuellement à la position x , l'axe sera réglé pour être en position x). Pour les axes à régulation de position, les entrées et affichages pour la position se réfèrent exactement à ces coordonnées d'axe.

Remarque

La correspondance entre les coordonnées de l'axe et la situation réelle est extrêmement importante. Cette étape est nécessaire pour garantir que la position cible absolue de l'axe est également atteinte exactement avec l'entraînement.

L'instruction MC_Home déclenche le référencement de l'axe.

Il existe quatre fonctions de référencement différentes. Les deux premières fonctions permettent à l'utilisateur de définir la position en cours de l'axe et les deux autres à positionner l'axe par rapport à un capteur de référencement.

- Mode 0 - Référencement direct absolu : Lorsqu'il est activé, ce mode indique à l'axe exactement où il est. Il donne à la variable de position interne la valeur de l'entrée Position de l'instruction de référencement. Ce mode est utilisé pour le calibrage et la configuration des machines.

La position de l'axe est définie sans tenir compte de la came de référence. Les déplacements en cours ne sont pas annulés. La valeur du paramètre d'entrée Position de l'instruction MC_Home est immédiatement prise comme point de référence de l'axe. Pour affecter le point de référence à une position mécanique précise, l'axe doit être immobile à cette position pendant l'opération de référencement.

- Mode 1 - Référencement direct relatif : Lorsqu'il est activé, ce mode utilise la variable de position interne à laquelle il ajoute la valeur de l'entrée Position de l'instruction de référencement. Ce mode sert typiquement à prendre en compte le décalage d'une machine.

La position de l'axe est définie sans tenir compte de la came de référence. Les déplacements en cours ne sont pas annulés. La position de l'axe après le référencement est définie comme suit : nouvelle position de l'axe = position en cours de l'axe + valeur du paramètre Position de l'instruction MC_Home

- Mode 2 - Référencement passif : Lorsque l'axe se déplace et franchit la came de référence, la position en cours est définie comme position de référence. Cette fonction permet de prendre en compte l'usure normale des machines et le jeu entre dents et d'éviter ainsi la compensation manuelle de l'usure. L'entrée Position de l'instruction de référencement s'ajoute, comme précédemment, à la position indiquée par la came de référence, ce qui permet un décalage aisée de la position de référence.

Lors du référencement passif, l'instruction MC_Home n'exécute aucune prise de référence. L'accostage de la came de référence nécessaire pour cette étape doit être réalisé par l'utilisateur au moyen d'autres instructions Motion Control. L'axe est référencé conformément à la configuration lorsque la came de référence est détectée. Les déplacements en cours ne sont pas annulés au démarrage du référencement passif.

- **Mode 3 - Référencement actif** : Ce mode constitue la méthode de référencement de l'axe la plus précise. Le sens et la vitesse de déplacement initiaux sont configurés dans les paramètres avancés de référencement de la configuration d'objet technologique. Cela dépend de la configuration de la machine. Il est également possible de déterminer si le front montant ou le front descendant du signal de came de référence constitue la position de référence. Pratiquement tous les capteurs ont une plage active et si la position Régime permanent activé était utilisée comme signal de référence, il y aurait la possibilité d'une erreur dans la position de référence puisque la plage active du signal activé couvrirait une plage de distance. L'utilisation du front montant ou du front descendant de ce signal permet d'obtenir des résultats beaucoup plus précis pour la position de référence. Comme avec tous les autres modes, la valeur de l'entrée Position de l'instruction de référencement est ajoutée à la position référencée matérielle.

En mode de référencement actif, l'instruction MC_Home exécute la prise de référence requise. L'axe est référencé conformément à la configuration lorsque la came de référence est détectée. Les déplacements en cours sont annulés.

Les modes 0 et 1 ne nécessitent aucun déplacement de l'axe. Ils servent typiquement lors de la configuration et du calibrage. Les modes 2 et 3 nécessitent un déplacement de l'axe et franchissent un capteur qui est configuré en tant que came de référence dans l'objet technologique Axe. Le point de référence peut être placé dans la zone de travail de l'axe ou hors de la zone de travail normale mais à l'intérieur de la plage de déplacement.

Configuration des paramètres de référencement

Vous configurez les paramètres pour le référencement actif et passif dans la fenêtre de configuration "Référencement". La méthode de référencement est définie à l'aide du paramètre d'entrée "Mode" de l'instruction Motion Control. Ici, Mode = 2 signifie référencement passif et Mode = 3 signifie référencement actif.

Remarque

Prenez l'une des mesures suivantes pour garantir que la machine n'atteint pas une butée mécanique en cas d'inversion de sens :

- Maintenez une vitesse d'accostage faible.
 - Augmentez l'accélération/décélération configurée.
 - Augmentez la distance entre le fin de course matériel et la butée mécanique.
-

Tableau 10- 51 Paramètres de configuration pour le référencement de l'axe

Paramètre	Description
Entrée de la came de référence (référencement actif et passif)	<p>Sélectionnez, dans la liste déroulante, l'entrée TOR pour la came de référence. L'entrée doit pouvoir émettre des alarmes. Les entrées intégrées de la CPU et les entrées d'un Signal Board inséré peuvent être sélectionnées comme entrées pour la came de référence.</p> <p>Le temps de filtre par défaut pour les entrées TOR est de 6,4 ms. Des décélérations involontaires et donc des imprécisions peuvent se produire lorsque les entrées TOR sont utilisées comme cames de référence. Selon la vitesse de référencement et l'étendue de la came de référence, le point de référence peut ne pas être détecté. Il est possible de régler le temps de filtre sous "Filtre d'entrée" dans la configuration d'appareil des entrées TOR.</p> <p>Le temps de filtre indiqué doit être inférieur à la durée du signal d'entrée au niveau de la came de référence.</p>
Came de référence (référencement actif et passif)	<ul style="list-style-type: none"> • Référencement actif : Choisissez si l'axe doit être référencé du côté inférieur ou supérieur de la came de référence. Selon la position de départ de l'axe et la configuration des paramètres de référencement, la séquence d'accostage du point de référence peut différer du diagramme affiché dans la fenêtre de configuration. • Référencement passif : Avec le référencement passif, l'accostage aux fins de référencement doit être réalisé par l'utilisateur au moyen de commandes de mouvement. Le côté de la came de référence où l'accostage se fait dépend des facteurs suivants : <ul style="list-style-type: none"> – Configuration de la came de référence – Sens de déplacement en cours pendant le référencement passif
Inversion automatique du sens lorsque les fins de course matériels sont atteints (référencement actif uniquement)	<p>Cochez cette case si vous souhaitez utiliser les fins de course matériels comme came d'inversion pour la prise de référence. Les fins de course matériels doivent être configurés et activés pour l'inversion de sens.</p> <p>Si le fin de course matériel est atteint pendant que le référencement est actif, l'axe freine avec la décélération configurée (et non avec la décélération d'urgence) et effectue une inversion de sens. La came de référence est ensuite recherchée en sens inverse.</p> <p>Si l'inversion de sens n'est pas active et que l'axe atteint le fin de course matériel pendant que le référencement est actif, la prise de référence est interrompue avec une erreur et l'axe est freiné avec la décélération d'urgence.</p>
Sens d'accostage (référencement actif et passif)	<p>Avec la sélection de sens, vous définissez le sens d'accostage utilisé pendant le référencement actif pour rechercher la came de référence, ainsi que le sens de référencement. Le sens de référencement indique le sens de déplacement utilisé par l'axe pour accoster le côté configuré de la came de référence afin d'effectuer le référencement.</p>
Vitesse d'attaque (référencement actif uniquement)	<p>Indiquez la vitesse à laquelle la came de référence est recherchée pendant la prise de référence.</p> <p>Valeurs limites (indépendantes de l'unité utilisateur sélectionnée) : vitesse de démarrage/d'arrêt ≤ vitesse d'accostage ≤ vitesse maximale</p>

Paramètre	Description
Vitesse de référencement (référencement actif uniquement)	Indiquez la vitesse à laquelle l'axe doit accoster la came de référence pour réaliser le référencement. Valeurs limites (indépendantes de l'unité utilisateur sélectionnée) : Vitesse de démarrage/d'arrêt \leq vitesse de référencement \leq vitesse maximale
Décalage de la position de référence (référencement actif uniquement)	Si le point de référence désiré diffère de la position de la came de référence, vous pouvez indiquer le décalage de la position de référence dans ce champ. Si cette valeur est différente de 0, l'axe exécute les actions suivantes après le référencement au niveau de la came de référence : <ol style="list-style-type: none"> 1. L'axe se déplace à la vitesse de référencement pour la valeur du décalage du point de référence. 2. Lorsque la position du décalage de la position de référence est atteinte, la position de l'axe prend la valeur de la position de référence absolue. La position de référence absolue est indiquée au moyen du paramètre "Position" de l'instruction Motion Control "MC_Home". Valeurs limites (indépendantes de l'unité sélectionnée par l'utilisateur) : $-1.0e12 \leq$ décalage du point de référence $\leq 1.0e12$

Tableau 10- 52 Facteurs affectant le référencement

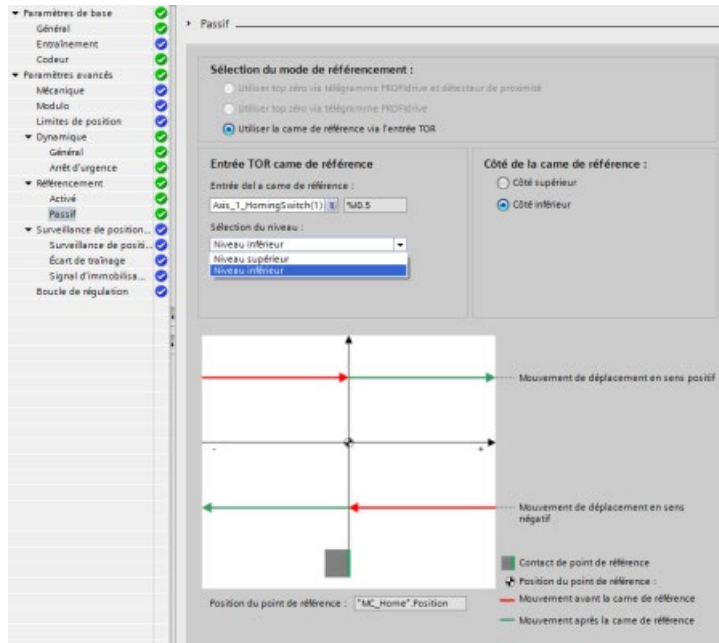
Facteurs déterminants :			Résultat :
Configuration du sens d'accostage	Configuration de la came de référence	Sens de déplacement en cours	Référencement sur la came de référence
Positif	"Côté inférieur (négatif)"	Sens positif	Bas
		Sens négatif	Haut
Positif	"Côté supérieur (positif)"	Sens positif	Haut
		Sens négatif	Bas
Négatif	"Côté inférieur (négatif)"	Sens positif	Haut
		Sens négatif	Bas
Négatif	"Côté supérieur (positif)"	Sens positif	Bas
		Sens négatif	Haut

Niveau de la came de référence

Sous Passif ou Actif dans la boîte de dialogue Référencement de la configuration d'axe, vous configurez l'entrée TOR de votre came de référence. Lors de cette configuration, vous pouvez également modifier le niveau (supérieur ou inférieur) de la came de référence des axes en boucle fermée (PROFIdrive et analogique). Le niveau supérieur est pris par défaut.

Exemple :sélection du niveau de la came de référence passive

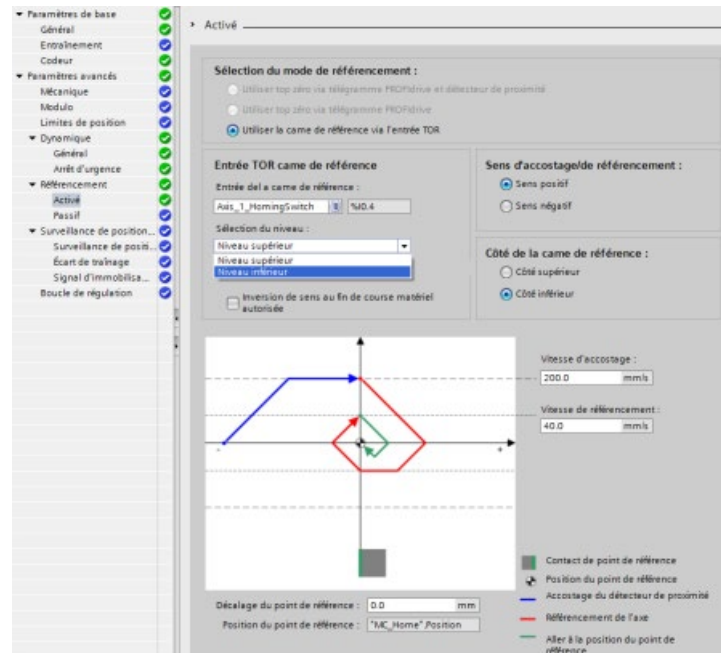
1. Vous avez configuré un projet S7-1200 avec un axe analogique/PROFdrive en référencement passif. Selon votre application, vous sélectionnez "Niveau supérieur" ou "Niveau inférieur" pour le niveau de la came de référence passive :



2. Le programme exécute un référencement passif.
3. À l'achèvement du référencement passif, l'axe est référencé.

Exemple : sélection du niveau de la came de référence active

1. Vous avez configuré un projet S7-1200 avec un axe analogique/PROFIdrive en référencement actif. Selon votre application, vous sélectionnez "Niveau supérieur" ou "Niveau inférieur" pour le niveau de la came de référence active :



2. Le programme exécute un référencement actif.
3. À l'achèvement du référencement actif, l'axe est référencé.

Séquence pour le référencement actif

Vous démarrez le référencement actif avec l'instruction Motion Control "MC_Home" (paramètre d'entrée Mode = 3). Dans ce cas, le paramètre d'entrée Position" indique les coordonnées absolues du point de référence. Vous pouvez également lancer le référencement actif dans le panneau de commande à des fins de test.

Le schéma suivant montre un exemple de courbe caractéristique pour une prise de référence active avec les paramètres de configuration suivants :

- sens d'accostage = sens d'accostage positif
- came de référence = côté supérieur (positif)
- valeur du décalage de la position de référence > 0

Tableau 10- 53 Caractéristiques de vitesse du référencement MC

Fonctionnement		Remarques	
		A	Vitesse d'attaque
		B	Vitesse de référencement
		C	Coordonnée de la position de référence
		D	Décalage de la position de référence
①	Phase de recherche (segment bleu de la courbe) : Au démarrage du référencement actif, l'axe accélère jusqu'à la vitesse d'attaque configurée et recherche la came de référence à cette vitesse.		
②	Accostage du point de référence (segment rouge de la courbe) : Une fois la came de référence détectée, l'axe freine et exécute une inversion de sens pour être référencé du côté configuré de la came de référence à la vitesse de référencement configurée.		
③	Déplacement vers la position du point de référence (segment vers de la courbe) : Après son référencement à la came de référence, l'axe se déplace vers les coordonnées du point de référence à la vitesse de référencement. Lorsqu'il atteint les coordonnées du point de référence, l'axe est arrêté à la valeur de position indiquée dans le paramètre d'entrée Position de l'instruction MC_Home.		

Remarque

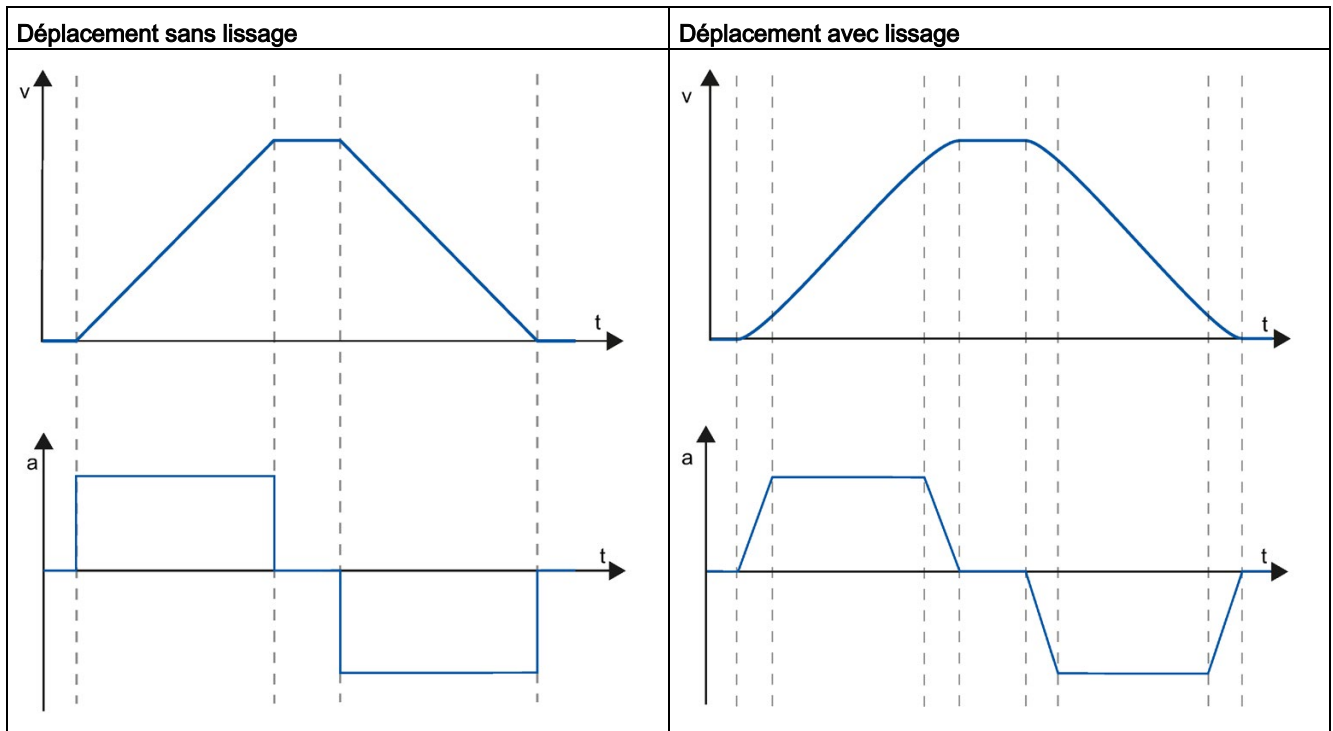
Si la recherche de référencement ne fonctionne pas comme vous le prévoyiez, vérifiez les entrées affectées aux fins de course matériels ou à la came de référence. Les alarmes sur front de ces entrées ont peut-être été désactivées dans la configuration d'appareil.

Examinez les données de configuration de l'objet technologique Axe concerné pour voir, le cas échéant, les entrées qui sont affectées à "HW Low Limit Switch Input", "HW High Limit Switch Input" et "Input reference point switch". Puis, ouvrez la configuration d'appareil pour la CPU et examinez chacune des entrées affectées. Vérifiez que les options "Activer la détection de front montant" et "Activer la détection de front descendant" sont toutes deux sélectionnées. Si elles ne le sont pas, effacez les entrées concernées dans la configuration de l'axe puis sélectionnez-les à nouveau.

10.3.6.4 Limitation d'à-coup

La limitation d'à-coup permet de réduire les contraintes auxquelles sont soumises les pièces mécaniques pendant une rampe d'accélération et de décélération. La valeur pour l'accélération et la décélération n'est pas modifiée soudainement lorsque la limitation est active ; elle est adaptée lors d'une phase de transition. La figure ci-dessous montre la courbe de vitesse et la courbe d'accélération avec et sans limitation d'à-coup.

Tableau 10- 54 Limitation d'à-coup



La limitation d'à-coup donne un profil de vitesse "lissé" au mouvement de l'axe. Cela garantit le démarrage et le freinage en douceur d'un convoyeur à bande, par exemple.

10.3.7 Instructions Motion Control

10.3.7.1 Présentation des instructions MC

Les instructions Motion Control utilisent un bloc de données technologique associé et les sorties PTO (sorties de trains d'impulsions) réservées de la CPU pour piloter le mouvement sur un axe.

- MC_Power (Page 756) active et désactive un axe Motion Control.
- MC_Reset (Page 759) remet à zéro toutes les erreurs Motion Control. Toutes les erreurs Motion Control pouvant être acquittées le sont.
- MC_Home (Page 760) établit la relation entre le programme de commande d'axe et le système de positionnement mécanique de l'axe.
- MC_Halt (Page 763) annule tous les processus de mouvement et provoque l'arrêt du mouvement de l'axe. La position d'arrêt n'est pas définie.
- MC_MoveAbsolute (Page 765) démarre le mouvement vers une position absolue. Le travail s'achève lorsque la position cible est atteinte.
- MC_MoveRelative (Page 768) démarre un mouvement de positionnement relatif par rapport à la position de départ.
- MC_MoveVelocity (Page 770) provoque le mouvement de l'axe à la vitesse indiquée.
- MC_MoveJog (Page 773) exécute le mode manuel à vue aux fins de test et de mise en route.
- MC_CommandTable (Page 776) exécute des commandes d'axe en tant que séquence de mouvement.
- MC_ChangeDynamic (Page 779) modifie les réglages dynamiques pour l'axe.
- MC_WriteParam (Page 782) écrit dans un nombre choisi de paramètres pour modifier la fonctionnalité de l'axe à partir du programme utilisateur.
- MC_ReadParam (Page 784) lit un nombre choisi de paramètres qui indiquent la position en cours, la vitesse, etc., de l'axe défini dans l'entrée Axis.

Niveaux de firmware de la CPU

Si vous avez une CPU S7-1200 avec un firmware de version V4.1 ou postérieure, sélectionnez la version V5.0 de chaque instruction Motion Control.

Si vous avez une CPU S7-1200 avec un firmware V4.0 ou antérieur, sélectionnez la version V4.0, V3.0, V2.0 ou V1.0 appropriée de chaque instruction de mouvement.

Remarque

Les instructions Motion Control V1.0 à V3.0 commandent activement la sortie de validation (ENO) de l'instruction. Lorsqu'une erreur se produit dans le bloc, la sortie de validation est mise à 0. Une erreur est signalée par les sorties ERROR, ErrorID et ErrorInfo dans le bloc. Il est possible d'évaluer l'état de l'instruction à l'aide de la sortie de validation et d'exécuter ensuite d'autres instructions en série.

Avec les instructions Motion Control V4.0 et V5.0, la sortie de validation reste vraie tant que l'instruction s'exécute, et ce quel que soit l'état d'erreur. Ainsi, un programme qui utilisait V3.0 ou une version antérieure de Motion Control et qui se basait sur l'état de la sortie de validation peut fonctionner incorrectement. Pour remédier à cela, vous devez utiliser les sorties DONE et ERROR, et non la sortie de validation, pour évaluer l'état de l'instruction avec Motion Control V4.0 ou plus.

Remarque


La CPU calcule les tâches de mouvement en "tranches" ou segments de 10 ms. Lorsqu'une tranche est en cours d'exécution, la prochaine attend son exécution dans la file d'attente. Si vous interrompez la tâche de mouvement sur un axe (en exécutant une autre nouvelle tâche de mouvement pour cet axe), la nouvelle tâche de mouvement ne peut pas être exécutée durant un maximum de 20 ms (le reste de la tranche actuelle plus la tranche en file d'attente).

10.3.7.2 MC_Power (Libérer/bloquer l'axe)

Remarque

Si l'axe est désactivé à la suite d'une erreur, il sera automatiquement réactivé une fois l'erreur éliminée et acquittée. Cela nécessite que le paramètre d'entrée Enable ait conservé la valeur VRAI pendant ce processus.

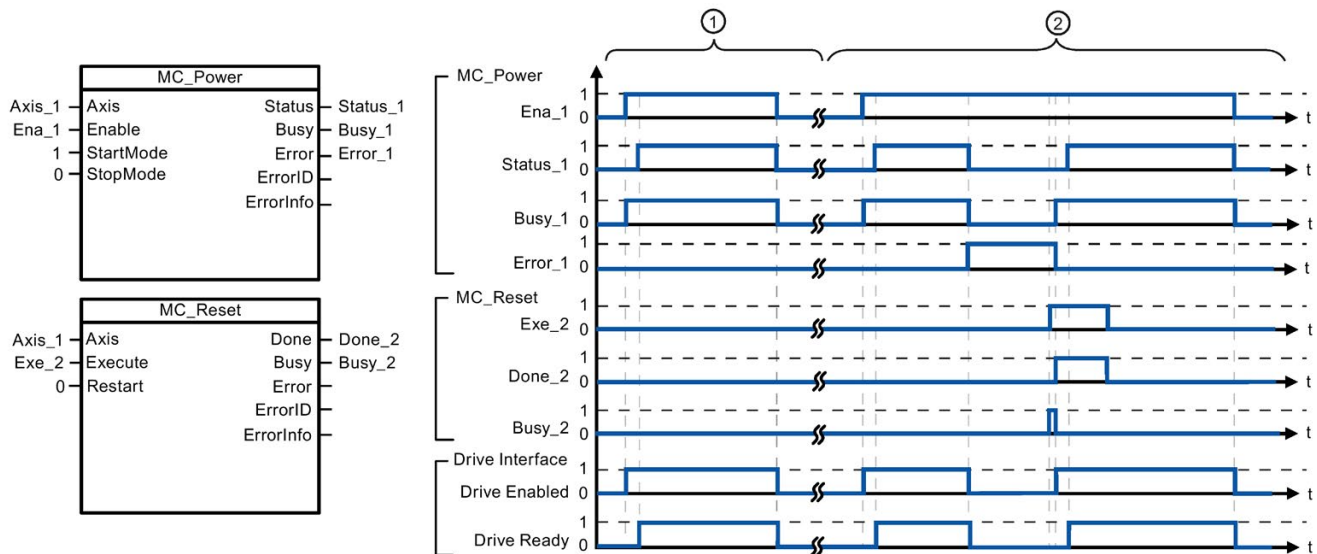
Tableau 10- 55 Instruction MC_Power

CONT/LOG	SCL	Description
	<pre>"MC_Power_DB" (Axis:= _multi_fb_in_, Enable:= _bool_in_, StartMode:= _int_in_, StopMode:= _int_in_, Status=> _bool_out_, Busy=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorIn- fo=> _word_out_) ;</pre>	<p>L'instruction Motion Control MC_Power débloque ou désactive un axe. Les conditions suivantes doivent être remplies pour que vous puissiez libérer ou bloquer l'axe :</p> <ul style="list-style-type: none"> • L'objet technologique a été configuré correctement. • Il n'y a pas d'erreur de libération/blocage en attente. <p>L'exécution de MC_Power ne peut pas être annulée par une tâche Motion Control. Le blocage de l'axe (paramètre d'entrée Enable = FAUX) annule toutes les tâches Motion Control pour l'objet technologique associé.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_Power_DB" est le nom du DB d'instance.

Tableau 10- 56 Paramètres pour l'instruction MC_Power

Paramètre et type		Type de données	Description
Axis	IN_OUT	TO_Axis	Objet technologique Axe
Enable	IN	Bool	<ul style="list-style-type: none"> FALSE (valeur par défaut) : Toutes les tâches actives sont annulées conformément au mode "StopMode" paramétré et l'axe s'arrête. TRUE : Motion Control tente de débloquent l'axe.
StartMode	IN	Int	<ul style="list-style-type: none"> 0 : Régulation de vitesse Remarque : Le paramètre StartMode est évalué uniquement pendant la détection de front (de 0 à 1). 1 : Régulation de position (valeur par défaut)
StopMode	IN	Int	<ul style="list-style-type: none"> 0 : Arrêt d'urgence : Si une demande d'inhibition de l'axe est en cours, l'axe exécute la décélération d'urgence configurée. L'axe est bloqué dès qu'il est immobilisé. 1 : Arrêt immédiat : Si une demande d'inhibition de l'axe est en cours, l'axe est arrêté sans décélération. La sortie d'impulsions est arrêtée immédiatement. 2 : Arrêt d'urgence avec contrôle de jerk : Si une demande d'inhibition de l'axe est en cours, l'axe exécute la décélération d'urgence configurée. Si la commande d'à-coup est activée, l'à-coup configuré est pris en compte. L'axe est bloqué dès qu'il est immobilisé.
Status	OUT	Bool	<p>Etat de libération de l'axe :</p> <ul style="list-style-type: none"> FALSE : L'axe est bloqué : <ul style="list-style-type: none"> L'axe n'exécute pas de tâches Motion Control et n'accepte pas de nouvelles tâches (à l'exception d'une tâche MC_Reset). L'axe n'est pas référencé. Lors du blocage, l'état ne passe pas à FAUX tant que l'axe n'est pas immobilisé. TRUE : L'axe est débloquent : <ul style="list-style-type: none"> L'axe est prêt à exécuter des tâches Motion Control. Lors de la libération de l'axe, l'état ne passe à VRAI que lorsque le signal "Entraînement prêt" est présent. Si l'interface d'entraînement "Entraînement prêt" n'a pas été configurée dans la configuration de l'axe, l'état passe à VRAI immédiatement.
Busy	OUT	Bool	<p>FALSE : MC_Power n'est pas active. TRUE : MC_Power est active.</p>
Error	OUT	Bool	<p>FALSE : Pas d'erreur TRUE : Une erreur s'est produite dans l'instruction Motion Control "MC_Power" ou dans l'objet technologique associé. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".</p>
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error"
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID"



- ① Un axe est validé puis à nouveau bloqué. Une fois que l'entraînement a renvoyé "Entraînement prêt" à la CPU, la réussite de la libération peut être lue via "Status_1".
- ② Après une libération d'axe, une erreur ayant entraîné Le blocage de l'axe s'est produite. L'erreur est supprimée et acquittée avec "MC_Reset". L'axe est alors à nouveau libéré.

Procédez comme suit pour libérer un axe à interface d'entraînement configurée :

1. Vérifiez que les conditions décrites plus haut sont bien remplies.
2. Initialisez le paramètre d'entrée "StopMode" à la valeur désirée. Définissez le paramètre d'entrée "Enable" à VRAI.

La sortie de libération pour "Entraînement libéré" passe à VRAI pour laisser passer le courant vers l'entraînement. La CPU attend le signal "Entraînement prêt" de l'entraînement.

Lorsque le signal "Entraînement prêt" est disponible sur l'entrée Prêt configurée de la CPU, l'axe est libéré. Le paramètre de sortie "Status" et la variable d'objet technologique <nom axe>.StatusBits.Enable prennent la valeur VRAI.

Procédez comme suit pour libérer un axe sans interface d'entraînement configurée :

1. Vérifiez que les conditions décrites plus haut sont bien remplies.
2. Initialisez le paramètre d'entrée "StopMode" à la valeur désirée. Définissez le paramètre d'entrée "Enable" à VRAI. L'axe est libéré. Le paramètre de sortie "Status" et la variable d'objet technologique <nom d'axe>.StatusBits.Enable prennent la valeur VRAI.

Procédez comme suit pour bloquer un axe :


1. Immobilisez l'axe.

Vous pouvez savoir quand l'axe est immobile à l'aide de la variable d'objet technologique <nom d'axe>.StatusBits.StandStill.

2. Définissez le paramètre d'entrée "Enable" à FAUX une fois l'axe immobilisé.
3. Les paramètres de sortie "Busy" et "Status" et la variable d'objet technologique <nom d'axe>.StatusBits.Enable prennent la valeur FAUX lorsque le blocage de l'axe est achevé.

10.3.7.3 MC_Reset (Acquitter l'erreur)

Tableau 10- 57 Instruction MC_Reset

CONT/LOG	SCL	Description
	<pre>"MC_Reset_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Restart:= _bool_in_, Done=> _bool_out_, Busy=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_) ;</pre>	<p>Utilisez l'instruction MC_Reset pour acquitter les erreurs "Erreur de fonctionnement avec arrêt de l'axe" et "Erreur de configuration". Vous trouverez les erreurs nécessitant un acquittement dans la liste des ErrorIDs et ErrorInfos, sous "Solution".</p> <p>Avant d'utiliser l'instruction MC_Reset, vous devez avoir éliminé la cause de l'erreur de configuration en attente nécessitant un acquittement (par exemple, en changeant une valeur d'accélération invalide dans l'objet technologique Axe en une valeur valide).</p> <p>A partir de V3.0, le paramètre Restart permet de charger la configuration d'axe en mémoire de travail à l'état de fonctionnement MARCHÉ.</p>

- STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- Dans l'exemple SCL, "MC_Reset_DB" est le nom du DB d'instance.

La tâche MC_Reset ne peut être annulée par aucune autre tâche Motion Control. La nouvelle tâche MC_Reset n'annule aucune autre tâche Motion Control active.

Tableau 10- 58 Paramètres pour l'instruction MC_Reset

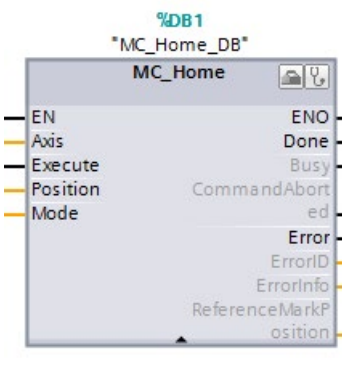
Paramètre et type	Type de données	Description	
Axis	IN	TO_Axis_1	Objet technologique Axe
Execute	IN	Bool	Démarrage de la tâche en présence d'un front montant
Restart	IN	Bool	TRUE = Chargement de la configuration d'axe de la mémoire de chargement dans la mémoire de travail. La commande ne peut être exécutée que lorsque l'axe est bloqué.
			FALSE = Acquittement des erreurs en attente
Done	OUT	Bool	VRAI = L'erreur a été acquittée.
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error"
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID"

Procédez comme suit pour acquitter une erreur avec MC_Reset :

- Vérifiez que les conditions décrites plus haut sont bien remplies.
- Lancez l'acquittement de l'erreur en présence d'un front montant dans le paramètre d'entrée Execute.
- On sait que l'erreur a été acquittée lorsque Done est égal à VRAI et que la variable d'objet technologique <nom d'axe>.StatusBits.Error est égal à FAUX.

10.3.7.4 Instruction MC_Home (Référencer l'axe)

Tableau 10- 59 Instruction MC_Home

CONT/LOG	SCL	Description
	<pre>"MC_Home_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_real_in_, Mode:=_int_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_, ReferenceMarkPosition=>_real_out_);</pre>	<p>Utilisez l'instruction MC_Home pour établir une correspondance entre les coordonnées de l'axe et la position physique réelle de l'entraînement. Le référencement est nécessaire pour un positionnement absolu de l'axe.</p> <p>L'utilisation de l'instruction MC_Home n'est possible que si l'axe est libéré</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

2 Dans l'exemple SCL, "MC_Home_DB" est le nom du DB d'instance.

Les modes de référencement suivants sont disponibles :

- Référencement direct absolu (Mode = 0) : La valeur du paramètre "Position" est prise comme position en cours de l'axe.
- Référencement direct relatif (Mode = 1) : La position en cours de l'axe est décalée de la valeur du paramètre "Position".
- Référencement passif (Mode = 2) : Lors du référencement passif, l'instruction MC_Home n'exécute aucune prise de référence. L'accostage de la came de référence nécessaire pour cette étape doit être réalisé par l'utilisateur au moyen d'autres instructions Motion Control. L'axe est référencé lorsque la came de référence est détectée.
- Référencement actif (Mode = 3) : La prise de référence est exécutée automatiquement.
- Référencement du codeur absolu (relatif) (Mode = 6) : La position en cours est décalée de la valeur du paramètre "MC_Home.Position".
- Référencement du codeur absolu (absolu) (Mode = 7) : La position en cours est réglée à la valeur du paramètre "MC_Home.Position".

Tableau 10- 60 Paramètres pour l'instruction MC_Home

Paramètre et type		Type de données	Description
Axis	IN_OUT	TO_Axis	Objet technologique Axe
Execute	IN	Bool	Démarrage de la tâche en présence d'un front montant
Position	IN	Real	<ul style="list-style-type: none"> Mode = 0, 2 et 3 (position absolue de l'axe à l'achèvement du référencement) Mode = 1 (valeur de correction pour la position en cours de l'axe) Mode = 6 (la position en cours est décalée de la valeur du paramètre "MC_Home.Position") Mode = 7 (la position en cours est réglée à la valeur du paramètre "MC_Home.Position") Valeurs limites : $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Mode	IN	Int	Mode de référencement : <ul style="list-style-type: none"> 0 : Référencement direct absolu La nouvelle position de l'axe correspond à la valeur de position du paramètre "Position". 1 : Référencement direct relatif La nouvelle position de l'axe correspond à la position en cours de l'axe + la valeur de position du paramètre "Position". 2 : Référencement passif Référencement selon la configuration de l'axe. Après le référencement, la valeur du paramètre "Position" est prise comme nouvelle position de l'axe. 3 : Référencement actif Prise de référence selon la configuration de l'axe. Après le référencement, la valeur du paramètre "Position" est prise comme nouvelle position de l'axe. 6 : La position en cours est décalée de la valeur du paramètre "MC_Home.Position". Le décalage de valeur absolue calculé est enregistré dans la CPU de manière rémanente. (<nom de l'axe>.StatusSensor.AbsEncoderOffset) 7 : La position en cours est réglée à la valeur du paramètre "MC_Home.Position". Le décalage de valeur absolue calculé est enregistré dans la CPU de manière rémanente. (<nom de l'axe>.StatusSensor.AbsEncoderOffset)
Done	OUT	Bool	VRAI = Tâche achevée
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = La tâche a été annulée par une autre tâche lors de l'exécution.

Paramètre et type		Type de données	Description
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error"
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID"
ReferenceMarkPosition	OUT	Real	Position de l'axe au repère de référence dans le système de coordonnées précédent.

Paramètre de sortie "ReferenceMarkPosition" : la fonction de référencement enregistre l'ancienne position avant de s'exécuter et fournit cette valeur dans le paramètre de sortie ReferenceMarkPosition. La sortie fournit les valeurs suivantes pour les différents modes de référencement :

- Référencement actif/passif : Elle fournit la position de l'axe au top zéro/repère de référence du système de coordonnées précédent pendant et après le référencement actif/passif.
- Référencement du codeur absolu : Elle fournit la position précédente de l'axe pendant et après le référencement du codeur absolu.
- Référencement direct : Elle fournit la position précédente de l'axe pendant et après le référencement direct.

Pendant le référencement, Motion Control définit la position de l'axe à la nouvelle valeur de l'entrée "MC_Home.Position". Les valeurs de "MC_Home.ReferenceMarkPosition" sont valides lorsque "MC_Home.Done" a la valeur TRUE.

Remarque

Perte du référencement de l'axe dans les cas suivants

- Blocage de l'axe par l'instruction MC_Power
 - Commutation entre mode automatique et mode manuel
 - Lors du démarrage d'un référencement actif (le référencement de l'axe est à nouveau disponible à l'achèvement de l'opération de référencement)
 - Après mise hors tension puis sous tension de la CPU
 - Après un redémarrage de la CPU (transition MARCHÉ à ARRET ou ARRET à MARCHÉ)
-

Procédez comme suit pour référencer l'axe :

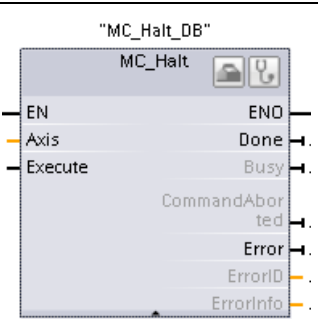
1. Vérifiez que les conditions décrites plus haut sont bien remplies.
2. Initialisez les paramètres d'entrée nécessaires et lancez l'opération de référencement à l'aide d'un front montant dans le paramètre d'entrée "Execute".
3. On sait que le référencement est achevé lorsque le paramètre de sortie "Done" et la variable d'objet technologique <nom d'axe>.StatusBits.HomingDone prennent la valeur VRAI.

Tableau 10- 61 Comportement d'annulation

Mode	Description		
0 ou 1	La tâche MC_Home ne peut être annulée par aucune autre tâche Motion Control. La nouvelle tâche MC_Home n'annule aucune tâche Motion Control active. Les tâches de déplacement par rapport à une position reprennent après le référencement en fonction de la nouvelle position de référence (valeur dans le paramètre d'entrée Position).		
2	La tâche MC_Home peut être annulée par les tâches Motion Control suivantes : Tâche MC_Home Mode = 2, 3 : La nouvelle tâche MC_Home annule la tâche Motion Control active suivante. Tâche MC_Home Mode = 2 : Les tâches de déplacement par rapport à une position reprennent après le référencement en fonction de la nouvelle position de référence (valeur dans le paramètre d'entrée Position).		
3	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> La tâche MC_Home peut être annulée par les tâches Motion Control suivantes : <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog </td> <td style="width: 50%; vertical-align: top;"> La nouvelle tâche MC_Home annule les tâches Motion Control actives suivantes : <ul style="list-style-type: none"> • MC_Home Mode = 2, 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog </td> </tr> </table>	La tâche MC_Home peut être annulée par les tâches Motion Control suivantes : <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 	La nouvelle tâche MC_Home annule les tâches Motion Control actives suivantes : <ul style="list-style-type: none"> • MC_Home Mode = 2, 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog
La tâche MC_Home peut être annulée par les tâches Motion Control suivantes : <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 	La nouvelle tâche MC_Home annule les tâches Motion Control actives suivantes : <ul style="list-style-type: none"> • MC_Home Mode = 2, 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 		

10.3.7.5 MC_Halt (Arrêter un axe)

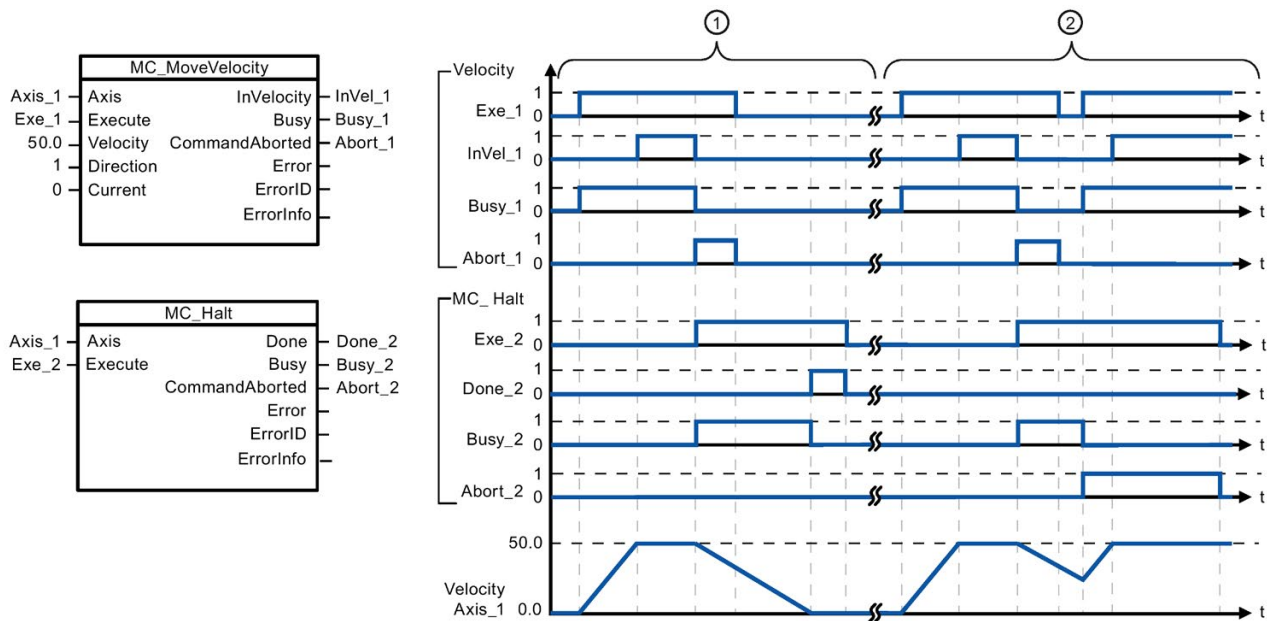
Tableau 10- 62 Instruction MC_Halt

CONT/LOG	SCL	Description
	<pre>"MC_Halt_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Utilisez l'instruction MC_Halt pour arrêter tout mouvement et immobiliser l'axe. La position d'immobilisation n'est pas définie. L'utilisation de l'instruction MC_Halt n'est possible que si l'axe est libéré.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_Halt_DB" est le nom du DB d'instance.

Tableau 10- 63 Paramètres pour l'instruction MC_Halt

Paramètre et type		Type de données	Description
Axis	IN	TO_Axis_1	Objet technologique Axe
Execute	IN	Bool	Démarrage de la tâche en présence d'un front montant
Done	OUT	Bool	VRAI = La vitesse zéro est atteinte.
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error"
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID"



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 5,0

- ① L'axe est freiné jusqu'à l'immobilisation par une tâche MC_Halt. L'immobilisation de l'axe est signalée via "Done_2".
- ② Alors qu'une tâche MC_Halt freine l'axe, cette tâche est annulée par une autre tâche de mouvement. L'annulation est signalée via "Abort_2".

Comportement d'annulation

La tâche MC_Halt peut être annulée par les tâches Motion Control suivantes :

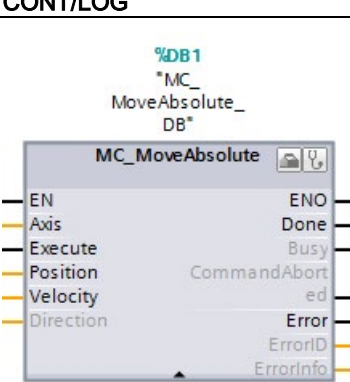
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

La nouvelle tâche MC_Halt annule les tâches Motion Control actives suivantes :

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.6 MC_MoveAbsolute (Positionner un axe de manière absolue)

Tableau 10- 64 Instruction MC_MoveAbsolute

CONT/LOG	SCL	Description
	<pre>"MC_MoveAbsolute_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_real_in_, Velocity:=_real_in_, Direction:=_int_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Utilisez l'instruction MC_MoveAbsolute pour démarrer un déplacement de positionnement de l'axe à une position absolue.</p> <p>L'utilisation de l'instruction MC_MoveAbsolute n'est possible que si l'axe est libéré et référencé.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_MoveAbsolute_DB" est le nom du DB d'instance.

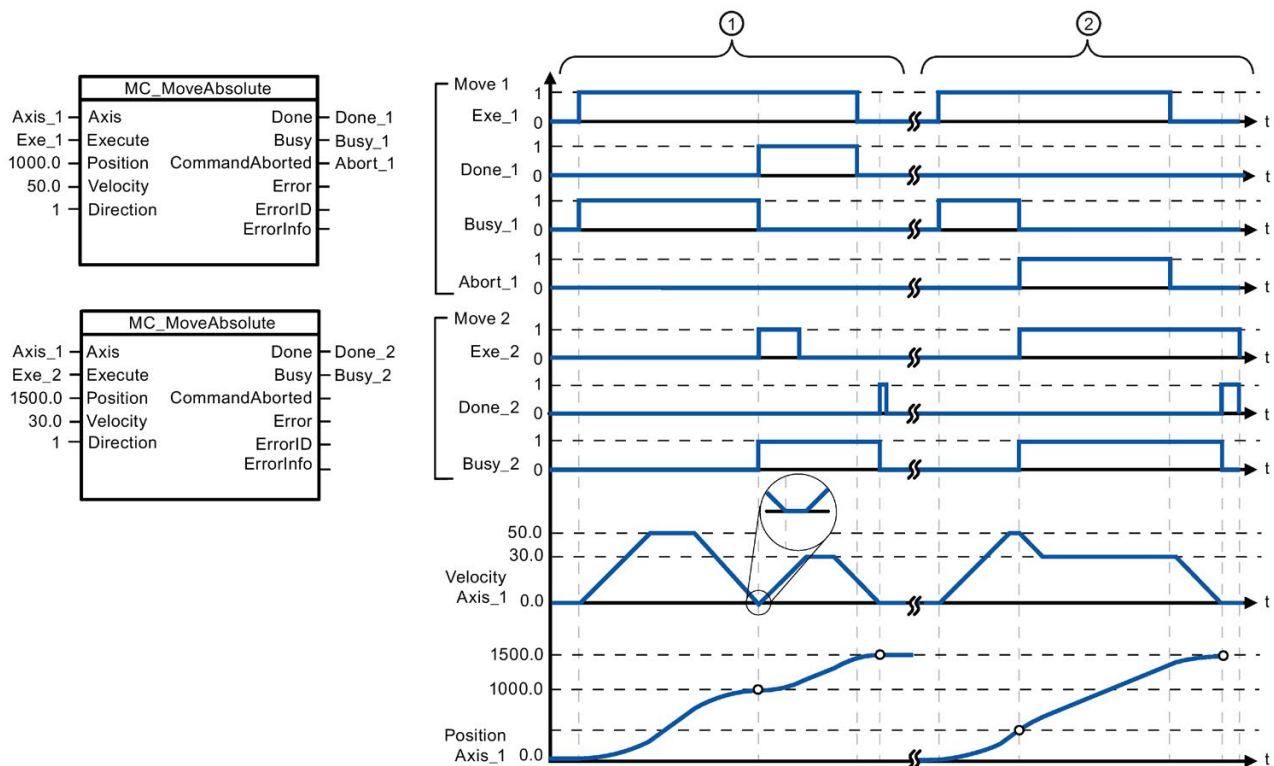
Tableau 10- 65 Paramètres pour l'instruction MC_MoveAbsolute

Paramètre et type		Type de données	Description
Axis	IN	TO_Axis_1	Objet technologique Axe
Execute	IN	Bool	Démarrage de la tâche en présence d'un front montant (valeur par défaut : faux)
Position	IN	Real	Position cible absolue (valeur par défaut : 0,0) 0,0) Valeurs limites : $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Velocity	IN	Real	Vitesse de l'axe (valeur par défaut : 10,0) Cette vitesse n'est pas toujours atteinte en raison de l'accélération et de la décélération configurées ainsi que de la position cible à accoster. Valeurs limites : vitesse de démarrage/d'arrêt \leq Velocity \leq vitesse maximale
Direction	IN	Int	Sens de rotation (valeur par défaut : 0)
Done	OUT	Bool	VRAI = La position cible absolue est atteinte.
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error" (valeur par défaut : 0000) 0000)
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID" (valeur par défaut : 0000) 0000)

Vous pouvez configurer l'axe de positionnement en tant qu'axe modulo. Lorsque vous utilisez un axe modulo, vous pouvez sélectionner le sens de déplacement avec le paramètre d'entrée "Direction". Motion Control ne tient pas compte de l'entrée "Sens" si l'axe n'est pas un axe modulo.

Le tableau suivant indique les valeurs valides pour l'entrée "Direction" :

Valeur	Signification	Commentaire
0	SIGN_OF_VELOCITY	Le signe de la vitesse définit le sens de déplacement.
1	POSITIVE	Déplacement à vitesse positive
2	NEGATIVE	Déplacement à vitesse négative
3	SHORTEST_WAY	Déplacement par le chemin le plus court jusqu'à la destination.



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 10,0

- ① Un axe est déplacé à la position absolue 1000,0 avec une tâche MC_MoveAbsolute. Lorsque l'axe atteint la position cible, cela est signalé via "Done_1". Lorsque "Done_1" = VRAI, une autre tâche MC_MoveAbsolute, avec une position cible 1500,0, est lancée. Du fait des temps de réponse (par exemple, temps de cycle du programme utilisateur, etc.), l'axe s'immobilise brièvement (voir le détail agrandi). Lorsque l'axe atteint la nouvelle position cible, cela est signalé via "Done_2".
- ② Une tâche MC_MoveAbsolute active est annulée par une autre tâche MC_MoveAbsolute. L'annulation est signalée via "Abort_1". L'axe est ensuite déplacé à la nouvelle vitesse vers la nouvelle position cible 1500,0. Lorsque la nouvelle position cible est atteinte, cela est signalé via "Done_2".

Comportement d'annulation

La tâche MC_MoveAbsolute peut être annulée par les tâches Motion Control suivantes :

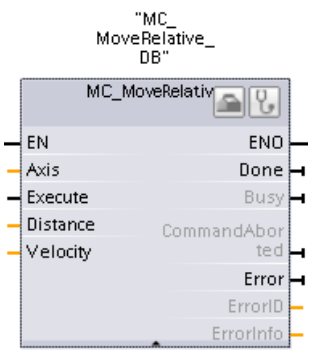
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

La nouvelle tâche MC_MoveAbsolute annule les tâches Motion Control actives suivantes :

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.7 MC_MoveRelative (Positionner un axe de manière relative)

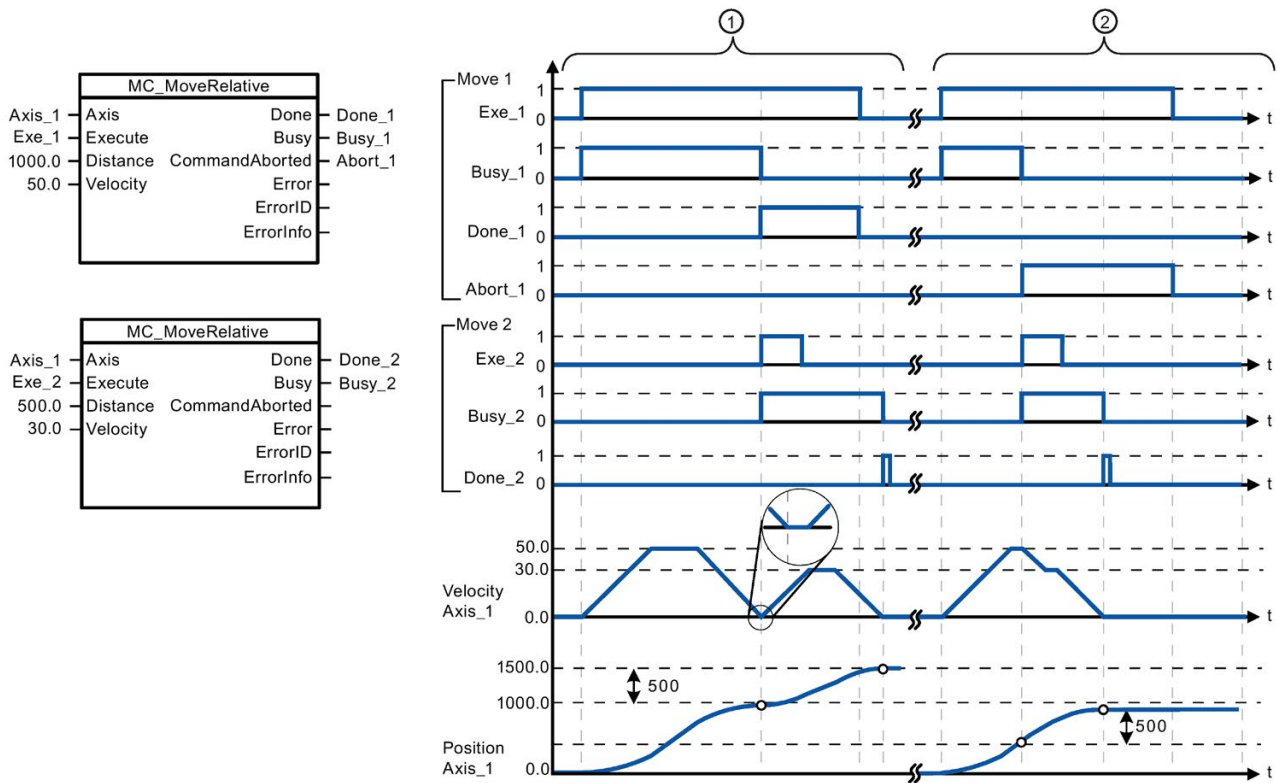
Tableau 10- 66 Instruction MC_MoveRelative

CONT/LOG	SCL	Description
	<pre>"MC_MoveRelative_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Distance:= _real_in_, Velocity:= _real_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>Utilisez l'instruction MC_MoveRelative pour démarrer un déplacement de positionnement relatif de l'axe par rapport à une position initiale..</p> <p>L'utilisation de l'instruction MC_MoveRelative n'est possible que si l'axe est libéré.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_MoveRelative_DB" est le nom du DB d'instance.

Tableau 10- 67 Paramètres pour l'instruction MC_MoveRelative

Paramètre et type	Type de données	Description
Axis	IN	TO_Axis_1 Objet technologique Axe
Execute	IN	Bool Démarrage de la tâche en présence d'un front montant (valeur par défaut : faux)
Distance	IN	Real Distance à parcourir pour l'opération de positionnement (valeur par défaut : 0,0) 0,0) Valeurs limites : $-1.0e^{12} \leq \text{Distance} \leq 1.0e^{12}$
Velocity	IN	Real Vitesse de l'axe (valeur par défaut : 10,0) Cette vitesse n'est pas toujours atteinte en raison de l'accélération et de la décélération configurées ainsi que de la distance à parcourir. Valeurs limites : vitesse de démarrage/d'arrêt $\leq \text{Velocity} \leq$ vitesse maximale
Done	OUT	Bool VRAI = La position cible est atteinte.
Busy	OUT	Bool VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word Code d'erreur pour le paramètre "Error" (valeur par défaut : 0000) 0000)
ErrorInfo	OUT	Word Information d'erreur pour le paramètre "ErrorID" (valeur par défaut : 0000) 0000)



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 10,0

- ① L'axe est déplacé par une tâche MC_MoveRelative de la distance ("Distance") 1000,0. Lorsque l'axe atteint la position cible, cela est signalé via "Done_1". Lorsque "Done_1" = VRAI, une autre tâche MC_MoveRelative, avec une distance de déplacement de 500,0, est lancée. Du fait des temps de réponse (par exemple, temps de cycle du programme utilisateur), l'axe s'immobilise brièvement (voir le détail agrandi). Lorsque l'axe atteint la nouvelle position cible, cela est signalé via "Done_2".
- ② Une tâche MC_MoveRelative active est annulée par une autre tâche MC_MoveRelative. L'annulation est signalée via "Abort_1". L'axe est ensuite déplacé de la nouvelle distance ("Distance") 500,0 à la nouvelle vitesse. Lorsque la nouvelle position cible est atteinte, cela est signalé via "Done_2".

Comportement d'annulation

La tâche MC_MoveRelative peut être annulée par les tâches Motion Control suivantes :

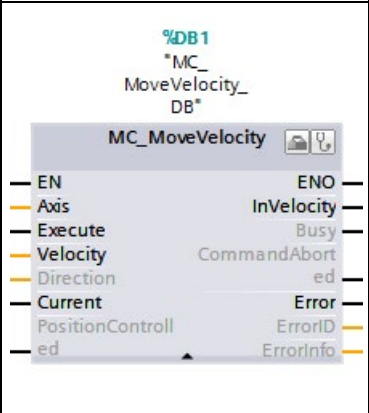
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

La nouvelle tâche MC_MoveRelative annule les tâches Motion Control actives suivantes :

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.8 MC_MoveVelocity (Déplacer un axe à une vitesse prédéfinie)

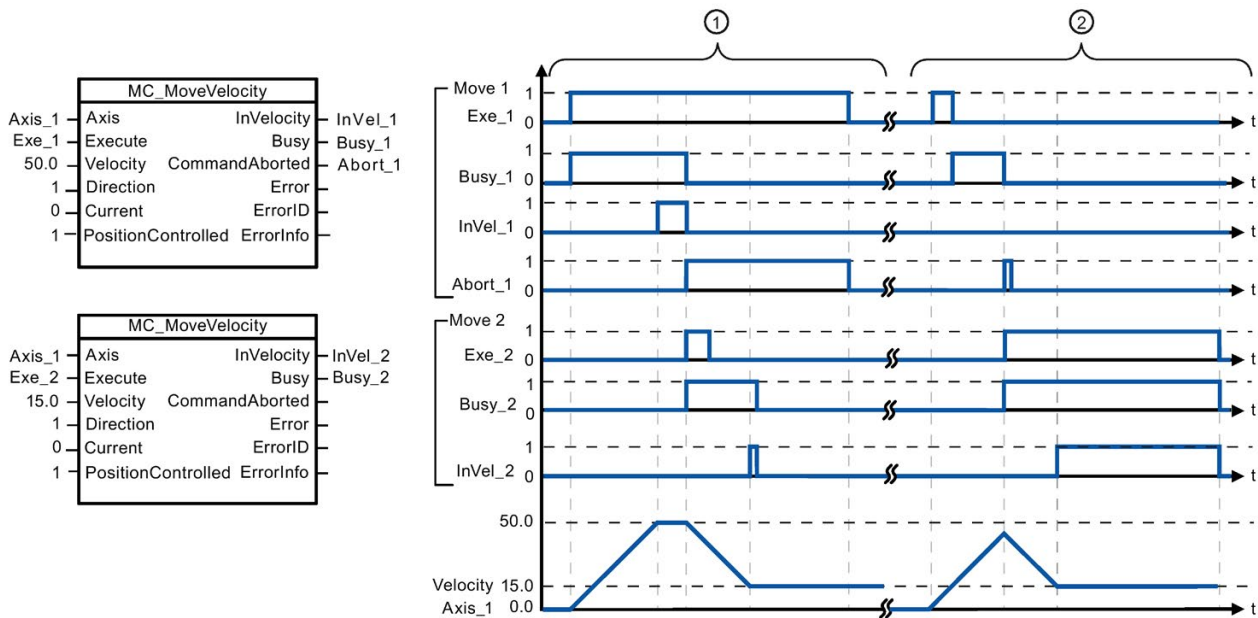
Tableau 10- 68 Instruction MC_MoveVelocity

CONT/LOG	SCL	Description
	<pre> "MC_MoveVelocity_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Velocity:=_real_in_, Direction:=_int_in_, Current:=_bool_in_, PositionControlled:=_bool_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_); </pre>	<p>Utilisez l'instruction MC_MoveVelocity pour déplacer l'axe à la vitesse prédéfinie.</p> <p>L'utilisation de l'instruction MC_MoveVelocity n'est possible que si l'axe est libéré.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_MoveVelocity_DB" est le nom du DB d'instance.

Tableau 10- 69 Paramètres pour l'instruction MC_MoveVelocity

Paramètre et type		Type de données	Description
Axis	IN	TO_SpeedAxis	Objet technologique Axe
Execute	IN	Bool	Démarrage de la tâche en présence d'un front montant (valeur par défaut : faux)
Velocity	IN	Real	Indication de vitesse pour le déplacement de l'axe (valeur par défaut : 100.0) Valeurs limites : vitesse de démarrage/d'arrêt $\leq \text{Velocity} \leq$ vitesse maximale (Velocity = 0.0 est autorisé)
Direction	IN	Int	Indication du sens : <ul style="list-style-type: none"> • 0 : 0 : Le sens de rotation correspond au signe de la valeur dans le paramètre "Velocity" (valeur par défaut). • 1 : 1 : Sens de rotation positif (le signe de la valeur dans le paramètre "Velocity" n'est pas pris en compte) • 2 : Sens de rotation négatif (le signe de la valeur dans le paramètre "Velocity" n'est pas pris en compte)
Current	IN	Bool	Conserver la vitesse actuelle : <ul style="list-style-type: none"> • FAUX : "Conserver la vitesse actuelle" est désactivé. Les valeurs des paramètres "Velocity" et "Direction" sont utilisées. (valeur par défaut) • VRAI : "Conserver la vitesse actuelle" est activé. Les valeurs des paramètres "Velocity" et "Direction" ne sont pas prises en compte. Lorsque l'axe reprend le déplacement à la vitesse en cours, le paramètre "InVelocity" renvoie la valeur TRUE.
PositionControlled	IN	Bool	<ul style="list-style-type: none"> • 0 : régulation de vitesse • 1 : régulation de position (valeur par défaut : vrai)
InVelocity	OUT	Bool	VRAI : <ul style="list-style-type: none"> • Si "Current" = FAUX (FALSE) : La vitesse indiquée dans le paramètre "Velocity" a été atteinte. • Si "Current" = VRAI (TRUE) : L'axe se déplace à la vitesse en cours au démarrage.
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error" (valeur par défaut : 0000) 0000)
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID" (valeur par défaut : 0000) 0000)



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 10,0

- ① Une tâche MC_MoveVelocity active utilise "InVel_1" pour signaler que sa vitesse cible a été atteinte. Elle est ensuite annulée par une autre tâche MC_MoveVelocity. L'annulation est signalée via "Abort_1". Lorsque la nouvelle vitesse cible de 15,0 est atteinte, cela est signalé via "InVel_2". L'axe poursuit alors son déplacement à la nouvelle vitesse constante.
- ② Une tâche active MC_MoveVelocity est annulée par une autre tâche MC_MoveVelocity avant d'atteindre sa vitesse cible. L'annulation est signalée via "Abort_1". Lorsque la nouvelle vitesse cible de 15,0 est atteinte, cela est signalé via "InVel_2". L'axe poursuit alors son déplacement à la nouvelle vitesse constante.

Comportement d'annulation

La tâche MC_MoveVelocity peut être annulée par les tâches Motion Control suivantes :

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

La nouvelle tâche MC_MoveVelocity annule les tâches Motion Control actives suivantes :

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Remarque**Comportement en cas de vitesse définie à 0 (Velocity = 0,0)**

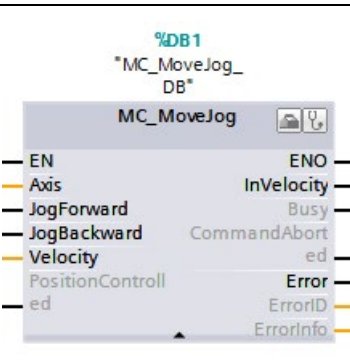
Une tâche MC_MoveVelocity avec "Velocity" = 0,0 (par exemple, une tâche MC_Halt) annule les tâches de déplacement actives et arrête l'axe avec la décélération configurée. Lorsque l'axe s'immobilise, le paramètre de sortie "InVelocity" indique VRAI pendant au moins un cycle du programme.

"Busy" indique la valeur VRAI pendant l'opération de décélération et passe à FAUX conjointement avec "InVelocity". Si le paramètre "Execute" est VRAI, "InVelocity" et "Busy" sont verrouillés.

Lorsque la tâche MC_MoveVelocity est démarrée, le bit d'état "SpeedCommand" est mis à 1 dans l'objet technologique. Le bit d'état "ConstantVelocity" est mis à 1 lors de l'immobilisation de l'axe. Les deux bits sont adaptés à la nouvelle situation lorsqu'une nouvelle tâche de déplacement est démarrée.

10.3.7.9 MC_MoveJog (Déplacer un axe en mode Manuel à vue)

Tableau 10- 70 Instruction MC_MoveJog

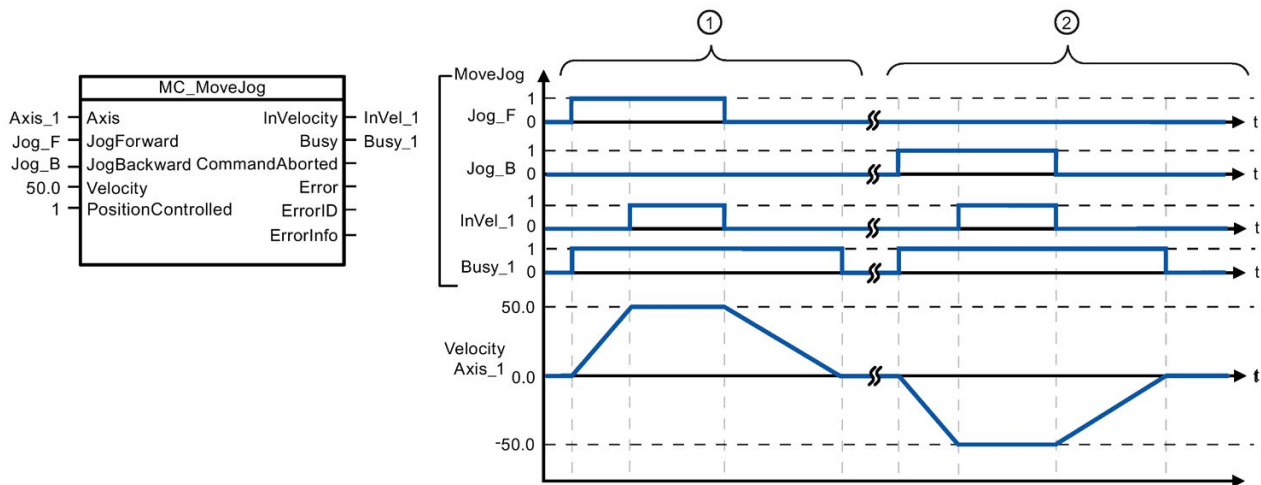
CONT/LOG	SCL	Description
	<pre>"MC_MoveJog_DB" (Axis:=_multi_fb_in_, JogForward:=_bool_in_, JogBackward:=_bool_in_, Velocity:=_real_in_, PositionControlled:=_bool_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Utilisez l'instruction MC_MoveJog pour déplacer l'axe à la vitesse prédéfinie en mode Marche par à-coups. Cette instruction sert typiquement au test et à la mise en service.</p> <p>L'utilisation de l'instruction MC_MoveJog n'est possible que si l'axe est libéré.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_MoveJog_DB" est le nom du DB d'instance.

Tableau 10- 71 Paramètres pour l'instruction MC_MoveJog

Paramètre et type		Type de données	Description
Axis	IN	TO_SpeedAxis	Objet technologique Axe
JogForward ¹	IN	Bool	Tant que le paramètre est VRAI, l'axe se déplace dans le sens positif à la vitesse indiquée dans le paramètre "Velocity". Le signe de la valeur dans le paramètre "Velocity" n'est pas pris en compte. (valeur par défaut : faux)
JogBackward ¹	IN	Bool	Tant que le paramètre est VRAI, l'axe se déplace dans le sens négatif à la vitesse indiquée dans le paramètre "Velocity". Le signe de la valeur dans le paramètre "Velocity" n'est pas pris en compte. (valeur par défaut : faux)
Velocity	IN	Real	Vitesse prédéfinie pour le mode Marche par à-coups (valeur par défaut : 100.0) Valeurs limites : vitesse de démarrage/d'arrêt ≤ Velocity ≤ vitesse maximale
PositionControlled	IN	Bool	<ul style="list-style-type: none"> 0 : régulation de vitesse 1 : régulation de position (valeur par défaut : vrai)
InVelocity	OUT	Bool	VRAI = La vitesse indiquée dans le paramètre "Velocity" a été atteinte.
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error" (valeur par défaut : 0000) 0000)
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID" (valeur par défaut : 0000) 0000)

¹ Si les deux paramètres JogForward et JogBackward sont simultanément à VRAI, l'axe s'arrête avec la décélération configurée. Une erreur est signalée dans les paramètres "Error", "ErrorID" et "ErrorInfo".



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 5,0

- ① "Jog_F" déplace l'axe dans le sens positif en mode Marche par à-coups. Lorsque la vitesse cible de 50,0 est atteinte, cela est signalé via "InVelo_1". L'axe freine jusqu'à s'immobiliser à nouveau après la réinitialisation de Jog_F.
- ② "Jog_B" déplace l'axe dans le sens négatif en mode Marche par à-coups. Lorsque la vitesse cible de 50,0 est atteinte, cela est signalé via "InVelo_1". L'axe freine jusqu'à s'immobiliser à nouveau après la réinitialisation de Jog_B.

Comportement d'annulation

La tâche MC_MoveJog peut être annulée par les tâches Motion Control suivantes :

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

La nouvelle tâche MC_MoveJog annule les tâches Motion Control actives suivantes :

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.10 MC_CommandTable (Exécuter les commandes de l'axe comme séquence de mouvements)

Tableau 10- 72 Instruction MC_CommandTable

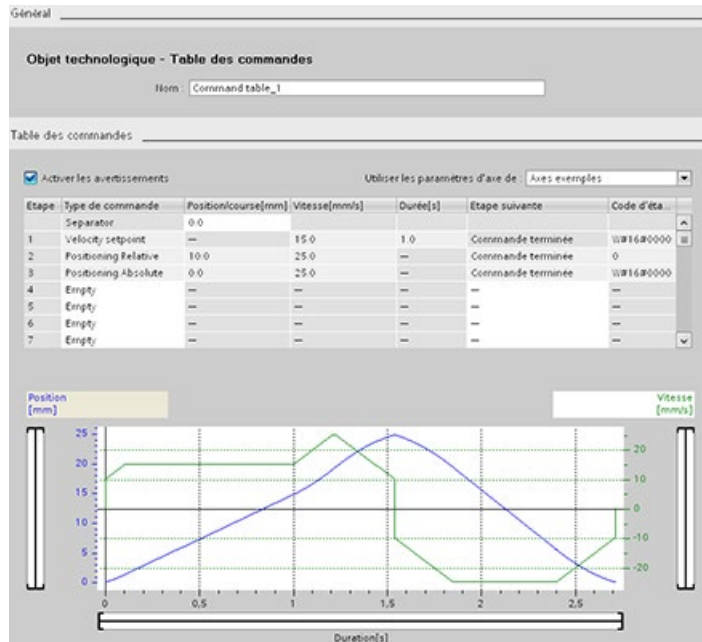
CONT/LOG	SCL	Description
	<pre> "MC_CommandTable_DB" (Axis:=_multi_fb_in_, CommandTable:=_multi_fb_in_, Execute:=_bool_in_, StartIndex:=_uint_in_, EndIndex:=_uint_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_, CurrentIndex=>_uint_out_, Code=>_word_out_); </pre>	<p>Exécute une série de mouvements individuels pour un axe de commande de mouvement, ces mouvements individuels pouvant être combinés en une séquence de mouvement.</p> <p>Les mouvements individuels sont configurés dans une table de commande d'objet technologique pour la sortie de trains d'impulsions (TO_CommandTable_PTO).</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_CommandTable_DB" est le nom du DB d'instance.

Tableau 10- 73 Paramètres pour l'instruction MC_CommandTable

Paramètre et type		Type de données	Valeur initiale	Description
Axis	IN	TO_Axis_1	-	Objet technologique Axe
Table	IN	TO_CommandTable_1	-	Objet technologique Table de commande
Execute	IN	Bool	FAUX	Démarrage de la tâche en présence d'un front montant
StartIndex	IN	Int	1	Démarrage du traitement de la table de commande à cet indice Limites : $1 \leq \text{StartIndex} \leq \text{EndIndex}$
EndIndex	IN	Int	32	Fin du traitement de la table de commande à cet indice Limites : $\text{StartIndex} \leq \text{EndIndex} \leq 32$
Done	OUT	Bool	FAUX	Le traitement de MC_CommandTable a été achevé avec succès.
Busy	OUT	Bool	FAUX	Opération en cours
CommandAborted	OUT	Bool	FAUX	Le traitement de la tâche a été interrompu par une autre tâche.
Error	OUT	Bool	FAUX	Une erreur s'est produite pendant le traitement. La cause est indiquée par les paramètres ErrorID et ErrorInfo..
ErrorID	OUT	Word	16#0000	Code d'erreur
ErrorInfo	OUT	Word	16#0000	Informations d'erreur
Step	OUT	Int	0	Indice en cours de traitement
Code	OUT	Word	16#0000	Identificateur défini par l'utilisateur de l'indice en cours de traitement

Vous pouvez créer la séquence de mouvement désirée dans la fenêtre de configuration "Table de commande" et vérifier le résultat dans la vue graphique du graphique de tendance.



Vous pouvez sélectionner les types de commandes qui doivent être utilisés pour le traitement de la table de commande. Il est possible d'entrer jusqu'à 32 tâches. Les commandes sont traitées l'une après l'autre.

Tableau 10- 74 Types de commandes MC_CommandTable

Type de commande	Description
Empty	Marque de réservation pour toute commande à ajouter. L'entrée Empty n'est pas prise en compte lors du traitement de la table de commande.
Halt	Pause de l'axe. Remarque : cette commande n'est exécutée qu'après une commande de consigne de vitesse.
Positioning Relative	Positionnement de l'axe en fonction de la distance. La commande déplace l'axe de la distance indiquée à la vitesse donnée.
Positioning Absolute	Positionnement de l'axe en fonction de l'emplacement. La commande déplace l'axe à l'emplacement indiqué, en utilisant la vitesse donnée.
Velocity setpoint	Déplacement de l'axe à la vitesse donnée.
Wait	Attente de l'expiration de la durée indiquée. "Wait" n'interrompt pas un accostage actif.
Separator	Ajout d'une ligne séparatrice au-dessus de la ligne sélectionnée. La ligne séparatrice permet de définir plus d'un profil dans une seule table de commande.

Conditions requises pour l'exécution de MC_CommandTable :

- L'objet technologique TO_Axis_PTO V2.0 doit être correctement configuré.
- L'objet technologique TO_CommandTable_PTO doit être correctement configuré.
- L'axe doit être débloqué.

Comportement d'annulation

La tâche MC_CommandTable peut être annulée par les tâches Motion Control suivantes :

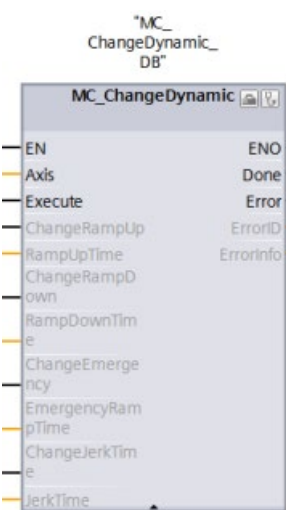
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable

La nouvelle tâche MC_CommandTable annule les tâches Motion Control actives suivantes :

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable
- La tâche Motion Control en cours avec le lancement de la première commande "Positioning Relative", "Positioning Absolute", "Velocity setpoint" ou "Halt"

10.3.7.11 MC_ChangeDynamic (Modifier les paramètres dynamiques de l'axe)

Tableau 10- 75 Instruction MC_ChangeDynamic

CONT/LOG	SCL	Description
	<pre>"MC_ChangeDynamic_DB" (Execute:=_bool_in_, ChangeRampUp:=_bool_in_, RampUpTime:=_real_in_, ChangeRampDown:=_bool_in_, RampDownTime:=_real_in_, ChangeEmergency:=_bool_in_, EmergencyRampTime:=_real_in_, ChangeJerkTime:=_bool_in_, JerkTime:=_real_in_, Done=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Change les paramètres dynamiques d'un axe Motion Control :</p> <ul style="list-style-type: none"> • Changement de la valeur du temps d'accélération • Changement de la valeur du temps de décélération • Changement de la valeur du temps de décélération d'urgence • Changement de la valeur du temps de lissage (à-coup)

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_ChangeDynamic_DB" est le nom du DB d'instance.

Tableau 10- 76 Paramètres pour l'instruction MC_ChangeDynamic

Paramètre et type		Type de données	Description
Axis	IN	TO_Axis_1	Objet technologique Axe
Execute	IN	Bool	Démarrage de la commande en présence d'un front montant. Valeur par défaut : FALSE
ChangeRampUp	IN	Bool	TRUE = Changement du temps d'accélération conformément au paramètre d'entrée "RampUpTime". Valeur par défaut : FALSE
RampUpTime	IN	Real	Temps (en secondes) pour accélérer de l'arrêt à la vitesse maximale configurée sans limite d'à-coup. Valeur par défaut : 5,00 La modification affectera la variable <nom de l'axe>.Config.DynamicDefaults.Acceleration. L'efficacité de la modification est détaillée dans la description de cette variable.
ChangeRampDown	IN	Bool	TRUE = Modifier le temps de décélération conformément au paramètre d'entrée "RampDownTime". Valeur par défaut : FALSE
RampDownTime	IN	Real	Temps (en secondes) pour décélérer l'axe de la vitesse maximale configurée jusqu'à l'immobilisation sans limite d'à-coup. Valeur par défaut : 5,00 La modification affectera la variable <nom de l'axe>.Config.DynamicDefaults.Deceleration. L'efficacité de la modification est détaillée dans la description de cette variable.
ChangeEmergency	IN	Bool	TRUE = Modifier le temps de décélération d'urgence conformément au paramètre d'entrée "EmergencyRampTime". Valeur par défaut : FALSE
EmergencyRampTime	IN	Real	Temps (en secondes) pour décélérer l'axe de la vitesse maximale configurée jusqu'à l'immobilisation sans limite d'à-coup en mode d'arrêt d'urgence. Valeur par défaut : 2,00 La modification affectera la variable <nom de l'axe>.Config.DynamicDefaults.EmergencyDeceleration. L'efficacité de la modification est détaillée dans la description de cette variable.
ChangeJerkTime	IN	Bool	TRUE = Changement du temps de lissage selon le paramètre d'entrée "JerkTime". Valeur par défaut : FALSE
JerkTime	IN	Real	Temps de lissage (en secondes) utilisé pour les rampes d'accélération et de décélération de l'axe. Valeur par défaut : 0,25 La modification affectera la variable <nom de l'axe>.Config.DynamicDefaults.Jerk. L'efficacité de la modification est détaillée dans la description de cette variable.
Done	OUT	Bool	TRUE = Les valeurs modifiées ont été écrites dans le bloc de données technologique. La description des variables montre quand le changement entre en vigueur. Valeur par défaut : FALSE
Error	OUT	Bool	TRUE = Une erreur s'est produite pendant l'exécution de la commande. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo". Valeur par défaut : FALSE

Paramètre et type		Type de données	Description
ErrorID	OUT	Word	Code d'erreur. Valeur par défaut : 16#0000
ErrorInfo	IN	Word	Informations d'erreur. Valeur par défaut : 16#0000

Conditions requises pour l'exécution de MC_ChangeDynamic :

- L'objet technologique TO_Axis_PTO V2.0 doit être correctement configuré.
- L'axe doit être débloqué.

Remarque

Vous ne pouvez utiliser l'instruction MC_ChangeDynamic que pour le raccordement d'un entraînement au moyen de PTO (Pulse Train Output).

Comportement d'annulation

Une instruction MC_ChangeDynamic ne peut être annulée par aucune autre instruction Motion Control.

Une nouvelle instruction MC_ChangeDynamic n'annule aucune tâche Motion Control active.

Remarque

Les paramètres d'entrée "RampUpTime", "RampDownTime", "EmergencyRampTime" et "RoundingOffTime" peuvent contenir des valeurs faisant passer les paramètres d'axe résultants "accélération", "décélération", "décélération d'urgence" et "à-coup" en dehors des limites autorisées.

Veillez à maintenir les paramètres de MC_ChangeDynamic à l'intérieur des limites des paramètres de configuration dynamiques pour l'objet technologique Axe.

10.3.7.12 MC_WriteParam (Écrire dans les paramètres d'un objet technologique)

Vous utilisez l'instruction MC_WriteParam pour écrire dans un nombre choisi de paramètres afin de modifier la fonctionnalité de l'axe à partir du programme utilisateur.

Tableau 10- 77 Instruction MC_WriteParam

CONT/LOG	SCL	Description
	<pre>"MC_WriteParam_DB" (Parameter:=_variant_in_, Value:=_variant_in_, Execute:=_bool_in_, Done:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	<p>L'instruction MC_WriteParam permet d'écrire dans des paramètres publics (par exemple, accélération et valeurs du DB utilisateur).</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC_WriteParam_DB" est le nom du DB d'instance.

Vous pouvez écrire dans les paramètres publics. Vous ne pouvez pas écrire dans "MotionStatus" et "StatusBits". Les paramètres valides sont énumérés dans le tableau ci-dessous :

Nom du paramètre inscriptible	Nom du paramètre inscriptible
Actor.InverseDirection	DynamicDefaults.Acceleration
Actor.DirectionMode	DynamicDefaults.Deceleration
Actor.DriveParameter.PulsesPerDriveRevolution	DynamicDefaults.Jerk
Sensor[1].ActiveHoming.Mode	DynamicDefaults.EmergencyDeceleration
Sensor[1].ActiveHoming.Sidelnput	PositionLimitsHW.Active
Sensor[1].ActiveHoming.Offset	PositionLimitsHW.MaxSwitchedLevel
Sensor[1].ActiveHoming.SwitchedLevel	PositionLimitsHW.MinSwitchedLevel
Sensor[1].PassiveHoming.Mode	PositionLimitsSW.Active
Sensor[1].PassiveHoming.Sidelnput	PositionLimitsSW.MinPosition
Sensor[1].PassiveHoming.SwitchedLevel	PositionLimitsSW.MaxPosition
Units.LengthUnit	Homing.AutoReversal
Mechanics.LeadScrew	Homing.ApproachDirection
DynamicLimits.MinVelocity	Homing.ApproachVelocity
DynamicLimits.MaxVelocity	Homing.ReferencingVelocity

Tableau 10- 78 Paramètres pour l'instruction MC_WriteParam

Paramètre et type		Type de données	Description
PARAMNAME	IN	Variant	Nom du paramètre dans lequel la valeur est écrite
VALUE	IN	Variant	Valeur à écrire dans le paramètre affecté
EXECUTE	IN	Bool	Démarrer l'instruction. Valeur par défaut : FALSE
DONE	OUT	Bool	La valeur a été écrite. Valeur par défaut : FALSE
BUSY	OUT	Bool	Si TRUE, l'instruction est en cours d'exécution. Valeur par défaut : FALSE
ERROR	OUT	Real	Si TRUE, une erreur s'est produite. Valeur par défaut : FALSE
ERRORID	OUT	Word	ID de l'erreur
ERRORINFO	OUT	Word	Information associée à ERRORID

Tableau 10- 79 Codes d'erreur pour ERRORID et ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Description
0	0	Modification réussie d'un paramètre du DB de l'objet technologique Axe
8410 ^[1]	0028 ^[1]	Réglage d'un paramètre invalide (paramètre du DB de l'objet technologique Axe de longueur incorrecte)
8410 ^[1]	0029 ^[1]	Réglage d'un paramètre invalide (pas de paramètre du DB de l'objet technologique Axe)
8410 ^[1]	002B ^[1]	Réglage d'un paramètre invalide (paramètre du DB de l'objet technologique Axe en lecture seule)
8410 ^[1]	002C ^[1]	Réglage d'un paramètre valide, mais l'axe n'est pas inhibé
Erreur config. ^[2]	Erreur config. ^[2]	Réglage d'un paramètre valide (paramètre du DB de l'objet technologique Axe public en lecture seule) hors plage
Erreur config. ^[3]	Erreur config. ^[3]	Réglage d'un paramètre valide (paramètre du DB de l'objet technologique Axe public) hors plage

[1] Erreur sur MC_WriteParam

[2] Erreur sur MC_Power

[3] Erreur sur MC_Power et MC_MoveXXX ou MC_CommandTable

Remarque

Pour le raccordement d'un entraînement par PROFIdrive/une sortie analogique, vous ne pouvez pas écrire, avec MC_WriteParam, des paramètres qui requièrent un redémarrage de l'objet technologique. L'affirmation de l'aide en ligne de TIA Portal concernant cette instruction est incorrecte.

10.3.7.13 Instruction MC_ReadParam (Lire les paramètres d'un objet technologique)

L'instruction MC_ReadParam vous permet de lire un nombre choisi de paramètres qui indiquent la position actuelle, la vitesse, etc. de l'axe défini dans l'entrée Axe.

Tableau 10- 80 Instruction MC_ReadParam

CONT/LOG	SCL	Description
	<pre>"MC_ReadParam_DB" (Enable:=_bool_in_, Parameter:=_variant_in_, Value:=_variant_in_out_, Valid:=_bool_out_, Busy:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	<p>L'instruction MC_ReadParam vous permet de lire des valeurs d'état individuelles, indépendamment du point de contrôle du cycle.</p>

- STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- Dans l'exemple SCL, "MC_ReadParam_DB" est le nom du DB d'instance.

L'instruction MC_ReadParam fonctionne suivant un comportement d'activation. Tant que l'entrée "Activer" est vrai, l'instruction lit le "Paramètre" indiqué dans le lieu de stockage de la "Valeur".

La valeur "MotionStatus" "Position" s'actualise à chaque point de contrôle du cycle (CCP) sur la base de la valeur HSC en cours.

La valeur "MotionStatus" "Velocity" est la vitesse de commande à la fin du segment en cours (actualisée ~10ms). MC_ReadParam peut également lire cette valeur.

Si une erreur se produit, l'instruction commute sur un état d'erreur qui ne peut être remis à zéro que par un nouveau front montant de l'entrée "Activer".

Tableau 10- 81 Paramètres pour l'instruction MC_ReadParam

Paramètre et type	Type de données	Description	
ENABLE	IN	Bool	Démarre l'instruction. Valeur par défaut : FALSE
PARAMETER	IN	Variant	Pointeur désignant le paramètre TO à lire.
VALID	OUT	Bool	Si TRUE, la valeur a été lue. Valeur par défaut : FALSE
BUSY	OUT	Bool	Si TRUE, l'instruction est utilisée. Valeur par défaut : FALSE
ERROR	OUT	Real	Si TRUE, une erreur s'est produite. Valeur par défaut : FALSE
ERRORID	OUT	Word	Code de l'erreur. Valeur par défaut : 0
ERRORINFO	OUT	Word	Information complétant ERRORID. Valeur par défaut : 0
VALUE	INOUT	Variant	Pointeur désignant l'emplacement où la valeur lue est stockée.

Tableau 10- 82 Codes d'erreur pour ERRORID et ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Description
0	0	Un paramètre a été lu avec succès
8410	0028	Paramètre incorrect (longueur incorrecte)
8410	0029	Paramètre incorrect (pas de DB TO)
8410	0030	Paramètre incorrect (pas lisible)
8411	0032	Paramètre incorrect (valeur erronée)

Paramètres TO

La variable "MotionStatus" de l'axe peut prendre quatre valeurs. Vous souhaitez visualiser des modifications apportées à ces valeurs lisibles pendant l'exécution du programme :

Nom de variable	Type de données	Lisible via MC_ReadParam
MotionStatus :	Structure	Non
• Position	REAL	Oui
• Vitesse	REAL	Oui
• Distance	REAL	Oui
• TargetPosition	REAL	Oui

10.3.8 Suivi des commandes actives

10.3.8.1 Suivi des instructions MC avec paramètre de sortie "Done"

Les instructions Motion Control avec paramètre de sortie "Done" sont démarrées par le paramètre d'entrée "Execute" et se concluent de manière définie (par exemple, la réussite du référencement pour l'instruction "MC_Home"). La tâche est achevée et l'axe est à l'arrêt.

- Le paramètre de sortie "Done" a la valeur VRAI si la tâche s'est achevée correctement.
- Les paramètres de sortie "Busy", "CommandAborted" et "Error" signalent respectivement que la tâche est encore en cours de traitement, qu'elle a été annulée ou qu'une erreur s'est produite. L'instruction Motion Control "MC_Reset" ne peut pas être annulée et n'a donc pas de paramètre de sortie "CommandAborted".
 - Pendant l'exécution de la tâche Motion Control, le paramètre de sortie "Busy" indique la valeur VRAI. Si la tâche est achevée, a été annulée ou a été arrêtée en raison d'une erreur, le paramètre de sortie "Busy" prend la valeur FAUX. Cette modification a lieu quel que soit le signal du paramètre d'entrée "Execute".
 - Les paramètres de sortie "Done", "CommandAborted" et "Error" prennent la valeur VRAI pendant au moins un cycle. Ces messages d'état sont rémanents pendant que le paramètre d'entrée "Execute" a la valeur VRAI.

Les tâches des instructions Motion Control suivantes ont une fin définie :

- MC_Reset
- MC_Home
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative

Le comportement des bits d'état est représenté ci-après pour divers types de situations.

- Le premier exemple montre le comportement de l'axe pour une tâche achevée. Si la tâche Motion Control a été exécutée complètement lorsqu'elle s'achève, le paramètre de sortie "Done" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "Done".
- Le deuxième exemple montre le comportement de l'axe pour une tâche annulée. Si la tâche Motion Control est annulée pendant son exécution, le paramètre de sortie "CommandAborted" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "CommandAborted".
- Le troisième exemple montre le comportement de l'axe si une erreur se produit. Si une erreur se produit pendant l'exécution de la tâche Motion Control, le paramètre de sortie "Error" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "Error".

Tableau 10- 83 Exemple 1 : Exécution complète de la tâche

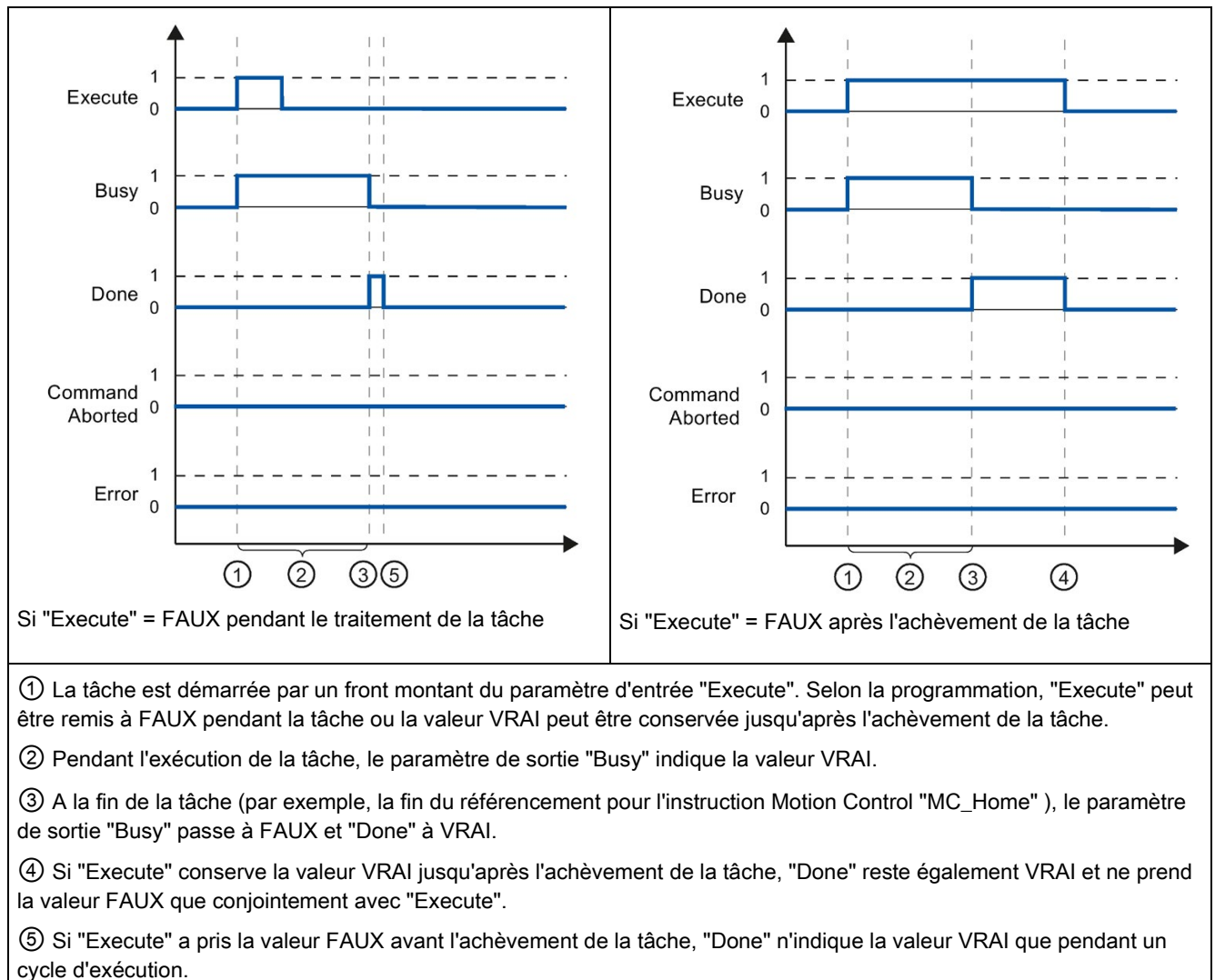


Tableau 10- 84 Exemple 2 : Annulation de la tâche

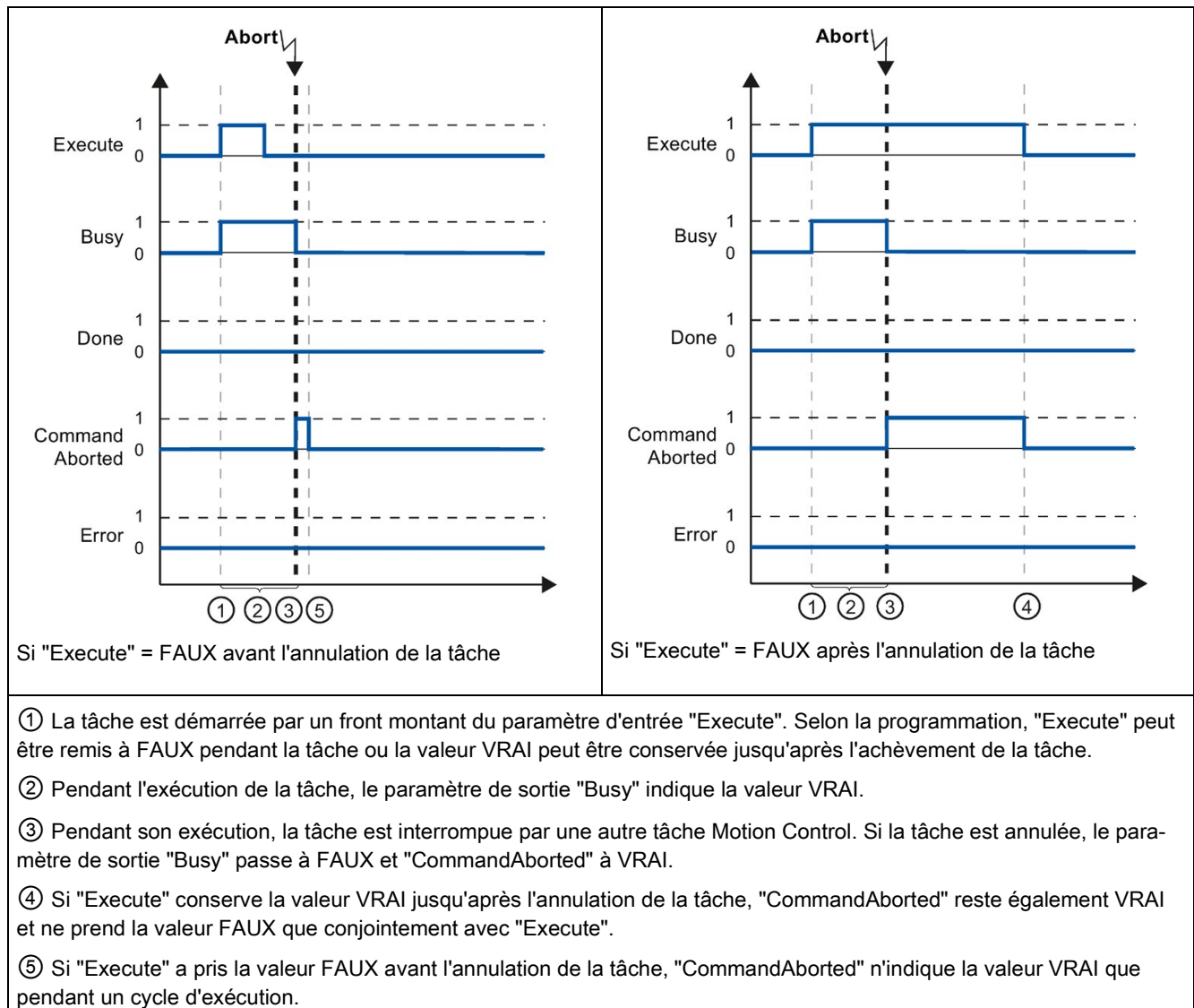
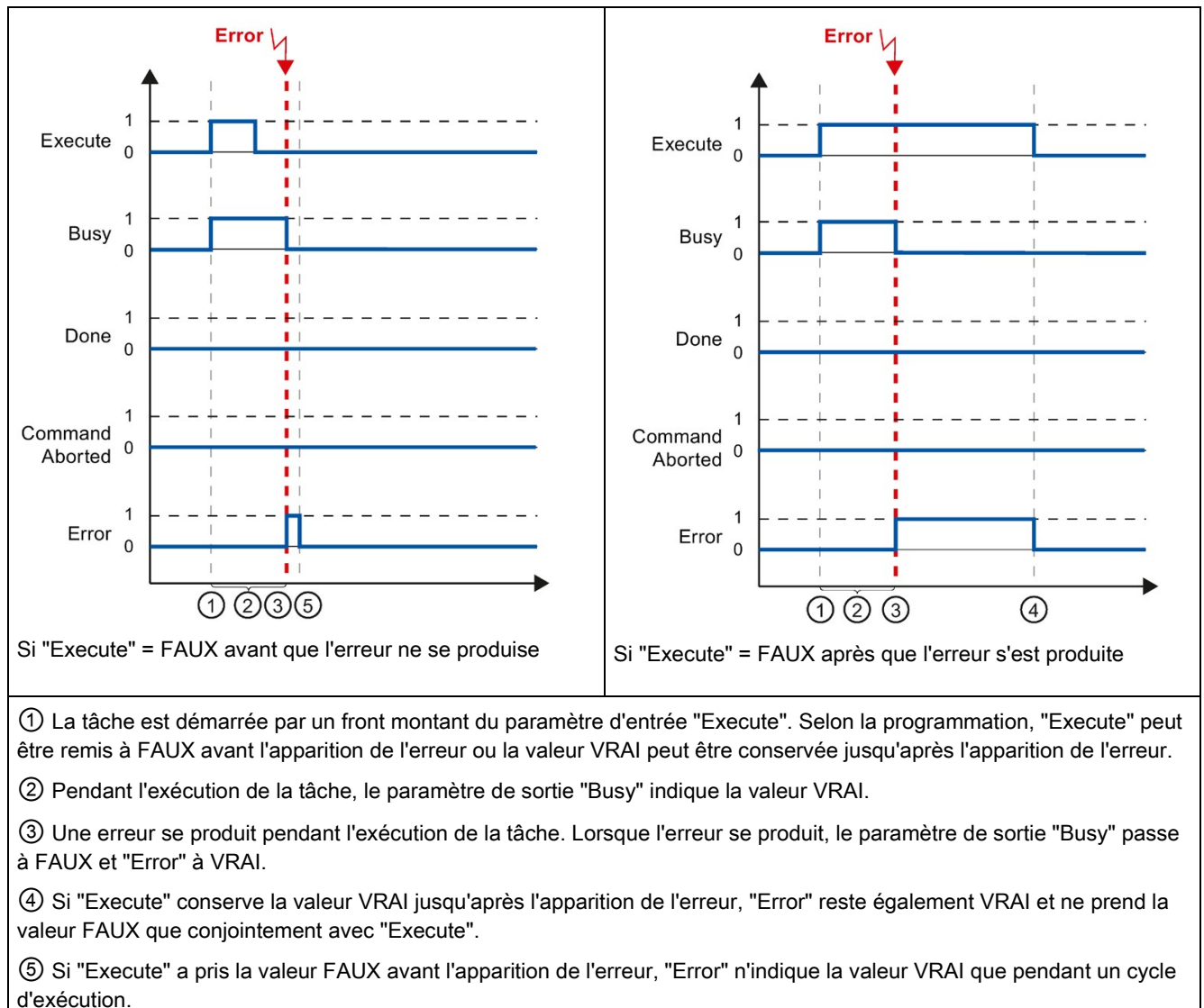


Tableau 10- 85 Exemple 3 : Erreur pendant l'exécution de la tâche



10.3.8.2 Surveillance de l'instruction MC_Velocity

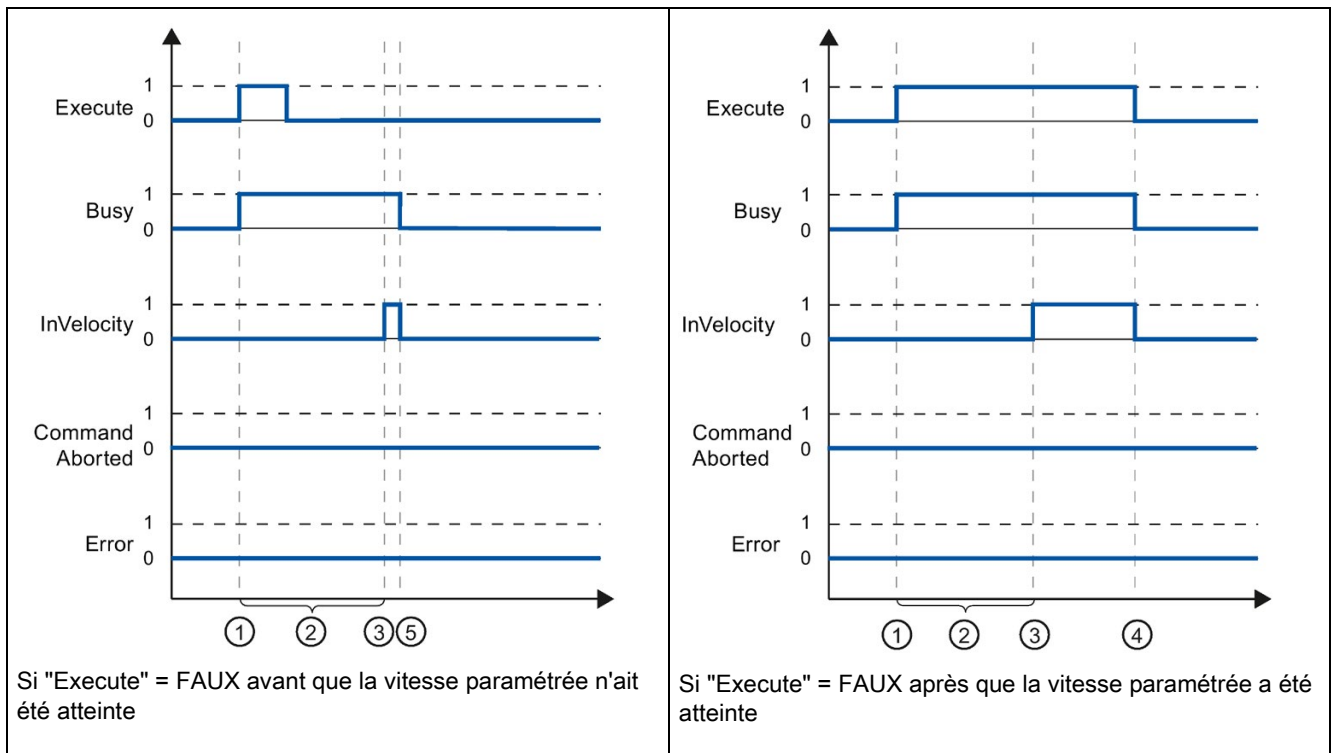
Les tâches de l'instruction Motion Control "MC_MoveVelocity" réalisent un déplacement à la vitesse indiquée :

- Les tâches de l'instruction Motion Control "MC_MoveVelocity" n'ont pas de fin définie. L'objectif de la tâche est rempli lorsque la vitesse paramétrée est atteinte pour la première fois et que l'axe se déplace à vitesse constante. Lorsque la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- La tâche est achevée lorsque la vitesse paramétrée a été atteinte et que le paramètre d'entrée "Execute" a pris la valeur FAUX. Toutefois, le mouvement de l'axe n'est pas encore achevé à l'achèvement de la tâche. Son mouvement peut, par exemple, être arrêté à l'aide de la tâche Motion Control "MC_Halt".
- Les paramètres de sortie "Busy", "CommandAborted" et "Error" signalent respectivement que la tâche est encore en cours de traitement, qu'elle a été annulée ou qu'une erreur s'est produite.
 - Pendant l'exécution de la tâche Motion Control, le paramètre de sortie "Busy" indique la valeur VRAI. Si la tâche est achevée, a été annulée ou a été arrêtée en raison d'une erreur, le paramètre de sortie "Busy" prend la valeur FAUX. Cette modification a lieu quel que soit le signal du paramètre d'entrée "Execute".
 - Les paramètres de sortie "InVelocity", "CommandAborted" et "Error" prennent la valeur VRAI pendant au moins un cycle lorsque leur condition est remplie. Ces messages d'état sont rémanents pendant que le paramètre d'entrée "Execute" a la valeur VRAI.

Le comportement des bits d'état est représenté ci-après pour divers types de situations :

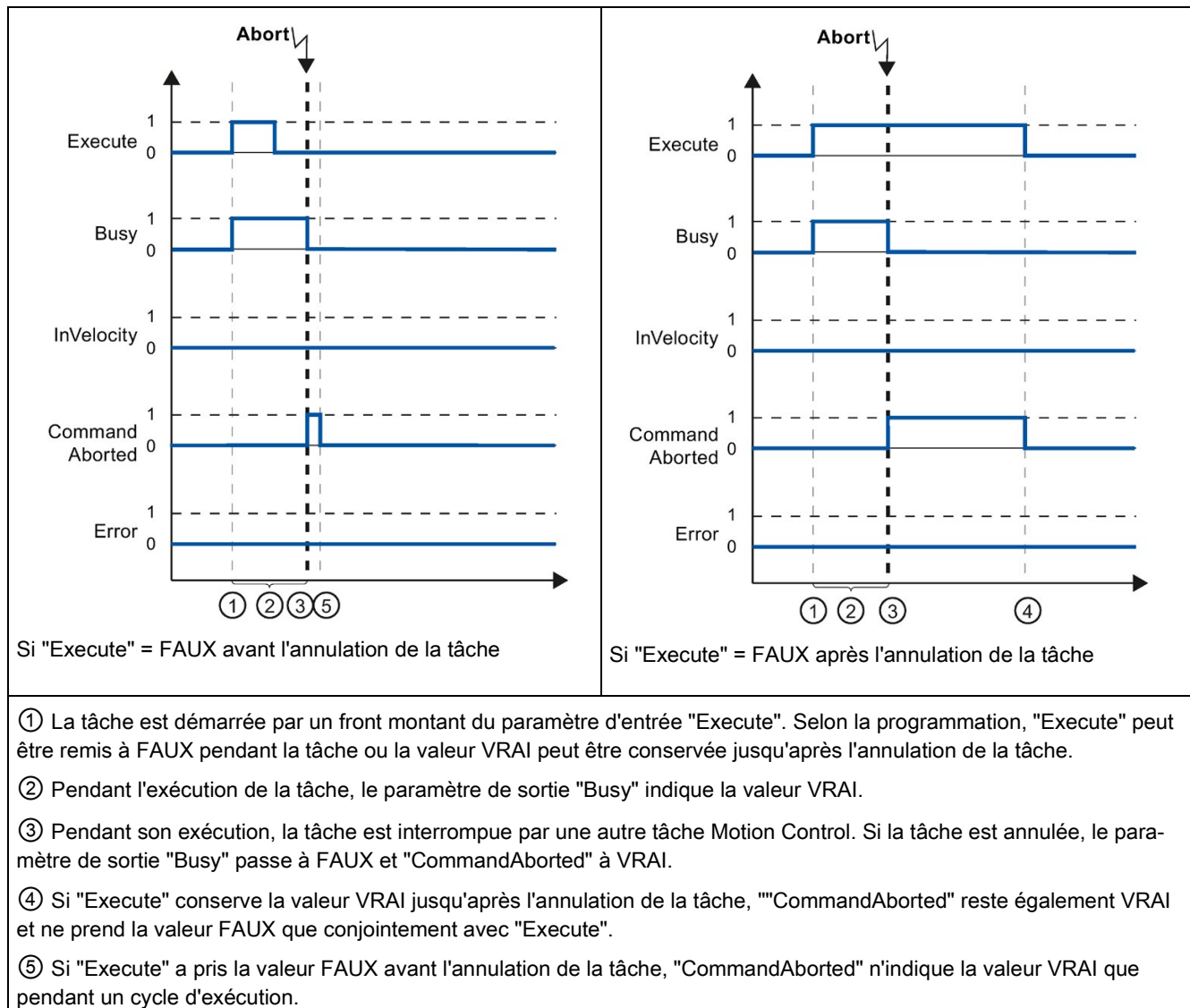
- Le premier exemple montre ce qui se passe lorsque l'axe atteint la vitesse paramétrée. Si la tâche Motion Control a été exécutée au moment où la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "InVelocity".
- Le deuxième exemple montre ce qui se passe si la tâche est annulée avant que la vitesse paramétrée n'ait été atteinte. Si la tâche Motion Control est annulée avant que la vitesse paramétrée n'ait été atteinte, le paramètre de sortie "CommandAborted" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "CommandAborted".
- Le troisième exemple montre le comportement de l'axe si une erreur survient avant que la vitesse paramétrée n'ait été atteinte. Si une erreur se produit pendant l'exécution de la tâche Motion Control avant que la vitesse paramétrée n'ait été atteinte, le paramètre de sortie "Error" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "Error".

Tableau 10- 86 Exemple 1 : La vitesse paramétrée est atteinte



- ① La tâche est démarrée par un front montant du paramètre d'entrée "Execute". Selon la programmation, "Execute" peut être remis à FAUX avant que la vitesse paramétrée n'ait été atteinte ou bien seulement après que la vitesse paramétrée a été atteinte.
- ② Pendant l'exécution de la tâche, le paramètre de sortie "Busy" indique la valeur VRAI.
- ③ Lorsque la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- ④ Si "Execute" conserve la valeur VRAI même après que la vitesse paramétrée a été atteinte, la tâche reste active. "InVelocity" et "Busy" conservent la valeur VRAI et leur état ne passe à FAUX que conjointement avec "Execute".
- ⑤ Si "Execute" a pris la valeur FAUX avant que la vitesse paramétrée n'ait été atteinte, la tâche est achevée lorsque la vitesse paramétrée est atteinte. "InVelocity" indique la valeur VRAI pendant un cycle d'exécution et passe à FAUX conjointement avec "Busy".

Tableau 10- 87 Exemple 2 : La tâche est annulée avant que la vitesse paramétrée n'ait été atteinte

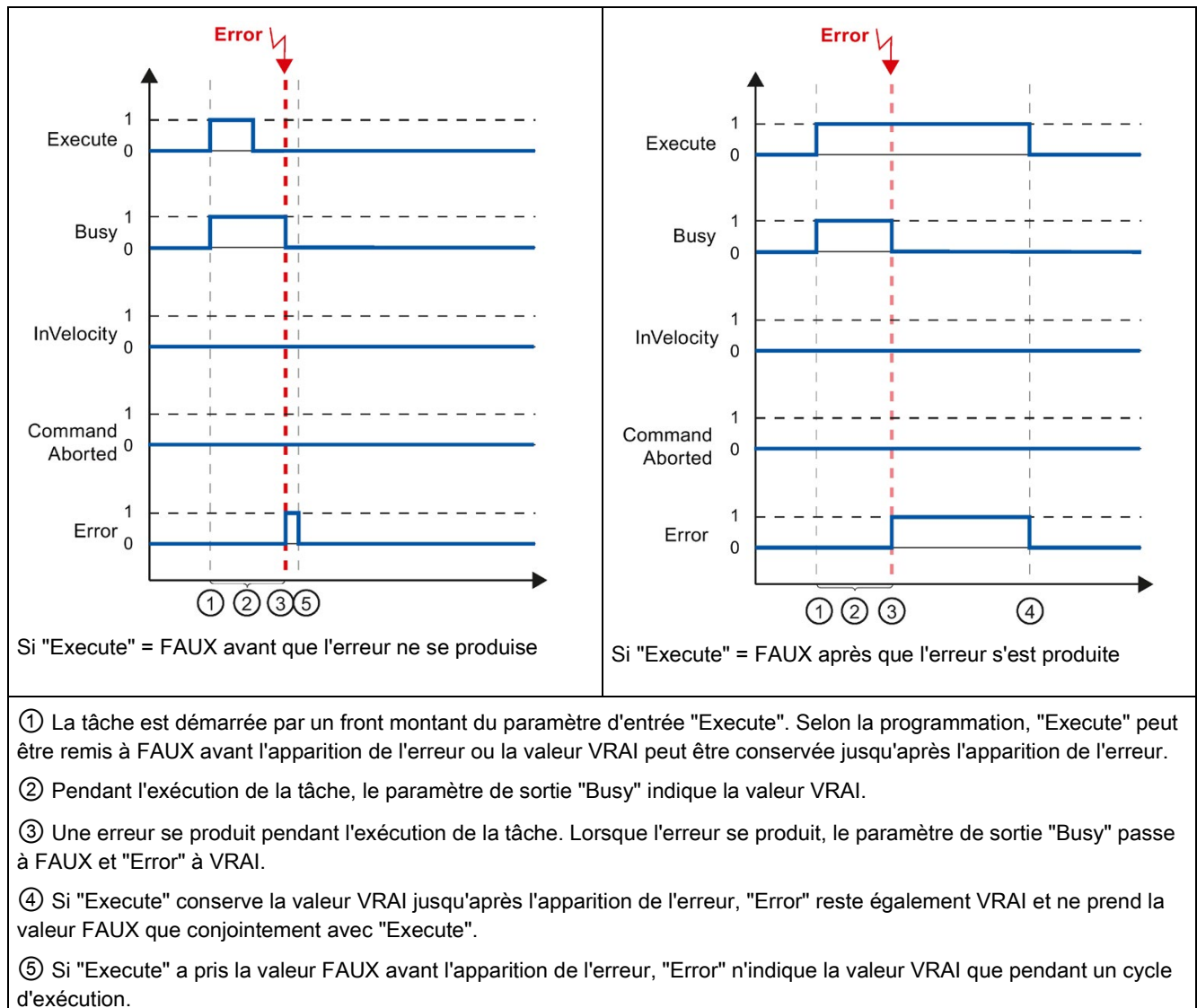


Remarque

Une annulation n'est pas signalée dans le paramètre de sortie "CommandAborted" dans les conditions suivantes :

- La vitesse paramétrée a été atteinte, le paramètre d'entrée "Execute" a la valeur FAUX et une nouvelle tâche Motion Control est déclenchée.
- Lorsque la vitesse paramétrée a été atteinte et que le paramètre d'entrée "Execute" a pris la valeur FAUX la tâche est achevée. Ainsi, le démarrage d'une nouvelle tâche n'est pas signalée comme une annulation.

Tableau 10- 88 Exemple 3 : Une erreur se produit avant que la vitesse paramétrée n'ait été atteinte



- ① La tâche est démarrée par un front montant du paramètre d'entrée "Execute". Selon la programmation, "Execute" peut être remis à FAUX avant l'apparition de l'erreur ou la valeur VRAI peut être conservée jusqu'après l'apparition de l'erreur.
- ② Pendant l'exécution de la tâche, le paramètre de sortie "Busy" indique la valeur VRAI.
- ③ Une erreur se produit pendant l'exécution de la tâche. Lorsque l'erreur se produit, le paramètre de sortie "Busy" passe à FAUX et "Error" à VRAI.
- ④ Si "Execute" conserve la valeur VRAI jusqu'après l'apparition de l'erreur, "Error" reste également VRAI et ne prend la valeur FAUX que conjointement avec "Execute".
- ⑤ Si "Execute" a pris la valeur FAUX avant l'apparition de l'erreur, "Error" n'indique la valeur VRAI que pendant un cycle d'exécution.

Remarque

Une erreur n'est pas signalée dans le paramètre de sortie "Error" dans les conditions suivantes :

- La vitesse paramétrée a été atteinte, le paramètre d'entrée "Execute" a la valeur FAUX et une erreur d'axe se produit (accostage du fin de course logiciel, par exemple).
- Lorsque la vitesse paramétrée a été atteinte et que le paramètre d'entrée "Execute" a pris la valeur FAUX la tâche est achevée. Après l'achèvement de la tâche, l'erreur d'axe est signalée uniquement dans l'instruction Motion Control "MC_Power".

10.3.8.3 Surveillance de l'instruction MC_MoveJog

Les tâches de l'instruction Motion Control "MC_MoveJog" réalisent un mouvement en mode Marche par à-coups.

- Les tâches de l'instruction Motion Control "MC_MoveJog" n'ont pas de fin définie. L'objectif de la tâche est rempli lorsque la vitesse paramétrée est atteinte pour la première fois et que l'axe se déplace à vitesse constante. Lorsque la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- La tâche est achevée lorsque le paramètre d'entrée "JogForward" ou "JogBackward" a pris la valeur FAUX et que l'axe s'est immobilisé.
- Les paramètres de sortie "Busy", "CommandAborted" et "Error" signalent respectivement que la tâche est encore en cours de traitement, qu'elle a été annulée ou qu'une erreur s'est produite.
 - Pendant l'exécution de la tâche Motion Control, le paramètre de sortie "Busy" indique la valeur VRAI. Si la tâche est achevée, a été annulée ou a été arrêtée en raison d'une erreur, le paramètre de sortie "Busy" prend la valeur FAUX.
 - Le paramètre de sortie "InVelocity" est à l'état VRAI tant que l'axe se déplace à la vitesse paramétrée. Les paramètres de sortie "CommandAborted" et "Error" indiquent l'état pendant au moins un cycle. Ces messages d'état sont rémanents aussi longtemps que l'un ou l'autre paramètre d'entrée "JogForward" ou "JogBackward" a la valeur VRAI.

Le comportement des bits d'état est représenté ci-après pour divers types de situations.

- Le premier exemple montre le comportement de l'axe si la vitesse paramétrée est atteinte et maintenue. Si la tâche Motion Control a été exécutée au moment où la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- Le deuxième exemple montre le comportement de l'axe si la tâche est annulée. Si la tâche Motion Control est annulée pendant son exécution, le paramètre de sortie "CommandAborted" prend la valeur VRAI. Ce comportement est indépendant du fait que la vitesse paramétrée a été ou non atteinte.
- Le troisième exemple montre le comportement de l'axe si une erreur survient. Si une erreur se produit pendant l'exécution de la tâche Motion Control, le paramètre de sortie "Error" prend la valeur VRAI. Ce comportement est indépendant du fait que la vitesse paramétrée a été ou non atteinte.

Tableau 10- 89 Exemple 1 : La vitesse paramétrée est atteinte et maintenue

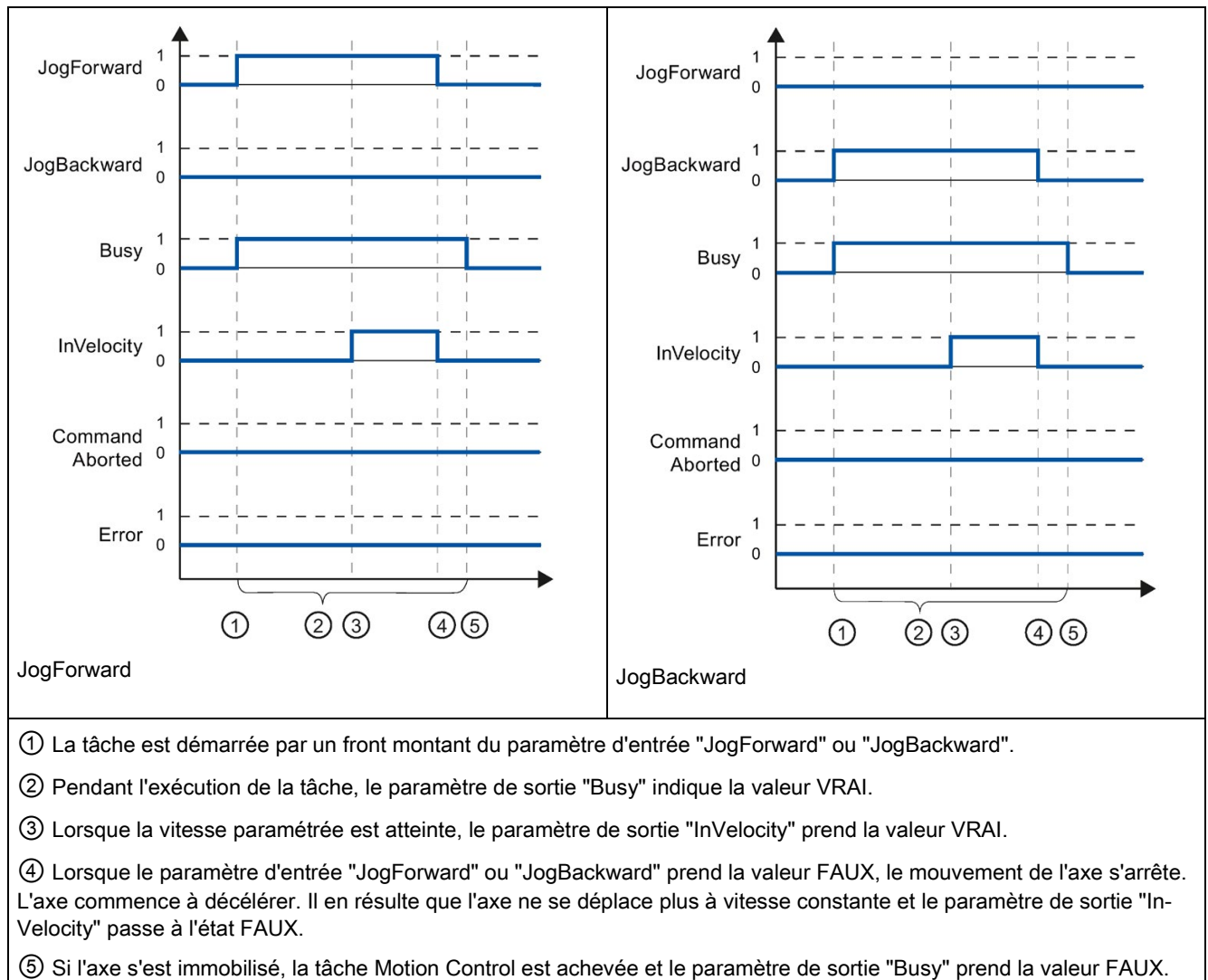
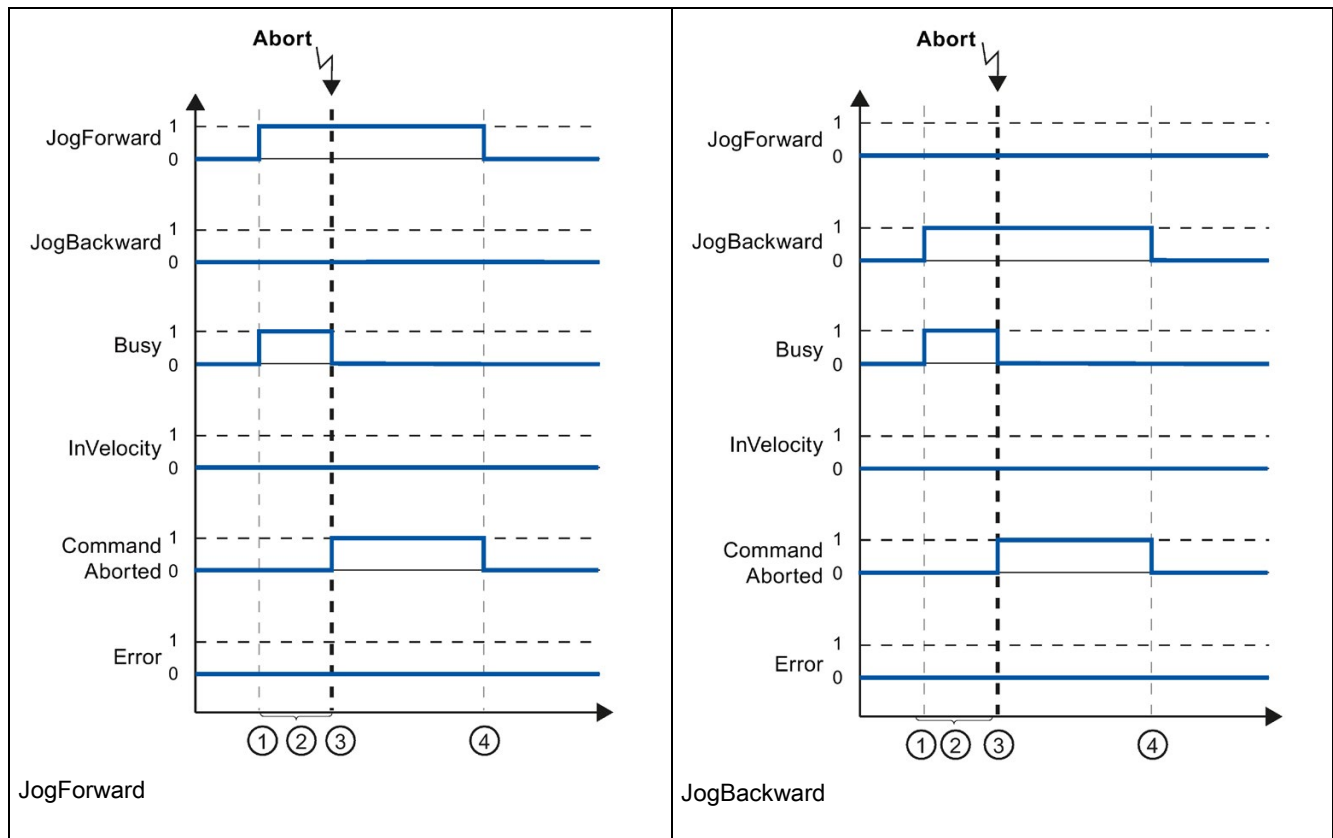


Tableau 10- 90 Exemple 2 : La tâche est annulée pendant l'exécution

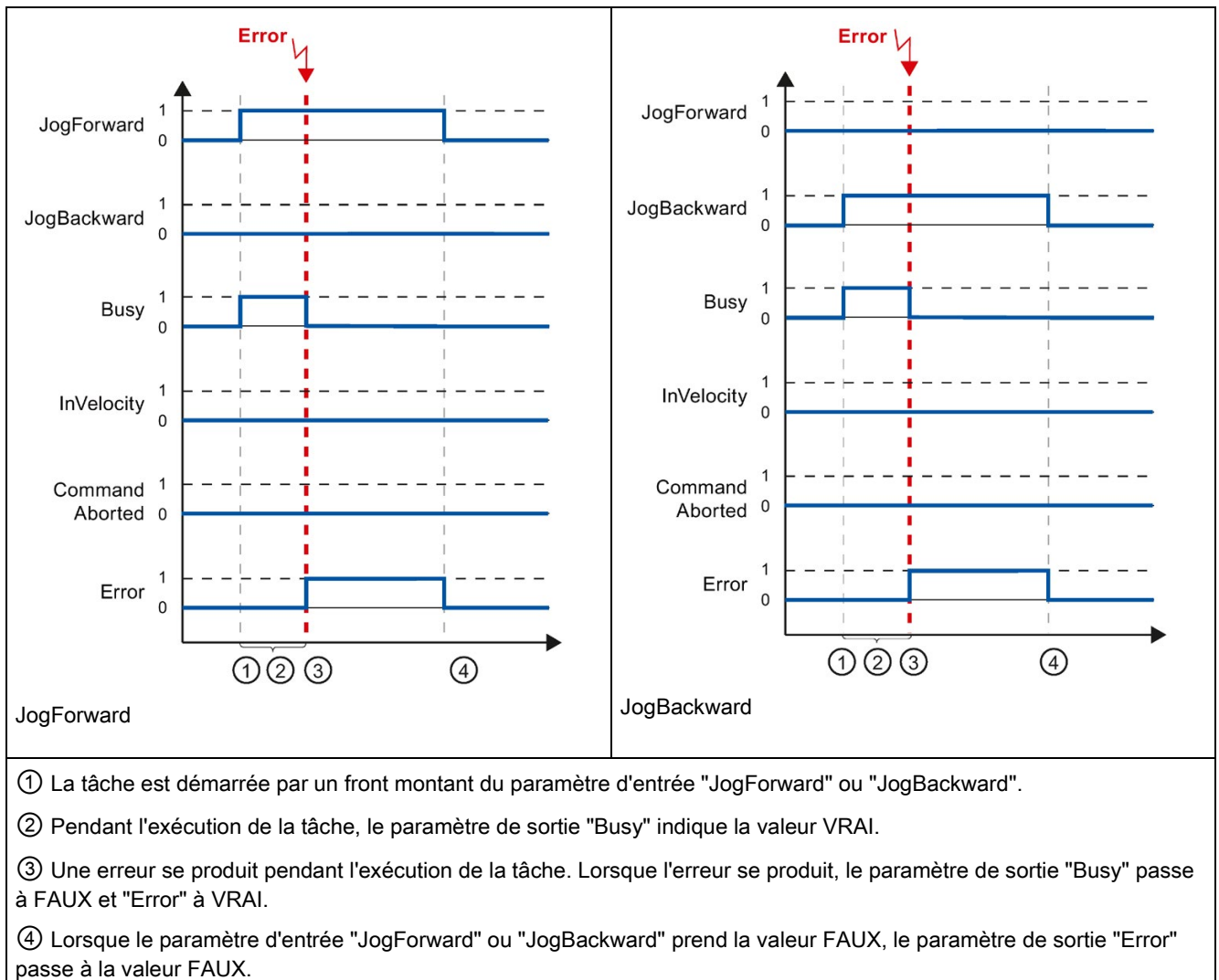


- ① La tâche est démarrée par un front montant du paramètre d'entrée "JogForward" ou "JogBackward".
- ② Pendant l'exécution de la tâche, le paramètre de sortie "Busy" indique la valeur VRAI.
- ③ Pendant son exécution, la tâche est interrompue par une autre tâche Motion Control. Si la tâche est annulée, le paramètre de sortie "Busy" passe à FAUX et "CommandAborted" à VRAI.
- ④ Lorsque le paramètre d'entrée "JogForward" ou "JogBackward" prend la valeur FAUX, le paramètre de sortie "CommandAborted" passe à la valeur FAUX.

Remarque

L'annulation de la tâche est signalée dans le paramètre de sortie "CommandAborted" pendant un cycle d'exécution seulement, si toutes les conditions ci-dessous sont remplies :
 Les paramètres d'entrée "JogForward" et "JogBackward" ont la valeur FAUX (mais l'axe est encore en cours de décélération) et une nouvelle tâche Motion Control est déclenchée.

Tableau 10- 91 Exemple 3 : Une erreur s'est produite pendant l'exécution de la tâche

**Remarque**

L'apparition de l'erreur est signalée dans le paramètre de sortie "Error" pendant un cycle d'exécution seulement, si toutes les conditions ci-dessous sont remplies :

Les paramètres d'entrée "JogForward" et "JogBackward" ont la valeur FAUX (mais l'axe est encore en cours de décélération) et une nouvelle erreur se produit (accostage du fin de course logiciel, par exemple).

10.3.9 Valeurs ErrorId et ErrorInfo pour Motion Control

Vous trouverez dans les tableaux suivants la liste de toutes les valeurs ErrorID et ErrorInfo pouvant être renvoyées dans les instructions Motion Control et l'adaptation des données. Ces tableaux indiquent non seulement la cause des erreurs, mais donnent également des solutions pour éliminer les erreurs.

En fonction de la réaction à l'erreur, l'axe est arrêté dans le cas d'erreurs de fonctionnement avec arrêt de l'axe. Les réactions suivantes sont possibles :

- **Retrait du déblocage**

La consigne zéro est sortie et le déblocage est retiré. L'axe est freiné en fonction de la configuration dans l'entraînement jusqu'à immobilisation.

- **Arrêt avec rampe d'arrêt d'urgence**

Les commandes de déplacement actives sont annulées. L'axe est freiné avec la décélération d'arrêt d'urgence configurée dans "Objet technologique > Paramètres avancés > Dynamique > Rampe d'arrêt d'urgence" jusqu'à l'immobilisation, sans limitation des à-coups.

Erreurs de fonctionnement avec arrêt de l'axe

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
16#8000		Erreur d'entraînement, "Entraînement prêt" désactivé		-
	16#0001	-	Acquitter l'erreur avec l'instruction "MC_Reset" ; fournir le signal d'entraînement ; redémarrer la commande si nécessaire.	
16#8001		Le fin de course logiciel inférieur a été déclenché.		Arrêt avec rampe d'arrêt d'urgence
	16#000E	La position du fin de course logiciel inférieur a été atteinte avec la décélération actuellement configurée.	Acquitter l'erreur avec l'instruction "MC_Reset" ; utiliser une commande de déplacement pour déplacer l'axe hors de la plage du fin de course logiciel dans le sens positif.	
	16#000F	La position du fin de course logiciel inférieur a été atteinte avec la décélération d'arrêt d'urgence.		
	16#0010	La position du fin de course logiciel inférieur a été dépassée avec la décélération d'arrêt d'urgence.		
16#8002		Le fin de course logiciel supérieur a été déclenché.		Arrêt avec rampe d'arrêt d'urgence
	16#000E	La position du fin de course logiciel supérieur a été atteinte avec la décélération actuellement configurée.	Acquitter l'erreur avec l'instruction "MC_Reset" ; utiliser une commande de déplacement pour déplacer l'axe hors de la plage du fin de course logiciel dans le sens négatif.	
	16#000F	La position du fin de course logiciel supérieur a été atteinte avec la décélération d'arrêt d'urgence.		
	16#0010	La position du fin de course logiciel supérieur a été dépassée avec la décélération d'arrêt d'urgence.		

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
16#8003		Le fin de course matériel inférieur a été accosté.		
	16#000E	Le fin de course matériel inférieur a été accosté. L'axe a été arrêté avec la décélération d'arrêt d'urgence. (La came de référence n'a pas été trouvée lors d'une prise de référence active)	Acquitter l'erreur pour un axe débloqué avec l'instruction "MC_Reset" ; utiliser une commande de déplacement pour déplacer l'axe hors de la plage du fin de course matériel dans le sens positif.	Pour un raccordement d'entraînement par PTO (Pulse Train Output) : Arrêt avec rampe d'arrêt d'urgence Pour un raccordement d'entraînement par PROFIdrive/sortie analogique : Retrait du déblocage
16#8004		Le fin de course matériel supérieur a été accosté.		
	16#000E	Le fin de course matériel supérieur a été accosté. L'axe a été arrêté avec la décélération d'arrêt d'urgence. (La came de référence n'a pas été trouvée lors d'une prise de référence active)	Acquitter l'erreur pour un axe débloqué avec l'instruction "MC_Reset" ; utiliser une commande de déplacement pour déplacer l'axe hors de la plage du fin de course matériel dans le sens négatif.	Pour un raccordement d'entraînement par PTO (Pulse Train Output) : Arrêt avec rampe d'arrêt d'urgence Pour un raccordement d'entraînement par PROFIdrive/sortie analogique : Retrait du déblocage
16#8005		PTO/HSC sont déjà utilisés par un autre axe.		
	16#0001	-	L'axe a été mal configuré : Corriger la configuration de PTO (Pulse Train Output) / HSC (High Speed Counter) et la charger dans l'automate. Plusieurs axes doivent fonctionner avec un PTO : Un autre axe utilise PTO / HSC. Si l'axe actuel doit prendre le contrôle, il faut bloquer l'autre axe avec "MC_Power" Enable = FALSE.	-
16#8006		Erreur de communication dans le panneau de commande		
	16#0012	Le délai imparti a été dépassé.	Vérifier le câblage et actionner de nouveau le bouton "Commande manuelle".	Retrait du déblocage
16#8007		Impossible de débloquent l'axe		
	16#0025	Redémarrage en cours d'exécution	Attendre que le redémarrage de l'axe soit terminé.	
	16#0026	La procédure de chargement à l'état MARCHE est en cours d'exécution.	Attendre que la procédure de chargement soit terminée.	

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
16#8008		Sens de déplacement invalide		-
	16#002E	Le sens de déplacement choisi n'est pas autorisé.	<ul style="list-style-type: none"> • Modifier le sens du déplacement, puis relancer la commande. • Modifier le sens de rotation autorisé dans la configuration de l'objet technologique sous "Paramètres avancés > Mécanique". Relancer la commande. 	
	16#002F	Un mouvement en sens inverse est impossible avec le sens de déplacement sélectionné.		
16#8009		Contact de référence/top zéro du codeur introuvable		Arrêt avec rampe d'arrêt d'urgence
	16#0033	Erreur de configuration, de matériel ou de pose du codeur ou erreur sur la came de référence.	<ul style="list-style-type: none"> • Raccorder un appareil approprié. • Vérifier l'appareil (E/S). • Comparer la configuration dans HW Config à celle de l'objet technologique. 	
16#800A		Signalisation de défaut du codeur		Retrait du déblocage
	16#0001	-	Vérifier le fonctionnement, les liaisons et les E/S de l'appareil.	
	16#0034	Défaut matériel sur le codeur		
	16#0035	Codeur encrassé		
	16#0036	Erreur lors de la lecture de la valeur absolue du codeur	Comparer le type de codeur dans le paramètre P979 de l'entraînement ou du codeur aux données de configuration de l'objet technologique.	
	16#0037	Surveillance du top zéro du codeur	Le codeur signale une erreur dans la surveillance du top zéro (code de défaut 0x0002 dans Gx_XIST2, voir profil PROFIdrive). Vérifier la compatibilité électromagnétique de l'installation (CEM).	
	16#0038	Le codeur est à l'état "Stationnement".	<ul style="list-style-type: none"> • Rechercher la cause de l'erreur dans l'entraînement ou le codeur raccordé. • Vérifier si la signalisation de défaut a éventuellement été déclenchée par une opération de mise en service de l'entraînement ou du codeur. 	
	16#0040	PROFIdrive : le codeur est défaillant au niveau du bus (défaillance de station).	Vérifier le fonctionnement, les liaisons et les E/S de l'appareil.	
	16#0041	PROFIdrive : signes de vie du codeur défaillants		

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
16#800B		Dépassement de plage de la position		Retrait du déblocage
	16#0039	Dépassement de plage dans le sens positif	Référencer l'axe dans une plage de valeurs réelles valide.	
	16#003A	Dépassement de plage dans le sens négatif		
	16#003B	La modification de la position réelle est supérieure à la longueur modulo dans un cycle d'horloge du régulateur de position.	Adapter la longueur modulo au codeur utilisé.	
16#800C		Signalisation de défaut de l'entraînement		Retrait du déblocage
	16#0001	-	Vérifier le fonctionnement, les liaisons et les E/S de l'appareil.	
	16#003C	PROFIdrive : le signal de l'entraînement "Commande demandée" est défaillant.		
	16#003D	PROFIdrive : l'entraînement s'est mis à l'arrêt.		
	16#003E	PROFIdrive : l'entraînement est défaillant au niveau du bus (défaillance de station)		
	16#003F	PROFIdrive : signes de vie de l'entraînement défaillants	<ul style="list-style-type: none"> Vérifier le fonctionnement, les liaisons et les E/S de l'appareil. Comparer les paramètres d'horloge dans HW Config (ligne PROFIBUS, OM esclave pour entraînement ou codeur) à ceux du système exécutif. Tmapc et Servo doivent être configurés avec la même période d'horloge. 	
16#800D		L'écart de traînage admissible a été dépassé.		Retrait du déblocage
	16#0001	-	<ul style="list-style-type: none"> Vérifier la configuration de la boucle de régulation. Vérifier le signal de sens du codeur. Vérifier la configuration de la surveillance de l'écart de traînage. 	
16#800E		Défaut sur le fin de course matériel		Retrait du déblocage
	16#0042	Sens de dégagement inadmissible avec le fin de course matériel actif	Le sens de déplacement programmé est inhibé à cause du fin de course matériel actif. Dégager l'axe dans le sens opposé.	
	16#0043	Inversion de polarité du fin de course matériel, impossible de dégager l'axe.	Vérifier la configuration mécanique du fin de course matériel.	
	16#0044	Les deux fins de course matériels sont actifs, impossible de dégager l'axe.		

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
16#800F		Erreur dans la plage de destination		Retrait du déblocage
	16#0045	Plage de destination pas atteinte	La plage de destination n'a pas été atteinte dans le temps de tolérance de positionnement. <ul style="list-style-type: none"> • Vérifier la configuration de la surveillance de positionnement. • Vérifier la configuration de la boucle de régulation. 	
	16#0046	Plage de destination de nouveau quittée	La plage de destination a été quittée pendant la durée de séjour minimum. <ul style="list-style-type: none"> • Vérifier la configuration de la surveillance de positionnement. • Vérifier la configuration de la boucle de régulation. 	
16#8010		La position du fin de course logiciel inférieur est supérieure à celle du fin de course logiciel supérieur alors que l'axe n'est pas un axe modulo		Retrait du déblocage
	16#0001	-	Modifier la position des fins de course logiciels.	

Erreurs de fonctionnement sans arrêt de l'axe

ErrorID	ErrorInfo	Description	Solution
16#8200		L'axe n'est pas déblocué.	
	16#0001	-	Débloquer l'axe ; redémarrer la commande.
	16#003D	Affiché si un entraînement avec une connexion analogique s'est mis à l'arrêt.	Débloquer l'axe ; redémarrer la commande.
16#8201		L'axe a déjà été déblocué par une autre instance "MC_Power".	
	16#0001	-	Débloquer l'axe avec une seule instance de "MC_Power".
16#8202		Nombre maximum de commandes Motion Control simultanées dépassé (200 commandes max. pour le raccordement d'entraînement par PTO (Pulse Train Output), 100 commandes max. pour le raccordement d'entraînement par PROFIdrive/sortie analogique)	
	16#0001	-	Réduire le nombre de commandes actives simultanément ; redémarrer la commande. Une commande est active si le paramètre "Busy" a la valeur VRAI dans l'instruction Motion Control.
16#8203		L'axe fonctionne actuellement en "commande manuelle" (panneau de commande de l'axe).	
	16#0001	-	Quitter la "commande manuelle" ; redémarrer la commande.
16#8204		L'axe n'est pas référencé.	
	16#0001	-	Référencer l'axe avec l'instruction "MC_Home" ; redémarrer la commande.

ErrorID	ErrorInfo	Description	Solution
16#8205		L'axe est actuellement commandé par le programme utilisateur (l'erreur ne s'affiche que dans le panneau de commande de l'axe).	
	16#0013	L'axe est débloqué dans le programme utilisateur.	Bloquer l'axe avec l'instruction "MC_Power" et sélectionner à nouveau "Commande manuelle" dans le panneau de commande de l'axe.
16#8206		L'objet technologique n'est pas encore activé.	
	16#0001	-	Débloquer l'axe avec l'instruction "MC_Power" Enable = TRUE ou débloquer l'axe dans le panneau de commande de l'axe.
16#8207		Commande rejetée	
	16#0016	Référencement actif en cours ; un autre mode de référencement ne peut pas être démarré.	Attendre la fin du référencement actif ou l'annuler avec une commande de mouvement, par ex. "MC_Halt".
	16#0018	L'axe ne peut pas être déplacé avec une table de commandes pendant son référencement actif ou passif.	Attendre la fin du référencement direct ou passif.
	16#0019	Le référencement actif ou passif de l'axe n'est pas possible pendant le traitement d'une table de commandes.	Attendre la fin de la table de commandes ou l'annuler avec une commande de mouvement, par ex. "MC_Halt".
	16#0052	La position indiquée dépasse la limite numérique.	Indiquer une valeur de position valide dans l'instruction Motion Control.
	16#0053	L'axe est en cours de démarrage.	Attendre jusqu'à ce que l'axe soit prêt à fonctionner.
	16#0054	Valeur réelle invalide	Pour exécuter une commande "MC_Home", les valeurs réelles doivent être valides. Vérifier l'état des valeurs réelles. La variable de l'objet technologique "<nom de l'axe>.StatusSensor.State" doit afficher la valeur 2 (valide).
16#8208		La différence entre vitesse maximale et vitesse de démarrage/d'arrêt est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#000A	La valeur est inférieure ou égale à 0.	
16#8209		Accélération invalide pour l'objet technologique "Axe"	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#000A	La valeur est inférieure ou égale à 0.	
16#820A		Le redémarrage de l'axe est impossible.	
	16#0013	L'axe est débloqué dans le programme utilisateur.	Bloquer l'axe avec l'instruction "MC_Power", puis redémarrer.
	16#0027	L'axe fonctionne actuellement en "commande manuelle" (panneau de commande de l'axe).	Quitter la "commande manuelle" ; redémarrer.
	16#002C	L'axe n'est pas bloqué.	Bloquer l'axe ; redémarrer la commande.
	16#0047	L'objet technologique n'est pas prêt pour le redémarrage.	Recharger le projet.
	16#0048	La condition pour le redémarrage de l'objet technologique n'est pas remplie.	Bloquer l'objet technologique.

ErrorID	ErrorInfo	Description	Solution
16#820B		L'exécution de la table de commandes est impossible.	
	16#0026	La procédure de chargement à l'état MARCHE est en cours d'exécution.	Attendre que la procédure de chargement soit terminée.
16#820C		Pas de configuration disponible	
	16#0001	-	Erreur interne Contacter le service d'assistance.

Erreur de paramètre de bloc

ErrorID	ErrorInfo	Description	Solution
16#8400		La valeur du paramètre "Position" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#0005	La valeur se trouve en dehors de la plage numérique (supérieure à 1E+12).	
	16#0006	La valeur se trouve en dehors de la plage numérique (inférieure à 1E+12).	
16#8401		La valeur du paramètre "Distance" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#0005	La valeur se trouve en dehors de la plage numérique (supérieure à 1E+12).	
	16#0006	La valeur se trouve en dehors de la plage numérique (inférieure à 1E+12).	
16#8402		La valeur du paramètre "Velocity" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#0008	La valeur est supérieure à la vitesse maximale configurée.	
	16#0009	La valeur est inférieure à la vitesse de démarrage/d'arrêt configurée.	
	16#0024	La valeur est inférieure à 0.	
16#8403		La valeur du paramètre "Direction" de l'instruction Motion Control est invalide.	
	16#0011	La valeur de sélection est invalide.	Corriger la valeur de sélection ; redémarrer la commande.

ErrorID	ErrorInfo	Description	Solution
16#8404		La valeur du paramètre "Mode" de l'instruction Motion Control est invalide.	
	16#0011	La valeur de sélection est invalide.	Corriger la valeur de sélection ; redémarrer la commande.
	16#0015	Le référencement actif/passif n'est pas configuré.	Corriger la configuration et la charger dans l'automate ; débloquer l'axe et redémarrer la commande.
	16#0017	L'inversion de sens au fin de course matériel est activée bien que les fins de course matériels soient désactivés.	<ul style="list-style-type: none"> Activer les fins de course matériels au moyen de la variable "<nom de l'axe>.PositionLimitsHW.Active" = TRUE ; redémarrer la commande. Corriger la configuration et la charger dans l'automate ; débloquer l'axe et redémarrer la commande.
	16#0055	Mode invalide pour le codeur incrémental	Démarrer une prise de référence pour un codeur incrémental avec le paramètre "Mode" = 0, 1, 2, 3.
	16#0056	Mode invalide pour le codeur absolu	<p>Le référencement passif et le référencement actif ("Mode" = 2, 3) ne sont pas possibles pour un codeur absolu.</p> <p>Démarrer une prise de référence pour un codeur absolu avec le paramètre "Mode" = 0, 1.</p>
16#8405		La valeur du paramètre "StopMode" de l'instruction Motion Control est invalide.	
	16#0011	La valeur de sélection est invalide.	Corriger la valeur de sélection ; redémarrer l'axe.
16#8406		La marche par à-coups simultanément vers l'avant et l'arrière n'est pas autorisée.	
	16#0001	-	Empêcher que les paramètres "JogForward" et "JogBackward" soient à l'état logique TRUE simultanément ; redémarrer la commande.
16#8407		Le changement d'axe avec l'instruction "MC_Power" est autorisé uniquement après que l'axe actif a été bloqué.	
	16#0001	-	Bloquer l'axe actif ; il est alors possible de commuter sur l'autre axe et de le débloquer.
16#8408		La valeur du paramètre "Axis" de l'instruction Motion Control est invalide.	
	16#001A	La valeur indiquée ne correspond pas à la version requise de l'objet technologique.	Corriger la valeur ; redémarrer la commande.
	16#001B	La valeur indiquée ne correspond pas au type requis de l'objet technologique.	
	16#001C	La valeur indiquée n'est pas un bloc de données technologique Motion Control.	
16#8409		La valeur du paramètre "CommandTable" de l'instruction Motion Control est invalide.	
	16#001A	La valeur indiquée ne correspond pas à la version requise de l'objet technologique.	Corriger la valeur ; redémarrer la commande.
	16#001B	La valeur indiquée ne correspond pas au type requis de l'objet technologique.	
	16#001C	La valeur indiquée n'est pas un bloc de données technologique Motion Control.	

ErrorID	ErrorInfo	Description	Solution
16#840A		La valeur du paramètre "StartStep" de l'instruction Motion Control est invalide.	
	16#000A	La valeur est inférieure ou égale à 0.	Corriger la valeur ; redémarrer la commande.
	16#001D	Le pas de début est plus grand que le pas de fin.	
	16#001E	La valeur est supérieure à 32.	
16#840B		La valeur du paramètre "EndStep" de l'instruction Motion Control est invalide.	
	16#000A	La valeur est inférieure ou égale à 0.	Corriger la valeur ; redémarrer la commande.
	16#001E	La valeur est supérieure à 32.	
16#840C		La valeur du paramètre "RampUpTime" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#000A	La valeur est inférieure ou égale à 0.	
16#840D		La valeur du paramètre "RampDownTime" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#000A	La valeur est inférieure ou égale à 0.	
16#840E		La valeur du paramètre "EmergencyRampTime" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#000A	La valeur est inférieure ou égale à 0.	
16#840F		La valeur du paramètre "JerkTime" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#000A	La valeur est inférieure ou égale à 0.	
16#8410		La valeur du paramètre "Parameter" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#000B	Adresse invalide	
	16#0028	Le type de données du pointeur VARIANT "Parameter" et "Value" ne correspondent pas l'un à l'autre.	Utiliser un type de données approprié ; redémarrer la commande.
	16#0029	Le pointeur VARIANT "Parameter" ne désigne pas un bloc de données de l'objet technologique.	Corriger le pointeur VARIANT ; redémarrer la commande.
	16#002A	Impossible de lire la valeur dans le pointeur VARIANT "Parameter".	Corriger le pointeur VARIANT ; redémarrer la commande.
	16#002B	Impossible d'écrire la valeur dans le pointeur VARIANT "Parameter".	Corriger le pointeur VARIANT ou la valeur ; redémarrer la commande.
	16#002C	L'axe n'est pas bloqué.	Bloquer l'axe ; redémarrer la commande.
16#8411		La valeur du paramètre "Value" de l'instruction Motion Control est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Corriger la valeur ; redémarrer la commande.
	16#0005	La valeur se trouve en dehors de la plage numérique (supérieure à 1E+12).	
	16#0006	La valeur se trouve en dehors de la plage numérique (inférieure à 1E+12).	

Erreur de configuration de l'axe

ErrorID	ErrorInfo	Description	Solution
16#8600		Le paramétrage du générateur d'impulsions (PTO) est invalide.	
	16#000B	L'adresse est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#0014	Le matériel sélectionné est utilisé par une autre application.	
16#8601		Le paramétrage du compteur rapide (HSC) est invalide.	
	16#000B	L'adresse est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#0014	Le matériel sélectionné est utilisé par une autre application.	
16#8602		Le paramétrage de la "sortie de validation" est invalide.	
	16#000B	L'adresse est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#8603		Le paramétrage de "l'entrée Prêt" est invalide.	
	16#000B	L'adresse est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#8604		La valeur pour "Impulsions par tour du moteur" est invalide.	
	16#000A	La valeur est inférieure ou égale à zéro.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#8605		La valeur pour "Distance par tours du moteur" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#0005	La valeur se trouve en dehors de la plage numérique (supérieure à 1E+12).	
	16#000A	La valeur est inférieure ou égale à zéro.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	
16#8606		La valeur pour "Vitesse de démarrage/d'arrêt" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#0003	La valeur est supérieure à la limite supérieure du matériel.	
	16#0004	La valeur est inférieure à la limite inférieure du matériel.	
	16#0007	La vitesse de démarrage/d'arrêt est supérieure à la vitesse maximale.	

ErrorID	ErrorInfo	Description	Solution
16#8607		La valeur pour "Vitesse maximale" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". • Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". • Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0003	La valeur est supérieure à la limite supérieure du matériel.	
	16#0004	La valeur est inférieure à la limite inférieure du matériel.	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	
16#8608		La valeur pour "Accélération" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	• Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". • Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande. • Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". • Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0003	La valeur est supérieure à la limite supérieure du matériel.	
	16#0004	La valeur est inférieure à la limite inférieure du matériel.	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	
16#8609		La valeur pour "Décélération" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	• Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". • Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande. • Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". • Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0003	La valeur est supérieure à la limite supérieure du matériel.	
	16#0004	La valeur est inférieure à la limite inférieure du matériel.	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	

ErrorID	ErrorInfo	Description	Solution
16#860A		La valeur pour "Décélération d'arrêt d'urgence" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0003	La valeur est supérieure à la limite supérieure du matériel.	
	16#0004	La valeur est inférieure à la limite inférieure du matériel.	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	
16#860B		La valeur de la position du fin de course logiciel inférieur est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0005	La valeur se trouve en dehors de la plage numérique (supérieure à 1E+12).	
	16#0006	La valeur se trouve en dehors de la plage numérique (inférieure à 1E+12).	
	16#0030	La valeur de position du fin de course logiciel inférieur est supérieure à celle du fin de course logiciel supérieur.	
16#860C		La valeur de la position du fin de course logiciel supérieur est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande. Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0005	La valeur se trouve en dehors de la plage numérique (supérieure à 1E+12).	
	16#0006	La valeur se trouve en dehors de la plage numérique (inférieure à 1E+12).	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	
16#860D		L'adresse du fin de course matériel inférieur est invalide.	
	16#000B	Adresse invalide	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#000C	L'adresse du front descendant est invalide.	
	16#000D	L'adresse du front montant est invalide.	
16#860E		L'adresse du fin de course matériel supérieur est invalide.	
	16#000B	Adresse invalide	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#000C	L'adresse du front descendant est invalide.	
	16#000D	L'adresse du front montant est invalide.	

ErrorID	ErrorInfo	Description	Solution
16#860F		La valeur pour "Décalage du point de référence" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#0005	La valeur se trouve en dehors de la plage numérique (supérieure à 1E+12).	
	16#0006	La valeur se trouve en dehors de la plage numérique (inférieure à 1E+12).	<ul style="list-style-type: none"> Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8610		La valeur pour "Vitesse d'accostage" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#0008	La vitesse est supérieure à la vitesse maximale.	
	16#0009	La vitesse est inférieure à la vitesse de démarrage/d'arrêt.	<ul style="list-style-type: none"> Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8611		La valeur pour "Vitesse de référencement" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#0008	La vitesse est supérieure à la vitesse maximale.	
	16#0009	La vitesse est inférieure à la vitesse de démarrage/d'arrêt.	<ul style="list-style-type: none"> Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.

ErrorID	ErrorInfo	Description	Solution
16#8612		L'adresse de la came de référence est invalide.	
	16#000B	Adresse invalide	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#000C	L'adresse du front descendant est invalide.	
	16#000D	L'adresse du front montant est invalide.	
16#8613		L'inversion de sens au fin de course matériel est activée lors du référencement actif bien que les fins de course matériels ne soient pas configurés.	
	16#0001	-	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8614		La valeur pour "À-coup" est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#001F	La valeur est supérieure à l'à-coup maximal.	
	16#0020	La valeur est inférieure à l'à-coup minimal.	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8615		La valeur pour "Unité de mesure" est invalide.	
	16#0011	La valeur de sélection est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#8616		L'adresse de la came de référence est invalide (référencement passif à partir de V4).	
	16#0011	La valeur de sélection est invalide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.

ErrorID	ErrorInfo	Description	Solution
16#8617			
La valeur de la variable <nom de l'axe>.Sensor.Sensor[1].ActiveHoming.Mode est invalide.			
	16#0011	La valeur de sélection est invalide. (valeur valide : 2 = référencement par entrée TOR)	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8618			
La valeur de la variable <nom de l'axe>.Sensor.Sensor[1].PassiveHoming.Mode est invalide.			
	16#0011	La valeur de sélection est invalide. (valeur valide : 2 = référencement par entrée TOR)	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8619			
La valeur de la variable <nom de l'axe>.Actor.Type est invalide.			
	16#0011	La valeur de sélection est invalide. (valeur valide : 2 = connexion par interface d'impulsion)	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#861A			
La valeur pour "Sens de rotation autorisé" est invalide.			
	16#0011	La valeur de sélection est invalide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#002D	"Deux sens" n'est pas autorisé lorsque la sortie de sens est désactivée.	
16#861B			
Facteurs du réducteur principal incorrects			
	16#0031	La valeur est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#861C			
Combinaison interdite de données de référencement pour le codeur incrémental			
	16#0031	La valeur est invalide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.

ErrorID	ErrorInfo	Description	Solution
16#861D		Le mode de fixation paramétré pour le codeur est invalide. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].MountingMode	
	16#0011	La valeur de sélection est invalide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#861E		Le périmètre configuré pour la roue de mesure du codeur est invalide. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].Parameter.DistancePerRevolution	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#861F		La résolution configurée pour le codeur linéaire est incorrecte. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].Parameter.Resolution	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8620		La résolution fine paramétrée pour Gn_XIST1 est invalide. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].Parameter.FineResolutionXist1	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8621		La résolution fine paramétrée pour Gn_XIST1 dans <nom de l'axe>.Sensor.Sensor[1].Parameter.FineResolutionXist1 n'est pas compatible avec le réglage dans le paramètre PROFIdrive P979.	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8622		Valeur invalide pour la donnée de configuration <nom de l'axe>.Actor.Interface.AddressIn ou <nom de l'axe>.Actor.Interface.AddressOut	
	16#0011	La valeur de sélection est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".

ErrorID	ErrorInfo	Description	Solution
16#8623		La valeur paramétrée dans la variable <nom de l'axe>.Sensor.Sensor[1].Type est invalide.	
	16#0011	La valeur de sélection est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#8624		Le système de codeur paramétré est invalide. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].System	
	16#0011	La valeur de sélection est invalide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8625		Le paramètre de surveillance du positionnement est incorrect. Valeur invalide dans <nom de l'axe>.PositioningMonitoring.MinDwellTime	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8626		Le paramètre de surveillance du positionnement est incorrect. Valeur invalide dans <nom de l'axe>.PositioningMonitoring.Window	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8627		La configuration de l'interface PROFIdrive de la valeur réelle est incorrecte. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].Interface.AddressIn ou <nom de l'axe>.Sensor.Sensor[1].Interface.AddressOut	
	16#0011	La valeur de sélection est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#8628		Facteurs du régulateur incorrects	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<p>La valeur pour le gain ou la commande anticipatrice de la boucle de régulation est incorrecte.</p> <ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande. (<nom de l'axe>.PositionControl.Kv, <nom de l'axe>.PositionControl.Kpc)

ErrorID	ErrorInfo	Description	Solution
16#8629		La valeur limite pour le signal d'immobilisation est incorrecte. Valeur invalide dans <nom de l'axe>.StandStillSignal.VelocityThreshold	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#862A		Le paramètre de surveillance du positionnement est incorrect. Valeur invalide dans <nom de l'axe>.PositioningMonitoring.ToleranceTime	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#862B		Paramétrage PROFIBUS incohérent ; la somme de Ti et To est supérieure à un cycle DP.	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#862C		Le paramètre de surveillance de l'immobilisation est incorrect. Valeur invalide dans <nom de l'axe>.StandStillSignal.MinDwellTime	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#862D		Le paramètre de surveillance de l'écart de traînage est incorrect. Valeur invalide dans <nom de l'axe>.FollowingError.MinValue	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#862E		Valeur invalide pour la donnée de configuration <nom de l'axe>.Modulo.Length	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.

ErrorID	ErrorInfo	Description	Solution
16#862F		Valeur invalide pour la donnée de configuration <nom de l'axe>.Modulo.StartValue	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8630		Valeur invalide pour la donnée de configuration <nom de l'axe>.Actor.DriveParameter.ReferenceSpeed	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8631		La résolution fine paramétrée pour Gn_XIST2 est invalide. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].Parameter.FineResolutionXist2	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8632		Le nombre de tours de codeur déterminables est invalide. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].Parameter.DeterminableRevolutions	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8633		Le sens d'accostage de la came de référence indiqué pour le référencement passif est invalide. Valeur invalide dans <nom de l'axe>.Sensor.Sensor[1].PassiveHoming.Direction	
16#8634		Le paramètre de surveillance de l'écart de traînage est incorrect. Valeur invalide dans <nom de l'axe>.FollowingError.MaxValue	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.

ErrorID	ErrorInfo	Description	Solution
16#8635		Le paramètre de surveillance de l'écart de traînage est incorrect. Valeur invalide dans <nom de l'axe>.FollowingError.MinVelocity	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8636		Facteur du régulateur incorrect. Valeur invalide du facteur de commande anticipatrice <nom de l'axe>.PositionControl.Kpc	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
16#8637		Valeur invalide pour la donnée de configuration <nom de l'axe>.Sensor.Sensor[1].Interface.Type	
	16#0011	La valeur de sélection est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#8638		Valeur invalide pour la donnée de configuration <nom de l'axe>.Sensor.Sensor[1].Interface.HSC	
	16#0011	La valeur de sélection est invalide.	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
16#8639		Défaut sur l'entraînement	
	16#0049	Erreur de configuration sur l'appareil	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#004A	La technologie requiert une cadence d'asservissement plus faible.	Erreur système interne. Vérifier la cohérence du projet et le recharger dans l'automate.
	16#004B	Le pilote de l'appareil n'a pas été initialisé au démarrage.	Pour débloquent un objet technologique, le pilote de l'actionneur doit être initialisé. Réexécuter la commande plus tard.
16#863A		La communication avec l'entraînement est incorrecte.	
	16#004C	Erreur de configuration sur l'appareil	<ul style="list-style-type: none"> Raccorder un appareil approprié. Vérifier l'appareil (E/S). Comparer la configuration dans HW Config à celle de l'objet technologique.
	16#004D	Le pilote de l'appareil requiert une cadence d'asservissement plus faible.	<ul style="list-style-type: none"> Raccorder un appareil approprié. Vérifier l'appareil (E/S). Comparer la configuration dans HW Config à celle de l'objet technologique.
	16#004E	Erreur dans la communication interne avec l'appareil	Vérifier la cohérence du projet et le recharger dans l'automate.

ErrorID	ErrorInfo	Description	Solution
16#863B		Défaut sur le codeur	
	16#0049	Erreur de configuration sur l'appareil	Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power".
	16#004A	La technologie requiert une cadence d'asservissement plus faible.	Erreur système interne. Vérifier la cohérence du projet et le recharger dans l'automate.
	16#004B	Le pilote de l'appareil n'a pas été initialisé au démarrage.	Pour débloquer un objet technologique, le pilote de l'actionneur doit être initialisé. Réexécuter la commande plus tard.
16#863C		La communication avec le codeur est incorrecte.	
	16#004C	Erreur de configuration sur l'appareil	<ul style="list-style-type: none"> • Raccorder un appareil approprié. • Vérifier l'appareil (E/S). • Comparer la configuration dans HW Config à celle de l'objet technologique.
	16#004D	Le pilote de l'appareil requiert une cadence d'asservissement plus faible.	<ul style="list-style-type: none"> • Raccorder un appareil approprié. • Vérifier l'appareil (E/S). • Comparer la configuration dans HW Config à celle de l'objet technologique.
	16#004E	Erreur dans la communication interne avec l'appareil	Vérifier la cohérence du projet et le recharger dans l'automate.
16#863D		La communication avec l'appareil (entraînement ou codeur) est incorrecte.	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	<ul style="list-style-type: none"> • Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". • Corriger la valeur incorrecte en ligne, acquitter l'erreur avec l'instruction "MC_Reset" et, le cas échéant, redémarrer la commande.
	16#0055	L'adresse logique demandée est invalide.	<ul style="list-style-type: none"> • Raccorder un appareil approprié. • Vérifier l'appareil (E/S). • Vérifier la configuration topologique dans HW Config. • Comparer la configuration dans HW Config à celle de l'objet technologique.
	16#0056	L'adresse de sortie logique demandée est invalide.	
	16#0057	L'adresse de sortie logique demandée est invalide.	
16#863E		La valeur de la variable "ControlPanel.Input.TimeOut" est invalide (panneau de commande de l'axe).	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	Corriger la valeur dans les variables de l'objet technologique <nom de l'axe>.ControlPanel.Input.TimeOut. La valeur est indiquée en millisecondes (ms).

ErrorID	ErrorInfo	Description	Solution
16#863F		Valeur invalide pour la donnée de configuration <nom de l'axe>.Actor.DriveParameter.MaxSpeed	
	16#0030	La valeur a un format numérique incorrect ou se situe en dehors de la plage numérique valide.	Rectifier la valeur de référence dans l'entraînement et dans la configuration de l'objet technologique à Actuator.MaxSpeed/2. En cas de raccordement d'entraînement analogique, rectifier la valeur de référence dans l'entraînement et dans la configuration de l'objet technologique à Actuator.MaxSpeed/1,17.

Erreurs d'adaptation des données

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
16#8640		Erreurs lors de l'adaptation de la configuration d'actionneur		
	16#0030	La valeur n'a pas le format numérique attendu ou se situe en dehors de la plage numérique valide.	Redémarrage	Erreur d'adaptation
	16#0059	L'appareil n'est affecté à aucun groupe d'entraînement SINAMICS ou ne prend pas en charge les services nécessaires pour l'adaptation.	Redémarrage	Erreur d'adaptation
	16#005A	L'adaptation a été annulée en raison d'un manque de ressources.	Redémarrage	Erreur d'adaptation
	16#005B	L'adaptation n'est possible que si l'appareil a été câblé directement à une zone d'E/S.	Redémarrage	Erreur d'adaptation
	16#005C	Vitesse maximale (p1082) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
	16#005D	Couple/force maximum (p1520) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
	16#005E	Couple/force maximum (p1521) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
	16#005F	Limite de couple/force pour la résolution fine (p1544) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
	16#0060	Vitesse de base/vitesse de rotation nominale (p2000) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
	16#0061	Couple assigné/force assignée (p2003) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
16#8041	Erreurs lors de l'adaptation de la configuration de codeur			
	16#0030	La valeur n'a pas le format numérique attendu ou se situe en dehors de la plage numérique valide.	Redémarrage	Erreur d'adaptation
	16#005A	L'adaptation a été annulée en raison d'un manque de ressources.	Redémarrage	Erreur d'adaptation
	16#005B	L'adaptation n'est possible que si l'appareil a été câblé directement à une zone d'E/S.	Redémarrage	Erreur d'adaptation
	16#0059	L'appareil n'est affecté à aucun groupe d'entraînement SINAMICS ou ne prend pas en charge les services nécessaires pour l'adaptation.	Redémarrage	Erreur d'adaptation
	16#0062	Système de codeur (r0979[1/11].0) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
	16#0063	Résolution du codeur (r0979[2/12]) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
	16#0064	Résolution du codeur Gx_XIST1 (r0979[3/13]) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
	16#0065	Résolution du codeur Gx_XIST2 (r0979[4/14]) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
	16#0066	Nombre de tours de codeur discernables (r0979[5/15]) : paramètre inexistant ou sa valeur ne peut pas être lue ou se situe en dehors des limites autorisées. La lecture des paramètres a été annulée en raison d'une erreur signalée par le matériel.	Redémarrage	Erreur d'adaptation
16#8642		Configuration adaptée en interne		
	16#0067	1 : Valeur interdite pour Actor.MaxSpeed (vitesse Actor.MaxSpeed supérieure à 2 * vitesse Actor.ReferenceSpeed) ; solution : dans l'entraînement, définir P2000 = P1082 par exemple.	Réinitialisation	Erreur de configuration
16#8643		Incohérence entre objet technologique et configuration de l'entraînement		
	16#0068	Le type de télégramme configuré n'est pas compatible avec le type de télégramme de l'appareil (P922 ou P2079).	Réinitialisation	Erreur de configuration
	16#0069	La résolution du couple n'est pas compatible.	Réinitialisation	Erreur de configuration
	16#006A	La période élémentaire du cycle d'application maître n'est pas identique à celle de l'horloge servo.	Réinitialisation	Erreur de configuration
	16#006B	L'horloge de traitement de l'objet technologique n'est pas identique au cycle d'application de l'entraînement.	Réinitialisation	Erreur de configuration
	16#006C	Un module de fonction avec moteur linéaire est défini dans l'entraînement.	Réinitialisation	Erreur de configuration

ErrorID	ErrorInfo	Description	Solution	Réaction à l'erreur
16#8644		Incohérence entre objet technologique et configuration du codeur		
	16#0068	Le type de télégramme configuré n'est pas compatible avec le type de télégramme de l'appareil (P922 ou P2079).	Réinitialisation	Erreur de configuration
	16#006A	La période élémentaire du cycle d'application maître n'est pas identique à celle de l'horloge servo.	Réinitialisation	Erreur de configuration
	16#006B	L'horloge de traitement de l'objet technologique n'est pas identique au cycle d'application de l'entraînement.	Réinitialisation	Erreur de configuration
	16#006D	Le codeur sur l'entraînement n'est pas un codeur absolu (P979).	Réinitialisation	Erreur de configuration
16#8645		La vitesse maximale ne peut pas être atteinte avec les paramètres d'entraînement et d'axe définis		
	16#0001	Général	Réinitialisation	Erreur de configuration

Erreur de configuration de la table de commandes

ErrorID	ErrorInfo	Description	Solution
16#8700		La valeur pour "Type de commande" dans la table de commandes est invalide.	
	16#0001	-	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne et, le cas échéant, redémarrer la commande.
16#8701		La valeur pour "Position/course" dans la table de commandes est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne et, le cas échéant, redémarrer la commande.
	16#0005	La valeur se trouve en dehors de la plage numérique (supérieure à 1E+12).	
	16#0006	La valeur se trouve en dehors de la plage numérique (inférieure à 1E+12).	
16#8702		La valeur pour "Vitesse" dans la table de commandes est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne et, le cas échéant, redémarrer la commande.
	16#0008	La valeur est supérieure à la vitesse maximale configurée.	
	16#0009	La valeur est inférieure à la vitesse de démarrage/d'arrêt configurée.	
16#8703		La valeur pour "Durée" dans la table de commandes est invalide.	
	16#0002	La valeur n'est pas un nombre valide.	<ul style="list-style-type: none"> Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". Corriger la valeur incorrecte en ligne et, le cas échéant, redémarrer la commande.
	16#0021	La valeur est supérieure à 64 800 s.	
	16#0022	La valeur est inférieure à 0,001 s.	

ErrorID	ErrorInfo	Description	Solution
16#8704		La valeur pour "Pas suivant" dans la table de commandes est invalide.	
	16#0011	La valeur de sélection est invalide.	<ul style="list-style-type: none"> • Charger une configuration sans erreur dans l'automate ; débloquer de nouveau l'axe avec l'instruction "MC_Power". • Corriger la valeur incorrecte en ligne et, le cas échéant, redémarrer la commande.
	16#0023	La transition entre commandes n'est pas autorisée pour cette commande.	

Erreurs internes

ErrorID	ErrorInfo	Description	Solution
16#8FFF		Erreur interne	
	16#F0**	-	<p>Mise hors tension puis sous tension de la CPU. Si cela n'a pas l'effet voulu, contacter l'assistance clients. Tenir à disposition les informations suivantes :</p> <ul style="list-style-type: none"> • ErrorID • ErrorInfo • Entrées du tampon de diagnostic

Le S7-1200 offre plusieurs types de communication entre les CPU et les consoles de programmation, IHM et autres CPU.

ATTENTION

Si un pirate parvient à accéder physiquement aux réseaux, il peut lire et écrire les données.

TIA Portal, la CPU et les IHM (à l'exception de celles utilisant GET/PUT) utilisent une communication sécurisée qui protège contre les attaques Replay et par l'homme du milieu. Une fois que la communication est activée, l'échange des messages signés s'effectue en texte clair, ce qui permet au pirate de lire les données, mais qui protège contre l'accès aux données en écriture. C'est le TIA Portal, et non le processus de communication, qui crypte les données des blocs avec protection Know-how.

Toutes les autres formes de communication (échange I/O via PROFIBUS, PROFINET, AS-i, ou autre bus I/O, GET/PUT, blocs T et modules de communication (CM)) n'ont pas de fonction de sécurité. Vous devez les protéger en limitant leur accès physique. Si un pirate parvient à accéder physiquement aux réseaux par le biais de ces modes de communication, il peut lire et écrire les données.

Pour obtenir des informations et des recommandations sur la sécurité, voir nos "Recommandations d'opération pour la sécurité industrielle"

(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) sur le site de service et d'assistance de Siemens :

PROFINET

PROFINET est utilisé pour l'échange de données par le biais du programme utilisateur avec d'autres partenaires de communication via Ethernet :

- Dans le S7-1200, PROFINET prend en charge 16 périphériques ES avec 256 sous-modules au maximum, et PROFIBUS autorise 3 maîtres PROFIBUS DP indépendants, prenant en charge 32 esclaves par maître DP, avec 512 modules au maximum par maître DP.
- Communication S7
- User Datagram Protocol (UDP)
- ISO sur TCP (RFC 1006)
- Transport Control Protocol (TCP)

Contrôleur PROFINET IO

En tant qu'automate d'E/S utilisant PROFINET IO, la CPU communique avec 16 périphériques PN au plus sur le réseau PN local ou via un coupleur (link) PN/PN. Reportez-vous à PROFIBUS et PROFINET International, PI (www.profinet.com) pour plus d'informations.

PROFIBUS

PROFIBUS est utilisé pour l'échange de données par le biais du programme utilisateur avec d'autres partenaires de communication via le réseau PROFIBUS :

- Avec le CM 1242-5, la CPU fonctionne en tant qu'esclave PROFIBUS DP.
- Avec le CM 1243-5, la CPU fonctionne en tant que maître PROFIBUS DP de classe 1.
- Les esclaves PROFIBUS DP, maîtres PROFIBUS DP et interfaces AS-i (les trois modules de communication du côté gauche) et PROFINET sont des réseaux de communications indépendants qui ne se limitent pas les uns les autres.

AS-i

Le module CM 1243-2 maître AS-i S7-1200 permet le raccordement d'un réseau interface AC à une CPU S7-1200.

Communication CPU à CPU S7

Vous pouvez créer une connexion de communication sur une station partenaire et utiliser les instructions GET et PUT pour communiquer avec des CPU S7.

Communication TeleService

Avec TeleService via GPRS, une station d'ingénierie sur laquelle STEP 7 est installé communique via le réseau GSM et Internet avec une station SIMATIC S7-1200 munie d'un CP 1242-7. La liaison passe par un serveur TeleControl qui sert d'intermédiaire et qui est connecté à Internet.

IO-Link

Le maître S7-1200 SM 1278 4xIO-Link active des périphériques IO-Link pour se connecter à un S7-1200 CPU.

11.1 Liaisons de communication asynchrones

Vue d'ensemble des services de communication

La CPU accepte les services de communication suivants :

Service de communication	Fonctionnalité	Utilisation de PROFIBUS DP		Utilisation d'Ethernet
		Module maître DP CM 1243-5	Module esclave DP CM 1242-5	
Communication PG	Mise en service, test, diagnostic	Oui	Non	Oui
Communication HMI	Contrôle et surveillance par l'opérateur	Oui	Non	Oui
Communication S7	Échange de données à l'aide de connexions configurées	Oui	Non	Oui
Routage de fonctions PG	Par exemple, test et diagnostic au-delà des limites du réseau	Non	Non	Non
PROFIBUS DP	Échange de données entre maître et esclave	Oui	Oui	Non
PROFINET IO	Échange de données entre contrôleurs d'E/S et dispositifs d'E/S	Non	Non	Oui
Serveur Web	Diagnostic	Non	Non	Oui
SNMP ¹ (protocole simple de gestion de réseau)	Protocole standard de diagnostic et de paramétrage du réseau	Non	Non	Oui
Routage S7	En utilisant des tables de routage, des partenaires de communication peuvent communiquer avec chaque appareil, et ce même si ces appareils se trouvent dans des sous-réseaux S7 différents.	Non	Non	Oui
Communication ouverte via TCP/IP	Échange de données via Industrial Ethernet avec protocole TCP/IP (avec FB chargeables)	Non	Non	Oui
Communication ouverte via ISO sur TCP	Échange de données via Industrial Ethernet avec ISO sur protocole TCP (avec FB chargeables)	Non	Non	Oui
Communication ouverte via UDP	Échange de données via Industrial Ethernet avec protocole UDP (avec FB chargeables)	Non	Non	Oui

¹ La CPU prend en charge SNMP V1 sans interruptions ("traps").

Connexions disponibles

La CPU prend en charge le nombre suivant de liaisons de communication asynchrones simultanées maximum pour PROFINET et PROFIBUS. Le nombre maximum de ressources de connexion affectées à chaque catégorie est fixe ; vous ne pouvez pas modifier ces valeurs. Cependant vous pouvez configurer les 6 "Connexions libres disponibles" pour augmenter le nombre de n'importe quelle catégorie comme requis par votre application.

Ressources de liaison

	Ressources de la station		Ressources du module	
	Réservée	Dynami...	PLC_2 [CPU 1217C DC/DC/...	
Nombre maximum de ressources :	62	6	68	
	Maximum	Configurées	Configurées	Configurées
Communication PG :	4	-	-	-
Communication IHM :	12	0	0	0
Communication S7 :	8	0	0	0
Open User Communication :	8	0	0	0
Communication Web :	30	-	-	-
Autre communication :	-	-	0	0
Ressources utilisées en tout :	0	0	0	
Ressources disponibles :	62	6	68	

Remarque

Le nombre total de liaisons de communication S7-1200 n'augmente pas lorsque vous ajoutez des modules CM/CP.

En fonction des ressources de connexion affectées, le nombre suivant de connexions par dispositif est disponible :

	Console de programmation (PG)	Interface homme-machine (IHM)	Client/serveur GET/PUT	Communications ouvertes (Open User Communications)	Navigateur web
Nombre maximum de ressources de connexion	4 (prise en charge d'1 PG garantie)	12 (prise en charge de 4 dispositifs IHM garantie)	8	8	30 (prise en charge de 3 navigateurs Web garantie)

La CPU dispose, par exemple, de quatre ressources de connexion PG. Selon les fonctions PG actuellement utilisées, la PG peut utiliser en réalité une, deux, trois ou quatre de ses ressources de connexion disponibles. Si une PG utilise une ressource de connexion et qu'une autre PG en utilise trois pour leurs fonctions actuelles, il est possible d'utiliser deux PG simultanément. La prise en charge d'au moins une PG est toujours garantie.

Le nombre d'IHM représente un autre exemple, tel qu'illustré à la figure ci-dessous. Les IHM ont 12 ressources de connexion disponibles. Selon le type ou le modèle d'IHM que vous possédez et les fonctions IHM que vous utilisez, chaque IHM peut utiliser en réalité une, deux ou trois de ses ressources de connexion disponibles. Compte tenu du nombre de ressources de connexion disponibles utilisées, il peut s'avérer possible d'utiliser plus de quatre IHM en même temps. Toutefois, la prise en charge d'au moins quatre IHM est toujours garantie. Une IHM peut utiliser ses ressources de connexion disponibles (chacune trois au total) pour les fonctions suivantes :

- Lecture
- Ecriture
- Déclenchement d'alarme plus diagnostic

Il s'agit uniquement d'un exemple. Le nombre réel de liaisons utilisées peut varier selon le type et la version de l'IHM.

Exemple	IHM 1	IHM 2	IHM 3	IHM 4	IHM 5	Ressources de connexion disponibles totales
Ressources de connexion utilisées	2	2	2	3	3	12

Remarque

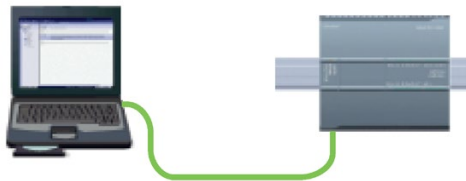
Connexions (HTTP) du serveur Web : La CPU fournit des liaisons pour plusieurs navigateurs web. Le nombre de navigateurs que la CPU peut prendre en charge simultanément dépend du nombre de connexions qu'un navigateur web donné demande/utilise.

Remarque

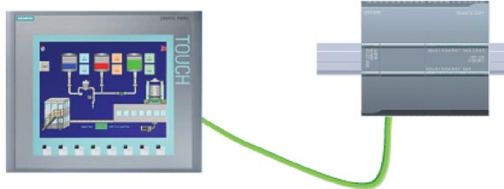
Les communications ouvertes, la liaison S7, l'IHM, la console de programmation et les liaisons de communication (HTTP) du serveur Web peuvent utiliser plusieurs ressources de connexion selon les fonctions utilisées.

11.2 PROFINET

La CPU peut communiquer avec d'autres CPU, avec des consoles de programmation, avec des appareils IHM et avec des appareils non Siemens utilisant des protocoles de communication TCP standard.



console de programmation connectée à la CPU



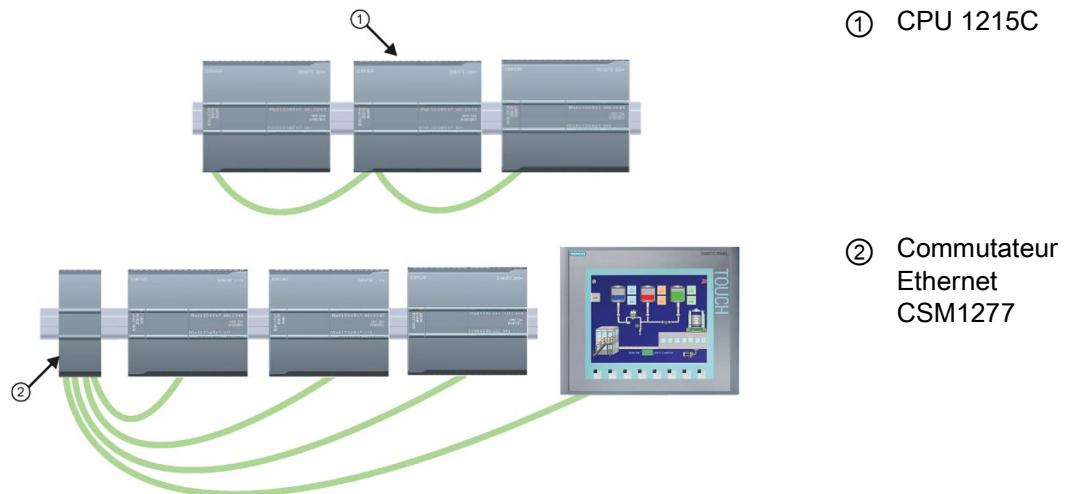
IHM connectée à la CPU



une CPU connectée à une autre CPU

Commutation Ethernet

Les CPU 1211C, 1212C et 1214C ont un port Ethernet unique et ne comprennent pas de commutateur Ethernet intégré. Une connexion directe entre une console de programmation ou une IHM et une CPU nécessite pas de commutateur Ethernet. Toutefois, un commutateur Ethernet est obligatoire pour un réseau comportant plus de deux CPU ou appareils IHM.

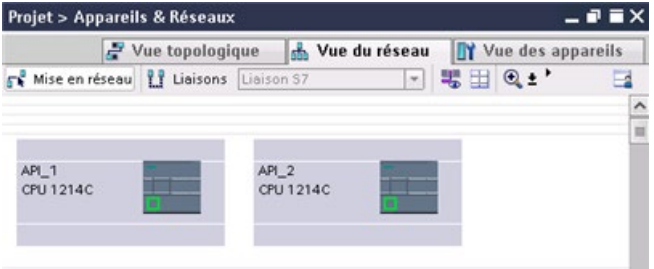
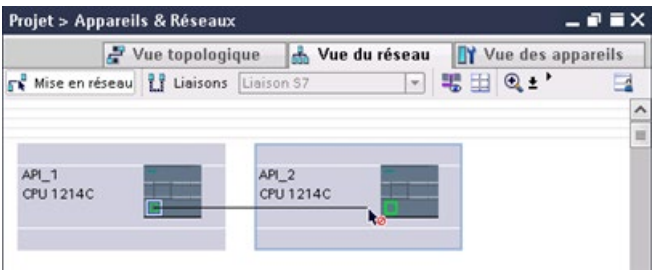
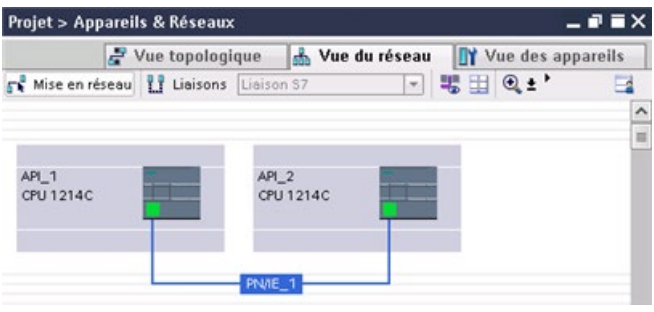


La CPU 1215C et la CPU 1217C comprennent un commutateur Ethernet intégré à 2 ports. Vous pouvez avoir un réseau avec une CPU 1215C et deux autres CPU S7-1200. Vous pouvez également utiliser le commutateur Ethernet à 4 ports CSM1277, monté sur châssis, pour connecter plusieurs CPU et appareils IHM.

11.2.1 Création d'une liaison réseau

Utilisez la "Vue du réseau" de la configuration des appareils pour créer les liaisons réseau entre les appareils dans votre projet. Une fois la liaison réseau créée, servez-vous de l'onglet "Propriétés" de la fenêtre d'inspection pour configurer les paramètres du réseau.

Tableau 11- 1 Création d'une liaison réseau

Action	Résultat
Sélectionnez la "Vue du réseau" pour afficher les appareils à connecter.	 <p>The screenshot shows the 'Projet > Appareils & Réseaux' window with the 'Vue du réseau' tab selected. Two devices, API_1 and API_2, are visible on the workspace. The interface includes a toolbar with options like 'Mise en réseau', 'Liaisons', and 'Liaison S7'.</p>
Sélectionnez le port sur un appareil et tracez la liaison vers le port sur le deuxième appareil.	 <p>The screenshot shows the same interface as the previous one, but with a mouse cursor hovering over a port on the second device (API_2). A faint line indicates the start of a connection being drawn from the first device.</p>
Relâchez le bouton de la souris pour créer la liaison réseau.	 <p>The screenshot shows the final result: a blue line representing the network link (PNNE_1) has been established between the two devices, API_1 and API_2.</p>

11.2.2 Configuration du routage local/partenaire

Une liaison appareil local / partenaire (distant) définit une affectation logique de deux partenaires de communication pour établir des services de communication. Une liaison définit les éléments suivants :

- Partenaires de communication concernés (un actif, un passif)
- Type de liaison (par exemple, liaison API, IHM ou d'appareil)
- Routage

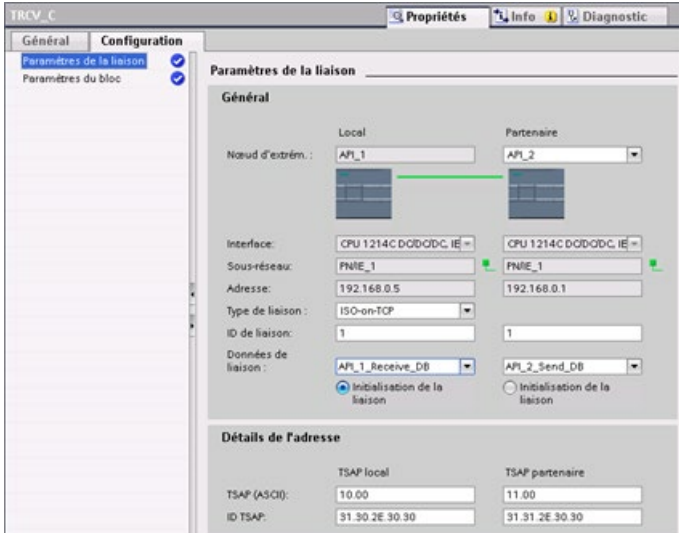
Les partenaires de communication exécutent les instructions pour configurer et établir la liaison de communication. Vous vous servez de paramètres pour indiquer les partenaires de nœud d'extrémité actif et passif. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.

Si la liaison est interrompue (en raison d'une coupure de ligne, par exemple), le partenaire actif tente de rétablir la liaison configurée. Vous n'avez pas besoin de réexécuter l'instruction de communication.

Routages

Après insertion d'une instruction TSEND_C, TRCV_C ou TCON dans le programme utilisateur, la fenêtre d'inspection affiche les propriétés de la liaison dès que vous sélectionnez une partie quelconque de l'instruction. Indiquez les paramètres de communication dans l'onglet "Configuration" des "Propriétés" de l'instruction de communication.

Tableau 11- 2 Configuration du routage (à l'aide des propriétés de l'instruction)

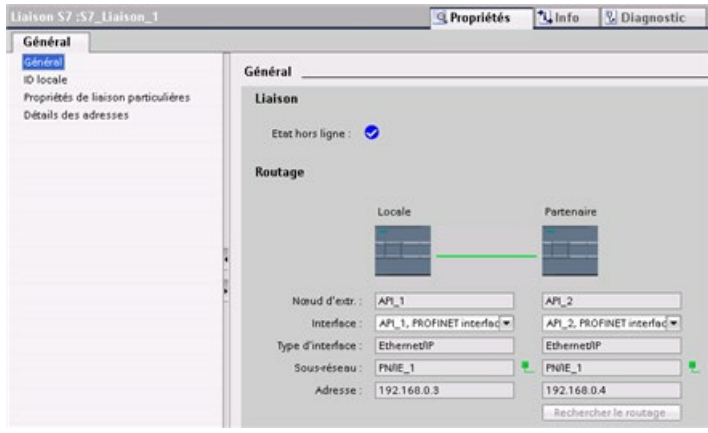
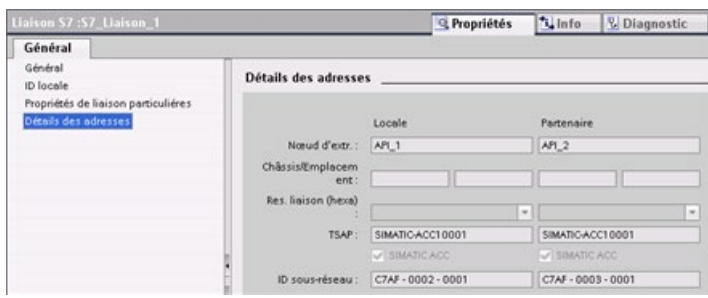
TCP, ISO sur TCP et UDP :	Propriétés de liaison
<p>Pour les protocoles Ethernet TCP, ISO sur TCP et UDP, utilisez les "Propriétés" de l'instruction (TSEND_C, TRCV_C ou TCON) pour configurer les liaisons "locale/partenaire".</p> <p>La figure montre les "Propriétés" d'une liaison ISO sur TCP dans l'onglet "Configuration".</p>	

Remarque

Lorsque vous configurez les propriétés de liaison pour une CPU, STEP 7 vous permet de sélectionner dans la CPU partenaire un DB de liaison spécifique (s'il existe) ou de créer le DB de liaison pour la CPU partenaire. La CPU partenaire doit avoir été préalablement créée pour le projet et ne peut pas être une CPU "non spécifiée".

Vous devez toujours insérer une instruction TSEND_C, TRCV_C ou TCON dans le programme utilisateur de la CPU partenaire. Lorsque vous insérez l'instruction, sélectionnez le DB de liaison qui a été créé par la configuration.

Tableau 11- 3 Configuration du routage pour la communication S7 (configuration des appareils)

Communication S7 (GET et PUT)	Propriétés de liaison
<p>Pour la communication S7, utilisez l'éditeur "Appareils & réseaux" du réseau pour configurer les liaisons locale/partenaire. Vous pouvez cliquer sur le bouton "Mis en valeur : Liaison" pour accéder aux "Propriétés".</p> <p>L'onglet "Général" fournit plusieurs propriétés :</p> <ul style="list-style-type: none"> • "Général" (montré) • "ID locale" • "Propriétés de liaison particulières" • "Détails de l'adresse" (montré) 	
	

Reportez-vous à "Protocoles" (Page 851) dans le chapitre "PROFINET" ou à "Création d'une liaison S7" (Page 1026) dans le chapitre "Communication S7" pour obtenir plus d'informations ainsi que la liste des instructions de communication disponibles.

Tableau 11- 4 Paramètres pour la liaison entre CPU

Paramètre		Définition
Adresse		Adresses IP affectées
Général	Noeud d'extrémité	Nom affecté à la CPU partenaire (réceptrice)
	Interface	Nom affecté aux interfaces
	Sous-réseau	Nom affecté aux sous-réseaux
	Type d'interface	<i>Communication S7 uniquement</i> : Type de l'interface
	Type de liaison	Type de protocole Ethernet
	ID de liaison	Numéro d'identification
	Données de liaison	Emplacement de stockage des données des CPU locale et partenaire
	Etablissement d'une liaison active	Bouton radio pour sélectionner la CPU locale ou partenaire en tant que liaison active
Détails de l'adresse	Noeud d'extrémité	<i>Communication S7 uniquement</i> : Nom affecté à la CPU partenaire (réceptrice)
	Châssis/emplacement	<i>Communication S7 uniquement</i> : Adresse du châssis et de l'emplacement
	Ressources de liaison	<i>Communication S7 uniquement</i> : Composant du TSAP utilisé lors de la configuration d'une liaison S7 avec une CPU S7-300 ou S7-400
	Port (décimal)	TCP et UDP : Port de la CPU partenaire en format décimal
	TSAP ¹ et ID de sous-réseau :	ISO sur TCP (RFC 1006) et communication S7 : TSAP des CPU locale et partenaire en formats ASCII et hexadécimal

¹ Lorsque vous configurez une liaison avec une CPU S7-1200 pour ISO sur TCP, utilisez uniquement des caractères ASCII dans l'extension TSAP pour les partenaires de communication passifs.

TSAP (points d'accès au service transport)

Grâce aux points d'accès au service transport (TSAP), le protocole ISO sur TCP et la communication S7 autorisent des liaisons multiples à une adresse IP unique. Les TSAP identifient de manière unique ces liaisons de noeud d'extrémité à une adresse IP.

Vous définissez les points TSAP à utiliser dans la section "Détails de l'adresse" de la boîte de dialogue "Paramètres de liaison". Vous indiquez le point TSAP d'une liaison dans la CPU dans la zone "TSAP local". Vous indiquez le point TSAP affecté à la liaison dans votre CPU partenaire dans la zone "TSAP partenaire".

Numéros de port

Avec les protocoles TCP et UDP, la configuration des paramètres de liaison de la CPU locale (active) doit indiquer l'adresse IP distante et le numéro de port de la CPU partenaire (passive).

Vous définissez les ports à utiliser dans la section "Détails de l'adresse" de la boîte de dialogue "Paramètres de liaison". Vous indiquez le port d'une liaison dans la CPU dans la zone "Port local". Vous indiquez le port affecté à la liaison dans votre CPU partenaire dans la zone "Port partenaire".

11.2.3 Affectation d'adresses IP (Internet Protocol)

11.2.3.1 Affectation d'adresses IP à des consoles de programmation et des dispositifs réseau

Si votre console de programmation utilise une carte adaptateur intégrée reliée à votre réseau local d'usine, la console de programmation et la CPU doivent toutes deux se trouver dans le même sous-réseau. Vous affectez le sous-réseau sous forme de combinaison de l'adresse IP et du masque de sous-réseau pour l'appareil. Adressez-vous à l'administrateur de votre réseau local si vous avez besoin d'aide.

L'ID de réseau correspond aux trois premiers octets de l'adresse IP, par exemple **211.154.184**.16. Cette ID de réseau identifie votre réseau IP de manière unique. Le masque de sous-réseau a normalement la valeur **255.255.255.0**. Toutefois, comme votre ordinateur se trouve dans un réseau local d'usine, le masque de sous-réseau peut avoir diverses valeurs (par exemple, **255.255.254.0**) pour permettre la configuration de sous-réseaux uniques. Lorsqu'il est combiné à l'adresse IP de l'appareil par une opération logique ET, le masque de sous-réseau définit les frontières d'un sous-réseau IP.

Remarque

Pour un scénario Internet où les consoles de programmation, les dispositifs réseau et les routeurs IP communiquent avec l'extérieur, vous devez affecter des adresses IP uniques afin d'éviter des conflits avec d'autres utilisateurs du réseau. Adressez-vous au personnel du service informatique de votre entreprise qui connaît vos réseaux d'usine pour l'affectation de vos adresses IP.

ATTENTION

Accès non autorisé à la CPU par le serveur Web

L'accès de personnes non autorisées à la CPU ou l'affectation de valeurs invalides aux variables API peut perturber le fonctionnement du processus, ce qui peut entraîner la mort, des blessures graves et des dommages matériels importants.

Comme la validation du serveur Web permet aux utilisateurs autorisés de changer l'état de fonctionnement, d'écrire dans des données de l'API et d'actualiser le firmware, Siemens vous recommande de mettre en œuvre la politique de sécurité suivante :

- N'autorisez l'accès au serveur Web qu'avec le protocole HTTPS.
- Protégez les ID utilisateur du serveur Web (Page 1042) avec un mot de passe fort. Les mots de passe forts contiennent au moins dix caractères, lettres, chiffres et caractères spéciaux mélangés, ne sont pas des mots du dictionnaire et ne sont pas des noms ou identifiants pouvant être déduits de vos données personnelles. Ne divulguez pas le mot de passe et changez-en fréquemment.
- N'étendez pas les droits minimum par défaut de l'utilisateur "Tous".
- Effectuez des recherches d'erreur et des vérifications de plage sur vos variables dans la logique de votre programme, car les utilisateurs des pages Web ont la possibilité de donner des valeurs invalides aux variables API.

Remarque

Une carte adaptateur de réseau secondaire est utile si vous ne voulez pas que votre CPU se trouve dans votre réseau d'entreprise. Cet agencement est particulièrement utile pendant le test initial et les tests de mise en service.

Affectation ou vérification de l'adresse IP de votre console de programmation via "Favoris réseau" (sur votre bureau)

Procédez comme suit pour affecter ou vérifier l'adresse IP de votre console de programmation :

1. Ouvrez le panneau de configuration à partir du menu Démarrer.
2. Ouvrez le "Centre Réseau et partage" et sélectionnez "Connexion au réseau local" pour l'adaptateur de réseau connecté à votre CPU.
3. Cliquez sur "Propriétés" dans la boîte de dialogue "État de Connexion au réseau local".
4. Dans la boîte de dialogue "Propriétés de Connexion au réseau local", sélectionnez "Protocole Internet version 4 (TCP/IPv4)" pour le champ "Cette connexion utilise les éléments suivants".
5. Cliquez sur le bouton "Propriétés".
6. Sélectionnez "Obtenir une adresse IP automatiquement" ou sélectionnez "Utiliser l'adresse IP suivante" pour saisir une adresse IP statique.
7. Si vous avez sélectionné "Utiliser l'adresse IP suivante", définissez l'adresse IP et le masque de sous-réseau :
 - Définissez l'adresse IP de manière à utiliser la même ID de réseau et le même sous-réseau que la CPU. Par exemple, si l'adresse IP de la CPU est **192.168.0.1**, vous pourriez définir l'adresse IP à **192.168.0.200**.
 - Sélectionnez **255.255.255.0** comme masque de sous-réseau.
 - N'indiquez rien pour "Passerelle par défaut".

Vous pouvez maintenant établir la connexion à la CPU.

Remarque

La carte d'interface réseau et la CPU doivent se trouver dans le même sous-réseau pour que STEP 7 trouve la CPU et puisse communiquer avec elle.

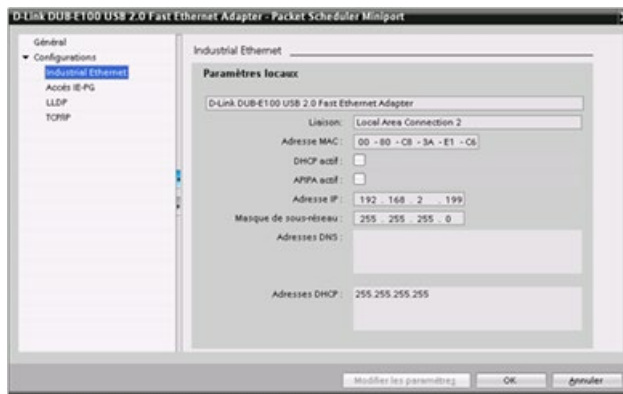
Votre service des technologies de l'information peut vous aider à configurer le réseau de façon que vous puissiez vous connecter à la CPU S7-1200.

11.2.3.2 Vérification de l'adresse IP de votre console de programmation

Vous pouvez vérifier les adresses MAC et IP de votre console de programmation à l'aide des sélections de menu suivantes :

1. Affichez le détail de "Accès en ligne" dans l'arborescence du projet.
2. Cliquez avec le bouton droit de la souris sur le réseau désiré et sélectionnez "Propriétés".
3. Dans la boîte de dialogue du réseau, affichez le détail de "Configurations" et sélectionnez "Industrial Ethernet".

Les adresses MAC et IP de la console de programmation s'affichent.



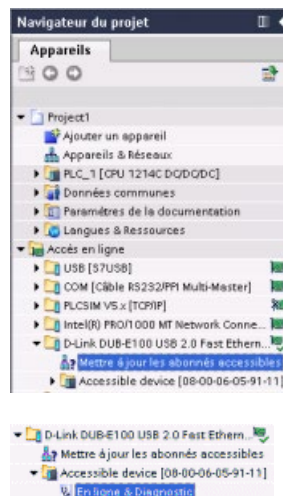
11.2.3.3 Affectation d'une adresse IP à une CPU en ligne

Vous pouvez affecter une adresse IP à un dispositif réseau en ligne. Cela s'avère particulièrement utile lors d'une configuration d'appareil initiale.

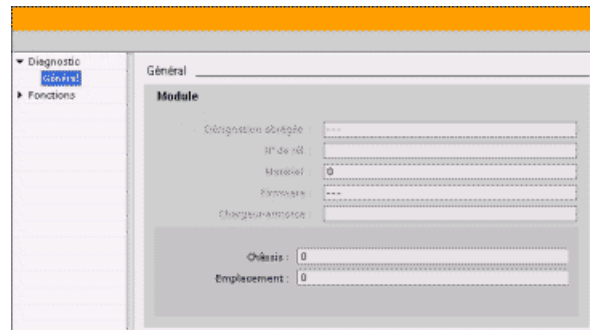
1. Vérifiez, dans l'arborescence du projet, que la CPU ne dispose pas d'une adresse IP configurée. Développez "Accès en ligne" > <carte adaptateur pour le réseau dans lequel se trouve l'appareil> et double-cliquez sur "Mettre à jour les abonnés accessibles".

Si STEP 7 affiche une adresse MAC et non une adresse IP, c'est qu'aucune adresse IP n'a été affectée.

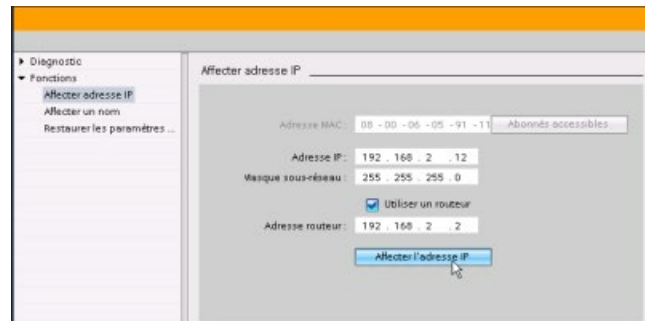
2. Doublez-cliquez sur "En ligne & Diagnostic" sous l'abonné accessible concerné.



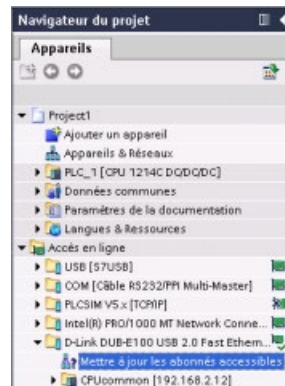
3. Sélectionnez "Fonctions" > "Affecter adresse IP" dans la boîte de dialogue "En ligne & Diagnostic".



4. Entrez votre nouvelle adresse IP dans le champ "Adresse IP" et cliquez sur le bouton "Affecter adresse IP".



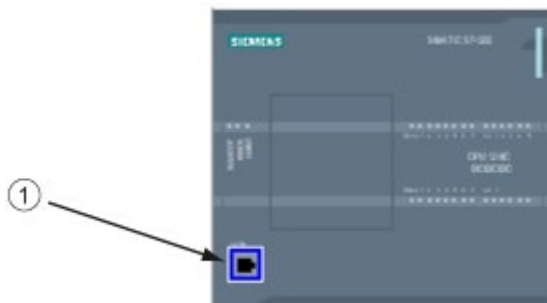
5. Vérifiez, dans l'arborescence du projet, que STEP 7 a affecté votre nouvelle adresse IP à la CPU. Double-cliquez sur "Mettre à jour les abonnés accessibles" pour afficher l'adresse IP que vous avez configurée.



11.2.3.4 Configuration d'une adresse IP pour une CPU dans votre projet

Configuration de l'interface PROFINET

Pour configurer les paramètres pour l'interface PROFINET, sélectionnez le carré PROFINET vert sur la CPU. L'onglet "Propriétés" de la fenêtre d'inspection affiche le port PROFINET.



① Port PROFINET

Configuration de l'adresse IP

Adresse Ethernet (MAC) : Dans un réseau PROFINET, chaque appareil reçoit une adresse MAC (Media Access Control : contrôle d'accès au support) du fabricant en vue de son identification. Une adresse MAC consiste en six groupes de deux chiffres hexadécimaux, séparés par un tiret (-) ou un deux-points (:), dans l'ordre de transmission (par exemple, 01-23-45-67-89-AB ou 01:23:45:67:89:AB).

Adresse IP : Chaque appareil doit également comporter une adresse IP (protocole Internet). Cette adresse permet à l'appareil de transmettre les données dans un réseau routé plus complexe.

Chaque adresse IP est divisée en quatre segments de 8 bits et est exprimée en format décimal avec séparation par des points (par exemple, 211.154.184.16). La première partie de l'adresse IP est utilisée pour l'ID de réseau (sur quel réseau est-on ?) et la deuxième partie de l'adresse correspond à l'ID d'hôte (unique pour chaque appareil dans le réseau). L'adresse IP 192.168.x.y est une désignation standard reconnue comme partie d'un réseau privé qui n'est pas routé sur Internet.

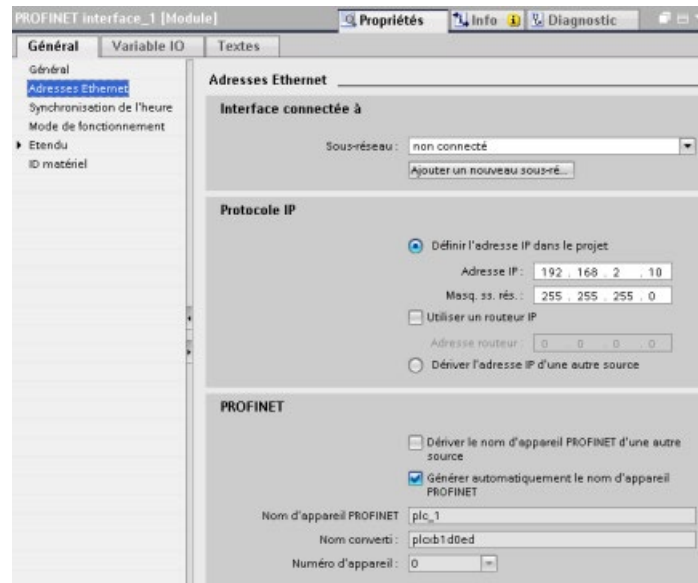
Masque de sous-réseau : Un sous-réseau est un groupement logique d'appareils réseau connectés. Les noeuds d'un sous-réseau tendent à être proches physiquement les uns des autres dans un réseau local. Un masque (appelé masque de sous-réseau ou masque de réseau) définit les frontières d'un sous-réseau IP.

Le masque de sous-réseau 255.255.255.0 convient généralement à un petit réseau local. Toutes les adresses IP dans ce réseau doivent avoir les trois mêmes premiers octets et les divers appareils dans ce réseau sont identifiés par le dernier octet (champ de 8 bits). Par exemple, on affectera le masque de sous-réseau 255.255.255.0 et les adresses IP 192.168.2.0 à 192.168.2.255 aux appareils dans un petit réseau local.

La seule liaison possible entre différents sous-réseaux se fait via un routeur. En présence de sous-réseaux, vous devez utiliser un routeur IP.

Routeur IP : Les routeurs constituent le lien entre réseaux locaux. Un ordinateur dans un réseau local peut envoyer par le biais d'un routeur des messages à d'autres réseaux qui peuvent abriter d'autres réseaux locaux. Si la destination des données ne se trouve pas à l'intérieur du réseau local, le routeur transmet les données à un autre réseau ou groupe de réseaux où les données peuvent être acheminées à destination.

Les routeurs se basent sur les adresses IP pour transmettre et recevoir des paquets de données.



Propriétés des adresses IP : Dans la fenêtre Propriétés, sélectionnez l'entrée de configuration "Adresses Ethernet". STEP 7 affiche la boîte de dialogue de configuration de l'adresse Ethernet qui associe le projet logiciel à l'adresse IP de la CPU qui recevra ce projet.

Tableau 11- 5 Paramètres pour l'adresse IP

Paramètre	Description	
Sous-réseau	Nom du sous-réseau auquel l'appareil est connecté. Cliquez sur le bouton "Ajouter nouveau sous-réseau" pour créer un nouveau sous-réseau. "non connecté" est la valeur par défaut. Deux types de connexion sont possibles : <ul style="list-style-type: none"> La valeur par défaut "non connecté" fournit une liaison locale. Un sous-réseau est nécessaire lorsque votre réseau contient au moins deux appareils. 	
Protocole IP	Adresse IP	Adresse IP affectée à la CPU
	Masque de sous-réseau	Masque de sous-réseau affecté
	Utiliser un routeur IP	Cochez cette case pour indiquer qu'un routeur IP est utilisé.
	Adresse du routeur	Adresse IP affectée au routeur, le cas échéant

Remarque

Toutes les adresses IP sont configurées lorsque vous chargez le projet dans la CPU. Si la CPU ne possède pas d'adresse IP préconfigurée, vous devez associer le projet à l'adresse MAC de l'appareil cible. Si votre CPU est connectée à un routeur sur un réseau, vous devez aussi entrer l'adresse IP du routeur.

Le bouton radio "Configurer l'adresse IP autrement" vous permet de modifier l'adresse IP en ligne ou à l'aide de l'instruction "T_CONFIG (Page 931)" après le chargement du programme. La méthode d'affectation de l'adresse IP concerne la CPU uniquement.

ATTENTION

Chargement d'une configuration matérielle avec l'option "Configurer l'adresse IP autrement"

Après chargement dans la CPU d'une configuration matérielle avec l'option "Configurer l'adresse IP autrement" activée, il n'est pas possible de faire passer la CPU de l'état de fonctionnement MARCHE à l'état ARRET ou de l'état ARRET à l'état MARCHE.

L'équipement utilisateur continue à fonctionner dans ces conditions et peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants en l'absence de précautions appropriées.

Vous devez donc veiller à que l'adresse ou les adresses IP de votre CPU soient définies avant d'utiliser la CPU dans un environnement d'automatisation réel. Pour ce faire, vous pouvez utiliser le progiciel de programmation STEP 7, l'outil SIMATIC Automation Tool ou un appareil IHM raccordé en combinaison avec l'instruction T_CONFIG.

ATTENTION

Situation dans laquelle le réseau PROFINET peut s'arrêter

La modification de l'adresse IP d'une CPU en ligne ou à partir du programme utilisateur peut créer une situation dans laquelle le réseau PROFINET est susceptible de s'arrêter.

Si l'adresse IP d'une CPU est changée en une adresse IP en dehors du sous-réseau, le réseau PROFINET perdra la communication et l'échange de données sera interrompu. L'équipement utilisateur pourrait être configuré pour continuer à fonctionner dans ces conditions. La perte de la communication PROFINET peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants en l'absence de précautions appropriées.

Si une adresse IP doit être modifiée à la main, veillez à ce que la nouvelle adresse IP se situe à l'intérieur du sous-réseau.

Configuration du port PROFINET

Par défaut, la CPU configure le ou les ports de l'interface PROFINET pour l'autonégociation. Pour que l'autonégociation fonctionne correctement, il faut activer cette option sur les deux stations. Si une station a une configuration fixe (par exemple, duplex intégral à 100 Mbit/s) et que l'autonégociation est activée sur l'autre station, l'autonégociation échouera et le fonctionnement se fera en semi-duplex.

Pour surmonter cette limitation de l'autonégociation, le S7-1200 fournit une option de désactivation de l'autonégociation. Lorsque vous désactivez l'autonégociation, le S7-1200 est automatiquement configuré en duplex intégral à 100 Mbit/s.

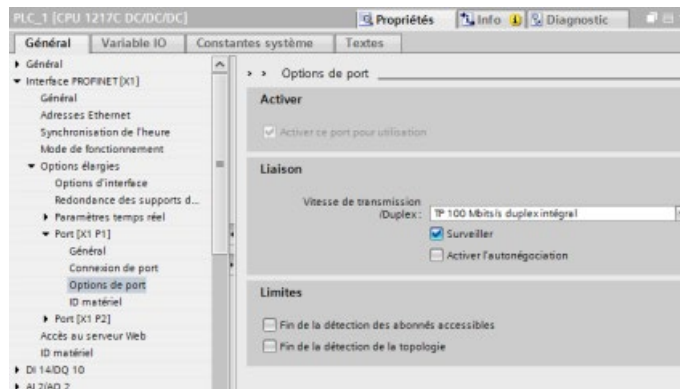
Vous pouvez donner à la vitesse de transmission et au mode duplex une valeur fixe pour chaque port :

1. Sélectionnez "Options élargies" et le port à configurer. Puis, sélectionnez "Options de port".
2. Dans le champ "Vitesse de transmission/Duplex" sous "Liaison", sélectionnez l'une des valeurs suivantes :
 - Automatique : La CPU et le partenaire déterminent la vitesse de transmission et le mode duplex du port par autonégociation.
 - TP 100 Mbit/s duplex intégral : Si vous désactivez l'autonégociation, le port fonctionne à 100 Mbit/s en mode duplex intégral. Si vous activez l'autonégociation, le port peut fonctionner à 100 Mbit/s duplex intégral ou à une autre combinaison vitesse de transmission/mode duplex autonégociée entre la CPU et le partenaire (un message est alors inscrit dans le tampon de diagnostic si l'option "Surveiller" est cochée ; voir ci-dessous).

3. Surveiller : Lorsque vous cochez cette case, un message est inscrit dans le tampon de diagnostic en présence de l'une des situations suivantes sur le port :
 - Impossible d'établir une liaison sur le port
 - Défaillance d'une liaison établie
 - Vous sélectionnez "TP 100 Mbit/s duplex intégral" dans le champ "Vitesse de transmission/Duplex" et la CPU établit par autonégociation une liaison avec une vitesse de transmission négociée différente de 100 Mbit/s ou semi-duplex comme mode duplex négocié.
4. Activer l'autonégociation : Si vous sélectionnez duplex intégral à 100 Mbit/s dans le champ "Vitesse de transmission/Duplex", vous pouvez désactiver l'autonégociation. Décochez la case "Activer l'autonégociation" pour désactiver l'autonégociation.

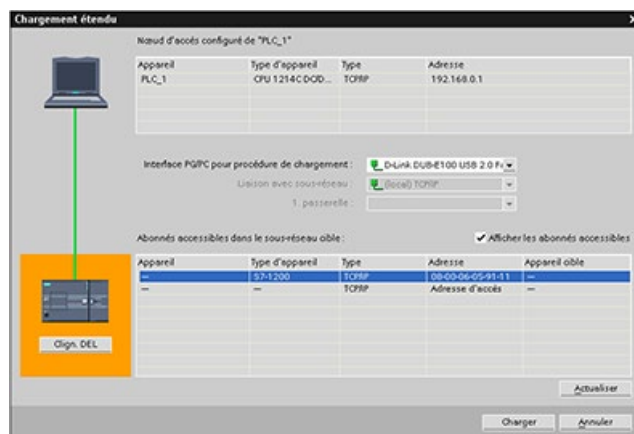
Remarque

Si vous ne désactivez pas l'autonégociation, la CPU et le partenaire négocient la vitesse de transmission et le mode duplex du port.



11.2.4 Test du réseau PROFINET

Une fois la configuration achevée, chargez le projet (Page 225) dans la CPU. Toutes les adresses IP sont configurées lorsque vous chargez le projet dans la CPU.



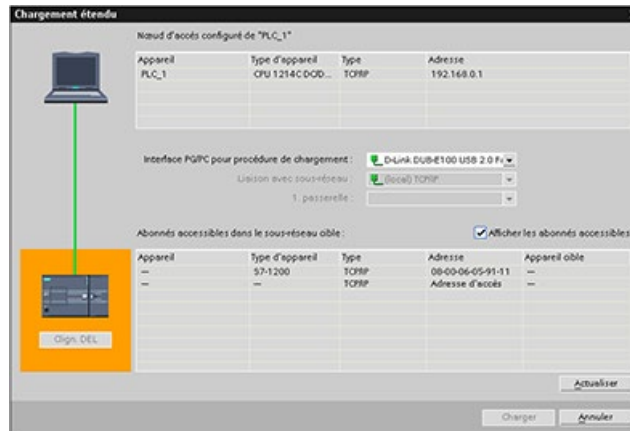
Affectation d'une adresse IP à un appareil en ligne

La CPU S7-1200 ne dispose pas d'une adresse IP préconfigurée. Vous devez affecter manuellement une adresse IP à la CPU :

- Pour affecter une adresse IP à un appareil en ligne, reportez-vous à "Configuration des appareils, Affectation d'une adresse IP à une CPU en ligne" (Page 838) où vous trouverez la marche à suivre étape par étape.
- Pour affecter une adresse IP dans votre projet, vous devez configurer l'adresse IP dans la configuration des appareils, sauvegarder la configuration et la charger dans l'automate. Reportez-vous à "Configuration des appareils, Configuration d'une adresse IP pour une CPU dans votre projet" (Page 840) pour plus d'informations.

Utilisation de la boîte de dialogue "Chargement élargi dans l'appareil" pour rechercher les dispositifs réseau connectés

La boîte de dialogue "Chargement élargi dans l'appareil" de la fonction "Charger dans l'appareil" de la CPU S7-1200 peut montrer tous les dispositifs réseau accessibles en précisant si des adresses IP uniques ont été affectées ou non à tous les appareils. Cochez la case "Afficher les abonnés accessibles" pour afficher tous les appareils accessibles et disponibles avec les adresses MAC ou IP qui leur ont été affectées.



Si le dispositif réseau requis n'est pas dans la liste, c'est que la communication avec ce dispositif a été interrompue pour une certaine raison. Vous devez alors rechercher d'éventuelles erreurs de matériel et/ou de configuration dans cet appareil et ce réseau.

11.2.5 Localisation de l'adresse Ethernet (MAC) sur la CPU

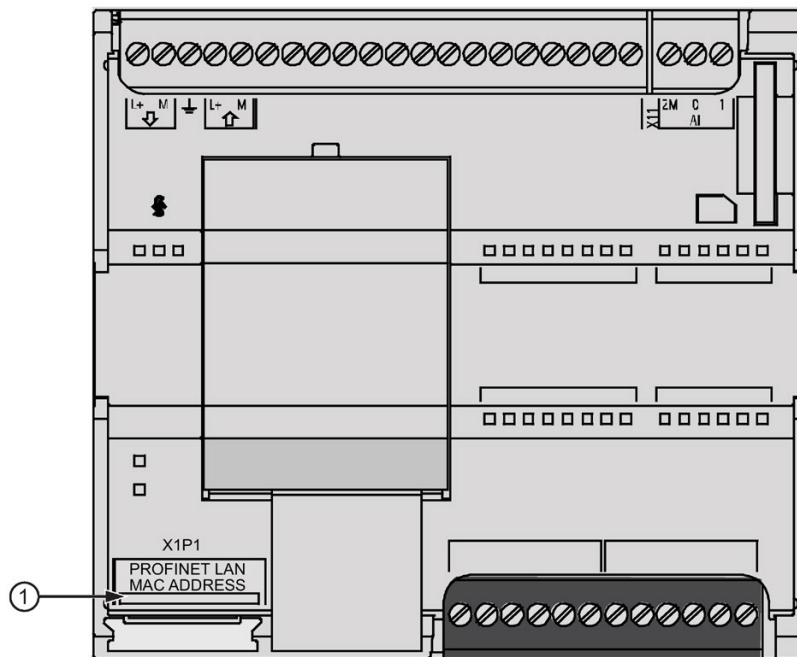
Dans les réseaux PROFINET, une adresse MAC (Media Access Control) est un identificateur que le fabricant attribue à l'interface de réseau aux fins d'identification. Une adresse MAC code généralement le numéro d'identification déposé du fabricant.

Le format standard (IEEE 802.3) pour l'impression des adresses MAC sous forme conviviale consiste en six groupes de deux chiffres hexadécimaux, séparés par un tiret (-) ou un deux-points (:), dans l'ordre de transmission (par exemple, 01-23-45-67-89-ab ou 01:23:45:67:89:ab).

Remarque

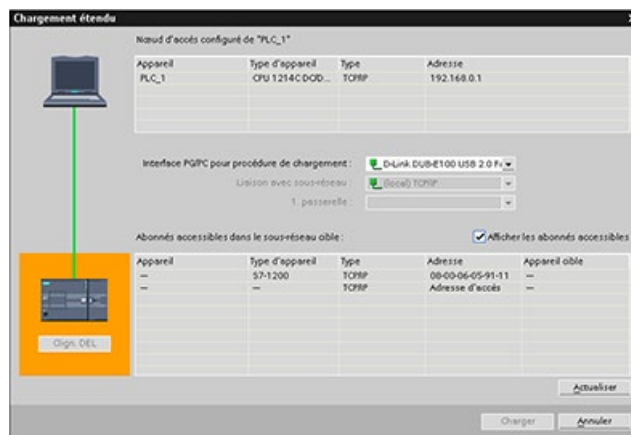
Chaque CPU reçoit en usine une adresse MAC unique permanente. Vous ne pouvez pas modifier l'adresse MAC d'une CPU.

L'adresse MAC est imprimée sur le coin inférieur gauche à l'avant de la CPU. Vous devez soulever le volet inférieur pour voir cette adresse MAC.



① Adresse MAC

Au départ, la CPU n'a pas d'adresse IP mais uniquement une adresse MAC reçue en usine. La communication PROFINET exige qu'une adresse IP unique soit affectée à tous les appareils.



Utilisez la fonction "Charger dans l'appareil" de la CPU et la boîte de dialogue "Chargement élargi dans l'appareil" pour visualiser tous les appareils réseau accessibles et vous assurer que des adresses IP uniques ont été affectées à tous les appareils. Cette boîte de dialogue affiche tous les appareils accessibles et disponibles avec les adresses MAC ou IP qui leur ont été affectées. Les adresses MAC sont capitales pour identifier les appareils auxquels manque l'adresse IP unique exigée.

11.2.6 Configuration de la synchronisation NTP

ATTENTION

Accès possible d'un pirate à vos réseaux par la synchronisation NTP

Si un pirate parvient à accéder à vos réseaux par la synchronisation NTP (Network Time Protocol), il aura la possibilité de perturber la commande de votre processus en décalant l'heure système de la CPU. De telles perturbations dans la commande du processus peuvent entraîner la mort, des blessures graves ou des dommages matériels.

La CPU S7-1200 désactive la fonction client NTP par défaut. Lorsque vous activez la fonction NTP, seules les adresses IP que vous configurez peuvent agir en tant que serveurs NTP. Vous devez configurer la fonction NTP pour permettre les corrections de l'heure système de la CPU à partir de serveurs distants.

La CPU S7-1200 prend en charge les alarmes horaires et les instructions d'horloge qui sont dépendantes de la précision de l'heure système de la CPU. Si vous configurez NTP et acceptez la synchronisation de l'heure par un serveur, vous devez vous assurer que le serveur est fiable. Vous risquez sinon une brèche de sécurité permettant à un utilisateur inconnu d'altérer la commande de votre processus par le décalage de l'heure système de la CPU.

Pour obtenir des informations et des recommandations sur la sécurité, voir nos "Recommandations d'opération pour la sécurité industrielle" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) sur le site de service et d'assistance de Siemens :

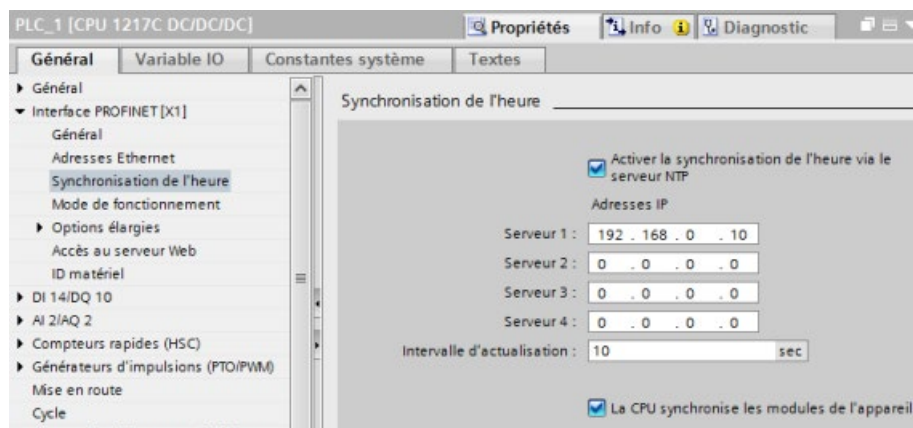
Le protocole NTP (Network Time Protocol) est largement utilisé pour synchroniser les horloges des systèmes informatiques avec des serveurs de temps Internet. En mode NTP, la CPU envoie des requêtes d'heure à intervalles réguliers (en mode client) au serveur NTP dans le sous-réseau (local). L'heure la plus précise et la plus fiable possible est calculée en fonction des réponses du serveur et l'horloge de la station est synchronisée.

L'avantage de ce mode est qu'il permet la synchronisation de l'heure au-delà des limites de sous-réseaux.

Vous configurez les adresses IP de quatre serveurs NTP au maximum. L'intervalle de mise à jour définit l'intervalle (en secondes) entre les requêtes d'heure. La valeur de l'intervalle doit être comprise entre 10 secondes et un jour.

En mode NTP, les serveurs transmettent généralement le temps UTC (temps universel coordonné) qui correspond à l'heure moyenne de Greenwich (GMT).

Dans la fenêtre Propriétés de la configuration d'appareil de la CPU, sélectionnez l'entrée de configuration "Synchronisation de l'heure". STEP 7 affiche la boîte de dialogue Synchronisation de l'heure :



Remarque

La CPU reçoit toutes les adresses IP lorsque vous chargez le projet.

Tableau 11- 6 Paramètres pour la synchronisation de l'heure

Paramètre	Définition
Activer la synchronisation de l'heure via le serveur NTP	Cochez cette case pour activer la synchronisation de l'heure au moyen du serveur NTP.
Serveur 1	Adresse IP affectée au serveur de temps réseau 1
Serveur 2	Adresse IP affectée au serveur de temps réseau 2
Serveur 3	Adresse IP affectée au serveur de temps réseau 3
Serveur 4	Adresse IP affectée au serveur de temps réseau 4
Intervalle d'actualisation de la synchronisation de l'heure	Valeur de l'intervalle (en s)
La CPU synchronise les modules de l'appareil.	Cochez cette case pour synchroniser l'horloge CP avec l'horloge de la CPU.

11.2.7 Temps de mise en route, affectation de nom et d'adresse pour un appareil PROFINET

PROFINET IO peut allonger le temps de mise en route de votre système (délai d'attente configurable). Un plus grand nombre d'appareils et des appareils lents influent sur la quantité de temps nécessaire pour passer à l'état MARCHE.

Dans la version V4.0 et ultérieures, votre réseau PROFINET S7-1200 peut comporter 16 périphériques PROFINET IO au maximum.

Chaque station (ou périphérique IO) démarre de manière indépendante à la mise en route, ce qui influe sur le temps de mise en route total de la CPU. Si vous donnez une valeur trop faible au délai d'attente configurable, le temps de mise en route pourrait ne pas être suffisant pour que toutes les stations mènent à bien leur démarrage. Si cette situation survient, des erreurs de station erronées seront signalées.

Vous trouverez le "temps d'affectation des paramètres pour la périphérie décentralisée" (délai d'attente) dans les propriétés de la CPU, sous "Mise en route". Le délai d'attente configurable par défaut est de 60 000 ms (1 minute) ; il est configurable par l'utilisateur.

Affectation de nom et d'adresse pour un appareil PROFINET dans STEP 7

Tous les appareils PROFINET **doivent** avoir un nom d'appareil et une adresse IP. Servez-vous de STEP 7 pour définir les noms d'appareils et configurer les adresses IP. Le téléchargement des noms d'appareils dans les périphériques IO se fait à l'aide de PROFINET DCP (Discovery and Configuration Protocol).

Affectation de l'adresse PROFINET au démarrage du système

Le contrôleur diffuse les noms des appareils sur le réseau et les appareils répondent avec leur adresse MAC. Le contrôleur affecte alors une adresse IP à l'appareil à l'aide du protocole DCP PROFINET :

- Si l'adresse MAC a une adresse IP configurée, la station exécute un démarrage.
- Si l'adresse MAC n'a pas d'adresse IP configurée, STEP 7 affecte l'adresse qui est configurée dans le projet et la station exécute alors un démarrage.
- En cas de problème lors de cette opération, une erreur de station survient et il n'y a pas de démarrage. Cette situation provoque le dépassement de la valeur de délai d'attente configurable.

11.2.8 Communication ouverte (Open User Communication)

11.2.8.1 Protocoles

Le port PROFINET intégré de la CPU prend en charge plusieurs normes de communication via un réseau Ethernet :

- Transport Control Protocol (TCP)
- ISO sur TCP (RFC 1006)
- User Datagram Protocol (UDP)

Tableau 11- 7 Protocoles et leurs instructions de communication respectives

Protocole	Exemples d'utilisation	Entrée des données dans la zone de réception	Instructions de communication	Type d'adressage
TCP	Communication CPU à CPU Transport de trames	Mode ad hoc	Uniquement TRCV_C et TRCV	Affecte des numéros de port aux appareils local (actif) et partenaire (passif).
		Réception de données avec longueur indiquée	TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV	
ISO sur TCP	Communication CPU à CPU Fragmentation et réassemblage des messages	Mode ad hoc	Uniquement TRCV_C et TRCV	Affecte des TSAP aux appareils local (actif) et partenaire (passif).
		Gestion par le protocole	TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV	
UDP	Communication CPU à CPU Communication du programme utilisateur	User Datagram Protocol (protocole de datagramme utilisateur)	TUSEND et TURCV	Affecte des numéros de port aux appareils local (actif) et partenaire (passif), mais ce n'est pas une liaison spécialisée.
Communication S7	Communication CPU à CPU Lecture/écriture de données depuis/vers une CPU	Emission et réception de données avec longueur indiquée	GET et PUT	Affecte des TSAP aux appareils local (actif) et partenaire (passif).
PROFINET IO	Communication CPU à périphérique PROFINET IO	Emission et réception de données avec longueur indiquée	Intégrées	Intégré

11.2.8.2 TCP et ISO sur TCP

TCP est un protocole standard décrit par RFC 793 : Transmission Control Protocol. L'objectif principal de TCP est de fournir un service de liaison sûr et fiable entre des paires de processus. Ce protocole présente les caractéristiques suivantes :

- Protocole de communication efficace car il est étroitement lié au matériel
- Convenant à des quantités moyennes à élevées de données (jusqu'à 8192 octets)
- Fournit considérablement plus de fonctions pour les applications, notamment la reprise sur erreur, le contrôle de flux et la fiabilité.
- Protocole orienté connexion
- Utilisable très souplement avec des systèmes tiers supportant exclusivement TCP
- Apte au routage
- Longueurs de données statiques uniquement
- Accusé de réception des messages
- Adressage des applications à l'aide de numéros de port
- La plupart des protocoles d'application utilisateur, tels que TELNET et FTP, utilisent TCP.
- Un effort de programmation est nécessaire pour la gestion des données en raison de l'interface de programmation EMISSION / RECEPTION.

ISO sur TCP (International Standards Organization on Transport Control Protocol ; RFC 1006) est un mécanisme qui permet aux applications ISO d'être portées sur le réseau TCP/IP. Ce protocole présente les caractéristiques suivantes :

- Protocole de communication efficace étroitement lié au matériel
- Convenant à des quantités moyennes à élevées de données (jusqu'à 8192 octets)
- Contrairement à TCP, les messages comportent une identification de fin de données et sont orientés messages.
- Apte au routage ; peut être utilisé dans des réseaux étendus
- Longueurs de données dynamiques possibles
- Un effort de programmation est nécessaire pour la gestion des données en raison de l'interface de programmation EMISSION / RECEPTION.

Grâce à des points d'accès au service transport (TSAP), le protocole TCP autorise plusieurs liaisons à une adresse IP unique (jusqu'à 64K liaisons). Avec RFC 1006, les TSAP identifient de manière unique ces liaisons de noeud d'extrémité à une adresse IP.

11.2.8.3 Services de communication et numéros de port utilisés

La CPU S7-1200 supporte les protocoles indiqués dans le tableau ci-dessous. Pour chaque protocole, la CPU affecte les paramètres d'adresse, la couche des communications respective ainsi que le rôle des communications et le sens des communications.

Ces informations permettent de faire correspondre les mesures de sécurité pour la protection du système d'automatisation concernant les protocoles utilisés (par exemple, pare-feu). Seuls les réseaux Ethernet ou PROFINET possèdent des mesures de sécurité. Puisque PROFIBUS ne possède aucune mesure de sécurité, le tableau n'inclut aucun protocole PROFIBUS.

Le tableau ci-dessous indique les différentes couches et protocoles que la CPU utilise :

Protocole	Numéro de port	(2) Couche de liaison (4) Couche de transport	Fonction	Description
Protocoles PROFINET				
DCP (Discovery and Configuration Protocol)	Sans objet	(2) Ethernet II et IEEE 802.1Q et Ethertype 0x8892 (PROFINET)	Découverte et configuration des appareils PROFINET accessibles	PROFINET utilise le DCP pour découvrir des appareils et fournir des paramètres de base. Le DCP utilise l'adresse MAC Multicast spéciale : xx-xx-xx-01-0E-CF, xx-xx-xx = Organizationally Unique Identifier
LLDP (Link Layer Discovery Protocol)	Sans objet	(2) Ethernet II et IEEE 802.1Q et Ethertype 0x88CC (PROFINET)	Protocole de découverte de couche de liaison PROFINET	PROFINET utilise le LLDP pour découvrir et gérer les rapports d'association entre les appareils PROFINET. Le LLDP utilise l'adresse MAC Multicast spéciale : 01-80-C2-00-00-0E

11.2.8.4 Mode ad hoc

Typiquement, TCP et ISO sur TCP reçoivent des paquets de données d'une longueur indiquée, comprise entre 1 et 8192 octets. Toutefois, les instructions de communication TRCV_C et TRCV fournissent aussi un mode de communication "ad hoc" qui permet de recevoir des paquets de données de longueur variable allant de 1 à 1472 octets.

Remarque

Si vous stockez les données dans un DB "optimisé" (symbolique uniquement), vous ne pouvez recevoir les données qu'en tableaux de types de données Byte, Char, USInt et SInt.

Pour activer le mode ad hoc de l'instruction TRCV_C ou TRCV, activez le paramètre d'entrée ADHOC de l'instruction.

Si vous n'appellez pas souvent l'instruction TRCV_C ou TRCV en mode ad hoc, vous recevrez peut-être plus d'un paquet lors d'un appel. Par exemple : Si vous deviez recevoir cinq paquets de 100 octets avec un appel, TCP transmettrait ces cinq paquets en un paquet de 500 octets alors que ISO-on-TCP restructurerait les paquets en cinq paquets de 100 octets.

11.2.8.5 ID de liaison pour les instructions Open User Communication

Lorsque vous insérez les instructions PROFINET TSEND_C, TRCV_C ou TCON dans votre programme utilisateur, STEP 7 crée un DB d'instance pour configurer la voie de communication (ou liaison) entre les appareils. Servez-vous des "Propriétés" (Page 833) de l'instruction pour configurer les paramètres de la liaison. L'ID de liaison pour cette liaison figure parmi ces paramètres.

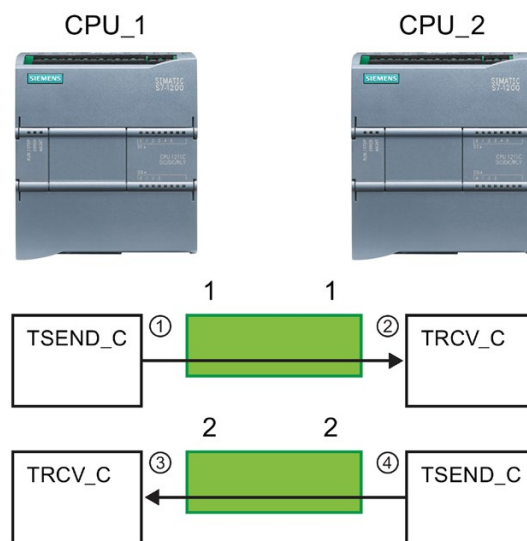
- L'ID de liaison doit être unique pour la CPU. Chaque liaison que vous créez doit avoir un DB et un ID de liaison différents.
- La CPU locale et la CPU partenaire peuvent toutes deux utiliser le même numéro d'ID de liaison pour la même liaison, mais les numéros d'ID de liaison ne doivent pas obligatoirement coïncider. Le numéro d'ID de liaison n'est significatif que pour les instructions PROFINET à l'intérieur du programme utilisateur de la CPU concernée.
- Vous pouvez utiliser n'importe quel nombre pour l'ID de liaison de la CPU. Toutefois, configurer les ID de liaison séquentiellement à partir de "1" constitue une méthode facile pour garder une trace du nombre de liaisons utilisées pour une CPU spécifique.

Remarque

Chaque instruction TSEND_C, TRCV_C ou TCON dans votre programme utilisateur crée une nouvelle liaison. Il est important d'utiliser l'ID de liaison correct pour chaque liaison.

L'exemple suivant montre la communication entre deux CPU qui utilisent deux liaisons distinctes pour émettre et recevoir des données.

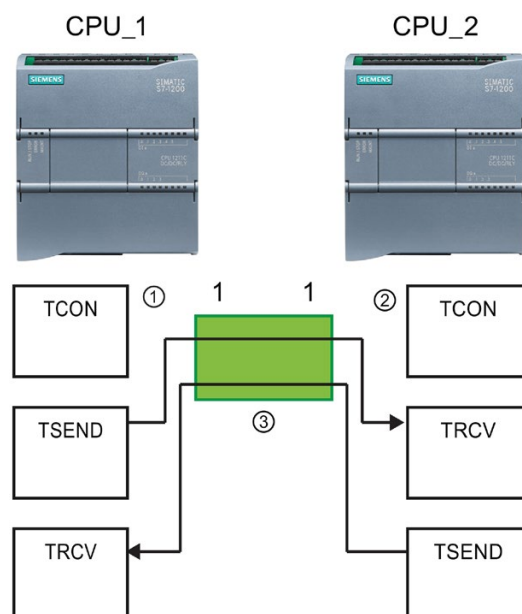
- L'instruction TSEND_C dans la CPU_1 se connecte à l'instruction TRCV_C dans la CPU_2 par le biais de la première liaison ("ID de liaison 1" à la fois sur la CPU_1 et la CPU_2).
- L'instruction TRCV_C dans la CPU_1 se connecte à l'instruction TSEND_C dans la CPU_2 par le biais de la deuxième liaison ("ID de liaison 2" à la fois sur la CPU_1 et la CPU_2).



- ① TSEND_C sur la CPU_1 crée une liaison et affecte un ID de liaison à cette liaison (ID de liaison 1 pour la CPU_1).
- ② TRCV_C sur la CPU_2 crée la liaison pour la CPU_2 et affecte l'ID de liaison (ID de liaison 1 pour la CPU_2).
- ③ TRCV_C sur la CPU_1 crée une deuxième liaison pour la CPU_1 et affecte un ID de liaison différent à cette liaison (ID de liaison 2 pour la CPU_1).
- ④ TSEND_C sur la CPU_2 crée une deuxième liaison et affecte un ID de liaison différent à cette liaison (ID de liaison 2 pour la CPU_2).

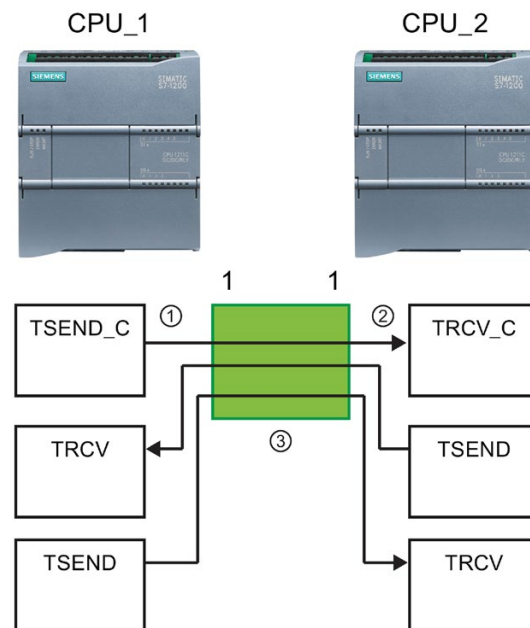
L'exemple suivant montre la communication entre deux CPU qui utilisent une même liaison pour émettre et recevoir des données.

- Chaque CPU utilise une instruction TCON pour configurer la liaison entre les deux CPU.
- L'instruction TSEND dans la CPU_1 se connecte à l'instruction TRCV dans la CPU_2 par le biais de l'ID de liaison ("ID de liaison 1") qui a été configurée par l'instruction TCON dans la CPU_1. L'instruction TRCV dans la CPU_2 se connecte à l'instruction TSEND dans la CPU_1 par le biais de l'ID de liaison ("ID de liaison 1") qui a été configurée par l'instruction TCON dans la CPU_2.
- L'instruction TSEND dans la CPU_2 se connecte à l'instruction TRCV dans la CPU_1 par le biais de l'ID de liaison ("ID de liaison 1") qui a été configurée par l'instruction TCON dans la CPU_2. L'instruction TRCV dans la CPU_1 se connecte à l'instruction TSEND dans la CPU_2 par le biais de l'ID de liaison ("ID de liaison 1") qui a été configurée par l'instruction TCON dans la CPU_1.



- ① TCON sur la CPU_1 crée une liaison et affecte un ID de liaison à cette liaison sur la CPU_1 (ID=1).
- ② TCON sur la CPU_2 crée une liaison et affecte un ID de liaison à cette liaison sur la CPU_2 (ID=1).
- ③ TSEND et TRCV sur la CPU_1 utilisent l'ID de liaison créé par l'instruction TCON sur la CPU_1 (ID=1). TSEND et TRCV sur la CPU_2 utilisent l'ID de liaison créé par l'instruction TCON sur la CPU_2 (ID=1).

Comme le montre l'exemple suivant, vous pouvez également utiliser des instructions TSEND et TRCV individuelles pour communiquer par le biais d'une liaison créée par une instruction TSEND_C ou TRCV_C. Les instructions TSEND et TRCV ne créent pas elles-mêmes de nouvelle liaison, de sorte que vous devez utiliser le DB et l'ID de liaison créés par une instruction TSEND_C, TRCV_C ou TCON.



- ① TSEND_C sur la CPU_1 crée une liaison et affecte un ID de liaison à cette liaison (ID=1).
- ② TRCV_C sur la CPU_2 crée une liaison et affecte un ID de liaison à cette liaison sur la CPU_2 (ID=1).
- ③ TSEND et TRCV sur la CPU_1 utilisent l'ID de liaison créé par l'instruction TSEND_C sur la CPU_1 (ID=1). TSEND et TRCV sur la CPU_2 utilisent l'ID de liaison créé par l'instruction TRCV_C sur la CPU_2 (ID=1).

11.2.8.6 Paramètres pour la liaison PROFINET

Les instructions TSEND_C, TRCV_C et TCON nécessitent les paramètres relatifs à la liaison pour que la liaison au partenaire soit établie. La structure TCON_Param affecte ces paramètres pour les protocoles TCP, ISO sur TCP et UDP. Typiquement, vous utilisez l'onglet "Configuration" (Page 833) des "Propriétés" de l'instruction pour fournir ces paramètres. Si l'onglet "Configuration" n'est pas accessible, vous devez fournir la structure TCON_Param dans les paramètres de l'instruction.

Avec V4.1, la structure TCON_IP_V4 affecte des paramètres pour le protocole TCP et la structure TCON_IP_RFC affecte des paramètres pour le protocole ISO sur TCP.

TCON_Param

Tableau 11- 8 Structure de la description de la liaison (TCON_Param)

Octet	Paramètre et type de données		Description
0 ... 1	block_length	UInt	Longueur : 64 octets (fixe)
2 ... 3	id	CONN_OUC (Word)	Référence à cette liaison : Plage de valeurs : 1 (valeur par défaut) à 4095. Indiquez la valeur de ce paramètre pour l'instruction TSEND_C, TRCV_C ou TCON sous ID.
4	connection_type	USInt	Type de liaison : <ul style="list-style-type: none"> • 17 : TCP (par défaut) • 18 : ISO sur TCP • 19 : UDP
5	active_est	Bool	ID pour le type de liaison : <ul style="list-style-type: none"> • TCP et ISO sur TCP : <ul style="list-style-type: none"> – FAUX : liaison passive – VRAI : liaison active (valeur par défaut) • UDP : FAUX
6	local_device_id	USInt	ID pour l'interface PROFINET ou Industrial Ethernet locale : 1 (valeur par défaut)
7	local_tsap_id_len	USInt	Longueur du paramètre local_tsap_id utilisé, en octets ; valeurs possibles : <ul style="list-style-type: none"> • TCP : 0 (actif, valeur par défaut) ou 2 (passif) • ISO sur TCP : 2 à 16 • UDP : 2
8	rem_subnet_id_len	USInt	Ce paramètre n'est pas utilisé.
9	rem_staddr_len	USInt	Longueur de l'adresse du noeud d'extrémité partenaire, en octets : <ul style="list-style-type: none"> • 0 : non précisé (paramètre rem_staddr sans objet) • 4 (valeur par défaut) : adresse IP valide dans le paramètre rem_staddr (uniquement pour TCP et ISO sur TCP)
10	rem_tsap_id_len	USInt	Longueur du paramètre rem_tsap_id utilisé, en octets ; valeurs possibles : <ul style="list-style-type: none"> • TCP : 0 (passif) ou 2 (actif, valeur par défaut) • ISO sur TCP : 2 à 16 • UDP : 0
11	next_staddr_len	USInt	Ce paramètre n'est pas utilisé.

Octet	Paramètre et type de données		Description
12 ... 27	local_tsap_id	Array [1..16] of Byte	<p>Elément d'adresse local de la liaison :</p> <ul style="list-style-type: none"> • TCP et ISO sur TCP : n° de port local (valeurs possibles : 1 à 49151 ; valeurs recommandées : 2000...5000) : <ul style="list-style-type: none"> – local_tsap_id[1] = octet de poids fort du numéro de port en notation hexadécimale – local_tsap_id[2] = octet de poids faible du numéro de port en notation hexadécimale – local_tsap_id[3-16] = sans objet • ISO sur TCP : ID TSAP local : <ul style="list-style-type: none"> – local_tsap_id[1] = B#16#E0 – local_tsap_id[2] = châssis et emplacement du noeud d'extrémité local (bits 0 à 4 : numéro d'emplacement, bits 5 à 7 : numéro de châssis) – local_tsap_id[3-16] = extension TSAP, facultative • UDP : Ce paramètre n'est pas utilisé. <p>Remarque : Veillez à ce que chaque valeur de local_tsap_id soit unique dans la CPU.</p>
28 ... 33	rem_subnet_id	Array [1..6] of USInt	Ce paramètre n'est pas utilisé.
34 ... 39	rem_staddr	Array [1..6] of USInt	<p>TCP et ISO sur TCP uniquement : Adresse IP du noeud d'extrémité partenaire (sans objet pour les liaisons passives). L'adresse IP 192.168.002.003 est, par exemple, stockée dans les éléments suivants du tableau :</p> <p>rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6]= sans objet</p>
40 ... 55	rem_tsap_id	Array [1..16] of Byte	<p>Elément d'adresse partenaire de la liaison :</p> <ul style="list-style-type: none"> • TCP : numéro de port partenaire. Plage : 1 à 49151 ; valeurs recommandées : 2000 à 5000) : <ul style="list-style-type: none"> – rem_tsap_id[1] = octet de poids fort du numéro de port en notation hexadécimale – rem_tsap_id[2] = octet de poids faible du numéro de port en notation hexadécimale – rem_tsap_id[3-16] = sans objet • ISO sur TCP : ID TSAP partenaire : <ul style="list-style-type: none"> – rem_tsap_id[1] = B#16#E0 – rem_tsap_id[2] = châssis et emplacement du noeud d'extrémité partenaire (bits 0 à 4 : numéro d'emplacement, bits 5 à 7 : numéro de châssis) – rem_tsap_id[3-16] = extension TSAP, facultative • UDP : Ce paramètre n'est pas utilisé.
56 ... 61	next_staddr	Array [1..6] of Byte	Ce paramètre n'est pas utilisé.
62 ... 63	spare	Word	Réservé : W#16#0000

TCON_IP_V4

Tableau 11- 9 Structure de la description de la liaison (TCON_IP_V4) : A utiliser avec TCP

Octet	Paramètre et type de données		Description
0 ... 1	InterfaceId	HW_ANY	Identifiant HW du sous-module d'interface IE
2 ... 3	ID	CONN_OUC (Word)	Référence à cette liaison : Plage de valeurs : 1 (valeur par défaut) à 4095. Indiquez la valeur de ce paramètre pour l'instruction TSEND_C, TRCV_C ou TCON sous ID.
4	ConnectionType	Byte	Type de liaison : <ul style="list-style-type: none"> • 11 : TCP/IP (par défaut) • 17 : TCP/IP (Ce type de liaison est inclus pour des raisons d'héritage. Il vous est recommandé d'utiliser "11 : TCP/IP (par défaut)".) • 19 : UDP
5	ActiveEstablished	Bool	Initialisation de la liaison active/passive : <ul style="list-style-type: none"> • VRAI : liaison active (valeur par défaut) • FAUX : liaison passive
	Adresse IP V4		
6	ADDR[1]	Byte	Octet 1
7	ADDR[1]	Byte	Octet 2
8	ADDR[1]	Byte	Octet 3
9	ADDR[1]	Byte	Octet 4
10 ... 11	RemotePort	UInt	Numéro de port distant UDP/TCP
12 ... 13	LocalPort	UInt	Numéro de port local UDP/TCP

TCON_IP_RFC

Tableau 11- 10 Structure de la description de la liaison (TCON_IP_RFC) : A utiliser avec ISO sur TCP

Octet	Paramètre et type de données		Description
0 ... 1	Interfaceld	HW_ANY	Identifiant HW du sous-module d'interface IE
2 ... 3	ID	CONN_OUC (Word)	Référence à cette liaison : Plage de valeurs : 1 (valeur par défaut) à 4095. Indiquez la valeur de ce paramètre pour l'instruction TSEND_C, TRCV_C ou TCON sous ID.
4	ConnectionType	Byte	Type de liaison : <ul style="list-style-type: none"> • 12 : ISO sur TCP (par défaut) • 17 : ISO sur TCP (Ce type de liaison est inclus pour des raisons d'héritage. Il vous est recommandé d'utiliser "12 : ISO sur TCP (par défaut)".)
5	ActiveEstablished	Bool	Initialisation de la liaison active/passive : <ul style="list-style-type: none"> • VRAI : liaison active (valeur par défaut) • FAUX : liaison passive
6 ... 7	En réserve		Inutilisé
Adresse IP V4			
8	ADDR[1]	Byte	Octet 1
9	ADDR[1]	Byte	Octet 2
10	ADDR[1]	Byte	Octet 3
11	ADDR[1]	Byte	Octet 4
Sélecteur de transport distant			
12 ... 13	TSelLength	UInt	Longueur de SélecteurT
14 ... 45	TSel	array [1..32] of Byte	Plage de caractères pour le nom TSAP
Sélecteur de transport local			
46 ... 47	TSelLength	UInt	Longueur de SélecteurT
48 ... 79	TSel	array [1..32] of Byte	Plage de caractères pour le nom TSAP

Voir aussi

CM/CP S7-1200 (<https://support.industry.siemens.com/cs/fr/fr/ps>)

11.2.8.7 Instructions TSEND_C et TRCV_C

À partir de la version V4.1 de la CPU S7-1200 avec STEP 7 V13 SP1, la CPU étend les fonctionnalités des instructions TSEND_C et TRCV_C en permettant l'utilisation de paramètres de liaison avec des structures selon "TCON_IP_v4" et "TCON_IP_RFC".

Pour cette raison, le S7-1200 accepte deux jeux d'instructions TSEND_C et TRCV_C :

- Instructions d'héritage TSEND_C et TRCV_C (Page 875): Ces instructions TSEND_C et TRCV_C existaient avant la version V4.0 du S7-1200 et fonctionnent uniquement avec les paramètres de liaison possédant des structures selon "TCON_Param".
- Instructions TSEND_C et TRCV_C (Page 863): Ces instructions TSEND_C et TRCV_C offrent toutes les fonctionnalités des instructions d'héritage, ainsi que la capacité à utiliser les paramètres de connexion possédant des structures selon "TCON_IP_v4" et "TCON_IP_RFC".

Sélection de la version des instructions TSEND_C et TRCV_C

Il existe deux versions des instructions TSEND_C et TRCV_C disponibles dans STEP 7 :

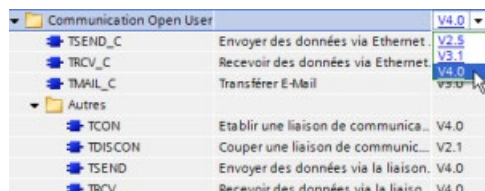
- Les versions 2.5 et 3.1 étaient disponibles dans STEP 7 Basic/Professional versions V13 ou antérieures.
- La version 4.0 est disponible dans STEP 7 Basic/Professional V13 SP1 ou ultérieur.

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

N'utilisez pas des versions d'instructions différentes dans le même programme CPU.



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.



Pour changer la version des instructions TSEND_C et TRCV_C, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction TSEND_C ou TRCV_C dans votre programme, une nouvelle instance de FB ou FC, en fonction de l'instruction TSEND_C ou TRCV_C sélectionnée, est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB ou FC dans l'arborescence du projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction TSEND_C ou TRCV_C dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB ou FC TSEND_C ou TRCV_C dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction TSEND_C ou TRCV_C.

TSEND_C et TRCV_C (Envoyer et recevoir des données via Ethernet)

L'instruction TSEND_C combine les fonctions des instructions TCON, TDISCON et TSEND . L'instruction TRCV_C combine les fonctions des instructions TCON, TDISCON et TRCV. Reportez-vous à "TCON, TDISCON, TSEND et TRCV (Page 885)" pour plus d'informations sur ces instructions.

La taille de données minimale que vous pouvez envoyer (TSEND_C) ou recevoir (TRCV_C) est d'un octet ; la taille maximale est de 8192 octets. TSEND_C n'accepte pas l'envoi de données à partir d'adresses booléennes et TRCV_C ne reçoit pas de données dans des adresses booléennes. Pour plus d'informations sur le transfert de données à l'aide de ces instructions, reportez-vous au paragraphe sur la cohérence des données (Page 204).



Remarque

Initialisation des paramètres de communication

Une fois que vous avez inséré l'instruction TSEND_C ou TRCV_C, utilisez les "Propriétés" de l'instruction (Page 833) pour configurer les paramètres de communication (Page 857). Lorsque vous entrez les paramètres pour les partenaires de communication dans la fenêtre d'inspection, STEP 7 entre les données correspondantes dans le DB pour l'instruction.

Si vous voulez utiliser un DB multi-instance, vous devez configurer le DB manuellement dans les deux CPU.

Tableau 11- 11 Instructions TSEND_C et TRCV_C

CONT/LOG	SCL	Description
	<pre>"TSEND_C_DB" (req:=_bool_in_, cont:=_bool_in_, len:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, connect:=_struct_inout_, data:=_variant_inout_, com rst:= bool_inout);</pre>	<p>TSEND_C établit une liaison de communication TCP ou ISO sur TCP à une station partenaire, envoie des données et peut mettre fin à la liaison. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.</p>
	<pre>"TRCV_C_DB" (en_r:=_bool_in_, cont:=_bool_in_, len:=_uint_in_, adhoc:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, rcvd_len=>_uint_out_, connect:=_struct_inout_, data:=_variant_inout_, com rst:= bool_inout);</pre>	<p>TRCV_C établit une liaison de communication TCP ou ISO sur TCP à une CPU partenaire, reçoit des données et peut mettre fin à la liaison. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 11- 12 Types de données pour les paramètres de TSEND_C et TRCV_C

Paramètre et type		Type de données	Description
REQ (TSEND_C)	IN	Bool	Lance la tâche d'émission en présence d'un front montant
EN_R (TRCV_C)	IN	Bool	Réception activée
CONT	IN	Bool	Contrôle la liaison de communication : <ul style="list-style-type: none"> • 0 : Déconnecter la liaison de communication après l'envoi des données. • 1 : Établir et conserver la liaison de communication Lors de l'envoi de données (TSEND_C) (front montant au niveau du paramètre REQ) ou de la réception de données (TRCV_C) (front montant au niveau du paramètre EN_R), le paramètre EN_R doit avoir la valeur TRUE afin d'établir ou de conserver une liaison.
LEN	IN	UDInt	Paramètre facultatif (masqué) Nombre maximum d'octets à envoyer (TSEND_C) ou à recevoir (TRCV_C) avec la tâche. Si, dans le paramètre DATA, vous utilisez des valeurs purement symboliques, le paramètre LEN doit avoir la valeur "0".
ADHOC (TRCV_C)	IN	Bool	Paramètre facultatif (masqué) Demande ad hoc pour le type de liaison TCP.
CONNECT	IN_OUT	TCON_Param	Pointeur sur la description de la liaison en fonction de la structure de la liaison à décrire : <ul style="list-style-type: none"> • Pour TCP ou UDP, utilisez la structure TCON_IP_v4 • Pour plus d'informations sur TCON_IP_v4, voir "Paramètres pour la liaison PROFINET" (Page 857). • Pour ISO-on-TCP, utilisez la structure TCON_IP_RFC • Pour plus d'informations sur TCON_IP_RFC, voir "Paramètres pour la liaison PROFINET" (Page 857). Le paramètre CONNECT n'est évalué qu'en cas de front montant sur REQ (TSEND_C), lorsque l'établissement de la liaison démarre (TRCV_C) ou lorsque COM_RST = "1".
DATA	IN_OUT	Variant	Pointeur sur la zone d'émission qui contient : <ul style="list-style-type: none"> • l'adresse et la longueur des données à envoyer (TSEND_C) • l'adresse et la longueur maximale des données reçues (TRCV_C).
ADDR	IN_OUT	Variant	Paramètre facultatif (masqué) Pointeur désignant l'adresse du destinataire pour le type de liaison UDP. Les informations d'adresse sont représentées dans la structure TADDR_Param ###.

Paramètre et type		Type de données	Description
COM_RST	IN_OUT	Bool	Paramètre facultatif (masqué) Redémarre l'instruction : <ul style="list-style-type: none"> 0 : Sans objet 1 : Redémarrage complet de l'instruction, par le biais duquel la liaison existante dépendante de CONT est soit interrompue, soit remise à 0 et rétablie. Le paramètre COM_RST est remis à 0 après l'évaluation de l'instruction TSEND_C ou TRCV_C et ne devrait donc pas être commuté statiquement.
DONE	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> 0 : La tâche d'émission n'a pas encore été démarrée ou elle est encore en cours d'exécution. 1 : Tâche d'émission exécutée sans erreur. Cet état s'affiche pour un cycle uniquement.
BUSY	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> 0 : La tâche d'émission n'a pas encore été démarrée ou elle est déjà achevée. 1 : La tâche d'émission n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche d'émission.
ERROR	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> 0 : Pas d'erreur 1 : Erreur survenue lors de l'établissement de la liaison, de l'émission de données ou de l'interruption de la liaison.
STATUS	OUT	Word	Etat de l'instruction (voir la description des paramètres ERROR et STATUS).
RCVD_LEN (TRCV_C)	OUT	Int	Quantité de données effectivement reçue (en octets).

Remarque

L'instruction TSEND_C requiert un front montant du paramètre d'entrée REQ pour démarrer une tâche d'émission. Le paramètre BUSY est mis à 1 pendant le traitement. L'achèvement de la tâche d'émission est signalé par la mise à 1 du paramètre DONE ou ERROR pendant un cycle. Pendant ce temps, aucun front montant du paramètre d'entrée REQ n'est pris en compte.

Remarque

Avec la valeur par défaut du paramètre LEN (LEN = 0), le paramètre DATA est utilisé pour déterminer la longueur des données transmises. Il est recommandé que les données envoyées par l'instruction TSEND_C aient la même taille que le paramètre DATA de l'instruction TRCV_C.

Si, en cas d'utilisation de la valeur par défaut du paramètre LEN, il s'avère nécessaire d'envoyer des données en segments plus petits que la taille du paramètre DATA, la règle suivante s'applique. Si la taille des données envoyées par TSEND_C est différente de la taille du paramètre TRCV_C DATA, TRCV_C reste à l'état occupé (code d'état : 7006) jusqu'à ce que la taille totale des données envoyées par TSEND_C soit égale à la taille du paramètre TRCV_C DATA.

Le tampon du paramètre TRCV_C DATA n'affiche pas les nouvelles données reçues tant que leur taille n'est pas égale à celle du tampon du paramètre DATA.

Fonctionnement de TSEND_C

L'instruction TSEND_C est exécutée de manière asynchrone et exécute tour à tour les fonctions suivantes :

1. Créer et établir une liaison de communication :

TSEND_C définit une liaison de communication et l'établit lorsqu'un front montant est détecté sur le paramètre REQ et qu'aucune liaison de communication n'a déjà été établie. Une fois la connexion définie et établie, elle est maintenue et surveillée automatiquement par la CPU. La définition de la liaison de communication s'effectue avec la description de la liaison fournie dans le paramètre CONNECT. Il est possible d'utiliser les types de liaison suivants :

- Structure TCON_Param pour les protocoles TCP, ISO sur TCP et UDP
- Avec V4.1, TCP/UDP : Description de la liaison via la structure TCON_IP_v4 dans le paramètre CONNECT
- Avec V4.1, ISO sur TCP : Description de la liaison via la structure TCON_IP_RFC dans le paramètre CONNECT

Lorsque la CPU passe à l'état ARRET, une liaison existante est interrompue et la liaison configurée est supprimée. Vous devez réexécuter TSEND_C pour reconfigurer et rétablir la liaison. Le nombre de liaisons de communication possibles est précisé dans les caractéristiques techniques de votre CPU.

2. Envoyer des données via la liaison de communication existante :

La tâche d'émission est exécutée en cas de détection d'un front montant dans le paramètre REQ. Comme décrit ci-dessus, la liaison de communication a été établie au préalable. Vous indiquez la zone d'émission dans le paramètre DATA. Celui-ci contient l'adresse et la longueur des données à envoyer. Dans le paramètre DATA, n'utilisez aucune zone de données ayant le type de données BOOL ou Array of BOOL. La définition du nombre maximum d'octets à émettre avec une tâche d'émission est effectuée dans le paramètre LEN. En cas d'utilisation d'un nom symbolique dans le paramètre DATA, le paramètre LEN devrait avoir la valeur "0".

Aucune édition des données à émettre ne doit être réalisée avant l'exécution complète de la tâche d'émission.

3. Interrompre la liaison de communication :

La liaison de communication est interrompue après l'envoi des données, si le paramètre CONT avait la valeur "0" pendant le front montant dans le paramètre REQ. Si tel n'est pas le cas, la liaison de communication est maintenue.

Si l'exécution de la tâche d'émission réussit, le paramètre DONE est mis à "1". Le cas échéant, la liaison de communication est interrompue au préalable (voir ci-dessus la description de la dépendance du paramètre CONT). L'état de signal "1" du paramètre DONE ne confirme pas que les données émises ont déjà été lues par le partenaire de communication.

A la mise à "1" du paramètre COM_RST, TSEND_C est remis à "0", Ce qui peut conduire à une perte de données en cas de transfert de données à cet instant.

Les cas suivants sont possibles en fonction du paramètre CONT :

- CONT = "0" :
Une liaison de communication existante est établie.
- CONT = "1" et une liaison de communication était établie :
Une liaison de communication existante est remise à 0 et rétablie.
- CONT = "1" et aucune liaison de communication n'était établie :
Aucune liaison de communication n'est établie.

Le paramètre COM_RST est remis à 0 après l'évaluation par l'instruction T_SEND. Pour réactiver TSEND_C après l'exécution (DONE = 1), appelez une fois l'instruction avec REQ = 0

Fonctionnement de TRCV_C

L'instruction TRCV_C est exécutée de manière asynchrone et exécute tour à tour les fonctions suivantes :

1. Créer et établir une liaison de communication :

TRCV_C définit une liaison de communication et l'établit lorsque le paramètre EN_R = "1" et qu'aucune liaison de communication n'a déjà été établie. Une fois la connexion définie et établie, elle est maintenue et surveillée automatiquement par la CPU.

La définition de la liaison de communication s'effectue avec la description de la liaison fournie dans le paramètre CONNECT. Il est possible d'utiliser les types de liaison suivants :

- Structure TCON_Param pour les protocoles TCP, ISO sur TCP et UDP
- Avec V4.1, TCP / UDP : Description de la liaison via la structure TCON_IP_v4 dans le paramètre CONNECT
- Avec V4.1, ISO sur TCP : Description de la liaison via la structure TCON_IP_RFC dans le paramètre CONNECT

Lorsque la CPU passe à l'état ARRET, une liaison existante est interrompue et la liaison configurée est supprimée. Vous devez réexécuter TRCV_C pour reconfigurer et rétablir la liaison avec EN_R = "1".

Si EN_R est mis à "0" avant l'établissement de la liaison de communication, celle-ci est établie et le reste, même si CONT = "0". Cependant, aucune donnée ne sera reçue (DONE reste à "0").

Le nombre de liaisons de communication possibles est précisé dans les caractéristiques techniques de votre CPU.

2. Recevoir des données via la liaison de communication existante :

La réception des données est activée lorsque le paramètre EN_R est mis à la valeur "1". Comme décrit ci-dessus, la liaison de communication a été établie au préalable. Les données reçues sont inscrites dans une plage de réception. La définition de la longueur de la plage de réception s'effectue, selon la variante de protocole utilisée, soit avec le paramètre LEN (si LEN <> 0) soit en indiquant la longueur du paramètre DATA (si LEN = 0). Si, dans le paramètre DATA, vous utilisez des valeurs purement symboliques, le paramètre LEN doit avoir la valeur "0".

Si EN_R est mis à "0" avant la première réception de données, la liaison de communication est maintenue, même si CONT = 0. Cependant, aucune donnée n'est reçue (DONE reste à "0").

3. Interrompre la liaison de communication :

La liaison de communication est interrompue après la réception des données, si le paramètre CONT avait la valeur "0" au début de l'établissement de la liaison. Si tel n'est pas le cas, la liaison de communication est maintenue.

Si l'exécution de la tâche de réception réussit, le paramètre DONE est mis à "1". Le cas échéant, la liaison de communication est interrompue au préalable (voir ci-dessus la description de la dépendance du paramètre CONT).

A la mise à "1" du paramètre COM_RST, TRCV_C est remis à "0". Une perte de données risque de se produire si la réception de données intervient au moment de la nouvelle exécution. Les cas suivants sont possibles en fonction du paramètre CONT :

- CONT = "0" :
Une liaison de communication existante est établie.
- CONT = "1" et une liaison de communication était établie :
Une liaison de communication existante est remise à 0 et rétablie.
- CONT = "1" et aucune liaison de communication n'était établie :
Aucune liaison de communication n'est établie.

Le paramètre COM_RST est remis à 0 après l'évaluation par l'instruction TRCV_.

TRCV_C gère les mêmes modes de réception que l'instruction TRCV. Le tableau suivant montre comment les données sont entrées dans la zone de réception.

Variante de protocole	Disponibilité des données dans la zone de réception	Paramètre Connection_type de la description de la liaison	Paramètre LEN	Paramètre RCVD_LEN
TCP (Mode ad hoc)	Les données sont immédiatement disponibles.	B#16#11	Sélectionné à l'aide de l'instruction TRCV_C sur l'entrée ADHOC	1 à 1472
TCP (réception de données avec longueur spécifiée)	Les données sont disponibles dès que la longueur des données indiquée dans le paramètre LEN a été intégralement reçue.	B#16#11	1 à 8192	Identique à la valeur du paramètre LEN
ISO sur TCP (transmission de données commandée par protocole)	Les données sont disponibles dès que la longueur des données indiquée dans le paramètre LEN a été intégralement reçue.	B#16#12	1 à 8192	Identique à la valeur du paramètre LEN

Remarque

Mode ad hoc

Le "mode ad hoc" n'est disponible qu'avec la variante de protocole TCP. Pour activer le mode ad hoc de l'instruction TRCV_C, définissez le paramètre d'entrée de l'instruction ADHOC. La longueur de la zone de réception est définie par le pointeur dans le paramètre DATA. La longueur de données réellement reçue est émise dans le paramètre RCVD_LEN. Le nombre maximum d'octets pouvant être reçus est égal à 1460.

Remarque

Importation dans le S7-1200 de projets STEP 7 S7-300/400 utilisant le "mode ad hoc"

Dans les projets STEP 7 S7-300/400, le "mode ad hoc" est sélectionné par affectation de 0 au paramètre LEN. Dans S7-1200, vous paramétrez le mode ad hoc de l'instruction TRCV_C en définissant le paramètre d'entrée de l'instruction ADHOC.

Si vous importez dans le S7-1200 un projet STEP 7 S7-300/400 utilisant le "mode ad hoc", vous devez donner la valeur 65535 au paramètre LEN.

Remarque

TCP (réception de données avec longueur spécifiée)

La valeur du paramètre LEN vous permet de définir la longueur des données reçues. Les données spécifiées dans le paramètre DATA sont disponibles dans la zone de réception dès que la longueur de données indiquée dans le paramètre LEN a été intégralement reçue.

Remarque

ISO sur TCP (transmission de données commandée par protocole)

Dans la variante de protocole ISO sur TCP, la transmission des données est commandée par le protocole. La zone de réception est définie par les paramètres LEN et DATA.

Paramètres BUSY, DONE et ERROR

Remarque

En raison du traitement asynchrone de TSEND_C, vous devez faire en sorte que les données dans la zone d'émission restent cohérentes jusqu'à ce que le paramètre DONE ou le paramètre ERROR prenne la valeur VRAI.

Pour TSEND_C, un état VRAI du paramètre DONE signifie que les données ont été envoyées avec succès. Cela ne signifie pas que la CPU partenaire de liaison ait effectivement lu la mémoire tampon de réception.

En raison du traitement asynchrone de TRCV_C, les données dans la zone de réception ne sont cohérentes que lorsque le paramètre DONE est égal à 1.

Tableau 11- 13 Paramètres BUSY, DONE et ERROR des instructions TSEND_C et TRCV_C

BUSY	DONE	ERROR	Description
1	0	0	La tâche d'émission est en cours d'exécution.
0	1	0	La tâche d'émission a été achevée avec succès.
0	0	1	L'établissement de la liaison ou la tâche d'émission s'est achevé sur une erreur. La cause de l'erreur figure dans le paramètre STATUS.
0	0	0	Aucune nouvelle tâche d'émission n'a été affectée.

Vous pouvez contrôler l'état d'exécution au moyen des paramètres BUSY, DONE, ERROR et STATUS. Le paramètre BUSY indique l'état du traitement. Avec le paramètre DONE, vous pouvez vérifier si une tâche s'est exécutée correctement. Le paramètre ERROR est mis à 1 si des erreurs se sont produites pendant l'exécution de TSEND_C ou TRCV_C. Les informations d'erreur sont transmises dans le paramètre STATUS.

Paramètres Error et Status

Tableau 11- 14 Codes d'erreur TSEND_C et TRCV_C pour ERROR et STATUS

ERROR	STATUS (W#16#...)	Description
0	0000	Tâche d'émission (TSEND_C) ou de réception (TRCV_C) exécutée sans erreur.
0	7000	Pas d'exécution de tâche d'émission en cours, pas de liaison de communication établie.
0	7001	<ul style="list-style-type: none"> • Démarrer l'exécution de la tâche d'émission (TSEND_C) ou de réception (TRCV_C). • Etablir la liaison. • Attendre le partenaire de liaison.
0	7002	Données en cours d'émission (TSEND_C) ou de réception (TRCV_C).
0	7003	La liaison de communication est en cours d'interruption.
0	7004	Liaison de communication établie et surveillée, aucune exécution de tâche d'émission (TSEND_C) ou de réception (TRCV_C) active
0	7005	La liaison de communication est remise à 0
1	80A0	Erreur groupée pour les codes d'erreur W#16#80A1 et W#16#80A2.
1	80A1	<ul style="list-style-type: none"> • Liaison ou port déjà occupé par un utilisateur. • Erreur de communication : <ul style="list-style-type: none"> – La liaison indiquée n'a pas encore été établie. – La liaison indiquée est en train d'être coupée. Une transmission via cette liaison n'est pas possible. – L'interface est en cours de réinitialisation.
1	80A2	Port local ou distant utilisé par le système
1	80A3	<ul style="list-style-type: none"> • Tentative de rétablir une liaison existante. • Tentative de mettre fin à une liaison inexistante.
1	80A4	L'adresse IP du nœud d'extrémité distant de la connexion est invalide, c'est-à-dire qu'elle ne correspond pas à l'adresse IP du partenaire local.
1	80A7	Erreur de communication : Vous avez appelé l'instruction avec COM_RST = 1 avant que la tâche d'émission ne soit terminée.
1	80B2	Le paramètre CONNECT pointe sur un bloc de données qui a été généré avec l'attribut "Sauvegarder uniquement dans la mémoire de chargement".
1	80B3	Paramétrage incohérent : Erreur groupée pour les codes d'erreur W#16#80A0 à W#16#80A2, W#16#80A4, W#16#80B4 à W#16#80B9.
1	80B4	<p>Lors de l'utilisation de la variante de protocole ISO sur TCP (connection_type = B#16#12) pour l'établissement passif d'une liaison (active_est = FALSE) l'une ou les deux conditions suivantes n'ont pas été respectées :</p> <ul style="list-style-type: none"> • local_tsap_id_len >= B#16#02 • local_tsap_id[1] = B#16#E0
1	80B5	Seul un établissement de liaison passif est autorisé pour le type de liaison 13 = UDP.
1	80B6	Erreur de paramétrage dans le paramètre connection_type du bloc de données servant à la description de la liaison
1	80B7	Erreur dans l'un des paramètres suivants du bloc de données servant à la description de la liaison : block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len, next_staddr_len.

ERROR	STATUS (W#16#...)	Description
1	8085	<ul style="list-style-type: none"> Le paramètre LEN est supérieur à la valeur maximale admise. La valeur du paramètre LEN ou DATA a été modifiée après le premier appel.
1	8086	Le paramètre ID à l'intérieur du paramètre CONNECT se situe en-dehors de la plage admise.
1	8087	Nombre maximal de liaisons atteint ; aucune liaison supplémentaire n'est possible.
1	8088	La valeur du paramètre LEN ne correspond pas à la zone de réception indiquée dans le paramètre DATA.
1	8089	Le paramètre CONNECT ne pointe pas sur un bloc de données.
1	8091	Profondeur d'imbrication maximale dépassée
1	809A	Le paramètre CONNECT pointe sur un champ qui ne correspond pas à la longueur de la description de liaison.
1	809B	L'ID de l'appareil local dans la description de la liaison ne correspond pas à la CPU.
1	80C3	<ul style="list-style-type: none"> Toutes les ressources de liaison (Page 854) sont utilisées. Un bloc avec cet ID est déjà en cours de traitement dans un groupe de priorité différent.
1	80C4	<p>Erreur de communication temporaire :</p> <ul style="list-style-type: none"> La liaison ne peut pas être établie en ce moment. L'interface est en train de recevoir de nouveaux paramètres ou la liaison est en cours d'établissement. La liaison configurée est en cours de suppression par une instruction "TDISCON (Page 885)". La liaison utilisée est en cours de fermeture par un appel avec COM_RST = 1.
1	8722	Erreur dans le paramètre CONNECT : La zone source est invalide (zone n'existe pas dans le DB).
1	873A	Erreur dans le paramètre CONNECT : Impossible d'accéder à la description de la liaison (pas d'accès au DB)
1	877F	Erreur dans le paramètre CONNECT : Erreur interne
1	8822	TSEND_C : Paramètre DATA : Zone source invalide, zone inexistante dans le DB.
1	8824	TSEND_C : Paramètre DATA : Erreur de zone dans le pointeur VARIANT.
1	8832	TSEND_C : Paramètre DATA : Le numéro de DB est trop grand.
1	883A	TSEND_C : Paramètre CONNECT : Impossible d'accéder aux données de connexion indiquées (par exemple, parce que le DB n'existe pas).
1	887F	TSEND_C : Paramètre DATA : Erreur interne (par exemple référence VARIANT invalide)
1	893A	TSEND_C : Paramètre DATA : Impossible d'accéder à la zone d'émission (par exemple, parce que le DB n'existe pas).
1	8922	TRCV_C : Paramètre DATA : Zone cible invalide, zone inexistante dans le DB.
1	8924	TRCV_C : Paramètre DATA : Erreur de zone dans le pointeur VARIANT.
1	8932	TRCV_C : Paramètre DATA : Le numéro de DB est trop grand.
1	893A	TRCV_C : Paramètre CONNECT : Impossible d'accéder aux données de connexion indiquées (par exemple, parce que le DB n'existe pas).
1	897F	TRCV_C : Paramètre DATA : Erreur interne (par exemple, référence VARIANT invalide).
1	8A3A	TRCV_C : Paramètre DATA : Impossible d'accéder à la zone de données (par exemple, parce que le DB n'existe pas).

Remarque**Messages d'erreur des instructions TCON, TSEND, TRCV et TDISCON**

En interne, l'instruction TSEND_C utilise les instructions TCON, TSEND et TDISCON ; et l'instruction TRCV_C utilise les instructions TCON, TRCV et TDISCON. Reportez-vous à "TCON, TDISCON, TSEND et TRCV (Page 885)" pour plus d'informations sur les messages d'erreur de ces instructions.

Protocoles de communication Ethernet

Chaque CPU comprend un port PROFINET intégré qui prend en charge la communication PROFINET standard. Les instructions TSEND_C, TRCV_C, TSEND et TRCV prennent toutes en charge les protocoles Ethernet TCP et ISO sur TCP.

Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 833)" pour plus d'informations.

11.2.8.8 Instructions d'héritage TSEND_C et TRCV_C

Avant le lancement de STEP 7 V13 SP1 et des CPU S7-1200 V4.1, les instructions TSEND_C et TRCV_C pouvaient fonctionner uniquement avec des paramètres de connexion possédant des structures selon "TCON_Param". Les concepts généraux s'appliquent aux deux jeux d'instructions. Reportez-vous aux instructions d'héritage individuelles TSEND_C et TRCV_C pour obtenir des informations relatives à la programmation.

Sélection de la version des instructions TSEND_C et TRCV_C

Il existe deux versions des instructions TSEND_C et TRCV_C disponibles dans STEP 7 :

- Les versions 2.5 et 3.1 étaient disponibles dans STEP 7 Basic/Professional versions V13 ou antérieures.
- La version 4.0 est disponible dans STEP 7 Basic/Professional V13 SP1 ou ultérieur.

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

N'utilisez pas des versions d'instructions différentes dans le même programme CPU.



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.

Open user communication		V3.1
TSEND_C	Send data via Ethernet (TCP)	V2.5
TRCV_C	Receive data via Ethernet (T...	V3.1
TMAIL_C	Send e-mail	V4.0
Others		V3.0
TCON	Establish communication c...	V3.0
TDISCON	Terminate communication ...	V2.1
TSEND	Send data via communicati...	V3.0
TRCV	Receive data via communic...	V3.0

Pour changer la version des instructions TSEND_C et TRCV_C, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction TSEND_C ou TRCV_C dans votre programme, une nouvelle instance de FB ou FC, en fonction de l'instruction TSEND_C ou TRCV_C sélectionnée, est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB ou FC dans l'arborescence du projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction TSEND_C ou TRCV_C dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB ou FC TSEND_C ou TRCV_C dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction TSEND_C ou TRCV_C.

TSEND_C et TRCV_C (Envoyer et recevoir des données via Ethernet) hérités

Les instructions d'héritage TSEND_C combinent les fonctions des instructions d'héritage TCON, TDISCON et TSEND . L'instruction TRCV_C combine les fonctions des instructions TCON, TDISCON et TRCV. (Reportez-vous à "Instructions d'héritage TCON, TDISCON, TSEND, et TRCV (communication TCP) (Page 896)" pour plus d'informations sur ces instructions.)

La taille de données minimale que vous pouvez envoyer (TSEND_C) ou recevoir (TRCV_C) est d'un octet ; la taille maximale est de 8192 octets. TSEND_C n'accepte pas l'envoi de données à partir d'adresses booléennes et TRCV_C ne reçoit pas de données dans des adresses booléennes. Pour plus d'informations sur le transfert de données à l'aide de ces instructions, reportez-vous au paragraphe sur la cohérence des données (Page 204).

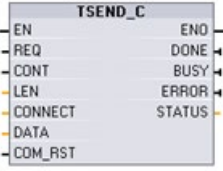
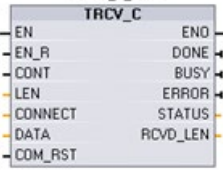
Remarque

Initialisation des paramètres de communication

Une fois que vous avez inséré l'instruction TSEND_C ou TRCV_C, utilisez les "Propriétés" de l'instruction (Page 833) pour configurer les paramètres de communication (Page 857). Lorsque vous entrez les paramètres pour les partenaires de communication dans la fenêtre d'inspection, STEP 7 entre les données correspondantes dans le DB pour l'instruction.

Si vous voulez utiliser un DB multi-instance, vous devez configurer le DB manuellement dans les deux CPU.

Tableau 11- 15 Instructions TSEND_C et TRCV_C

CONT/LOG	SCL	Description
<p>"TSEND_C_DB"</p> 	<pre>"TSEND_C_DB" (req:=_bool_in_, cont:=_bool_in_, len:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, connect:=_struct_inout_, data:=_variant_inout_, com_rst:=_bool_inout_);</pre>	<p>TSEND_C établit une liaison de communication TCP ou ISO sur TCP à une station partenaire, envoie des données et peut mettre fin à la liaison. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.</p>
<p>"TRCV_C_DB"</p> 	<pre>"TRCV_C_DB" (en_r:=_bool_in_, cont:=_bool_in_, len:=_uint_in_, adhoc:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, rcvd_len=>_uint_out_, connect:=_struct_inout_, data:=_variant_inout_, com_rst:=_bool_inout_);</pre>	<p>TRCV_C établit une liaison de communication TCP ou ISO sur TCP à une CPU partenaire, reçoit des données et peut mettre fin à la liaison. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 11- 16 Types de données pour les paramètres de TSEND_C et TRCV_C

Paramètre et type		Type de données	Description
REQ (TSEND_C)	IN	Bool	REQ = 1 lance la tâche d'émission TSEND_C en présence d'un front montant avec la liaison décrite dans le paramètre CONNECT (CONT = 1 est également requis pour établir et conserver la liaison de communication).
EN_R (TRCV_C)	IN	Bool	Lorsque EN_R est égal à 1, TRCV_C est prêt à recevoir. La tâche de réception est traitée (CONT = 1 est également requis pour établir et conserver la liaison de communication).
CONT	IN	Bool	Contrôle la liaison de communication : <ul style="list-style-type: none"> • 0 : Déconnecter la liaison de communication • 1 : Etablir et conserver la liaison de communication Lors de l'envoi de données (TSEND_C) (front montant au niveau du paramètre REQ), le paramètre CONT doit avoir la valeur TRUE afin d'établir ou de conserver une liaison. Lors de la réception de données (TRCV_C) (front montant au niveau du paramètre EN_R), le paramètre CONT doit avoir la valeur TRUE afin d'établir ou de conserver une liaison.
LEN	IN	UInt	Nombre maximum d'octets à envoyer (TSEND_C) ou à recevoir (TRCV_C) : <ul style="list-style-type: none"> • Valeur par défaut = 0 : Le paramètre DATA détermine la longueur des données à envoyer (TSEND_C) ou à recevoir (TRCV_C). • Mode ad hoc = 65535 : Une longueur de données variable est définie pour la réception (TRCV_C).
CONNECT	IN_OUT	TCON_Param	Pointeur désignant la description de la liaison (Page 857)
DATA	IN_OUT	Variant	<ul style="list-style-type: none"> • Contient l'adresse et la longueur des données à envoyer (TSEND_C) • Contient l'adresse de début et la longueur maximale des données reçues (TRCV_C).
COM_RST	IN_OUT	Bool	Permet le redémarrage de l'instruction : <ul style="list-style-type: none"> • 0 : Sans objet • 1 : Redémarrage complet du bloc fonctionnel avec coupure de la liaison existante
DONE	OUT	Bool	<ul style="list-style-type: none"> • 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. • 1 : Tâche achevée sans erreur
BUSY	OUT	Bool	<ul style="list-style-type: none"> • 0 : Tâche achevée • 1 : La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche.
ERROR	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> • 0 : Pas d'erreur • 1 : Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.

Paramètre et type		Type de données	Description
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir le tableau "Paramètres Error et Status" ci-après)
RCVD_LEN (TRCV_C)	OUT	Int	Quantité de données effectivement reçue en octets

Remarque

L'instruction TSEND_C requiert un front montant du paramètre d'entrée REQ pour démarrer une tâche d'émission. Le paramètre BUSY est mis à 1 pendant le traitement. L'achèvement de la tâche d'émission est signalé par la mise à 1 du paramètre DONE ou ERROR pendant un cycle. Pendant ce temps, aucun front montant du paramètre d'entrée REQ n'est pris en compte.

Remarque

Avec la valeur par défaut du paramètre LEN (LEN = 0), le paramètre DATA est utilisé pour déterminer la longueur des données transmises. Il est recommandé que les données envoyées par l'instruction TSEND_C aient la même taille que le paramètre DATA de l'instruction TRCV_C.

Si, en cas d'utilisation de la valeur par défaut du paramètre LEN, il s'avère nécessaire d'envoyer des données en segments plus petits que la taille du paramètre DATA, la règle suivante s'applique. Si la taille des données envoyées par TSEND_C est différente de la taille du paramètre DATA de TRCV_C, TRCV_C reste à l'état occupé (code d'état : 7006) jusqu'à ce que la taille totale des données envoyées par TSEND_C soit égale à la taille du paramètre DATA de TRCV_C.

Le tampon du paramètre DATA de TRCV_C n'affiche pas les nouvelles données reçues tant que leur taille n'est pas égale à celle du tampon du paramètre DATA.

Fonctionnement de TSEND_C

Les fonctions suivantes décrivent le fonctionnement de l'instruction TSEND_C :

- Pour établir une liaison, exécutez TSEND_C avec CONT = 1.
- Une fois la liaison établie avec succès, TSEND_C met le paramètre DONE à 1 pour un cycle.
- Pour mettre fin à la liaison de communication, exécutez TSEND_C avec CONT = 0. La liaison est immédiatement interrompue. Cela affecte également la station réceptrice. La liaison y est également coupée et il peut y avoir perte de données dans la mémoire tampon de réception.
- Pour envoyer des données par le biais d'une liaison établie, exécutez TSEND_C avec un front montant en REQ. Après une opération d'envoi réussie, TSEND_C met le paramètre DONE à 1 pour un cycle.
- Pour établir une liaison et envoyer des données, exécutez TSEND_C avec CONT = 1 et REQ = 1. Après une opération d'envoi réussie, TSEND_C met le paramètre DONE à 1 pour un cycle.

Fonctionnement de TRCV_C

Les fonctions suivantes décrivent le fonctionnement de l'instruction TRCV_C :

- Pour établir une liaison, exécutez TRCV_C avec le paramètre CONT = 1.
- Pour recevoir des données, exécutez TRCV_C avec le paramètre EN_R = 1. TRCV_C reçoit les données en continu lorsque les paramètres EN_R = 1 et CONT = 1.
- Pour mettre fin à la liaison, exécutez TRCV_C avec le paramètre CONT = 0. La liaison est immédiatement interrompue et une perte des données est possible.

TRCV_C gère les mêmes modes de réception que l'instruction TRCV. Le tableau suivant montre comment les données sont entrées dans la zone de réception.

Tableau 11- 17 Entrée des données dans la zone de réception

Variante de protocole	Entrée des données dans la zone de réception	Paramètre "con- nection_type"	Valeur du paramètre LEN	Valeur du paramètre RCVD_LEN (octets)
TCP	Mode ad hoc	B#16#11	65535	1 à 1472
TCP	Réception de données avec longueur indiquée	B#16#11	0 (recommandé) ou 1 à 8192, excepté 65535	1 à 8192
ISO sur TCP	Mode ad hoc	B#16#12	65535	1 à 1472
ISO sur TCP	Gestion par le protocole	B#16#12	0 (recommandé) ou 1 à 8192, excepté 65535	1 à 8192

Remarque

Mode ad hoc

Le "mode ad hoc" existe avec les variantes de protocole TCP et ISO sur TCP. Vous paramétrez le "mode ad hoc" en affectant "65535" au paramètre LEN. La zone de réception est identique à la zone formée par DATA. La longueur des données reçues est transmise au paramètre RCVD_LEN.

Si vous stockez les données dans un DB "optimisé" (symbolique uniquement), vous ne pouvez recevoir les données qu'en tableaux de types de données Byte, Char, USInt et SInt.

Remarque

Importation dans le S7-1200 de projets STEP 7 S7-300/400 utilisant le "mode ad hoc"

Dans les projets STEP 7 S7-300/400, le "mode ad hoc" est sélectionné par affectation de 0 au paramètre LEN. Dans le S7-1200, vous paramétrez le "mode ad hoc" en affectant "65535" au paramètre LEN.

Si vous importez dans le S7-1200 un projet STEP 7 S7-300/400 utilisant le "mode ad hoc", vous devez donner la valeur 65535 au paramètre LEN.

Remarque

Vous devez faire en sorte que les données dans la zone d'émission restent cohérentes jusqu'à ce que le paramètre DONE ou le paramètre ERROR prenne la valeur VRAI.

En raison du traitement asynchrone de TSEND_C, vous devez faire en sorte que les données dans la zone d'émission restent cohérentes jusqu'à ce que le paramètre DONE ou le paramètre ERROR prenne la valeur VRAI.

Pour TSEND_C, un état VRAI du paramètre DONE signifie que les données ont été envoyées avec succès. Cela ne signifie pas que la CPU partenaire de liaison ait effectivement lu la mémoire tampon de réception.

En raison du traitement asynchrone de TRCV_C, les données dans la zone de réception ne sont cohérentes que lorsque le paramètre DONE est égal à 1.

Tableau 11- 18 Paramètres BUSY, DONE et ERROR des instructions TSEND_C et TRCV_C

BUSY	DONE	ERROR	Description
VRAI	Sans objet	Sans objet	La tâche est en cours de traitement.
FAUX	VRAI	FAUX	La tâche a été achevée avec succès.
FAUX	FAUX	VRAI	La tâche s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS.
FAUX	FAUX	FAUX	Aucune nouvelle tâche n'a été affectée.

Codes d'erreur de TSEND_C, TRCV_C Error et Status

ERROR	STATUS	Description
0	0000	Tâche exécutée sans erreur
0	7000	Pas de traitement de tâche actif
0	7001	Démarrage du traitement de la tâche, établissement de la liaison, attente du partenaire de liaison
0	7002	Données en cours d'émission ou de réception
0	7003	Coupure de la liaison
0	7004	Liaison établie et surveillée ; pas de traitement de tâche actif
1	8085	Le paramètre LEN est supérieur à la plus grande valeur autorisée.
1	8086	Le paramètre CONNECT se situe hors de la plage permise.
1	8087	Nombre maximal de liaisons atteint ; aucune liaison supplémentaire n'est possible.
1	8088	Le paramètre LEN n'est pas valide pour la zone de mémoire indiquée dans DATA.
1	8089	Le paramètre CONNECT ne pointe pas sur un bloc de données.
1	8091	Profondeur d'imbrication maximale dépassée
1	809A	Le paramètre CONNECT pointe sur un champ qui ne correspond pas à la longueur de la description de liaison.
1	809B	L'identificateur local_device_id dans la description de la liaison ne correspond pas à la CPU.

ERROR	STATUS	Description
1	80A1	Erreur de communication : <ul style="list-style-type: none"> • La liaison indiquée n'a pas encore été établie. • La liaison indiquée est en train d'être coupée ; une transmission via cette liaison n'est pas possible. • L'interface est en cours de réinitialisation.
1	80A3	Tentative de mettre fin à une liaison inexistante
1	80A4	L'adresse IP de la liaison au partenaire distant est invalide. Par exemple, l'adresse IP du partenaire distant est identique à l'adresse IP du partenaire local.
1	80A5	ID de liaison (Page 854) déjà utilisé
1	80A7	Erreur de communication : Vous avez appelé TDISCON avant que TSEND_C ne soit achevé.
1	80B2	Le paramètre CONNECT pointe sur un bloc de données qui a été généré avec le mot-clé UNLINKED.
1	80B3	Paramètres incohérents : <ul style="list-style-type: none"> • Erreur dans la description de la liaison • Le port local (paramètre local_tsap_id) est déjà présent dans une autre description de liaison. • ID dans la description de liaison différent de l'ID indiqué comme paramètre.
1	80B4	Lors de l'utilisation de ISO-on-TCP (connection_type = B#16#12) pour établir une liaison passive, le code d'erreur 80B4 vous avertit que le TSAP entré n'était pas conforme à l'une des exigences suivantes eu égard à l'adresse : <ul style="list-style-type: none"> • Pour une longueur TSAP locale égale à 2 et une valeur ID TSAP égale à E0 ou E1 (hexadécimal) pour le premier octet, le deuxième octet doit être 00 ou 01. • Pour une longueur TSAP locale supérieure ou égale à 3 et une valeur ID TSAP égale à E0 ou E1 (hexadécimal) pour le premier octet, le deuxième octet doit être 00 ou 01 et tous les autres octets doivent être des caractères ASCII valides. • Pour une longueur TSAP locale supérieure ou égale à 3 et une valeur ID TSAP différente de E0 et E1 (hexadécimal) pour le premier octet, tous les octets de l'ID TSAP doivent être des caractères ASCII valides. Les caractères ASCII valides ont des valeurs d'octet allant de 20 à 7E (hexadécimal).
1	80B7	Type de données et/ou longueur des données émises incompatibles avec la zone de la CPU partenaire dans laquelle elles doivent être écrites.
1	80C3	Toutes les ressources de liaison sont utilisées.
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none"> • La liaison ne peut pas être établie en ce moment. • L'interface est en train de recevoir de nouveaux paramètres. • La liaison configurée est en cours de suppression par TDISCON.
1	8722	Paramètre CONNECT : Zone source invalide : la zone n'existe pas dans le DB.
1	873A	Paramètre CONNECT : Impossible d'accéder à la description de la liaison (DB non disponible par exemple)
1	877F	Paramètre CONNECT : Erreur interne telle qu'une référence ANY invalide
1	893A	Le paramètre contient le numéro d'un DB qui n'est pas chargé.

Protocoles de communication Ethernet

Chaque CPU comprend un port PROFINET intégré qui prend en charge la communication PROFINET standard. Les instructions TSEND_C, TRCV_C, TSEND et TRCV prennent toutes en charge les protocoles Ethernet TCP et ISO sur TCP.

Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 833)" pour plus d'informations.

11.2.8.9 Instructions TCON, TDISCON, TSEND et TRCV

À partir de la version V4.1 de la CPU S7-1200 avec STEP 7 V13 SP1, la CPU étend les fonctionnalités des instructions TCON, TDISCON, TSEND et TRCV en permettant l'utilisation de paramètres de liaison avec des structures selon "TCON_IP_v4" et "TCON_IP_RFC".

Pour cette raison, le S7-1200 accepte deux jeux d'instructions TCON, TDISCON, TSEND et TRCV :

- Instructions d'héritage TCON, TDISCON, TSEND et TRCV (Page 896): Ces instructions TCON, TDISCON, TSEND et TRCV existaient avant la version V4.0 du S7-1200 et fonctionnent uniquement avec les paramètres de connexion possédant des structures selon "TCON_Param".
- Instructions TCON, TDISCON, TSEND et TRCV (Page 885): Ces instructions TCON, TDISCON, TSEND et TRCV offrent toutes les fonctionnalités des instructions d'héritage, ainsi que la capacité à utiliser les paramètres de connexion possédant des structures selon "TCON_IP_v4" et "TCON_IP_RFC".

Sélection de la version des instructions TCON, TDISCON, TSEND et TRCV

Il existe deux versions des instructions TCON, TDISCON, TSEND, ou TRCV disponibles dans STEP 7 :

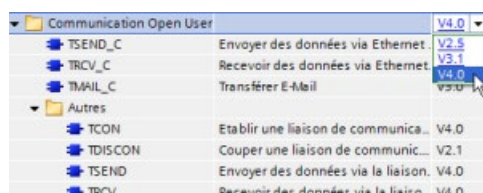
- Les versions 2.5 et 3.1 étaient disponibles dans STEP 7 Basic/Professional versions V13 ou antérieures.
- La version 4.0 est disponible dans STEP 7 Basic/Professional V13 SP1 ou ultérieur.

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

N'utilisez pas des versions d'instructions différentes dans le même programme CPU.



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.



Pour changer la version des instructions TCON, TDISCON, TSEND ou TRCV, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction TCON, TDISCON, TSEND ou TRCV dans votre programme, une nouvelle instance de FB ou FC, en fonction de l'instruction TCON, TDISCON, TSEND ou TRCV sélectionnée, est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB ou FC dans l'arborescence du projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction TCON, TDISCON, TSEND ou TRCV dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB ou FC TCON, TDISCON, TSEND ou TRCV dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction TCON, TDISCON, TSEND ou TRCV.

Instructions TCON, TDISCON, TSEND et TRCV (communication TCP)

Communication Ethernet à l'aide des protocoles TCP et ISO sur TCP

Remarque

Instructions TSEND_C et TRCV_C

Pour simplifier la programmation de la communication PROFINET/Ethernet, l'instruction TSEND_C et l'instruction TRCV_C combinent les fonctionnalités des instructions TCON, TDISCON, TSEND et TRCV :

- TSEND_C combine les instructions TCON, TDISCON et TSEND.
- TRCV_C combine les instructions TCON, TDISCON et TRCV.

Les instructions suivantes gèrent le processus de communication :

- TCON établit une liaison TCP/IP entre un PC client et serveur (CPU).
- TSEND et TRCV envoient et reçoivent les données.
- TDISCON coupe la liaison.

La taille de données minimale que vous pouvez envoyer (TSEND) ou recevoir (TRCV) est de un octet ; la taille maximale est de 8 192 octets. TSEND n'accepte pas l'envoi de données à partir d'adresses booléennes et TRCV ne reçoit pas de données dans des adresses booléennes. Pour plus d'informations sur le transfert de données à l'aide de ces instructions, reportez-vous au paragraphe sur la cohérence des données (Page 204).

TCON, TDISCON, TSEND et TRCV opèrent de manière asynchrone, ce qui signifie que le traitement de la tâche s'étend sur plusieurs exécutions de l'instruction. Par exemple, vous lancez une tâche pour la configuration et l'établissement d'une liaison en exécutant une instruction TCON avec le paramètre REQ = 1. Puis, vous réexécutez TCON pour surveiller l'avancement de la tâche et tester son achèvement au moyen du paramètre DONE.

Le tableau suivant montre les relations entre BUSY, DONE et ERROR. Servez-vous-en pour déterminer l'état de la tâche en cours.

Tableau 11- 19 Interactions entre les paramètres BUSY, DONE et ERROR

BUSY	DONE	ERROR	Description
1	0	0	La tâche est en cours de traitement.
0	1	0	La tâche a été achevée avec succès.
0	0	1	La tâche s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS.
0	0	0	Aucune nouvelle tâche n'a été affectée.

TCON et TDISCON


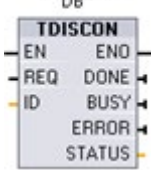
Remarque

Initialisation des paramètres de communication

Une fois que vous avez inséré l'instruction TCON, utilisez les "Propriétés" de l'instruction (Page 833) pour configurer les paramètres de communication (Page 857). Lorsque vous entrez les paramètres pour les partenaires de communication dans la fenêtre d'inspection, STEP 7 entre les données correspondantes dans le DB d'instance pour l'instruction.

Si vous voulez utiliser un DB multi-instance, vous devez configurer le DB manuellement dans les deux CPU.

Tableau 11- 20 Instructions TCON et TDISCON

CONT/LOG		Description
	<pre>"TCON_DB" (req:=_bool_in_, ID:=_undef_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, connect:= struct inout);</pre>	TCP et ISO sur TCP : TCON établit une liaison de communication de la CPU vers un partenaire de communication.
	<pre>"TDISCON_DB" (req:=_bool_in_, ID:=_word_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	TCP et ISO sur TCP : TDISCON met fin à une liaison de communication de la CPU vers un partenaire de communication.

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 11- 21 Types de données pour les paramètres de TCON et TDISCON

Paramètre	Déclaration	Type de données	Description
REQ	IN	Bool	Démarre la tâche d'établissement de la liaison indiquée dans le paramètre ID en cas de front montant.
ID	IN	CONN_OUC (Word)	Référence à la liaison affectée. Plage de valeurs : W#16#0001 à W#16#0FFF
CONNECT (TCON)	IN_OUT	VARIANT	Pointeur désignant la description de la liaison <ul style="list-style-type: none"> Pour TCP ou UDP, utilisez la structure TCON_IP_v4. Pour plus d'informations sur TCON_IP_v4, voir : "Paramètres pour la liaison PROFINET" (Page 857). Pour ISO sur TCP, utilisez la structure TCON_IP_RFC. Pour plus d'informations sur TCON_IP_RFC, voir : "Paramètres pour la liaison PROFINET" (Page 857).
DONE	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> 0 : La tâche n'a pas encore été démarrée ou elle est encore en cours d'exécution 1 : Tâche exécutée sans erreur
BUSY	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> 0 : La tâche n'a pas encore été démarrée ou elle est déjà achevée 1 : La tâche n'est pas encore achevée. Il n'est pas possible de démarrer une nouvelle tâche
ERROR	OUT	Bool	Paramètre d'état ERROR : <ul style="list-style-type: none"> 0 : Pas d'erreur 1 : Une erreur s'est produite
STATUS	OUT	Word	Etat de l'instruction

Les deux partenaires de communication exécutent l'instruction TCON pour configurer et établir la liaison de communication. Vous vous servez de paramètres pour indiquer les partenaires de noeud d'extrémité actif et passif. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.

Si la liaison est interrompue en raison d'une coupure de ligne ou à cause du partenaire de communication distant par exemple, le partenaire actif tente de rétablir la liaison configurée. Vous n'avez pas besoin de réexécuter TCON.

Une liaison existante est coupée et la liaison configurée est supprimée lors de l'exécution de l'instruction TDISCON ou lorsque la CPU passe à l'état ARRET. Vous devez réexécuter TCON pour reconfigurer et rétablir la liaison.

Tableau 11- 22 Codes d'erreur pour les paramètres ERROR et STATUS de TCON et TDISCON

ERROR	STATUS (W#16#...)	Explication
0	0000	La liaison a été établie avec succès.
0	7000	Pas de traitement de tâche actif
0	7001	Démarrage de l'exécution de la tâche ; établissement de la liaison (TCON) ou déconnexion (TDISCON).
0	7002	Etablissement de la liaison (REQ sans objet) ; établissement de la liaison (TCON) ou déconnexion (TDISCON).
1	8085	TCON : ID de liaison déjà utilisée
1	8086	TCON : Le paramètre ID se situe hors de la plage valide.
1	8087	TCON : Nombre maximal de liaisons atteint ; aucune liaison supplémentaire n'est possible.
1	8089	TCON : Le paramètre CONNECT ne pointe pas sur une description de liaison ou la description de liaison a été créée manuellement.
1	809A	TCON : La structure dans le paramètre CONNECT n'est pas prise en charge ou la longueur est invalide.
1	809B	TCON : L'ID de l'appareil local dans la description de la liaison ne correspond pas à la CPU ou au CP ou bien est égale à 0.
1	80A0	Erreur groupée pour les codes d'erreur W#16#80A1 et W#16#80A2.
1	80A1	TCON : Pour TCP/UDP (TCON_IP_v4) : Liaison ou port déjà utilisé.
1	80A2	TCON: port local ou distant actuellement utilisé par le système.
1	80A3	TCON : La valeur dans le paramètre ID est déjà utilisée par une liaison (TCON) créée par le programme utilisateur. La liaison utilise le même identifiant mais des paramètres de liaison différents dans le paramètre CONNECT.
1	80A4	TCON : L'adresse IP du nœud d'extrémité distant de la liaison est invalide ou elle correspond à l'adresse IP du partenaire local.
1	80A5	TCON : ID de liaison déjà utilisée
1	80A7	TCON : Erreur de communication : Vous avez exécuté "TDISCON" avant que "TCON" ne soit achevé.
1	80B2	TCON : Le paramètre CONNECT pointe sur un bloc de données qui a été généré avec l'attribut "Sauvegarder uniquement dans la mémoire de chargement".
1	80B3	Paramétrage incohérent : Erreur groupée pour les codes d'erreur W#16#80A0 à W#16#80A2, W#16#80A4, W#16#80B4 à W#16#80B9.
1	80B4	TCON : Uniquement avec TCON_IP_RFC. Le sélecteur T local n'a pas été indiqué, le premier octet ne contient pas la valeur 0x0E ou le sélecteur T local commence par "SIMATIC-".
1	80B5	TCON : Seul un établissement de liaison passif est autorisé pour le type de liaison 13 = UDP (le paramètre active_est de la structure TCON_IP_v4 a la valeur TRUE).
1	80B6	TCON : Erreur de paramétrage dans le paramètre connection_type du bloc de données servant à la description de la liaison <ul style="list-style-type: none"> • Valide seulement avec TCON_IP_v4 : 0x11, 0x0B et 0x13. • Valide seulement avec TCON_IP_RFC : 0x0C et 0x12

ERROR	STATUS (W#16#...)	Explication
1	80B7	TCON : Avec TCON_IP_v4 : <ul style="list-style-type: none"> • TCP (initialisation de la liaison active) : Le port distant est égal à "0". • TCP (initialisation de la liaison passive) : Le port local est égal à "0". • UDP : Le port local est égal à "0". TCON : Avec TCON_IP_RFC : <ul style="list-style-type: none"> • Le sélecteur T local (local_tselector) ou distant (remote_tselector) a été indiqué avec une longueur de plus de 32 octets. • Pour TSELlength du sélecteur T (local ou distant), une longueur supérieure à 32 a été entrée. • Erreur dans la longueur de l'adresse IP du partenaire de liaison spécifique.
1	80B8	TCON : Le paramètre ID dans la description de liaison locale (structure dans le paramètre CONNECT) et le paramètre ID de l'instruction sont différents.
1	80C3	TCON : Toutes les ressources de liaison (Page 854) sont utilisées.
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none"> • La liaison ne peut pas être établie en ce moment (TCON). • L'interface est en train de recevoir de nouveaux paramètres (TCON et TDISCON). • La liaison configurée est en cours de suppression par une instruction TDISCON (TCON).
1	80C5	TCON : Le partenaire distant refuse d'établir la liaison, a coupé la liaison ou y a mis fin de manière active.
1	80C6	TCON : Impossible d'atteindre le partenaire distant (erreur de réseau)
1	80C7	TCON : Dépassement du délai d'exécution
1	80C8	TCON : L'ID est utilisée par une liaison créée par le programme utilisateur qui utilise la même description de liaison qu'au paramètre CONNECT.
1	80C9	TCON : La validation du partenaire distant a échoué. Le partenaire distant qui veut établir la liaison ne correspond pas au partenaire défini de la structure dans le paramètre CONNECT.
1	80CE	TCON : L'adresse IP de l'interface locale est 0.0.0.0.

TSEND et TRCV

Remarque

Lors de l'utilisation de la communication PROFINET Open User, si vous exécutez une opération TSEND sans qu'une instruction correspondante TRCV ne soit exécutée sur l'appareil distant, l'instruction TSEND peut rester indéfiniment "occupée", attendant que l'instruction TRCV reçoive les données. Dans cet état, la sortie "Busy" de l'instruction TSEND est mise à 1, et la sortie "Status" a la valeur "0x7002". Cette condition peut se produire si vous transférez plus de 4 096 octets de données. Ce problème est résolu à la nouvelle exécution de l'instruction TRCV.

Tableau 11- 23 Instructions TSEND et TRCV

CONT/LOG	SCL	Description
	<pre>"TSEND_DB" (req:=_bool_in_, ID:=_word_in_, len:=_udint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, data:=_variant_inout_);</pre>	<p>TCP et ISO sur TCP : TSEND envoie des données par le biais d'une liaison de communication de la CPU vers une station partenaire.</p>
	<pre>"TRCV_DB" (en_r:=_bool_in_, ID:=_word_in_, len:=_udint_in_, adhoc:=_bool_in_, ndr=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, rcvd_len=>_udint_out_, data:=_variant_inout_);</pre>	<p>TCP et ISO sur TCP : TRCV reçoit des données par le biais d'une liaison de communication d'une station partenaire vers la CPU.</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 11- 24 Types de données pour les paramètres de TSEND et TRCV

Paramètre et type		Type de données	Description
REQ	IN	Bool	TSEND : Lance la tâche d'émission en présence d'un front montant. Les données sont prises dans la zone indiquée par DATA et LEN.
EN_R	IN	Bool	TRCV : Valide la réception par la CPU ; lorsque EN_R = 1, TRCV est prêt à recevoir. La tâche de réception est traitée.
ID	IN	CONN_OUC (Word)	Référence à la liaison associée. L'ID doit être identique au paramètre ID associé dans la description de la liaison locale. Plage de valeurs : W#16#0001 à W#16#0FFF
LEN	IN	UDInt	Nombre maximum d'octets à envoyer (TSEND) ou à recevoir (TRCV) : <ul style="list-style-type: none"> • Valeur par défaut = 0 : Le paramètre DATA détermine la longueur des données à envoyer (TSEND) ou à recevoir (TRCV). • Mode ad hoc = 65535 : Une longueur de données variable est définie pour la réception (TRCV).
ADHOC	IN	Bool	TRCV : Paramètre facultatif (masqué) Demande ad hoc pour le type de liaison TCP.
DATA	IN_OUT	Variant	Pointeur désignant la zone de données à envoyer (TSEND) ou à recevoir (TRCV). La zone de données contient l'adresse et la longueur. L'adresse fait référence à de la mémoire I, de la mémoire Q, de la mémoire M ou à un DB.
DONE	OUT	Bool	TSEND : <ul style="list-style-type: none"> • 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. • 1 : Tâche exécutée sans erreur
NDR	OUT	Bool	TRCV : <ul style="list-style-type: none"> • NDR = 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. • NDR = 1 : La tâche a été achevée avec succès.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • BUSY = 1 : La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche. • BUSY = 0 : La tâche est achevée.
ERROR	OUT	Bool	ERROR = 1 : Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)
RCVD_LEN	OUT	UDInt	TRCV : Volume de données effectivement reçu en octets

Remarque

L'instruction TSEND requiert un front montant du paramètre d'entrée REQ pour démarrer une tâche d'émission. Le paramètre BUSY est mis à 1 pendant le traitement. L'achèvement de la tâche d'émission est signalé par la mise à 1 du paramètre DONE ou ERROR pendant un cycle. Pendant ce temps, aucun front montant du paramètre d'entrée REQ n'est pris en compte.

Fonctionnement de TRCV

L'instruction TRCV écrit les données reçues dans une zone de réception qui est définie par les deux variables suivantes :

- Pointeur désignant le début de la zone
- Longueur de la zone ou valeur fournie à l'entrée LEN si différente de 0

Remarque

Avec la valeur par défaut du paramètre LEN (LEN = 0), le paramètre DATA est utilisé pour déterminer la longueur des données transmises. Il est recommandé que les données envoyées par l'instruction TSEND aient la même taille que le paramètre DATA de l'instruction TRCV.

Si, en cas d'utilisation de la valeur par défaut du paramètre LEN, il s'avère nécessaire d'envoyer des données en segments plus petits que la taille du paramètre DATA, la règle suivante s'applique. Il est recommandé de garder le bit EN_R au niveau haut jusqu'à ce que l'instruction TSEND correspondante ait transféré le volume de données approprié pour remplir le paramètre DATA de TRCV. Si la taille des données envoyées par TSEND est différente de la taille du paramètre DATA de TRCV, TRCV reste à l'état occupé (code d'état : 7002) tant que le bit EN_R est au niveau haut et jusqu'à ce que la taille totale des données envoyées par TSEND soit égale à la taille du paramètre DATA de TRCV. Si le bit EN_R de TRCV est généré par impulsion, une impulsion doit y être appliquée un nombre de fois égal au nombre d'exécutions de l'instruction TSEND pour que les données soient reçues.

Le tampon du paramètre DATA de TRCV n'affiche pas les nouvelles données reçues tant que leur taille n'est pas égale à celle du tampon du paramètre DATA.

Dès que toutes les données de la tâche ont été reçues, TRCV les transfère dans la zone de réception et met NDR à 1.

Tableau 11- 25 Entrée des données dans la zone de réception

Variante de protocole	Entrée des données dans la zone de réception	Paramètre "connection_type"	Valeur du paramètre LEN	Valeur du paramètre RCVD_LEN (octets)
TCP	Mode ad hoc	B#16#11	Sélectionné à l'aide de l'instruction TRCV sur l'entrée ADHOC	1 à 1472
TCP	Réception de données avec longueur indiquée	B#16#11	0 (recommandé) ou 1 à 8192, 65535 excepté	1 à 8192
ISO sur TCP	Mode ad hoc	B#16#12	65535	1 à 1472
ISO sur TCP	Gestion par le protocole	B#16#12	0 (recommandé) ou 1 à 8192, 65535 excepté	1 à 8192

Remarque

Mode ad hoc

Le "mode ad hoc" existe avec les variantes de protocole TCP et ISO sur TCP. Pour activer le mode ad hoc de l'instruction TRCV, définissez le paramètre d'entrée de l'instruction ADHOC. La zone de réception est identique à la zone formée par DATA. La longueur des données reçues est transmise au paramètre RCVD_LEN. Tout de suite après la réception d'un bloc de données, TRCV entre les données dans la zone de réception et met NDR à 1.

Si vous stockez les données dans un DB "optimisé" (symbolique uniquement), vous ne pouvez recevoir les données qu'en tableaux de types de données Byte, Char, USInt et SInt.

Remarque

Importation dans le S7-1200 de projets STEP 7 S7-300/400 utilisant le "mode ad hoc"

Dans les projets STEP 7 S7-300/400, le "mode ad hoc" est sélectionné par affectation de 0 au paramètre LEN. Dans le S7-1200, vous activez le mode ad hoc de l'instruction TRCV en activant le paramètre d'entrée ADHOC de l'instruction.

Si vous importez dans le S7-1200 un projet STEP 7 S7-300/400 utilisant le "mode ad hoc", vous devez donner la valeur 65535 au paramètre LEN.

Tableau 11- 26 Codes d'erreur pour les paramètres ERROR et STATUS de TSEND et TRCV

ERROR	STATUS	Description
0	0000	<ul style="list-style-type: none"> Tâche d'émission achevée sans erreur (TSEND) Nouvelles données acceptées : La longueur en cours des données reçues est indiquée dans RCVD_LEN (TRCV).
0	7000	<ul style="list-style-type: none"> Pas de traitement de tâche actif (TSEND) Bloc pas prêt à recevoir (TRCV)
0	7001	<ul style="list-style-type: none"> Démarrage du traitement de la tâche, données en cours d'envoi : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement (TSEND). Bloc prêt à recevoir ; une tâche de réception a été activée (TRCV).
0	7002	<ul style="list-style-type: none"> Exécution intermédiaire de l'instruction (REQ sans objet), tâche en cours de traitement : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement (TSEND). Exécution intermédiaire de l'instruction, tâche de réception en cours de traitement : des données sont écrites dans la zone de réception pendant ce traitement. C'est pourquoi une erreur pourrait être à l'origine de données incohérentes dans la zone de réception (TRCV).
1	8085	<ul style="list-style-type: none"> Le paramètre LEN est supérieur à la plus grande valeur autorisée (TSEND et TRCV). Paramètre LEN ou DATA modifié depuis la première exécution de l'instruction (TRCV).
1	8086	Le paramètre ID ne se situe pas dans la plage d'adresses permise.
1	8088	Le paramètre LEN est plus grand que la zone de mémoire indiquée dans DATA.
1	80A1	Erreur de communication : <ul style="list-style-type: none"> La liaison indiquée n'a pas encore été établie (TSEND et TRCV). La liaison indiquée est en train d'être coupée. Une tâche d'émission ou de réception via cette liaison n'est pas possible (TSEND et TRCV). L'interface est en cours de réinitialisation (TSEND). L'interface est en train de recevoir de nouveaux paramètres (TRCV).
1	80C3	Manque interne de ressources de liaison (Page 854) : Un bloc avec cet ID est déjà en cours de traitement dans une classe de priorité différente.
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none"> La liaison au partenaire de communication ne peut pas être établie en ce moment. L'interface est en train de recevoir de nouveaux paramètres ou la liaison est en cours d'établissement.

Protocoles de communication Ethernet

Chaque CPU comprend un port PROFINET intégré qui prend en charge la communication PROFINET standard. Les instructions TSEND_C, TRCV_C, TSEND et TRCV prennent toutes en charge les protocoles Ethernet TCP et ISO sur TCP.

Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 833)" pour plus d'informations.

11.2.8.10 Instructions d'héritage TCON, TDISCON, TSEND et TRCV

Avant le lancement de STEP 7 V13 SP1 et des CPU S7-1200 V4.1, les instructions TCON, TDISCON, TSEND et TRCV pouvaient fonctionner uniquement avec des paramètres de liaison possédant des structures selon "TCON_Param". Les concepts généraux s'appliquent aux deux jeux d'instructions. Reportez-vous aux instructions d'héritage individuelles TCON, TDISCON, TSEND et TRCV pour obtenir des informations relatives à la programmation.

Sélection de la version des instructions TCON, TDISCON, TSEND et TRCV

Il existe deux versions des instructions TCON, TDISCON, TSEND, ou TRCV disponibles dans STEP 7 :

- Les versions 2.5 et 3.1 étaient disponibles dans STEP 7 Basic/Professional versions V13 ou antérieures.
- La version 4.0 est disponible dans STEP 7 Basic/Professional V13 SP1 ou ultérieur.

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

N'utilisez pas des versions d'instructions différentes dans le même programme CPU.



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.

Open user communication		V3.1
TSEND_C	Send data via Ethernet (TCP)	V2.5
TRCV_C	Receive data via Ethernet (T...	V3.1
TMAIL_C	Send e-mail	V4.0
Others		V2.1
TCON	Establish communication c...	V3.0
TDISCON	Terminate communication ...	V2.1
TSEND	Send data via communicati...	V3.0
TRCV	Receive data via communic...	V3.0

Pour changer la version des instructions TCON, TDISCON, TSEND ou TRCV, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction TCON, TDISCON, TSEND ou TRCV dans votre programme, une nouvelle instance de FB ou FC, en fonction de l'instruction TCON, TDISCON, TSEND ou TRCV sélectionnée, est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB ou FC dans l'arborescence du projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction TCON, TDISCON, TSEND ou TRCV dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB ou FC TCON, TDISCON, TSEND ou TRCV dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction TCON, TDISCON, TSEND ou TRCV.

Instructions d'héritage TCON, TDISCON, TSEND, et TRCV (communication TCP)

Communication Ethernet à l'aide des protocoles TCP et ISO sur TCP

Remarque

Instructions TSEND_C et TRCV_C

Pour simplifier la programmation de la communication PROFINET/Ethernet, l'instruction TSEND_C et l'instruction TRCV_C combinent les fonctionnalités des instructions TCON, TDISCON, TSEND et TRCV :

- TSEND_C combine les instructions TCON, TDISCON et TSEND.
- TRCV_C combine les instructions TCON, TDISCON et TRCV.

Les instructions suivantes gèrent le processus de communication :

- TCON établit une liaison TCP/IP entre un PC client et serveur (CPU).
- TSEND et TRCV envoient et reçoivent les données.
- TDISCON coupe la liaison.

La taille de données minimale que vous pouvez envoyer (TSEND) ou recevoir (TRCV) est de un octet ; la taille maximale est de 8 192 octets. TSEND n'accepte pas l'envoi de données à partir d'adresses booléennes et TRCV ne reçoit pas de données dans des adresses booléennes. Pour plus d'informations sur le transfert de données à l'aide de ces instructions, reportez-vous au paragraphe sur la cohérence des données (Page 204).

TCON, TDISCON, TSEND et TRCV opèrent de manière asynchrone, ce qui signifie que le traitement de la tâche s'étend sur plusieurs exécutions de l'instruction. Par exemple, vous lancez une tâche pour la configuration et l'établissement d'une liaison en exécutant une instruction TCON avec le paramètre REQ = 1. Puis, vous réexécutez TCON pour surveiller l'avancement de la tâche et tester son achèvement au moyen du paramètre DONE.

Le tableau suivant montre les relations entre BUSY, DONE et ERROR. Servez-vous-en pour déterminer l'état de la tâche en cours.

Tableau 11- 27 Interactions entre les paramètres BUSY, DONE et ERROR

BUSY	DONE	ERROR	Description
VRAI	Sans objet	Sans objet	La tâche est en cours de traitement.
FAUX	VRAI	FAUX	La tâche a été achevée avec succès.
FAUX	FAUX	VRAI	La tâche s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS.
FAUX	FAUX	FAUX	Aucune nouvelle tâche n'a été affectée.

TCON et TDISCON

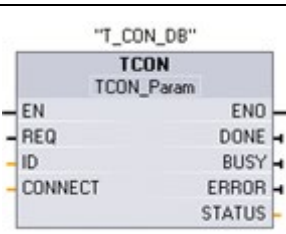
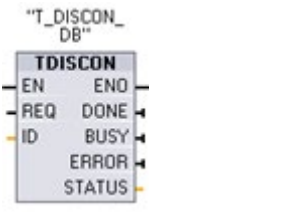
Remarque

Initialisation des paramètres de communication

Une fois que vous avez inséré l'instruction TCON, utilisez les "Propriétés" de l'instruction (Page 833) pour configurer les paramètres de communication (Page 857). Lorsque vous entrez les paramètres pour les partenaires de communication dans la fenêtre d'inspection, STEP 7 entre les données correspondantes dans le DB d'instance pour l'instruction.

Si vous voulez utiliser un DB multi-instance, vous devez configurer le DB manuellement dans les deux CPU.

Tableau 11- 28 Instructions TCON et TDISCON

CONT/LOG		Description
<p>"T_CON_DB"</p>  <p>The diagram shows the TCON instruction logic. It is a ladder logic instruction with the following inputs and outputs: EN (enable), END (end), REQ (request), DONE (done), ID (identifier), BUSY (busy), CONNECT (connect), ERROR (error), and STATUS (status). The instruction is labeled 'TCON' and 'TCON_Param'.</p>	<pre>"TCON_DB" (req:=_bool_in_, ID:=_undef_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, connect:=_struct_inout_);</pre>	<p>TCP et ISO sur TCP : TCON établit une liaison de communication de la CPU vers un partenaire de communication.</p>
<p>"T_DISCON_DB"</p>  <p>The diagram shows the TDISCON instruction logic. It is a ladder logic instruction with the following inputs and outputs: EN (enable), END (end), REQ (request), DONE (done), ID (identifier), BUSY (busy), ERROR (error), and STATUS (status). The instruction is labeled 'TDISCON'.</p>	<pre>"TDISCON_DB" (req:=_bool_in_, ID:=_word_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	<p>TCP et ISO sur TCP : TDISCON met fin à une liaison de communication de la CPU vers un partenaire de communication.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 11- 29 Types de données pour les paramètres de TCON et TDISCON

Paramètre et type		Type de données	Description
REQ	IN	Bool	Le paramètre de commande REQ lance la tâche en établissant la liaison indiquée par ID. La tâche commence en présence d'un front montant.
ID	IN	CONN_OUC (Word)	Référence à la liaison à établir (TCON) ou à interrompre (TDISCON) vers le partenaire éloigné ou entre le programme utilisateur et la couche de communication du système d'exploitation. L'ID doit être identique au paramètre ID associé dans la description de la liaison locale. Plage de valeurs : W#16#0001 à W#16#0FFF
CONNECT (TCON)	IN_OUT	TCON_Param	Pointeur désignant la description de la liaison (Page 857)
DONE	OUT	Bool	<ul style="list-style-type: none"> 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. 1 : Tâche achevée sans erreur
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 : Tâche achevée 1 : La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche.
ERROR	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> 0 : Pas d'erreur 1 : Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)

Les deux partenaires de communication exécutent l'instruction TCON pour configurer et établir la liaison de communication. Vous vous servez de paramètres pour indiquer les partenaires de noeud d'extrémité actif et passif. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.

Si la liaison est interrompue en raison d'une coupure de ligne ou à cause du partenaire de communication éloigné par exemple, le partenaire actif tente de rétablir la liaison configurée. Vous n'avez pas besoin de réexécuter TCON.

Une liaison existante est coupée et la liaison configurée est supprimée lors de l'exécution de l'instruction TDISCON ou lorsque la CPU passe à l'état ARRET. Vous devez réexécuter TCON pour reconfigurer et rétablir la liaison.

Tableau 11- 30 Codes d'erreur pour les paramètres ERROR et STATUS de TCON et TDISCON

ERROR	STATUS	Description
0	0000	La liaison a été établie avec succès.
0	7000	Pas de traitement de tâche actif
0	7001	Démarrage du traitement de la tâche ; établissement de la liaison (TCON) ou interruption de la liaison (TDISCON)
0	7002	Appel intermédiaire (REQ sans objet) ; établissement de la liaison (TCON) ou interruption de la liaison (TDISCON)
1	8086	Le paramètre ID se situe hors de la plage d'adresses permise.
1	8087	TCON : Nombre maximal de liaisons atteint ; aucune liaison supplémentaire n'est possible.
1	809B	TCON : L'identification local_device_id dans la description de la liaison ne correspond pas à la CPU.
1	80A1	TCON : Liaison ou port déjà occupé par un utilisateur
1	80A2	TCON : Port local ou éloigné occupé par le système
1	80A3	Tentative de rétablir une liaison existante (TCON) ou de mettre fin à une liaison inexistante (TDISCON)
1	80A4	TCON : L'adresse IP du nœud d'extrémité distant de la liaison est invalide ; elle correspond à l'adresse IP du partenaire local.
1	80A5	TCON : ID de liaison (Page 854) déjà utilisée
1	80A7	TCON : Erreur de communication : Vous avez appelé TDISCON avant que TCON ne soit achevé. TDISCON doit d'abord couper complètement la liaison référencée par l'ID.
1	80B2	TCON : Le paramètre CONNECT pointe sur un bloc de données qui a été généré avec l'attribut "Sauvegarder uniquement dans la mémoire de chargement".
1	80B4	<p>TCON : Lors de l'utilisation de ISO sur TCP (connection_type = B#16#12) pour établir une liaison passive, le code d'erreur 80B4 vous avertit que le TSAP entré n'était pas conforme à l'une des exigences suivantes eu égard à l'adresse :</p> <ul style="list-style-type: none"> • Pour une longueur TSAP locale égale à 2 et une valeur ID TSAP égale à E0 ou E1 (hexadécimal) pour le premier octet, le deuxième octet doit être 00 ou 01. • Pour une longueur TSAP locale supérieure ou égale à 3 et une valeur ID TSAP égale à E0 ou E1 (hexadécimal) pour le premier octet, le deuxième octet doit être 00 ou 01 et tous les autres octets doivent être des caractères ASCII valides. • Pour une longueur TSAP locale supérieure ou égale à 3 et une valeur ID TSAP différente de E0 et E1 (hexadécimal) pour le premier octet, tous les octets de l'ID TSAP doivent être des caractères ASCII valides. <p>Les caractères ASCII valides ont des valeurs d'octet allant de 20 à 7E (hexadécimal).</p>
1	80B5	TCON : Seul un établissement de liaison passif est autorisé pour le type de liaison "13 = UDP".
1	80B6	TCON : Erreur de paramétrage dans le paramètre CONNECTION_TYPE du SDT TCON_Param

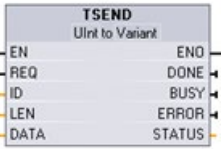
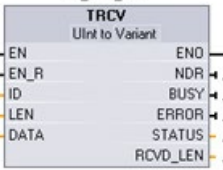
ERROR	STATUS	Description
1	80B7	<p>TCON : Erreur dans l'un des paramètres suivants du bloc de données servant à la description de la liaison :</p> <ul style="list-style-type: none"> • block_length • local_tsap_id_len • rem_subnet_id_len • rem_staddr_len • rem_tsap_id_len • next_staddr_len <p>Remarque : Lors du fonctionnement de TCON en mode passif TCP, le LOCAL_TSAP_ID_LEN doit être de "2" et le REM_TSAP_ID_LEN doit être de "0".</p>
1	80B8	TCON : Le paramètre dans la description de liaison locale et le paramètre ID sont différents.
1	80C3	TCON : Toutes les ressources de liaison sont utilisées.
1	80C4	<p>Erreur de communication temporaire :</p> <ul style="list-style-type: none"> • La liaison ne peut pas être établie en ce moment (TCON). • La liaison configurée est en cours de suppression par TDISCON (TCON). • La liaison est en cours d'établissement (TDISCON). • L'interface est en train de recevoir de nouveaux paramètres (TCON et TDISCON).

TSEND et TRCV

Remarque

Lors de l'utilisation de la communication PROFINET Open User, si vous exécutez une opération TSEND sans qu'une instruction correspondante TRCV ne soit exécutée sur l'appareil distant, l'instruction TSEND peut rester indéfiniment "occupée", attendant que l'instruction TRCV reçoive les données. Dans cet état, la sortie "Busy" de l'instruction TSEND est mise à 1, et la sortie "Status" a la valeur "0x7002". Cette condition peut se produire si vous transférez plus de 4 096 octets de données. Ce problème est résolu à la nouvelle exécution de l'instruction TRCV.

Tableau 11- 31 Instructions TSEND et TRCV

CONT/LOG	SCL	Description
<p>"T_SEND_DB_1"</p>  <p>The diagram shows a data block for the TSEND instruction. It is titled "TSEND" and "UInt to Variant". It contains the following fields: EN, END, REQ, DONE, ID, BUSY, LEN, ERROR, and DATA, STATUS.</p>	<pre>"TSEND_DB" (req:=_bool_in_, ID:=_word_in_, len:=_udint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, data:=_variant_inout_);</pre>	<p>TCP et ISO sur TCP : TSEND envoie des données par le biais d'une liaison de communication de la CPU vers une station partenaire.</p>
<p>"T_RCV_DB"</p>  <p>The diagram shows a data block for the TRCV instruction. It is titled "TRCV" and "UInt to Variant". It contains the following fields: EN, EN_R, NDR, ID, BUSY, LEN, ERROR, DATA, STATUS, RCV_LEN.</p>	<pre>"TRCV_DB" (en_r:=_bool_in_, ID:=_word_in_, len:=_udint_in_, ndr=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, rcvd_len=>_udint_out_, data:=_variant_inout_);</pre>	<p>TCP et ISO sur TCP : TRCV reçoit des données par le biais d'une liaison de communication d'une station partenaire vers la CPU.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 11- 32 Types de données pour les paramètres de TSEND et TRCV

Paramètre et type		Type de données	Description
REQ	IN	Bool	TSEND : Lance la tâche d'émission en présence d'un front montant. Les données sont prises dans la zone indiquée par DATA et LEN.
EN_R	IN	Bool	TRCV : Valide la réception par la CPU ; lorsque EN_R = 1, TRCV est prêt à recevoir. La tâche de réception est traitée.
ID	IN	CONN_OUC (Word)	Référence à la liaison associée. L'ID doit être identique au paramètre ID associé dans la description de la liaison locale. Plage de valeurs : W#16#0001 à W#16#0FFF
LEN	IN	UInt	Nombre maximum d'octets à envoyer (TSEND) ou à recevoir (TRCV) : <ul style="list-style-type: none"> Valeur par défaut = 0 : Le paramètre DATA détermine la longueur des données à envoyer (TSEND) ou à recevoir (TRCV). Mode ad hoc = 65535 : Une longueur de données variable est définie pour la réception (TRCV).
DATA	IN_OUT	Variante	Pointeur désignant la zone de données à envoyer (TSEND) ou à recevoir (TRCV). La zone de données contient l'adresse et la longueur. L'adresse fait référence à de la mémoire I, de la mémoire Q, de la mémoire M ou à un DB.
DONE	OUT	Bool	TSEND : <ul style="list-style-type: none"> 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. 1 : Tâche exécutée sans erreur
NDR	OUT	Bool	TRCV : <ul style="list-style-type: none"> NDR = 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. NDR = 1 : La tâche a été achevée avec succès.
BUSY	OUT	Bool	<ul style="list-style-type: none"> BUSY = 1 : La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche. BUSY = 0 : La tâche est achevée.
ERROR	OUT	Bool	ERROR = 1 : Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)
RCVD_LEN	OUT	Int	TRCV : Volume de données effectivement reçu en octets

Remarque

L'instruction TSEND requiert un front montant du paramètre d'entrée REQ pour démarrer une tâche d'émission. Le paramètre BUSY est mis à 1 pendant le traitement. L'achèvement de la tâche d'émission est signalé par la mise à 1 du paramètre DONE ou ERROR pendant un cycle. Pendant ce temps, aucun front montant du paramètre d'entrée REQ n'est pris en compte.

Fonctionnement de TRCV

L'instruction TRCV écrit les données reçues dans une zone de réception qui est définie par les deux variables suivantes :

- Pointeur désignant le début de la zone
- Longueur de la zone ou valeur fournie à l'entrée LEN si différente de 0

Remarque

Avec la valeur par défaut du paramètre LEN (LEN = 0), le paramètre DATA est utilisé pour déterminer la longueur des données transmises. Il est recommandé que les données envoyées par l'instruction TSEND aient la même taille que le paramètre DATA de l'instruction TRCV.

Si, en cas d'utilisation de la valeur par défaut du paramètre LEN, il s'avère nécessaire d'envoyer des données en segments plus petits que la taille du paramètre DATA, la règle suivante s'applique. Il est recommandé de garder le bit EN_R au niveau haut jusqu'à ce que l'instruction TSEND correspondante ait transféré le volume de données approprié pour remplir le paramètre DATA de TRCV. Si la taille des données envoyées par TSEND est différente de la taille du paramètre DATA de TRCV, TRCV reste à l'état occupé (code d'état : 7002) tant que le bit EN_R est au niveau haut et jusqu'à ce que la taille totale des données envoyées par TSEND soit égale à la taille du paramètre DATA de TRCV. Si le bit EN_R de TRCV est généré par impulsion, une impulsion doit y être appliquée un nombre de fois égal au nombre d'exécutions de l'instruction TSEND pour que les données soient reçues.

Le tampon du paramètre DATA de TRCV n'affiche pas les nouvelles données reçues tant que leur taille n'est pas égale à celle du tampon du paramètre DATA.

Dès que toutes les données de la tâche ont été reçues, TRCV les transfère dans la zone de réception et met NDR à 1.

Tableau 11- 33 Entrée des données dans la zone de réception

Variante de protocole	Entrée des données dans la zone de réception	Paramètre "connection_type"	Valeur du paramètre LEN	Valeur du paramètre RCVD_LEN (octets)
TCP	Mode ad hoc	B#16#11	65535	1 à 1472
TCP	Réception de données avec longueur indiquée	B#16#11	0 (recommandé) ou 1 à 8192, 65535 excepté	1 à 8192
ISO sur TCP	Mode ad hoc	B#16#12	65535	1 à 1472
ISO sur TCP	Gestion par le protocole	B#16#12	0 (recommandé) ou 1 à 8192, 65535 excepté	1 à 8192

Remarque

Mode ad hoc

Le "mode ad hoc" existe avec les variantes de protocole TCP et ISO sur TCP. Vous paramétrez le "mode ad hoc" en affectant "65535" au paramètre LEN. La zone de réception est identique à la zone formée par DATA. La longueur des données reçues est transmise au paramètre RCVD_LEN. Tout de suite après la réception d'un bloc de données, TRCV entre les données dans la zone de réception et met NDR à 1.

Si vous stockez les données dans un DB "optimisé" (symbolique uniquement), vous ne pouvez recevoir les données qu'en tableaux de types de données Byte, Char, USInt et SInt.

Remarque

Importation dans le S7-1200 de projets STEP 7 S7-300/400 utilisant le "mode ad hoc"

Dans les projets STEP 7 S7-300/400, le "mode ad hoc" est sélectionné par affectation de 0 au paramètre LEN. Dans le S7-1200, vous paramétrez le "mode ad hoc" en affectant "65535" au paramètre LEN.

Si vous importez dans le S7-1200 un projet STEP 7 S7-300/400 utilisant le "mode ad hoc", vous devez donner la valeur 65535 au paramètre LEN.

Codes d'erreur de TSEND , TRCV Error et Status

ERROR	STATUS	Description
0	0000	<ul style="list-style-type: none"> Tâche d'émission achevée sans erreur (TSEND) Nouvelles données acceptées : La longueur en cours des données reçues est indiquée dans RCVD_LEN (TRCV).
0	7000	<ul style="list-style-type: none"> Pas de traitement de tâche actif (TSEND) Bloc pas prêt à recevoir (TRCV)
0	7001	<ul style="list-style-type: none"> Démarrage du traitement de la tâche, données en cours d'envoi : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement (TSEND). Bloc prêt à recevoir ; une tâche de réception a été activée (TRCV).
0	7002	<ul style="list-style-type: none"> Exécution intermédiaire de l'instruction (REQ sans objet), tâche en cours de traitement : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement (TSEND). Exécution intermédiaire de l'instruction, tâche de réception en cours de traitement : des données sont écrites dans la zone de réception pendant ce traitement. C'est pourquoi une erreur pourrait être à l'origine de données incohérentes dans la zone de réception (TRCV).
1	8085	<ul style="list-style-type: none"> Le paramètre LEN est supérieur à la plus grande valeur autorisée (TSEND et TRCV). Paramètre LEN ou DATA modifié depuis la première exécution de l'instruction (TRCV).
1	8086	Le paramètre ID ne se situe pas dans la plage d'adresses permise.
1	8088	Le paramètre LEN est plus grand que la zone de mémoire indiquée dans DATA.
1	80A1	Erreur de communication : <ul style="list-style-type: none"> La liaison indiquée n'a pas encore été établie (TSEND et TRCV). La liaison indiquée est en train d'être coupée. Une tâche d'émission ou de réception via cette liaison n'est pas possible (TSEND et TRCV). L'interface est en cours de réinitialisation (TSEND). L'interface est en train de recevoir de nouveaux paramètres (TRCV).
1	80C3	Manque interne de ressources : Un bloc avec cet ID est déjà en cours de traitement dans une classe de priorité différente.
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none"> La liaison au partenaire de communication ne peut pas être établie en ce moment. L'interface est en train de recevoir de nouveaux paramètres ou la liaison est en cours d'établissement.

Protocoles de communication Ethernet

Chaque CPU comprend un port PROFINET intégré qui prend en charge la communication PROFINET standard. Les instructions TSEND_C, TRCV_C, TSEND et TRCV prennent toutes en charge les protocoles Ethernet TCP et ISO sur TCP.

Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 833)" pour plus d'informations.

11.2.8.11 Instruction T_RESET (Couper et rétablir une liaison existante)

L'instruction "T_RESET" s'achève puis réétablit une connexion existante :

Tableau 11- 34 Instruction T_RESET

CONT/LOG	SCL	Description
	<pre>"T_RESET_DB" (req:=_bool_in_, id:=_word_in_, done=>_bool_out_, error=>_bool_out_, sta- tus=>_word_out_);</pre>	<p>Utilisez l'instruction "T_RESET" pour achever puis rétablir une connexion existante.</p>

Les nœuds d'extrémité locaux de la liaison sont conservés. Ils sont générés automatiquement :

- Si une liaison a été configurée et chargée dans la CPU ;
- Si une liaison a été générée par le programme utilisateur, par appel de l'instruction "TCON (Page 885)" par exemple.

L'instruction "T_RESET" peut être exécutée pour tous les types de liaisons, que l'interface locale de la CPU ou que l'interface d'un module ou d'un processeur de communication ait été utilisée pour la liaison. Les liaisons pour les transferts de données en mode ad hoc avec TCP constituent toutefois une exception, puisque de telles liaisons ne peuvent pas être référencées avec un ID de liaison.

Une fois appelée au moyen du paramètre REQ, l'instruction "T_RESET" met fin à la liaison indiquée dans le paramètre ID et efface les mémoires tampons d'émission et de réception des données si nécessaire. L'annulation de la liaison annule également toute transmission de données en cours. Il y a donc un risque de perdre des données si une transmission est en cours. La CPU définie comme partenaire de liaison actif tente alors automatiquement de restaurer la liaison de communication interrompue. Par conséquent, vous n'avez pas besoin d'appeler l'instruction "TCON (Page 885)" pour rétablir la connexion de communication.

Les paramètres de sortie DONE, BUSY et STATUS indiquent l'état de la tâche.

Types de données pour les paramètres

Le tableau suivant présente les paramètres de l'instruction T_RESET :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L, T, C ou constante	Le paramètre de commande REQUEST démarre la tâche d'annulation de la liaison indiquée dans ID. La tâche démarre en présence d'un front montant.
ID	Input	CONN_OUC (WORD)	L, D ou constante	Référence à la liaison vers le partenaire passif à laquelle il faut mettre fin. L'ID doit être identique à l'ID du paramètre correspondant dans la description de la liaison locale. Plage de valeurs : W#16#0001 à W#16#0FFF
DONE	Output	BOOL	I, Q, M, D, L	Paramètre d'état DONE <ul style="list-style-type: none"> 0 : 0 : tâche pas encore commencée ou encore en cours d'exécution. 1 : Tâche exécutée sans erreur.
BUSY	Output	BOOL	I, Q, M, D, L	Paramètre d'état BUSY <ul style="list-style-type: none"> 0 : 0 : tâche achevée 1 : Tâche pas encore achevée.
ERROR	Output	BOOL	I, Q, M, D, L	Paramètre d'état ERROR <ul style="list-style-type: none"> 0 : Pas d'erreur. 1 : Une erreur s'est produite pendant le traitement. Le paramètre STATUS contient des informations détaillées sur le type d'erreur.
STATUS	Output	WORD	I, Q, M, D, L	Paramètre d'état STATUS Informations d'erreur (voir le tableau "Paramètre STATUS").


Paramètre STATUS

Bit d'erreur	STATUS* (W#16#...)	Description
0	0000	Pas d'erreur.
0	0001	La liaison n'a pas été établie.
0	7001	La coupure de la liaison a été lancée.
0	7002	Coupure de la liaison en cours.
1	8081	Liaison inconnue indiquée dans le paramètre ID.

11.2.8.12 Instruction T_DIAG (Vérifie l'état d'une liaison et lire des informations)

L'instruction "T_DIAG" permet de vérifier l'état d'une liaison et de lire d'autres informations sur le nœud d'extrémité local de cette liaison :

Tableau 11- 35 Instruction T_DIAG

CONT/LOG	SCL	Description
	<pre>"T_DIAG_DB" (req:=_bool_in_, id:=_word_in_, done=>_bool_out_, error=>_bool_out_, sta- tus=>_dword_out_);</pre>	<p>Utilisez l'instruction "T_DIAG" pour vérifier l'état d'une liaison et lire d'autres informations sur le nœud d'extrémité local de cette liaison.</p>

L'instruction "T_DIAG" fonctionne comme suit :

- La liaison est référencée avec le paramètre ID. Vous pouvez lire aussi bien des nœuds d'extrémité de liaison configurés dans l'éditeur de liaison que ceux qui ont été programmés (par ex. avec l'instruction "TCON").

Les nœuds d'extrémité de liaison temporaires (créés par exemple lors de la connexion d'une station d'ingénierie) ne peuvent pas faire l'objet d'un diagnostic car aucune ID de liaison n'est créée dans ce cas.

- Les informations de liaison lues sont placées dans une structure référencée avec le paramètre RESULT.
- Le paramètre de sortie STATUS indique si les informations de liaison ont pu être lues. Les informations de liaison dans la structure au paramètre RESULT sont valides uniquement si l'instruction "T_DIAG" a été terminée avec STATUS = W#16#0000 et ERROR = FALSE.

En cas d'erreur, les informations de liaison ne peuvent pas être interprétées.

Informations de liaison possibles

La structure "TDiag_Status" peut être utilisée pour lire les informations de liaison au paramètre RESULT. La structure TDiag_Status contient uniquement les informations les plus importantes concernant un nœud d'extrémité de liaison (par exemple, le protocole utilisé, l'état de liaison ou le nombre de données envoyées ou reçues en octets).

La composition et les paramètres de la structure TDiag_Status sont décrits plus bas (voir le tableau "Structure TDIAG_Status").

Types de données pour les paramètres

Le tableau suivant présente les paramètres de l'instruction T_DIAG :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L, T, C ou constante	Lance l'instruction de contrôle de la liaison indiquée au paramètre ID en cas de front montant.
ID	Input	CONN_OUC (WORD)	L, D ou constante	Référence à la liaison affectée. Plage de valeurs : W#16#0001 à W#16#0FFF
RESULT	InOut	VARIANT	D	Pointeur désignant la structure dans laquelle les informations de liaison sont archivées. La structure TDiag_Status peut être utilisée au paramètre RESULT (pour une description, voir le tableau "Structure TDIAG_Status").
DONE	Output	BOOL	I, Q, M, D, L	Paramètre d'état : <ul style="list-style-type: none"> 0 : L'instruction n'a pas encore été démarrée ou elle est encore en cours d'exécution. 1 : Instruction exécutée sans erreur.
BUSY	Output	BOOL	I, Q, M, D, L	Paramètre d'état : <ul style="list-style-type: none"> 0 : L'instruction n'a pas encore été démarrée ou elle est déjà achevée. 1 : L'instruction n'est pas encore achevée. Il n'est pas possible de démarrer une nouvelle tâche.
ERROR	Output	BOOL	I, Q, M, D, L	Paramètre d'état : <ul style="list-style-type: none"> 0 : Pas d'erreur. 1 : Une erreur s'est produite.
STATUS	Output	WORD	I, Q, M, D, L	Etat de l'instruction

Paramètres BUSY, DONE et ERROR

Vous pouvez contrôler l'état d'exécution de l'instruction "T_DIAG" au moyen des paramètres BUSY, DONE, ERROR et STATUS. Le paramètre BUSY indique l'état du traitement. Avec le paramètre DONE, vous vérifiez si l'instruction s'est exécutée correctement. Le paramètre ERROR est mis à 1 si des erreurs se sont produites pendant l'exécution de "T_DIAG".

Le tableau suivant montre les relations entre les paramètres BUSY, DONE et ERROR :

BUSY	DONE	ERROR	Description
1	-	-	L'instruction est en cours de traitement.
0	1	0	L'instruction a été correctement exécutée. Les données dans la structure référencée avec RESULT sont valides uniquement dans ce cas.
0	0	1	L'instruction s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS.
0	0	0	Aucune nouvelle instruction n'a été affectée.

Paramètre STATUS

Le tableau suivant donne la signification des valeurs du paramètre STATUS :

Bit d'erreur	STATUS* (W#16#...)	Description
0	0000	L'instruction "T_DIAG" a été correctement exécutée. Les données dans la structure référencée au paramètre RESULT peuvent être interprétées.
0	7000	Aucun traitement d'instruction actif.
0	7001	Traitement d'instruction lancé.
0	7002	Les informations de liaison sont en train d'être lues (paramètre REQ non pertinent).
1	8086	La valeur du paramètre ID n'est pas dans la plage admise (W#16#0001 ... W#16#0FFF).
1	8089	Le paramètre RESULT pointe sur un type de données non valide (seules les structures TDIAG_Status et TDIAG_StatusExt sont admises).
1	80A3	Le paramètre ID référence un nœud d'extrémité de liaison qui n'existe pas. Cette erreur peut également survenir après l'appel de l'instruction "TDISCON" avec les liaisons programmées.
1	80C4	Erreur interne. L'accès au nœud d'extrémité de liaison est temporairement impossible.

Structure TDIAG_Status

Le tableau suivant décrit dans les détails la composition de la structure TDIAG_Status. Tous les éléments n'ont de valeur valide que si l'instruction a pu être exécutée sans erreur. En cas d'erreur, le contenu des paramètres reste inchangé :

Nom	Type de données	Description
Les paramètres suivants existent pour la structure TDIAG_Status :		
InterfaceID	HW_ANY	ID d'interface (LADDR) de la CPU ou du CM/CP.
ID	CONN_OUC	ID de la liaison diagnostiquée. Après un appel réalisé avec succès, la valeur de cet élément est identique au paramètre ID de l'instruction "T_DIAG".
ConnectionType	BYTE	Type de protocole utilisé pour la liaison : <ul style="list-style-type: none"> • 0x01 : non utilisé. • ... • 0x0B : protocole TCP (IP_v4) • 0x0C : protocole ISO-on-TCP (RFC1006) • 0x0D : protocole TCP (DNS) • 0x0E : protocole Dial in • 0x0F : protocole WDC • 0x10 : protocole SMTP • 0x11 : protocole TCP • 0x12 : protocole TCP et ISO-on-TCP (RFC1006) • 0x13 : protocole UDP • 0x14 : réservé • 0x15 : protocole d'accès au bus PROFIBUS (FDL) • 0x16 : protocole de transport ISO 8073 (ISONative) • ... • 0x20 : protocole SMTP ou SMTPS basé sur IPv4 • 0x21 : protocole SMTP ou SMTPS basé sur IPv6 • 0x22 : protocole SMTP ou SMTPS basé sur FQDN (Fully Qualified Domain Name) • ... • 0x70 : liaison S7 • Autres : réservés
ActiveEstablished	BOOL	<ul style="list-style-type: none"> • FALSE: Localement, le point de fin de connexion passif • TRUE: Localement, le point de fin de connexion actif

Nom	Type de données	Description
State	BYTE	<p>Etat actuel du nœud d'extrémité de liaison</p> <ul style="list-style-type: none"> • 0x00 : non utilisé. • 0x01 : liaison coupée. Etat temporaire pris par exemple après l'appel de l'instruction "T_RESET". Le système essaie ensuite de rétablir automatiquement la connexion. • 0x02 : le nœud d'extrémité de liaison actif tente d'établir la liaison avec le partenaire de communication distant. • 0x03 : le nœud d'extrémité de liaison passif attend l'établissement de la liaison du partenaire de communication distant. • 0x04 : liaison établie. • 0x05 : la liaison est en cours d'interruption. La coupure peut être due à l'appel de l'instruction "T_RESET" ou "T_DISCON". D'autres causes possibles sont une erreur de protocole ou une rupture de fil. • 0x06..0xFF : non utilisé.
Kind	BYTE	<p>Mode de fonctionnement du nœud d'extrémité de liaison :</p> <ul style="list-style-type: none"> • 0x00 : non utilisé. • 0x01 : liaison statique configurée ayant fait l'objet d'une configuration puis d'un chargement sur la CPU. • 0x02 : liaison dynamique configurée ayant fait l'objet d'une configuration puis d'un chargement sur la CPU (non pris en charge pour le moment). • 0x03 : liaison programmée établie à partir du programme utilisateur avec l'instruction "TCON". Le nœud d'extrémité de liaison a été détruit par un appel de l'instruction "TDISCON" ou un passage de la CPU à l'état ARRET. • 0x04 : Temporairement, la connexion dynamique établie par la station d'ingénierie (ES) ou par la station opérateur (OS), par exemple. (ce type de connexion ne peut actuellement pas être diagnostiqué car il n'y a pas d'ID). • 0x05..0xFF : non utilisé.
SentBytes	UDINT	Nombre d'octets de données envoyés.
ReceivedBytes	UDINT	Nombre d'octets de données reçus.

11.2.8.13 Instruction TMAIL_C (Envoyer un courriel à l'aide de l'interface Ethernet de la CPU)

Présentation

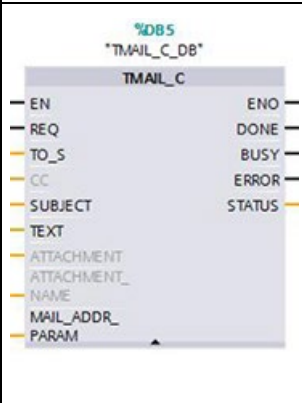
Utilisez l'instruction "TMAIL_C" pour envoyer un courriel au moyen de l'interface Ethernet de la CPU S7-1200.

L'instruction TMAIL_C possède deux fonctionnalités :

- Courriel sur l'interface de la CPU (uniquement SMTP sans SSL)
- Courriel sur une interface CP (soit SMTP sans SSL ou SMTP avec SSL). Si vous souhaitez utiliser la fonctionnalité SSL, vous devez définir le paramètre d'entrée TMAIL_C CERTINDEX = 1 et utiliser l'interface CP. De plus, le certificat correct doit être sauvegardé dans le stockage des certificats CP.

Cette instruction ne peut être utilisée qu'une fois le matériel configuré et si l'infrastructure du réseau autorise une liaison de communication avec le serveur de messagerie.

Tableau 11- 36 Instruction TMAIL_C

CONT/LOG	SCL	Description
	<pre>"TMAIL_C_DB" (req:=_bool_in_, to_s:=_string_in_, cc:=_string_in_, subject:=_string_in_, text:=_string_in_, attachment:=_variant_in_, attachment_name:=_string_in_, mail_addr_param:=_string_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	<p>L'instruction "TMAIL_C" permet d'envoyer un courriel au moyen de l'interface Ethernet de la CPU S7-1200.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Vous définissez le contenu du courriel et les données de liaison au moyen des paramètres suivants :

- Vous définissez les adresses des destinataires avec les paramètres TO_S et CC.
- Vous définissez le contenu du courriel avec les paramètres SUBJECT et TEXT.
- Vous pouvez définir une pièce jointe à l'aide de pointeurs VARIANT dans les paramètres ATTACHMENT et ATTACHMENT_NAME.
- Vous définissez les données de la liaison et vous effectuez l'adressage et l'authentification pour le serveur de messagerie à l'aide du type de données système Tmail_v4 ou Tmail_FQDN dans le paramètre MAIL_ADDR_PARAM. Si vous vous servez de l'interface de la CPU S7-1200, il faut utiliser le type de données système Tmail_v4. Dans ce cas, le courriel peut uniquement être envoyé via SMTP.

- L'envoi d'un courriel démarre en présence d'un front montant (de 0 à 1) dans le paramètre REQ.
- L'état de la tâche est indiqué par les paramètres de sortie "BUSY", "DONE", "ERROR" et "STATUS".

L'instruction "TMAIL_C" ne permet pas d'envoyer de SMS directement. La possibilité ou non de transmission du courriel en tant que SMS par le serveur de messagerie dépend de votre fournisseur de services de télécommunication.

Fonctionnement de l'instruction

L'instruction "TMAIL_C" fonctionne de manière asynchrone, ce qui signifie que son exécution s'étend sur plusieurs cycles. Vous devez indiquer une instance lorsque vous appelez l'instruction "TMAIL_C".

La connexion au serveur de messagerie est perdue dans les cas suivants :

- Si la CPU passe à l'état ARRET alors que l'instruction "TMAIL_C" est active.
- Si des problèmes de communication se produisent sur le bus Industrial Ethernet. Dans ce cas, la transmission du courriel sera interrompue et le courriel n'atteindra pas son destinataire.

La connexion est également coupée une fois l'instruction exécutée avec succès et le courriel envoyé.

IMPORTANT

Modification du programme utilisateur

Vous pouvez modifier les parties de votre programme utilisateur qui affectent directement les appels de TMAIL_C uniquement lorsque :

- La CPU est à l'état ARRET.
- Aucun courriel n'est en cours d'envoi (REQ = 0 et BUSY = 0).

Cela vaut, en particulier, pour la suppression ou le remplacement de blocs de programme qui contiennent des appels de TMAIL_C ou des appels de l'instance de TMAIL_C.

Des ressources de liaison risquent d'être mobilisées inutilement si vous ne tenez pas compte de cette restriction. Le système d'automatisation peut passer dans un état non défini avec les fonctions de communication TCP/IP via Industrial Ethernet.

Un démarrage à chaud ou à froid de la CPU est nécessaire une fois les modifications transférées.

Cohérence des données

L'instruction TMAIL_C applique les paramètres TO_S, CC, SUBJECT, TEXT, ATTACHMENT et MAIL_ADDR_PARAM alors qu'elle est en cours d'exécution, ce qui signifie qu'il ne faut pas modifier ces paramètres avant que la tâche ne soit achevée (BUSY = 0).

Authentification SMTP

L'authentification désigne une procédure de vérification d'identité, en demandant un mot de passe par exemple.

Si vous utilisez l'interface de la CPU S7-1200, l'instruction "TMAIL_C" prend en charge la procédure d'authentification SMTP AUTH-LOGIN qui est exigée par la plupart des serveurs de messagerie. Pour plus d'informations sur la procédure d'authentification utilisée par votre serveur de messagerie, consultez le manuel de votre serveur ou le site Web de votre fournisseur de services Internet.

- Pour que vous puissiez utiliser la procédure d'authentification AUTH-LOGIN, l'instruction "TMAIL_C" demande le nom d'utilisateur avec lequel elle doit ouvrir une session sur le serveur de messagerie. Ce nom d'utilisateur correspond au nom d'utilisateur avec lequel vous avez configuré un compte de messagerie sur votre serveur de messagerie. Il est transmis par le paramètre UserName à la structure dans le paramètre MAIL_ADDR_PARAM.

La procédure d'authentification AUTH-LOGIN n'est pas utilisée si aucun nom d'utilisateur ne figure dans le paramètre MAIL_ADDR_PARAM. Le courriel est alors envoyé sans authentification.

- Pour ouvrir une session, l'instruction "TMAIL_C" demande également le mot de passe associé. Ce mot de passe correspond au mot de passe que vous avez défini lorsque vous avez configuré votre compte de messagerie. Il est transmis par le biais du paramètre PassWord à la structure dans le paramètre MAIL_ADDR_PARAM.

Types de données pour les paramètres

Le tableau suivant présente les paramètres de l'instruction TMAIL_C :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L, T, C ou constante	Paramètre de commande REQUEST : Active l'envoi d'un courriel en présence d'un front montant.
TO_S (Page 920)	Input	STRING	D	Adresses des destinataires. Chaîne STRING avec 180 caractères (octets) au maximum. Pour le format de l'adresse électronique, voir l'exemple dans la description des paramètres.
CC (Page 920)	Input	STRING	D	Adresses de copie conforme (facultatives). Chaîne STRING avec 180 caractères (octets) au maximum. Même format d'adresse électronique que pour le paramètre TO_S. Si une chaîne vide est affectée ici, le courriel n'est pas envoyé à des destinataires CC.
SUBJECT	Input	STRING	D	Objet du courriel. Chaîne STRING avec 180 caractères (octets) au maximum.
TEXT	Input	STRING	D	Texte du courriel (facultatif). Chaîne STRING avec 180 caractères (octets) au maximum. Si une chaîne vide est affectée à ce paramètre, le courriel est envoyé sans texte.
ATTACHMENT	Input	VARIANT	D	Pièce jointe (facultatif) Référence à un tableau d'octets/mots/doubles mots (ArrayOfByte, ArrayOfWord ou ArrayOfDWord) ayant une longueur maximale de 64 Ko. Si vous n'affectez pas de valeur à ce paramètre, le courriel est envoyé sans pièce jointe.
ATTACHMENT_NAME	Input	VARIANT	D	Nom de la pièce jointe (facultatif) Référence à une chaîne de 50 caractères (octets) au maximum pour définir le nom de fichier de la pièce jointe. Si une chaîne vide est affectée à ce paramètre, la pièce jointe sera envoyée sous le nom de fichier "attachment.bin".

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
MAIL_ADDR_P ARAM (Page 918)	Input	VARIANT	D	Paramètres de liaison et adresse du serveur de messagerie Vous définissez les paramètres de liaison au moyen de la structure Tmail_v4 ou Tmail_FQDN (voir la description des paramètres).
DONE (Page 920)	Output	BOOL	I, Q, M, D, L	Paramètre d'état <ul style="list-style-type: none"> DONE = 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. DONE = 1 : Tâche exécutée sans erreur
BUSY (Page 920)	Output	BOOL	I, Q, M, D, L	Paramètre d'état <ul style="list-style-type: none"> BUSY = 0 : Le traitement de "TMAIL_C" s'est arrêté. BUSY = 1 : La transmission du courriel n'est pas encore achevée.
ERROR (Page 920)	Output	BOOL	I, Q, M, D, L	Paramètre d'état <ul style="list-style-type: none"> ERROR = 0 : Pas d'erreur ERROR = 1 : Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.
STATUS (Page 921)	Output	WORD	I, Q, M, D, L	Paramètre d'état Valeur ou information d'erreur renvoyée par l'instruction "TMAIL_C" (voir la description des paramètres).

Vous trouverez des informations plus détaillées sur les types de données valides au paragraphe "Vue d'ensemble des types de données valides".

Remarque

Paramètres facultatifs

Les paramètres facultatifs CC, TEXT et ATTACHMENT sont envoyés avec le courriel uniquement si les paramètres correspondants contiennent une chaîne non vide.

Paramètre MAIL_ADDR_PARAM

Dans le paramètre MAIL_ADDR_PARAM, vous définissez la liaison pour l'envoi du courriel au moyen de la structure Tmail_v4 ou Tmail_FQDN et vous sauvegardez l'adresse du serveur de messagerie ainsi que les identifiants.

La structure que vous utilisez dans le paramètre MAIL_ADDR_PARAM dépend du format dans lequel vous devez accéder au serveur de messagerie :

- Tmail_v4 : accès par adresse IP (IPv4).
- Tmail_FQDN : accès par nom de domaine complet (FQDN).

La structure que vous pouvez utiliser dépend de l'interface indiquée dans le paramètre Interfaceld. Si vous voulez utiliser l'instruction "TMAIL_C" avec l'interface interne, il faut utiliser la structure Tmail_v4 dans le paramètre MAIL_ADDR_PARAM.

Tableau 11- 37 Tmail_v4: accès au serveur de messagerie par l'adresse IP (IPv4)

Paramètre	Type de données	Description
Tmail_v4	Struct	
Interfaceld	LADDR	Identificateur matériel de l'interface
ID	CONN_OUC	ID de liaison
ConnectionType	BYTE	Type de liaison. Sélectionnez 16#20 comme type de liaison pour IPv4.
ActiveEstablished	BOOL	Bit d'état. Mis à 1 une fois la liaison établie.
CertIndex	BYTE	=0 : SMTP (S imple M ail T ransfer P rotocol) utilisé. SMTP doit être utilisé si le courriel est envoyé via l'interface d'une CPU S7-1200.
WatchDogTime	TIME	Chien de garde pour l'exécution. Utilisez ce paramètre pour définir le temps d'exécution maximum pour l'émission. Remarque : l'établissement de la liaison peut durer plus longtemps que prévu (une minute environ) si la connexion est lente. Si vous définissez le paramètre WATCH_DOG_TIME, n'oubliez pas de prévoir le temps nécessaire à l'établissement de la liaison. La liaison est coupée une fois le temps indiqué écoulé.
MailServerAddress	IP_v4	Adresse IP du serveur de messagerie. IPv4 au format suivant : XXX.XXX.XXX.XXX (décimal). Exemple : 192.142.131.237.
UserName	STRING[254]	Identifiant pour le serveur de messagerie
PassWord	STRING[254]	Mot de passe pour le serveur de messagerie
From	EMAIL_ADDR	Adresse de l'expéditeur du courriel, définie au moyen des deux paramètres suivants de type STRING. Exemple : "myname@mymailserver.com"
LocalPartPlusAtSign	STRING[64]	Partie locale de l'adresse de l'expéditeur, signe @ compris. Exemple : "myname@"
FullQualified-DomainName	STRING[254]	Nom de domaine complet (FQDN : Fully Qualified Domain Name) du serveur de messagerie. Exemple : "mymail-server.com"

Tableau 11- 38 Tmail_FQDN : accès au serveur de messagerie par nom de domaine complet FQDN

Paramètre	Type de données	Description
Tmail_v6	Struct	
Tmail_FQDN	LADDR	Identificateur matériel de l'interface
ID	CONN_OUC	ID de liaison
ConnectionType	BYTE	Type de liaison. Sélectionnez 16#22 comme type de liaison pour FQDN.
ActiveEstablished	BOOL	Bit d'état. Mis à 1 une fois la liaison établie.
CertIndex	BYTE	=0 : SMTP (S imple M ail T ransfer P rotocol) utilisé. SMTP doit être utilisé si le courriel est envoyé via l'interface d'une CPU S7-1200.
WatchDogTime	TIME	Chien de garde pour l'exécution. Utilisez ce paramètre pour définir le temps d'exécution maximum pour l'émission. Remarque : l'établissement de la liaison peut durer plus longtemps que prévu (une minute environ) si la connexion est lente. Si vous définissez le paramètre WATCH_DOG_TIME, n'oubliez pas de prévoir le temps nécessaire à l'établissement de la liaison. La liaison est coupée une fois le temps indiqué écoulé.
MailServerAddress	STRING[254]	Nom de domaine complet (Fully Qualified Domain Name : FQDN) du serveur de messagerie. L'accès au serveur de messagerie se fait par le nom de domaine complet. Exemple : "www.mymailserver.com."
UserName	STRING[254]	Identifiant pour le serveur de messagerie
PassWord	STRING[254]	Mot de passe pour le serveur de messagerie
From	Struct	Adresse de l'expéditeur du courriel, définie au moyen des deux paramètres suivants de type STRING. Exemple : "myname@mymailserver.com"
LocalPartPlu-sAtSign	STRING[64]	Partie locale de l'adresse de l'expéditeur, signe @ compris. Exemple : "myname@"
FullQualified-DomainName	STRING[254]	Nom de domaine complet (FQDN : Fully Qualified Domain Name) du serveur de messagerie. Exemple : "mymail-server.com"

Paramètres TO_S et CC

Les paramètres TO_S et CC sont des chaînes, avec le contenu suivant par exemple :

- <wenna@mydomain.com>, <ruby@mydomain.com>
- <admin@mydomain.com>, <judy@mydomain.com>

Tenez compte des règles suivantes lorsque vous entrez ces paramètres :

- Il faut insérer un espace et le signe "<" avant chaque adresse.
- Il faut terminer chaque adresse par le signe ">".
- Il faut insérer une virgule entre les adresses dans TO et CC.

Pour des raisons liées à l'exécution et à l'espace mémoire, l'instruction "TMAIL_C" ne contrôle pas la syntaxe des paramètres TO_S et CC.

Paramètres DONE, BUSY et ERROR

Les paramètres de sortie DONE, BUSY et ERROR ne sont affichés que pendant un cycle si l'état du paramètre de sortie BUSY passe de 1 à 0.

Le tableau suivant montre les relations entre DONE, BUSY et ERROR. A l'aide de ce tableau, vous pouvez déterminer l'état en cours de l'instruction TMAIL_C et le moment où l'envoi du courriel est achevé.

DONE	BUSY	ERROR	Description
0	1	0	La tâche est en cours de traitement.
1	0	0	Tâche achevée avec succès.
0	0	1	La tâche s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS (Page 921).
0	0	0	Aucune (nouvelle) tâche n'a été affectée à l'instruction TMAIL_C.

Paramètre STATUS

Le tableau suivant présente les valeurs que TMAIL_C renvoie dans le paramètre STATUS :

Valeur en retour STATUS* (W#16#...)	Explication	Remarques
0000	Le traitement de TMAIL_C s'est achevé sans erreur.	L'achèvement sans erreur de TMAIL_C ne signifie pas obligatoirement que le courriel envoyé arrivera à destination. La saisie incorrecte de l'adresse d'un destinataire ne génère pas d'erreur d'état pour l'instruction TMAIL_C. Dans ce cas, il n'est pas garanti que le courriel parviendra aux autres destinataires même s'ils ont été entrés correctement.
7001	TMAIL_C est actif (BUSY = 1).	Premier appel : la tâche est déclenchée.
7002	TMAIL_C est actif (BUSY = 1).	Appel intermédiaire : tâche déjà active.
8xxx	Le traitement de TMAIL_C s'est achevé avec un code d'erreur des instructions de communication appelées en interne.	Pour plus d'informations, reportez-vous aux descriptions du paramètre STATUS pour les instructions de communication TCON, TDISCON, TSEND et TRCV (Page 885).
8010	Erreur lors de l'établissement de la liaison	Vous trouverez plus d'informations sur l'évaluation dans le paramètre SFB_STATUS du bloc de données d'instance. Le code d'erreur contenu dans le paramètre SFB_STATUS est expliqué dans la description du paramètre STATUS pour l'instruction TCON (Page 885).
8011	Erreur lors de l'envoi des données.	Vous trouverez plus d'informations sur l'évaluation dans le paramètre SFB_STATUS du bloc de données d'instance. Le code d'erreur contenu dans le paramètre SFB_STATUS est expliqué dans la description du paramètre STATUS pour l'instruction TSEND (Page 885).
8012	Erreur lors de la réception des données.	Vous trouverez plus d'informations sur l'évaluation dans le paramètre SFB_STATUS du bloc de données d'instance. Le code d'erreur contenu dans le paramètre SFB_STATUS est expliqué dans la description du paramètre STATUS pour l'instruction TRCV (Page 885).
8013	Erreur lors de l'établissement de la liaison	Vous trouverez plus d'informations sur l'évaluation dans le paramètre SFB_STATUS du bloc de données d'instance. Le code d'erreur contenu dans le paramètre SFB_STATUS est expliqué dans la description du paramètre STATUS pour les instructions TCON (Page 885) et TDISCON (Page 885).

Valeur en retour STATUS* (W#16#...)	Explication	Remarques
8014	L'établissement de la liaison n'est pas possible.	Vous avez peut-être entré une adresse IP de serveur de messagerie incorrecte (MailServerAddress (Page 918)) ou indiqué un intervalle de temps trop court (WatchDogTime (Page 918)) pour l'établissement de la liaison. Il est également possible que la CPU ne soit pas connectée au réseau ou que la configuration de la CPU soit incorrecte.
8015	Type de données incorrect pour MAIL_ADDR_PARAM	Les seuls types de données valides sont les types de données système (structures) Tmail_v4 et TMail_FQDN.
8016	Type de données incorrect pour le paramètre ATTACHMENT	Les seuls types de données valides sont ArrayOfByte, ArrayOfWord et ArrayOfDWord.
8017	Longueur de données incorrecte pour le paramètre ATTACHMENT	La longueur des données doit être <= 65534 octets.
82xx, 84xx, ou 85xx	Le message d'erreur provient du serveur de messagerie et correspond, à l'exception du "8", au code d'erreur du protocole SMTP. Les lignes suivantes présentent plusieurs codes d'erreur qui peuvent se produire.	Vous trouverez des informations plus détaillées sur le code d'erreur SMTP et sur d'autres codes d'erreur du protocole SMTP sur Internet ou dans la documentation du serveur de messagerie relative aux erreurs. Vous pouvez également voir le message d'erreur le plus récent du serveur de messagerie dans le paramètre BUFFER1 du DB d'instance. Vous trouverez les dernières données envoyées par l'instruction TMAIL_C sous DATEN dans le DB d'instance.
8450	Action non exécutée : la boîte aux lettres électronique n'est pas disponible/accessible	Réessayez ultérieurement.
8451	Action annulée : erreur de traitement local	Réessayez ultérieurement.
8500	Erreur de syntaxe : erreur non reconnue. Il peut également s'agir d'une chaîne de commande trop longue. Ce message peut également apparaître lorsque le serveur de messagerie ne prend pas en charge la procédure d'authentification LOGIN.	Vérifiez les paramètres de TMAIL_C. Essayez d'envoyer un courriel sans authentification. Pour ce faire, remplacez le contenu du paramètre UserName par une chaîne vide. Si aucun nom d'utilisateur n'est indiqué, la procédure d'authentification LOGIN n'est pas utilisée.
8501	Erreur de syntaxe : entrée incorrecte dans un paramètre	Cause possible : adresse incorrecte dans le paramètre TO_S ou CC (voir aussi : paramètres TO_S et CC (Page 920)).
8502	Commande inconnue ou non implémentée.	Vérifiez vos entrées, en particulier le paramètre FROM. Il est peut-être incomplet ou vous avez omis le "@" ou le "." (voir aussi : paramètres TO_S et CC (Page 920)).
8535	Authentification SMTP incomplète	Vous avez peut-être entré un nom d'utilisateur ou un mot de passe incorrect.

Valeur en retour STATUS* (W#16#...)	Explication	Remarques
8550	Impossible d'accéder au serveur de messagerie. Vous n'avez pas de droits d'accès.	Vous avez peut-être entré un nom d'utilisateur ou un mot de passe incorrect ou votre serveur de messagerie n'accepte peut-être pas votre procédure d'ouverture de session. Une autre explication pourrait être une erreur dans le nom de domaine après le "@" dans le paramètre TO_S ou CC (voir aussi : paramètres TO_S et CC (Page 920)).
8552	Action annulée : la taille de mémoire affectée a été dépassée.	Réessayez ultérieurement.
8554	La transmission a échoué.	Réessayez ultérieurement.
* Vous pouvez afficher les codes d'erreur sous forme de valeurs entières ou hexadécimales dans l'éditeur de programmes.		

11.2.8.14 UDP

UDP est un protocole standard décrit par RFC 768 : User Datagram Protocol (protocole de datagramme utilisateur). UDP fournit un mécanisme permettant à une application d'envoyer un datagramme à une autre application. Toutefois, la distribution des données n'est pas garantie. Ce protocole présente les caractéristiques suivantes :

- Protocole de communication rapide
- Convenant à des quantités faibles à moyennes de données (jusqu'à 1 472 octets)
- UDP est un protocole de commande de transport plus simple que TCP avec une couche fine présentant des trafics de service faibles.
- Utilisable très souplement avec de nombreux systèmes tiers
- Apte au routage
- Utilisation de numéros de port pour diriger les datagrammes
- Il n'y a pas d'accusé de réception des messages. C'est à l'application qu'il incombe de gérer la reprise sur erreur et la sécurité.
- Un effort de programmation est nécessaire pour la gestion des données en raison de l'interface de programmation EMISSION / RECEPTION.

UDP prend en charge la diffusion générale. Pour utiliser la diffusion générale, vous devez configurer la partie "adresse IP" du paramètre ADDR. Ainsi, une CPU ayant l'adresse IP 192.168.2.10 et le masque de sous-réseau 255.255.255.0 utilisera l'adresse de diffusion générale 192.168.2.255.

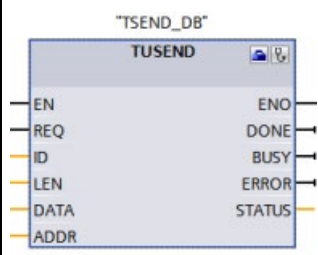

11.2.8.15 TUSEND et TURCV

Les instructions suivantes gèrent le processus de communication UDP :

- TCON établit la communication entre le PC client et serveur (CPU).
- TUSEND et TURCV envoient et reçoivent les données.
- TDISCON coupe la communication entre le client et le serveur.

Reportez-vous à TCON, TDISCON, TSEND et TRCV (Page 885) dans la rubrique "TCP et ISO sur TCP" pour plus d'informations sur les instructions de communication TCON et TDISCON.

Tableau 11- 39 Instructions TUSEND et TURCV

CONT/LOG	SCL	Description
	<pre>"TUSEND_DB" (req:=_bool_in_, ID:=_word_in_, len:=_udint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, data:=_variant_inout_);</pre>	<p>L'instruction TUSEND envoie des données via UDP au partenaire éloigné indiqué par le paramètre ADDR.</p> <p>Pour démarrer la tâche d'envoi des données, appelez l'instruction TUSEND avec REQ = 1.</p>
	<pre>"TURCV_DB" (en_r:=_bool_in_, ID:=_word_in_, len:=_udint_in_, ndr=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, rcvd_len=>_udint_out_, data:=_variant_inout_);</pre>	<p>L'instruction TURCV reçoit des données via UDP. Le paramètre ADDR indique l'adresse de l'émetteur. Lorsque l'exécution de TURCV s'est achevée avec succès, le paramètre ADDR contient l'adresse du partenaire éloigné (l'émetteur).</p> <p>TURCV ne prend pas en charge le mode ad hoc.</p> <p>Pour démarrer la tâche de réception des données, appelez l'instruction TURCV avec EN_R = 1.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

TCON, TDISCON, TUSEND et TURCV opèrent de manière asynchrone, ce qui signifie que le traitement de la tâche s'étend sur plusieurs exécutions de l'instruction.

Tableau 11- 40 Types de données pour les paramètres de TUSEND et TURCV

Paramètre et type		Type de données	Description
REQ (TUSEND)	IN	Bool	Lance la tâche d'émission en présence d'un front montant. Les données sont prises dans la zone indiquée par DATA et LEN.
EN_R (TURCV)	IN	Bool	<ul style="list-style-type: none"> 0 : La CPU ne peut pas recevoir. 1 : Valide la réception par la CPU. L'instruction TURCV est prête à recevoir et la tâche de réception est traitée.
ID	IN	Word	Référence à la liaison associée entre le programme utilisateur et la couche de communication du système d'exploitation. ID doit être identique au paramètre ID associé dans la description de la liaison locale. Plage de valeurs : W#16#0001 à W#16#0FFF.
LEN	IN	UDInt	Nombre d'octets à envoyer (TUSEND) ou à recevoir (TURCV) : <ul style="list-style-type: none"> Default = 0. La paramètre DATA détermine la longueur des données à envoyer ou à recevoir. Sinon, plage de valeurs : 1 à 1472
DONE (TUSEND)	IN	Bool	Paramètre d'état DONE (TUSEND) : <ul style="list-style-type: none"> 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. 1 : Tâche achevée sans erreur
NDR (TURCV)	OUT	Bool	Paramètre d'état NDR (TURCV) : <ul style="list-style-type: none"> 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. 1 : La tâche a été achevée avec succès.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 1 : La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche. 0 : La tâche est achevée.
ERROR	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> 0 : Pas d'erreur 1 : Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)
RCVD_LEN	OUT	UDInt	Nombre d'octets reçus (TURCV)

Paramètre et type		Type de données	Description
DATA	IN_OUT	Variante	Adresse de la zone d'émission (TUSEND) ou de la zone de réception (TURCV) : <ul style="list-style-type: none"> • la mémoire image des entrées • la mémoire image des sorties • un memento • un bloc de données
ADDR	IN_OUT	Variante	Pointeur désignant l'adresse du récepteur (pour TUSEND) ou de l'émetteur (pour TURCV) (par exemple, P#DB100.DBX0.0 byte 8). Le pointeur peut désigner n'importe quelle zone de mémoire. Une structure de 8 octets du type suivant est requise : <ul style="list-style-type: none"> • Les 4 premiers octets contiennent l'adresse IP éloignée. • Les 2 octets suivants indiquent le numéro du port partenaire. • Les 2 derniers octets sont réservés.

L'état de la tâche est indiqué dans les paramètres de sortie BUSY et STATUS. STATUS correspond au paramètre de sortie RET_VAL des instructions à fonctionnement asynchrone.

Le tableau suivant montre les relations entre BUSY, DONE (TUSEND), NDR (TURCV) et ERROR. A l'aide de ce tableau, vous pouvez déterminer l'état en cours de l'instruction (TUSEND ou TURCV) ou le moment où l'opération d'émission / réception est achevée.

Tableau 11- 41 Etat des paramètres BUSY, DONE (TUSEND) / NDR (TURCV) et ERROR

BUSY	DONE / NDR	ERROR	Description
VRAI	Sans objet	Sans objet	La tâche est en cours de traitement.
FAUX	VRAI	FAUX	La tâche a été achevée avec succès.
FAUX	FAUX	VRAI	La tâche s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS.
FAUX	FAUX	FAUX	Aucune (nouvelle) tâche n'a été affectée à l'instruction.

¹ En raison du traitement asynchrone des instructions : Pour TUSEND, vous devez faire en sorte que les données dans la zone d'émission restent cohérentes jusqu'à ce que le paramètre DONE ou le paramètre ERROR prenne la valeur VRAI. Pour TURCV, les données dans la zone de réception ne sont cohérentes que lorsque le paramètre NDR prend la valeur VRAI.

Tableau 11- 42 Codes d'erreur pour les paramètres ERROR et STATUS de TUSEND et TURCV

ERROR	STATUS	Description
0	0000	<ul style="list-style-type: none"> Tâche d'émission achevée sans erreur (TUSEND). De nouvelles données ont été acceptées : La longueur en cours des données reçues est indiquée dans RCVD_LEN (TURCV).
0	7000	<ul style="list-style-type: none"> Pas de traitement de tâche actif (TUSEND) Bloc pas prêt à recevoir (TURCV)
0	7001	<ul style="list-style-type: none"> Démarrage du traitement de la tâche, données en cours d'envoi (TUSEND) : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement. Bloc prêt à recevoir ; une tâche de réception a été activée (TURCV).
0	7002	<ul style="list-style-type: none"> Exécution intermédiaire de l'instruction (REQ sans objet), tâche en cours de traitement (TUSEND) : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement. Exécution intermédiaire de l'instruction, tâche en cours de traitement : l'instruction TURCV écrit les données dans la zone de réception pendant ce traitement. C'est pourquoi une erreur pourrait être à l'origine de données incohérentes dans la zone de réception.
1	8085	Le paramètre LEN est supérieur à la plus grande valeur autorisée ou a la valeur 0 (TUSEND) ou vous avez modifié la valeur du paramètre LEN ou DATA depuis la première exécution de l'instruction (TURCV).
1	8086	Le paramètre ID ne se situe pas dans la plage d'adresses permise.
1	8088	<ul style="list-style-type: none"> Le paramètre LEN est plus grand que la zone d'émission (TUSEND) ou la zone de réception (TURCV) indiquée dans DATA. La zone de réception est trop petite (TURCV).
1	8089	Le paramètre ADDR ne pointe pas sur un bloc de données.
1	80A1	<p>Erreur de communication :</p> <ul style="list-style-type: none"> La liaison indiquée entre le programme utilisateur et la couche de communication du système d'exploitation n'a pas encore été établie. La liaison indiquée entre le programme utilisateur et la couche de communication du système d'exploitation est en train d'être coupée. Une tâche d'émission (TUSEND) ou de réception (TURCV) via cette liaison n'est pas possible. L'interface est en cours de réinitialisation.
1	80A4	L'adresse IP du noeud d'extrémité de la liaison éloignée est invalide ; il est possible qu'elle corresponde à l'adresse IP locale (TUSEND).
1	80B3	<ul style="list-style-type: none"> La variante de protocole paramétrée (paramètre connection_type dans la description de la liaison) n'est pas UDP. Veuillez utiliser l'instruction TSEND ou TRCV. Paramètre ADDR : paramétrage invalide pour le numéro de port (TUSEND)

ERROR	STATUS	Description
1	80C3	<ul style="list-style-type: none">• Un bloc avec cet ID est déjà en cours de traitement dans une classe de priorité différente.• Manque interne de ressources
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none">• La liaison entre le programme utilisateur et la couche de communication du système d'exploitation ne peut pas être établie en ce moment (TUSEND).• L'interface est en train de recevoir de nouveaux paramètres (TUSEND).• La liaison est en cours de réinitialisation (TURCV).

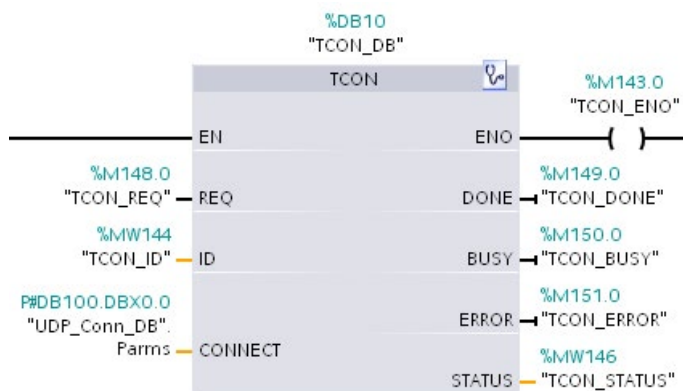
Protocoles de communication Ethernet

Chaque CPU comprend un port PROFINET intégré qui prend en charge la communication PROFINET standard. Les instructions TUSEND et TURCV prennent en charge le protocole Ethernet UDP.

Reportez-vous à "Configuration du routage local/partenaire" (Page 833) dans le chapitre "Configuration des appareils" pour plus d'informations.

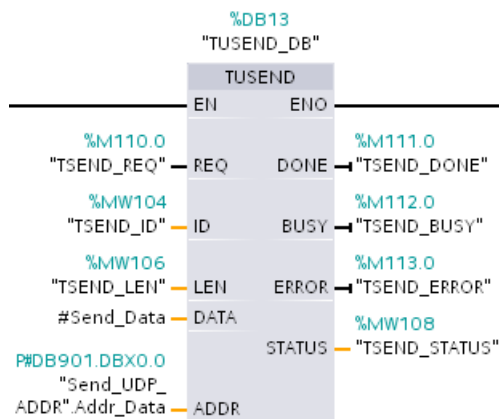
Fonctionnement

Les deux partenaires sont passifs dans la communication UDP. Des valeurs initiales de paramètre typiques pour le type de données "TCON_Param" sont présentées dans les figures suivantes. Les numéros de port (LOCAL_TSAP_ID) sont écrits en format à deux octets. Tous les ports sont autorisés à l'exception de 161, 34962, 34963 et 34964.



UDP_Conn_DB					
	Nom	Type de données	Décalage	Valeur initiale	Commentaire
1	Static				
2	Params	TCON_Param	0.0		
3	BLOCK_LENGTH	UInt	0.0	64	byte length of SDT
4	ID	CONN_OUC	2.0	1	reference to the connection
5	CONNECTION_TYPE	USInt	4.0	19	17: TCP/IP, 18: ISO on TCP
6	ACTIVE_EST	Bool	5.0	false	active/passive connection establishment
7	LOCAL_DEVICE_ID	USInt	6.0	1	1: local I/E interface
8	LOCAL_TSAP_ID_LEN	USInt	7.0	2	byte length of local TSAP id/port number
9	REM_SUBNET_ID_LEN	USInt	8.0	0	byte length of remote subnet id
10	REM_STADDR_LEN	USInt	9.0	0	byte length of remote IP address
11	REM_TSAP_ID_LEN	USInt	10.0	0	byte length of remote port/TSAP id
12	NEXT_STADDR_LEN	USInt	11.0	0	byte length of next station address
13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0		TSAP id/local port number
14	LOCAL_TSAP_ID[1]	Byte		0#16#07	
15	LOCAL_TSAP_ID[2]	Byte		0#16#00	

L'instruction TUSEND envoie des données par le biais d'UDP au partenaire éloigné indiqué dans le type de données "TADDR_Param". L'instruction TURCV reçoit des données par le biais d'UDP. Lorsque l'exécution de TURCV s'est achevée avec succès, le type de données "TADDR_Param" montre l'adresse du partenaire éloigné (l'émetteur), comme illustré dans les figures ci-après.



Send_UDP_ADDR					
	Nom	Type de données	Décalage	Valeur initiale	Commentaire
1	Static				
2	Addr_Data	TADDR_Param	0.0		
3	REM_IP_ADDR	Array[1..4] of USint	0.0		remote station address
4	REM_IP_ADDR{1}	USint		0	
5	REM_IP_ADDR{2}	USint		0	
6	REM_IP_ADDR{3}	USint		0	
7	REM_IP_ADDR{4}	USint		0	
8	REM_PORT_NR	Ulnr	4.0	0	remote port number
9	RESERVED	Word	6.0	0	unused; has to be 0

11.2.8.16 T_CONFIG

L'instruction T_CONFIG permet de modifier à partir du programme utilisateur l'adresse Ethernet, le nom d'appareil PROFINET ou les adresses IP des serveurs NTP pour la synchronisation de l'heure. Les éléments suivants peuvent être modifiés de manière permanente ou temporaire :

- Adresse IP
- Masque de sous-réseau
- Adresse du routeur
- Nom de station
- Adresses IP de quatre serveurs NTP au maximum

Remarque

Le bouton radio "Permettre la modification de l'adresse IP directement sur l'appareil" (Page 939), situé sur la page "Adresse Ethernet" des propriétés de la CPU, permet de modifier l'adresse IP en ligne ou à l'aide de l'instruction "T_CONFIG" après le chargement du programme.

La case à cocher "Permettre la modification du nom d'appareil PROFINET directement sur l'appareil" (Page 941), située sur la page "Adresse Ethernet" des propriétés de la CPU, permet de modifier le nom d'appareil PROFINET en ligne ou à l'aide de l'instruction "T_CONFIG" après le chargement du programme.

La case à cocher "Activer la synchronisation de l'heure via le serveur NTP" (Page 942), située sur la page "Synchronisation de l'heure" des propriétés de la CPU, permet de modifier les adresses IP de quatre serveurs NTP au maximum.

Remarque

Vous ne pouvez pas exécuter plus d'une instruction T_CONFIG à la fois.

Remarque

Les modifications apportées à l'adresse IP ou au nom de station de la CPU peuvent être temporaires ou permanentes. Les modifications apportées aux adresses IP des serveurs NTP sont uniquement temporaires.

- "Modification permanente" signifie que la modification est rémanente, c'est-à-dire conservée en cas de coupure de courant.
 - "Modification temporaire" signifie que la modification est volatile et que la valeur initiale est rétablie en cas de coupure de courant.
-

Tableau 11- 43 Instruction T_CONFIG

CONT/LOG	SCL	Description
	<pre>"T_CONFIG_DB" (Req:=_bool_in_, Interface:=_uint_in_, Conf_Data:=_variant_in_, Done=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, Status=>_dword_out_, Err_Loc=>_dword_out_);</pre>	<p>Utilisez l'instruction T_CONFIG pour modifier les paramètres de configuration IP à partir de votre programme utilisateur.</p> <p>T_CONFIG fonctionne de manière asynchrone. Son exécution s'étend sur plusieurs appels.</p>

Tableau 11- 44 Types de données T_CONFIG pour les paramètres

Paramètre et type	Type de données	Description
REQ Input	Bool	Démarre l'instruction en présence d'un front montant.
INTERFACE Input	HW_Interface	ID de l'interface de réseau
CONF_DATA Input	Variant	Référence à la structure des données de configuration ; CONF_DATA est défini par une structure contenant jusqu'à quatre types de données système (SDT).
DONE Output	Bool	<ul style="list-style-type: none"> 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. 1 : Tâche exécutée sans erreur
BUSY Output	Bool	<ul style="list-style-type: none"> 0 : La tâche est achevée. 1 : La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche.
ERROR Output	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> 0 : Pas d'erreur 1 : Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.
STATUS Output	DWord	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)
ERR_LOC Output	DWord	Emplacement de l'erreur (ID de bloc et emplacement de sous-bloc dans la structure CONF_DATA)

Les informations de configuration IP sont placées dans le bloc de données CONF_DATA conjointement avec un pointeur Variant désignant le paramètre CONF_DATA référencé ci-avant. L'exécution réussie de l'instruction T_CONFIG se termine avec la transmission des données de configuration IP à l'interface de réseau.

Les messages d'état et d'erreur de l'instruction T_CONFIG sont transmis dans les paramètres STATUS et ERR_LOC :

- La cause de l'erreur figure dans le paramètre STATUS.
- L'emplacement de l'erreur survenue figure dans le paramètre ERR_LOC. Les options suivantes sont disponibles :
 - 16#0000_0000 : Pas d'erreur ou erreur à l'appel de l'instruction (par exemple, erreurs lors du paramétrage de l'instruction ou lors de la communication avec l'interface PROFINET)
 - 16#0001_0000 : Données de configuration erronées dans les paramètres du type de données système IF_CONF_HEADER
 - 16#0001_000x : Données de configuration erronées dans les paramètres du type de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP. x correspond à la position du sous-bloc erroné dans la structure T_CONFIG. Si la structure T_CONFIG contient, par exemple, un sous-bloc pour l'adresse IP et un sous-bloc pour le nom de station et que l'erreur se situe dans le sous-bloc du nom de station, ERR_LOC a la valeur 0001_0002.

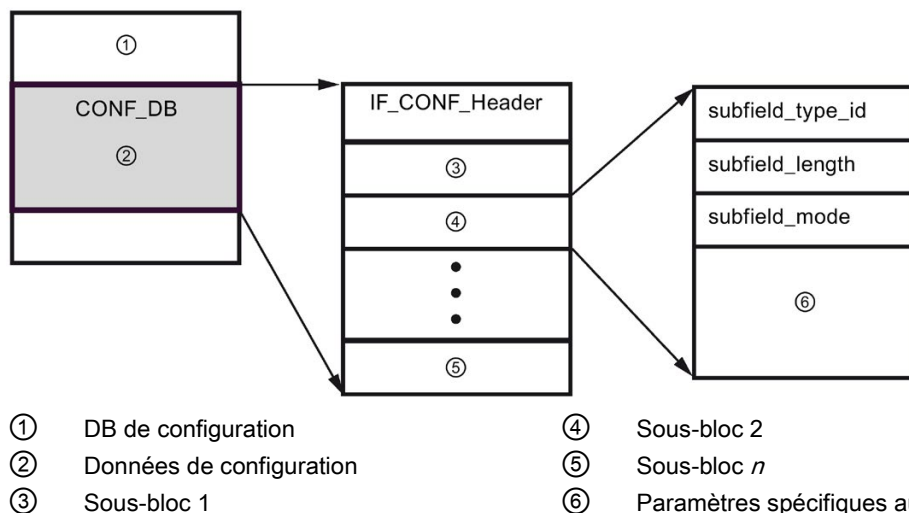
Le tableau suivant indique les valeurs possibles pour les paramètres STATUS et ERR_LOC :

STATUS*	ERR_LOC*	Explication
0000_0000	0000_0000	Traitement de la tâche achevé sans erreur
0070_0000	0000_0000	Pas de traitement de tâche actif
0070_0100	0000_0000	Démarrage du traitement de la tâche
0070_0200	0000_0000	Appel intermédiaire (REQ sans objet).
C08x_yy00	0000_0000	Information d'erreur générale
C080_8000	0000_0000	Erreur à l'appel de l'instruction : L'ID matérielle dans le paramètre Interface est invalide.
C080_8100	0000_0000	Erreur à l'appel de l'instruction : L'ID matérielle dans le paramètre Interface ne désigne pas une interface PROFINET.
C080_8700	0000_0000	Erreur à l'appel de l'instruction : Longueur incorrecte du bloc de données dans le paramètre CONF_DATA
C080_8800	0001_0000	Erreur dans le type de données système IF_CONF_HEADER : Le paramètre FieldType a une valeur invalide. Utilisez la valeur "0" pour FieldType.
C080_8900	0001_0000	Erreur dans le type de données système IF_CONF_HEADER : Le paramètre FieldId a une valeur invalide ou est utilisé plusieurs fois. Utilisez la valeur "0" pour FieldId.
C080_8A00	0001_0000	Erreur dans le type de données système IF_CONF_HEADER : Nombre incorrect dans le paramètre SubfieldCount. Entrez le nombre correct de types de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP utilisés.
C080_8B00	0001_000x	Erreur dans le type de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP : Le paramètre Id a une valeur invalide. Utilisez "30" pour IF_CONF_V4, "40" pour IF_CONF_NOS et "17" pour IF_CONF_NTP.
C080_8C00	0001_000x	Erreur dans le type de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP : Type de données système utilisé incorrect, ordre erroné ou utilisation multiple d'un type de données système

STATUS*	ERR_LOC*	Explication
C080_8D00	0001_000x	Erreur dans le type de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP : Le paramètre Length a une valeur incorrecte ou invalide.
C080_8E00	0001_000x	Erreur dans le type de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP : Le paramètre Mode a une valeur incorrecte ou invalide. <ul style="list-style-type: none"> • Seules les valeurs "1" (permanent) ou "2" (temporaire) sont autorisées avec IF_CONF_V4 et IF_CONF_NOS . • Seule la valeur "2" (temporaire) est autorisée avec IF_CONF_NTP.
C080_9000	0001_000x	Erreur dans le type de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP : Impossible d'appliquer les données de configuration. Cause possible : <ul style="list-style-type: none"> • Pour IF_CONF_V4 : L'option "Régler l'adresse IP sur l'appareil" n'était pas sélectionnée dans la configuration matérielle. • Pour IF_CONF_NOS : L'option "Régler le nom d'appareil PROFINET sur l'appareil" n'était pas sélectionnée dans la configuration matérielle. • Pour IF_CONF_NTP : L'option "Activer la synchronisation de l'heure via le serveur NTP" n'était pas sélectionnée dans la configuration matérielle et aucune adresse IP n'était définie pour les serveurs NTP.
C080_9400	0001_000x	Erreur dans le type de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP : Une valeur de paramètre n'est pas définie ou est invalide.
C080_9500	0001_000x	Erreur dans le type de données système IF_CONF_V4, IF_CONF_NOS ou IF_CONF_NTP : Les valeurs de deux paramètres sont incompatibles.
C080_C200	0000_0000	Erreur à l'appel de l'instruction : Impossible de transférer les données de configuration. Cause possible : l'interface PROFINET n'est pas accessible.
C080_C300	0000_0000	Erreur à l'appel de l'instruction : Ressources insuffisantes (par exemple, plusieurs appels de "T_CONFIG" avec différents paramètres)
C080_C400	0000_0000	Erreur à l'appel de l'instruction : Erreur de communication temporaire. Indication temporelle pour le passage à l'heure d'été.
C080_D200	0000_0000	Erreur à l'appel de l'instruction : Appel impossible. L'instruction n'est pas prise en charge par l'interface PROFINET sélectionnée.

Bloc de données CONF_DATA

Le schéma suivant montre comment les données de configuration à transférer sont stockées dans le DB de configuration.



Les données de configuration de CONF_DB comprennent un bloc contenant un en-tête (IF_CONF_Header) et plusieurs sous-blocs. IF_CONF_Header fournit les éléments suivants :

- field_type_id (type de données UInt) : Zéro
- field_id (type de données UInt) : Zéro
- subfield_cnt (type de données UInt) : Nombre de sous-blocs

Chaque sous-bloc est constitué à son tour d'un en-tête (subfield_type_id, subfield_length, subfield_mode) et des paramètres spécifiques du sous-bloc. Chaque sous-bloc doit comporter un nombre pair d'octets. "subfield_mode" peut avoir la valeur 1 ou 2. Voir les tableaux ci-après.

Remarque

Un seul bloc (IF_CONF_Header) est autorisé actuellement. Ses paramètres field_type_id et field_id doivent avoir la valeur zéro. D'autres blocs ayant des valeurs différentes pour field_type_id et field_id sont envisageables dans le futur.

Tableau 11- 45 Sous-blocs pris en charge

subfield_type_id	Type de données	Explication
30	IF_CONF_V4	Paramètres IP : adresse IP, masque de sous-réseau, adresse du routeur
40	IF_CONF_NOS	Nom de station PROFINET IO (Name of station)
17	IF_CONF_NTP	Protocole NTP (Network Time Protocol)

Tableau 11- 46 Eléments du type de données IF_CONF_V4

Nom	Type de données	Valeur initiale	Description
Id	UInt	30	subfield_type_id
Length	UInt	18	subfield_length
Mode	UInt	0	subfield_mode (1 = permanent ou 2 = temporaire)
InterfaceAddress	IP_V4	-	Adresse d'interface
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte	0	Adresse IP, octet de poids fort : 200
ADDR[2]	Byte	0	Adresse IP, octet de poids fort : 12
ADDR[3]	Byte	0	Adresse IP, octet de poids faible : 1
ADDR[4]	Byte	0	Adresse IP, octet de poids faible : 144
SubnetMask	IP_V4	-	Masque de sous-réseau
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte	0	Masque de sous-réseau, octet de poids fort : 255
ADDR[2]	Byte	0	Masque de sous-réseau, octet de poids fort : 255
ADDR[3]	Byte	0	Masque de sous-réseau, octet de poids faible : 255
ADDR[4]	Byte	0	Masque de sous-réseau, octet de poids faible : 0
DefaultRouter	IP_V4	-	Routeur par défaut
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte	0	Routeur, octet de poids fort : 200
ADDR[2]	Byte	0	Routeur, octet de poids fort : 12
ADDR[3]	Byte	0	Routeur, octet de poids faible : 1
ADDR[4]	Byte	0	Routeur, octet de poids faible : 1

Tableau 11- 47 Eléments du type de données IF_CONF_NOS

Nom	Type de données	Valeur initiale	Description
Id	UInt	40	subfield_type_id
Length	UInt	246	subfield_length
Mode	UInt	0	subfield_mode (1 = permanent ou 2 = temporaire)
NOS (Name of station)	Array[1..240] of Byte	0	Nom de station : Vous devez occuper le tableau ARRAY à partir du premier octet. Si le tableau ARRAY est plus long que le nom de station à affecter, vous devez entrer un octet nul après le nom de station réel (conformément à CEI 61158-6-10). Sinon, l'élément NOS est rejeté et l'instruction T_CONFIG (Page 931) inscrit le code d'erreur DW#16#C0809400 dans STATUS. Si vous écrivez zéro dans le premier octet, le nom de la station est effacé.

Les noms de station sont soumis aux limitations suivantes :

- Un nom d'élément à l'intérieur du nom de station, c'est-à-dire une chaîne de caractères entre deux points, ne doit pas dépasser 63 caractères.
- Les caractères spéciaux, tels que tréma, crochet, trait de soulignement, barre oblique, espace, etc., sont interdits. Le seul caractère spécial permis est le trait d'union.
- Le nom de station ne doit pas commencer ni se terminer par un trait d'union (-).
- Le nom de station ne doit pas commencer par un chiffre.
- La forme n.n.n.n (n = 0, ... 999) du nom de station n'est pas permise.
- Le nom de station ne doit pas commencer par la chaîne "port-xyz" ou "port-xyz-abcde" (a, b, c, d, e, x, y, z = 0, ... 9).

Remarque

Vous pouvez également créer un tableau (ARRAY) "NOS" qui est plus court que 240 octets, mais il ne doit pas faire moins de 2 octets. Dans ce cas, vous devez ajuster en conséquence la variable "Length" (longueur du sous-bloc).

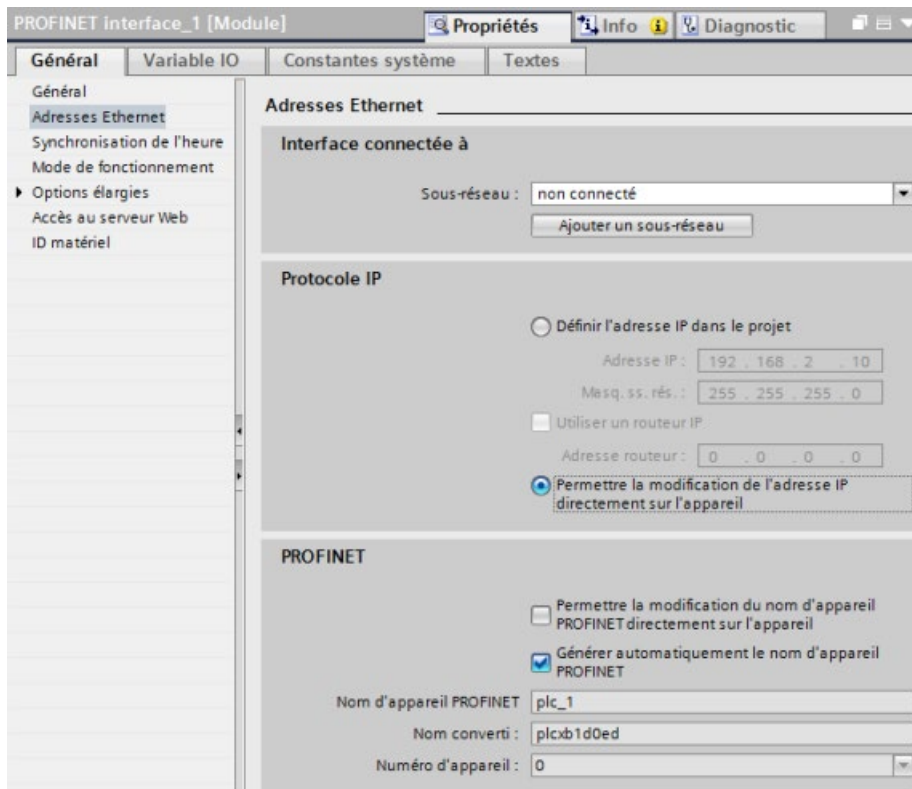
Tableau 11- 48 Éléments du type de données IF_CONF_NTP

Nom		Type de données	Valeur initiale	Description
Id		UInt	17	subfield_type_id
Length		UInt	22	subfield_length
Mode		UInt	0	subfield_mode (2 = temporaire)
NTP_IP		Array[1...4] of IP_V4	-	Adresses IP des serveurs NTP
	NTP_IP[1]	IP_V4		Adresse IP du serveur NTP 1
	ADDR	Array[1...4] of Byte	0	
	ADDR[1]	Byte	0	Octet de poids fort de l'adresse IP
	ADDR[2]	Byte	0	Octet de poids fort de l'adresse IP
	ADDR[3]	Byte	0	Octet de poids faible de l'adresse IP
	ADDR[4]	Byte	0	Octet de poids faible de l'adresse IP
	NTP_IP[2]	IP_V4		Adresse IP du serveur NTP 2
	ADDR	Array[1...4] of Byte	0	
	ADDR[1]	Byte	0	Octet de poids fort de l'adresse IP
	ADDR[2]	Byte	0	Octet de poids fort de l'adresse IP
	ADDR[3]	Byte	0	Octet de poids faible de l'adresse IP
	ADDR[4]	Byte	0	Octet de poids faible de l'adresse IP
	NTP_IP[3]	IP_V4		Adresses IP du serveur NTP 3
	ADDR	Array[1...4] of Byte	0	
	ADDR[1]	Byte	0	Octet de poids fort de l'adresse IP
	ADDR[2]	Byte	0	Octet de poids fort de l'adresse IP
	ADDR[3]	Byte	0	Octet de poids faible de l'adresse IP
	ADDR[4]	Byte	0	Octet de poids faible de l'adresse IP
	NTP_IP[4]	IP_V4		Adresses IP du serveur NTP 4
	ADDR	Array[1...4] of Byte	0	
	ADDR[1]	Byte	0	Octet de poids fort de l'adresse IP
	ADDR[2]	Byte	0	Octet de poids fort de l'adresse IP
	ADDR[3]	Byte	0	Octet de poids faible de l'adresse IP
	ADDR[4]	Byte	0	Octet de poids faible de l'adresse IP

Exemple : Utilisation de l'instruction T_CONFIG pour modifier les paramètres IP

Dans l'exemple suivant, l'adresse IP "InterfaceAddress", le masque de sous-réseau "SubnetMask" et le routeur IP "DefaultRouter" sont modifiés dans le sous-bloc "addr". Vous devez sélectionner le bouton radio "Permettre la modification de l'adresse IP directement sur l'appareil" dans la page "Adresse Ethernet" des propriétés de la CPU pour pouvoir modifier les paramètres IP à l'aide de l'instruction "T_CONFIG" après le chargement du programme.

CONF_DATA_1			
	Nom	Type de données	Valeur de départ
1	▼ Static		
2	▼ Conf_data	Struct	
3	▼ header	IF_CONF_Header	
4	Fieldtype	UInt	0
5	Fieldid	UInt	0
6	SubfieldCount	UInt	1
7	▼ addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	▼ InterfaceAddress	IP_V4	
12	▼ ADDR	array [1..4] of Byte	
13	ADDR[1]	Byte	192
14	ADDR[2]	Byte	168
15	ADDR[3]	Byte	2
16	ADDR[4]	Byte	30
17	▼ SubnetMask	IP_V4	
18	▼ ADDR	array [1..4] of Byte	
19	ADDR[1]	Byte	255
20	ADDR[2]	Byte	255
21	ADDR[3]	Byte	255
22	ADDR[4]	Byte	0
23	▼ DefaultRouter	IP_V4	
24	▼ ADDR	array [1..4] of Byte	
25	ADDR[1]	Byte	192
26	ADDR[2]	Byte	168
27	ADDR[3]	Byte	2
28	ADDR[4]	Byte	1



Exemple : Utilisation de l'instruction T_CONFIG pour modifier les paramètres IP et les noms des périphériques PROFINET IO

Dans l'exemple suivant, les sous-blocs "addr" et "nos" (Name of station) sont tous deux modifiés. Vous devez sélectionner la case à cocher "Permettre la modification du nom d'appareil PROFINET directement sur l'appareil" dans la page "Adresse Ethernet" des propriétés de la CPU pour pouvoir modifier le nom d'appareil PROFINET à l'aide de l'instruction "T_CONFIG" après le chargement du programme.

CONF_DATA_2			
	Nom	Type de données	Valeur de départ
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	Uint	0
5	FieldId	Uint	0
6	SubfieldCount	Uint	2
7	addr	IF_CONF_v4	
8	Id	Uint	30
9	Length	Uint	18
10	Mode	Uint	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	SubnetMask	IP_V4	
14	ADDR	array [1..4] of Byte	
15	DefaultRouter	IP_V4	
16	ADDR	array [1..4] of Byte	
17	nos	IF_CONF_IOS	
18	Id	Uint	40
19	Length	Uint	246
20	Mode	Uint	1
21	IOS	array [1..240] of Byte	

The screenshot shows the 'PROFINET interface_1 [Module]' configuration window. The 'Adresses Ethernet' section is expanded, showing the following settings:

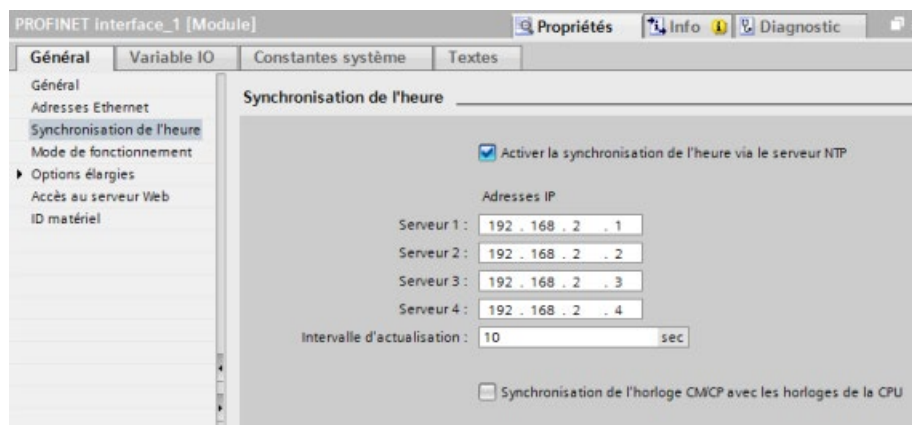
- Interface connectée à:** Sous-réseau : non connecté (with an 'Ajouter un sous-réseau' button).
- Protocole IP:**
 - Définir l'adresse IP dans le projet (Adresse IP: 192.168.2.10, Masq. ss. rés.: 255.255.255.0)
 - Utiliser un routeur IP (Adresse routeur: 0.0.0.0)
 - Permettre la modification de l'adresse IP directement sur l'appareil
- PROFINET:**
 - Permettre la modification du nom d'appareil PROFINET directement sur l'appareil
 - Générer automatiquement le nom d'appareil PROFINET
 - Nom d'appareil PROFINET: plc_1
 - Nom converti: plcb1d0ed
 - Numéro d'appareil: 0

Exemple : Utilisation de l'instruction T_CONFIG pour modifier les adresses IP des serveurs NTP

Dans l'exemple suivant, l'instruction T_CONFIG modifie les adresses IP de quatre serveurs NTP (Network Time Protocol) au maximum dans le sous-bloc "ntp".

Dans les propriétés de la CPU, "Interface PROFINET [X1]", page "Synchronisation de l'heure", vous configurez la synchronisation NTP en cochant la case "Activer la synchronisation de l'heure via le serveur NTP" comme illustré dans la figure ci-après. Vous pouvez ensuite modifier les adresses IP des serveurs NTP à l'aide de l'instruction "T_CONFIG" après le chargement du programme.

CONF_DATA_3			
	Nom	Type de données	Valeur de départ
1	Static		
2	Conf_Data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	1
7	ntp	IF_CONF_NTP	
8	Id	UInt	17
9	Length	UInt	22
10	Mode	UInt	2
11	NTP_IP	Array[1..4] of IP_V4	
12	NTP_IP[1]	IP_V4	
13	ADDR	Array[1..4] of Byte	
14	ADDR[1]	Byte	192
15	ADDR[2]	Byte	168
16	ADDR[3]	Byte	2
17	ADDR[4]	Byte	5
18	NTP_IP[2]	IP_V4	
19	ADDR	Array[1..4] of Byte	
20	ADDR[1]	Byte	192
21	ADDR[2]	Byte	168
22	ADDR[3]	Byte	2
23	ADDR[4]	Byte	6
24	NTP_IP[3]	IP_V4	
25	ADDR	Array[1..4] of Byte	
26	ADDR[1]	Byte	192
27	ADDR[2]	Byte	168
28	ADDR[3]	Byte	2
29	ADDR[4]	Byte	7
30	NTP_IP[4]	IP_V4	
31	ADDR	Array[1..4] of Byte	
32	ADDR[1]	Byte	192
33	ADDR[2]	Byte	168
34	ADDR[3]	Byte	2
35	ADDR[4]	Byte	8



11.2.8.17 Paramètres communs des instructions

Paramètre d'entrée REQ

De nombreuses instructions de communication ouverte (Open User Communication) utilisent une entrée REQ pour déclencher l'opération en présence d'une transition du niveau bas au niveau haut. L'entrée REQ doit être au niveau haut (VRAI) pour une exécution de l'instruction, mais elle peut rester à VRAI aussi longtemps que désiré. L'instruction ne déclenche pas de nouvelle opération tant qu'elle n'a pas été exécutée avec l'entrée REQ à FAUX de manière à ce qu'elle réinitialise l'historique de l'entrée REQ. Cela est nécessaire pour que l'instruction puisse à nouveau détecter une transition du niveau bas au niveau haut afin de déclencher l'opération suivante.

Lorsque vous insérez l'une de ces instructions dans votre programme, STEP 7 vous demande d'identifier le DB d'instance. Utilisez un DB unique pour chaque appel d'instruction. Cela garantit que chaque instruction gère correctement les entrées telles que REQ.

Paramètre entrée ID

Ce paramètre fait référence à "ID local (hexa)" dans la "Vue du réseau" de "Appareils & Réseaux" dans STEP 7 ; il s'agit de l'ID du réseau que vous voulez utiliser pour ce bloc de communication. L'ID doit être identique au paramètre ID associé dans la description de la liaison locale.

Paramètres de sortie DONE, NDR, ERROR et STATUS

Ces instructions fournissent des sorties décrivant l'état d'achèvement.

Tableau 11- 49 Paramètres de sortie des instructions pour la communication ouverte

Paramètre	Type de données	Valeur par défaut	Description
DONE	Bool	FAUX	Est mis à VRAI pour un cycle pour indiquer que la dernière demande s'est achevée sans erreur. FAUX sinon.
NDR	Bool	FAUX	Est mis à VRAI pour un cycle pour indiquer que la dernière action demandée s'est achevée sans erreur et que de nouvelles données ont été reçues. FAUX sinon.
BUSY	Bool	FAUX	Est mis à VRAI pour indiquer que : <ul style="list-style-type: none"> • La tâche n'est pas encore achevée. • Il n'est pas possible de déclencher une nouvelle tâche. Est mis à FAUX quant la tâche est achevée.
ERROR	Bool	FAUX	Est mis à VRAI pour un cycle pour indiquer que la dernière demande s'est achevée avec des erreurs, le code d'erreur correspondant étant contenu dans STATUS. FAUX sinon.
STATUS	Word	0	Etat du résultat : <ul style="list-style-type: none"> • Si le bit DONE ou NDR est à 1, STATUS est mis à 0 ou reçoit un code informatif. • Si le bit ERROR est à 1, STATUS aura la valeur du code d'erreur. • Si aucun des bits ci-dessus n'est à 1, l'instruction renvoie des résultats d'état qui décrivent l'état en cours de la fonction. STATUS conserve sa valeur pendant la durée de l'exécution de la fonction.

Remarque

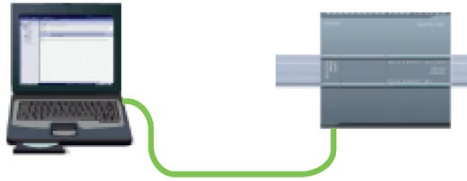
Notez que DONE, NDR et ERROR sont à 1 pour un cycle uniquement.

TSAP et numéros de port limités pour la communication TCP et ISO passive

Si vous vous servez de l'instruction TCON pour configurer et établir la liaison de communication passive, n'utilisez pas les adresses de port suivantes qui sont réservées :

- TSAP ISO (passif) :
 - 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
 - 10.00, 10.01, 11.00, 11.01, ... BF.00, BF.01
- Port TCP (passif) : 5001, 102, 123, 20, 21, 25, 34962, 34963, 34964, 80
- Port UDP (passif) : 161, 34962, 34963, 34964

11.2.9 Communication avec une console de programmation



Une CPU peut communiquer avec une console de programmation STEP 7 dans un réseau.

Tenez compte des points suivants lorsque vous établissez une communication entre une CPU et une console de programmation :

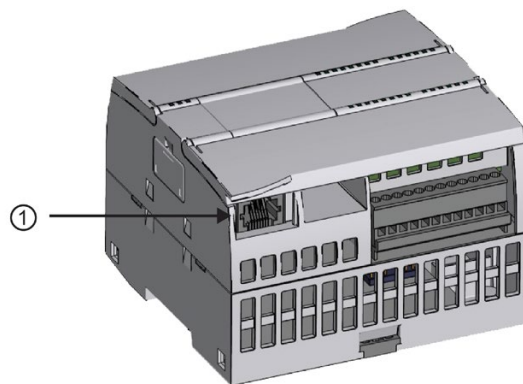
- Configuration/installation : Une configuration matérielle est requise.
- Aucun commutateur Ethernet n'est nécessaire pour la communication un à un ; un commutateur Ethernet est obligatoire pour plus de deux unités dans un réseau.

11.2.9.1 Etablissement de la liaison de communication matérielle

Les interfaces PROFINET établissent les connexions physiques entre une console de programmation et une CPU. Comme la fonction Auto-Cross-Over est intégrée à la CPU, un câble Ethernet standard ou croisé peut être utilisé pour l'interface. Un commutateur Ethernet n'est pas nécessaire pour connecter une console de programmation directement à une CPU.

Procédez comme suit pour créer la liaison matérielle entre une console de programmation et une CPU :

1. Installez la CPU (Page 62).
2. Branchez le câble Ethernet dans le port PROFINET montré ci-dessous.
3. Connectez le câble Ethernet à la console de programmation.



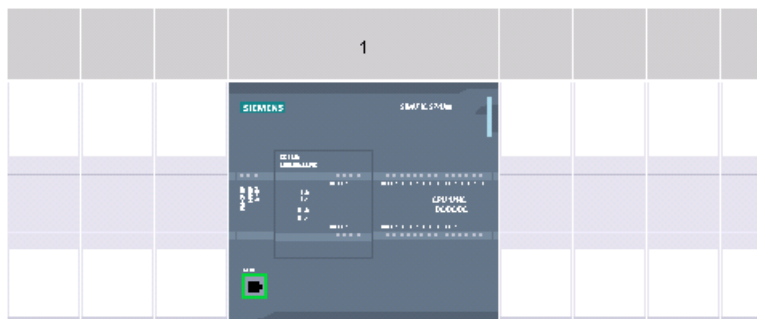
① Port PROFINET

Un serre-câble optionnel est disponible pour renforcer la connexion PROFINET. Pour des informations concernant la commande, voir Pièces détachées et autres matériels (Page 1657).

11.2.9.2 Configuration des appareils

Si vous avez déjà créé un projet avec une CPU, ouvrez votre projet dans STEP 7.

Si ce n'est pas le cas, créez un projet et insérez une CPU (Page 160) dans le châssis. Dans le projet ci-dessous, vous voyez une CPU dans la "Vue des appareils".



11.2.9.3 Affectation d'adresses IP (Internet Protocol)

Affectation des adresses IP

Dans un réseau PROFINET, chaque appareil doit également comporter une adresse IP (protocole Internet). Cette adresse permet à l'appareil de transmettre des données dans un réseau routé plus complexe :

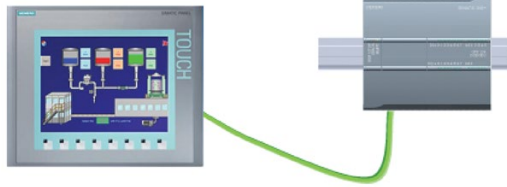
- Si vous avez des consoles de programmation ou d'autres périphériques réseau qui utilisent une carte adaptateur intégrée connectée à votre réseau local d'usine ou une carte adaptateur Ethernet vers USB connectée à un réseau isolé, vous devez leur affecter des adresses IP. Pour plus d'informations, reportez-vous à "Affectation d'adresses IP aux consoles de programmation et périphériques réseau" (Page 836).
- Vous pouvez également affecter une adresse IP à une CPU ou un périphérique réseau en ligne. Cela s'avère particulièrement utile lors d'une configuration d'appareil initiale. Pour plus d'informations, reportez-vous à "Affectation d'une adresse IP à une CPU en ligne" (Page 836)".
- Une fois que vous avez configuré votre CPU ou périphérique réseau dans votre projet, vous pouvez configurer les paramètres pour l'interface PROFINET, dont son adresse IP. Reportez-vous à "Configuration d'une adresse IP pour une CPU dans votre projet" (Page 838) pour plus d'informations.

11.2.9.4 Test de votre réseau PROFINET

Une fois la configuration achevée, vous devez charger le projet dans la CPU. Toutes les adresses IP sont configurées lorsque vous chargez le projet dans la CPU.

La boîte de dialogue "Chargement étendu" de la fonction "Charger dans l'appareil" de la CPU peut montrer tous les dispositifs réseau accessibles en précisant si des adresses IP uniques ont été affectées ou non à tous les appareils. Reportez-vous à "Test du réseau PROFINET" (Page 845) pour plus d'informations.

11.2.10 Communication IHM vers automate



La CPU prend en charge les liaisons de communication PROFINET vers des IHM (Page 33) (interfaces homme-machine). Il faut tenir compte des exigences suivantes lors de la configuration de la communication entre CPU et IHM :

Configuration/installation :

- Le port PROFINET de la CPU doit être configuré pour la connexion avec l'IHM.
- L'IHM doit être installée et configurée.
- Les informations de configuration IHM font partie du projet CPU et peuvent être configurées et chargées de l'intérieur du projet.
- Aucun commutateur Ethernet n'est nécessaire pour la communication un à un ; un commutateur Ethernet est obligatoire pour plus de deux unités dans un réseau.

Remarque

Vous pouvez utiliser le commutateur Ethernet à 4 ports CSM1277 monté sur châssis pour connecter vos CPU et appareils IHM. Le port PROFINET de la CPU ne contient pas de dispositif de commutation Ethernet.

Fonctions prises en charge :

- L'IHM peut lire/écrire des données dans la CPU.
- Des messages peuvent être lancés, sur la base d'informations récupérées dans la CPU.
- Diagnostic système

Tableau 11- 50 Etapes requises pour configurer la communication entre une IHM et une CPU

Etape	Tâche
1	<p>Etablissement de la liaison de communication matérielle</p> <p>Une interface PROFINET établit la connexion physique entre une IHM et une CPU. Comme la fonction Auto-Cross-Over est intégrée à la CPU, vous pouvez utiliser un câble Ethernet standard ou croisé pour l'interface. Un commutateur Ethernet n'est pas nécessaire pour connecter une IHM et une CPU.</p> <p>Reportez-vous à "Communication avec une console de programmation, Etablissement de la liaison de communication matérielle" (Page 945) pour plus d'informations.</p>
2	<p>Configuration des appareils</p> <p>Reportez-vous à "Communication avec une console de programmation, Configuration des appareils" (Page 946) pour plus d'informations.</p>
3	<p>Configuration des liaisons réseau logiques entre une IHM et une CPU</p> <p>Reportez-vous à "Communication IHM vers API, Configuration des liaisons réseau logiques entre deux appareils" (Page 948) pour plus d'informations.</p>
4	<p>Configuration d'une adresse IP dans votre projet</p> <p>Utilisez le même processus de configuration ; vous devez toutefois configurer des adresses IP pour l'IHM et la CPU.</p> <p>Reportez-vous à "Configuration des appareils, Configuration d'une adresse IP pour une CPU dans votre projet" (Page 840) pour plus d'informations.</p>
5	<p>Test du réseau PROFINET</p> <p>Vous devez charger la configuration pour chaque CPU et appareil IHM.</p> <p>Reportez-vous à "Configuration des appareils, Test du réseau PROFINET" (Page 845) pour plus d'informations.</p>

11.2.10.1 Configuration des liaisons réseau logiques entre deux appareils

Une fois que vous avez configuré le châssis avec la CPU, vous être prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Cliquez d'abord sur l'onglet "Liaisons", puis sélectionnez le type de liaison dans la liste déroulante juste à droite (une liaison ISO sur TCP, par exemple).

Pour créer une liaison PROFINET, cliquez sur le carré vert (PROFINET) du premier appareil et tracez une ligne vers le carré PROFINET du deuxième appareil. Relâchez le bouton de la souris : la liaison PROFINET est créée.

Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 832) pour plus d'informations.

11.2.11 Communication API-API



Une CPU peut communiquer avec une autre CPU dans un réseau à l'aide des instructions TSEND_C et TRCV_C.

Tenez compte des points suivants lors de la configuration de la communication entre deux CPU :

- Configuration/installation : Une configuration matérielle est requise.
- Fonctions prises en charge : Lecture/écriture de données vers une CPU homologue
- Aucun commutateur Ethernet n'est nécessaire pour la communication un à un ; un commutateur Ethernet est obligatoire pour plus de deux unités dans un réseau.

Tableau 11- 51 Etapes requises pour configurer la communication entre deux CPU

Etape	Tâche
1	<p>Etablissement de la liaison de communication matérielle</p> <p>Une interface PROFINET établit la connexion physique entre deux CPU. Comme la fonction Auto-Cross-Over est intégrée à la CPU, vous pouvez utiliser un câble Ethernet standard ou croisé pour l'interface. Un commutateur Ethernet n'est pas nécessaire pour connecter les deux CPU.</p> <p>Reportez-vous à "Communication avec une console de programmation, Etablissement de la liaison de communication matérielle" (Page 945) pour plus d'informations.</p>
2	<p>Configuration des appareils</p> <p>Vous devez configurer deux CPU dans votre projet.</p> <p>Reportez-vous à "Communication avec une console de programmation, Configuration des appareils" (Page 946) pour plus d'informations.</p>
3	<p>Configuration des liaisons réseau logiques entre deux CPU</p> <p>Reportez-vous à "Communication API-API, Configuration des liaisons réseau logiques entre deux appareils" (Page 950) pour plus d'informations.</p>
4	<p>Configuration d'une adresse IP dans votre projet</p> <p>Utilisez le même processus de configuration ; vous devez toutefois configurer des adresses IP pour les deux CPU (par exemple, API_1 et API_2).</p> <p>Reportez-vous à "Configuration des appareils, Configuration d'une adresse IP pour une CPU dans votre projet" (Page 840) pour plus d'informations.</p>
5	<p>Configuration des paramètres d'émission et de réception</p> <p>Vous devez configurer des instructions TSEND_C et TRCV_C dans les deux CPU pour activer la communication entre elles.</p> <p>Reportez-vous à "Configuration de la communication entre deux CPU, Configuration des paramètres d'émission et de réception" (Page 951) pour plus d'informations.</p>
6	<p>Test du réseau PROFINET</p> <p>Vous devez charger la configuration pour chaque CPU.</p> <p>Reportez-vous à "Configuration des appareils, Test du réseau PROFINET" (Page 845) pour plus d'informations.</p>

11.2.11.1 Configuration des liaisons réseau logiques entre deux appareils

Une fois que vous avez configuré le châssis avec la CPU, vous être prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Cliquez d'abord sur l'onglet "Liaisons", puis sélectionnez le type de liaison dans la liste déroulante juste à droite (une liaison ISO sur TCP, par exemple).

Pour créer une liaison PROFINET, cliquez sur le carré vert (PROFINET) du premier appareil et tracez une ligne vers le carré PROFINET du deuxième appareil. Relâchez le bouton de la souris : la liaison PROFINET est créée.

Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 832) pour plus d'informations.

11.2.11.2 Configuration du routage local/partenaire entre deux appareils

Configuration des paramètres généraux

Vous indiquez les paramètres de communication dans la boîte de dialogue de configuration "Propriétés" de l'instruction de communication. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction.

Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 833)" pour plus d'informations.

Vous définissez les TSAP ou ports à utiliser dans la section "Détails de l'adresse" de la boîte de dialogue "Paramètres de liaison". Vous indiquez le point TSAP ou le port d'une liaison dans la CPU dans la zone "TSAP local". Vous indiquez le point TSAP ou le port affecté à la liaison dans votre CPU partenaire dans la zone "TSAP partenaire".

11.2.11.3 Configuration des paramètres d'émission et de réception

On utilise des blocs de communication (par exemple, TSEND_C et TRCV_C) pour établir des liaisons entre deux CPU. Avant que la CPU ne puisse entamer une communication PROFINET, vous devez configurer les paramètres pour l'émission et le réception de messages. Ces paramètres conditionnent la façon dont la communication fonctionne lorsque des messages sont envoyés à un appareil cible ou reçu d'un appareil cible.

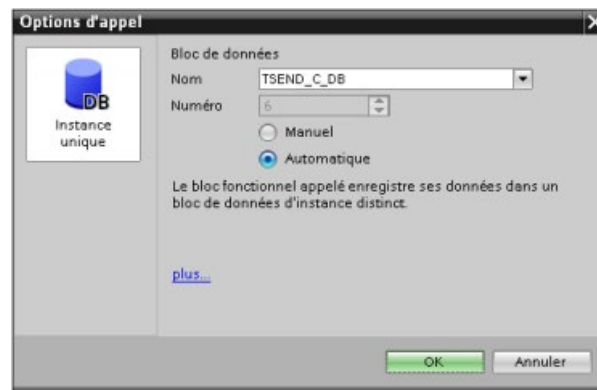
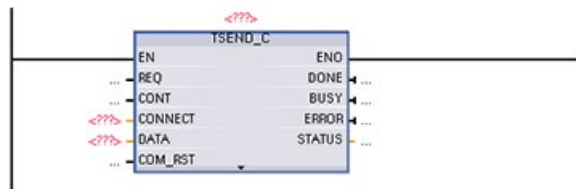
Configuration des paramètres d'émission de l'instruction TSEND_C

TSEND_C, instruction

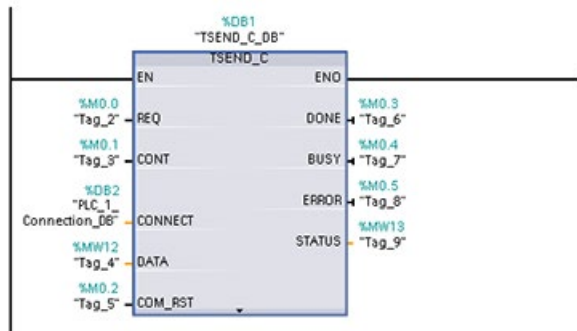
L'instruction TSEND_C (Page 863) crée une liaison de communication vers une station partenaire. La liaison est configurée, établie et automatiquement surveillée jusqu'à ce que sa coupure soit demandée par l'instruction. L'instruction TSEND_C combine les fonctions des instructions TCON, TDISCON et TSEND.

Vous pouvez configurer comment une instruction TSEND_C envoie des données dans la Configuration des appareils de STEP 7. Pour commencer, vous insérez l'instruction dans le programme à partir du dossier "Communication" dans la task card "Instructions".

L'instruction TSEND_C s'affiche avec la boîte de dialogue d'options d'appel dans laquelle vous indiquez un DB pour le stockage des paramètres de l'instruction.



Vous pouvez affecter des adresses de variables aux entrées et sorties comme illustré dans la figure ci-après :



Configuration des paramètres généraux

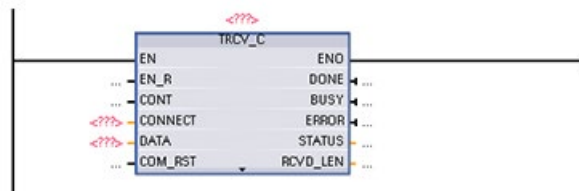
Vous indiquez les paramètres dans la boîte de dialogue de configuration Propriétés de l'instruction TSEND_C. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction TSEND_C.

Configuration des paramètres de réception de l'instruction TRCV_C

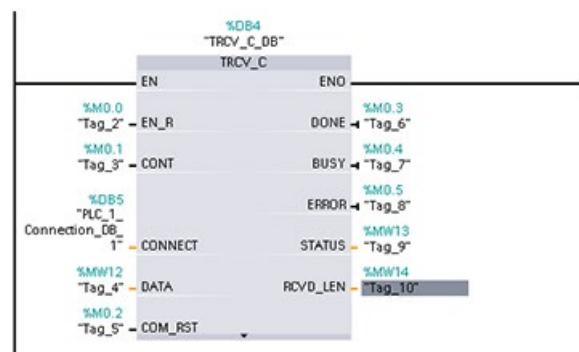
Instruction TRCV_C

L'instruction TRCV_C (Page 863) crée une liaison de communication vers une station partenaire. La liaison est configurée, établie et automatiquement surveillée jusqu'à ce que sa coupure soit demandée par l'instruction. L'instruction TRCV_C combine les fonctions des instructions TCON, TDISCON et TRCV.

Vous pouvez configurer comment une instruction TRCV_C reçoit des données dans la configuration CPU de STEP 7. Pour commencer, insérez l'instruction dans le programme à partir du dossier "Communication" dans la task card "Instructions". L'instruction TRCV_C s'affiche avec la boîte de dialogue d'options d'appel dans laquelle vous indiquez un DB pour le stockage des paramètres de l'instruction.



Vous pouvez affecter des adresses de variables aux entrées et sorties comme illustré dans la figure ci-après :



Configuration des paramètres généraux

Vous indiquez les paramètres dans la boîte de dialogue de configuration Propriétés de l'instruction TRCV_C. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction TRCV_C.

11.2.12 Configuration d'une CPU et d'un périphérique PROFINET IO

11.2.12.1 Ajout d'un périphérique PROFINET IO

Ajout d'un périphérique PROFINET IO


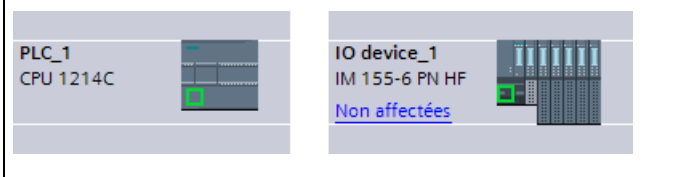
Dans le portail "Appareils & Réseaux", utilisez le catalogue du matériel pour ajouter des périphériques PROFINET IO.

Remarque

Vous pouvez utiliser STEP 7 Professional ou Basic, V11 ou une version supérieure, pour ajouter un périphérique PROFINET IO.

Développez p. ex. les dossiers suivants dans le catalogue du matériel pour ajouter un périphérique IO ET 200SP : Périphérie décentralisée, ET 200SP, modules d'interface et PROFINET. Vous pouvez alors sélectionner le coupleur dans la liste des périphériques ET 200SP (classés par numéro de référence) et ajouter le périphérique IO ET 200SP.

Tableau 11- 52 Ajout d'un périphérique IO ET 200SP à la configuration des appareils

Insérez le périphérique IO	Résultat
 <p>API_1 CPU 1214C</p> <p>6ES7 155-6AU00-0CNO</p>	 <p>PLC_1 CPU 1214C</p> <p>IO device_1 IM 155-6 PN HF Non affectées</p>

Vous pouvez maintenant connecter le périphérique PROFINET IO à la CPU :

1. Cliquez avec le bouton droit de la souris sur le lien "Non affecté" sur l'appareil et sélectionnez "Affecter un nouveau contrôleur IO" dans le menu contextuel pour afficher la boîte de dialogue "Sélectionner un contrôleur IO".
2. Sélectionnez votre CPU S7-1200 ("PLC_1" dans cet exemple) dans la liste des contrôleurs IO du projet.
3. Cliquez sur "OK" pour créer la connexion réseau.

Vous pouvez également aller dans le portail "Appareils & Réseaux" et utiliser la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet :

1. Pour créer une liaison PROFINET, cliquez sur le carré vert (PROFINET) du premier appareil et tracez une ligne vers le carré PROFINET du deuxième appareil.
2. Relâchez le bouton de la souris : la liaison PROFINET est créée.

Pour plus d'informations, reportez-vous à "Configuration des appareils : configurer la CPU pour la communication" (Page 187).

11.2.12.2 Affectation de CPU et de noms d'appareils

Affectation de CPU et de noms d'appareils

Les liaisons réseau entre les appareils affectent également le périphérique PROFINET IO à la CPU, ce qui est nécessaire pour que cette CPU puisse piloter le périphérique. Pour modifier cette affectation, cliquez sur le nom d'API figurant sur le périphérique PROFINET IO. Une boîte de dialogue s'ouvre, permettant de déconnecter le périphérique PROFINET IO de la CPU en cours et de le réaffecter ou de le laisser sans affectation, au choix.

Vous devez affecter un nom aux appareils sur votre réseau PROFINET pour pouvoir vous connecter à la CPU. Servez-vous de la "Vue du réseau" pour affecter un nom à vos appareils PROFINET s'ils n'en ont pas déjà un ou s'il faut le modifier. Pour ce faire, cliquez avec le bouton droit de la souris sur le périphérique PROFINET IO et sélectionnez "Affecter un nom d'appareil".

Pour chaque périphérique PROFINET IO, vous devez définir le même nom dans le projet STEP 7 et dans le périphérique PROFINET IO dans le réseau PROFINET (vous pouvez utiliser soit l'outil "En ligne & diagnostic" de STEP 7, soit l'outil de mise en service, configuration et diagnostic PRONETA pour affecter le nom d'appareil dans le réseau PROFINET). Si un nom manque ou ne coïncide pas à l'un ou l'autre endroit, le mode d'échange de données PROFINET IO ne fonctionnera pas. Reportez-vous à "Outils en ligne et de diagnostic, Affectation d'un nom à un appareil PROFINET en ligne (Page 1384)" pour plus d'informations.

11.2.12.3 Affectation d'adresses IP (Internet Protocol)

Affectation des adresses IP

Dans un réseau PROFINET, chaque appareil doit également comporter une adresse IP (protocole Internet). Cette adresse permet à l'appareil de transmettre des données dans un réseau routé plus complexe :

- Si vous avez des consoles de programmation ou d'autres périphériques réseau qui utilisent une carte adaptateur intégrée connectée à votre réseau local d'usine ou une carte adaptateur Ethernet vers USB connectée à un réseau isolé, vous devez leur affecter des adresses IP. Pour plus d'informations, reportez-vous à "Affectation d'adresses IP aux consoles de programmation et périphériques réseau" (Page 836).
- Vous pouvez également affecter une adresse IP à une CPU ou un périphérique réseau en ligne. Cela s'avère particulièrement utile lors d'une configuration d'appareil initiale. Pour plus d'informations, reportez-vous à "Affectation d'une adresse IP à une CPU en ligne (Page 838)".
- Une fois que vous avez configuré votre CPU ou périphérique réseau dans votre projet, vous pouvez configurer les paramètres pour l'interface PROFINET, dont son adresse IP. Reportez-vous à "Configuration d'une adresse IP pour une CPU dans votre projet" (Page 840) pour plus d'informations.

11.2.12.4 Configuration du temps de cycle IO

Configuration du temps de cycle IO

Un périphérique PROFINET IO reçoit de nouvelles données de la CPU périodiquement selon un "cycle IO". Le temps d'actualisation peut être configuré séparément pour chaque périphérique et détermine l'intervalle de temps auquel des données sont transmises entre la CPU et le périphérique.

STEP 7 calcule automatiquement le temps d'actualisation "cycle IO" dans le paramétrage par défaut de chaque périphérique du réseau PROFINET, en tenant compte du volume de données à échanger et du nombre de périphériques affectés à ce contrôleur. Si vous ne voulez pas que le temps d'actualisation soit calculé automatiquement, vous pouvez modifier ce paramètre.

Vous indiquez les paramètres de "cycle IO" dans la boîte de dialogue de configuration "Propriétés" du périphérique PROFINET IO. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction.

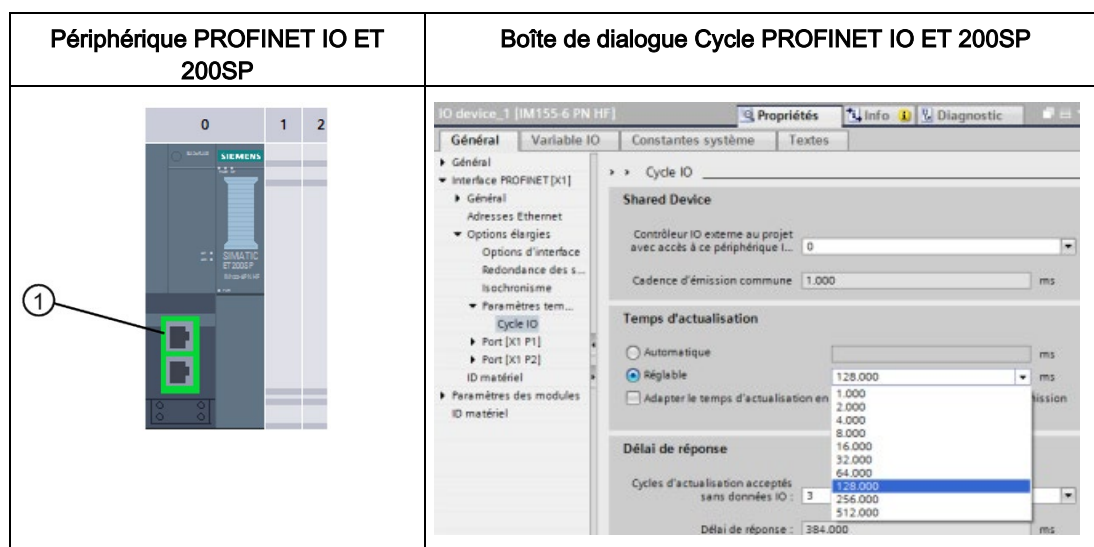
Cliquez sur le port PROFINET dans la "Vue des appareils" du périphérique PROFINET IO. Dans la boîte de dialogue "Interface PROFINET", accédez aux paramètres "Cycle IO" au moyen des sélections de menu suivantes :

- "Options élargies"
- "Paramètres temps réel"
- "Cycle IO"

Définissez le "Temps d'actualisation" du cycle IO avec les sélections suivantes :

- Sélectionnez "Calculé automatiquement" pour avoir un temps d'actualisation approprié calculé automatiquement.
- Pour définir le temps d'actualisation vous-même, sélectionnez "Réglable" et entrez le temps d'actualisation requis en ms.

Tableau 11- 53 Configuration du temps de cycle PROFINET IO ET 200SP



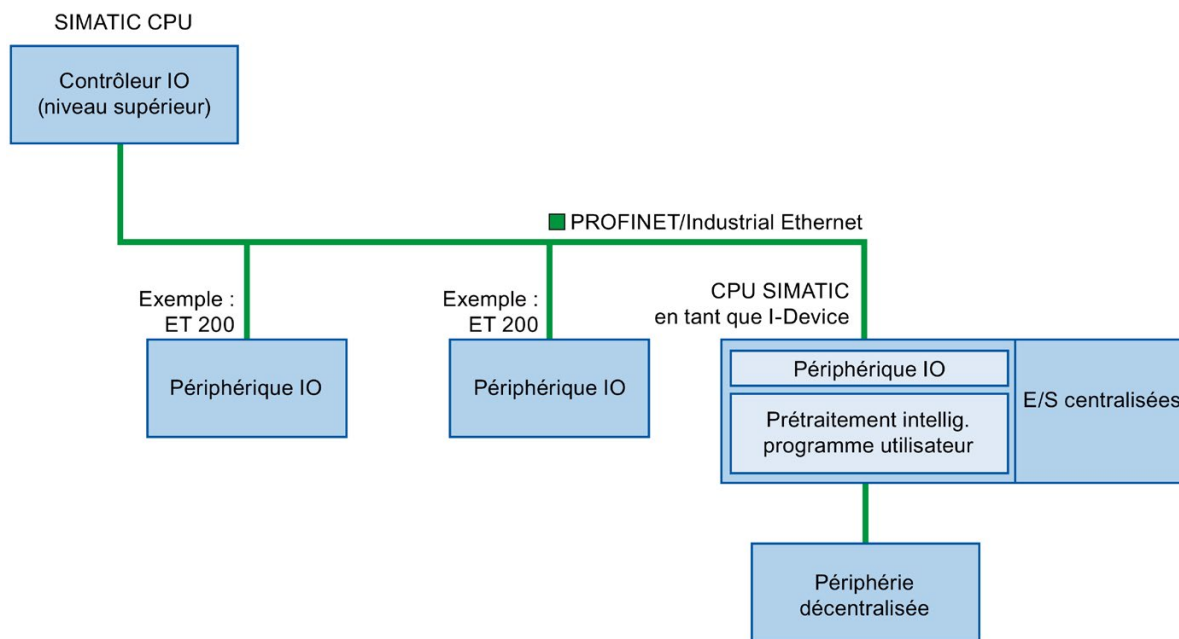
① Port PROFINET

11.2.13 Configuration d'une CPU et d'un I-device PROFINET

11.2.13.1 Fonctionnalité I-device

La fonctionnalité "I-device" (périphérique IO intelligent) d'une CPU facilite l'échange de données avec un contrôleur IO et le fonctionnement de la CPU en tant qu'unité intelligente de prétraitement de sous-processus, par exemple. L'I-device est relié en tant que périphérique IO à un contrôleur IO de "niveau supérieur".

Le prétraitement est géré par le programme utilisateur dans la CPU. Les valeurs de processus acquises dans les E/S centralisées ou la périphérie décentralisée (PROFINET IO ou PROFIBUS DP) subissent un prétraitement par le programme utilisateur et sont mises à disposition de la CPU d'une station de niveau supérieur via une interface PROFINET IO.



Conventions d'appellation pour "I-device"

Dans le reste de cette description, une CPU ou un CP possédant une fonctionnalité I-device sera simplement appelé "I-device".

11.2.13.2 Propriétés et avantages d'un I-device

Domaines d'application

Domaines d'application d'un I-device :

- Traitement réparti :
Une tâche d'automatisation complexe peut être subdivisée en unités/sous-processus plus petits. On obtient ainsi des processus gérables plus facilement et donc des sous-tâches simplifiées.
- Séparation des sous-processus :
De grands processus complexes et largement distribués peuvent être subdivisés en plusieurs sous-processus avec des interfaces gérables grâce à l'utilisation d'I-devices. Ces sous-processus peuvent, si nécessaire, être stockés dans des projets STEP 7 individuels qui pourront ultérieurement être fusionnés pour former un unique projet maître.
- Protection du savoir-faire :
Les composants peuvent être fournis avec uniquement un fichier GSD pour la description de l'interface I-device plutôt qu'avec un projet STEP 7. L'utilisateur peut donc protéger son programme puisqu'il n'est plus nécessaire de le publier.

Propriétés

Propriétés d'un I-device :

- Dissociation des projets STEP 7 :
Les créateurs et les utilisateurs d'un I-device peuvent avoir des projets d'automatisation STEP 7 complètement séparés. Le fichier GSD constitue l'interface entre les différents projets STEP 7, autorisant ainsi un lien à des contrôleurs IO standard par le biais d'une interface standardisée.
- Communication en temps réel :
L'I-device comporte un système PROFINET IO déterministe par le biais d'une interface PROFINET IO.

Avantages

L'I-device présente les avantages suivants :

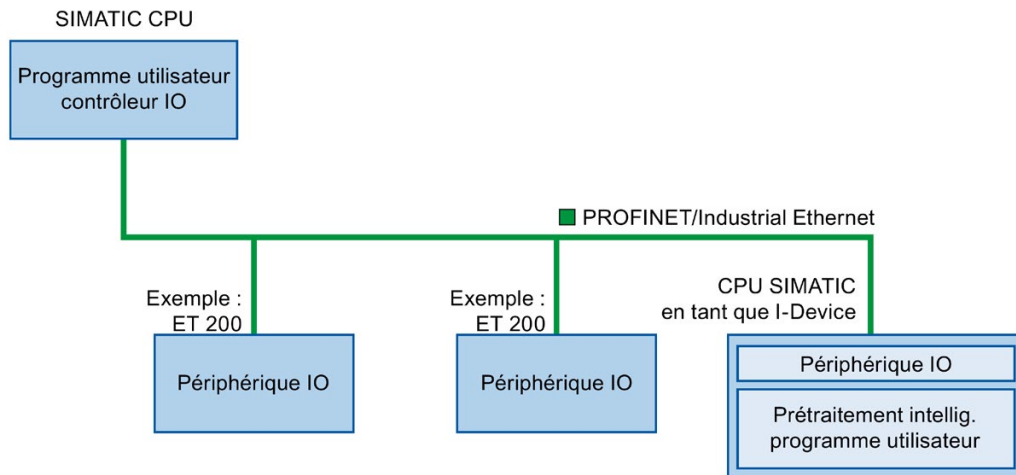
- Facilité de liaison des contrôleurs IO
- Communication en temps réel entre contrôleurs IO
- Décharge du contrôleur IO grâce à la distribution de la capacité de calcul aux I-devices
- Charge de communication réduite grâce au traitement local des données
- Gestion facilitée grâce au traitement des sous-tâches dans des projets STEP 7 séparés

11.2.13.3 Caractéristiques d'un I-device

On inclut un I-device dans un système IO comme un périphérique IO standard.

I-device sans système PROFINET IO de niveau inférieur

L'I-device ne dispose pas de sa propre périphérie décentralisée. L'affectation de la configuration et du paramétrage des I-devices dans le rôle d'un périphérique IO est la même que pour un système de périphérie décentralisée (par exemple, ET 200).



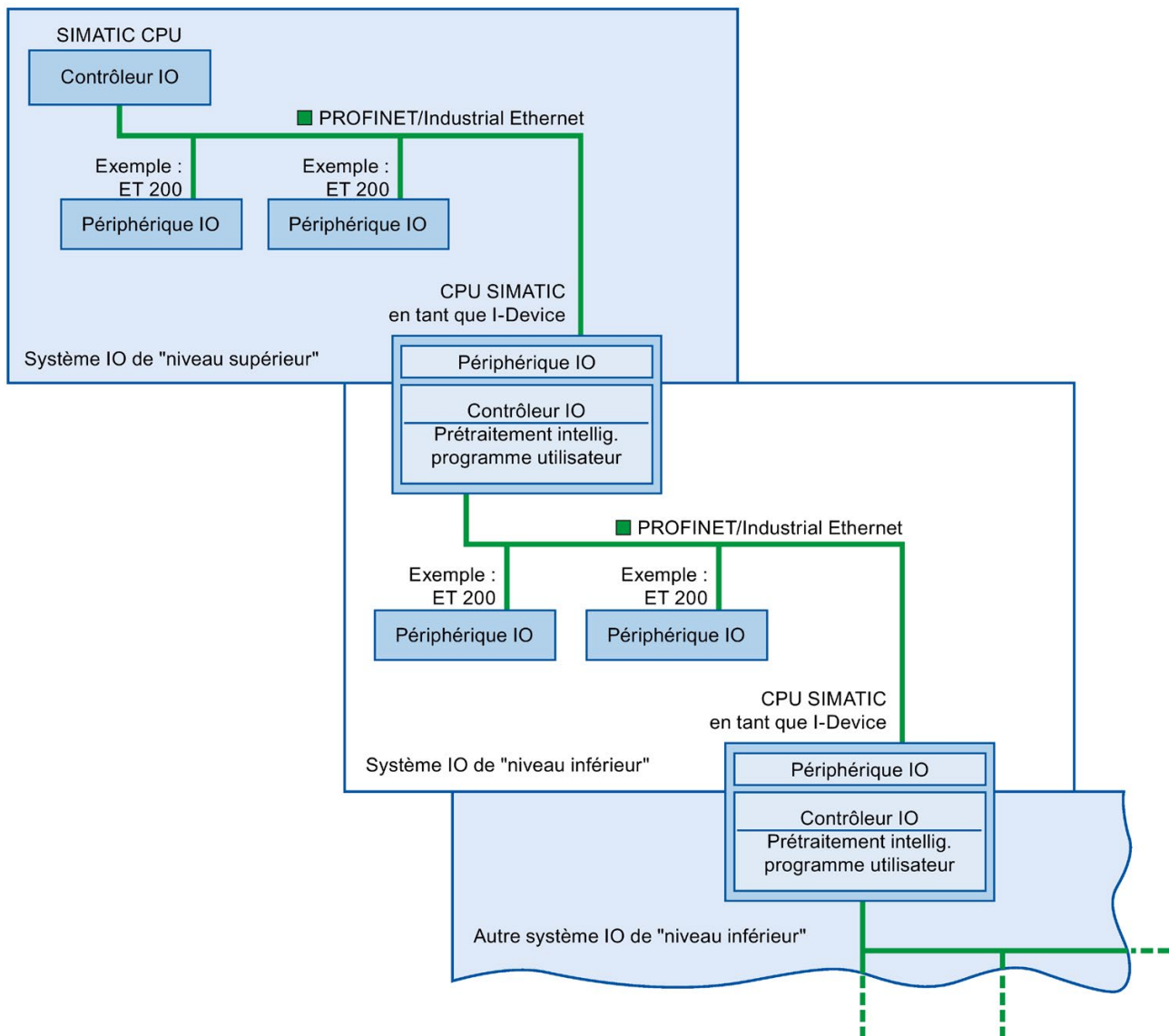
I-device avec système PROFINET IO de niveau inférieur

Selon la configuration, un I-device peut, en plus d'avoir le rôle d'un périphérique IO, également être un contrôleur IO sur une interface PROFINET.

Cela signifie que l'I-device peut faire partie d'un système IO de niveau supérieur via son interface PROFINET et peut, en tant que contrôleur IO, gérer son propre système IO de niveau inférieur.

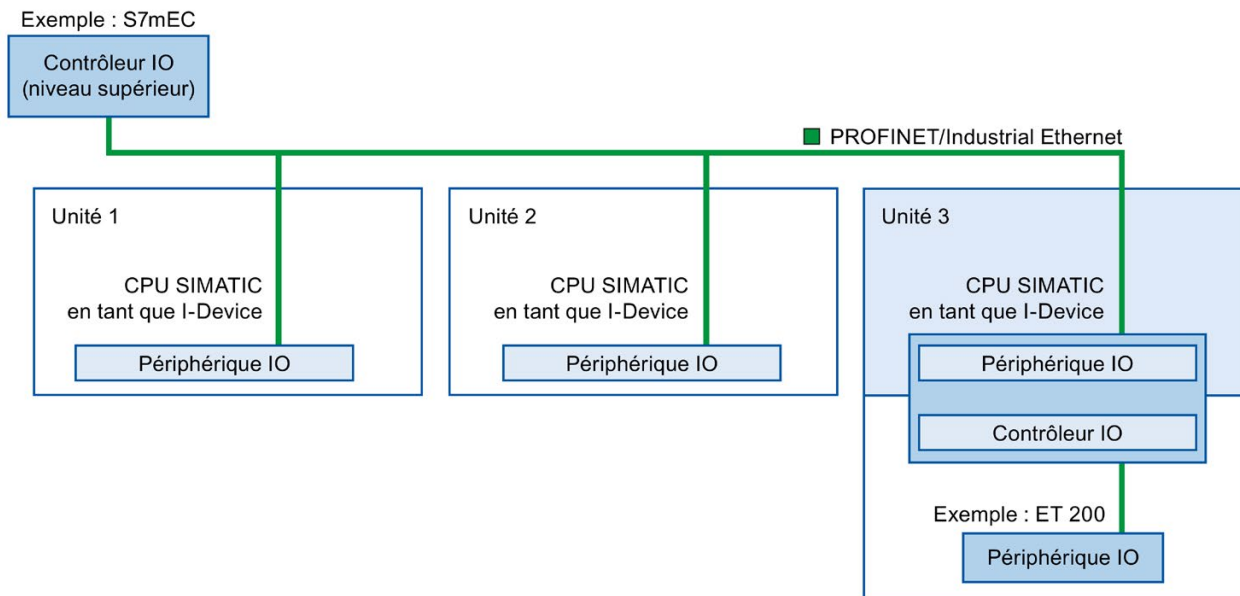
Le système IO de niveau inférieur peut, à son tour, contenir des I-devices (voir la figure ci-dessous). Il est ainsi possible d'avoir des systèmes IO structurés hiérarchiquement.

En plus de son rôle comme contrôleur IO, un I-device peut être utilisé, via une interface PROFIBUS, en tant que maître DP pour un système PROFIBUS de niveau inférieur.



Exemple : I-device en tant que périphérique IO et contrôleur IO

L'utilisation de l'I-device en tant que périphérique IO et contrôleur IO est illustrée dans l'exemple de processus d'impression ci-dessous. L'I-device pilote une unité (un sous-processus). Une unité sert, par exemple, à insérer des feuilles supplémentaires telles que des prospectus ou des brochures dans un paquet de documents imprimés.



L'unité 1 et l'unité 2 consistent chacune en un I-device à E/S centralisées. L'I-device et le système de périphérie décentralisée (ET 200, par exemple) forment conjointement l'unité 3.

Le programme utilisateur sur l'I-device est responsable du prétraitement des données du processus. Pour cette tâche, le programme utilisateur de l'I-device a besoin de paramètres par défaut (données de commande, par exemple) en provenance du contrôleur IO de niveau supérieur. L'I-device transmet les résultats (l'état de sa sous-tâche, par exemple) au contrôleur IO de niveau supérieur.

11.2.13.4 Echange de données entre système IO de niveau supérieur et système IO de niveau inférieur

Les zones de transfert constituent une interface avec le programme utilisateur de la CPU d'un I-device. Les entrées sont traitées dans le programme utilisateur et les sorties sont le résultat du traitement dans le programme utilisateur.

Les données pour la communication entre le contrôleur IO et l'I-device sont mises à disposition dans les zones de transfert. Une zone de transfert contient une unité d'information qui est échangée de manière cohérente entre le contrôleur IO et l'I-device. Vous trouverez plus d'informations sur la configuration et l'utilisation des zones de transfert sous "Configuration d'un I-device (Page 965)".

Comportement différent des zones de transfert d'entrée sur l'automate et l'I-device en cas de perte de réseau

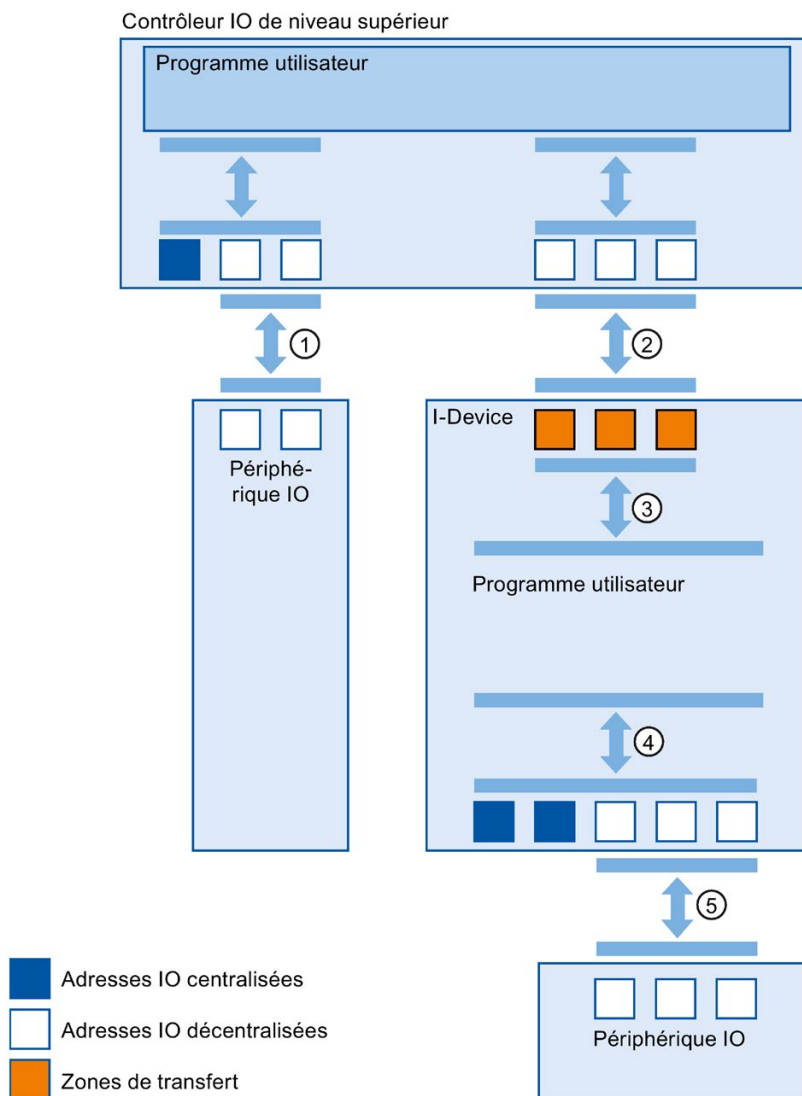
Sur l'automate, la CPU écrit zéro dans les zones de transfert d'entrée lors d'une perte de réseau. Sur l'I-device, les zones de transfert d'entrée conservent leurs dernières valeurs.

Vous pouvez configurer votre système pour éviter cette situation pour le cas général de l'I-device (périphérique I non partagé). Effacez, à cet effet, les zones de transfert d'entrée pour l'I-device dans un OB de défaillance de châssis ou de station pour un événement apparaissant. Procédez comme suit :

1. Ajoutez un OB de défaillance de châssis ou de station à votre projet (ce sera l'OB 86 par défaut).
2. Ajoutez à l'OB une logique mettant les valeurs des entrées pour l'I-device à zéro quand la variable de démarrage de LADDR indique la valeur de l'ID matérielle de l'I-device et que la variable de démarrage de Event_Class indique un événement apparaissant :
 - Vous trouverez l'ID matérielle de l'I-device dans la table de variables standard dans l'onglet "Constantes système". L'ID matérielle est un type de "HW_Device" et le nom de la variable indique qu'il s'agit d'un périphérique I (par exemple, "Local~PROFINET_interface_1~IODevice").
 - La valeur "16#39" dans "Event_Class" indique un événement apparaissant. Si la variable d'entrée "Event_Class" contient la valeur "16#39", cela signifie que l'OB de défaillance de châssis ou de station est actuellement actif (et non désactivé).

Flux d'échange de données

La figure suivante montre l'échange de données entre le système IO de niveau supérieur et le système IO de niveau inférieur. La légende numérotée explique les relations de communication individuelles.



- ① **Echange de données entre contrôleur IO de niveau supérieur et périphérique IO normal**
De cette manière, le contrôleur IO et les périphériques IO échangent des données par le biais de PROFINET.

- ② **Echange de données entre contrôleur IO de niveau supérieur et I-device**
De cette manière, le contrôleur IO et l'I-device échangent des données par le biais de PROFINET.
L'échange de données entre un contrôleur IO de niveau supérieur et un I-device est basé sur la relation contrôleur IO/périphérique IO classique.
Pour le contrôleur IO de niveau supérieur, les zones de transfert des I-devices représentent des sous-modules d'une station préconfigurée.
Les données de sortie du contrôleur IO sont les données d'entrée de l'I-device. De manière analogue, les données d'entrée du contrôleur IO sont les données de sortie de l'I-device.
- ③ **Relation de transfert entre le programme utilisateur et la zone de transfert**
De cette manière, le programme utilisateur et la zone de transfert échangent des données d'entrée et de sortie.
- ④ **Echange de données entre le programme utilisateur et les E/S de l'I-device**
De cette manière, le programme utilisateur et les E/S centralisées/la périphérie décentralisée échangent des données d'entrée et de sortie.
- ⑤ **Echange de données entre l'I-device et un périphérique IO de niveau inférieur**
De cette manière, l'I-device et ses périphériques IO échangent des données. Le transfert de données se fait par le biais de PROFINET.

11.2.13.5 Configurer un I-Device

Vous avez, au fond, deux possibilités de configuration :

- Configuration d'un périphérique I au sein d'un projet
- Configuration d'un périphérique I qui est utilisé dans un autre projet ou dans un autre système d'ingénierie.

Si vous configurez un I-device pour un autre projet ou un autre système d'ingénierie, vous pouvez le faire avec STEP 7 en exportant un I-device configuré dans un fichier GSD. Vous importez le fichier GSD dans l'autre projet ou dans l'autre système d'ingénierie comme les autres fichiers GSD. Dans ce fichier GSD sont enregistrées entre autres les zones de transfert pour l'échange de données.

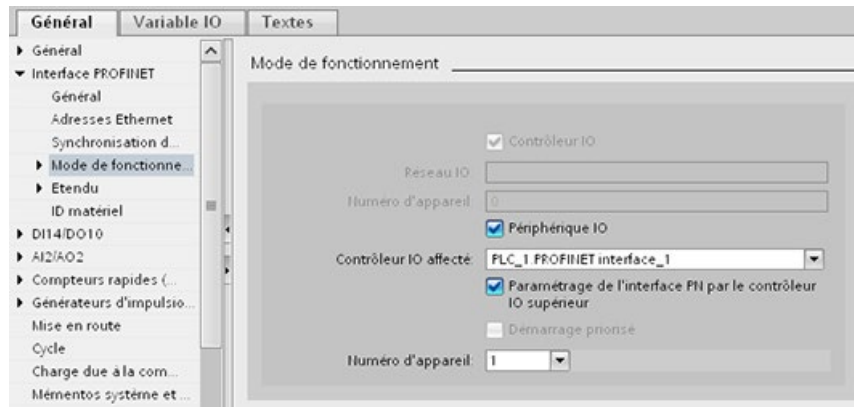
Remarque

Lorsque vous utilisez le S7-1200 en tant que I-device partagé et en tant que contrôleur, veillez à augmenter les temps d'actualisation de l'I-device PROFINET et de PROFINET IO afin d'alléger l'impact sur les performances de communication. Le système est très stable et fonctionne bien lorsque vous sélectionnez 2 ms comme temps d'actualisation d'un temps I-device PROFINET individuel et que vous sélectionnez 2 ms comme temps d'actualisation d'un temps PROFINET IO individuel.

Configuration d'un périphérique I au sein d'un projet

1. Déplacez par glisser-déposer une CPU PROFINET du catalogue du matériel dans la vue de réseau.
2. Déplacez par glisser-déposer une CPU PROFINET, qu'il est aussi possible de paramétrer comme périphérique IO, du catalogue du matériel dans la vue de réseau. Cet appareil est paramétré comme périphérique I (par ex. CPU 1215C).
3. Sélectionnez l'interface PROFINET du périphérique I.
4. Sélectionnez dans la fenêtre d'inspection dans la fenêtre de navigation locale l'entrée "Type de fonctionnement" et cochez la case "Périphérique IO".
5. La liste déroulante "Contrôleur IO affecté" vous permet maintenant de sélectionner le contrôleur IO.

Lorsque vous avez sélectionné le contrôleur IO, la mise en réseau et le système IO entre les deux appareils s'affichent dans la vue réseau.



6. Avec la case "Paramétrage de l'interface PN par le contrôleur IO supérieur", vous déterminez si l'interface sera paramétrée par le périphérique I lui-même ou par le contrôleur IO de niveau supérieur.

Si vous utilisez le périphérique I avec un réseau IO subordonné, l'interface PROFINET du périphérique I (par ex. paramètres de port) ne peut pas être paramétrée par le contrôleur IO supérieur.

7. Configurez les zones de transfert. Vous trouverez les zones de transfert dans la navigation locale, section "Communication périphérique I" :
- Cliquez dans le premier champ de la colonne "Zones de transfert". STEP 7 attribue un nom prédéfini que vous pouvez modifier.
 - Sélectionnez le type de la relation de communication : actuellement, seul CD ou F-CD peut être sélectionné.
 - Les adresses sont paramétrées automatiquement ; corrigez les adresses si nécessaire et déterminez la longueur de la zone de transfert qui doit assurer un transfert cohérent.



8. Pour chaque zone de transfert, une entrée distincte est générée dans la fenêtre de navigation locale. Si vous sélectionnez l'une de ces entrées, vous pouvez adapter ou corriger et commenter les informations sur la zone de transfert.

Remarque

Si vous configurez un S7-1200 en tant qu'I-device, la taille maximale d'une zone de transfert est de 1 024 octets d'entrée ou de sortie. Il peut exister des facteurs restrictifs en fonction des E/S locales ainsi que des limitations d'espace d'adressage sur le contrôleur.

Configuration d'un périphérique I à l'aide d'un fichier GSD

Si vous utilisez le périphérique I dans un autre projet ou si le périphérique I est utilisé dans un autre système d'ingénierie, configurez toujours le contrôleur IO de rang supérieur et le périphérique I de la manière décrite ci-dessus.

Cependant, vous cliquerez après la configuration des zones de transfert sur le bouton "Export" pour générer un fichier GSD à partir du périphérique I. Ce fichier GSD représente le périphérique I configuré dans l'autre projet.

Le bouton "Export" se trouve à la section "Communication périphérique I" dans la fenêtre d'inspection.

La configuration matérielle est compilée et la boîte de dialogue "Export" s'ouvre.

Attribuez un nom au représentant du périphérique I ainsi qu'une description dans les champs prévus à cet effet. Cliquez sur le bouton "Export" pour terminer la procédure.

Importez ensuite le fichier GSD par ex. dans un autre projet.

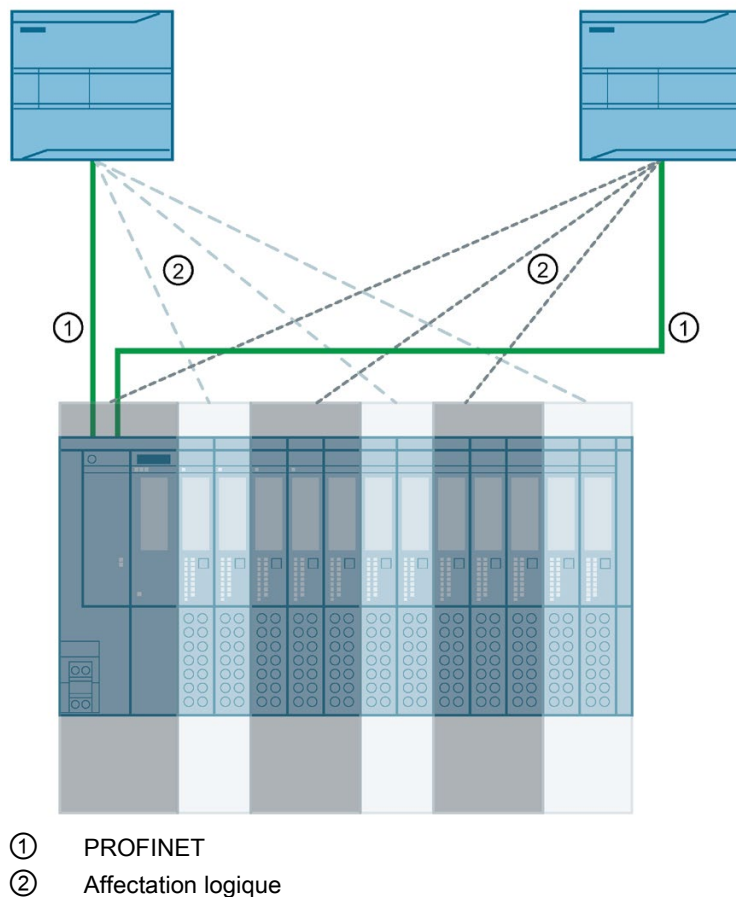
11.2.14 Appareils partagés

11.2.14.1 Fonctionnalité appareil partagé

De nombreux contrôleurs IO sont souvent utilisés dans des systèmes plus importants ou distribués à plus grande échelle.

Sans la fonction "Appareil partagé", chaque module de périphérie décentralisée d'un périphérique IO est affecté au même contrôleur IO. Si les capteurs physiquement proches les uns des autres doivent fournir des données à différents contrôleurs IO, plusieurs périphériques IO sont requis.

La fonction "Appareil partagé" permet aux modules ou aux sous-modules d'un périphérique IO d'être répartis entre différents contrôleurs IO. Cela permet aux concepts d'automatisation d'être flexibles. Vous avez, par exemple, la possibilité de combiner des modules de périphérie décentralisée posés les uns près des autres avec un périphérique IO.



Principe

L'accès aux sous-modules de l'appareil partagé est alors réparti entre les contrôleurs IO individuels. Chaque sous-module de l'appareil partagé est exclusivement affecté à un contrôleur IO.

Exigence (configuration GSD)

- STEP 7 V12 Service Pack 1 ou version ultérieure
- CPU S7-1200 avec firmware de version V4.1 ou ultérieure comme contrôleur IO
- Le périphérique IO supporte la fonction appareil partagé, par exemple, module d'interface IM 155-5 PN ST
- Le fichier GSD pour configurer le périphérique IO est installé
- Une CPU S7-1200 configurée comme un I-device supporte la fonctionnalité Appareil partagé. Vous devez exporter le fichier GSD PROFINET pour l'I-device depuis STEP 7 (V5.5) puis l'importer dans STEP 7 (Portail TIA).

Configurer l'accès

Le périphérique IO doit être présent dans plusieurs projets de sorte que les modules ou les sous-modules d'un périphérique IO puissent être affectés à différents contrôleurs IO. Un projet séparé est requis pour chaque contrôleur IO.

Vous utilisez le paramètre "Appareil partagé" du module d'interface pour déterminer les modules ou sous-modules auxquels le contrôleur IO a accès :

- Si le contrôleur IO local a accès au module configuré, sélectionnez le nom du contrôleur IO dans la liste.
- Si le contrôleur IO d'un projet différent et qui n'est pas le contrôleur IO local doit avoir accès au module configuré, sélectionnez l'entrée "---".

La configuration est cohérente vis-à-vis de l'accès si chaque module ou sous-module dans exactement un projet est affecté à un contrôleur IO.

Module ou sous-module affecté à un autre contrôleur IO

Le paragraphe ci-dessous décrit les conséquences du paramétrage "---" du paramètre "Appareil partagé" à partir du point de vue du contrôleur IO local.

Dans ce cas, le contrôleur IO local n'a pas accès au module configuré de cette manière. Plus particulièrement, cela signifie :

- Aucun échange de données avec le module ou sous-module
- Aucune réception d'alarmes ou de diagnostics, ce qui signifie aucun affichage du statut des diagnostics dans la vue en ligne
- Aucune affectation de paramètre du module ou du sous-module

Paramétrage des propriétés temps réel

STEP 7 calcule la charge due à la communication et par conséquent les temps d'actualisation résultants. Vous devez entrer le nombre de contrôleurs IO externes au projet dans le projet dans lequel l'interface PROFINET de l'appareil partagé est affectée au contrôleur IO de sorte qu'un calcul soit possible avec des configurations d'appareil partagé.

Le nombre possible maximum de contrôleurs IO pour l'appareil partagé dépend de l'appareil. Ce nombre est sauvegardé dans le fichier GSD de l'appareil partagé.

Vous pouvez paramétrer une cadence d'émission très courte (minimum de 1 ms) avec une CPU S7-1200 en tant que contrôleur IO. L'horloge de transmission peut être plus courte que l'horloge de transmission la plus courte supportée par l'appareil partagé. Dans ce cas, l'appareil partagé fonctionne au moyen du contrôleur IO avec une horloge de transmission qu'il supporte (adaptation à l'horloge de transmission).

Exemple : Une CPU prend en charge des cadences d'émission à partir de 1 ms. Un périphérique IO configuré prend en charge des cadences d'émission à partir de 1,25 ms ; un autre périphérique IO prend en charge des cadences d'émission à partir de 1 ms. Dans ce cas, vous avez la possibilité de paramétrer la cadence d'émission courte de 1 ms pour la CPU. La CPU fera fonctionner le périphérique IO "lent" avec la cadence d'émission de 1,25 ms.

Règles pour la configuration

- Les contrôleurs IO qui utilisent l'appareil partagé sont créés dans différents projets. Dans chaque projet, veillez soigneusement à ce que l'appareil partagé soit configuré de manière identique dans chaque station. Seul un contrôleur IO peut avoir l'accès intégral à un sous-module. Des défaillances dans la configuration entraînent un dysfonctionnement de l'appareil partagé.
- Les adresses E/S d'un module ou d'un sous-module peuvent uniquement être modifiées si un module ou un sous-module est affecté au contrôleur IO dans le même projet.
- L'appareil partagé doit avoir les mêmes paramètres IP et le même nom d'appareil dans chaque projet.
- L'horloge de transmission doit être identique pour tous les contrôleurs IO qui ont accès à l'appareil partagé.
- L'ID sous-réseau S7 du sous-réseau auquel l'appareil partagé est connecté doit être identique dans tous les projets.
- Les fonctions suivantes sont uniquement disponibles si l'interface PROFINET de l'appareil partagé est affectée au contrôleur IO local :
 - Démarrage prioritaire
 - Affectation de paramètres des propriétés de port

Conditions limites

Les conditions limites suivantes s'appliquent lorsqu'une configuration d'appareil partagé s'applique à plusieurs projets :

- Les adresses de modules ou de sous-modules qui ne sont pas affectées à ce contrôleur IO sont manquantes dans la présentation de l'adresse de chaque contrôleur IO qui a accès à un appareil partagé.
- Les modules ou sous-modules qui ne sont pas affectés ne sont pas pris en compte dans le calcul des limites de configuration pour l'appareil partagé pendant le contrôle de cohérence. Pour cette raison, vous devez vérifier vous-même que le nombre maximum de sous-modules ou que la quantité maximale de données IO cycliques pour l'appareil partagé ne soit pas dépassée. Pour plus d'informations sur les quantités maximales, reportez-vous à la documentation concernant les appareils que vous utilisez.
- Les erreurs de configuration comme l'affectation d'un module ou d'un sous-module à plusieurs contrôleurs IO ne sont pas détectées dans STEP 7.
- Les CPU qui sont chargées avec une configuration d'appareil partagé ne possèdent aucune information quant au fait que le périphérique IO soit ou non un appareil partagé. Les modules ou les sous-modules qui sont affectés à d'autres contrôleurs IO et par conséquent à d'autres CPU sont manquants dans la configuration chargée. Ces modules ou sous-modules ne sont par conséquent affichés ni dans le serveur web de la CPU ni sur l'écran de la CPU.

11.2.14.2 Exemple : Configurer un appareil partagé (configuration GSD)

Cet exemple décrit la manière dont configurer un système de périphérie décentralisée comme un appareil partagé avec STEP 7 V13 SP1 ou une version ultérieure.

Une configuration "décentralisée" avec différents outils d'ingénierie pour différentes familles de contrôleurs IO est possible. La procédure décrite ci-dessous se base sur STEP 7 dans sa version V13 SP1 et se limite à une configuration avec deux contrôleurs IO de la série S7-1200 qui partagent un appareil partagé.

Cet exemple crée deux projets avec un contrôleur IO chacun :

- Contrôleur1
- Contrôleur2

Vous devez créer l'appareil partagé dans les deux projets, même s'il compte physiquement pour un et qu'il s'agit du même périphérique IO.

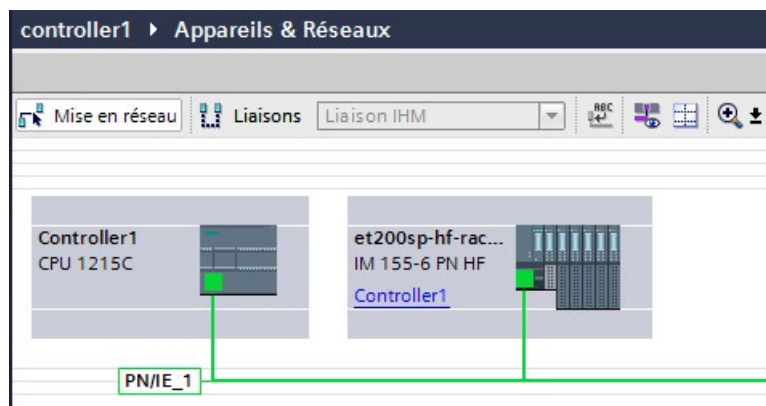
Exigences

- STEP 7 V13 SP1 ou version ultérieure
- Le périphérique IO supporte la fonctionnalité appareil partagé, (par exemple, ET 200SP IM 155-6 PN HF V3.1).
- Le fichier GSD pour configurer le périphérique IO comme appareil partagé est installé.

Procédure : Créer le projet 1

Pour créer le premier projet avec un appareil partagé, suivez ces étapes :

1. Démarrez STEP 7.
2. Créez un nouveau projet avec le nom "Contrôleur1".
3. Insérez une CPU 1215C à partir du catalogue du matériel dans la vue de réseau. Nommez-la "Contrôleur1".
4. Insérez un périphérique IO avec la fonction "Appareil partagé" (par exemple, un ET 200SP) à partir du catalogue du matériel (catalogue du matériel : Autres appareils de champ > PROFINET IO > I/O).
5. Affectez le contrôleur IO "Contrôleur1" au périphérique IO.

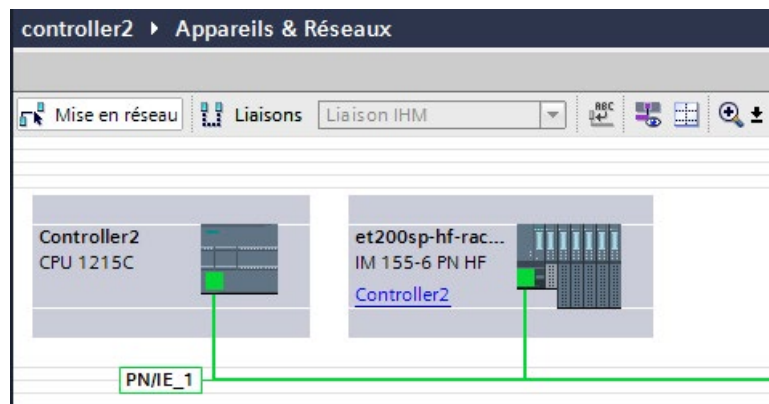


6. Double-cliquez sur le périphérique IO et insérez tous les modules et sous-modules requis à partir du catalogue du matériel dans le tableau de présentation des appareils.
7. Affectez les paramètres du module.
8. Sauvegardez le projet.

Procédure : Créer le projet 2

Pour créer le deuxième projet avec un appareil partagé, suivez ces étapes :

1. Démarrez à nouveau STEP 7.
Une nouvelle instance de STEP 7 s'ouvre.
2. Dans la nouvelle instance, créez un nouveau projet avec le nom "Contrôleur2".
3. Insérez une CPU 1215C dans la vue de réseau. Nommez-la "Contrôleur2".
4. Copiez le périphérique IO à partir du projet "Contrôleur1" et insérez le dans la vue de réseau du projet "Contrôleur2".
5. Affectez le contrôleur IO "Contrôleur2" au périphérique IO.



6. Sauvegardez le projet.

Les deux projets ont maintenant un périphérique IO structuré de manière identique qui doit être configuré dans la prochaine étape pour les différents types d'accès au contrôleur IO.

Procédure : Configurer l'accès à l'appareil partagé

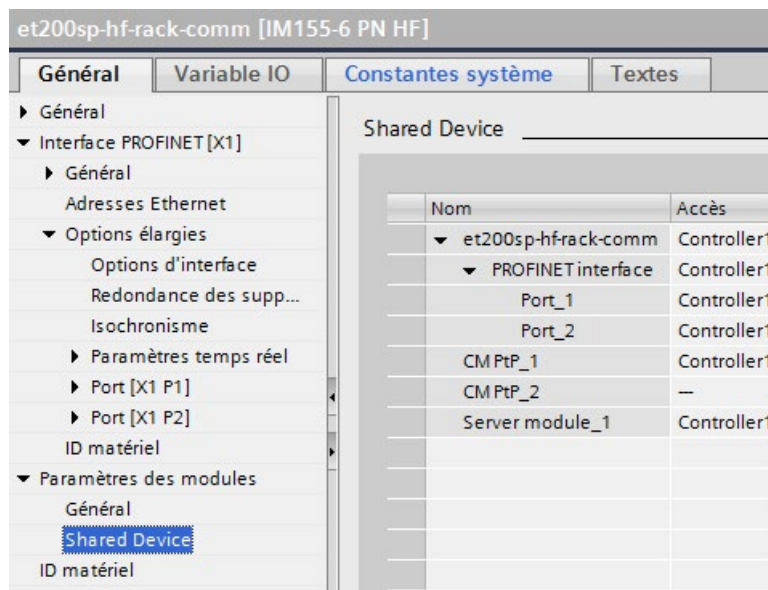
Les modules et sous-modules que vous insérez dans l'appareil partagé sont automatiquement affectés à la CPU locale. Pour modifier l'affectation, suivez ces étapes :

1. Sélectionnez le module d'interface dans la vue de réseau ou d'appareil du projet "Contrôleur1".
2. Sélectionnez la rubrique "Appareil partagé" dans la fenêtre d'inspection.

Un tableau indique quelle CPU a accès au module ou sous-module respectif pour tous les modules configurés. Le paramétrage par défaut établit que la CPU locale a accès à tous les modules et sous-modules.

3. Conservez le paramétrage "Contrôleur1" pour tous les modules et sous-modules qui doivent rester dans la plage d'adresses de la CPU locale.

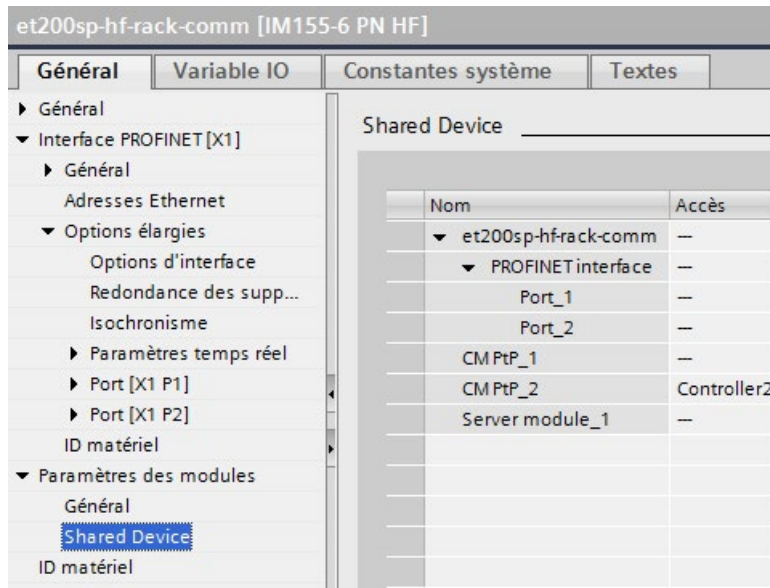
Sélectionnez le paramétrage "---" pour tous les modules et sous-modules qui doivent être situés dans la plage d'adresses de la CPU à partir du projet "Contrôleur2" (Contrôleur2). Cela signifie qu'un contrôleur IO en dehors du projet doit avoir accès au module ou au sous-module.



4. Sélectionnez le module d'interface dans la vue de réseau ou d'appareil du projet "Contrôleur2".
5. Sélectionnez la rubrique "Appareil partagé" dans la fenêtre d'inspection.

Un tableau indique quelle CPU a accès au module ou sous-module respectif pour tous les modules configurés.

6. Sélectionnez le paramétrage "---" pour tous les modules et sous-modules qui doivent être situés dans la plage d'adresses de la CPU à partir du projet "Contrôleur1" (Contrôleur1).



7. Enfin, vérifiez si les paramètres pour l'accès sont "complémentaires" pour chaque module ou sous-module dans les deux projets. Cela signifie que si la CPU locale a accès à un projet, l'option "---" doit être paramétrée dans l'autre projet et vice versa.

Remarque : L'option "---" pour l'interface PROFINET et par conséquent pour les ports établit les paramètres associés comme en lecture seule et non modifiables. Les paramètres de l'interface PROFINET et les paramètres des ports peuvent uniquement être modifiés dans le projet dans lequel l'interface PROFINET est affectée à la CPU locale. Les ports peuvent être interconnectés dans les deux projets indépendamment de cela.

8. Vérifiez si les mêmes paramètres d'adresse IP et si le même nom de l'appareil sont paramétrés pour l'appareil partagé dans tous les projets.

Vérifiez si la même ID de sous-réseau S7 est paramétrée dans tous les projets pour le sous-réseau auquel l'appareil partagé est connecté (propriétés de sous-réseau, rubrique "Général" dans la fenêtre d'inspection).

Remarque

Si vous effectuez des modifications quant à l'appareil partagé : Effectuez les mêmes modifications dans chaque projet sur l'appareil partagé. Assurez-vous qu'un seul contrôleur IO a accès à un module ou à un sous-module.

Procédure : Ajuster les paramètres temps réel

Pour s'assurer que tous les contrôleurs IO et les appareils partagés fonctionnent avec l'horloge de transmission appropriée et que les temps d'actualisation sont calculés correctement sur la base de la charge due à la communication, vous devez ajuster et vérifier les paramètres suivants :

1. Sélectionnez le projet dont les contrôleurs IO ont accès à l'interface PROFINET et aux ports de l'appareil partagé.
2. Sélectionnez le module d'interface de l'appareil partagé dans la vue de réseau.
3. Dans la fenêtre d'inspection, naviguez vers la rubrique "interface PROFINET > Options avancées > Paramétrages temps réel > cycle IO".
4. Dans la rubrique "Appareil partagé", paramétrez le nombre de contrôleurs IO externes au projet. Le nombre maximum dépend du périphérique IO (spécification dans le fichier GSD).
5. Vous devez paramétrer la même horloge de transmission pour chaque contrôleur IO qui a accès aux modules et aux sous-modules de l'appareil partagé :
 - Si vous configurez le contrôleur IO avec STEP 7 (TIA Portal) :
 - Ouvrez le projet correspondant.
 - Sélectionnez l'interface PROFINET du contrôleur IO.
 - Sélectionnez la rubrique "Options avancées > Paramétrages temps réel > Communication IO" dans la fenêtre d'inspection et paramétrez l'horloge de transmission partagée.
 - Si vous configurez le contrôleur IO avec un outil d'ingénierie différent :
 - Sélectionnez l'interface PROFINET de l'appareil partagé dans STEP 7 (TIA Portal) et lisez l'horloge de transmission sur l'appareil partagé ("Options avancées > Paramétrages temps réel").
 - Entrez l'horloge de transmission lue dans l'outil d'ingénierie.

Remarque

Si vous configurez tous les contrôleurs IO qui ont accès à l'appareil partagé dans STEP 7 (TIA Portal), vous pouvez paramétrer des horloges de transmission plus courtes sur le contrôleur IO que celles supportées par l'appareil partagé (adaptation de l'horloge de transmission).

Compilation et chargement

Vous devez compiler les configurations pour les différents contrôleurs IO et les charger dans les CPU l'une après l'autre.

En raison de la configuration appliquée à des projets séparés, STEP 7 ne met pas évidence les erreurs de cohérence en cas d'affectation de paramètres d'accès incorrecte. Voici des exemples d'affectation de paramètres d'accès incorrecte :

- Plusieurs contrôleurs IO ont accès au même module
- Les paramètres d'adresse IP ou les horloges de transmission ne sont pas identiques

Ces erreurs ne se manifestent pas avant le fonctionnement du contrôleur et sont mises en évidence comme erreurs de configuration.

11.2.14.3 Exemple : Configurer un I-device comme appareil partagé

Cet exemple décrit comment configurer un S7-1200 comme un I-device avec STEP 7 Version V13 SP1 ou une version ultérieure puis comment l'utiliser dans deux projets comme appareil partagé.

Une configuration "décentralisée" avec différents outils d'ingénierie pour différentes familles de contrôleurs IO est possible. La procédure décrite ci-dessous se base sur STEP 7 dans sa version V13 SP1 et se limite à une configuration avec deux contrôleurs IO de la famille S7-1200 qui partagent les zones de transfert d'un I-device comme appareil partagé. L'I-device lui-même est une CPU 1215C.

Cet exemple crée trois projets avec un contrôleur IO chacun :

- S7-1200-I-Device
- Contrôleur1
- Contrôleur2

Vous utilisez le projet S7-1200-I-Device pour configurer l'I-device. Vous utilisez la variante GSD PROFINET de S7-1200-I-Device dans les projets Contrôleur1 et Contrôleur2 afin d'affecter les zones de transfert dans le contrôleur IO de niveau supérieur respectif.

Concept I-device partagé

Le concept I-device partagé requiert un minimum de trois projets séparés :

- **Projet I-device** : vous configurez et programmez un I-device pour effectuer une tâche d'automatisation particulière. Vous définissez des zones de transfert comme l'interface I/O pour les contrôleurs de niveau supérieur et affectez ces zones de transfert à différents contrôleurs IO. Pour la liaison à des contrôleurs IO de niveau supérieur, vous fournissez un fichier GSD PROFINET et utilisez les zones de transfert pour accéder à l'I-device.
- **Contrôleurs qui partagent l'I-device (deux projets)** : vous utilisez l'I-device comme variante GSD PROFINET pendant la configuration du système IO PROFINET et, dans ce processus, spécifiez les adresses d'E/S sous lesquelles les contrôleurs IO accèdent aux zones de transfert.

Remarque

Si vous configurez un S7-1200 en tant qu'I-device, la taille maximale d'une zone de transfert est de 1 024 octets d'entrée ou de sortie. Il peut exister des facteurs restrictifs en fonction des E/S locales ainsi que des limitations d'espace d'adressage sur le contrôleur.

I-device

Vous affectez les paramètres suivants pour une CPU S7-1200 en tant qu'I-device :

- E/S centralisées et périphérie décentralisée
- Zones de transfert souhaitées
- Nombre de contrôleurs IO ayant accès à cet I-device (toujours plus élevé qu'1 pour un appareil partagé)

Remarque

Vous configurez l'I-device sans un contrôleur IO de niveau supérieur. En conséquence, vous pouvez uniquement utiliser les adresses d'E/S locales de la zone de transfert (correspond à "Adresse dans l'I-device") pour créer le programme utilisateur afin de modifier les adresses de la zone de transfert. Vous téléchargez l'I-device, entièrement configuré excepté pour la liaison au contrôleur IO de niveau supérieur à la CPU S7-1200.

Vous exportez un fichier GSD PROFINET à partir de la configuration de l'I-device.

Contrôleurs qui partagent l'I-device

Vous devez installer le fichier GSD PROFINET créé à partir de la configuration de l'I-device dans tous les systèmes d'ingénierie que vous utilisez pour configurer un système IO PROFINET avec cet I-device partagé. Si vous configurez toutes les utilisations de cet I-device avec STEP 7 V13 SP1, il est suffisant d'installer le fichier GSD dans STEP 7.

Vous configurez l'I-device en tant que variante GSD sur le système IO PROFINET dans les projets concernés. Dans STEP 7 V13 SP1, vous trouvez cet I-device sous "Autres appareils de champ > PROFINET IO > PLC et CP" après l'installation.

Dans chacun des projets concernés, vous affectez des zones de transfert exclusivement aux contrôleurs IO de niveau supérieur (paramétrage par défaut : tout). Vous paramétrez les autres zones de transfert sur "---" (non affecté). Lorsque vous procédez ainsi, le contrôleur IO local ne peut pas accéder à cette zone de transfert, et vous pouvez affecter la zone de transfert à un autre contrôleur IO dans un autre projet.

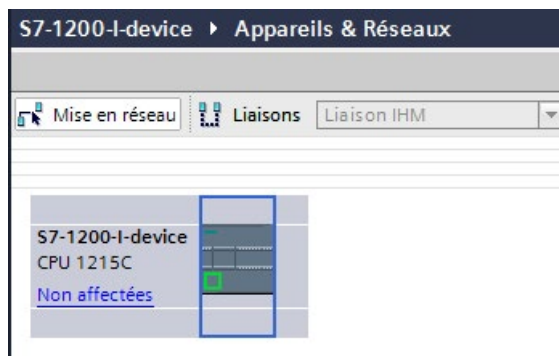
Exigences

- STEP 7 V13 SP1 ou version ultérieure
- Le périphérique IO supporte la fonctionnalité appareil partagé, (par exemple, ET 200SP IM 155-6 PN HF V3.1).
- Le fichier GSD pour configurer le périphérique IO comme appareil partagé est installé.

Procédure : Créer le projet S7-1200-I-Device

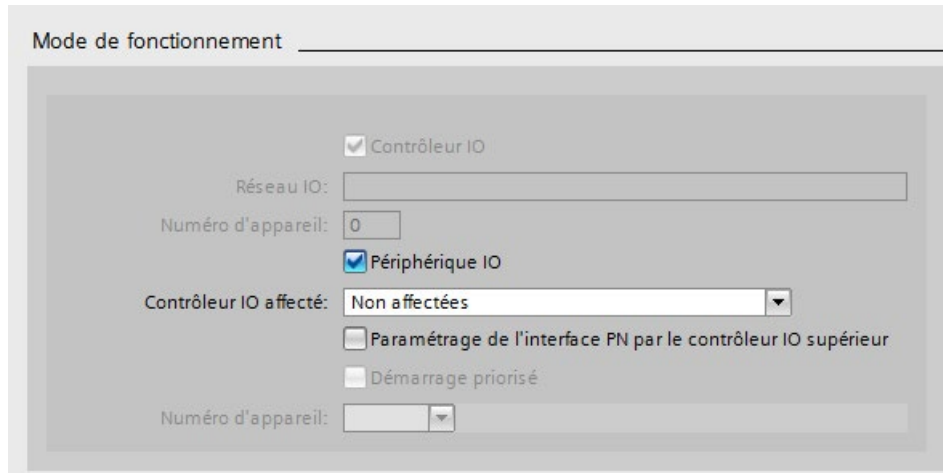
Pour créer le projet avec un I-device partagé, suivez ces étapes :

1. Démarrez STEP 7.
2. Créez un nouveau projet avec le nom "S7-1200-I-device".
3. Insérez une CPU 1215C à partir du catalogue du matériel dans la vue de réseau. Affectez le nom "S7-1200-I-device".

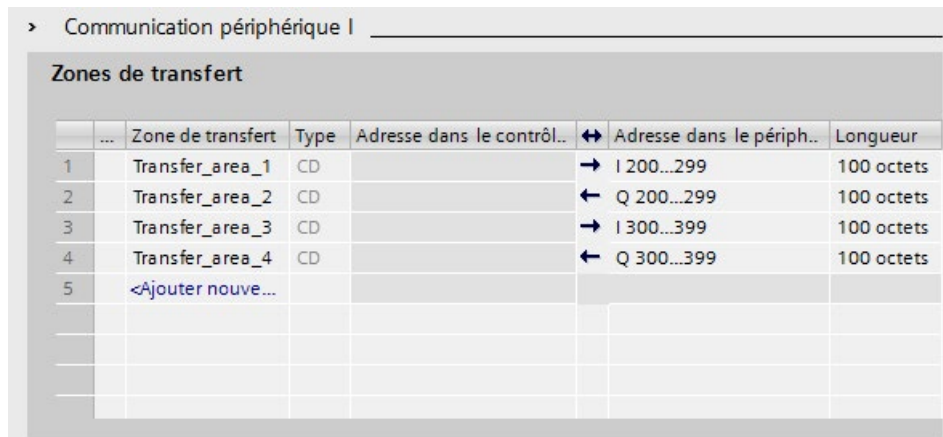


4. Double-cliquez sur le périphérique IO et configurez tous les modules et sous-modules requis.

5. Affectez les paramètres du module. Plus particulièrement, vous devez configurer les paramètres suivants pour la CPU dans la zone de l'interface PROFINET [X1] :
 - Activez l'option "Périphérique IO" dans la rubrique "Mode de fonctionnement".

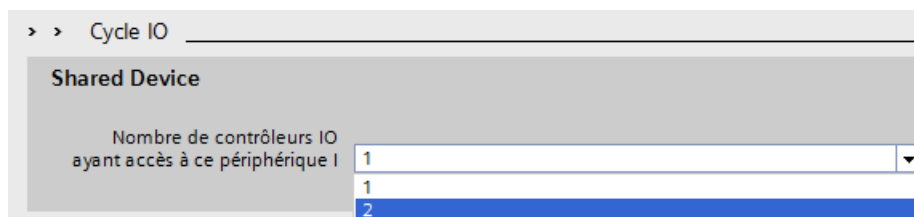


- Configurez les zones de transfert dans la rubrique "Mode de fonctionnement" > "Configuration I-device". La colonne "Adresse dans le contrôleur IO" reste vide car aucun contrôleur IO n'est affecté.

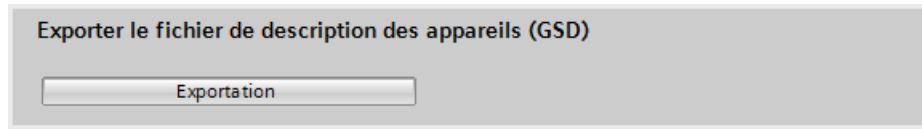


Remarque : Pour modifier une zone d'entrée en zone de sortie, et vice versa, vous devez naviguer vers la zone de la zone de transfert correspondante.

- Sélectionnez le nombre de contrôleurs IO (au moins deux) qui accéderont à l'I-device partagé pendant le fonctionnement (rubrique "Mode de fonctionnement" > "Paramétrages temps réel", rubrique "Appareil partagé").



6. Sauvegardez le projet.
7. Cliquez sur le bouton "Export" (rubrique "Mode" > "configuration I-device", paragraphe "Exporter le fichier de description de station générale (GSD)"). Si vous ne modifiez pas le nom dans la boîte de dialogue Export, le fichier GSD utilise un nom de format affecté (par exemple, "GSDML-V2.31-#Siemens-PreConf_S7-1200-I-Device-20130925-123456").



Procédure : Créer le projet Contrôleur1

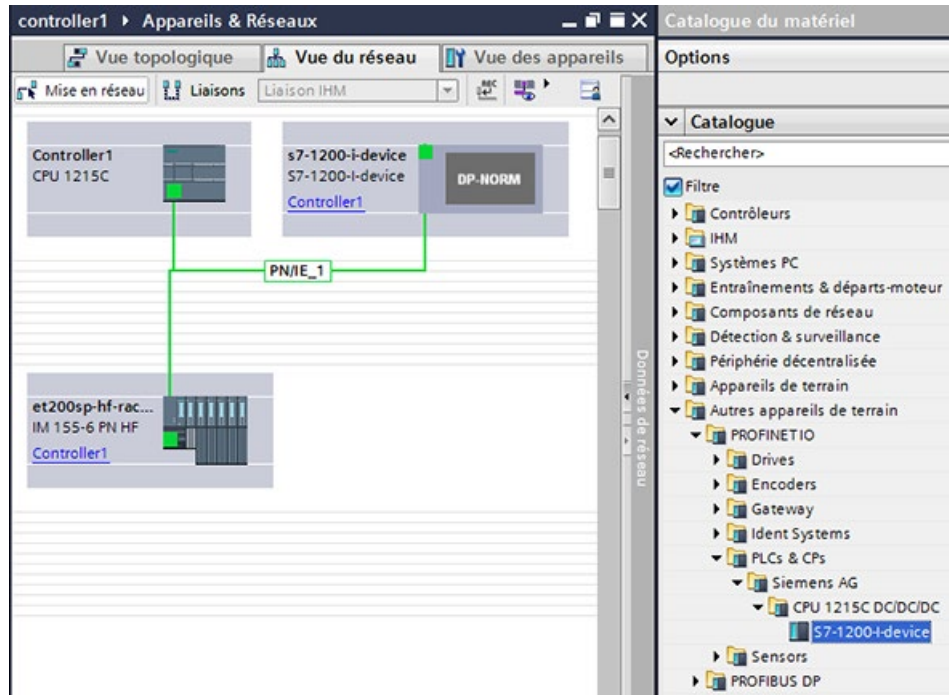
Pour créer le premier projet avec un I-device partagé, suivez ces étapes :

1. Démarrez STEP 7.
2. Installez le fichier GSD PROFINET à partir de l'exportation de la CPU de l'I-device (S7-1200-I-Device).



3. Créez un nouveau projet avec le nom "Contrôleur1".
4. Insérez la CPU 1215C dans la vue de réseau. Le nom de la CPU doit être "Contrôleur1".
5. Insérez l'I-device à partir du catalogue du matériel (catalogue du matériel : Autres appareils de champ > PROFINET IO > API et CP).

6. Affectez le contrôleur IO "Contrôleur1" à l'I-device.



7. Sélectionnez la rubrique "Appareil partagé" dans toutes les propriétés de l'I-device :
 - Dans le tableau, toutes les zones de transfert et l'interface PROFINET sont affectées au contrôleur IO local (Contrôleur1).
 - Définissez les zones de transfert auxquelles la CPU du Contrôleur1 ne doit **pas** avoir accès. Sélectionnez l'entrée "---" pour ces zones. Ces zones de transfert sont fournies pour le Contrôleur2.

The screenshot displays the SIMATIC Manager interface for configuring a shared device. The top section shows a network diagram with three main components: 'Controller1 CPU 1215C', 's7-1200-i-device S7-1200-I-device Controller1', and 'et200sp-hf-rac... IM 155-6 PN HF Controller1'. These are connected via a 'PN/IE_1' link. The bottom section shows the configuration window for the 's7-1200-i-device [Module]' with the 'Shared Device' tab selected. The 'Shared Device' tab contains a table with the following data:

Nom	Accès
s7-1200-i-device	Controller1
Interface	---
Port 1	---
Port 2	---
Transfer_area_1	Controller1
Transfer_area_2	Controller1
Transfer_area_3	---
Transfer_area_4	Controlli
	Controller1

- Vous pouvez adapter les adresses à partir de la vue des appareils du contrôleur IO dans la présentation des appareils. Pour ouvrir la présentation des appareils, double-cliquez sur l'I-device.

Module	Châssis	Empla.	Adresse I	Adresse...	Type	Numéro de article	Firmware	C...	Accès
▼ s7-1200-i-device	0	1		256...355	S7-1200-I-device	6ES7 215-1AG40-0XB0	V4.1		Controller1
Transfer_area_1	0	1 1000		256...355	Transfer_area_1				Controller1
Transfer_area_2	0	1 1001	256...355		Transfer_area_2				Controller1
Transfer_area_3	0	1 1002			Transfer_area_3				—
Transfer_area_4	0	1 1003			Transfer_area_4				—
▶ Interface	0	1 X1			s7-1200-i-device				—

- Sauvegardez le projet.

Procédure - Créer le projet Contrôleur2

Pour créer le deuxième projet avec un appareil partagé, suivez ces étapes :

- Démarrez à nouveau STEP 7.
Une nouvelle instance de STEP 7 s'ouvre.
- Dans la nouvelle instance, créez un nouveau projet avec le nom "Contrôleur2".
- Insérez la CPU 1215C dans la vue de réseau. Affectez le nom "Contrôleur2".
- Insérez l'I-device à partir du catalogue du matériel (catalogue du matériel : Autres appareils de champ > PROFINET IO > API et CP).
- Affectez le contrôleur IO "Contrôleur2" à l'I-device.
- Adaptez l'accès aux zones de transfert comme dans le projet Contrôleur1. Assurez-vous qu'aucune affectation en double n'a lieu.
- Adaptez les paramètres du sous-réseau et l'interface PROFINET. Puisque l'I-device partagé concerne le même appareil dans différents projets, ces données doivent concorder.
- Sauvegardez le projet.

Les deux projets ont maintenant un I-device partagé configuré de manière identique. L'accès au contrôleur IO et les paramètres de l'interface PROFINET doivent encore être vérifiés dans les différents projets pendant la prochaine étape.

Résumé - Affecter des paramètres pour l'accès à l'appareil partagé

Les zones de transfert sont automatiquement affectées au contrôleur IO local. Pour modifier l'affectation, suivez ces étapes :

1. Cliquez sur l'appareil "S7-1200-I-Device" dans la vue de réseau du projet "Contrôleur1", et sélectionnez la rubrique "Appareil partagé".
2. Un tableau indique quelle CPU a accès à chacune des zones de transfert configurées. Le paramétrage par défaut établit que la CPU locale a accès à tous les modules et sous-modules.
3. Conservez le paramétrage "Contrôleur1" pour toutes les zones de transfert qui doivent rester dans la plage d'adresses de la CPU locale.

Sélectionnez le paramétrage "---" pour toutes les zones de transfert qui doivent être situées dans la plage d'adresses de la CPU du "Contrôleur2" à partir du projet "Contrôleur2". Cela signifie qu'un contrôleur IO en dehors du projet doit avoir accès à la zone de transfert.

4. Suivez la même procédure pour les projets restants.
5. Enfin, vérifiez si les paramétrages pour l'accès sont "complémentaires" pour chaque module ou sous-module dans les deux projets. Cela signifie que si la CPU locale a accès à un projet, l'option "---" doit être paramétrée dans l'autre projet et vice versa.

Remarque : L'option "---" pour l'interface PROFINET et par conséquent pour les ports établit les paramètres associés comme en lecture seule et non modifiables. Les paramètres de l'interface PROFINET et les paramètres des ports peuvent uniquement être modifiés dans le projet dans lequel l'interface PROFINET est affectée à la CPU locale. Les ports peuvent être interconnectés dans les deux projets indépendamment de cela.

6. Vérifiez si les mêmes paramètres d'adresse IP et si le même nom de l'appareil sont paramétrés pour l'appareil partagé dans tous les projets.

Vérifiez si la même ID de sous-réseau S7 est paramétrée dans tous les projets pour le sous-réseau auquel l'appareil partagé est connecté (propriétés de sous-réseau, rubrique "Général" dans la fenêtre d'inspection).

Remarque

Si vous effectuez des modifications de l'I-device (par exemple, modification du nombre ou de la longueur des zones de transfert), exportez l'I-device à nouveau en tant que fichier GSD. Installez à nouveau le fichier GSD dans chaque projet qui utilise l'I-device comme appareil partagé. Assurez-vous qu'un seul contrôleur IO ait accès à une zone de transfert.

Remarque

Lorsque vous utilisez le S7-1200 en tant que I-device partagé et en tant que contrôleur, veillez à augmenter les temps d'actualisation de l'I-device PROFINET et de PROFINET IO afin d'alléger l'impact sur les performances de communication. Le système est très stable et fonctionne bien lorsque vous sélectionnez 2 ms comme temps d'actualisation d'un temps I-device PROFINET individuel et que vous sélectionnez 2 ms comme temps d'actualisation d'un temps PROFINET IO individuel.

Vous indiquez les paramètres "Cycle IO" dans la boîte de dialogue de configuration "Propriétés" de l'I-device PROFINET ou de PROFINET IO. Pour plus d'informations, voir "Configuration du temps de cycle IO" (Page 956).

Procédure - Ajuster les paramètres temps réel

Pour s'assurer que tous les contrôleurs IO et les appareils partagés fonctionnent avec l'horloge de transmission appropriée et que les temps d'actualisation sont calculés correctement sur la base de la charge due à la communication, vous devez ajuster et vérifier les paramètres suivants :

1. Vous devez paramétrer la même horloge de transmission pour chaque contrôleur IO qui a accès aux modules et aux sous-modules de l'appareil partagé :
 - Si vous configurez le contrôleur IO avec STEP 7 (TIA Portal), effectuez ces étapes :
 - Ouvrez le projet correspondant.
 - Sélectionnez l'interface PROFINET du contrôleur IO.
 - Sélectionnez la rubrique "Options avancées > Paramètres temps réel > Communication IO" dans la fenêtre d'inspection et paramétrez l'horloge de transmission partagée.
 - Si vous configurez le contrôleur IO avec un outil d'ingénierie différent, effectuez ces étapes :
 - Sélectionnez l'interface PROFINET de l'appareil partagé dans STEP 7 (Portail TIA) et lisez l'horloge de transmission sur l'appareil partagé (rubrique "Options avancées > Paramètres temps réel").
 - Entrez l'horloge de transmission lue dans l'outil d'ingénierie.

Remarque

Si vous configurez **tous** les contrôleurs IO qui ont accès à l'I-device partagé dans STEP 7 (Portail TIA), vous pouvez paramétrer des horloges de transmission plus courtes sur le contrôleur que celles supportées par l'appareil partagé (adaptation de l'horloge de transmission).

Compilation et chargement

Vous devez compiler les configurations pour les différents contrôleurs IO et les télécharger dans les CPU l'une après l'autre.

En raison de la configuration appliquée à des projets séparés, STEP 7 ne met pas évidence les erreurs de cohérence en cas d'affectation de paramètres d'accès incorrecte. Voici des exemples d'affectation de paramètres d'accès incorrecte :

- Plusieurs contrôleurs IO ont accès au même module.
- Les paramètres d'adresse IP ou les horloges de transmission ne sont pas identiques.

Ces erreurs ne se manifestent pas avant le fonctionnement du contrôleur et sont mises en évidence comme erreurs de configuration.

11.2.15 Protocole MRP (Media Redundancy Protocol)

Les CPU S7-1200 V4.2.x suivantes peuvent fonctionner en tant que client MRP, mais n'assurent pas le rôle de gestionnaire MRP :

- CPU 1215C
- CPU 1217C
- CPU 1215FC

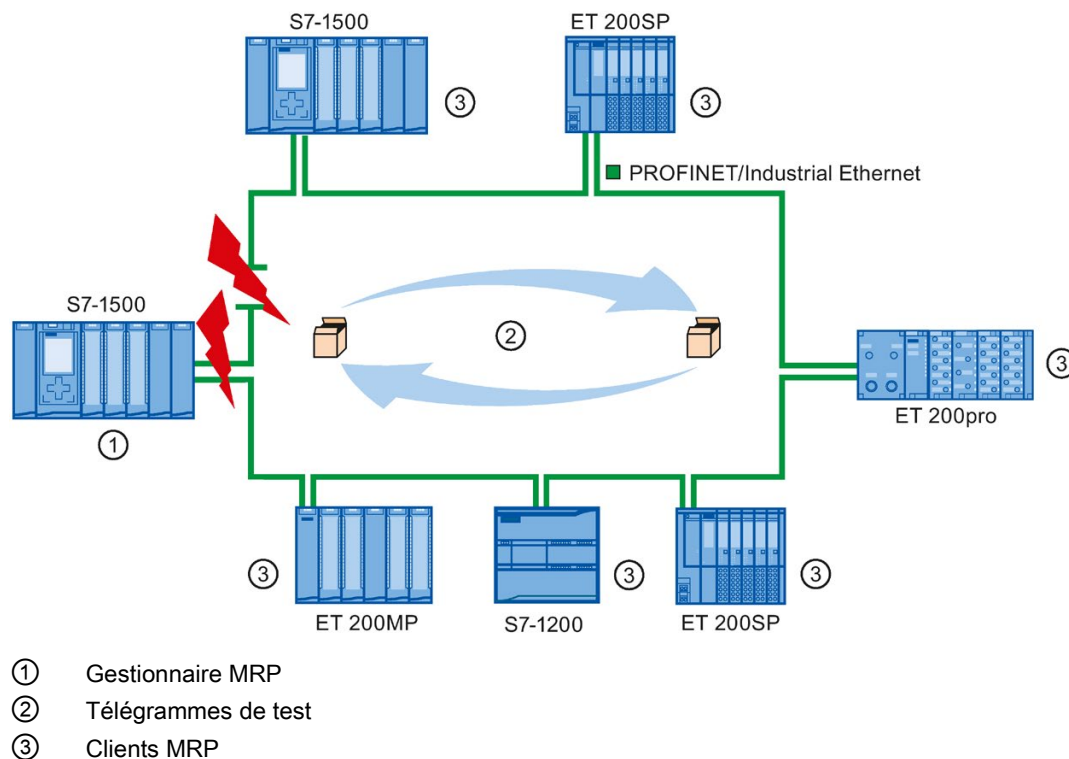
Ces trois CPU S7-1200 ont toutes deux ports PN, ce qui est nécessaire pour la prise en charge du protocole MRP et des paramètres de configuration utilisés pour initialiser le fonctionnement client MRP.

11.2.15.1 Redondance des supports dans les technologies en anneau

Pour accroître la disponibilité d'un réseau Industrial Ethernet à topologies réseau linéaires optiques ou électriques, vous pouvez convertir une topologie réseau linéaire en topologie en anneau en réunissant ses extrémités.

Les appareils dans une topologie en anneau peuvent être des périphériques IO, des contrôleurs IO, des commutateurs externes ou des commutateurs intégrés de modules de communication.

Pour configurer une topologie en anneau avec redondance des supports, vous devez réunir les deux extrémités libres d'une topologie réseau linéaire dans un appareil. Boucler la topologie réseau linéaire pour former un anneau se fait avec deux ports (ports anneau) d'un appareil dans l'anneau. L'un des appareils de l'anneau résultant assure alors le rôle de gestionnaire MRP. Tous les autres appareils dans l'anneau sont des clients MRP.



Les ports anneau d'un appareil sont les ports qui établissent la liaison aux deux appareils voisins dans la topologie en anneau. Vous sélectionnez et paramétrez les ports anneau dans la configuration de l'appareil correspondant (prédéfini le cas échéant).

Fonctionnement de la redondance de supports dans une topologie en anneau

Si l'anneau est interrompu à un point, les chemins de données entre les différents appareils sont reconfigurés automatiquement. Après la reconfiguration, les appareils sont à nouveau accessibles.

En cas de fonctionnement ininterrompu du réseau en communication normale, l'un des deux ports anneau est bloqué dans le gestionnaire MRP afin qu'aucun télégramme de données ne tourne en boucle. Du point de vue de la transmission des données, la topologie en anneau devient linéaire. Le gestionnaire MRP surveille l'anneau à la recherche d'interruptions. Pour ce faire, il envoie des télégrammes de test aussi bien à partir du port anneau 1 qu'à partir du port anneau 2. Les télégrammes de test circulent le long de l'anneau dans les deux sens jusqu'à ce qu'ils atteignent le port anneau opposé du gestionnaire MRP.

Une interruption de l'anneau peut être due à une coupure de la connexion entre deux appareils ou à la défaillance d'un appareil dans l'anneau.

Si les télégrammes de test du gestionnaire MRP n'atteignent plus l'autre port anneau lors d'une interruption de l'anneau, le gestionnaire MRP interconnecte ses deux ports anneau.

Ce chemin de substitution permet de rétablir une liaison fonctionnelle entre tous les autres appareils sous la forme d'une topologie réseau linéaire.

La durée entre l'interruption et le rétablissement de l'anneau d'une topologie linéaire fonctionnelle est appelée temps de reconfiguration.

Dès que l'interruption a été éliminée, les chemins de transmission initiaux sont rétablis, les deux ports anneau du gestionnaire MRP sont déconnectés l'un de l'autre et les clients MRP sont informés de ce changement. Les clients MRP réutilisent alors les chemins initiaux vers les autres appareils.

Procédé de redondance des supports

Le procédé de redondance des supports standard dans SIMATIC est le protocole "Media Redundancy Protocol" (MRP) avec un temps de reconfiguration typique de 200 ms. Jusqu'à 50 appareils peuvent participer à chaque anneau.

11.2.15.2 Utilisation du protocole MRP (Media Redundancy Protocol)

Le procédé "MRP" fonctionne conformément au protocole "Media Redundancy Protocol" (MRP) qui est défini dans la norme IEC 61158 type 10 "PROFINET".

Conditions

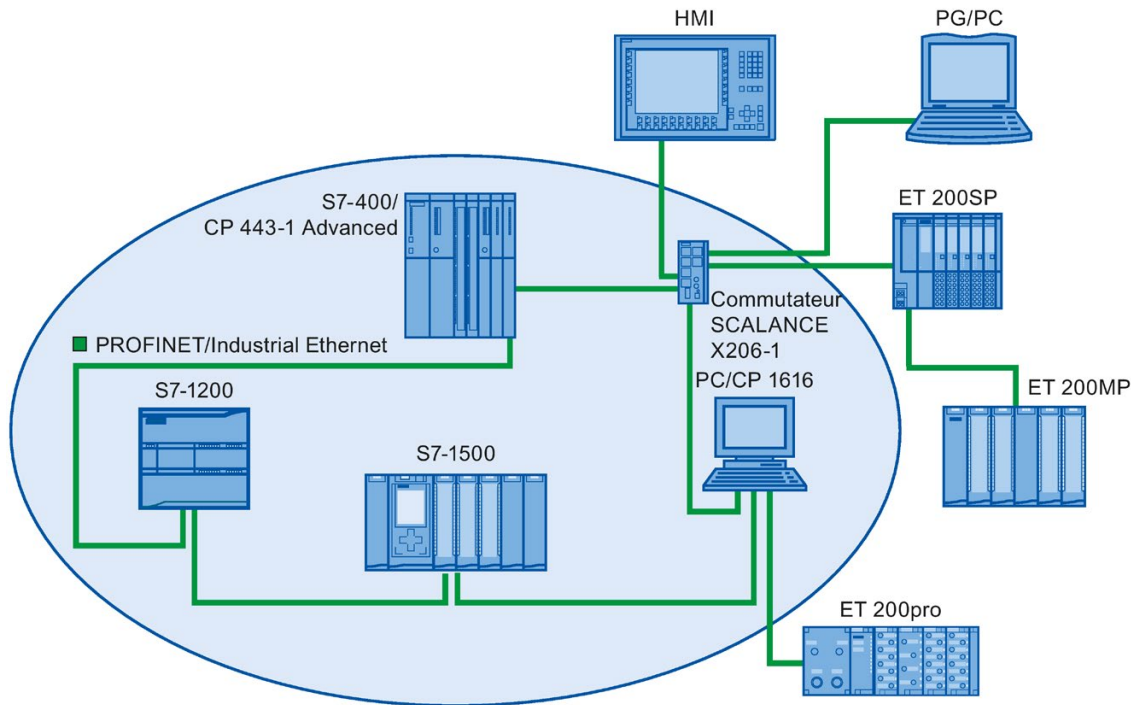
Les conditions suivantes doivent être remplies pour un fonctionnement sans perturbation de MRP :

- L'anneau dans lequel vous voulez utiliser MRP doit être exclusivement constitué d'appareils qui prennent cette fonction en charge.
- MRP doit être activé dans tous les appareils de l'anneau.
- Tous les appareils doivent être interconnectés par leurs ports anneaux.
- Il doit y avoir au moins un gestionnaire MRP (rôle "Gestionnaire (auto)").
- L'anneau ne doit pas contenir plus de 50 appareils. Un dépassement du nombre d'appareils peut entraîner des temps de reconfiguration supérieurs ou égaux à 200 ms.
- Tous les ports partenaires dans l'anneau doivent avoir des paramètres identiques.

Topologie

La figure suivante montre une topologie possible pour les appareils dans un anneau avec MRP. Les appareils dans l'ovale ombré se trouvent dans le domaine de redondance.

Voici un exemple de topologie en anneau avec MRP :



Les règles suivantes s'appliquent pour une topologie en anneau avec redondance des supports selon MRP :

- Tous les appareils dans l'anneau appartiennent au même domaine de redondance.
- Un appareil dans l'anneau a le rôle de gestionnaire MRP.
- Tous les autres appareils dans l'anneau sont des clients MRP.

Vous pouvez relier au réseau des appareils non compatibles MRP par le biais de ports non configurés comme ports anneaux. Cela est uniquement possible avec des appareils ayant plus de deux ports (par exemple, un commutateur SCALANCE X ou un PC avec une CP1616).

Conditions limites

Vous disposez des possibilités de communication suivantes :

- MRP et RT : Le fonctionnement RT est possible avec MRP.

Remarque

La communication RT est interrompue (défaillance de station) si le temps de reconfiguration de l'anneau est supérieur au délai de réponse sélectionné pour le périphérique IO. Vous devez sélectionner un délai de réponse supérieur à 200 ms pour vos périphériques IO. Pour plus d'informations, voir le paragraphe "Délai de réponse" ci-après.

- MRP et TCP/IP (TSEND, HTTP...) : La communication TCP/IP avec MRP est possible, car les paquets de données perdus sont renvoyés le cas échéant.
- MRP et démarrage priorisé :
 - Si vous configurez MRP dans un anneau, vous ne pouvez pas utiliser la fonction "Démarrage priorisé" dans les applications PROFINET sur les appareils concernés.
 - Si vous voulez utiliser la fonction "Démarrage priorisé", vous devez alors désactiver MRP dans la configuration (l'appareil ne peut pas faire partie de l'anneau).
- MRP dans des appareils PROFINET ayant plus de deux ports : Si vous utilisez dans un anneau un appareil PROFINET ayant plus de deux ports, vous devez paramétrer une limite Sync sur les ports qui ne sont pas dans l'anneau. Vous définissez ainsi une limite pour un domaine Sync. Vous ne pouvez pas transmettre de trames Sync émises pour synchroniser des appareils dans un domaine Sync.

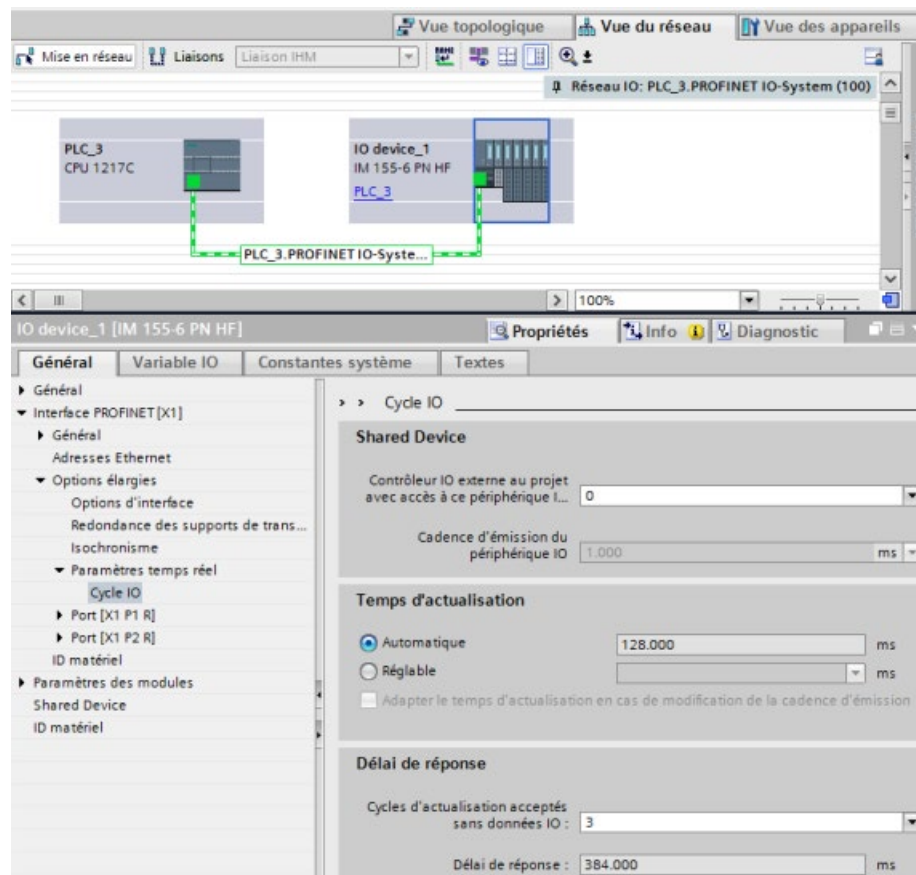
Délai de réponse

Le délai de réponse est l'intervalle de temps pendant lequel un contrôleur IO ou un périphérique IO est autorisé à ne pas recevoir de données d'entrée/sortie (données IO). Si le périphérique IO ne reçoit pas de données du contrôleur IO pendant le délai de réponse, il détecte les trames manquantes et transmet des valeurs de remplacement. Une défaillance de station est alors signalée dans le contrôleur IO.

Vous pouvez configurer un délai de réponse pour les périphériques PROFINET IO. Vous n'indiquez pas le délai de réponse directement, mais sous la forme "Cycles d'actualisation acceptés sans données IO". Le délai de réponse résultant est automatiquement calculé à partir du nombre de cycles d'actualisation acceptés.

Procédez comme suit pour définir le délai de réponse :

1. Sélectionnez l'interface PROFINET du périphérique IO dans la vue du réseau ou dans la vue des appareils.
2. Dans les propriétés de l'interface, sélectionnez "Options élargies" > "Paramètres temps réel" > "Cycle IO".
3. Sélectionnez le nombre de cycles requis dans la liste déroulante.



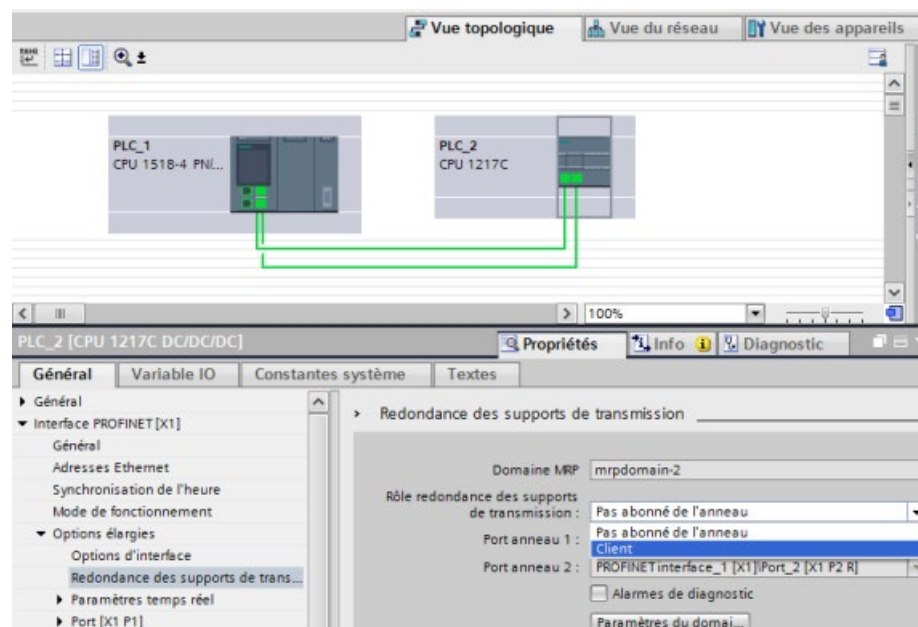
11.2.15.3 Configuration de la redondance des supports

Tous les composants dans votre application doivent prendre en charge le protocole MRP (Media Redundancy Protocol).

Marche à suivre

Procédez comme suit pour configurer la redondance des supports :

1. Créez un anneau par des interconnexions de port appropriées (par exemple, dans la vue topologique).
2. Sélectionnez un périphérique PROFINET pour lequel vous voulez configurer la redondance des supports.
3. Dans la fenêtre d'inspection, naviguez jusqu'à "Interface PROFINET [X1]" > "Options élargies" > "Redondance des supports de transmission".



4. Dans "Rôle redondance des supports de transmission", affectez à l'appareil le rôle "Gestionnaire (auto)", "Client" ou "Pas d'abonné de l'anneau".

Lorsque vous configurez un anneau dans la vue topologique de TIA Portal, ce dernier affecte automatiquement les rôles de redondance des supports pour vous. Si un appareil peut être gestionnaire, TIA Portal définit le rôle de redondance des supports à "Gestionnaire (auto)". Pour le S7-1200, le rôle de redondance des supports est automatiquement défini à "Client".

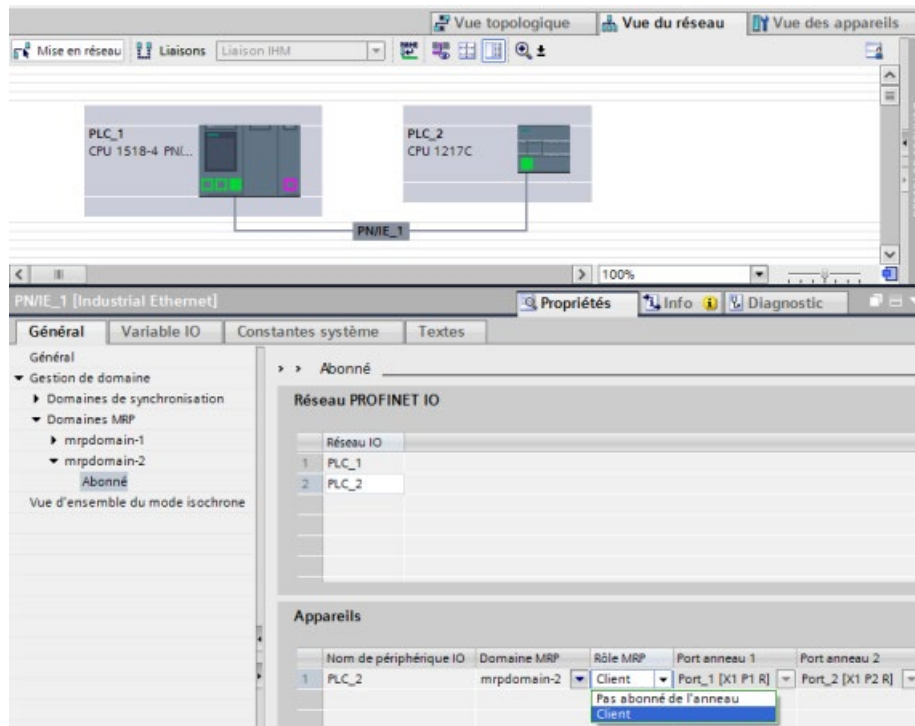
Remarque

Vous ne pouvez pas affecter le rôle de redondance des supports "Gestionnaire (auto)" à la CPU S7-1200.

5. Répétez les étapes 2 à 4 pour tous les appareils PROFINET dans l'anneau.

Ou bien :

1. Mettez en évidence le réseau PROFINET IO dans la vue du réseau.
2. Cliquez sur le réseau PROFINET IO.
3. Naviguez jusqu'à l'appareil du domaine MRP requis dans la fenêtre d'inspection.



4. Pour les périphériques PROFINET, sélectionnez "Gestionnaire (auto)", "Client" ou "Pas d'abonné de l'anneau".

Remarque

Vous ne pouvez pas affecter le rôle MRP "Gestionnaire (auto)" à la CPU S7-1200.

Option de configuration "redondance des supports" : rôle MRP

Les rôles "Gestionnaire", "Gestionnaire (auto)", "Client" et "Pas d'abonné de l'anneau" sont disponibles selon l'appareil utilisé.

Règles :

- Un anneau ne peut avoir qu'un appareil avec le rôle "Gestionnaire". Aucun autre appareil avec le rôle "Gestionnaire" ou "Gestionnaire (auto)" n'est autorisé. Tous les autres appareils dans l'anneau peuvent uniquement avoir le rôle "Client". Les appareils qui ne font pas partie de l'anneau peuvent avoir le rôle "Pas d'abonné de l'anneau".
- Si un anneau n'a pas d'appareil avec le rôle "Gestionnaire", il doit avoir au moins un appareil avec le rôle "Gestionnaire (auto)". Un anneau peut avoir un nombre quelconque d'appareils assurant les rôles "Client" et "Gestionnaire (auto)".

Remarque

Vous ne pouvez pas affecter les rôles MRP "Gestionnaire" ou "Gestionnaire (auto)" à la CPU S7-1200.

Option de configuration "redondance des supports" : port anneau 1 et port anneau 2

Sélectionnez l'un après l'autre les ports que vous voulez configurer comme port anneau 1 ou port anneau 2. La liste déroulante affiche les ports possibles pour chaque type d'appareil. Si les ports sont réglés en usine, les champs ne sont pas disponibles.

Remarque

La configuration des ports anneau n'est pas nécessaire dans le S7-1200 puisque la CPU S7-1200 n'a que deux ports.

Alarmes de diagnostic

Si des alarmes de diagnostic concernant l'état MRP doivent être transmises à la CPU locale, cochez la case "Alarmes de diagnostic". Les alarmes de diagnostic suivantes peuvent être configurées :

- Erreur de câblage ou de port :

La CPU génère des alarmes de diagnostic pour les erreurs suivantes sur les ports anneau :

- Un voisin du port anneau ne prend pas en charge MRP.
- Un port anneau est relié à un port qui n'est pas un port anneau.
- Un port anneau est relié au port anneau d'un autre domaine MRP.

- Interruption/rétablissement (gestionnaire MRP uniquement) :

La CPU génère des alarmes de diagnostic en cas d'interruption de l'anneau et de rétablissement de la configuration initiale. Si ces deux alarmes se produisent à moins de 0,2 seconde d'intervalle, cela signale une interruption de l'anneau.

Vous pouvez réagir à ces événements dans le programme utilisateur en programmant la réaction appropriée dans l'OB d'alarme de diagnostic (OB 82).

Remarque**Appareils tiers comme gestionnaire MRP**

Pour assurer un fonctionnement sans erreur lorsqu'un appareil tiers est utilisé comme gestionnaire MRP dans un anneau, vous devez affecter le rôle fixe "Client" à tous les autres appareils de l'anneau avant de fermer l'anneau. Il est possible sinon que des télégrammes tournent en boucle ou que des défaillances du réseau se produisent.

11.2.16 Routage S7

Dans la vue du réseau de STEP 7, vous pouvez créer une topologie de communication complexe en reliant des appareils dans différents sous-réseaux S7. Vous pouvez relier des CPU et CP classiques ainsi que les CPU et CP S7 les plus récents. Vous pouvez également inclure des IHM et des stations PC, un serveur OPC par exemple.

Une fois que vous avez décidé des appareils qui doivent communiquer et établi les liaisons nécessaires à l'aide de STEP 7, le système d'ingénierie (ES) peut charger les tables de routage correspondantes dans les différents routeurs S7 comme partie de la configuration matérielle. Lorsque vous avez chargé les tables de routage dans les différents routeurs S7, le système d'ingénierie ou d'autres partenaires de communication peuvent communiquer avec chaque appareil, et ce même si ces appareils se trouvent dans des sous-réseaux S7 différents. Cela est possible parce que les CPU et/ou CP intercalés agissent en tant que routeurs S7. Les CPU et/ou CP transmettent les demandes de liaison entrantes au routeur S7 suivant jusqu'à ce que la demande de liaison atteigne l'appareil concerné. Les appareils établissent alors la liaison S7.

La CPU utilise le mécanisme d'écriture d'enregistrement pour transférer les tables de routage nécessaires aux processeurs CP dans l'appareil de base. Les tables de routage établissent le chemin d'un appareil à un autre au moment de la demande de liaison qui comprend une identification de sous-réseau S7 distant (Subnet_ID). L'appareil qui reçoit la demande de liaison interroge sa table de routage, trouve la station suivante dans le chemin vers le sous-réseau S7 cible et transmet la demande de liaison. Finalement, la demande de liaison atteint la cible voulue et la réponse retrace le chemin en sens inverse.

Les CPU S7-1200 ont une interface PN unique et jusqu'à trois CP raccordés au bus de communication local. Vous disposez donc de deux options pour le routage dans la station S7-1200 :

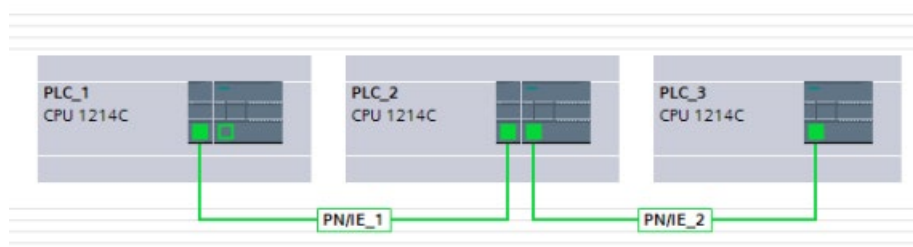
- Routage entre la CPU et un CP
- Routage d'un CP à un autre CP

Pour plus d'informations sur les CP S7-1200 qui prennent en charge la fonction de routage S7, voir les CP S7-1200 sur la page Support produit du site Industry Online Support de Siemens. Le CP 1243-1 (<https://support.industry.siemens.com/cs/fr/fr/view/584459>) est montré dans l'exemple de recherche de processeur de communication apte au routage S7.

11.2.16.1 Routage S7 entre interfaces CPU et CP

Comme les CPU S7-1200 sont limitées à une interface PN unique, une CPU individuelle ne peut pas servir de routeur. Vous ne pouvez jamais raccorder une CPU individuelle à plus d'un sous-réseau S7 à la fois. Lorsque vous installez des modules CP dans l'appareil de base de la CPU, vous pouvez vous relier à plusieurs sous-réseaux S7 et utiliser le routage.

Pour que PLC_1 puisse communiquer avec PLC_3 dans l'exemple de système ci-dessous, le système d'ingénierie doit acheminer les messages via PLC_2. Le système d'ingénierie doit charger la table de routage pour PLC_2 et PLC_2 doit fournir la table de routage pour le module CP dans son appareil de base. Avec ces tables de routage en place, PLC_1 et PLC_3 peuvent communiquer l'un avec l'autre même s'ils ne sont pas directement reliés.

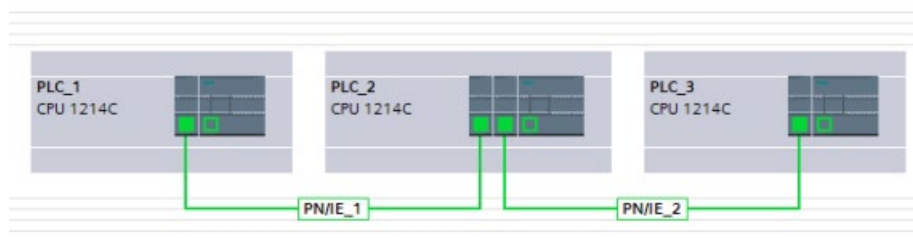


Pour vérifier le routage de l'un quelconque des sous-réseaux S7 vers l'autre sous-réseau S7, PLC_1 doit établir une connexion de transport vers PLC_3 et PLC_3 doit établir une liaison vers PLC_1. Cette procédure assure que le routage de l'interface PN/IE de l'API vers un module CP et le routage d'un module CP vers l'interface PN/IE de l'API sont possibles.

11.2.16.2 Routage S7 entre deux interfaces CP

Comme les CPU S7-1200 prennent en charge jusqu'à trois modules CP, vous pouvez relier la totalité de ces modules à des sous-réseaux S7 différents. Vous pouvez utiliser le routage si vous installez au moins deux modules CP dans l'appareil de base de la CPU et les reliez à différents sous-réseaux S7.

Pour que PLC_1 puisse communiquer avec PLC_3 dans l'exemple de système ci-dessous, le système d'ingénierie doit acheminer les messages par PLC_2 du module CP au module CP dans son appareil de base. Le système d'ingénierie doit charger la table de routage pour PLC_2 et PLC_2 doit fournir les tables de routage pour les deux modules CP. Avec ces tables de routage en place, PLC_1 et PLC_3 peuvent communiquer l'un avec l'autre même s'ils ne sont pas directement reliés. Notez également que le routage se fait de module CP à module CP sans messages envoyés par le biais de l'interface PN/IE de PLC_2.



11.2.17 Désactivation de SNMP

Le protocole SNMP (Simple Network Management Protocol) est un protocole de norme Internet permettant de collecter et d'organiser des informations sur les appareils gérés dans les réseaux IP et de modifier ces informations afin de changer le comportement de l'appareil. Les appareils prenant généralement en charge SNMP sont les routeurs, commutateurs, serveurs, postes de travail, imprimantes, châssis de modem, etc.

SNMP est largement utilisé dans les systèmes de gestion de réseau pour surveiller, dans les appareils reliés au réseau, les situations justifiant une attention administrative. SNMP utilise divers services et outils pour la détection et le diagnostic de la topologie du réseau. Les informations sur les propriétés des appareils compatibles SNMP sont enregistrées dans des fichiers MIB (Management Information Base) pour lesquels l'utilisateur doit posséder les droits requis. SNMP présente les données de gestion sous la forme de variables sur les systèmes gérés qui décrivent la configuration système. Ces variables peuvent être interrogées (et parfois paramétrées) par des applications de gestion.

SNMP utilise le protocole de transport UDP et comprend les deux composants réseau suivants :

- Gestionnaire SNMP : Surveille les noeuds du réseau.
- Client SNMP : Collecte diverses informations spécifiques du réseau dans les noeuds individuels et les stocke sous forme structurée dans la base MIB (Management Information Base). Des diagnostics réseau détaillés peuvent être effectués avec ces données.

Dans certaines conditions, votre application peut vous demander de désactiver SNMP, par exemple :

- si les paramétrages de sécurité dans votre réseau n'autorisent pas l'utilisation de SNMP ;
- si vous utilisez votre propre solution SNMP (par exemple, avec vos propres instructions de communication).

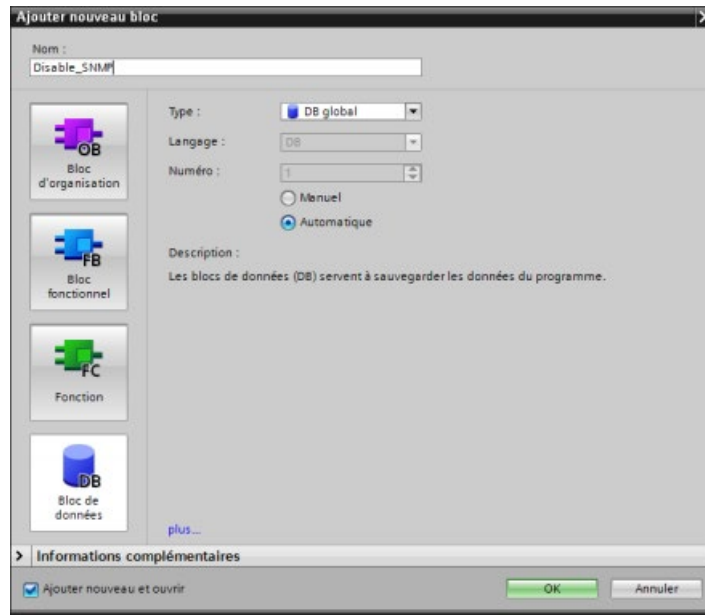
Si vous désactivez SNMP pour un appareil, vous ne disposerez plus de certaines options pour le diagnostic de la topologie du réseau (par exemple, l'outil PRONETA ou le serveur Web de la CPU).

11.2.17.1 Désactivation de SNMP

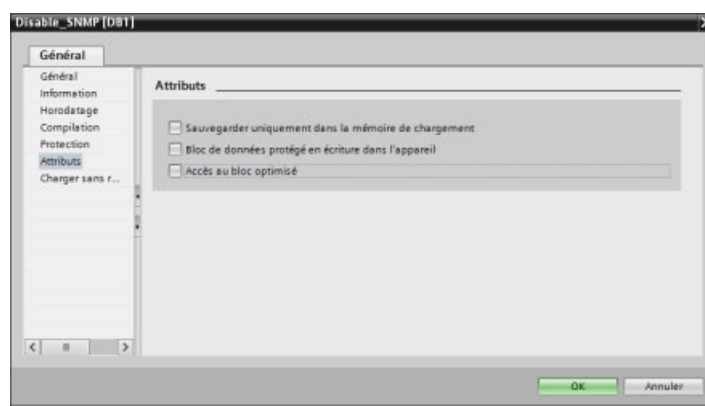
Désactivation de SNMP

Procédez comme suit pour désactiver SNMP dans la CPU S7-1200 :

1. Créez un bloc de données (DB) classique :



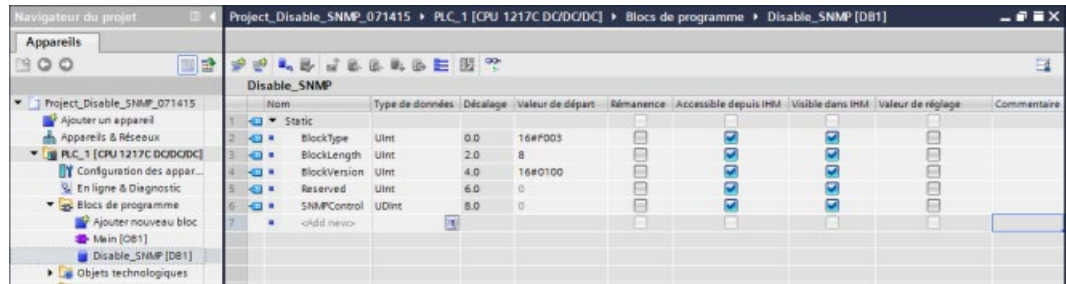
2. Sélectionnez les propriétés du DB que vous venez de créer.
3. Sélectionnez l'onglet "Attributs". Décochez la case "Accès au bloc optimisé" :



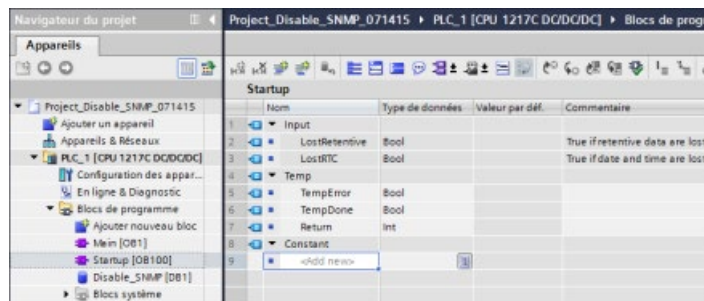
4. Cliquez sur le bouton "OK".

Un message vous recommandant de recompiler votre programme s'affiche. Recompilez votre programme à ce moment.

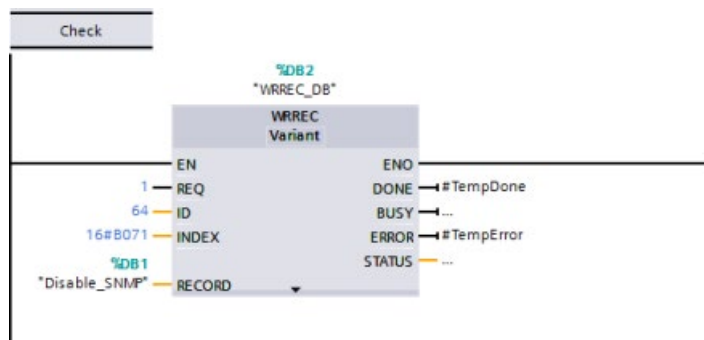
- Dans l'interface du bloc DB classique, créez les variables statiques suivantes avec les valeurs indiquées. Vous utiliserez ces variables dans votre programme pour désactiver l'implémentation SNMP interne :



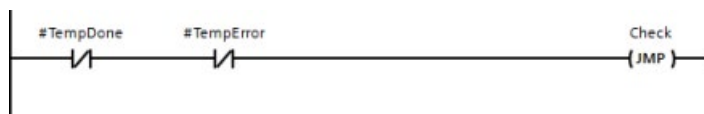
- Dans l'OB de démarrage (OB 100), ajoutez les variables temporaires comme indiqué :



- Avec l'éditeur CONT dans le réseau 1 de l'OB de démarrage (OB 100), insérez un repère (Repère de saut) – dans l'exemple ci-dessous, le repère est nommé "Check" – et une instruction WRREC (Écrire l'enregistrement) avec les entrées et les sorties indiquées :



- Insérez la boucle et le code de contrôle suivants avec la sortie JMP (Sauter au repère). Ce code garantit l'achèvement de l'appel et la désactivation de SNMP avant de quitter l'OB de démarrage :



11.2.18 Diagnostic

"Blocs d'organisation (OB)" (Page 94) explique comment utiliser des blocs d'organisation (OB) à des fins de diagnostic avec ces réseaux de communication.

11.2.19 Instructions de périphérie décentralisée

Reportez-vous à "Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)" (Page 394) pour obtenir des informations sur la manière d'utiliser les instructions de périphérie décentralisée avec ces réseaux de communication.

11.2.20 Instructions de diagnostic

Reportez-vous à "Diagnostic (PROFINET ou PROFIBUS)", "Instructions de diagnostic" (Page 449) pour obtenir des informations sur la manière d'utiliser ces instructions avec ces réseaux de communication.

11.2.21 Événements de diagnostic pour la périphérie décentralisée

Reportez-vous à "Diagnostic (PROFINET ou PROFIBUS)", "Événements de diagnostic pour la périphérie décentralisée" (Page 498) pour obtenir des informations sur la manière d'utiliser ces informations de diagnostic avec ces réseaux de communication.

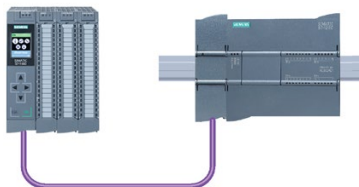
11.3 PROFIBUS

Un système PROFIBUS utilise un maître du bus pour interroger des esclaves répartis en mode multipoint sur un bus série RS485. Un esclave PROFIBUS est n'importe quel périphérique (transducteur d'E/S, vanne, entraînement de moteur ou autre appareil de mesure) qui traite des informations et envoie ses sorties au maître. L'esclave constitue une station passive sur le réseau puisqu'il n'a pas de droits d'accès au bus et peut seulement accuser réception des messages reçus ou envoyer des messages sur demande du maître. Tous les esclaves PROFIBUS ont la même priorité et le maître est à l'origine de toutes les communications de réseau

Un maître PROFIBUS constitue une station active sur le réseau. PROFIBUS DP définit deux classes de maîtres. Un maître de classe 1 (normalement un automate programmable industriel API ou un PC exécutant un logiciel spécial) gère la communication normale ou l'échange de données avec les esclaves qui lui sont affectés. Un maître de classe 2 (généralement un appareil de configuration, tel qu'un ordinateur portable ou une console de programmation servant à la mise en service, à la maintenance ou au diagnostic) est un appareil spécial utilisé avant tout pour la mise en service des esclaves et à des fins de diagnostic.

Le S7-1200 est raccordé à un réseau PROFIBUS en tant qu'esclave DP avec le module de communication CM 1242-5. Le module CM 1242-5 (esclave DP) peut être le partenaire de communication de maîtres DP V0/V1. Si vous voulez configurer le module dans un système tiers, il y a un fichier GSD disponible pour le CM 1242-5 (esclave DP) sur le CD qui est envoyé avec le module et sur les pages de l'Assistance client Siemens Automatisation (<https://support.industry.siemens.com/cs/ww/fr/ps/6GK7242-5DX30-0XE0>) sur Internet.

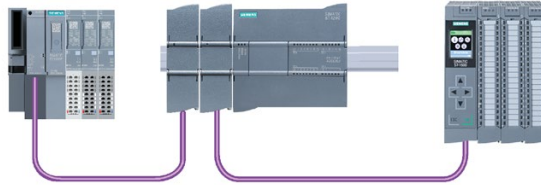
Dans la figure ci-dessous, le S7-1200 est un esclave DP affecté à un automate S7-300 :



Le S7-1200 est raccordé à un réseau PROFIBUS en tant que maître DP avec le module de communication CM 1243-5. Le module CM 1243-5 (maître DP) peut être le partenaire de communication d'esclaves DP V0/V1. Dans la figure ci-dessous, le S7-1200 est un maître pilotant un esclave DP ET 200SP :



Si l'on a installé à la fois un CM 1242-5 et un CM 1243-5, un S7-1200 peut opérer simultanément comme esclave d'un réseau maître DP de niveau supérieur et comme maître d'un réseau maître DP de niveau inférieur :



À partir de V3.0, vous pouvez configurer trois modules de communication PROFIBUS au maximum par station, toute combinaison de CM maîtres DP ou esclaves DP étant possible. Dans une implémentation du firmware V3.0 ou ultérieur de la CPU, les maîtres DP peuvent piloter chacun 32 esclaves au maximum.

Les données de configuration des CM PROFIBUS sont stockées sur la CPU locale. Cela permet un simple remplacement de ces modules de communication si nécessaire.

Pour utiliser PROFIBUS avec les CPU S7-1200 V4.0 ou ultérieures, vous devez mettre à niveau le firmware du CM PROFIBUS maître au moins à la version V1.3.

Remarque

Nous vous conseillons de toujours mettre le firmware du CM PROFIBUS à niveau sur la version actuelle (<http://support.automation.siemens.com/WW/view/fr/42131407>). Vous pouvez mettre à jour le firmware au moyen de l'une des méthodes suivantes :

- Utilisation des Outils en ligne et de diagnostic de STEP 7 (Page 1389)
 - Utilisation d'une Carte mémoire SIMATIC (Page 153)
 - Utilisation de la page Web standard "Information sur les modules" du serveur Web (Page 1060)
 - Utilisation de SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/fr/view/98161300>)
-

11.3.1 Services de communication des CM PROFIBUS

Les CM PROFIBUS utilisent le protocole PROFIBUS DP-V1.

Types de communication avec DP-V1

Les types de communication suivants sont disponibles avec DP-V1 :

- Communication cyclique (CM 1242-5 et CM 1243-5)

Les deux modules PROFIBUS prennent en charge la communication cyclique pour le transfert de données du processus entre esclave DP et maître DP.

La communication cyclique est gérée par le système d'exploitation de la CPU. Elle ne nécessite pas de blocs logiciels. Les données d'E/S sont lues ou écrites directement dans la mémoire image de la CPU.

- Communication acyclique (CM 1243-5 uniquement)

Le module maître DP prend également en charge la communication acyclique à l'aide de blocs logiciels :

- L'instruction "RALRM" est disponible pour la gestion des alarmes.
- Les instructions "RDREC" et "WRREC" permettent de transférer les données de configuration et de diagnostic.

Fonctions non prises en charge par le CM 1243-5 : SYNC/FREEZE et Get_Master_Diag

Autres services de communication du CM 1243-5

Le module maître CM 1243-5 DP prend en charge les services de communication supplémentaires suivants :

- Communication S7

- Services PUT/GET

Le maître DP fonctionne en tant que client et serveur pour les requêtes provenant d'autres automates S7 ou de PC via PROFIBUS.

- Communication PG/OP

Les fonctions PG permettent de charger les données de configuration et les programmes utilisateur d'une console de programmation dans la CPU et de transférer les données de diagnostic dans une console de programmation.

Les partenaires possibles pour la communication OP sont les pupitres HMI Panels, les SIMATIC Panel PC avec WinCC flexible ou les systèmes SCADA qui supportent la communication S7.

11.3.2 Référence aux manuels utilisateurs des CM PROFIBUS

Informations supplémentaires

Vous trouverez des informations détaillées sur les modules de communication PROFIBUS dans les manuels de ces appareils. Ces manuels sont disponibles sur Internet, dans les pages du service client de Siemens Industrial Automation sous les entrées suivantes :

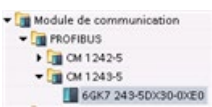
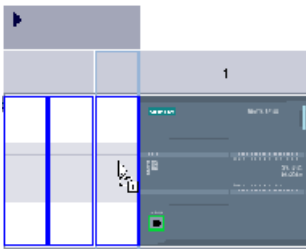
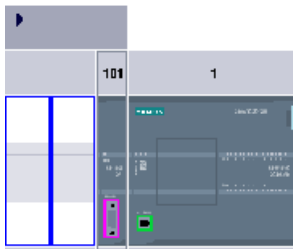
- CM 1242-5 (<https://support.industry.siemens.com/cs/ww/fr/ps/15667>)
- CM 1243-5 (<https://support.industry.siemens.com/cs/ww/fr/ps/15669>)

11.3.3 Configuration d'un maître et d'un esclave DP

11.3.3.1 Ajout du module CM 1243-5 (maître DP) et d'un esclave DP

Dans le portail "Appareils & Réseaux", utilisez le catalogue du matériel pour ajouter des modules PROFIBUS à la CPU. Ces modules se raccordent sur le côté gauche de la CPU. Pour insérer un module dans la configuration matérielle, sélectionnez le module concerné dans le catalogue du matériel et double-cliquez ou faites glisser le module dans l'emplacement mis en évidence.

Tableau 11- 54 Ajout d'un module CM 1243-5 PROFIBUS (maître DP) à la configuration des appareils

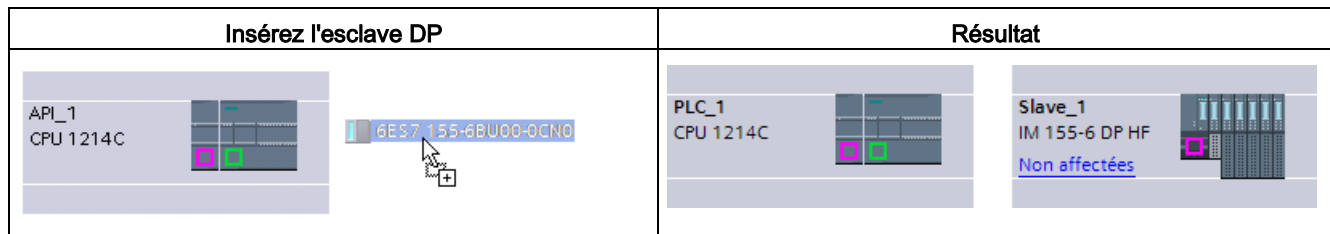
Module	Sélectionnez le module	Insérez le module	Résultat
CM 1243-5 (maître DP)			

Utilisez également le catalogue du matériel pour ajouter des esclaves DP. Par exemple, pour ajouter un esclave DP ET 200SP, affichez le détail des conteneurs suivants dans le Catalogue du matériel :

- Périphérie décentralisée
- ET 200SP
- Coupleurs
- PROFIBUS

Puis, sélectionnez "6ES7 155-6BU00-0CN0" (IM155-6 DP HF) dans la liste des numéros de référence et ajoutez l'esclave DP ET 200SP comme illustré dans la figure ci-dessous.

Tableau 11- 55 Ajout d'un esclave DP ET 200SP à la configuration des appareils



11.3.3.2 Configuration des liaisons réseau logiques entre deux appareils PROFIBUS

Maintenant que vous avez configuré le module CM 1243-5 (maître DP), vous êtes prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Pour créer la liaison PROFIBUS, sélectionnez le carré violet (PROFIBUS) sur le premier appareil. Tracez une ligne vers le carré PROFIBUS sur le deuxième appareil. Relâchez le bouton de la souris : la liaison PROFIBUS est créée.

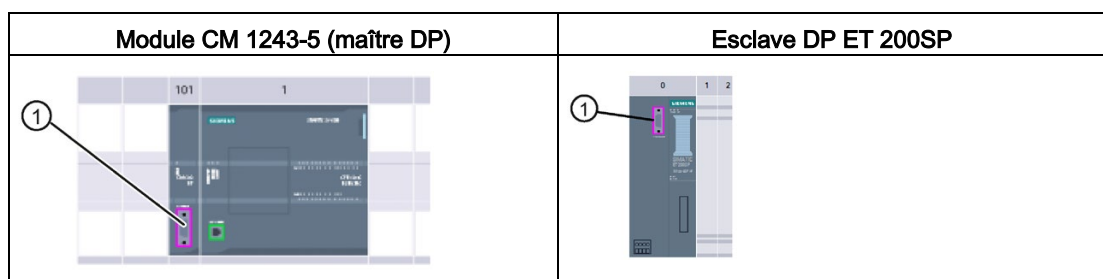
Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 832) pour plus d'informations.

11.3.3.3 Affectation d'adresses PROFIBUS au module CM 1243-5 et à l'esclave DP

Configuration de l'interface PROFIBUS

Une fois que vous avez configuré les liaisons réseau logiques entre deux appareils PROFIBUS, vous pouvez configurer les paramètres des interfaces PROFIBUS. Pour ce faire, cliquez sur le carré PROFIBUS violet sur le module CM 1243-5. L'onglet "Propriétés" de la fenêtre d'inspection affiche l'interface PROFIBUS. L'interface PROFIBUS de l'esclave DP se configure de la même manière.

Tableau 11- 56 Configuration des interfaces PROFIBUS du module CM 1243-5 (maître DP) et de l'esclave DP ET 200SP



① port PROFIBUS

Affectation de l'adresse PROFIBUS

Dans un réseau PROFIBUS, une adresse PROFIBUS est affectée à chaque appareil. Cette adresse est comprise entre 0 et 127, aux exceptions suivantes près :

- Adresse 0 : Réservée aux outils de configuration et/ou de programmation du réseau raccordés au bus
- Adresse 1 : Réservée au premier maître par Siemens
- Adresse 126 : Réservée aux appareils sortant de l'usine qui n'ont pas de réglage par commutateur et doivent être réadressés via le réseau
- Adresse 127 : Réservée aux messages à diffusion générale vers tous les appareils dans le réseau et ne pouvant pas être affectée à des appareils opérationnels.

Ainsi, les adresses pouvant être utilisées par des appareils PROFIBUS opérationnels vont de 2 à 125.

Dans la fenêtre Propriétés, sélectionnez l'entrée de configuration "Adresse PROFIBUS". STEP 7 affiche la boîte de dialogue de configuration de l'adresse PROFIBUS qui sert à définir l'adresse PROFIBUS de l'appareil.

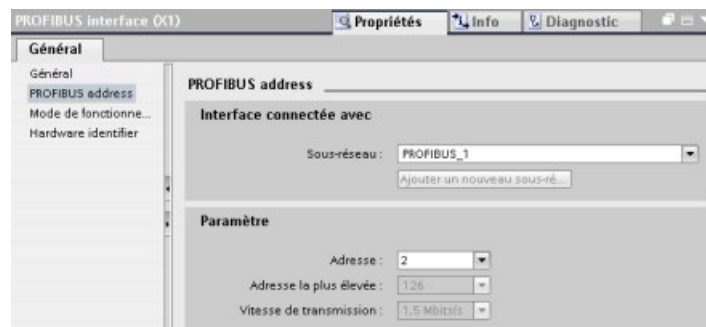


Tableau 11- 57 Paramètres pour l'adresse PROFIBUS

Paramètre	Description	
Sous-réseau	Nom du sous-réseau auquel l'appareil est connecté. Cliquez sur le bouton "Ajouter nouveau sous-réseau" pour créer un nouveau sous-réseau. "non connecté" est la valeur par défaut. Deux types de connexion sont possibles : <ul style="list-style-type: none"> • La valeur par défaut "non connecté" fournit une liaison locale. • Un sous-réseau est nécessaire lorsque votre réseau contient au moins deux appareils. 	
Paramètres	Adresse	Adresse PROFIBUS affectée à l'appareil
	Adresse la plus élevée	L'adresse PROFIBUS la plus élevée est basée sur les stations actives dans le réseau PROFIBUS (par exemple, maître DP). Les esclaves DP passifs ont, indépendamment de cela, des adresses PROFIBUS allant de 1 à 125 même si, par exemple, l'adresse PROFIBUS la plus élevée est définie à 15. L'adresse PROFIBUS la plus élevée est pertinente pour le passage du jeton (c'est-à-dire des droits d'émission), le jeton étant uniquement transmis à des stations actives. Indiquer l'adresse PROFIBUS la plus élevée permet d'optimiser le bus.
	Vitesse de transmission	Vitesse de transmission du réseau PROFIBUS configuré : Les vitesses de transmission PROFIBUS vont de 9,6 Kbits/s à 12 Mbits/s. Le réglage de la vitesse de transmission dépend des propriétés des noeuds PROFIBUS utilisés. La vitesse de transmission ne doit pas dépasser la vitesse prise en charge par le noeud le plus lent. La vitesse de transmission est normalement définie pour le maître sur le réseau PROFIBUS, tous les esclaves DP utilisant automatiquement la même vitesse de transmission (détection automatique de débit).

11.3.4 Instructions de périphérie décentralisée

Reportez-vous à "Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)" (Page 394) pour obtenir des informations sur la manière d'utiliser les instructions de périphérie décentralisée avec ces réseaux de communication.

11.3.5 Instructions de diagnostic

Reportez-vous à "Diagnostic (PROFINET ou PROFIBUS)", "Instructions de diagnostic" (Page 449) pour obtenir des informations sur la manière d'utiliser ces instructions avec ces réseaux de communication.

11.3.6 Evénements de diagnostic pour périphérie décentralisée

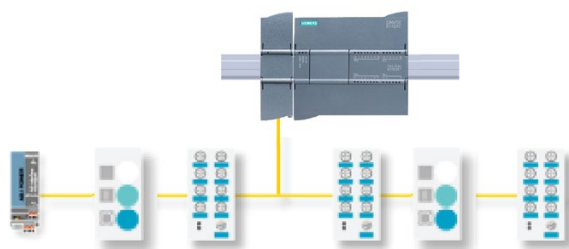
Reportez-vous à "Diagnostic (PROFINET ou PROFIBUS)", "Evénements de diagnostic pour la périphérie décentralisée" (Page 498) pour obtenir des informations sur la manière d'utiliser ces informations de diagnostic avec ces réseaux de communication.

11.4 Interface AS-i

Le module CM 1243-2 maître AS-i S7-1200 permet le raccordement d'un réseau AS-i à une CPU S7-1200.

L'interface actionneur/capteur, interface AS-i, est un réseau d'interconnexion à un seul maître pour le niveau le plus bas dans les systèmes d'automatisation. Le module CM 1243-2 sert de maître AS-i pour le réseau. Avec un câble AS-i unique, il est possible de connecter les capteurs et actionneurs (esclaves AS-i) à la CPU par le biais du CM 1243-2. Le CM 1243-2 gère toute la coordination du réseau AS-i et transmet les données et informations d'état des actionneurs et capteurs à la CPU par le biais des adresses d'E/S qui lui sont affectées. Vous pouvez accéder aux valeurs binaires ou analogiques en fonction du type d'esclave. Les esclaves AS-i sont les voies d'entrée et de sortie du réseau AS-i et ne sont actifs que si le CM 1243-2 les appelle.

Dans la figure ci-dessous, le S7-1200 est un maître AS-i qui pilote des modules esclaves TOR/analogiques AS-i.



Pour utiliser AS-i avec les CPU S7-1200 V4.0, vous devez mettre à niveau le firmware du CM AS-i maître à la version V1.1.

Vous pouvez effectuer cette mise à niveau à l'aide du serveur Web ou d'une carte mémoire SIMATIC.

Remarque

Pour les CPU S7-1200 V4.0, si le serveur web ou une carte mémoire SIMATIC est utilisé pour mettre à niveau le firmware AS-i de V1.0 à V1.1, vous devez mettre à jour le firmware AS-i dans le maître AS-i CM 1243-2 conformément à la procédure suivante :

1. Télécharger la mise à niveau du firmware sur le maître AS-i CM 1243-2.
 2. Une fois le téléchargement terminé, initialiser la CPU S7-1200 pour effectuer le processus de mise à niveau du firmware dans le maître AS-i CM 1243-2.
 3. Répéter les étapes 1 et 2 pour chaque autre CM 1243-2 maître AS-i. L'automate S7-1200 autorise trois CM 1243-2 maître AS-i au maximum.
-

Remarque

Nous vous conseillons de toujours mettre le firmware du CM AS-i à niveau sur la version actuelle disponible (<http://support.automation.siemens.com/WW/view/fr/43416171>) sur le site Web de service et d'assistance Siemens.

11.4.1 Configuration d'un maître et d'un esclave AS-i




Le maître AS-i CM 1243-2 est intégré dans le système d'automatisation S7-1200 en tant que module de communication.

Vous trouverez des informations détaillées sur le module maître AS-i CM 1243-2 dans le manuel "Maître AS-i CM 1243-2 et module AS-i de découplage des données DCM 1271 pour SIMATIC S7-1200" (<https://support.industry.siemens.com/cs/ww/fr/ps/15750/man>).

11.4.1.1 Ajout du module maître AS-i CM 1243-2 et d'un esclave AS-i

Utilisez le catalogue du matériel pour ajouter des modules maître AS-i CM 1243-2 à la CPU. Ces modules se raccordent sur le côté gauche de la CPU, trois modules maître AS-i CM 1243-2 pouvant être utilisés au maximum. Pour insérer un module dans la configuration matérielle, sélectionnez le module concerné dans le catalogue du matériel et double-cliquez ou faites glisser le module dans l'emplacement mis en évidence.

Tableau 11- 58 Ajout d'un module maître AS-i CM 1243-2 à la configuration des appareils

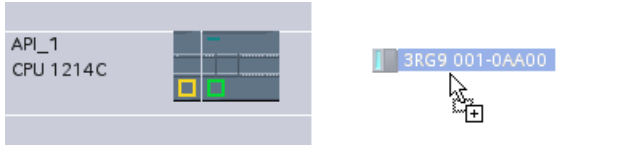
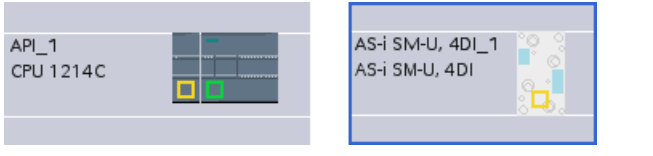
Module	Sélectionnez le module	Insérez le module	Résultat
CM 1243-2 maître AS-i			

Utilisez également le catalogue du matériel pour ajouter des esclaves AS-i. Par exemple, pour ajouter un esclave qui soit "module d'E/S, compact, TOR, entrées", affichez le détail des dossiers suivants dans le Catalogue du matériel :

- Appareils de terrain
- Esclaves AS Interface

Puis, sélectionnez "3RG9 001-0AA00" (AS-i SM-U, 4DI) dans la liste des numéros de référence et ajoutez l'esclave "module d'E/S, compact, TOR, entrées" comme illustré dans la figure ci-après.

Tableau 11- 59 Ajout d'un esclave AS-i à la configuration des appareils

Insérez l'esclave AS-i	Résultat
	

11.4.1.2 Configuration des liaisons réseau logiques entre deux appareils AS-i

Maintenant que vous avez configuré le module maître AS-i CM 1243-2, vous êtes prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Pour créer la liaison AS-i, sélectionnez le carré jaune (AS-i) sur le premier appareil. Tracez une ligne vers le carré AS-i sur le deuxième appareil. Relâchez le bouton de la souris : la liaison AS-i est créée.

Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 832) pour plus d'informations.

11.4.1.3 Configuration des propriétés du maître AS-i CM 1243-2

Pour paramétrer l'interface AS-i, cliquez sur le carré AS-i jaune sur le module maître AS-i CM 1243-2. L'onglet "Propriétés" de la fenêtre d'inspection affiche l'interface AS-i.

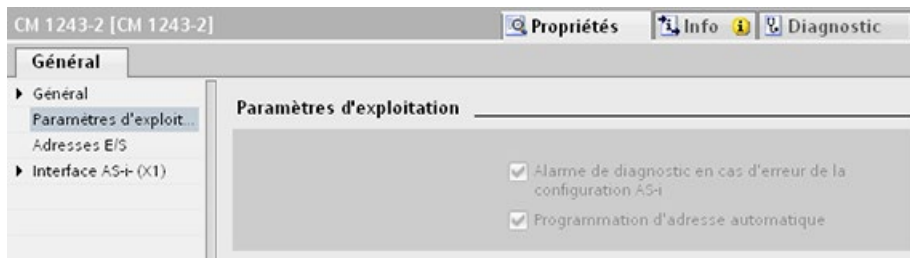
Dans la fenêtre d'inspection STEP 7, vous pouvez afficher, configurer et modifier les informations générales, les adresses et les paramètres de fonctionnement :

Tableau 11- 60 Propriétés du module maître AS-i CM 1243-2

Propriété	Description
Général	Nom du module maître AS-i CM 1243-2
Paramètres de fonctionnement	Paramètres pour la réponse du maître AS-i
Adresses E/S	Zone pour les adresses d'E/S des esclaves
Interface AS-i (X1)	Réseau AS-i affecté

Remarque

"Alarme de diagnostic pour erreurs dans la configuration AS-i" et "Programmation d'adresses automatique" sont toujours actifs et sont donc affichés en gris.



11.4.1.4 Affectation d'une adresse AS-i à un esclave AS-i

Configuration de l'interface esclave AS-i

Pour paramétrer l'interface AS-i, cliquez sur le carré AS-i jaune sur l'esclave AS-i. L'onglet "Propriétés" de la fenêtre d'inspection affiche l'interface AS-i.



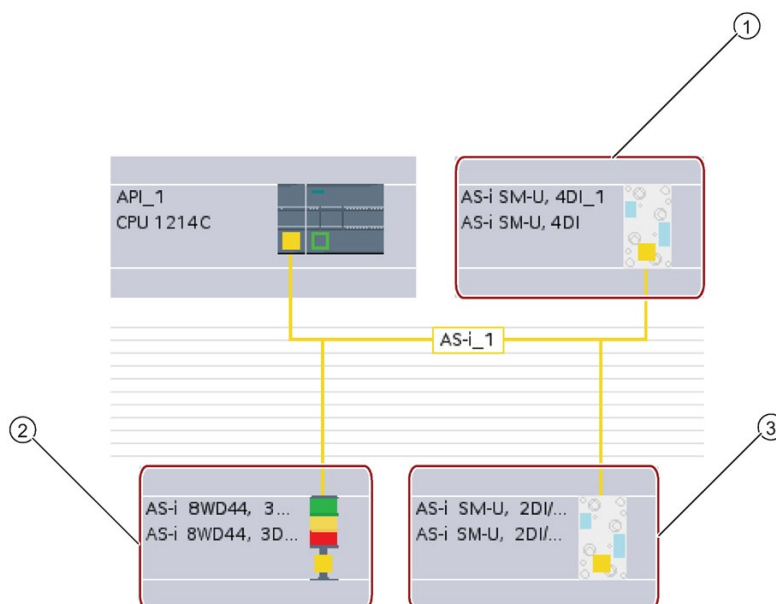
Affectation de l'adresse esclave AS-i

Dans un réseau AS-i, une adresse d'esclave AS-i est affectée à chaque appareil. Cette adresse est comprise entre 0 et 31, l'adresse 0 étant toutefois réservée aux nouveaux esclaves. Les adresses d'esclaves vont de 1(A ou B) à 31(A ou B) pour un total maximal de 62 esclaves.

Les appareils AS-i "standard" utilisent l'adresse entière et ont une adresse numérotée sans la dénomination A ou B. Les appareils AS-i "abonné A/B" utilisent la partie A ou B de chaque adresse qui permet d'utiliser deux fois chacune des 31 adresses. Les plages d'adresse vont de 1A à 31A et de 1B à 31B.

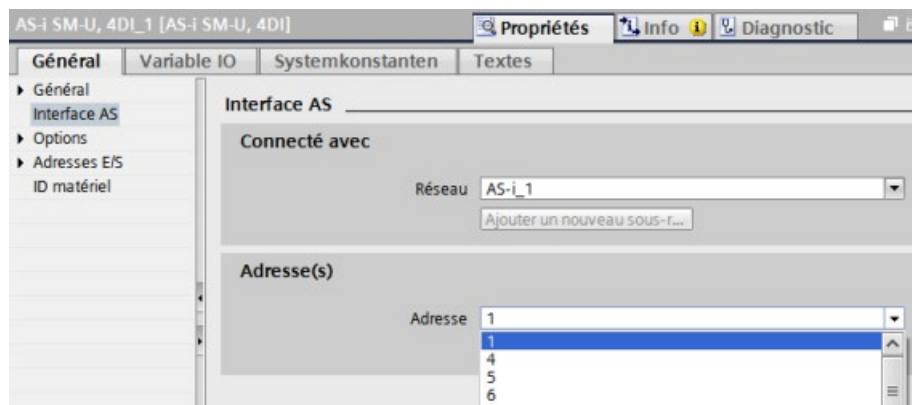
Toute adresse dans la plage de 1 à 31 peut être affectée à un esclave AS-i ; il est indifférent que les esclaves commencent à l'adresse 21 ou que le premier esclave reçoive effectivement l'adresse 1.

Dans l'exemple ci-dessous, trois appareils AS-i ont été adressés comme "1" (un appareil de type standard), "2A" (un appareil de type abonné A/B) et "3" (un appareil de type standard) :



- ① Adresse esclave AS-i 1, Appareil : SM-U AS-i, 4DI, référence : 3RG9 001-0AA00
- ② Adresse esclave AS-i 2A ; Appareil : 8WD44 AS-i, 3DO, A/B, référence : 8WD4 428-0BD
- ③ Adresse esclave AS-i 3, Appareil : SM-U AS-i, 2DI/2DO, référence : 3RG9 001-0AC00

Entrez l'adresse d'esclave AS-i ici :



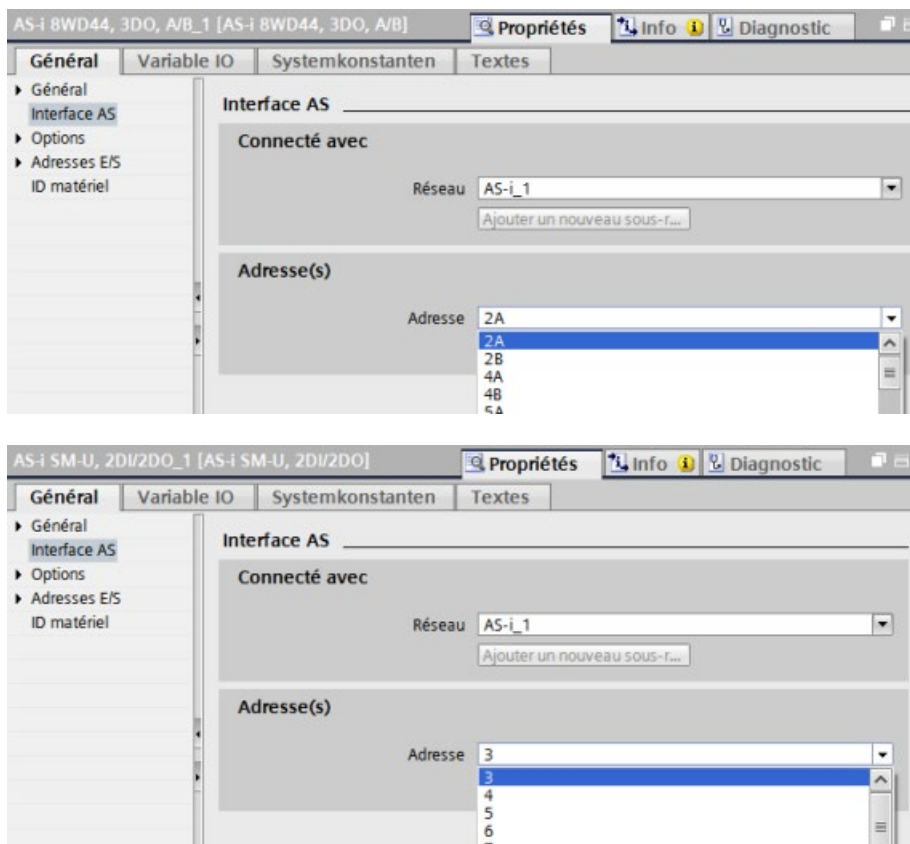


Tableau 11- 61 Paramètres pour l'interface AS-i

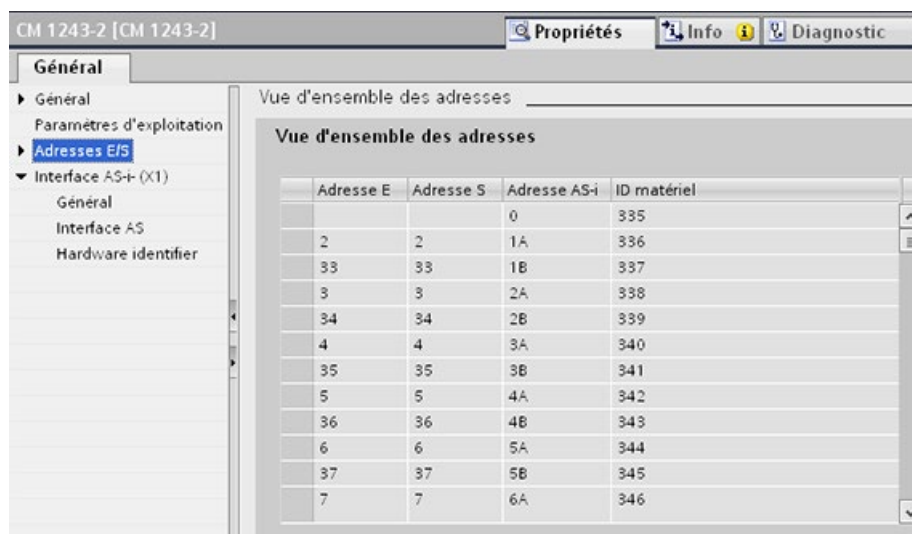
Paramètre	Description
Réseau	Nom du réseau auquel l'appareil est connecté.
Adresse(s)	Adresse AS-i affectée à l'esclave, comprise dans la plage allant de 1(A ou B) à 31(A ou B) pour un total maximal de 62 esclaves

11.4.2 Echange de données entre le programme utilisateur et les esclaves AS-i

11.4.2.1 Configuration STEP 7 de base

Le maître AS-i réserve une zone de données de 62 octets dans la zone d'E/S de la CPU. L'accès aux données TOR s'y fait en octets ; il y a un octet de données d'entrée et un octet de données de sortie pour chaque esclave.

L'affectation des liaisons AS-i des esclaves TOR AS-i aux bits de données de l'octet affecté est indiquée dans la fenêtre d'inspection du module CM 1243-2 maître AS-i.



Adresse E	Adresse S	Adresse AS-i	ID matériel
		0	335
2	2	1A	336
33	33	1B	337
3	3	2A	338
34	34	2B	339
4	4	3A	340
35	35	3B	341
5	5	4A	342
36	36	4B	343
6	6	5A	344
37	37	5B	345
7	7	6A	346

Vous pouvez accéder aux données des esclaves AS-i dans le programme utilisateur en utilisant les adresses d'E/S affichées avec les instructions logiques sur bits (par exemple, AND) ou les affectations de bits appropriées.

Remarque

"Affectation système" est automatiquement activé si vous ne configurez pas les esclaves AS-i avec STEP 7.

Si vous ne configurez pas d'esclaves, vous devez informer le module maître AS-i CM 1243-2 de la configuration de bus actuelle à l'aide de la fonction en ligne "Réelle > prévue".

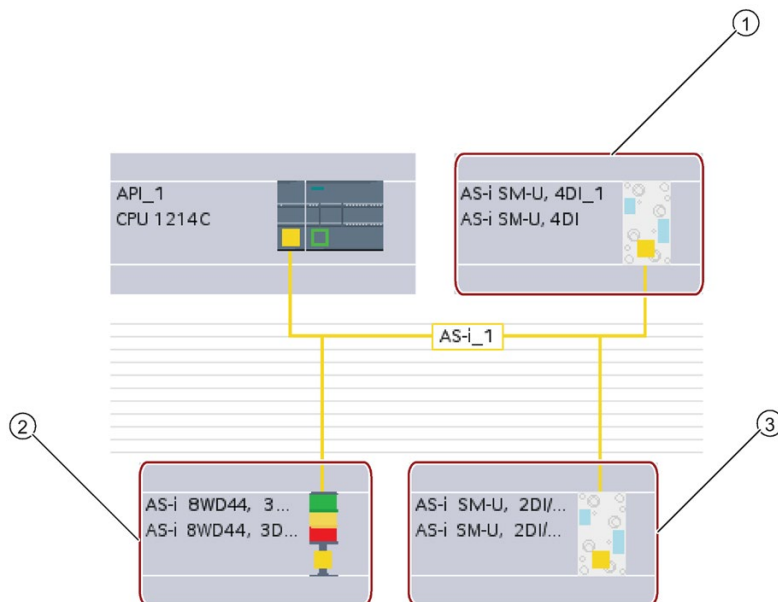
Informations supplémentaires

Vous trouverez des informations détaillées sur le module maître AS-i CM 1243-2 dans le manuel "Maître AS-i CM 1243-2 et module AS-i de découplage des données DCM 1271 pour SIMATIC S7-1200" (<https://support.industry.siemens.com/cs/ww/fr/ps/15750/man>).

11.4.2.2 Configuration d'esclaves avec STEP 7

Transfert de valeurs TOR AS-i

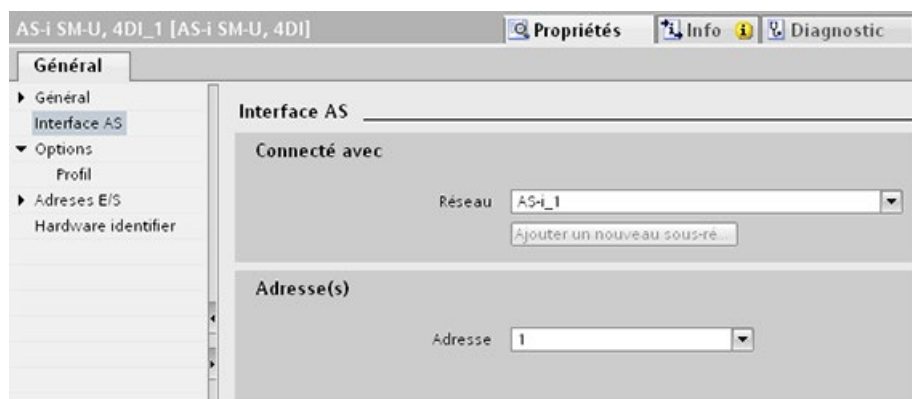
La CPU accède cycliquement aux entrées et sorties TOR des esclaves AS-i par le biais du module maître AS-i CM 1243-2. L'accès aux données se fait par les adresses d'E/S ou via un transfert d'enregistrements.



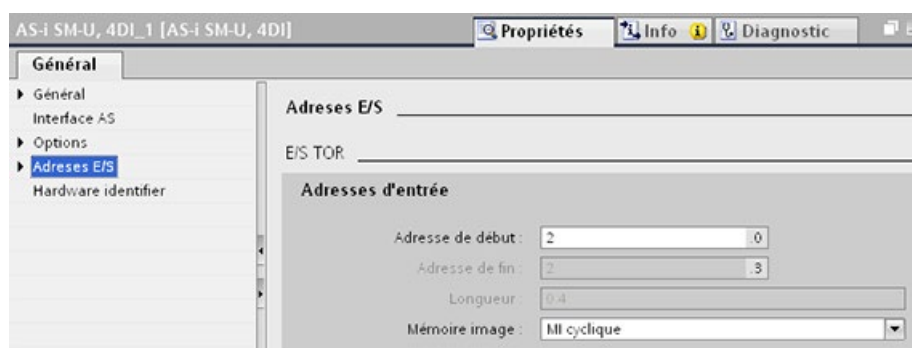
- ① Adresse esclave AS-i 1
- ② Adresse esclave AS-i 2A
- ③ Adresse esclave AS-i 3

L'accès aux données TOR s'y fait en octets (en d'autres termes, un octet est affecté à chaque esclave TOR AS-i). Lorsque vous configurez les esclaves AS-i dans STEP 7, l'adresse d'E/S pour l'accès aux données depuis le programme utilisateur apparaît dans la fenêtre d'inspection de l'esclave AS-i correspondant.

On a affecté l'adresse d'esclave 1 au module d'entrées TOR (AS-i SM-U, 4DI) dans le réseau AS-i ci-dessus. Lorsqu'on clique sur le module d'entrées TOR, l'onglet "Interface AS" des propriétés de l'appareil affiche l'adresse d'esclave, comme illustré ci-dessous :

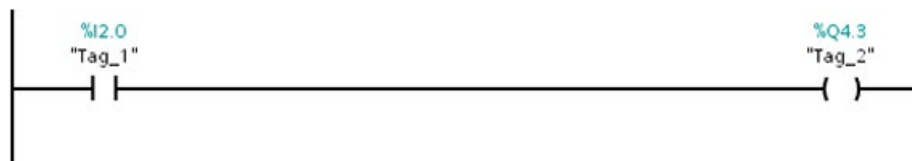


On a affecté l'adresse d'E/S 2 au module d'entrées TOR (AS-i SM-U, 4DI) dans le réseau AS-i ci-dessus. Lorsqu'on clique sur le module d'entrées TOR, l'onglet "Adresses E/S" des propriétés de l'appareil affiche l'adresse d'E/S, comme illustré ci-dessous :



Vous pouvez accéder aux données des esclaves AS-i dans le programme utilisateur en utilisant leurs adresses d'E/S avec les instructions logiques sur bits (par exemple, AND) ou les affectations de bits appropriées. Le programme simple suivant illustre comment fonctionne l'affectation :

L'entrée 2.0 est interrogée dans ce programme. Dans le système AS-i, cette entrée appartient à l'esclave 1 (octet d'entrée 2, bit 0). La sortie 4.3, qui est alors mise à 1, correspond à l'esclave AS-i (octet de sortie 4, bit 3).



Transfert de valeurs analogiques AS-i

Vous pouvez accéder aux données analogiques d'un esclave AS-i via la mémoire image de la CPU si vous avez configuré cet esclave AS-i comme esclave analogique dans STEP 7.

Si vous n'avez pas configuré l'esclave analogique dans STEP 7, vous pouvez accéder aux données de l'esclave AS-i uniquement par le biais des fonctions acycliques (interface d'enregistrement). Dans le programme utilisateur de la CPU, la lecture et l'écriture des appels AS-i se font à l'aide des instructions de périphérie décentralisée RDREC (Lire enregistrement) et WRREC (Ecrire enregistrement).

Remarque

Une configuration des esclaves AS-i définie via STEP 7 et chargée dans la station S7 est transférée par la CPU dans le module maître AS-i CM 1243-2 au démarrage de la station S7. Toute configuration existante déterminée via la fonction en ligne "Affectation système" (Page 1015) (réelle -> prévue) est écrasée.

Informations supplémentaires

Vous trouverez des informations détaillées sur le module maître AS-i CM 1243-2 dans le manuel "Maître AS-i CM 1243-2 et module AS-i de découplage des données DCM 1271 pour SIMATIC S7-1200" (<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

11.4.3 Instructions de périphérie décentralisée

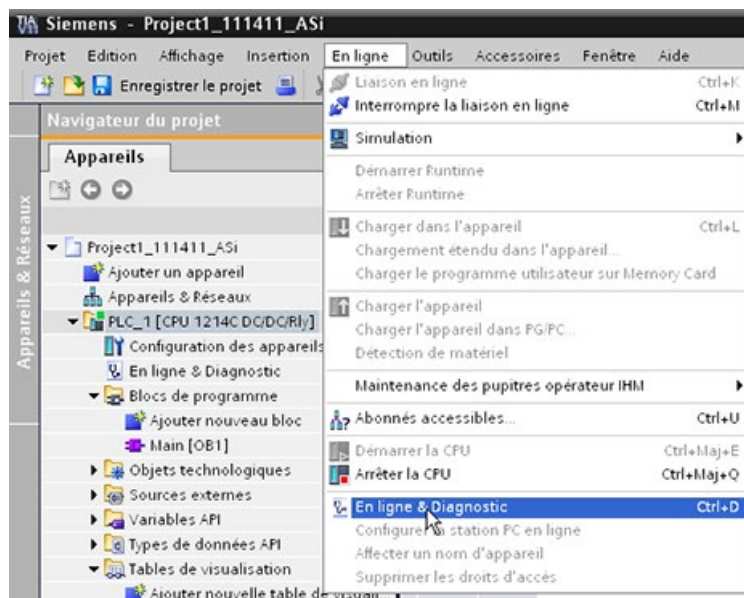
Reportez-vous à "Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)" (Page 394) pour obtenir des informations sur la manière d'utiliser les instructions de périphérie décentralisée avec ces réseaux de communication.

11.4.4 Utilisation d'outils en ligne AS-i

Changement des modes de fonctionnement AS-i en ligne

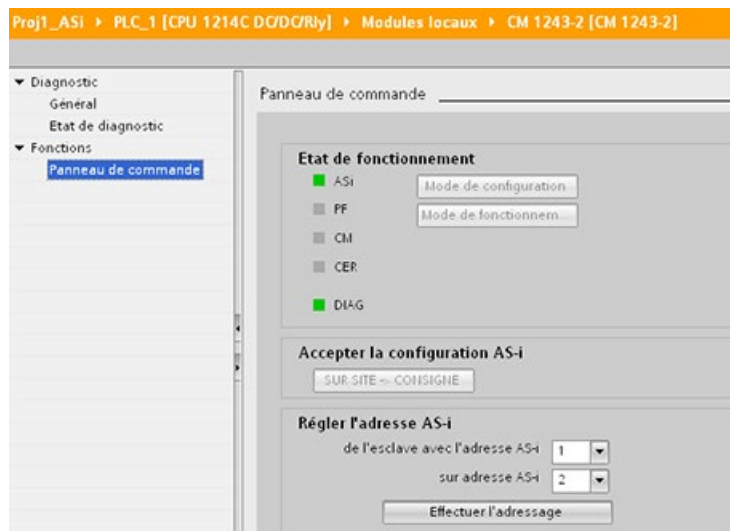
Vous devez aller en ligne pour afficher les modes de fonctionnement AS-i et en changer.

Pour aller en ligne, vous devez d'abord sélectionner le module maître AS-i CM1243-2 dans la configuration des appareils, puis cliquer sur le bouton "En ligne" dans la barre d'outils. Sélectionnez ensuite la commande "En ligne et diagnostic" du menu "En ligne".



Il existe deux modes de fonctionnement AS-i :

- le mode protégé :
 - Vous ne pouvez pas modifier les adresses d'esclave AS et d'E/S CPU.
 - La DEL "CM" verte est éteinte.
- le mode de configuration :
 - Vous pouvez apporter les modifications requises aux adresses d'esclave AS et d'E/S CPU.
 - La DEL "CM" verte est allumée.



Vous pouvez modifier l'adresse d'esclave AS-i dans la zone "Régler l'adresse AS-i". Un nouvel esclave n'ayant pas encore reçu d'adresse a toujours l'adresse 0. Il est identifié par le maître comme nouvel esclave auquel aucune adresse n'a encore été affectée et n'est pas inclus dans la communication normale tant qu'il n'a pas reçu d'adresse.

Erreur de configuration

Lorsque la DEL "CER" jaune est allumée, il y a une erreur dans la configuration de l'esclave AS-i. Cliquez sur le bouton "Réelle > prévue" pour substituer à la configuration d'esclave du module maître AS-i CM 1243-2 la configuration d'esclave du réseau de terrain AS-i.

11.5 Communication S7

11.5.1 GET et PUT (Lire les données d'une CPU distante/Écrire les données dans une CPU distante)

Vous pouvez utiliser les instructions GET et PUT pour communiquer avec des CPU S7 via des liaisons PROFINET et PROFIBUS. C'est possible uniquement si la fonction "Autoriser accès via communication PUT/GET" est activée pour la CPU partenaire dans la propriété "Protection" des propriétés de la CPU locale :

- Accès aux données dans une CPU distante : Une CPU S7-1200 peut utiliser uniquement des adresses absolues dans le champ d'entrée ADDR_x pour accéder à des variables dans des CPU distantes (S7-200/300/400/1200).
- Accès aux données dans un DB standard : Une CPU S7-1200 peut utiliser uniquement des adresses absolues dans le champ d'entrée ADDR_x pour accéder aux variables d'un DB standard dans une CPU S7 éloignée.
- Accès aux données dans un DB optimisé : Une CPU S7-1200 ne peut pas accéder aux variables d'un DB optimisé dans une CPU S7-1200 éloignée.
- Accès aux données dans une CPU locale : Une CPU S7-1200 peut utiliser soit des adresses absolues, soit des adresses symboliques comme entrées pour les champs d'entrée RD_x ou SD_x des instructions GET et PUT respectivement.

Remarque

L'opération GET/PUT du programme CPU V4.0 n'est pas activée automatiquement

L'opération GET/PUT du programme CPU V3.0 est activée automatiquement dans une CPU V4.0.

Cependant, une opération GET/PUT du programme de CPU V4.0 n'est pas activée automatiquement dans une CPU V4.0. Vous devez vous rendre sur "Configuration de l'appareil" de la CPU, l'onglet "Propriétés" de la fenêtre d'inspection puis la propriété "Protection" pour activer l'accès GET/PUT (Page 218).

Tableau 11- 62 Instructions GET et PUT

CONT/LOG	SCL	Description																						
<p style="text-align: center;">"GET_SFB_DB_1"</p> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> <p style="text-align: center;">GET Remote - Variant</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">EN</td> <td style="width: 50%;">END</td> </tr> <tr> <td>REQ</td> <td>NDR</td> </tr> <tr> <td>ID</td> <td>ERROR</td> </tr> <tr> <td>ADDR_1</td> <td>STATUS</td> </tr> <tr> <td>ADDR_2</td> <td></td> </tr> <tr> <td>ADDR_3</td> <td></td> </tr> <tr> <td>ADDR_4</td> <td></td> </tr> <tr> <td>RD_1</td> <td></td> </tr> <tr> <td>RD_2</td> <td></td> </tr> <tr> <td>RD_3</td> <td></td> </tr> <tr> <td>RD_4</td> <td></td> </tr> </table> </div>	EN	END	REQ	NDR	ID	ERROR	ADDR_1	STATUS	ADDR_2		ADDR_3		ADDR_4		RD_1		RD_2		RD_3		RD_4		<pre>"GET_DB" (req:=_bool_in_, ID:=_word_in_, ndr=>_bool_out_, error=>_bool_out_, status=>_word_out_, addr_1:=_remote_inout_, [...addr_4:=_remote_inout_,] rd_1:=_variant_inout_ [...rd_4:=_variant_inout_]);</pre>	<p>Utilisez l'instruction GET pour lire des données dans une CPU S7 éloignée. La CPU éloignée peut être à l'état MARCHÉ ou à l'état ARRÊT.</p> <p>STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.</p>
EN	END																							
REQ	NDR																							
ID	ERROR																							
ADDR_1	STATUS																							
ADDR_2																								
ADDR_3																								
ADDR_4																								
RD_1																								
RD_2																								
RD_3																								
RD_4																								
<p style="text-align: center;">"PUT_SFB_DB"</p> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> <p style="text-align: center;">PUT Remote - Variant</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">EN</td> <td style="width: 50%;">END</td> </tr> <tr> <td>REQ</td> <td>DONE</td> </tr> <tr> <td>ID</td> <td>ERROR</td> </tr> <tr> <td>ADDR_1</td> <td>STATUS</td> </tr> <tr> <td>ADDR_2</td> <td></td> </tr> <tr> <td>ADDR_3</td> <td></td> </tr> <tr> <td>ADDR_4</td> <td></td> </tr> <tr> <td>SD_1</td> <td></td> </tr> <tr> <td>SD_2</td> <td></td> </tr> <tr> <td>SD_3</td> <td></td> </tr> <tr> <td>SD_4</td> <td></td> </tr> </table> </div>	EN	END	REQ	DONE	ID	ERROR	ADDR_1	STATUS	ADDR_2		ADDR_3		ADDR_4		SD_1		SD_2		SD_3		SD_4		<pre>"PUT_DB" (req:=_bool_in_, ID:=_word_in_, done=>_bool_out_, error=>_bool_out_, status=>_word_out_, addr_1:=_remote_inout_, [...addr_4:=_remote_inout_,] sd_1:=_variant_inout_, [...sd_4:=_variant_inout_]);</pre>	<p>Utilisez l'instruction PUT pour écrire des données dans une CPU S7 éloignée. La CPU éloignée peut être à l'état MARCHÉ ou à l'état ARRÊT.</p> <p>STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.</p>
EN	END																							
REQ	DONE																							
ID	ERROR																							
ADDR_1	STATUS																							
ADDR_2																								
ADDR_3																								
ADDR_4																								
SD_1																								
SD_2																								
SD_3																								
SD_4																								

Tableau 11- 63 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	Input	Bool	Un front montant déclenche l'opération.
ID	Input	CONN_PRG (Word)	ID de liaison S7 (hexa)
NDR (GET)	Output	Bool	Nouvelles données prêtes (New Data Ready) : <ul style="list-style-type: none"> 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. 1 : La tâche a été achevée avec succès.
DONE (PUT)	Output	Bool	DONE : <ul style="list-style-type: none"> 0 : La tâche n'a pas encore commencé ou est encore en cours d'exécution. 1 : La tâche a été achevée avec succès.
ERROR STATUS	Output Output	Bool Word	<ul style="list-style-type: none"> ERROR=0 Valeur de STATUS : <ul style="list-style-type: none"> 0000H : Ni erreur ni avertissement <> 0000H : Avertissement ; STATUS fournit des informations détaillées. ERROR=1 Il y a une erreur. STATUS fournit des informations détaillées sur la nature de l'erreur.
ADDR_1	InOut	Distant	Pointeur désignant les zones de mémoire dans la CPU éloignée qui contiennent les données à lire (GET) ou les données envoyées (PUT).
ADDR_2	InOut	Distant	
ADDR_3	InOut	Distant	
ADDR_4	InOut	Distant	
RD_1 (GET) SD_1 (PUT)	InOut	Variante	Pointeur désignant les zones de mémoire dans la CPU locale qui contiennent les données lues (GET) ou à envoyer (PUT).
RD_2 (GET) SD_2 (PUT)	InOut	Variante	Types de données autorisés : Bool (un bit unique autorisé), Byte, Char, Word, Int, DWord, DInt ou Real.
RD_3 (GET) SD_3 (PUT)	InOut	Variante	Remarque : Si le pointeur accède à un DB, vous devez indiquer l'adresse absolue, par exemple
RD_4 (GET) SD_4 (PUT)	InOut	Variante	P# DB10.DBX5.0 Byte 10 Dans ce cas, 10 représente le nombre d'octets à lire ou à envoyer.

Vous devez vous assurer que la longueur (nombre d'octets) et les types de données pour les paramètres ADDR_x (CPU éloignée) et RD_x ou SD_x (CPU locale) coïncident. Le nombre après l'identificateur "Byte" est le nombre d'octets désignés par le paramètre ADDR_x, RD_x ou SD_x .

Remarque

Le nombre total d'octets reçus dans une instruction GET et le nombre total d'octets envoyés dans une instruction PUT sont limités. Ces limites dépendent de combien des quatre zones de mémoire et d'adresses possibles vous utilisez :

- Si vous utilisez uniquement ADDR_1 et RD_1/SD_1, une instruction GET peut lire 222 octets et une instruction PUT peut envoyer 212 octets.
- Si vous utilisez ADDR_1, RD_1/SD_1, ADDR_2 et RD_2/SD_2, une instruction GET peut lire 218 octets au total et une instruction PUT peut envoyer 196 octets au total.
- Si vous utilisez ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3 et RD_3/SD_3 , une instruction GET peut lire 214 octets au total et une instruction PUT peut envoyer 180 octets au total.
- Si vous utilisez ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3, RD_3/SD_3, ADDR_4, RD_4/SD_4, une instruction GET peut lire 210 octets au total et une instruction PUT peut envoyer 164 octets au total.

La somme des nombres d'octets de chacun de vos paramètres de zone de mémoire et d'adresses doit être inférieure ou égale aux limites définies. Si vous dépassez ces limites, l'instruction GET ou PUT renvoie une erreur.

En présence d'un front montant du paramètre REQ, l'opération de lecture (GET) ou l'opération d'écriture (PUT) charge les paramètres ID, ADDR_1 et RD_1 (GET) ou SD_1 (PUT).

- Pour GET : La CPU éloignée renvoie les données demandées dans les zones de réception (RD_x), en commençant au cycle suivant. Le paramètre NDR est mis à 1 lorsque l'opération de lecture s'achève sans erreur. Une nouvelle opération ne peut démarrer qu'une fois que l'opération précédente s'est achevée.
- Pour PUT : La CPU locale commence à envoyer les données (SD_x) vers l'emplacement de mémoire (ADDR_x) dans la CPU éloignée. La CPU éloignée envoie un accusé de réception lorsque l'opération d'écriture s'est achevée sans erreur. Le paramètre DONE de l'instruction PUT est alors mis à 1. Une nouvelle opération d'écriture ne peut démarrer qu'une fois que l'opération précédente s'est achevée.

Remarque

Pour garantir la cohérence des données, évaluez toujours le moment où l'opération s'est achevée (NDR = 1 pour GET ou DONE = 1 pour PUT) avant d'accéder aux données ou de déclencher une nouvelle opération de lecture ou d'écriture.

Les paramètres ERROR et STATUS fournissent des informations sur l'état de l'opération de lecture (GET) ou d'écriture (PUT).

Tableau 11- 64 Informations d'erreur

ERROR	STATUS (décimal)	Description
0	11	<ul style="list-style-type: none"> La nouvelle tâche ne peut démarrer, car la tâche précédente n'est pas encore achevée. La tâche est en cours de traitement dans une classe de priorité de priorité inférieure.
0	25	La communication a démarré. La tâche est en cours de traitement.
1	1	Problèmes de communication, tels que : <ul style="list-style-type: none"> Description de la liaison non chargée (locale ou éloignée) Interruption de la liaison (par exemple, câble, CPU hors tension ou CM/CB/CP à l'état ARRET) Liaison au partenaire non encore établie
1	2	Accusé de réception négatif du partenaire. La tâche ne peut pas être exécutée.
1	4	Erreurs de longueur ou de type de données dans les pointeurs de zone de réception ou d'émission (RD_x pour GET ou SD_x pour PUT).
1	8	Erreur d'accès sur la CPU partenaire
1	10	Accès à la mémoire utilisateur locale impossible (par exemple, tentative d'accéder à un DB effacé)
1	12	A l'appel du SFB : <ul style="list-style-type: none"> Un DB d'instance qui n'appartient pas à GET ou PUT a été indiqué. Ce n'est pas un DB d'instance qui était indiqué mais un DB global. Le DB d'instance est resté introuvable (chargement d'un nouveau DB d'instance).
1	20	<ul style="list-style-type: none"> Dépassement du nombre maximum de tâches/instances en parallèle Les instances ont été rechargées à l'état MARCHE de la CPU. Cet état est possible à la première exécution de l'instruction GET ou PUT.
1	27	Il n'y a pas d'instruction GET ou PUT correspondante dans la CPU.

11.5.2 Création d'une liaison S7

Mécanismes de connexion

Pour accéder à des partenaires de liaison éloignés avec des instructions PUT/GET, l'utilisateur doit également avoir l'autorisation d'accès.

Par défaut, l'option "Autoriser accès via communication PUT/GET" n'est pas activée. Dans ce cas, l'accès en lecture et en écriture aux données de la CPU est possible uniquement pour les liaisons de communication qui nécessitent la configuration ou la programmation à la fois de la CPU locale et du partenaire de communication. L'accès via les instructions BSEND/BRCV est possible, par exemple.

Les liaisons pour lesquelles la CPU locale est uniquement un serveur (aucune configuration/programmation de la communication avec le partenaire de communication n'existe dans la CPU locale) ne sont donc pas possibles pendant le fonctionnement de la CPU. Exemple :

- Accès PUT/GET, FETCH/WRITE ou FTP via des modules de communication
- Accès PUT/GET à partir d'autres CPU S7
- Accès IHM via la communication PUT/GET

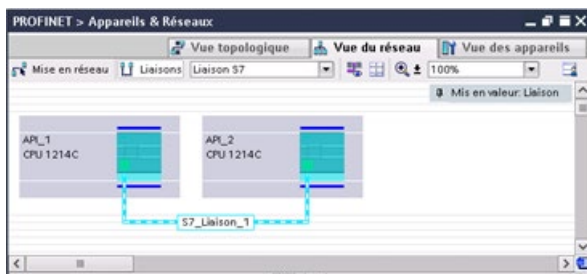
Si vous voulez autoriser l'accès aux données de la CPU à partir du client, c'est-à-dire si vous ne voulez pas restreindre les services de communication de la CPU, vous pouvez configurer la protection d'accès pour la CPU S7-1200 (Page 218) pour ce niveau de sécurité.

Types de liaisons

Le type de liaison que vous sélectionnez crée une liaison de communication vers une station partenaire. La liaison est configurée, établie et automatiquement surveillée.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Cliquez d'abord sur l'onglet "Liaisons", puis sélectionnez le type de liaison dans la liste déroulante juste à droite (une liaison S7, par exemple). Cliquez sur le carré vert (PROFINET) du premier appareil et tracez une ligne vers le carré PROFINET du deuxième appareil. Relâchez le bouton de la souris : la liaison PROFINET est créée.

Reportez-vous à "Création d'une liaison réseau" (Page 832) pour plus d'informations.



Cliquez sur le bouton "Mis en valeur : Liaison" pour accéder à la boîte de dialogue de configuration "Propriétés" de l'instruction de communication.

11.5.3 Configuration du routage local/partenaire entre deux appareils

Configuration des paramètres généraux

Vous indiquez les paramètres de communication dans la boîte de dialogue de configuration "Propriétés" de l'instruction de communication. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction.

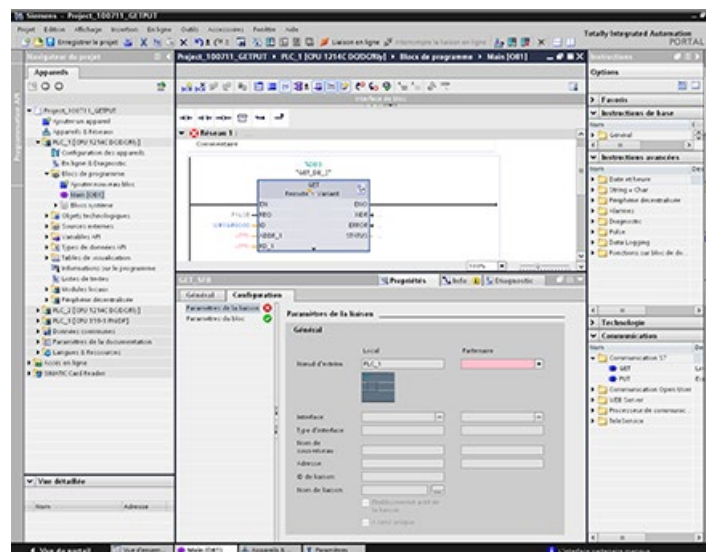
Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 833)" pour plus d'informations.

Vous définissez les TSAP ou ports à utiliser dans la section "Détails de l'adresse" de la boîte de dialogue "Paramètres de liaison". Vous indiquez le point TSAP ou le port d'une liaison dans la CPU dans la zone "TSAP local". Vous indiquez le point TSAP ou le port affecté à la liaison dans votre CPU partenaire dans la zone "TSAP partenaire".

11.5.4 Paramétrage de la liaison GET/PUT

Le paramétrage de la liaison pour instructions GET/PUT est une aide utilisateur qui vous assiste lors de la configuration de liaisons de communication CPU à CPU S7.

Après avoir inséré un bloc GET ou PUT, STEP 7 affiche la boîte de dialogue d'affectation de paramètres de liaison pour les instructions GET/PUT :



La fenêtre d'inspection affiche les propriétés de la liaison lorsque vous sélectionnez n'importe quelle partie de l'instruction. Vous pouvez configurer les paramètres de communication dans l'onglet "Configuration" des "Propriétés" pour les instructions de communication.

Remarque

L'opération GET/PUT du programme de CPU V4.1 et des versions ultérieures n'est pas activée automatiquement

L'opération GET/PUT du programme de CPU V3.0 est activée automatiquement dans une CPU V4.1 et les versions ultérieures.

Cependant, une opération GET/PUT du programme de CPU V4.1 et des versions ultérieures et d'une CPU de version ultérieure n'est pas activée automatiquement. Vous devez vous rendre sur "Configuration de l'appareil" de la CPU, l'onglet "Propriétés" de la fenêtre d'inspection puis la propriété "Protection" pour activer l'accès GET/PUT (Page 218).

11.5.4.1 Paramètres de la liaison

La page "Paramètres de la liaison" vous permet de configurer la liaison S7 nécessaire et de définir le paramètre "ID de liaison" auquel fait référence le paramètre "ID" du bloc GET/PUT. La page contient des informations sur le nœud d'extrémité local et vous permet de définir l'interface locale. Vous pouvez également définir le nœud d'extrémité partenaire.

La page "Paramètres du bloc" vous permet de configurer les autres paramètres du bloc.

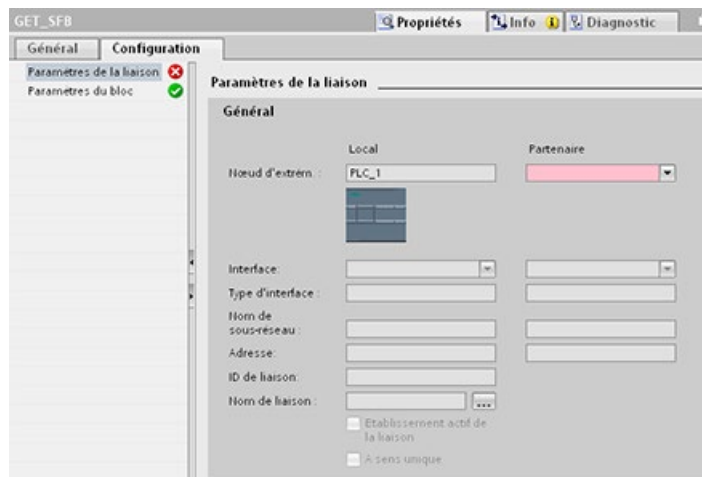


Tableau 11- 65 Paramètres de la liaison : définitions générales

Paramètre	Définition	
Paramètres de la liaison : Général	Noeud d'extrémité	Noeud d'extrémité local : Nom affecté à la CPU locale Noeud d'extrémité partenaire : nom affecté à la CPU partenaire (éloignée) Remarque : Le système affiche tous les partenaires de liaison S7 potentiels du projet en cours ainsi que l'option "non spécifié" dans la liste déroulante "Noeud d'extrémité partenaire". Un partenaire non spécifié représente un partenaire de communication ne se trouvant pas actuellement dans le projet STEP 7 (par exemple, un partenaire de communication d'appareil tiers).
	Interface	Nom affecté aux interfaces Remarque : vous pouvez modifier la liaison en changeant les interfaces locale et partenaire.
	Type d'interface	Type de l'interface
	Nom de sous-réseau	Nom affecté aux sous-réseaux
	Adresse	Adresses IP affectées Remarque : vous pouvez indiquer l'adresse éloignée d'un appareil tiers pour un partenaire de communication "non spécifié".
	ID de liaison	Numéro d'identification : généré automatiquement par le paramétrage de la liaison pour GET/PUT
	Nom de la liaison	Emplacement de stockage des données des CPU locale et partenaire : généré automatiquement par le paramétrage de la liaison pour GET/PUT
	Initialisation de la liaison	Case à cocher pour sélectionner la CPU locale en tant que liaison active
A sens unique	Case en lecture seule indiquant s'il s'agit d'une liaison à sens unique ou à double sens. Remarque : Dans une liaison GET/PUT PROFINET, l'appareil local et l'appareil partenaire peuvent tous deux agir comme client ou serveur. Cela permet une liaison à double sens, auquel cas la case "A sens unique" n'est pas cochée. Dans certains cas pour une liaison GET/PUT PROFIBUS, l'appareil partenaire peut agir uniquement en tant que serveur (S7-300, par exemple) et la case "A sens unique" est cochée.	

Paramètre ID de liaison

Il existe trois manières de modifier les ID de liaison définis par le système :

1. Vous modifiez l'ID en cours directement dans le bloc GET/PUT. Si le nouvel ID correspond à une liaison existante, la liaison est modifiée.
2. Vous modifiez l'ID en cours directement dans le bloc GET/PUT, mais cet ID n'existe pas encore. Le système crée une nouvelle liaison S7.
3. Vous modifiez l'ID en cours par le biais de la boîte de dialogue "Vue d'ensemble des liaisons" ; votre entrée est synchronisée avec le paramètre ID dans le bloc GET/PUT correspondant.

Remarque

Le paramètre ID du bloc GET/PUT n'est pas un nom de liaison, mais une expression numérique ayant, par exemple, la forme suivante : W#16#1

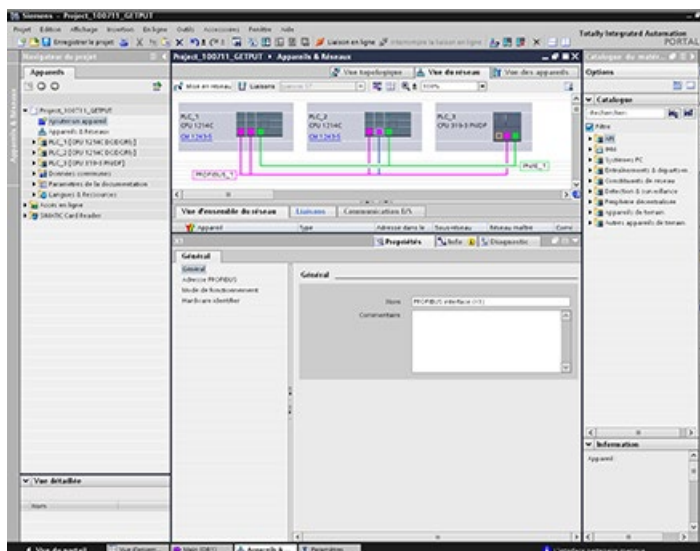
Paramètre Nom de la liaison

Vous pouvez éditer le nom de la liaison par le biais d'un contrôle utilisateur spécial, la boîte de dialogue "Vue d'ensemble des liaisons". Cette boîte de dialogue présente toutes les liaisons S7 disponibles pouvant être sélectionnées comme alternative à la communication GET/PUT en cours. Vous pouvez créer une liaison totalement nouvelle dans cette table. Cliquez sur le bouton à droite du champ "Nom de la liaison" pour ouvrir la boîte de dialogue "Vue d'ensemble des liaisons".



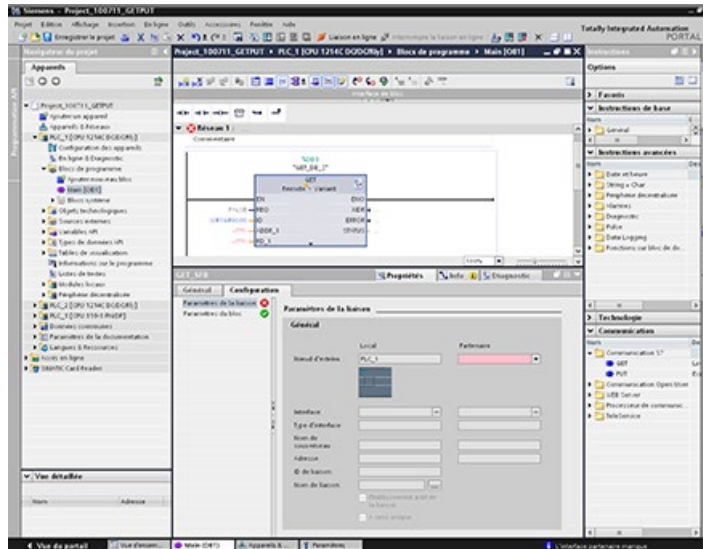
11.5.4.2 Configuration d'une liaison S7 CPU à CPU

Etant donné la configuration de PLC_1, PLC_2 et PLC_3 illustrée dans la figure ci-dessous, insérez des blocs GET ou PUT pour "PLC_1".



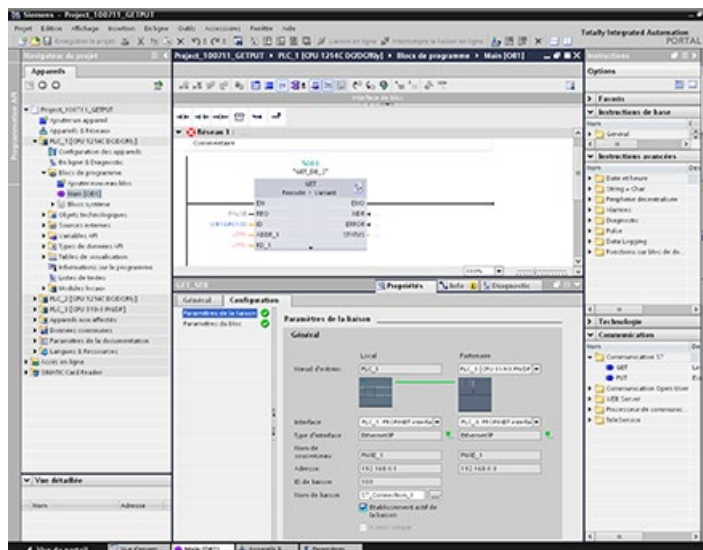
L'onglet "Propriétés" s'affiche automatiquement dans la fenêtre d'inspection avec les sélections de menu suivantes pour l'instruction GET ou PUT :

- "Configuration"
- "Paramètres de la liaison"



Configuration d'une liaison S7 PROFINET

Sélectionnez "PLC_3" pour "Nœud d'extrémité partenaire".



Le système réagit avec les modifications suivantes :

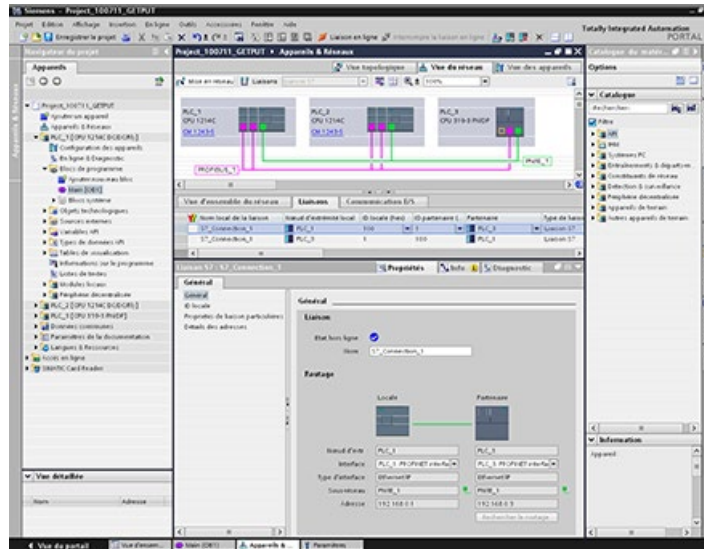
Tableau 11- 66 Paramètre de la liaison : Valeurs générales

Paramètre	Définition	
Paramètre de la liaison : Généralités	Noeud d'extrémité	"Noeud d'extrémité local" contient "PLC_1" en lecture seule. Le champ "Noeud d'extrémité partenaire" contient "PLC_3[CPU319-3PN/DP]". <ul style="list-style-type: none"> • La couleur passe de rouge à blanc. • L'image de l'appareil partenaire s'affiche. • Une ligne de connexion apparaît entre l'image de "PLC_1" et l'image de "PLC_3" (ligne Ethernet verte).
	Interface	"Interface" pour "Local" contient "CPU1214C DC/DC/DC, interface PROFINET (R0/S1)". "Interface partenaire" contient : "CPU319-3PN/DP, interface PROFINET (R0/S2)".
	Type d'interface	"Type d'interface" pour "Local" contient "Ethernet/IP" ; le contrôle est en lecture seule. "Type d'interface" pour "Partenaire" contient "Ethernet/IP" ; le contrôle est en lecture seule. Des images des types d'interface sont montrées à droite des "types d'interface" local et partenaire (icône Ethernet verte).
	Nom de sous-réseau	"Nom de sous-réseau" pour "Local" contient "PN/IE_1" ; le contrôle est en lecture seule. "Nom de sous-réseau" pour "Partenaire" contient "PN/IE_1" ; le contrôle est en lecture seule.
	Adresse	"Adresse" pour "Local" contient l'adresse IP locale ; le contrôle est en lecture seule. "Adresse" pour "Partenaire" contient l'adresse IP partenaire ; le contrôle est en lecture seule.
	ID de liaison	"ID de liaison" contient "100". Dans le bloc Main [OB1] dans l'éditeur de programmes, le paramètre "ID de liaison" du bloc GET/PUT contient également la valeur "100".
	Nom de la liaison	Ce champ contient le nom de liaison par défaut (par exemple, "S7_Connection_1") ; le contrôle est activé.
	Initialisation de la liaison	Case activée et cochée pour sélectionner la CPU locale en tant que liaison active
	A sens unique	Case en lecture seule et non cochée. Remarque : "PLC_1" (une CPU S7-1200 1214C DC/DC/Relais) et "PLC_3" (une CPU S7-300 319-3PN/DP) peuvent tous deux agir comme serveur et comme client dans une liaison GET/PUT PROFINET, ce qui permet une liaison à double sens.

La couleur de l'icône GET/PUT dans la vue des propriétés change également de rouge à vert.

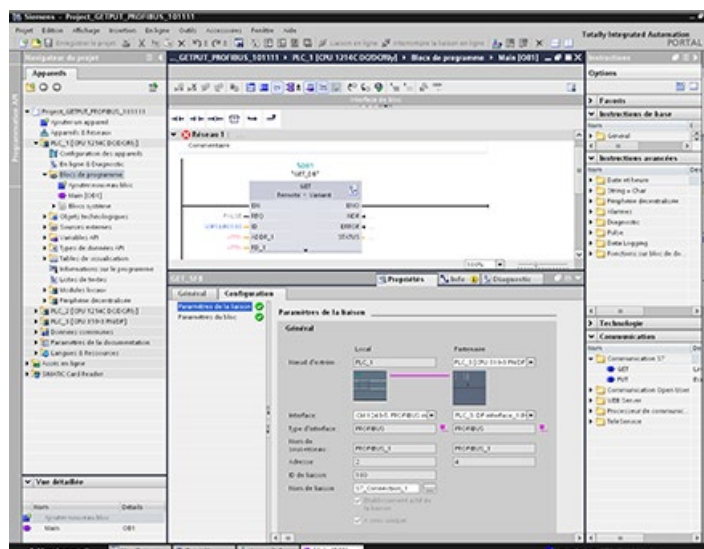
Liaison S7 PROFINET achevée

Dans la vue du réseau, une liaison S7 bilatérale est visible dans la table des liaisons entre "PLC_1" et "PLC_3".



Configuration d'une liaison S7 PROFIBUS

Sélectionnez "PLC_3" pour "Nœud d'extrémité partenaire".



Le système réagit avec les modifications suivantes :

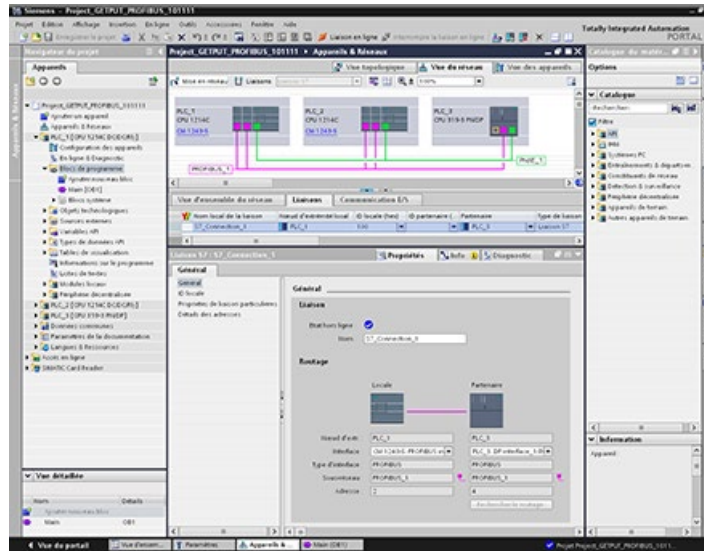
Tableau 11- 67 Paramètre de la liaison : Valeurs générales

Paramètre	Définition	
Paramètre de la liaison : Généralités	Noeud d'extrémité	"Noeud d'extrémité local" contient "PLC_1" en lecture seule. Le champ "Noeud d'extrémité partenaire" contient "PLC_3[CPU319-3PN/DP]". <ul style="list-style-type: none"> • La couleur passe de rouge à blanc. • L'image de l'appareil partenaire s'affiche. • Une ligne de connexion apparaît entre l'image de "PLC_1" et l'image de "PLC_3" (ligne PROFIBUS violette).
	Interface	"Interface" pour "Local" contient "CPU1214C DC/DC/DC, interface PROFIBUS (R0/S1)". "Interface partenaire" contient : "CPU319-3PN/DP, interface PROFIBUS (R0/S2)".
	Type d'interface	"Type d'interface" pour "Local" contient "PROFIBUS" ; le contrôle est en lecture seule. "Type d'interface" pour "Partenaire" contient "PROFIBUS" ; le contrôle est en lecture seule. Des images des types d'interface sont montrées à droite des "types d'interface" local et partenaire (icône PROFIBUS violette).
	Nom de sous-réseau	"Nom de sous-réseau" pour "Local" contient "PROFIBUS_1" ; le contrôle est en lecture seule. "Nom de sous-réseau" pour "Partenaire" contient "PROFIBUS_1" ; le contrôle est en lecture seule.
	Adresse	"Adresse" pour "Local" contient l'adresse IP locale ; le contrôle est en lecture seule. "Adresse" pour "Partenaire" contient l'adresse IP partenaire ; le contrôle est en lecture seule.
	ID de liaison	"ID de liaison" contient "100". Dans le bloc Main [OB1] dans l'éditeur de programmes, le paramètre "ID de liaison" du bloc GET/PUT contient également la valeur "100".
	Nom de la liaison	Ce champ contient le nom de liaison par défaut (par exemple, "S7_Connection_1") ; le contrôle est activé.
	Initialisation de la liaison	Case en lecture seule et cochée pour sélectionner la CPU locale en tant que liaison active
	A sens unique	Case en lecture seule et cochée. Remarque : "PLC_3" (une CPU 319-3PN/DP S7-300) peut agir uniquement en tant que serveur (et ne peut pas être également client) dans une liaison GET/PUT PROFIBUS, ce qui permet uniquement une liaison à sens unique.

La couleur de l'icône GET/PUT dans la vue des propriétés change également de rouge à vert.

Liaison S7 PROFIBUS achevée

Dans la vue du réseau, une liaison S7 à sens unique est visible dans la table des liaisons entre "PLC_1" et "PLC_3".



11.6 Que faire lorsqu'il est impossible d'accéder à la CPU par l'adresse IP

Si vous ne pouvez pas accéder à une CPU par l'adresse IP, vous pouvez définir une adresse IP d'urgence (temporaire) pour la CPU. Cette adresse IP d'urgence vous permet de rétablir la communication avec la CPU pour charger une configuration d'appareil avec une adresse IP valide.

Raisons pouvant rendre une adresse IP d'urgence nécessaire

Votre CPU peut s'avérer inaccessible si un programme présentant l'un des problèmes suivants y a été chargé :

- L'adresse IP de l'interface PROFINET de la CPU est déjà utilisée pour un autre appareil dans le réseau.
- Le sous-réseau est incorrect pour la CPU.
- Le masque de sous-réseau rend la CPU inaccessible.

Dans ces cas, la CPU n'est plus accessible à partir de STEP 7.

Affectation d'une adresse IP d'urgence

Vous pouvez affecter une adresse IP d'urgence si les conditions suivantes sont remplies :

- La configuration d'appareil dans STEP 7 comporte "Définir l'adresse IP dans le projet" pour le protocole IP.
- La CPU est à l'état ARRÊT.

Dans ces conditions, vous pouvez utiliser un outil DCP pour définir une adresse IP d'urgence comme adresse IP de l'appareil. L'outil SIMATIC Automation Tool, par exemple, possède une commande DCP de définition d'adresse IP. Vous pouvez définir une adresse IP d'urgence quel que soit le niveau de protection (Page 218) de la CPU. Une fois que vous avez défini une adresse IP temporaire avec un outil DCP, la DEL Maintenance de la CPU s'allume. Une entrée signalant que vous avez activé une adresse d'urgence d'une interface Ethernet est également inscrite dans le tampon de diagnostic.

Restauration d'une adresse IP après l'affectation d'une adresse IP d'urgence

Le tampon de diagnostic vous informe lorsque vous avez activé ou désactivé une adresse IP d'urgence. Vous pouvez remettre l'adresse IP d'urgence à zéro par mise hors tension puis sous tension de la CPU.

Après avoir défini une adresse IP d'urgence, vous pouvez charger un projet STEP 7 avec une adresse IP valide pour la CPU. Une fois le projet chargé, mettez la CPU hors tension puis sous tension.

Serveur Web

Le serveur Web pour le S7-1200 fournit un accès par pages Web aux données concernant votre CPU et aux données de processus.

Vous pouvez accéder aux pages Web du S7-1200 depuis un PC ou un appareil mobile. Pour les appareils avec de petits écrans, le serveur Web prend en charge une série de pages de base (Page 1049).

Pour établir la liaison, vous utilisez un serveur web pour accéder à l'adresse IP de la CPU S7-1200 ou l'adresse IP d'un module de CP (processeur de communication) fonctionnant sur un serveur web (Page 1047) dans le châssis local avec la CPU. Le S7-1200 accepte des liaisons simultanées multiples.



Pages Web standard

Le S7-1200 comprend des pages Web standard (Page 1048) auxquelles vous pouvez accéder à partir du navigateur Internet de votre PC (Page 1044) ou d'un dispositif mobile (Page 1046) :

- Introduction (Page 1054) : point d'entrée dans les pages Web standard
- Page d'accueil (Page 1054) : informations générales sur la CPU
- Diagnostic (Page 1056) : informations détaillées sur la CPU, y compris numéros de série, de référence et de version, protection du programme et utilisation de la mémoire
- État du module (Page 1060) : informations sur les modules locaux et distants et sur la possibilité d'actualiser le firmware pour les modules locaux
- Communication (Page 1064) : informations sur les adresses de réseau, les propriétés physiques des interfaces de communication, les statistiques, les paramètres, ainsi qu'un récapitulatif de la liaison et des informations de diagnostic
- Tampon de diagnostic (Page 1059) : mémoire tampon de diagnostic
- État des variables (Page 1068) : variables et E/S de la CPU accessibles par adresse ou nom de variable API
- Tables de visualisation (Page 1069) : tables de visualisation que vous avez configurées dans STEP 7
- Sauvegarde en ligne (Page 1072) : possibilité de sauvegarder une CPU en ligne ou de restaurer une sauvegarde en ligne effectuée précédemment

- File Browser (Page 1075) : navigateur pour fichiers stockés en interne dans la CPU ou sur une carte mémoire, par exemple, journaux et recettes
- Login (Page 1050) - se connecter en tant qu'autre utilisateur, ou se déconnecter.

Ces pages sont incluses dans la CPU S7-1200 et sont disponibles en anglais, français, allemand, espagnol, italien et chinois simplifié. Toutes les pages, à l'exception des pages d'introduction et d'accueil, nécessitent pour leur visualisation des droits utilisateur (Page 1042) supplémentaires que vous configurez dans STEP 7.

Pages Web personnalisées

Le S7-1200 fournit également une aide à la création de pages Web personnalisées qui peuvent accéder aux données de la CPU. Vous pouvez concevoir ces pages avec le logiciel auteur HTML de votre choix et inclure des commandes AWP (Automation Web Programming) prédéfinies dans votre code HTML pour accéder aux données de la CPU. Reportez-vous au chapitre Pages Web personnalisées (Page 1079) pour obtenir des informations spécifiques sur la conception de pages Web personnalisées et sur la configuration et la programmation associées dans STEP 7.

Vous pouvez accéder aux pages personnalisées à partir des pages Web standard ou de base sur un PC ou un dispositif mobile. Vous pouvez également configurer l'une de vos pages Web personnalisées comme page d'entrée (Page 1098) pour le serveur Web.

Exigences concernant le navigateur Web

Siemens a testé les pages standard du serveur Web ainsi que la prise en charge des navigateurs Web suivants :

- Internet Explorer 8 à 11
- Microsoft Edge
- Mozilla Firefox V22 à V32, V42 à V47
- Google Chrome V33 à V38, V46 à V47
- Mobile Safari et Mobile Chrome pour appareils iOS 9
- Navigateur Android pour les versions suivantes :
 - Jellybean v4.3
 - Kitkat v4.4
 - Lollipop V5.0 à v5.1
 - Marshmellow v6.0
- Mobile Chrome pour Google Android

Lors de l'utilisation du contrôle Navigateur HTML dans un projet WinCC, le serveur Web prend en charge les dispositifs Siemens HMI Panels suivants pour les pages standard :

- Basic Panels
 - Gen 2 KTP400 à KTP1200
- Comfort Panels
 - TP700 à TP2200
 - KP400 à KP1500
 - KTP400
 - TP700 Comfort Outdoor
- Mobile Panels
 - Gen 2 KTP700[F], KTP900[F]

Les restrictions liées au navigateur, en raison des interférences possibles avec l'affichage de pages Web standard ou personnalisées, sont présentées au paragraphe Contraintes (Page 1124).

12.1 Activation du serveur Web

Vous activez le serveur Web dans STEP 7, dans la configuration d'appareil pour la CPU à laquelle vous avez l'intention de vous connecter.

Procédez comme suit pour activer le serveur Web :

1. Sélectionnez la CPU dans la vue Configuration des appareils.
2. Dans la fenêtre d'inspection, sélectionnez "Serveur Web" dans les propriétés de la CPU.
3. Cochez la case "Activer le serveur Web sur tous les modules de cet appareil".
4. Pour une sécurité renforcée, vérifiez que "Autoriser l'accès uniquement via HTTPS" est sélectionné si vous avez besoin d'un accès sécurisé au Serveur web.
5. Si vous sélectionnez "Activer l'actualisation automatique" pour "Actualisation automatique", les pages Web standard seront rafraîchies toutes les dix secondes par défaut. Vous pouvez également saisir une fréquence de rafraîchissement personnalisée en secondes dans le champ "Intervalle d'actualisation".

ATTENTION

Accès de personnes non autorisées à la CPU via le serveur Web

L'accès de personnes non autorisées à la CPU ou l'affectation de valeurs invalides aux variables API peut perturber le fonctionnement du processus, ce qui peut entraîner la mort, des blessures graves et des dommages matériels importants.

Comme la validation du serveur Web permet à des utilisateurs ayant les droits correspondants de changer l'état de fonctionnement, d'écrire dans des données de l'API et d'actualiser le firmware, Siemens vous recommande de mettre en œuvre la politique de sécurité suivante :

- N'autorisez l'accès au serveur Web qu'avec le protocole HTTPS.
- Protégez les ID d'accès au serveur Web (Page 1042) avec un mot de passe fort. Les mots de passe forts contiennent au moins dix caractères, lettres, chiffres et caractères spéciaux mélangés, ne sont pas des mots du dictionnaire et ne sont pas des noms ou identifiants pouvant être déduits de vos données personnelles. Ne divulguez pas le mot de passe et changez-en fréquemment.
- N'étendez pas le privilège minimal par défaut à l'utilisateur "Everybody".
- Effectuez des recherches d'erreur et des vérifications de plage sur vos variables dans la logique de programme, car les utilisateurs des pages Web ont la possibilité de donner des valeurs invalides aux variables API.
- Utilisez un réseau virtuel privé (VPN) pour vous connecter au serveur web API S7-1200 à partir d'un site en dehors du réseau protégé.

Une fois que vous avez chargé la configuration d'appareil, vous pouvez utiliser les pages Web standard pour accéder aux pages d'introduction et d'accueil de la CPU. Pour accéder à d'autres pages, vous devez configurer un ou plusieurs utilisateurs du serveur Web (Page 1042).

Si vous avez créé et activé des pages Web personnalisées (Page 1079), vous pouvez y accéder par le menu de navigation des pages Web standard ou de base.

Remarque**Remplacement de l'appareil : remplacer une CPU V3.0 par une CPU V4.x**

Si vous remplacez une CPU V3.0 existante par une CPU V4.x (Page 1665) et convertissez votre projet V3.0 en un projet V4.x, notez que STEP 7 et la CPU V4.x conservent les paramètres du serveur Web pour :

- "Activer le serveur Web sur tous les modules de cet appareil"
 - "Autoriser l'accès uniquement via HTTPS"
-

Remarque

Si un "chargement dans la CPU à l'état MARCHE" (Page 1406) est en cours, les valeurs de données ne sont pas actualisées et vous ne pouvez pas écrire de valeurs dans les pages Web standard et personnalisées tant que le chargement dans la CPU n'est pas achevé. Le serveur Web rejette toute tentative d'écriture de valeurs de données alors que le chargement est en cours.

12.2 Configuration d'utilisateurs serveur Web

Vous pouvez configurer des utilisateurs dotés de différents niveaux de privilège pour l'accès à la CPU depuis le serveur Web.

Procédez comme suit pour configurer les utilisateurs serveur Web et leurs privilèges associés :

1. Sélectionnez la CPU dans la vue Configuration des appareils.
2. Dans la fenêtre d'inspection, sélectionnez "Serveur Web" dans les propriétés de la CPU et activez le serveur Web (Page 1040).
3. Sélectionnez "Gestion des utilisateurs" dans les propriétés du serveur Web.
4. Saisissez les noms d'utilisateur, les niveaux d'accès et les mots de passe pour les comptes utilisateur que vous souhaitez créer.

Une fois que vous avez chargé la configuration dans la CPU, seuls les utilisateurs ayant les droits correspondants pourront accéder aux fonctions du serveur Web.

Niveaux d'accès au serveur Web

STEP 7 contient un utilisateur standard "Everybody", sans mot de passe. Normalement, cet utilisateur ne dispose pas de privilèges supplémentaires et il ne peut afficher que les pages Web standard Démarrage (Page 1054) et Introduction (Page 1054). Vous pouvez cependant configurer des droits supplémentaires pour l'utilisateur "Everybody" ainsi que pour d'autres utilisateurs :

- Interroger les diagnostics
- Lecture des variables
- Ecriture des variables
- Lecture de l'état des variables
- Ecriture dans l'état des variables
- Ouverture des pages Web personnalisées
- Écriture dans les pages Web personnalisées
- Lecture de fichiers
- Ecriture/suppression de fichiers
- Changement de l'état de fonctionnement
- Faire clignoter les LED
- Mise à jour du firmware
- Sauvegarde de la CPU
- Restauration de la CPU
- Modification de paramètres système
- Modification de paramètres d'application

Si vous avez défini une page Web personnalisée comme page d'entrée (Page 1098) pour le serveur Web, l'utilisateur "Everybody" doit disposer des droits "Appeler des pages personnalisées".

 **ATTENTION**

Accès au serveur Web

Attribuer des privilèges à l'utilisateur "Everybody" permet de se connecter au serveur Web sans mot de passe. L'accès de personnes non autorisées à la CPU ou l'affectation de valeurs invalides aux variables API peut perturber le fonctionnement du processus, ce qui peut entraîner la mort, des blessures graves et des dommages matériels importants.

Comme l'utilisateur "Everybody", à condition de lui attribuer les privilèges correspondants, peut sans mot de passe changer un état de fonctionnement, écrire dans des données de l'API et actualiser le firmware, Siemens vous recommande de mettre en œuvre la politique de sécurité suivante :

- N'autorisez l'accès au serveur Web qu'avec le protocole HTTPS.
- Protégez les ID d'accès au serveur Web avec un mot de passe fort. Les mots de passe forts contiennent au moins dix caractères, lettres, chiffres et caractères spéciaux mélangés, ne sont pas des mots du dictionnaire et ne sont pas des noms ou identifiants pouvant être déduits de vos données personnelles. Ne divulguez pas le mot de passe et changez-en fréquemment.
- N'étendez pas le privilège minimal par défaut à l'utilisateur "Everybody".
- Effectuez des recherches d'erreur et des vérifications de plage sur vos variables dans la logique de programme, car les utilisateurs des pages Web ont la possibilité de donner des valeurs invalides aux variables API.
- Utilisez un réseau virtuel privé (VPN) pour vous connecter au serveur web API S7-1200 à partir d'un site en dehors du réseau protégé.

12.3 Accès aux pages Web depuis un PC

Vous pouvez accéder aux pages Web S7-1200 standard à partir d'un PC ou d'un dispositif mobile à travers l'adresse IP de la CPU S7-1200 ou l'adresse IP de n'importe quel CP fonctionnant sur un serveur web (Page 1047) dans le châssis local.

Procédez comme suit pour accéder aux pages Web standard du S7-1200 depuis un PC :

1. Vérifiez que le S7-1200 et le PC se trouvent sur un réseau Ethernet commun ou sont reliés directement l'un à l'autre au moyen d'un câble Ethernet standard.
2. Ouvrez un navigateur Web et entrez l'URL "https://ww.xx.yy.zz", "ww.xx.yy.zz" correspondant à l'adresse IP de la CPU S7-1200 ou à l'adresse IP d'un CP dans le châssis local.

Le navigateur Web ouvre la page Web standard Introduction (Page 1054) ou la page HTML par défaut de vos pages Web personnalisées si vous l'avez configurée comme page d'entrée (Page 1098).

Remarque

Utilisez un réseau virtuel privé (VPN) pour vous connecter au serveur web API S7-1200 à partir d'un site en dehors du réseau protégé. Pensez aussi que votre système d'exploitation ou environnement Web peut également imposer d'autres contraintes (Page 1124).

Accès aux pages Web standard par saisie de l'URL de la page

Vous pouvez accéder à une page Web standard spécifique avec l'URL de la page. Pour ce faire, saisissez l'URL sous la forme "https://ww.xx.yy.zz/<page>.html", "ww.xx.yy.zz" correspondant à l'adresse IP de la CPU S7-1200 ou à l'adresse IP d'un CP dans le châssis local :

- <https://ww.xx.yy.zz/start.html> : page d'accueil (Page 1054) avec des informations générales sur la CPU
- <https://ww.xx.yy.zz/identification.html> : informations d'identification (Page 1056) sur la CPU, y compris numéros de série, de référence et de version, appelée maintenant page "Diagnostic"
- <https://ww.xx.yy.zz/module.html> : informations sur les modules dans le châssis local et sur la possibilité d'actualiser le firmware (Page 1060)
- <https://ww.xx.yy.zz/communication.html> : informations de communication (Page 1064) sur les adresses réseau, les propriétés physiques des interfaces de communication et les statistiques de communication
- <https://ww.xx.yy.zz/diagnostic.html> : tampon de diagnostic (Page 1059)
- <https://ww.xx.yy.zz/variable.html> : variables et E/S de la CPU (Page 1068) accessibles par adresse, nom de variable API ou nom de variable de bloc de données
- <https://ww.xx.yy.zz/watch.html> : tables de visualisation (Page 1069)
- <https://ww.xx.yy.zz/filebrowser.html> : navigateur pour accéder aux fichiers journaux ou aux fichiers de recette (Page 1075) stockés en interne dans la CPU ou sur une carte mémoire

- <https://ww.xx.yy.zz/index.html> : page d'introduction pour entrer dans les pages Web standard
- <https://ww.xx.yy.zz/login.html> : page d'ouverture de session (Page 1050) si aucun utilisateur n'est actuellement connecté ; sinon, la page est vide.

Par exemple, si vous entrez "<https://ww.xx.yy.zz/communication.html>", le navigateur affiche la page sur la communication.

Remarque

Notez que toute page Web standard qui n'est pas énumérée explicitement ci-dessus (par exemple la page Sauvegarde en ligne (Page 1072)) n'a pas d'accès URL direct.

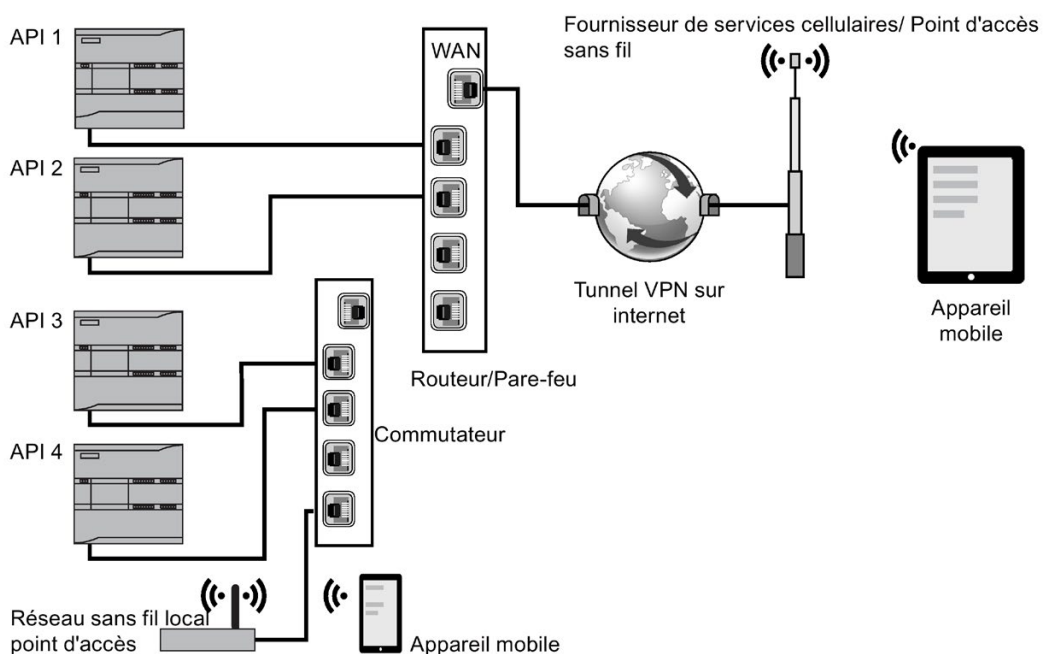
Accès sécurisé

Utilisez un réseau virtuel privé (VPN) pour vous connecter au serveur web API S7-1200 à partir d'un site en dehors du réseau protégé. Nécessitez l'utilisation et utilisez <https://> instead of <http://> pour un accès sécurisé (Page 1040) aux pages Web standard. Lorsque vous vous connectez au S7-1200 avec <https://>, le site Web crypte la session avec un certificat numérique. Le serveur Web transmet les données de manière sécurisée et il n'est pas visible à tous. Vous recevez normalement un avertissement de sécurité que vous pouvez acquitter avec "Oui" pour atteindre les pages Web standard. Pour éviter cet avertissement de sécurité à chaque accès sécurisé, vous pouvez importer le certificat de sécurité Siemens dans votre navigateur Web (Page 1126).

12.4 Accès aux pages Web depuis un dispositif mobile

Pour accéder à un S7-1200 à partir d'un dispositif mobile, vous devez connecter votre API à un réseau qui se connecte à Internet ou à un point d'accès sans fil local. Utilisez un réseau virtuel privé (VPN) sécurisé pour connecter un dispositif mobile au serveur Web du S7-1200. Vous pouvez utiliser la redirection de port dans le routeur sans fil pour mapper l'adresse IP de l'API sur une adresse par laquelle un dispositif mobile peut y accéder à l'aide d'Internet. Pour configurer la redirection de port, suivez les instructions de configuration logicielle de votre routeur. Vous pouvez connecter autant d'API et de dispositifs de commutation que votre routeur peut accepter.

Sans redirection de port, vous pouvez vous connecter à un API, mais uniquement localement dans la portée du signal sans fil.



Dans cet exemple, un dispositif mobile qui est à portée du point d'accès sans fil local peut se connecter à l'API 3 et à l'API 4 via leur adresse IP. Un dispositif mobile peut se connecter à l'API 1 et à l'API 2 à partir d'Internet en étant hors de portée du réseau sans fil local et en utilisant l'adresse de redirection de port pour chaque API.

Pour accéder aux pages Web standard, vous devez avoir accès à un service cellulaire ou à un point d'accès sans fil. Pour accéder à un API par Internet, saisissez l'adresse de redirection de port dans le navigateur Web de votre dispositif mobile, par exemple `http://ww.xx.yy.zz:pppp` ou `https://ww.xx.yy.zz:pppp`, "ww.xx.yy.zz" représentant l'adresse du routeur et "pppp" l'affectation de port pour un API spécifique.

Pour un accès local via un point d'accès sans fil local, entrez l'adresse IP de la CPU S7-1200 ou un CP fonctionnant sur un serveur web (Page 1047) dans le châssis local :

- `http://ww.xx.yy.zz` ou `https://ww.xx.yy.zz` pour accéder aux pages Web standard (Page 1048)
- `http://ww.xx.yy.zz/basic` ou `https://ww.xx.yy.zz/basic` pour accéder aux pages Web de base (Page 1049)

Pour une sécurité renforcée, configurez le serveur Web comme étant uniquement accessible par accès sécurisé (HTTPS) (Page 1040).

12.5 Utilisation d'un module CP pour accéder à des pages Web

Que vous accédiez au serveur Web depuis un PC ou un dispositif mobile, vous pouvez vous connecter aux pages Web standard à travers l'un des modules CP suivants lorsque vous l'avez configuré dans STEP 7 et installé dans le châssis local avec la CPU S7-1200 :

- CP 1242-7 GPRS V2
- CP 1243-1
- CP 1243-1 PCC
- CP 1243-7 LTE-EU
- CP 1243-7 LTE-US
- CP 1243-8 IRC

Vous utilisez la page Web standard de démarrage (Page 1054) pour accéder aux pages Web à travers ces modules CP. La page de démarrage (Start Page) affiche tous les modules CP configurés et installés dont vous disposez dans votre châssis local, mais vous ne pouvez accéder aux pages Web qu'à partir de ceux répertoriés ci-dessus.

Remarque

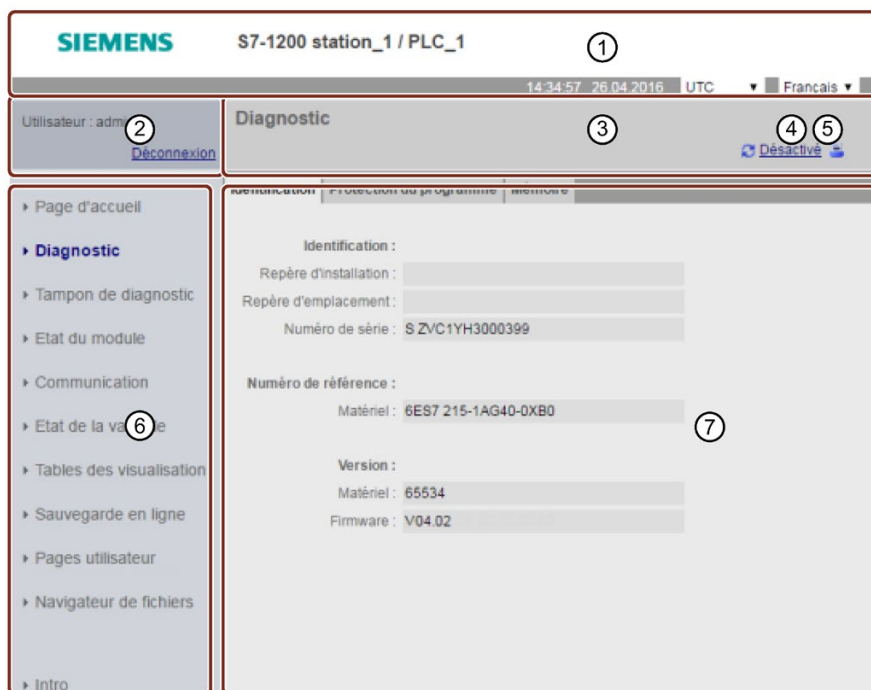
Accès à des pages Web standard lorsque les CP fonctionnant sur un serveur Web se trouvent dans le châssis local

Il se peut que vous constatiez des retards pouvant aller jusqu'à une ou deux minutes lorsque vous vous connectez aux pages Web S7-1200 standard quand des CP fonctionnant sur un serveur Web se trouvent dans le châssis local. Si les pages ne s'avèrent pas disponibles, ou si vous rencontrez des erreurs, patientez une ou deux minutes puis rafraîchissez la page.

12.6 Pages Web standard

12.6.1 Disposition des pages Web standard

Les pages Web S7-1200 standard ont une disposition commune avec des liens de navigation et des contrôles de page. Que vous affichiez la page sur un PC ou sur un dispositif mobile, chaque page a la même zone de contenu, mais la disposition et les contrôles de navigation varient en fonction de la taille de l'écran et de la résolution de l'appareil. Sur un PC standard ou un grand dispositif mobile, la disposition d'une page Web standard se présente comme suit :



- ① En-tête du serveur Web avec un sélecteur pour afficher l'heure locale de l'API ou l'heure UTC et un sélecteur pour la langue d'affichage (Page 177)
- ② Ouverture ou fermeture de session
- ③ En-tête de page Web standard avec le nom de la page que vous regardez. Cet exemple montre la page d'identification de la CPU. Certaines pages Web standard, telles que les informations sur les modules, affichent également ici un chemin de navigation si plusieurs écrans de ce type sont accessibles.
- ④ Bouton d'actualisation : pour les pages à actualisation automatique, il active ou désactive la fonction d'actualisation automatique ; pour les pages sans actualisation automatique, il provoque l'actualisation avec des données actuelles.
- ⑤ Bouton d'impression : prépare et affiche une version imprimable des informations disponibles dans la page affichée.
- ⑥ Zone de navigation pour passer à une autre page
- ⑦ Zone de contenu de la page Web standard que vous consultez. Cet exemple montre la page de diagnostic.

Remarque**Pages Web standard de modules CP**

Certains processeurs de communication (Page 1047) fournissent des pages Web standard à l'apparence et aux fonctionnalités similaires à celles des pages Web standard de la CPU S7-1200. La documentation de ces CP contient une description de ces pages Web CP standard.

12.6.2 Pages de base

Le serveur Web fournit des pages de base conçues pour être utilisées sur des dispositifs mobiles. Vous y accédez par l'URL composée de l'adresse IP de votre dispositif et de "basic" : <http://ww.xx.yy.zz/basic> ou <https://ww.xx.yy.zz/basic>

Les pages de base ont une apparence similaire aux pages standard, mais présentent quelques différences. La page omet la zone de navigation, la zone d'ouverture de session et la zone d'en-tête et comprend des boutons pour avancer et reculer dans les pages Web. Les pages de base comportent également un bouton Page d'accueil qui vous amène à une page de navigation. Vous pouvez également utiliser les contrôles de navigation fournis par votre dispositif mobile pour la navigation. Par exemple, la page de base Diagnostic se présente verticalement comme suit :

La résolution minimale requise pour l'affichage d'une page de base est de 240 x 240 pixels.



Notez que les illustrations des pages Web standard dans ce chapitre représentent l'apparence standard sur PC. La plupart des pages Web standard ont des pages de base équivalentes.

12.6.3 Ouverture de session et privilèges utilisateur

Chaque page Web standard sur votre PC comporte une fenêtre d'ouverture de session au-dessus de la zone de navigation. Pour des raisons d'espace, les pages Web de base offrent une page d'ouverture de session séparée. S7-1200 prend en charge des connexions utilisateur simultanées avec différents niveaux d'accès (de privilège) :

- Interroger les diagnostics
- Lecture des variables
- Ecriture des variables
- Lecture de l'état des variables
- Ecriture dans l'état des variables
- Ouvrir des pages personnalisées
- Ecrire dans des pages personnalisées
- Lecture de fichiers
- Ecriture/suppression de fichiers
- Changement de l'état de fonctionnement
- Faire clignoter les LED
- Effectuer la mise à jour du firmware
- Sauvegarde de la CPU
- Restauration de la CPU
- Modifier le paramètre système
- Modifier le paramètre d'application

Vous pouvez configurer des rôles utilisateur, les niveaux (privilèges) d'accès associés et les mots de passe (Page 1042) dans les propriétés de la gestion des utilisateurs du serveur Web de la configuration d'appareil STEP 7 de la CPU.

Ouverture de session

STEP 7 contient un utilisateur standard "Everybody", sans mot de passe. Normalement, cet utilisateur ne dispose pas de privilèges supplémentaires et il ne peut afficher que les pages Web standard Démarrage (Page 1054) et Introduction (Page 1054). Vous pouvez cependant attribuer à l'utilisateur "Everybody" les mêmes privilèges qu'aux autres utilisateurs que vous configurez :

ATTENTION

Accès au serveur Web

Attribuer des privilèges à l'utilisateur "Everybody" permet de se connecter au serveur Web sans mot de passe. L'accès de personnes non autorisées à la CPU ou l'affectation de valeurs invalides aux variables API peut perturber le fonctionnement du processus, ce qui peut entraîner la mort, des blessures graves et des dommages matériels importants.

Comme l'utilisateur "Everybody", à condition de lui attribuer les privilèges correspondants, peut sans mot de passe changer un état de fonctionnement, écrire dans des données de l'API et actualiser le firmware, Siemens vous recommande de mettre en œuvre la politique de sécurité suivante :

- N'autorisez l'accès au serveur Web qu'avec le protocole HTTPS.
- Protégez les ID utilisateur du serveur Web (Page 1042) avec un mot de passe fort. Les mots de passe forts contiennent au moins dix caractères, lettres, chiffres et caractères spéciaux mélangés, ne sont pas des mots du dictionnaire et ne sont pas des noms ou identifiants pouvant être déduits de vos données personnelles. Ne divulguez pas le mot de passe et changez-en fréquemment.
- N'étendez pas les droits minimum par défaut de l'utilisateur "Tous".
- Effectuez des recherches d'erreur et des vérifications de plage sur vos variables dans la logique de programme, car les utilisateurs des pages Web ont la possibilité de donner des valeurs invalides aux variables API.
- Utilisez un réseau virtuel privé (VPN) pour vous connecter au serveur web API S7-1200 à partir d'un site en dehors du réseau protégé.

Mais vous devez posséder les privilèges requis pour pouvoir exécuter certaines actions, comme changer l'état de fonctionnement de l'automate, écrire des valeurs en mémoire ou mettre à jour le firmware de la CPU. Notez que si vous avez défini le niveau de protection de la CPU (Page 218) à "Aucun accès (protection complète)", l'utilisateur "Everybody" n'aura pas la permission d'accéder au serveur Web, et ce quels que soient les paramètres d'autorisation utilisateur du serveur Web.



Le cadre d'ouverture de session se situe près du coin supérieur gauche de chaque page Web standard lorsqu'on y accède depuis un PC ou un large appareil mobile.



La page d'ouverture de session est une page séparée sur les petits dispositifs mobiles qui affichent les pages de base. Elle peut être sélectionnée dans la page d'accueil.

Procédez comme suit pour ouvrir une session :

1. Entrez le nom d'utilisateur dans le champ Nom.
2. Entrez le mot de passe dans le champ Mot de passe.

Votre session expire après trente minutes d'inactivité. Si la page actuellement chargée est rafraîchie en permanence, le délai pour la session est réinitialisé, empêchant ainsi la session d'expirer.

Remarque

En cas de problème à l'ouverture de session, téléchargez le certificat de sécurité Siemens (Page 1126) sur la page Introduction (Page 1054). Vous pourrez alors ouvrir une session sans rencontrer d'erreurs.

Fermeture de session



Pour fermer la session, il suffit de cliquer sur le lien de fermeture de session "Déconnexion" dans n'importe quelle page affichée depuis un PC ou un large appareil mobile.

À partir des pages de base, naviguez jusqu'à la page d'ouverture/de fermeture de session de la page d'accueil (Home page) et appuyez sur le bouton "Déconnexion".

Après avoir fermé la session, vous pouvez uniquement accéder et consulter les pages Web standard selon les privilèges de l'utilisateur "Everybody". Chacune des descriptions de page Web standard définit les privilèges requis pour cette page.

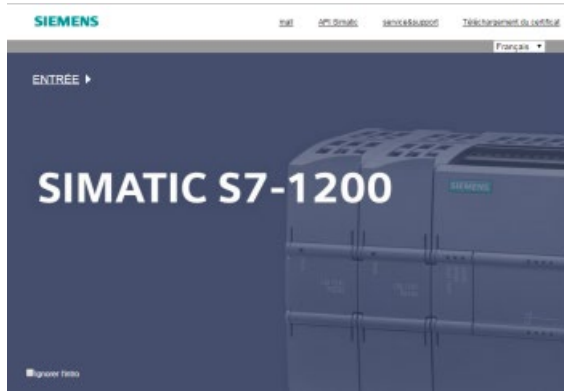
Remarque

Déconnectez-vous avant de fermer le serveur Web

Si vous êtes connecté au serveur Web, assurez-vous de vous déconnecter avant de fermer le navigateur web. Le serveur Web prend en charge au plus 7 connexions simultanées.

12.6.4 Introduction

La page Introduction est l'écran d'accueil pour entrer dans les pages Web standard du S7-1200.



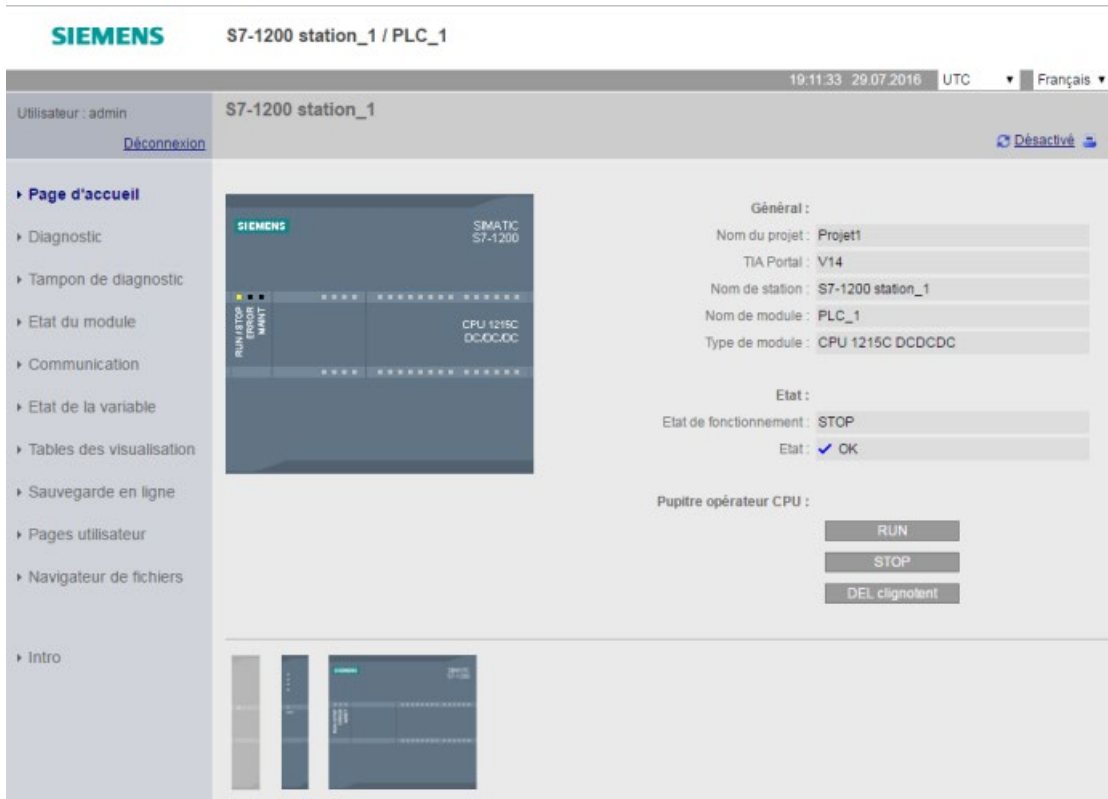
Dans cette page, cliquez sur "Entrer" pour accéder aux pages Web standard du S7-1200. En haut de l'écran se trouvent des liens vers des sites Web utiles de Siemens, ainsi qu'un lien pour télécharger le certificat de sécurité Siemens (Page 1126). Vous pouvez également choisir de sauter la page d'introduction pour des accès futurs au serveur Web.

12.6.5 Page de démarrage

La page d'accueil affiche une représentation de la CPU ou du CP auquel vous êtes connecté et donne des informations générales sur l'appareil et la version de TIA Portal que vous avez utilisée pour charger le projet dans la CPU. Pour la CPU, vous pouvez également utiliser les boutons pour changer l'état de fonctionnement et faire clignoter les LED à condition d'avoir ouvert une session (Page 1050) avec les droits (Page 1042) "Changer le mode de fonctionnement".

La partie inférieure de l'écran est visible si vous avez configuré et installé les modules CP fonctionnant sur un serveur web (Page 1047) dans le châssis local avec la CPU S7-1200. Vous pouvez survoler et cliquer sur un module CP fonctionnant sur un serveur web pour accéder aux pages Web standard. Reportez-vous à la documentation de votre module CP pour obtenir des informations sur les pages Web du module CP. Vous voyez le nom du module CP lorsque vous passez dessus.

Le serveur Web affiche également tous les autres modules CM et CP dans le châssis local, mais vous ne pouvez pas cliquer dessus puisqu'ils ne contiennent pas de pages Web. L'apparence du module pour ces CM et CP est en gris clair (désensibilisé) pour indiquer qu'il s'agit de module en affichage seul et non cliquables.



Remarquez que les CPU à sécurité intégrée S7-1200 affichent des données supplémentaires sur cette page liées à la sécurité fonctionnelle.

12.6.6 Diagnostic

La page Diagnostic affiche des caractéristiques identifiant la CPU, les paramètres de configuration pour la protection du savoir-faire et l'utilisation de la mémoire de chargement, de la mémoire de travail et de la mémoire rémanente.

Cette page comporte trois onglets :

- Identification : caractéristiques d'identification du module et informations sur l'installation et l'emplacement provenant de STEP 7
- Protection du programme : état de la protection du savoir-faire et lien CPU, ce qui peut être utile dans la planification des pièces de rechange, ainsi que le paramétrage de configuration STEP 7 pour autoriser ou empêcher la copie de la mémoire de chargement interne dans une mémoire de chargement externe (carte mémoire SIMATIC).
- Mémoire : utilisation de la mémoire de chargement, de la mémoire de travail et de la mémoire rémanente

Pour les CPU F, il y a un onglet "Sécurité" en plus.

L'affichage de l'onglet Identification nécessite les droits (Page 1042) "Interroger le diagnostic".

Onglet Identification



Onglet Protection du programme

L'onglet Protection du programme comprend les informations suivantes :

- Protection Know-How (Page 222) : Indique si vous avez configuré une protection du savoir-faire pour l'un quelconque des blocs de programme dans STEP 7.
- Lien (Page 223) : Indique si vous avez lié le programme à la CPU ou à la carte mémoire SIMATIC.
- Copie du programme en carte mémoire (Page 221) : Indique si vous avez autorisé la copie du programme de la mémoire de chargement interne dans une mémoire de chargement externe (carte mémoire SIMATIC).

The screenshot shows the Siemens SIMATIC Manager web interface. At the top, it displays 'SIEMENS' and 'S7-1200 station_1 / PLC_1'. The main header includes the time '20:58:35', the date '28.07.2016', the time zone 'UTC', and the language 'Français'. The user is identified as 'admin' with a 'Déconnexion' link. The 'Diagnostic' section is active, showing three tabs: 'Identification', 'Protection du programme', and 'Mémoire'. The 'Protection du programme' tab is selected and contains the following information:

- Protection du savoir-faire :** Protection du savoir-faire : Non présent
- Liaison :**
 - Numéro de série de la CPU : Aucune liaison
 - Numéro de série de la carte mémoire : Aucune liaison
- Copie du programme sur la carte mémoire :** Depuis la mémoire de chargement interne : Activé

A left-hand navigation menu is visible, listing various diagnostic and system pages such as 'Page d'accueil', 'Diagnostic', 'Tampon de diagnostic', 'Etat du module', 'Communication', 'Etat de la variable', 'Tables des visualisation', 'Sauvegarde en ligne', 'Pages utilisateur', 'Navigateur de fichiers', and 'Intro'.

Onglet Mémoire

The screenshot shows the Siemens S7-1200 diagnostic web interface. The title bar displays 'SIEMENS S7-1200 station_1 / PLC_1' and the current time/date is '20:23:48 28.07.2016 UTC'. The user is logged in as 'admin' with a 'Déconnexion' link. The 'Diagnostic' section is active, and the 'Mémoire' tab is selected. The interface displays three memory usage metrics:

Mémoire de chargement	Mémoire de travail	Mémoire rémanente
1.6% occupée(s) Libre : 3.94 MB / Total : 4.00 MB	6.2% occupée(s) Libre : 117.20 KB / Total : 125.00 KB	43.4% occupée(s) Libre : 5.66 KB / Total : 10.00 KB

Onglet Sécurité

Pour plus d'informations sur l'onglet Sécurité de la page Diagnostic, voir Manuel de sécurité fonctionnelle du S7-1200

(<https://support.industry.siemens.com/cs/ww/fr/view/104547552/en>).

12.6.7 Mémoire tampon de diagnostic

La page de mémoire tampon de diagnostic (Diagnostic Buffer) affiche des événements de diagnostic. L'événement le plus récent est le numéro 1 en haut et le plus ancien est le numéro 50. Le champ de sélection à gauche permet de sélectionner la plage d'entrées de mémoire tampon de diagnostic à afficher, 1 à 25 ou bien 26 à 50. Dans le champ de sélection à droite, vous pouvez choisir d'afficher l'heure en heures UTC ou en heures locales API. La partie supérieure de la page affiche les entrées de diagnostic avec l'horodatage auquel l'événement s'est produit.

Dans la partie supérieure de la page, vous pouvez sélectionner n'importe quelle entrée individuelle pour afficher des informations détaillées sur cette entrée dans la partie inférieure de la page. Notez que la langue d'affichage des entrées de la mémoire tampon de diagnostic dépend du paramétrage pour le multilinguisme (Page 183) dans la configuration d'appareil.

The screenshot shows the Siemens S7-1200 station diagnostic buffer page. The page title is "SIEMENS S7-1200 station_1 / PLC_1". The user is logged in as "admin" and the page is in French. The diagnostic buffer is set to show entries 1-25. The table below shows the following events:

Numéro	Heure	Date	Etat	Evénement
1	19:43:05	28.07.2016	Evénement apparaissant	New startup information - Current CPU oper
2	19:43:05	28.07.2016	Evénement apparaissant	Communication initiated request: STOP - Cf
3	19:38:24	28.07.2016	Evénement apparaissant	Follow-on operating mode change - CPU ct
4	19:38:24	28.07.2016	Evénement apparaissant	Communication initiated request: WARM RE
5	19:38:24	28.07.2016	Evénement apparaissant	New startup information - Current CPU oper
6	19:38:20	28.07.2016	Evénement apparaissant	New startup information - Current CPU oper
7	19:38:19	28.07.2016	Evénement apparaissant	New startup information - Current CPU oper
8	19:38:04	28.07.2016	Evénement apparaissant	New startup information - Current CPU oper
9	19:38:04	28.07.2016	Evénement apparaissant	Communication initiated request: STOP - Cf
10	19:32:58	28.07.2016	Evénement apparaissant	Follow-on operating mode change - CPU ct
11	19:32:57	28.07.2016	Evénement apparaissant	Communication initiated request: WARM RE

The detailed view for event 1 shows the following information:

- Détails : 1** ID d'événement : 16# 02:4000
- CPU info: New startup information
- Pending startup inhibit(s):
 - Manual restart required
 - Current CPU operating mode: STOP
- HW_ID= 52
- Evénement apparaissant

Vous devez posséder le privilège (Page 1042) "interroger des diagnostics" pour afficher la page Tampon de diagnostic.

12.6.8 Informations sur les modules

La page d'information sur les modules (Module Information) fournit des informations sur tous les modules dans le châssis local. La partie supérieure de cette page affiche un récapitulatif des modules basé sur la configuration des appareils dans STEP 7. La partie inférieure montre des informations sur l'état, l'identification et le firmware du module sélectionné, informations basées sur le module connecté correspondant. La page d'information sur les modules offre également la possibilité d'effectuer une mise à jour du firmware.

Vous devez posséder le privilège (Page 1042) d'"interroger des diagnostics" pour afficher la page d'information sur les modules.

Informations modules : onglet Etat

L'onglet Etat (Status) dans la partie inférieure de la page d'informations sur les modules affiche une description de l'état en cours du module sélectionné dans la partie supérieure. Cet onglet est vide si le module n'a pas d'état de diagnostic en attente.

SIEMENS S7-1200 station_1 / PLC_1

13:48:46 29.07.2016 UTC Français

Utilisateur : admin Déconnexion Désactivé









Etat du module - S7-1200 station_1

Emplacement	Etat	Nom	Détails	Numéro de référence	Adresse d'entrée	Adresse de sortie	Commentaire
1	✓	PLC_1	Détails	6ES7 215-1AG40-0XB0			
2	✓	DI 16/DQ 16x24VDC_1	Détails	6ES7 223-1BL32-0XB0	8	8	

État Identification Firmware

Icônes d'état pour les modules

Pour chaque module, la colonne "État" dans la partie supérieure affiche une icône qui indique l'état de ce module :

Icône	Signification
	Pas de défaut
	Désactivé
	Maintenance nécessaire
	Maintenance requise
	Défaut
	La CPU ne peut pas accéder au module ou à l'appareil (pour les appareils autres que la CPU).
	La CPU a établi une liaison à l'appareil, mais l'état du module est inconnu (pour les appareils autres que la CPU).
	Les données d'entrée et de sortie ne sont pas disponibles parce que le sous-module a bloqué ses voies d'E/S (pour les appareils autres que la CPU).

Zoom avant

Vous pouvez sélectionner un lien dans la partie supérieure pour faire un zoom avant sur les informations concernant ce module particulier. Les modules munis de sous-modules ont des liens pour chaque sous-module. Le type d'informations affichées varie selon le module sélectionné. Ainsi, la page d'information sur les modules affiche initialement le nom de la station S7-12000, un indicateur d'état et un commentaire. Si vous faites un zoom avant sur la CPU, les informations de module affichent le nom des entrées et sorties TOR et analogiques fournies par ce modèle de CPU, des informations d'adressage pour les E/S, des indicateurs d'état, des numéros d'emplacement et des commentaires.

Emplacement	Etat	Nom	Numéro de référence	Adresse d'entrée	Adresse de sortie	Commentaire
1.1		DI14/DQ10_1 Détails	6ES7 214-1AG40-0XB0	0	0	
1.2		AI2_1 Détails	6ES7 214-1AG40-0XB0	64	---	
1.16		HSC_1 Détails	6ES7 214-1AG40-0XB0	1000	---	
1.17		HSC_2 Détails	6ES7 214-1AG40-0XB0	1004	---	
1.18		HSC_3 Détails	6ES7 214-1AG40-0XB0	1008	---	
1.19		HSC_4 Détails	6ES7 214-1AG40-0XB0	1012	---	
1.20		HSC_5 Détails	6ES7 214-1AG40-0XB0	1016	---	
1.21		HSC_6 Détails	6ES7 214-1AG40-0XB0	1020	---	
1.32		Pulse_1 Détails	6ES7 214-1AG40-0XB0	---	1000	

La page d'information sur les modules montre le chemin que vous avez suivi au fur et à mesure que vous zoomez. Vous pouvez cliquer sur n'importe quel lien dans ce chemin pour revenir à un niveau supérieur.



Informations modules : onglet Identification

L'onglet Identification affiche les informations d'identification et de maintenance (I&M) du module sélectionné.

The screenshot shows the Siemens S7-1200 station web interface. The top bar includes the Siemens logo, the station name 'S7-1200 station_1 / PLC_1', the date and time '20 28 06 26.04 2016', the time zone 'UTC', and the language 'Français'. The user is logged in as 'admin' with a 'Déconnexion' link. The main content area is titled 'Etat du module' and contains a table of modules. Below the table, there are tabs for 'État', 'Identification', and 'Firmware'. The 'Identification' tab is active, showing the following details:

Emplacement	Etat	Nom	Numéro de référence	Adresse d'entrée	Adresse de sortie
1	✓	PLC_1	Détails 6ES7 215-1AG40-0XB0		
2	✓	DI 16/DQ 16x24VDC_1	Détails 6ES7 223-1BL32-0XB0	8	8

Below the table, the 'Identification' tab shows the following information:

- Constructeur : Siemens
- Version du firmware : V4.2
- Classe d'appareil : CPU 1215C DCDCDC
- Repère d'installation :
- Repère d'emplacement :
- Date d'installation : 2016-04-25 19:48
- Description :

Notez que si vous cliquez sur un module d'entrées-sorties F dans la partie supérieure de l'écran, la partie inférieure affiche un onglet Sécurité. Dans cet onglet, vous pouvez voir des données spécifiques relatives au module sélectionné comme décrit dans Manuel de sécurité fonctionnelle du S7-1200

(<https://support.industry.siemens.com/cs/ww/fr/view/104547552/en>).

Informations modules : onglet Firmware

L'onglet firmware de la page d'informations sur les modules affiche des informations sur le firmware du module sélectionné. Vous pouvez également effectuer une mise à jour du firmware de la CPU ou de tout autre module dans le châssis local qui prend en charge la fonction mise à jour du firmware si vous possédez le privilège (Page 1042) d'"effectuer une mise à jour du firmware". Pour les modules distants, vous pouvez voir les informations sur le firmware mais vous ne pouvez pas effectuer une mise à jour du firmware.

Remarque

En ce qui concerne la mise à jour du firmware, vous pouvez uniquement mettre à jour les CPU S7-1200 de version 3.0 ou plus.



Mise à jour du firmware

La CPU doit être à l'état ARRET lorsque vous actualisez le firmware. Lorsque la CPU est à l'état ARRET, cliquez sur le bouton Parcourir pour naviguer jusqu'au fichier et sélectionner celui-ci, à partir duquel vous pouvez charger la mise à jour de la version du firmware. Les mises à jour du firmware sont disponibles sur le site Web Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/fr>).

Pendant la mise à jour, la page affiche un message signalant que la mise à jour est en cours. Une fois la mise à jour terminée, la page affiche la référence et le numéro de version du firmware mis à jour. Si vous mettez à jour le firmware de la CPU ou d'un Signal Board, le serveur Web redémarre la CPU.

Vous pouvez également mettre à jour le firmware au moyen de l'une des autres méthodes suivantes :

- Utilisation des Outils en ligne et de diagnostic de STEP 7 (Page 1389)
- Utilisation d'une Carte mémoire SIMATIC (Page 153)
- Utilisation de SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/fr/view/98161300>)

Remarque

Problèmes potentiels durant une mise à jour du firmware à partir du serveur Web

En cas de perturbation de la communication pendant une mise à jour du firmware à partir du serveur Web, votre navigateur Web peut afficher un message demandant si vous voulez quitter la page actuelle ou rester sur cette page. Pour éviter d'éventuels problèmes, sélectionnez l'option Rester sur la page actuelle.

Si vous fermez le navigateur Web durant la mise à jour du firmware depuis le serveur Web, vous ne serez plus en mesure de faire passer l'état de fonctionnement de la CPU à MARCHE. Si cette situation se produit, vous devez couper/rétablir l'alimentation électrique de la CPU pour pouvoir faire passer son état de fonctionnement à MARCHE.

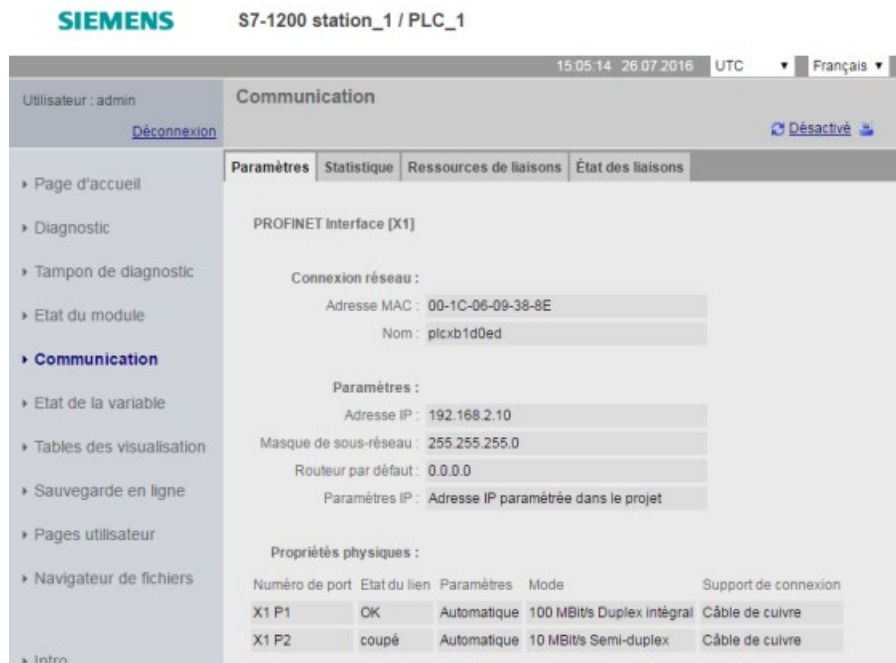
12.6.9 Communication

La page Communication affiche les paramètres de la CPU connectée, les statistiques de communication, les ressources et des informations sur les liaisons.

L'affichage de la page Communication nécessite les droits "Interroger le diagnostic".

Onglet Paramètres

L'onglet Paramètres affiche l'adresse MAC de la CPU, l'adresse IP et le paramétrage IP de la CPU, ainsi que les propriétés physiques :



Onglet Statistiques

L'onglet Statistiques affiche les statistiques de communication pour l'émission et la réception :

SIEMENS S7-1200 station_1 / PLC_1

20:49:52 27.07.2016 UTC Français

Utilisateur : admin [Déconnexion](#) [Désactivé](#)

Communication

Paramètres **Statistique** Ressources de liaisons État des liaisons

Page d'accueil
Diagnostic
Tampon de diagnostic
Etat du module
Communication
Etat de la variable
Tables des visualisation
Sauvegarde en ligne
Pages utilisateur
Navigateur de fichiers
Intro

Statistiques totales

Paquets de données émis :
Emis sans erreur : 325339112 octets
Collisions à la tentative d'émission : 0
Annulés après une autre erreur : 0

Paquets de données reçus :
Reçus sans erreur : 388087772 octets
Rejetés pour cause d'erreurs : 0
Rejetés car ressources insuffisantes : 0

X1 P1

Paquets de données émis :
Emis sans erreur : 325339112 octets
Collisions à la tentative d'émission : 0
Annulés après une autre erreur : 0

Paquets de données reçus :
Reçus sans erreur : 388087772 octets
Rejetés pour cause d'erreurs : 0
Rejetés car ressources insuffisantes : 0

X1 P2

Paquets de données émis :
Emis sans erreur : 0 octets
Collisions à la tentative d'émission : 0
Annulés après une autre erreur : 0

Paquets de données reçus :
Reçus sans erreur : 0 octets
Rejetés pour cause d'erreurs : 0
Rejetés car ressources insuffisantes : 0

Onglet Ressources de liaison

L'onglet Ressources de liaison affiche des informations sur le nombre total de ressources de liaison et sur la manière dont elles sont allouées aux différents types de communication :

The screenshot shows the Siemens S7-1200 station web interface. The title bar includes the Siemens logo, the station name 'S7-1200 station_1 / PLC_1', the time '20:58:38', the date '27.07.2016', the time zone 'UTC', and the language 'Français'. The user is identified as 'admin' with a 'Déconnexion' link. The main menu includes 'Paramètres', 'Statistique', 'Ressources de liaisons' (selected), and 'État des liaisons'. A 'Désactivé' button is also visible.

Navigation menu items:

- Page d'accueil
- Diagnostic
- Tampon de diagnostic
- Etat du module
- Communication**
- Etat de la variable
- Tables des visualisation
- Sauvegarde en ligne
- Pages utilisateur
- Navigateur de fichiers
- Intro

Summary of link resources:

Nombre de liaisons :

- Liaisons maximales : 128
- Liaisons non occupées : 126

Table of link reservations:

Liaisons :	réservee(s)	occupée(s)
Communication ES	4	1
Communication HMI	12	0
communication S7	8	0
Communication OpenUser	8	0
communication Web	0	1
Autre communication	---	0

Onglet État des liaisons

L'onglet État des liaisons affiche les liaisons pour la CPU, ainsi que des détails sur la liaison sélectionnée.

SIEMENS S7-1200 station_1 / PLC_1

20:19:13 27.07.2016 UTC Français

Utilisateur : admin [Déconnexion](#) [Désactivé](#)

Communication

Paramètres	Statistique	Ressources de liaisons		État des liaisons			
Etat	ID local (Hex)	Port de passerelle	Type d'adresse distante	Adresse distante	Type	Type	
✓ Connexion établie	0	1 (PLC_1)	IPv4	192.168.2.250	ES	Adhoc	
✓ Connexion établie	0	1 (PLC_1)	IPv4	192.168.2.250	WEB	Adhoc	
✓ Connexion établie	0	1 (PLC_1)	IPv4	192.168.2.250	WEB	Adhoc	
✓ Connexion établie	0	1 (PLC_1)	IPv4	192.168.2.250	WEB	Adhoc	

Details:

Détails de l'adresse

Adresse locale : 192.168.2.10

TSAP local (hexadécimal) : 53 49 4D 41 54 49 43 2D 52 4F 4F 54 2D 45 53

TSAP local (ASCII) : SIMATIC-ROOT-ES

Adresse distante : 192.168.2.250

TSAP distant (hexadécimal) : 06 00

Statistique

Tentatives d'établissement de liaison en cours : 0

Tentatives d'établissement de liaison réussies : 1

octets envoyés : 27391

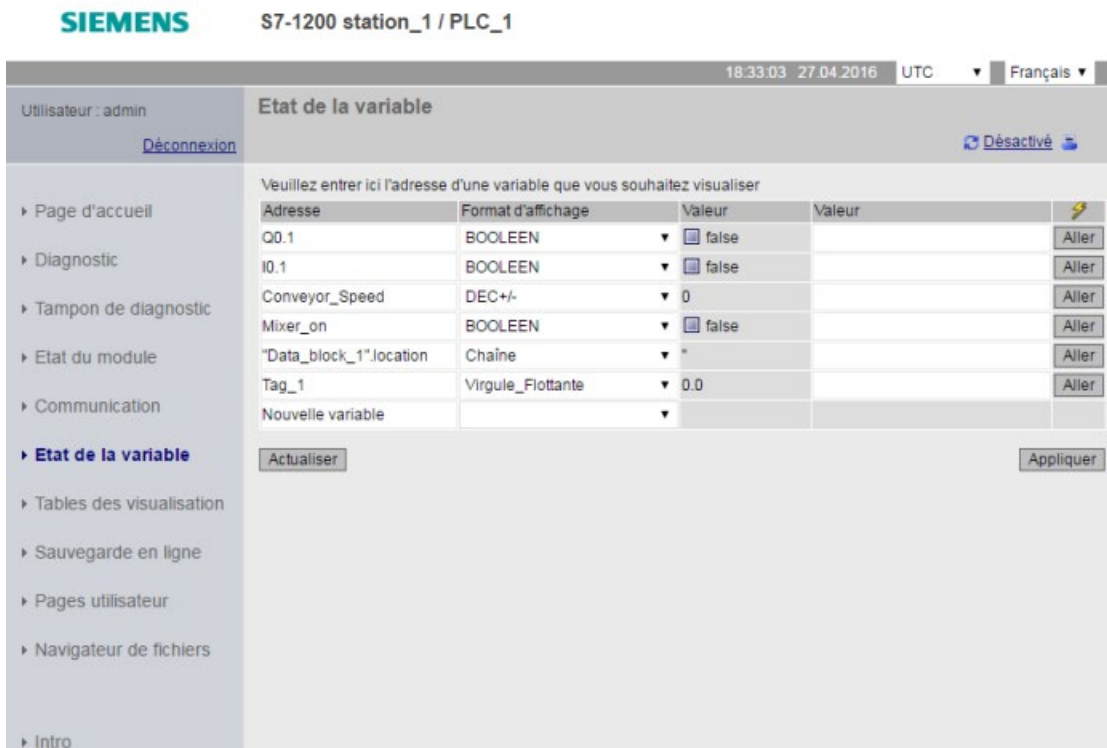
octets reçus : 7058

12.6.10 État des variables

La page État des variables vous permet de visualiser n'importe quelle donnée d' E/S ou de mémoire dans votre CPU. Vous pouvez entrer une adresse directe (comme %I0.0), un nom de variable API ou une variable d'un bloc de données spécifique. Pour les variables de bloc de données, vous devez mettre le nom du bloc de données entre guillemets. Vous pouvez sélectionner un format d'affichage des données pour chaque valeur à visualiser. Vous pouvez continuer à entrer et à spécifier toutes les valeurs que vous voulez dans les limites de la page. Les valeurs à visualiser s'affichent automatiquement. Vous pouvez cliquer sur le bouton "Rafraîchir" à tout moment pour rafraîchir toutes les valeurs visualisées. Si vous avez activé l'actualisation automatique dans STEP 7 (Page 1040), vous pouvez cliquer sur le bouton "Off" en haut à droite de la page pour la désactiver. Lorsque l'actualisation automatique est désactivée, vous pouvez cliquer sur "On" pour la réactiver.

Vous devez posséder les droits "Lire l'état des variables" pour afficher la page État des variables.

Si vous avez ouvert une session en tant qu'utilisateur possédant les droits (Page 1050) "Écrire l'état des variables", vous pouvez également forcer les valeurs de données. Entrez les valeurs que vous voulez donner dans le champ "Valeur de forçage" approprié. Cliquez sur le bouton "Aller" à côté d'une valeur pour écrire cette valeur dans la CPU. Vous pouvez également saisir plusieurs valeurs et cliquer sur "Appliquer" pour écrire toutes les valeurs dans la CPU. Les boutons et les titres de colonnes pour le forçage ne sont visibles que si vous possédez les droits "Écrire l'état des variables".



La page État des variables ne conserve pas vos entrées si vous y revenez après l'avoir quittée. Vous pouvez marquer la page d'un signet et revenir au signet pour voir les mêmes entrées. Si vous ne marquez pas la page d'un signet, vous devez ressaisir les variables.

Envisagez plutôt d'utiliser une table de visualisation (Page 1069) pour les valeurs que vous visualisez et forcez fréquemment.

Remarque

Tenez compte des particularités suivantes lors de l'utilisation de la page État des variables standard :

- Ecrivez toutes les modifications apportées aux chaînes entre apostrophes.
 - La page État des variables permet de visualiser et forcer des variables contenant les caractères suivants : &, <, (, +, ,(virgule), ., [,], \$ ou %, à condition que vous écriviez le nom de variable entre guillemets, par exemple "Cadence_2,5Hz".
 - Pour visualiser ou modifier un seul champ d'une variable DTL, insérez le champ dans l'adresse, par exemple, "Data_block_1".DTL_tag.Year. Saisissez une valeur entière pour la valeur de forçage conformément au type de données du champ spécifique du DTL. Par exemple, le champ Year est un UInt.
 - Le nombre maximum d'entrées de variable par page est de 50.
 - Si un nom de variable comprend des caractères spéciaux de sorte qu'il est refusé comme entrée dans la page État des variables, vous pouvez l'écrire entre guillemets. Dans la plupart des cas, la page reconnaîtra ensuite le nom de variable.
-

Voir aussi

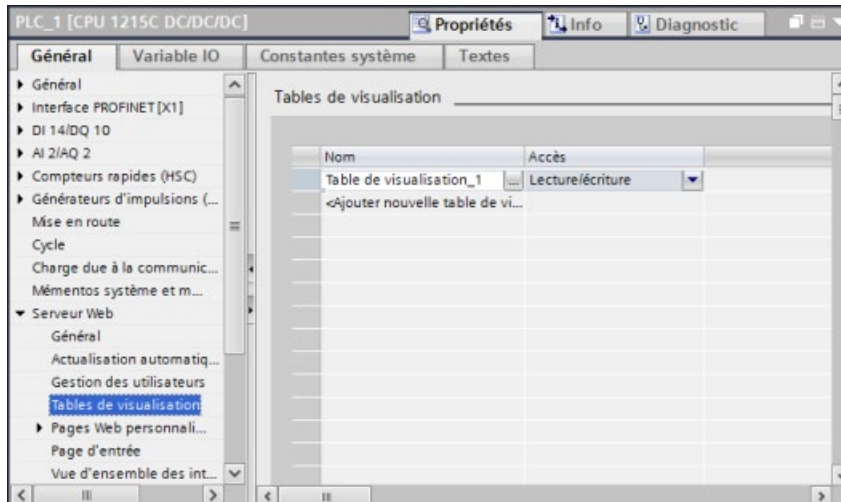
Règles de saisie des noms de variable et des valeurs (Page 1126)

12.6.11 Tables de visualisation

Le serveur Web vous permet d'accéder aux tables de visualisation que vous avez configurées dans STEP 7 et que vous avez chargées dans la CPU. Les tables de visualisation avec 50 entrées ou moins offrent les meilleures performances sur le serveur Internet.

Configuration STEP 7 pour sélectionner des tables de visualisation pour le serveur Web

Dans la configuration d'appareil de la CPU dans STEP 7, vous pouvez ajouter les tables de visualisation que le serveur Web doit afficher. Vous définissez des droits en lecture ou en lecture/écriture pour chaque table de visualisation que vous sélectionnez dans la liste des tables existantes. Une fois qu'elles sont chargées dans la CPU, vous pouvez uniquement visualiser les tables de visualisation pour lesquelles vous avez sélectionné les droits de lecture, mais vous pouvez visualiser et modifier les variables des tables de visualisation pour lesquelles vous avez sélectionné les droits de lecture/écriture.



Lorsque vous avez achevé la configuration des tables de visualisation dans la section Serveur Web de la configuration d'appareil, chargez votre configuration matérielle dans la CPU.

Visualisation des tables de visualisation dans le serveur Web

Si vous avez les droits (Page 1042) "Lire les variables", vous pouvez sélectionner Tables de visualisation dans le menu de navigation du serveur Web pour accéder aux tables de visualisation que vous avez configurées et chargées dans la CPU. Si vous avez chargé plus d'une table de visualisation, vous pouvez sélectionner celle à afficher dans la liste déroulante. Le serveur Web affiche la table de visualisation que vous avez créée dans STEP 7 et les valeurs en cours conformément au format d'affichage. Vous pouvez modifier le format d'affichage si vous le souhaitez mais, lorsque vous reviendrez à la page des tables de visualisation, le serveur Web reprendra les formats d'affichage par défaut dans la table de visualisation STEP 7.

Modification des variables des tables de visualisation dans le serveur Web

Si vous avez chargé une table de visualisation avec le niveau d'accès "lecture/écriture" et que vous avez ouvert une session Serveur Web avec les droits (Page 1042) "Écrire les variables", vous pouvez également modifier les valeurs des variables tout comme vous le faites dans une table de visualisation dans STEP 7. Vous pouvez modifier des valeurs de variables individuelles et cliquer sur "Aller" pour modifier uniquement la valeur concernée ou vous pouvez saisir plusieurs valeurs et cliquer sur "Appliquer" pour modifier toutes ces valeurs en même temps.

The screenshot shows the Siemens S7-1200 station_1 / PLC_1 web interface. The top bar includes the Siemens logo, the station name, and the date/time (18.10.12 25.08.2015) along with UTC and Français language settings. The user is identified as 'admin' with a 'Déconnexion' link. The main content area is titled 'Tables des visualisation' and shows a dropdown menu for 'Table de visualisation_1' and a 'Désactivé' button. Below this is a table with the following data:

Nom	Adresse	Format d'affichage	Valeur	Valeur	Commentaire
"Data_block_1".Location		Chaîne	▼ East		Aller
"Data_block_1".ManualOverrideEnable		BOOLEEN	▼ <input type="checkbox"/> false		Aller
"Data_block_1".TurbineNumber		DEC+/-	▼ 2		Aller
"Data_block_1".WindSpeed		Virgule_Flottante	▼ 17.5		Aller
"Conveyor_speed"	%MW102	DEC+/-	▼ 0		Aller

At the bottom of the table area, there are 'Actualiser' and 'Appliquer' buttons. A left sidebar contains navigation links such as 'Page d'accueil', 'Diagnostic', 'Etat du module', 'Communication', 'Etat de la variable', 'Tables des visualisation' (highlighted), 'Sauvegarde en ligne', 'Pages utilisateur', and 'Navigateur de fichiers'.

Remarque

Avantages des tables de visualisation pour la modification de variables

Pour qu'un utilisateur puisse modifier des variables et des variables de blocs de données dans la CPU à partir d'une table de visualisation, vous devez configurer la table de visualisation dans les propriétés Serveur Web de la configuration d'appareil STEP 7 et vous devez accorder l'accès en lecture/écriture. Vous pouvez ainsi restreindre les variables que peut modifier un utilisateur ayant les droits "Écrire les variables" aux seules variables figurant dans les tables de visualisation de serveur Web configurées.

Dans la CPU en revanche, sur la page "État des variables (Page 1068)", tout utilisateur avec les droits "Écrire l'état des variables" peut écrire dans n'importe quelle variable ou variable de bloc de données.

Vous pouvez, par une configuration prudente des droits utilisateur (Page 1042) sur le serveur Web, contribuer à sécuriser l'accès à vos données API.

Voir aussi

Règles de saisie des noms de variable et des valeurs (Page 1126)

12.6.12 Sauvegarde en ligne

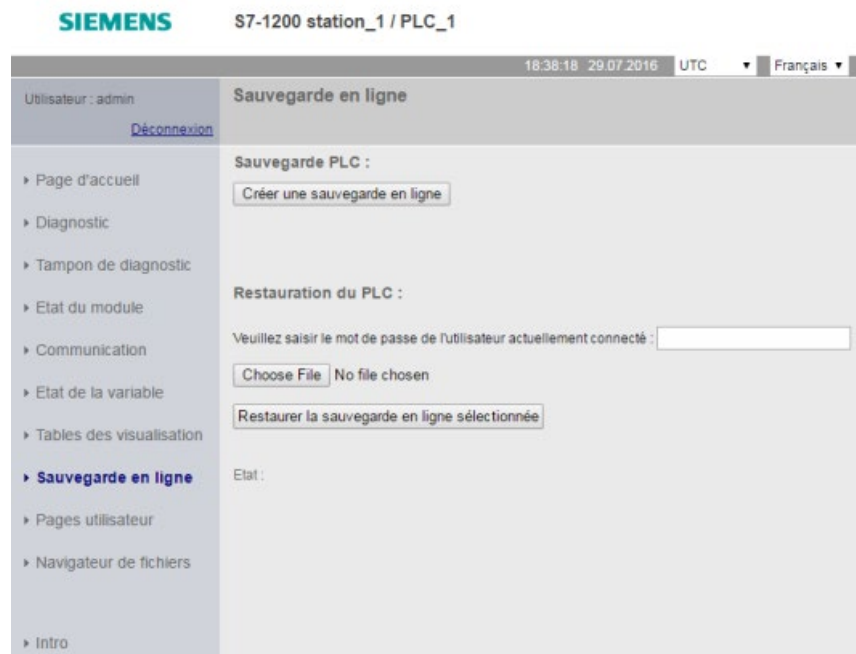
La page Web standard Sauvegarde en ligne permet d'exécuter une sauvegarde du projet STEP 7 pour l'API en ligne ainsi que de restaurer une sauvegarde de l'API exécutée auparavant. Avant de créer ou de restaurer une sauvegarde, faites passer l'API à l'état ARRÊT et cessez toute communication avec l'API, tels l'accès IHM et l'accès Serveur Web. Si votre CPU n'est pas à l'état ARRÊT, les fonctions de sauvegarde et de restauration vous demanderont de confirmer le passage à l'état ARRÊT avant de poursuivre.

Si vous avez accédé à la page Sauvegarde en ligne par l'un des modules CP à fonction Web, vous pouvez effectuer une sauvegarde mais vous ne pouvez pas procéder à une restauration.

Remarque

Vous pouvez également exécuter des opérations de sauvegarde et de restauration à partir de STEP 7 (Page 1423). Reportez-vous à ces rubriques pour une description complète des données que vous pouvez sauvegarder et restaurer. L'outil SIMATIC Automation Tool (SAT) fournit également des options de sauvegarde et de restauration.

Lorsque vous sauvegardez des fichiers à partir du serveur Web, votre PC ou votre appareil enregistre ces fichiers dans le dossier pris par défaut pour les téléchargements. Lorsque vous sauvegardez des fichiers à partir de STEP 7, STEP 7 enregistre ces fichiers à l'intérieur du projet STEP 7. Vous ne pouvez pas restaurer de fichiers de sauvegarde STEP 7 dans le serveur Web et vous ne pouvez pas restaurer de fichiers de sauvegarde Serveur Web dans STEP 7. Toutefois, vous pouvez enregistrer les fichiers de sauvegarde STEP 7 directement dans le dossier de téléchargement sur votre PC ou votre appareil. Dans ce cas, vous pourrez restaurer ces fichiers dans le serveur Web.



Sauvegarde de l'API

Dans la section Sauvegarde API de la page, cliquez sur le bouton "Créer sauvegarde en ligne" pour effectuer une sauvegarde du projet actuellement enregistré dans l'API. Cette fonction nécessite les droits (Page 1042) utilisateur "Effectuer une copie de sauvegarde de la CPU". Si la CPU est à l'état MARCHE et que vous devez la faire passer à l'état ARRÊT, vous devez également disposer des droits "Changer le mode de fonctionnement". Le PC ou l'appareil enregistrera le fichier de sauvegarde à l'emplacement par défaut pour les téléchargements. Selon le paramétrage de votre navigateur et de votre appareil, vous recevrez peut-être un message de confirmation pour l'enregistrement du fichier.

Restauration de l'API

Dans la section Restaurer l'API de la page, saisissez le mot de passe utilisateur du serveur Web et cliquez sur le bouton "Parcourir" ou "Choisir fichier" (selon votre navigateur) pour sélectionner un fichier de sauvegarde enregistré préalablement. Cliquez sur le bouton "Charger la sauvegarde en ligne" et confirmez le message qui s'affiche afin de charger le fichier dans l'API connecté. Cette page nécessite les droits (Page 1042) utilisateur "Restaurer la CPU". Si la CPU est à l'état MARCHE et que vous devez la faire passer à l'état ARRÊT, vous devez également disposer des droits "Changer le mode de fonctionnement".

Au fur et à mesure de la restauration, des messages s'affichent pour vous informer sur l'avancement de l'opération et vous devez saisir à nouveau votre identifiant et votre mot de passe. Les indicateurs d'achèvement suivants s'affichent après chaque étape du processus menée à bien, ainsi qu'un lien pour le rechargement de la page :

Etat :

- Le téléchargement de la sauvegarde en ligne a démarré.
- Vérification du fichier de sauvegarde.
- Formatage de la carte mémoire et réinitialisation de la CPU.
- Chargement de la configuration.
- Réinitialisation de la CPU.

Le téléchargement pour le dispositif a été effectué sans erreur. [Rechargement de la page...](#)

ATTENTION

Restauration de sauvegardes au contenu inconnu

La restauration d'une sauvegarde au contenu inconnu peut provoquer des dégâts matériels importants ou des blessures graves en présence de dysfonctionnements ou d'erreurs dans le programme.

En outre, si vous restaurez une sauvegarde dans laquelle le serveur Web n'est pas activé dans la configuration d'appareil de la CPU, vous ne serez pas en mesure d'accéder à la CPU à partir du serveur Web.

Assurez-vous que la sauvegarde correspond à une configuration dont le contenu est connu.

Remarque

Restauration d'une sauvegarde dans laquelle l'adresse IP de la CPU est différente

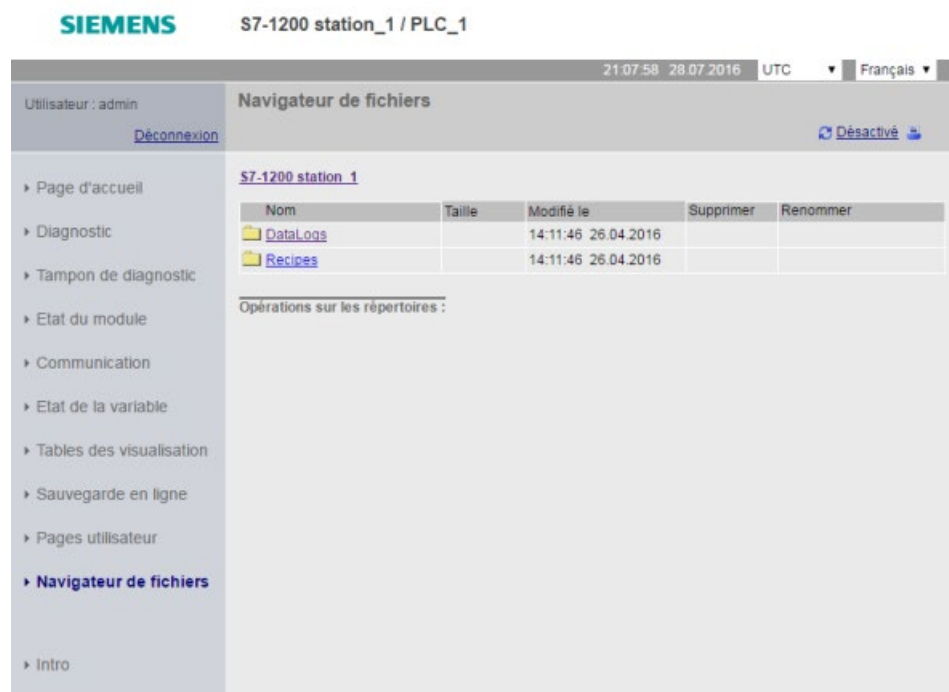
Si vous tentez de restaurer une sauvegarde dans laquelle l'adresse IP de la CPU est différente de l'adresse IP de la CPU en cours, le serveur Web ne peut pas afficher le message signalant que la restauration est achevée. Lorsque plus de cinq minutes se seront écoulées après l'apparition du message de réinitialisation de la CPU, saisissez la nouvelle adresse IP correspondant à l'adresse dans le fichier de sauvegarde. La CPU aura alors cette adresse et vous pourrez à nouveau accéder au serveur Web.

12.6.13 Navigateur de fichiers

La page du navigateur de fichiers File Browser fournit l'accès aux fichiers de la mémoire de chargement interne de la CPU ou de la carte mémoire (mémoire de chargement externe). La page du navigateur de fichiers affiche initialement le dossier racine de la mémoire de chargement, qui contient les dossiers "Journaux" et "Recettes", mais il affiche aussi tout autre dossier que vous avez éventuellement créé en cas d'utilisation d'une carte mémoire.

Le type d'accès aux fichiers et aux dossiers dont vous disposez dépend de vos privilèges (Page 1042) utilisateur. Tout utilisateur disposant des privilèges de "lecture de fichiers" peut visualiser des fichiers et des dossiers à l'aide du navigateur de fichiers. Quels que soient vos privilèges d'accès, vous ne pouvez pas supprimer le dossier Journaux ou Recettes ; toutefois, si vous avez créé des dossiers utilisateur sur la carte mémoire et que vous avez ouvert une session en tant qu'utilisateur avec des privilèges d'"écriture/suppression de fichiers", vous pouvez supprimer ces fichiers.

Cliquez sur un dossier pour accéder aux différents fichiers qu'il contient.



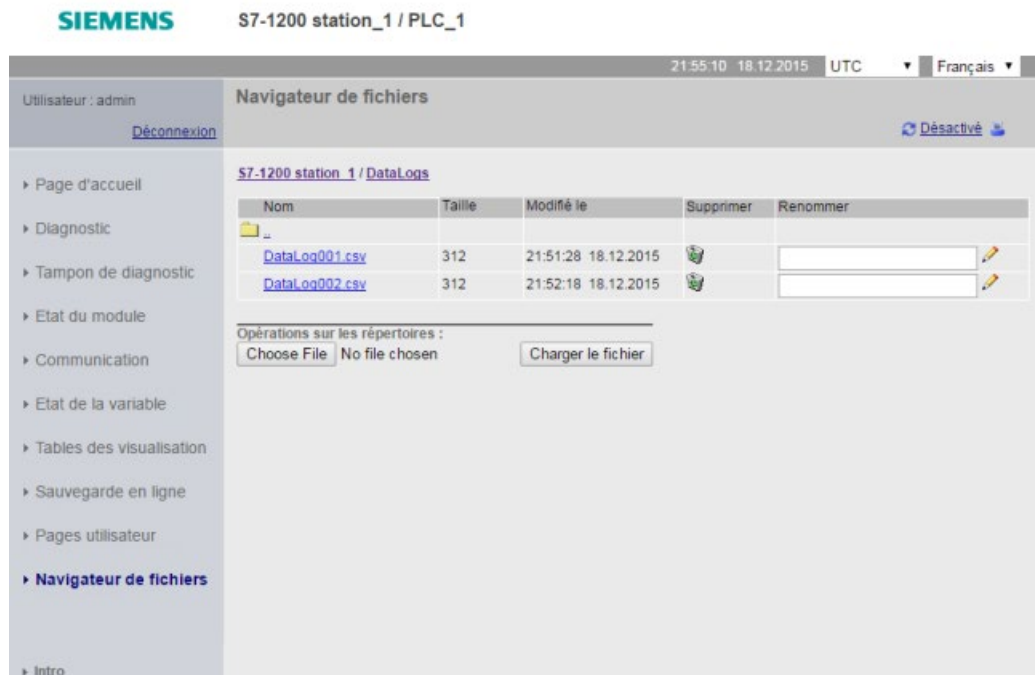
Journaux de données

Vous pouvez ouvrir chaque fichier journal à partir du dossier "Journaux". Si vous avez ouvert une session avec le privilège (Page 1042) d'"écriture/suppression de fichiers", vous pouvez également effacer, renommer et télécharger ces fichiers. Les fichiers journaux sont au format CSV (valeurs séparées par des virgules). Vous pouvez les enregistrer sur votre ordinateur ou les ouvrir dans Microsoft Excel (par défaut) ou un autre programme.

Remarque

Horodatages pour les journaux

Le serveur Web affiche les horodatages pour les journaux en heure UTC ou en heure locale API, en fonction de votre sélection tout en haut de la page.



Re- Les options "Delete" (supprimer) et "Rename" (renommer) ne sont pas disponibles si vous **marq** n'avez pas ouvert de session avec le privilège d'"écriture/suppression de fichiers".
ue :

Remarque**Gestion des journaux de données**

Ne conservez pas plus de 1 000 journaux de données dans un système de fichiers. Dépasser ce nombre peut empêcher le serveur web d'avoir suffisamment de ressources CPU pour afficher les journaux de données.

Si vous constatez que la page web du navigateur de fichiers n'est pas en mesure d'afficher les journaux de données, alors vous devez mettre la CPU à l'état ARRET afin d'afficher et de supprimer des journaux de données.

Gérez vos journaux de données pour vous assurer que vous en conservez uniquement le nombre que vous devez respecter, et ne pas dépasser 1 000 journaux de données.

Travailler avec un journal de données dans Excel

Le fichier journal est en format CSV anglais/américain. Pour l'ouvrir dans Excel sur des systèmes non anglais/américains, vous devez l'y importer avec des paramètres spécifiques (Page 1128).

Fichiers de recette

A l'image du dossier des journaux, le dossier des recettes affiche tout fichier de recette présents dans la mémoire de chargement. Les fichiers de recette sont également au format CSV et vous pouvez les ouvrir dans Microsoft Excel ou dans un autre programme. Comme dans le cas des journaux, vous devez posséder des privilèges de modification pour supprimer, modifier et enregistrer, renommer ou télécharger des fichiers de recette.

Téléchargement des fichiers et rafraîchissement automatique de la page

Si vous lancez le téléchargement d'un fichier, l'opération continue tant que vous restez sur la page web du navigateur de fichiers. Si vous activez la mise à jour automatique pour rafraîchir les pages du serveur web toutes les dix secondes, vous voyez la progression du téléchargement à chaque fois de la page se rafraîchit. Par exemple, si vous téléchargez un fichier de 2 Mo, vous pouvez voir la mise à jour qui montre la taille du fichier en octets à 2 500, 5 000, 10 000, 15 000 et 20 000 lors de la progression du chargement.

Si vous quittez la page du navigateur de fichiers pour afficher une autre page avant la fin du chargement, le serveur Web supprime le fichier incomplet.

Informations supplémentaires

Remarque

Conventions pour les noms de fichier

Pour que le serveur Web puisse opérer sur les fichiers journaux et de recette, les noms des fichiers doivent contenir des caractères du jeu ASCII à l'exception des caractères \ / : * ? " < > | et de l'espace.

Si les noms de vos fichiers ne sont pas conformes à cette règle, le serveur Web peut être confronté à des erreurs lors d'opérations telles que charger, supprimer ou renommer des fichiers. Dans ce cas, vous devrez peut-être utiliser un lecteur de carte et l'explorateur de fichiers Windows pour renommer les fichiers qui se trouvaient en mémoire de chargement externe.

Reportez-vous au chapitre Recettes et journaux (Page 512) pour plus d'informations sur la programmation avec les instructions de journal, ainsi que l'importation (Page 519) et l'exportation (Page 517) de recettes.

12.7 Pages Web personnalisées

Le serveur Web du S7-1200 vous offre la possibilité de créer vos propres pages HTML propres à une application qui incorporent les données provenant de l'API.

ATTENTION

Accès de personnes non autorisées à la CPU via les pages web personnalisées

L'accès de personnes non autorisées à la CPU via les pages web personnalisées peut perturber le fonctionnement du processus, ce qui peut entraîner la mort, des blessures graves et des dommages matériels importants.

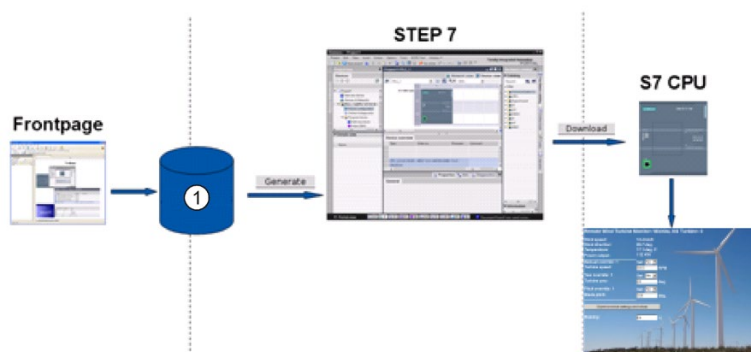
Le codage non sécurisé des pages web personnalisées crée des failles comme le cross-site scripting (XSS), l'injection de codes et autres.

Protégez votre CPU S7-1200 contre un accès non autorisé en l'installant de manière sécurisée comme précisé dans les recommandations d'opération qui se trouvent sur le Site Internet Industrial Security (<http://www.siemens.com/industrialsecurity>).

Vous créez les pages personnalisées à l'aide de l'éditeur HTML de votre choix et vous les téléchargez dans la CPU où elles sont accessibles dans le menu des pages Web standard. Ce processus implique plusieurs tâches :

- Créer des pages HTML avec un éditeur HTML, par exemple Microsoft Frontpage (Page 1080)
- Inclure des commandes AWP dans des commentaires HTML dans le code HTML (Page 1081) : Les commandes AWP constituent un jeu fixe de commandes que Siemens fournit pour accéder aux informations de la CPU.
- Configurer STEP 7 pour lire et traiter les pages HTML (Page 1097)
- Générer les blocs à partir des pages HTML (Page 1097)
- Programmer STEP 7 pour commander l'utilisation des pages HTML (Page 1099)
- Compiler et charger les blocs dans la CPU (Page 1101)
- Accéder aux pages Web personnalisées depuis votre PC (Page 1101)

Ce processus est illustré ci-dessous :



① Fichiers HTML avec commandes AWP intégrées

12.7.1 Création de pages HTML

Vous pouvez utiliser le progiciel de votre choix pour créer vos propres pages HTML à utiliser avec le serveur Web. Assurez-vous que votre code HTML est conforme aux standards HTML du consortium W3C (World Wide Web Consortium). STEP 7 ne procède à aucune vérification de votre syntaxe HTML.

Vous pouvez utiliser un progiciel qui permet la conception en mode WYSIWYG ou mise en page, mais vous devez pouvoir éditer votre code HTML en HTML pur. La plupart des outils de création Web fournissent ce type d'édition ; sinon, vous pouvez toujours utiliser un éditeur de texte simple pour éditer le code HTML. Insérez la ligne suivante dans votre page HTML pour définir le jeu de caractères de la page à UTF-8 :

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Veillez également à enregistrer le fichier en codage de caractères UTF-8 depuis l'éditeur.

Vous utilisez STEP 7 pour compiler tout le contenu de vos pages HTML en blocs de données STEP 7. Ces blocs de données consistent en un bloc de données de commande qui pilote l'affichage des pages Web et en un ou plusieurs blocs de données de fragment qui contiennent les pages Web compilées. Soyez conscient que les jeux importants de pages HTML, en particulier celles avec de nombreuses images, nécessitent une quantité significative d'espace en mémoire de chargement (Page 1102) pour les DB de fragment. Si la mémoire de chargement interne de votre CPU n'est pas suffisante pour vos pages Web personnalisées, utilisez une carte mémoire (Page 143) pour fournir une mémoire de chargement externe.

Pour que votre code HTML utilise des données dans le S7-1200, vous incluez des commandes AWP (Page 1081) sous forme de commentaires HTML. Lorsque vous avez terminé, sauvegardez vos pages HTML sur votre PC et notez le chemin du dossier où vous les avez sauvegardées.

Remarque

La taille des fichiers HTML contenant des commandes AWP est limitée à 64 Ko. Veillez à ce que la taille de vos fichiers reste en dessous de cette limite afin que STEP 7 soit dans la mesure de compiler vos pages.

Actualisation de pages Web personnalisées

Les pages Web personnalisées ne sont pas actualisées automatiquement. C'est à vous de choisir si votre programme HTML doit actualiser la page ou non. Le rafraîchissement périodique des pages qui affichent des données de l'API permet d'avoir des données toujours actuelles. Dans les pages HTML servant de formulaires pour la saisie de données, l'actualisation peut interférer avec la saisie des données par l'utilisateur. Si vous voulez que votre page entière soit actualisée automatiquement, vous pouvez ajouter cette ligne dans votre en-tête HTML, "10" étant le nombre de secondes entre les rafraîchissements :

```
<meta http-equiv="Refresh" content="10">
```

Vous pouvez aussi utiliser JavaScript ou d'autres techniques HTML pour gérer l'actualisation de la page ou des données. Consultez la documentation HTML et JavaScript à ce sujet.

12.7.2 Commandes AWP prises en charge par le serveur Web du S7-1200

Le serveur Web S7-1200 fournit des commandes AWP que vous intégrez dans vos pages Web personnalisées sous forme de commentaires HTML aux fins suivantes :

- Lecture de variables (Page 1082)
- Ecriture de variables (Page 1083)
- Lecture de variables spéciales (Page 1085)
- Ecriture de variables spéciales (Page 1087)
- Définition de types Enum (Page 1089)
- Affectation de variables aux types Enum (Page 1090)
- Création de blocs de données de fragment (Page 1091)

Syntaxe générale

Mis à part la commande de lecture d'une variable, les commandes AWP ont la syntaxe suivante :

```
<!-- AWP_ <nom et paramètres de la commande> -->
```

Vous utilisez les commandes AWP conjointement avec des commandes de formulaire HTML typiques pour écrire des variables dans la CPU.

Les descriptions des commandes AWP dans les pages suivantes utilisent les conventions suivantes :

- Les éléments entre crochets [] sont facultatifs.
- Les éléments entre chevrons < > sont des valeurs de paramètres à préciser.
- Les apostrophes et guillemets font partie intégrante de la commande. Ils doivent être présents comme indiqués.
- Les caractères spéciaux dans les noms de variables ou de blocs de données doivent être précédés d'un caractère d'échappement ou être entourés d'apostrophes ou de guillemets (Page 1094) selon l'usage.

Utilisez un éditeur de texte ou le mode d'édition HTML pour insérer des commandes AWP dans vos pages.

Remarque

Règles de syntaxe des commandes AWP

L'espace après "<!--" et l'espace avant "-->" dans la formulation d'une commande AWP sont indispensables à la compilation correcte de la commande. En cas d'omission de ces espaces, le compilateur peut être incapable de générer le code correct. Le compilateur n'affiche pas d'erreur dans ce cas.

Récapitulatif des commandes AWP

Les détails d'utilisation de chaque commande AWP sont précisés dans les paragraphes suivants, mais voici un bref récapitulatif des commandes :

Lecture de variables

```
:=<NomVar>:
```

Écriture de variables

```
<!-- AWP_In_Variable Name='<NomVar1>' [Use='<NomVar2>'] ... -->
```

Cette commande AWP ne fait que déclarer que la variable dans la clause Name est inscriptible. C'est votre code HTML qui exécute des écritures dans la variable en indiquant son nom dans des instructions HTML <input>, <select> ou autres à l'intérieur d'un formulaire HTML.

Lecture de variables spéciales

```
<!-- AWP_Out_Variable Name='<Type>:<Nom>' [Use='<NomVar>'] -->
```

Écriture de variables spéciales

```
<!-- AWP_In_Variable Name='<Type>:<Nom>' [Use='<NomVar>'] -->
```

Définition de types Enum

```
<!-- AWP_Enum_Def Name='<nom du type Enum>' Values='<valeur>, <valeur>,... ' -->
```

Référencement de types Enum

```
<!-- AWP_In_Variable Name='<NomVar>' Enum="<Nom du type Enum>" -->
<!-- AWP_Out_Variable Name='<NomVar>' Enum="<Nom du type Enum>" -->
```

Création de fragments

```
<!-- AWP_Start_Fragment Name='<Nom>' [Type=<Type>] [ID=<id>] -->
```

Importation de fragments

```
<!-- AWP_Import_Fragment Name='<Nom>' -->
```

12.7.2.1 Lecture de variables

Les pages Web personnalisées peuvent lire des variables (variables API) et des variables de blocs de données dans la CPU, à condition que vous ayez configuré les variables comme étant accessibles à partir d'un HMI.

Syntaxe

```
:=<NomVar>:
```

Paramètres

<NomVar>	Variable à lire qui peut être un nom de variable API dans votre programme STEP 7, une variable de bloc de données, une E/S ou un memento adressable. N'utilisez pas de guillemets autour du nom de variable pour les mémentos, les adresses d'E/S et les noms d'alias (Page 1094). Pour les variables API, utilisez des guillemets de part et d'autre du nom de variable. Pour les variables de bloc de données, vous devez mettre uniquement le nom du bloc entre guillemets. Le nom de la variable se trouve en dehors des guillemets. Notez que vous indiquez le nom du bloc et non son numéro. Référez les éléments de tableau à l'aide de la syntaxe d'élément de tableau.
----------	---

Exemples

```
:= "Vitesse_convoyeur":
:= "Mon_bloc_données".memento1:
:= I0.0:
:= MW100:
:= "Mon_bloc_données".Tableau_dim1[0]:
:= "Mon_bloc_données".Tableau_dim2[0,0]:
```

Exemple de lecture d'une variable avec alias

```
<!-- AWP_Out_Variable Name='memento1'
Use=' "Mon_bloc_données".memento1' -->
:=memento1:
```

Remarque

La définition d'alias pour les variables API et les variables de blocs de données est décrite à la rubrique Utilisation d'un alias pour une référence de variable (Page 1088).

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique Gestion des noms de variables contenant des caractères spéciaux (Page 1094).

12.7.2.2 Ecriture de variables

Les pages personnalisées peuvent écrire des données dans la CPU. Pour ce faire, vous utilisez une commande AWP afin d'identifier une variable dans la CPU dans laquelle il est possible d'écrire depuis la page HTML. Il faut indiquer la variable par le nom de variable API ou par le nom de variable de bloc de données. Vous pouvez déclarer plusieurs noms de variables dans une instruction. Utilisez les commandes HTTP POST standard pour écrire les données dans la CPU.

Un usage typique est de concevoir un formulaire dans votre page HTML avec des champs de saisie de texte ou des choix de liste de sélection qui correspondent aux variables CPU inscriptibles. Comme avec toutes les pages personnalisées, vous générez alors les blocs dans STEP 7 de manière à les inclure dans votre programme STEP 7. Lorsqu'un utilisateur avec des privilèges de modification de variables accède par la suite à cette page et entre des données dans les champs d'entrée ou sélectionne un choix dans une liste de sélection, le serveur Web convertit l'entrée au type de données approprié pour la variable et écrit la valeur dans la variable dans la CPU. Notez que la clause Name pour les champs de saisie HTML et les listes de sélection HTML utilise une syntaxe typique de la clause Name de la commande AWP_In_Variable. Typiquement, vous écrivez le nom entre apostrophes et, si vous référencez un bloc de données, vous écrivez le nom du bloc de données entre guillemets.

Reportez-vous à la documentation HTML pour plus de détails sur la gestion des formulaires.

Syntaxe

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

Paramètres

<NomVar1>	Si aucune clause d'utilisation (Use) n'est fournie, "NomVar1" est la variable dans laquelle il faut écrire. Il peut s'agir d'un nom de variable API dans votre programme STEP 7, d'une variable d'un bloc de données spécifique ou d'un bloc de données. En présence d'une clause Use, NomVar1 est un nom de remplacement pour la variable référencée dans <NomVar2> (Page 1088). Il s'agit d'un nom local à l'intérieur de la page HTML.
<NomVar2>	En présence d'une clause Use, "NomVar2" est la variable dans laquelle il faut écrire. Il peut s'agir d'un nom de variable API dans votre programme STEP 7 ou d'une variable d'un bloc de données spécifique.

Pour les clauses Name et Use, le nom complet doit être entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API et d'un nom de bloc de données. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro. Référez les éléments de tableau à l'aide de la syntaxe d'élément de tableau.

Si vous utilisez une commande AWP_In_Variable pour rendre un bloc de données inscriptible, chaque variable dans le bloc de données sera inscriptible.

Exemples d'utilisation d'un champ de saisie HTML

```
<!-- AWP_In_Variable Name=' "Niveau_cible" ' -->
<form method="post">
<p>Saisir niveau cible : <input name=' "Niveau_cible" ' type="text" />
</p>
</form>
```

```
<!-- AWP_In_Variable Name=' "Bloc_données_1".Freinage' -->
<form method="post">
<p>Freinage : <input name=' "Bloc_données_1".Freinage' type="text" />
%</p>
</form>
```

```
<!-- AWP_In_Variable Name=' "Bloc_données_1".Tableau_dim2' -->
<form method="post">
<p>Valeur de tableau bidimensionnel [2,1] : <input
name=' "Bloc_données_1".Tableau_dim2[2,1]' type="text" /> %</p>
</form>
```

Exemple d'utilisation d'une clause Use

```
<!-- AWP_In_Variable Name=' "Freinage" '
Use=' "Bloc_données_1".Freinage' -->
<form method="post">
<p>Freinage : <input name=' "Freinage" ' type="text" /> %</p>
</form>
```


Exemple d'utilisation d'un bloc de données inscriptible

```

<!-- AWP_In_Variable Name='Bloc_données_1' -->
<form method="post">
<p>Freinage : <input name='Bloc_données_1.Freinage' type="text" />
%
</p>
<p>Vitesse de l'éolienne : <input
name='Bloc_données_1.VitesseEolienne' size="10"
value='Bloc_données_1.VitesseEolienne' type="text" />
</p>
</form>

```

Exemple d'utilisation d'une liste de sélection HTML

```

<!-- AWP_In_Variable
Name='Bloc_données_1.ActiverManuelPrioritaire'-->
<form method="post">
<select name='Bloc_données_1.ActiverManuelPrioritaire'>
<option value='Bloc_données_1.ActiverManuelPrioritaire:> </option>
<option value=1>Oui</option>
<option value=0>Non</option>
</select><input type="submit" value="Submit setting" /></form>

```

Remarque

Seul un utilisateur disposant du droit (Page 1042) "Écrire dans les pages personnalisées" peut écrire des données dans la CPU. Les commandes ne sont pas prises en compte par le serveur Web si l'utilisateur ne dispose pas des privilèges de modification.

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique "Gestion des noms de variables contenant des caractères spéciaux (Page 1094)".

12.7.2.3 Lecture de variables spéciales

Le serveur Web offre la possibilité de lire des valeurs dans l'API afin de les stocker dans des variables spéciales dans l'en-tête de réponse HTTP. Vous pourriez, par exemple, vouloir lire un nom de chemin dans une variable API pour réorienter l'URL vers un autre emplacement à l'aide de la variable spéciale HEADER:Location.

Syntaxe

```

<!-- AWP_Out_Variable Name='<Type>:<Nom>' [Use='<NomVar>'] -->

```

Paramètres

<Type>	Type de la variable spéciale. Ce peut être : HEADER COOKIE_VALUE COOKIE_EXPIRES
<Nom>	Vous trouverez dans la documentation HTTP la liste de tous les noms de variables HEADER. En voici quelques exemples : Status : code de réponse Location : chemin pour un réacheminement Retry-After : durée pendant laquelle le service risque d'être indisponible pour le client demandeur Pour les types COOKIE_VALUE et COOKIE_EXPIRES, <Nom> est le nom du cookie spécifique. COOKIE_VALUE:nom : valeur du cookie nommé COOKIE_EXPIRES:nom : durée d'expiration en secondes du cookie nommé La clause Name doit être écrite entre apostrophes ou entre guillemets. Si aucune clause Use n'est indiquée, le nom de variable spéciale correspond à un nom de variable API. Ecrivez la clause Name entière entre apostrophes et la variable API entre guillemets. Le nom de variable spéciale et le nom de variable API doivent correspondre exactement.
<NomVar>	Nom de la variable API ou de bloc de données dans laquelle la variable doit être lue NomVar doit être écrit entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API ou d'un nom de bloc de données. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro.

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique Gestion des noms de variables contenant des caractères spéciaux (Page 1094).

Exemple : Lecture d'une variable spéciale sans clause Use

```
<!-- AWP_Out_Variable Name="HEADER:Status" -->
```

Dans cet exemple, la variable spéciale HTTP "HEADER:Status" reçoit la valeur de la variable API "HEADER:Status". Le nom dans la table de variables API doit correspondre exactement au nom de la variable spéciale si aucune clause Use n'est indiquée.

Exemple : Lecture d'une variable spéciale avec clause Use

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use='Etat' -->
```

Dans cet exemple, la variable spéciale "HEADER:Status" reçoit la valeur de la variable API "Etat".

12.7.2.4 Ecriture de variables spéciales

Le serveur Web offre la possibilité d'écrire dans la CPU des valeurs provenant de variables spéciales dans l'en-tête de requête HTTP. Vous pouvez, par exemple, stocker dans STEP 7 des informations sur le cookie associé à une page Web personnalisée, sur l'utilisateur qui accède à une page ou sur des informations d'en-tête. Le serveur Web permet d'accéder à des variables spéciales spécifiques que vous pouvez écrire dans la CPU si vous avez ouvert une session avec des privilèges de modification de variables.

Syntaxe

```
<!-- AWP_In_Variable Name='<Type>:<Nom>' [Use='<NomVar>']-->
```

Paramètres

<Type>	Type de la variable spéciale. Ce peut être : HEADER SERVER COOKIE_VALUE
<Nom>	Variable spécifique dans les types définis ci-dessus, comme illustré dans ces exemples : HEADER:Accept : types de contenu qui sont acceptables HEADER:User-Agent : informations sur l'agent utilisateur à l'origine de la requête SERVER:current_user_id : ID de l'utilisateur en cours ; 0 si aucun utilisateur n'a ouvert de session SERVER:current_user_name : nom de l'utilisateur en cours COOKIE_VALUE:<name> : valeur du cookie nommé Ecrivez la clause Name entre apostrophes. Si aucune clause Use n'est indiquée, le nom de variable spéciale correspond à un nom de variable API. Ecrivez la clause Name entière entre apostrophes et la variable API entre guillemets. Le nom de variable spéciale et le nom de variable API doivent correspondre exactement. Vous trouverez dans la documentation HTTP la liste de tous les noms de variables HEADER.
<NomVar>	Nom de la variable dans votre programme STEP 7 dans laquelle vous voulez écrire la variable spéciale. Ce peut être un nom de variable API ou une variable de bloc de données. NomVar doit être écrit entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API ou d'un nom de bloc de données. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro.

Exemples

```
<!-- AWP_In_Variable Name="SERVER:current_user_id" -->
```

Dans cet exemple, la page Web écrit la valeur de la variable spéciale HTTP "SERVER:current_user_id" dans la variable API de nom "SERVER:current_user_id".

```
<!-- AWP_In_Variable Name=SERVER:current_user_id'
Use="mon_IDutilisateur" -->
```

Dans cet exemple, la page Web écrit la valeur de la variable spéciale HTTP "SERVER:current_user_id" dans la variable API de nom "mon_IDutilisateur".

Remarque

Seul un utilisateur avec des privilèges de modification de variables peut écrire des données dans la CPU. Les commandes ne sont pas prises en compte par le serveur Web si l'utilisateur ne dispose pas des privilèges de modification.

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique "Gestion des noms de variables contenant des caractères spéciaux (Page 1094)".

12.7.2.5 Utilisation d'un alias pour une référence de variable

Vous pouvez utiliser dans votre page Web personnalisée un alias pour une variable In_Variable ou Out_Variable. Par exemple, vous pouvez utiliser dans votre page HTML un nom symbolique différent de celui utilisé dans la CPU ou établir une correspondance entre une variable dans la CPU et une variable spéciale. La clause AWP Use offre cette possibilité.

Syntaxe

```
<-- AWP_In_Variable Name='<NomVar1>' Use='<NomVar2>' -->
<-- AWP_Out_Variable Name='<NomVar1>' Use='<NomVar2>' -->
```

Paramètres

<NomVar1>	Nom de l'alias ou de la variable spéciale NomVar1 doit être écrit entre apostrophes ou entre guillemets.
<NomVar2>	Nom de la variable API à laquelle vous voulez affecter un alias. Il peut s'agir d'une variable API, d'une variable de bloc de données ou d'une variable spéciale. NomVar2 doit être écrit entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API, de variable spéciale ou de bloc de données. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro.

Exemples

```
<-- AWP_In_Variable Name='SERVER:current_user_id'
Use=' "Data_Block_10".utilisateur_serveur' -->
```

Dans cet exemple, la variable spéciale SERVER:current_user_id est écrite dans la variable "utilisateur_serveur" du bloc de données "Data_Block_10".

```
<-- AWP_Out_Variable Name='Poids'
Use=' "Data_Block_10".Données_cuve.Poids' -->
```

Dans cet exemple, la valeur dans l'élément de structure de bloc de données Data_Block_10.Données_cuve.Poids peut être plus simplement référencé par son alias "Poids" dans tout le reste de la page Web personnalisée.

```
<-- AWP_Out_Variable Name='Poids' Use=' "Poids_cuve_lait_cru"' -->
```

Dans cet exemple, la valeur dans la variable API "Poids_cuve_lait_cru" peut être plus simplement référencée par son alias "Poids" dans tout le reste de la page Web personnalisée.

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique Gestion des noms de variables contenant des caractères spéciaux (Page 1094).

12.7.2.6 Définition de types Enum

Vous pouvez définir des types Enum dans vos pages personnalisées et affecter les éléments dans une commande AWP.

Syntaxe

```
<!-- AWP_Enum_Def Name='<nom du type Enum>' Values='<valeur>,
<valeur>, ... ' -->
```

Paramètres

<nom du type Enum>	Nom du type énumération entre apostrophes ou entre guillemets
<valeur>	<constante>:<nom> La constante indique la valeur numérique pour l'affectation du type Enum. Le nombre total n'est pas borné. Le nom est la valeur affectée à l'élément Enum.

Notez que la chaîne entière d'affectations de valeur Enum est placée entre apostrophes, chaque affectation individuelle d'un élément du type Enum étant placée entre guillemets. La portée d'une définition de type Enum est globale pour les pages Web personnalisées. Si vous avez configuré vos pages Web personnalisées dans des dossiers de langue (Page 1115), la définition du type Enum est globale pour toutes les pages dans le dossier de langue.

Exemple

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Pas d'alarmes",
1:"Cuve pleine", 2:"Tank is empty"' -->
```

12.7.2.7 Référencage de variables CPU avec un type Enum

Vous pouvez affecter une variable dans la CPU à un type Enum. Cette variable peut être utilisée à un autre endroit de votre page Web personnalisée dans une opération de lecture (Page 1082) ou d'écriture (Page 1083). Dans une opération de lecture, le serveur Web remplace la valeur numérique lue dans la CPU par la valeur textuelle Enum correspondante. Dans une opération d'écriture, le serveur Web remplace la valeur textuelle par la valeur entière de l'énumération correspondant au texte avant d'écrire la valeur dans la CPU.

Syntaxe

```
<!-- AWP_In_Variable Name='<NomVar>' Enum="<TypeEnum>" -->
<!-- AWP_Out_Variable Name='<NomVar>' Enum="<TypeEnum>" -->
```

Paramètres

<NomVar>	Nom de la variable API ou de la variable de bloc de données à associer au type Enum ou nom d'alias pour une variable API (Page 1088) le cas échéant. NomVar doit être écrit entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API ou d'un nom de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données.
<EnumTypeEnum>	Nom du type énumération entre apostrophes ou entre guillemets

La portée d'une référence au type Enum est le fragment en cours.

Exemple d'utilisation dans une lecture de variables

```
<!-- AWP_Out_Variable Name='Alarme' Enum='EnumAlarme' -->...
<p>La valeur en cours de "Alarme" est :=Alarme:</p>
```

Si la valeur de "Alarme" dans la CPU est 2, la page HTML affiche 'La valeur en cours de "Alarme" est Cuve vide' car la définition du type Enum (Page 1089) affecte la chaîne littérale "Cuve vide" à la valeur numérique 2.

Exemple d'utilisation dans une écriture de variables

```
<!-- AWP_Enum_Def Name='EnumAlarme' Values='0:"Pas d'alarmes",
1:"Cuve pleine", 2:"Cuve vide"' -->
<!-- AWP_In_Variable Name='Alarme' Enum='EnumAlarme' -->...
<form method="POST">
<p><input type="hidden" name='Alarme' value="Cuve pleine" /></p>
<p><input type="submit" value='Activer Cuve pleine' /></p>
</form>
```

Comme la définition du type Enum (Page 1089) affecte "Cuve pleine" à la valeur numérique 1, la valeur 1 est écrite dans la variable API nommée "Alarm" dans la CPU.

Remarquez que la clause Enum dans la déclaration AWP_In_Variable doit correspondre exactement à la clause Name dans la déclaration AWP_Enum_Def .

Exemple d'utilisation dans une écriture de variables avec utilisation d'un alias

```
<!-- AWP_Enum_Def Name='EnumAlarme' Values='0:"Pas d'alarmes",
1:"Cuve pleine", 2:"Cuve vide"' -->
<!-- AWP_In_Variable Name='"Alarme"' Enum='EnumAlarme'
Use=' "Data_block_4".Moteur1.Alarme'-->...
<form method="POST">
<p><input type="hidden" name='"Alarme"' value="Cuve pleine" /></p>
<p><input type="submit" value='Activer Cuve pleine' /></p>
</form>
```

Comme la définition du type Enum (Page 1089) affecte "Cuve pleine" à la valeur numérique 1, la valeur 1 est écrite dans l'alias "Alarme" qui correspond à la variable API nommée "Moteur1.Alarme" dans le bloc de données "Data_Block_4" dans la CPU.

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique Gestion des noms de variables contenant des caractères spéciaux (Page 1094).

Remarque

Dans des versions précédentes, une déclaration AWP_Enum_Ref séparée était nécessaire pour associer une variable à un type Enum défini. STEP 7 et S7-1200 prennent en charge le code existant avec les déclarations AWP_Enum_Ref ; cependant, cette commande n'est plus nécessaire.

12.7.2.8 Création de fragments

STEP 7 convertit et sauvegarde les pages Web personnalisées sous la forme d'un DB de commande et de DB de fragment lorsque vous cliquez sur "Générer blocs" dans les propriétés CPU pour le serveur Web. Vous pouvez configurer des fragments spécifiques pour des pages spécifiques ou pour des parties de pages spécifiques. Vous pouvez identifier ces fragments par un nom et un numéro à l'aide de la commande AWP "AWP_Start_Fragment". Tout ce qu'il y a dans la page après la commande AWP_Start_Fragment appartient à ce fragment jusqu'à ce qu'une autre commande AWP_Start_Fragment soit émise ou jusqu'à ce que la fin de fichier soit atteinte.

Syntaxe

```
<!-- AWP_Start_Fragment Name='<Nom>'
[Type=<Type>] [ID=<id>] [Mode=<Mode>] -->
```

Paramètres

<Nom>	<p>Chaîne de texte : nom du DB de fragment</p> <p>Les noms de fragment doivent commencer par une lettre ou un trait de soulignement et contenir des lettres, des chiffres et des traits de soulignement. Le nom de fragment est une expression régulière de la forme : [a-zA-Z_][a-zA-Z_0-9]*</p>
<Type>	<p>Manuel ou automatique</p> <p>Manuel : Le programme STEP 7 doit demander ce fragment et peut réagir en conséquence. L'exploitation du fragment doit être gérée avec STEP 7 et les variables du DB de commande.</p> <p>Automatique : Le serveur Web traite le fragment automatiquement.</p> <p>Si vous n'indiquez pas le paramètre "type", "automatique" est pris par défaut.</p>
<id>	<p>Numéro d'identification entier. Si vous n'indiquez pas le paramètre "id", le serveur Web affecte un numéro par défaut. Donnez un numéro d'ID faible aux fragments manuels. C'est au moyen de l'ID que le programme STEP 7 gère un fragment manuel.</p>
<Mode>	<p>Visible ou masqué</p> <p>visible : le contenu du fragment s'affichera sur la page Web personnalisée.</p> <p>masqué : le contenu du fragment ne s'affichera pas sur la page Web personnalisée.</p> <p>Si vous n'indiquez pas le paramètre "mode", "visible" est pris par défaut.</p>

Fragments manuels

Si vous créez un fragment manuel pour une page Web personnalisée ou une partie d'une telle page, votre programme STEP 7 doit gérer le moment où le fragment est envoyé. Le programme STEP 7 doit définir les paramètres appropriés dans le DB de commande pour une page personnalisée en mode manuel, puis appeler l'instruction WWW avec le DB de commande modifié. Consultez le paragraphe Commande avancée de pages Web personnalisées (Page 1119) pour plus d'informations sur la structure du DB de commande et la manipulation de pages et de fragments individuels.

12.7.2.9 Importation de fragments

Vous pouvez créer un fragment avec un nom à partir d'une partie de votre code HTML, puis importer ce fragment à un autre endroit dans votre jeu de pages Web personnalisées. Considérons, par exemple, un jeu de pages Web personnalisées qui comprend une page de démarrage et plusieurs autres pages HTML accessibles à l'aide de liens sur cette page de démarrage. Supposons que chacune de ces pages distinctes doive afficher le logo de l'entreprise. Vous pouvez réaliser cette opération en créant un fragment (Page 1091) qui charge l'image du logo de l'entreprise. Chaque page HTML individuelle peut alors importer ce fragment pour afficher le logo de l'entreprise. Utilisez pour ce faire une commande AWP_Import_Fragment. Le code HTML pour ce fragment existe uniquement dans un fragment, mais vous pouvez importer ce DB de fragment autant de fois que nécessaire dans autant de pages Web que vous le désirez.

Syntaxe

```
<!-- AWP_Import_Fragment Name='<Nom>' -->
```


Paramètres

<Nom>	Chaîne de texte : nom du DB de fragment à importer
-------	--

Exemple

Extrait de code HTML qui crée un fragment pour afficher une image :

```
<!-- AWP_Start_Fragment Name='Mon_logo_dentreprise' --><p></p>
```

Extrait de code HTML d'un autre fichier .html qui importe le fragment affichant l'image du logo :

```
<!-- AWP_Import_Fragment Name='Mon_logo_dentreprise' -->
```

Les deux fichiers .html (celui qui crée le fragment et celui qui l'importe) sont dans la structure de dossiers que vous définissez lorsque vous configurez les pages personnalisées dans STEP 7 (Page 1097).

12.7.2.10 Combinaison de définitions

Lorsque vous déclarez des variables à utiliser dans vos pages Web personnalisées, vous pouvez combiner une déclaration de variable et un alias pour la variable (Page 1088). Vous pouvez aussi déclarer plusieurs variables In_Variables et plusieurs variables Out_Variables dans une instruction.

Exemples

```
<!-- AWP_In_Variable Name='"Niveau"', Name='"Poids"', Name='"Temp"'
-->
<--! AWP_Out_Variable Name='HEADER:Status', Use='"Etat"',
      Name='HEADER:Location', Use='"Emplacement"',
      Name='COOKIE_VALUE:name', Use='"mon_cookie"' -->
<!-- AWP_In_Variable Name='Alarme' Use='"Data_block_10".Alarme' -->
```

12.7.2.11 Gestion des noms de variables contenant des caractères spéciaux

Lorsque vous indiquez des noms de variables dans des pages Web personnalisées, vous devez être particulièrement prudent si ces noms contiennent des caractères ayant une signification particulière.

Lecture de variables

Vous utilisez la syntaxe suivante pour lire une variable (Page 1082) :

`:=<NomVar>` :

Les règles suivantes s'appliquent à la lecture de variables :

- Pour les noms de variables provenant de la table de variables API, placez le nom de variable entre guillemets.
- Pour les noms de variables qui sont des variables de bloc de données, placez le nom du bloc de données entre guillemets. La variable se trouve en dehors des guillemets.
- Pour les noms de variables qui sont des adresses d'E/S directes, des adresses de memento ou des noms d'alias, n'utilisez pas de guillemets ou apostrophes autour de la variable lue.
- Pour les noms de variables ou de variables de blocs de données contenant une barre oblique inversée, faites précéder la barre oblique inversée d'une autre barre oblique inversée.
- Si un nom de variable ou de variable de bloc de données contient un point, un signe inférieur à, un signe supérieur à ou une perluète, définissez pour la variable lue un alias sans caractères spéciaux et lisez la variable à l'aide de cet alias. Faites précéder les deux-points de noms de variables dans une clause Use d'une barre oblique inversée.

Tableau 12- 1 Exemples de lecture de variables

Nom du bloc de données	Nom de la variable	Commande de lecture
Sans objet	ABC:DEF	<pre><!--AWP_Out_Variable Name='variable_speciale' Use ="ABC:DEF" --> :=variable_speciale:</pre>
Sans objet	T\	<pre>:"T\":</pre>
Sans objet	A \B 'C :D	<pre><!--AWP_Out_Variable Name='autre_variable_speciale' Use="A \B \C :D" --> :=autre_variable_speciale:</pre>
Sans objet	a<b	<pre><!--AWP_Out_Variable Name='a_inférieur_à_b' Use="a<b" --> :=a inférieur à b:</pre>
Data_block_1	Tag_1	<pre>:"Data_block_1".Tag_1:</pre>
Data_block_1	ABC:DEF	<pre><!-- AWP_Out_Variable Name='variable_speciale' Use="Data_block_1".ABC\ :DEF'--> :=variable_speciale:</pre>
DB A' B C D\$ E	Tag	<pre>:"DB A' B C D\$ E".Tag:</pre>
DB:DB	Tag:Tag	<pre><!--AWP_Out_Variable Name='ma_variable' Use ="DB:DB".Tag\ :Tag' --> :=ma_variable:</pre>

Clauses Name et Use

Les commandes AWP `AWP_In_Variable`, `AWP_Out_Variable`, `AWP_Enum_Def`, `AWP_Enum_Ref`, `AWP_Start_Fragment` et `AWP_Import_Fragment` comportent des clauses Name. Les commandes de formulaire HTML telles que `<input>` et `<select>` ont également des clauses Name. `AWP_In_Variable` et `AWP_Out_Variable` peuvent en outre comporter des clauses Use. Quelle que soit la commande, la syntaxe des clauses Name et Use est identique en ce qui concerne la gestion des caractères spéciaux :

- Le texte que vous fournissez pour une clause Name ou Use doit être entre apostrophes. Si le nom concerné est une variable API ou un nom de bloc de données, utilisez des apostrophes pour la clause entière.
- A l'intérieur de la clause Name ou Use, les noms de blocs de données et les noms de variables API doivent être entre guillemets.
- Si un nom de variable ou de bloc de données comprend une apostrophe ou une barre oblique inversée, faites précéder ce caractère d'une barre oblique inversée. La barre oblique inversée est le caractère d'échappement dans le compilateur de commandes AWP.

Tableau 12- 2 Exemples de clauses Name

Nom du bloc de données	Nom de la variable	Exemples de clauses Name
Sans objet	ABC'DEF	Name=' "ABC\ 'DEF" '
Sans objet	A\B'C:D	Name=' "A \\B \'C :D" '
Data_block_1	Tag_1	Name=' "Data_block_1".Tag_1'
Data_block_1	ABC'DEF	Name=' "Data_block_1".ABC\ 'DEF'
Data_block_1	A\B'C:D	Name=' "Data_block_1".A \\B \'C :D'
DB A' B C D\$ E	Tag	Name=' "DB A\ \' B C D\$ E".Tag'

Les clauses Use utilisent les mêmes conventions que les clauses Name.

Remarque

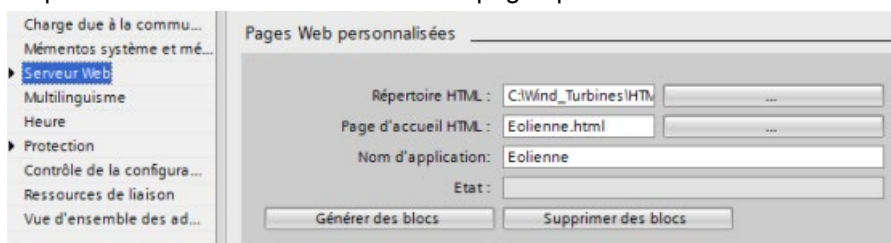
Quels que soient les caractères que vous utilisez dans votre page HTML, définissez le jeu de caractères de la page HTML à UTF-8 et sauvegardez-la à partir de l'éditeur avec le codage de caractères UTF-8.

12.7.3 Configuration de l'utilisation de pages Web personnalisées

Procédez comme suit pour configurer des pages Web personnalisées dans STEP 7 :

1. Sélectionnez la CPU dans la vue Configuration des appareils.
2. Affichez les propriétés "Serveur Web" dans la fenêtre d'inspection pour la CPU.
3. Si elle n'est pas déjà cochée, cochez la case "Activer le serveur Web sur le module".
4. Sélectionnez "Autoriser l'accès uniquement via HTTPS" pour vous assurer que le serveur Web utilise une communication cryptée et pour augmenter la sécurité de votre CPU accessible par le Web.
5. Entrez le nom du dossier sur votre PC où vous avez sauvegardé la page HTML par défaut (page d'accueil) ou naviguez jusqu'à ce dossier.
6. Entrez le nom de la page par défaut.
7. Indiquez un nom pour votre application (facultatif). Le serveur Web utilise le nom de l'application pour former des sous-catégories ou des groupes de pages Web. Lorsque vous fournissez un nom d'application, le serveur Web crée une URL pour votre page personnalisée dans le format suivant :
 http[s]://ww.xx.yy.zz/awp/<nom_application>/<nom_page>.html. L'URL est
 http[s]://ww.xx.yy.zz/awp/<nom_page>.html si vous ne fournissez pas de nom d'application.

Évitez des caractères spéciaux dans le nom de l'application. Certains caractères peuvent empêcher le serveur Web d'afficher les pages personnalisées.



8. Dans la section Avancés, entrez les extensions des fichiers qui incluent des commandes AWP. Par défaut, STEP 7 analyse les fichiers d'extension .htm, .html et .js. Si vous avez des extensions de fichier supplémentaires, ajoutez-les. Pour économiser les ressources de traitement, n'indiquez pas d'extensions correspondant à des fichiers dont aucun ne contient de commandes AWP.
9. Conservez la valeur par défaut pour le numéro du DB Web ou entrez un numéro de votre choix. Il s'agit du numéro du DB de commande pilotant l'affichage des pages Web.
10. Conservez la valeur par défaut pour le numéro de début des DB de fragment ou entrez un numéro de votre choix. Il s'agit du premier des DB de fragment qui contiennent les pages Web.

Génération de blocs de programme

Lorsque vous cliquez sur le bouton "Générer blocs", STEP 7 génère des blocs de données à partir des pages HTML dans le répertoire source HTML que vous avez indiqué, ainsi qu'un bloc de données de commande pour la gestion de vos pages Web. Vous pouvez définir les attributs selon les besoins de votre application (Page 1099). STEP 7 génère aussi un jeu de blocs de données de fragment pour contenir la représentation de toutes vos pages HTML. Lorsque vous générez les blocs de données, STEP 7 actualise les propriétés afin qu'elles affichent le numéro du bloc de données de commande et le numéro du premier des blocs de données de fragment. Une fois les blocs de données générés, vos pages Web personnalisées font partie de votre programme STEP 7. Les blocs correspondant à ces pages apparaissent dans le dossier de serveur Web qui se trouve dans le dossier Blocs système sous Blocs de programme dans le navigateur du projet.

Suppression de blocs de programme

Pour supprimer les blocs de données précédemment générés, cliquez sur le bouton "Supprimer blocs". STEP 7 supprime de votre projet le bloc de données de commande et tous les blocs de données de fragment correspondant aux pages Web personnalisées.

12.7.4 Configuration de la page d'entrée

Dans la configuration d'appareil de la CPU, vous avez la possibilité de désigner une page Web personnalisée comme page d'entrée qui s'affiche lorsque vous accédez au serveur Web à partir d'un PC ou d'un appareil mobile. Si vous ne le faites pas, la page d'entrée sera la page Web standard Introduction (Page 1054).

Procédez comme suit pour définir une page Web personnalisée comme page d'entrée :

1. Sélectionnez la CPU dans la vue Configuration des appareils.
2. Dans la fenêtre d'inspection, sélectionnez "Serveur Web" dans les propriétés de la CPU et activez le serveur Web (Page 1040).
3. Sélectionnez "Page d'entrée" dans les propriétés du serveur Web.
4. Sélectionnez "UP1" dans la liste déroulante pour configurer le serveur Web de telle manière qu'il affiche une page personnalisée lorsqu'on y accède. (L'autre choix, "Page d'accueil", paramètre le serveur Web pour qu'il affiche la page Web standard Introduction lors de l'accès.)

Il faut également configurer l'utilisateur "Everybody" de manière qu'il dispose des droits (Page 1042) "Appeler des pages personnalisées" et inclure un appel à l'instruction WWW (Page 1099) dans votre programme.

Une fois la configuration achevée et le projet chargé dans la CPU, le serveur Web peut utiliser comme page d'entrée la page HTML par défaut que vous avez sélectionnée lorsque vous avez configuré vos pages Web personnalisées (Page 1097).

Remarque

La CPU doit être à l'état MARCHE pour afficher une page d'entrée personnalisée.


12.7.5 Programmation de l'instruction WWW pour les pages Web personnalisées

Votre programme utilisateur STEP 7 doit contenir et exécuter l'instruction WWW afin que les pages Web personnalisées soient accessibles à partir des pages Web standard. Le bloc de données de commande est le paramètre d'entrée de l'instruction WWW et indique le contenu des pages tel que représenté dans les blocs de données de fragment, ainsi que des informations d'état et de commande. STEP 7 crée le bloc de données de commande lorsque vous cliquez sur le bouton "Générer blocs" dans la configuration des pages Web personnalisées (Page 1097).

Programmation de l'instruction WWW

Le programme STEP 7 doit exécuter l'instruction WWW pour rendre les pages Web personnalisées accessibles à partir des pages Web standard. Mais vous voudrez peut-être que les pages Web personnalisées ne soient disponibles que dans certaines circonstances, en fonction des exigences de votre application et de vos préférences. Dans ce cas, la logique de votre programme peut gérer le moment où l'instruction WWW est appelée.

Tableau 12- 3 Instruction WWW

CONT/LOG	SCL	Description
	<pre>ret_val := WWW(ctrl_db:=_uint_in_);</pre>	Permet l'accès aux pages Web personnalisées à partir des pages Web standard.

Vous devez indiquer le paramètre d'entrée bloc de données de commande (CTRL_DB) qui correspond au numéro de DB entier du DB de commande. Vous trouverez ce numéro de DB de commande (numéro du DB Web) dans les propriétés du serveur Web de la CPU après avoir créé les blocs pour les pages Web personnalisées. Entrez le numéro de DB entier comme paramètre CTRL_DB de l'instruction WWW. La valeur en retour RET_VAL contient le résultat de la fonction. Notez que l'instruction WWW s'exécute de manière asynchrone et que la sortie RET_VAL peut avoir une valeur initiale égale à 0 même si une erreur peut se produire par la suite. Le programme peut vérifier l'état du DB de commande pour s'assurer que l'application a été lancée avec succès ou contrôler RET_VAL à l'aide d'un appel ultérieur de WWW.

Tableau 12- 4 Valeur en retour

RET_VAL	Description
0	Pas d'erreur
16#00yx	<p>x : La requête représentée par le bit respectif est en état d'attente :</p> <p>x=1 : requête 0</p> <p>x=2 : requête 1</p> <p>x=4 : requête 2</p> <p>x=8 : requête 3</p> <p>Une opération OU logique peut être effectuée sur les valeurs de x, le résultat représentant alors les états d'attente de plusieurs requêtes. Par exemple, x égal à 6 signifie que les requêtes 1 et 2 sont en attente.</p> <p>y : 0 : pas d'erreur ; 1 : il y a une erreur et "last_error" a été mis à 1 dans le DB de commande (Page 1119)</p>
16#803a	Le DB de commande n'est pas chargé.
16#8081	Le DB de commande a un type, un format ou une version incorrecte.
16#80C1	Il n'y a pas de ressources disponibles pour initialiser l'application Web.

Utilisation du DB de commande

STEP 7 crée le bloc de données de commande lorsque vous cliquez sur "Générer blocs" et affiche le numéro du DB de commande dans les propriétés des pages Web personnalisées. Vous pouvez également trouver le DB de commande dans le dossier Blocs de programme du navigateur de projet.

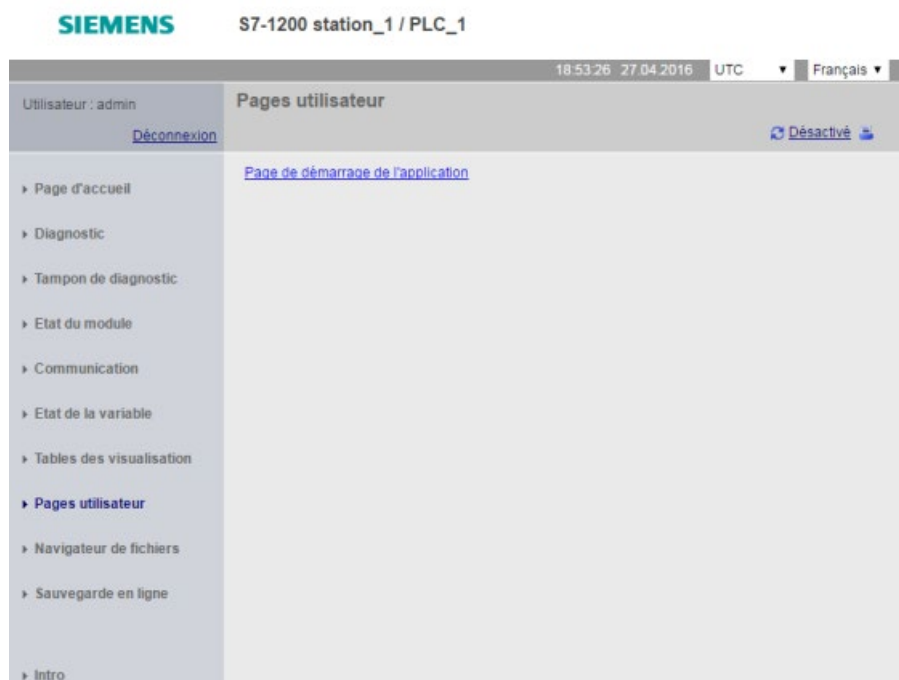
Typiquement, votre programme STEP 7 utilise le DB de commande directement tel qu'il a été créé par l'opération "Générer blocs" sans manipulation supplémentaire. Toutefois, le programme utilisateur STEP 7 peut définir des commandes globales dans le DB de commande afin de désactiver le serveur Web ou le réactiver par la suite. En outre, le programme utilisateur STEP 7 doit gérer le comportement des pages personnalisées que vous créez comme DB de fragment manuels (Page 1097) à l'aide d'une table de requête dans le DB de commande. Pour plus d'informations sur ces tâches avancées, reportez-vous à la rubrique Commande avancée de pages Web personnalisées (Page 1119).

12.7.6 Chargement des blocs de programme dans la CPU

Une fois que vous avez généré les blocs pour les pages Web personnalisées, ils font partie de votre programme STEP 7 tout comme tous les autres blocs de programme. Vous procédez de la manière usuelle pour les charger dans la CPU. Notez que vous pouvez charger les blocs de programme des pages Web personnalisées uniquement lorsque la CPU est à l'état ARRET.

12.7.7 Accès aux pages Web personnalisées

Vous accédez à vos pages Web personnalisées à partir des pages Web standard (Page 1044). Les pages Web standard affichent un lien pour les "Pages utilisateur" dans le menu de navigation sur le côté gauche. La navigation de base entre pages comporte également un lien vers les "Pages utilisateur". Lorsque vous cliquez sur le lien "Pages utilisateur", votre navigateur Web ouvre la page fournissant un lien vers votre page par défaut. Dans les pages personnalisées, la navigation se fait conformément à la manière dont vous avez conçu les différentes pages.



Remarque

Vous pouvez également définir une page personnalisée comme page d'entrée (Page 1098) pour le serveur Web.

12.7.8 Contraintes spécifiques aux pages Web personnalisées

Les contraintes pour les pages Web standard (Page 1124) s'appliquent également aux pages Web personnalisées. Il faut en outre tenir compte de quelques points spécifiques pour les pages Web personnalisées.

Espace en mémoire de chargement

Lorsque vous cliquez sur "Générer blocs", vos pages Web personnalisées deviennent des blocs de données qui ont besoin d'espace en mémoire de chargement. Si une carte mémoire est installée, vous disposez de la capacité de votre carte mémoire comme espace de mémoire de chargement externe pour les pages Web personnalisées.

Si vous n'avez pas installé de carte mémoire, ces blocs prennent de l'espace en mémoire de chargement interne qui est limitée selon votre modèle de CPU.

Vous pouvez vérifier la quantité de mémoire de chargement utilisée et disponible à l'aide des outils en ligne et de diagnostic dans STEP 7. Vous pouvez également consulter les propriétés des blocs individuels que STEP 7 génère à partir de vos pages Web personnalisées et voir la mémoire de chargement qu'ils consomment.

Remarque

Si vous devez réduire l'espace nécessaire pour vos pages Web personnalisées, utilisez moins d'images si c'est possible.

Guillemets dans les chaînes de texte

Évitez d'utiliser des chaînes de texte qui contiennent des apostrophes ou des guillemets insérées dans des variables de blocs de données que vous utilisez à n'importe quelle fin dans des pages Web personnalisées. Étant donné que la syntaxe HTML utilise souvent des apostrophes ou des guillemets comme séparateurs, les apostrophes ou les guillemets dans les chaînes de texte peuvent endommager l'affichage des pages Web personnalisées.

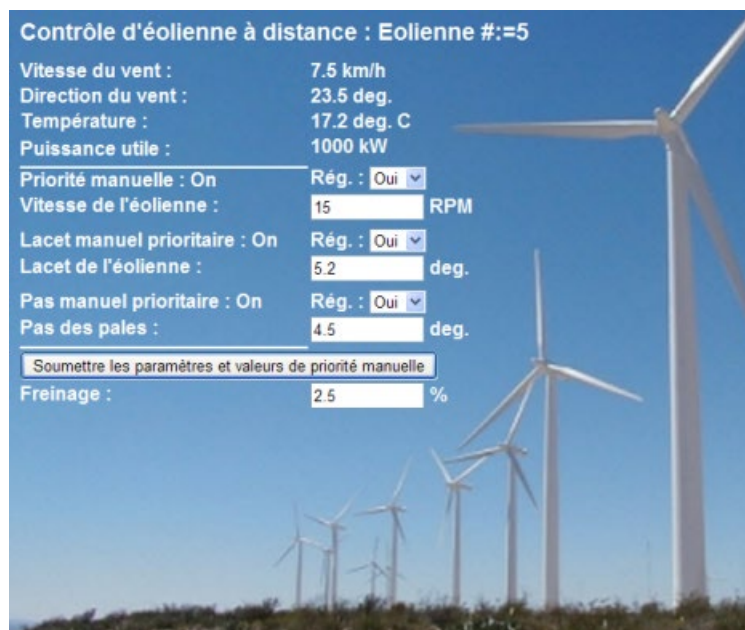
Pour les variable de blocs de données de type String que vous utilisez dans les pages Web personnalisées, observez les règles suivantes :

- N'entrez pas d'apostrophes ou de guillemets dans la valeur de la chaîne de variables de blocs de données dans STEP 7.
- Ne laissez pas le programme utilisateur effectuer des affectations de chaînes contenant des apostrophes ou des guillemets dans ces variables de blocs de données.

12.7.9 Exemple de page Web personnalisée

12.7.9.1 Page Web pour le contrôle-commande d'une éolienne

Prenons, comme exemple de page Web personnalisée, une page Web utilisée pour le contrôle-commande à distance d'une éolienne :



Contrôle d'éolienne à distance : Eolienne #:=5	
Vitesse du vent :	7.5 km/h
Direction du vent :	23.5 deg.
Température :	17.2 deg. C
Puissance utile :	1000 kW
Priorité manuelle : On	Rég. : <input type="button" value="Oui"/>
Vitesse de l'éolienne :	<input type="text" value="15"/> RPM
Lacet manuel prioritaire : On	Rég. : <input type="button" value="Oui"/>
Lacet de l'éolienne :	<input type="text" value="5.2"/> deg.
Pas manuel prioritaire : On	Rég. : <input type="button" value="Oui"/>
Pas des pales :	<input type="text" value="4.5"/> deg.
<input type="button" value="Soumettre les paramètres et valeurs de priorité manuelle"/>	
Freinage :	<input type="text" value="2.5"/> %

Description

Dans cette application, chaque éolienne d'une ferme d'éoliennes est équipée d'un S7-1200 pour la commande de l'éolienne. Dans le programme STEP 7, chaque éolienne dispose d'un bloc de données avec des données spécifiques concernant cette éolienne.

La page Web personnalisée fournit un accès à distance à l'éolienne à partir d'un PC. Un utilisateur peut se connecter aux pages Web standard de la CPU d'une éolienne particulière et accéder à la page Web personnalisée de contrôle d'éolienne à distance pour visualiser les données de cette éolienne. Un utilisateur avec les privilèges de modification de variables peut également faire passer l'éolienne en mode manuel et piloter les variables pour la vitesse, le lacet et le pas à partir de la page Web. Il peut en outre définir une valeur de freinage, que l'éolienne soit en mode manuel ou automatique.

Le programme STEP 7 doit surveiller les valeurs booléennes de substitution du mode automatique et, si elles sont à 1, utiliser les valeurs de vitesse d'éolienne, lacet et pas entrées par l'utilisateur. Si ce n'est pas le cas, le programme ne tient pas compte de ces valeurs.

Fichiers utilisés

Cet exemple de page Web personnalisée consiste en trois fichiers :

- **Wind_turbine.html** : Il s'agit de la page HTML qui implémente l'écran montré plus haut à l'aide de commandes AWP pour accéder aux données de l'automate.
- **Wind_turbine.css** : Il s'agit de la feuille de style en cascade qui contient les règles de formatage de la page HTML. L'utilisation d'une feuille de style en cascade est facultative, mais elle peut simplifier la conception des pages Web.
- **Wind_turbine.jpg** : Il s'agit de l'image d'arrière-plan utilisée par la page HTML. L'utilisation d'images dans des pages Web personnalisées est, bien sûr, optionnelle et requiert en outre de l'espace supplémentaire dans la CPU.

Ces fichiers ne sont pas fournis avec votre installation, mais sont décrits comme exemple.

Mise en oeuvre

La page HTML utilise des commandes AWP pour lire des valeurs dans l'API (Page 1082) pour les champs d'affichage et pour écrire des valeurs dans l'API (Page 1083) correspondant à des données saisies par l'utilisateur. Cette page utilise également des commandes AWP pour la définition (Page 1089) et la référence (Page 1090) de types Enum pour la gestion des réglages ON/OFF.

La première partie de la page affiche une ligne d'en-tête qui comprend le numéro de l'éolienne.

Contrôle d'éolienne à distance : Eolienne #5

La partie suivante de la page affiche les conditions atmosphériques au niveau de l'éolienne. Les entrées/sorties qui sont situées sur le site de l'éolienne fournissent la vitesse et la direction du vent ainsi que la température en cours.

Vitesse du vent :	7.5 km/h
Direction du vent :	23.5 deg.
Température :	17.2 deg. C

La page affiche ensuite la puissance utile de l'éolienne telle qu'elle est lue dans le S7-1200.

Puissance utile :	1000 kW
-------------------	---------

La partie suivante permet la commande manuelle de l'éolienne qui se substitue à la commande automatique normale par le S7-1200. Elle comprend :

- **Manual override (Manuel prioritaire) :** active le mode manuel de la turbine. Le programme utilisateur STEP 7 requiert que le mode manuel soit sur vrai pour utiliser l'un des réglages manuels pour la vitesse d'éolienne, le lacet ou le pas quelconque.

Priorité manuelle : On	Rég. : Oui
Vitesse de l'éolienne :	15 RPM

- **Lacet manuel prioritaire :** active la priorité manuelle du réglage du lacet et un réglage manuel pour le lacet. Le programme utilisateur STEP 7 n'applique le réglage du lacet que si Manuel prioritaire et Lacet manuel prioritaire sont tous deux vrais.

Lacet manuel prioritaire : On	Rég. : Oui
Lacet de l'éolienne :	5.2 deg.

- **Pas manuel prioritaire :** active la priorité manuelle pour le pas des pales. Le programme utilisateur STEP 7 n'applique le réglage du pas des pales que si Manuel prioritaire et Pas manuel prioritaire sont tous deux vrais.

Pas manuel prioritaire : On	Rég. : Oui
Pas des pales :	4.5 deg.

La page HTML comprend un bouton Soumettre pour transmettre les réglages de priorité à l'automate.

Soumettre les paramètres et valeurs de priorité manuelle

Le champ d'entrée de freinage utilisateur fournit une valeur manuelle pour un pourcentage de freinage. Le programme utilisateur STEP 7 accepte la valeur de freinage même si le mode manuel n'est pas activé.

Freinage :	2.5 %
------------	-------

En outre, la page HTML utilise une commande AWP pour écrire la variable spéciale (Page 1087) contenant l'identification de l'utilisateur qui accède à la page dans une variable de la table de variables API.

12.7.9.2 Lecture et affichage des données de l'automate

La page HTML de contrôle d'éolienne à distance utilise de nombreuses commandes AWP pour lire des données dans l'automate (Page 1082) et les afficher sur la page. Considérons, par exemple, le code HTML pour l'affichage de la puissance utile, comme illustré dans cette partie de l'exemple de page Web :



Exemple de code HTML

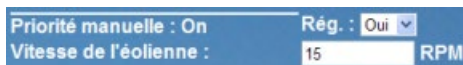
L'extrait suivant de la page HTML de contrôle d'éolienne à distance affiche le texte ""Energie produite :"" dans la cellule gauche d'une ligne du tableau et lit la variable pour la puissance utile puis l'affiche dans la cellule droite de la ligne du tableau avec l'abréviation pour kilowatts, kW.

La commande AWP :=`"Bloc_données_1".PuissanceUtile`: exécute l'opération de lecture. Notez que les blocs de données sont référencés par leur nom et pas par leur numéro (c'est-à-dire `"Bloc_données_1"` et non `"DB1"`).

```
<tr style="height:2%;">
<td>
<p>Puissance utile :</p>
</td>
<td>
<p style="margin-bottom:5px;"> :="Bloc_données_1".PuissanceUtile:
kW</p>
</td>
</tr>
```

12.7.9.3 Utilisation d'un type Enum

La page HTML de contrôle d'éolienne à distance utilise des types Enum pour trois instances dans lesquelles la page HTML affiche "ON" ou "OFF" pour une valeur booléenne et pour lesquelles l'utilisateur définit une valeur booléenne. Le type Enum pour "ON" correspond à une valeur de 1 et le type Enum pour "OFF" correspond à une valeur de 0. Considérons, par exemple, le code HTML pour la lecture et l'écriture du paramètre d'activation du manuel prioritaire dans la valeur `"Bloc_données_1".ActiverManuelPrioritaire` à l'aide d'un type Enum :



Exemple de code HTML

Les extraits suivants de la page HTML de contrôle d'éolienne à distance montrent comment déclarer un type Enum appelé "EtatPriorité" ayant les valeurs "Off" et "On" pour 0 et 1 et comment affecter une référence de type Enum à EtatPriorité pour la variable booléenne ActiverManuelPrioritaire dans le bloc de données nommé "Bloc_données_1".

```
<!-- AWP_In_Variable
Name=' "Bloc_données_1".ActiverManuelPrioritaire' Enum="EtatPriorité"
-->
```

```
<!-- AWP_Enum_Def Name="EtatPriorité" Values='0:"Off",1:"On"' -->
```

La où la page HTML comprend un champ d'affichage dans une cellule de tableau pour l'état en cours de ManualOverrideEnable, elle n'utilise qu'une commande normale de lecture de variable mais grâce à l'utilisation du type Enum préalablement déclaré et référencé, la page affiche "Off" ou "On" au lieu de 0 ou 1.

```
<td style="width:24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Priorité manuelle :
:="Bloc_données_1".ActiverManuelPrioritaire:</p>
</td>
```

La page HTML inclut une liste de sélection déroulante pour que l'utilisateur puisse changer la valeur de ActiverManuelPrioritaire. La liste de sélection utilise les textes "Oui" et "Non" à afficher dans les listes de sélection. Grâce à l'utilisation du type Enum, "Oui" est corrélé à la valeur "On" du type Enum et "Non" est corrélé à la valeur "Off". La sélection vide laisse la valeur de ActiverManuelPrioritaire telle quelle.

```
<select name=' "Bloc_données_1".ActiverManuelPrioritaire'>
<option value=' : "Bloc_données_1".ActiverManuelPrioritaire: '>
</option>
<option value="On">Oui</option>
<option selected value="Off">Non</option>
</select>
```

La liste de sélection est incluse dans un formulaire sur la page HTML. Lorsque l'utilisateur clique sur le bouton Soumettre, la page transmet le formulaire, ce qui inscrit la valeur "1" dans la variable booléenne ActiverManuelPrioritaire du bloc Bloc_données_1 si l'utilisateur a sélectionné "Oui" ou la valeur "0" si l'utilisateur a sélectionné "Non".

12.7.9.4 Ecriture d'une entrée utilisateur dans l'automate

La page HTML de contrôle d'éolienne à distance comprend plusieurs commandes AWP pour l'écriture de données dans le contrôleur (Page 1083). La page HTML déclare des variables AWP_In_Variables pour les variables booléennes de sorte qu'un utilisateur avec des privilèges de modification de variables peut placer l'éolienne en mode manuel et activer la substitution de valeurs manuelles pour la vitesse de l'éolienne, le lacet et/ou le pas des pales. La page utilise également des variables AWP_In_Variables pour permettre à un utilisateur avec des privilèges de modification de variables de régler par la suite des valeurs à virgule flottante pour la vitesse, le lacet et le pas de l'éolienne ainsi que pour le pourcentage de freinage. La page utilise une commande d'envoi de formulaire HTTP pour écrire les variables AWP_In_Variables dans l'automate.

Considérons, par exemple, le code HTML pour le réglage manuel de la valeur de freinage :

Exemple de code HTML

L'extrait suivant de la page HTML de contrôle d'éolienne à distance déclare d'abord pour "Bloc_données_1" une variable AWP_In_Variable qui permet à la page HTML d'écrire dans n'importe quelle variable du bloc de données "Bloc_données_1". La page affiche le texte "Freinage :." dans la cellule gauche d'une ligne du tableau. La cellule droite de la ligne de tableau correspond au champ qui accepte l'entrée utilisateur pour la variable "Freinage" de "Bloc_données_1". La valeur entrée par l'utilisateur se situe dans un formulaire HTML qui utilise la méthode HTML "POST" pour envoyer les données textuelles entrées à la CPU. La page lit alors la valeur de freinage en cours dans l'automate et l'affiche dans le champ d'entrée de données.

Un utilisateur avec des privilèges de modification de variables peut alors utiliser cette page pour écrire une valeur de freinage dans le bloc de données de la CPU qui commande le freinage.

```

<!-- AWP_In_Variable Name='Bloc_données_1' -->
...
<tr style="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Freinage :</p></td>
<td>
<form method="POST">
<p><input name='Bloc_données_1'.Freinage' size="10" type="text">
%</p>
</form>
</td>
</tr>

```

Remarque

Notez que, si une page personnalisée comporte un champ d'entrée de données pour une variable de bloc de données inscriptible, l'utilisateur doit écrire la chaîne entre apostrophes lorsqu'il l'entre dans le champ.

Remarque

Notez que si vous déclarez l'ensemble d'un bloc de données dans une déclaration AWP_In_Variable (par exemple, <!-- AWP_In_Variable Name=""Data_block_1" -->), il est possible d'écrire dans chaque variable de ce bloc de données à partir de la page Web personnalisée. Utilisez cette méthode lorsque vous voulez que toutes les variables d'un bloc de données soient inscriptibles. Sinon, si vous voulez qu'il ne soit possible d'écrire que dans certaines variables spécifiques du bloc de données à partir de la page Web personnalisée, déclarez spécifiquement les variables concernées à l'aide d'une déclaration telle que <!-- AWP_In_Variable Name=""Data_block_1".Braking' -->

12.7.9.5 Ecriture d'une variable spéciale

La page Web de contrôle d'éolienne à distance écrit la variable spéciale SERVER:current_user_id dans une variable API de la CPU, à condition que cet utilisateur possède les droits de modification. Dans ce cas, la valeur de la variable API contient l'ID de l'utilisateur qui accède à la page Web de contrôle d'éolienne à distance.

La page Web écrit la variable spéciale dans l'API et ne nécessite pas d'interface utilisateur.

Exemple de code HTML

```
<!-- AWP_In_Variable Name="SERVER:current_user_id"
Use="ID_utilisateur"-->
```

12.7.9.6 Référence : listage HTML de la page Web de contrôle d'éolienne à distance

Eolienne.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!--
Ce programme de test simule une page Web pour le contrôle-commande
d'une éolienne.
Variables API et variables de blocs de données requises dans STEP 7
:

Variable API :
ID_utilisateur : Int

Blocs de données :
Bloc_données_1

Variables dans Data_Block_1 :

NuméroEolienne : Int
VitesseVent : Real
DirectionVent : Real
Température : Real
PuissanceUtile : Real
ActiverManuelPrioritaire : Bool
```

```
VitesseEolienne : Real
LacetPrioritaire : Bool
Lacet : Real
PasPrioritaire : Bool
Pas : Real
Freinage : Real
```

La page Web personnalisée affiche les valeurs actuelles pour les données API et fournit une liste de sélection pour définir les trois valeurs booléennes à l'aide d'une affectation de type d'énumération. Le bouton "Soumettre" permet de transmettre les valeurs booléennes sélectionnées ainsi que les champs de saisie de données pour la vitesse (TurbineSpeed), le lacet (Yaw) et le pas (Pitch). La valeur pour le freinage peut être définie sans utiliser le bouton "Soumettre".

Aucun programme STEP 7 réel n'est nécessaire pour utiliser cette page. Théoriquement, le programme STEP 7 agirait uniquement sur les valeurs de vitesse (TurbineSpeed), lacet (Yaw) et pas (Pitch) si les valeurs booléennes associées étaient à 1. La seule exigence de STEP 7 est d'appeler l'instruction WWW avec les numéros des blocs de données générés pour cette page.

```
-->
<!-- AWP_In_Variable Name='Bloc_données_1' -->
<!-- AWP_In_Variable
Name='Bloc_données_1'.ActiverManuelPrioritaire' Enum='EtatPriorité'
-->
<!-- AWP_In_Variable Name='Bloc_données_1'.PasPrioritaire'
Enum='EtatPriorité' -->
<!-- AWP_In_Variable Name='Bloc_données_1'.LacetPrioritaire'
Enum='EtatPriorité' -->
<!-- AWP_In_Variable Name='SERVER:current_user_id'
Use='ID_utilisateur'-->
<!-- AWP_Enum_Def Name='EtatPriorité' Values='0:"Off",1:"On"' -->
```

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<link rel="stylesheet" href="Eolienne.css">
<title>Contrôle d'éolienne à distance</title>
</head>
<body>
<table cellpadding="0" cellspacing="2">
<tr style="height: 2%;">
<td colspan="2">
<h2>Contrôle d'éolienne à distance : Eolienne
#:="Bloc_données_1".NuméroEolienne:</h2>
</td>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Vitesse du vent
</p></td>
<td><p> :="Bloc_données_1".VitesseVent : km/h</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Direction du vent :</p></td>
```

```

<td><p> :="Bloc_données_1".DirectionVent : deg.</p></td>
</tr>

<tr style="height: 2%;"><td style="width:
25%;"><p>Température :</p></td>
<td><p> :="Bloc_données_1".Température : deg. C</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Puissance utile :</p></td>
<td><p style="margin-bottom:5px;"> :="Bloc_données_1".PuissanceUtile
: kW</p>
</td>
</tr>

<form method="POST" action="">
<tr style="height: 2%;" >
<td style="width=25%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Priorité manuelle :
:="Bloc_données_1".ActiverManuelPrioritaire:</p>
</td>
<td class="Text">Rég. :

<select name='"Bloc_données_1".ActiverManuelPrioritaire'>
<option value=':="Bloc_données_1".ActiverManuelPrioritaire'>
</option>
<option value="On">Oui</option>
<option value="Off">Non</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;"><td style="width:
25%;"><p>Vitesse de l'éolienne :</p></td>
<td>
<p style="margin-bottom:5px;"><input
name='"Bloc_données_1".VitesseEolienne' size="10"
value=':="Bloc_données_1".VitesseEolienne :' type="text"> RPM</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Lacet manuel prioritaire :
:="Bloc_données_1".LacetManuelPrioritaire : </p>
</td>
<td class="Text">Rég. :

<select name='"Bloc_données_1".LacetManuelPrioritaire'>
<option value=':="Bloc_données_1".LacetManuelPrioritaire : '>
</option>
<option value="On">Oui</option>
<option value="Off">Non</option>

```

```

</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Lacet de l'éolienne :</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name="Bloc_données_1".Lacet'
size="10" value=":"="Bloc_données_1".Lacet:' type="text"> deg.</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Pas manuel prioritaire : :="Bloc_données_1".PasManuelPrioritaire:
</p>
</td>
<td class="Text">Rég. :

<select name="Bloc_données_1".PasManuelPrioritaire'>
<option value=":"="Bloc_données_1".PasManuelPrioritaire :> </option>
<option value="On">Oui</option>
<option value="Off">Non</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width=25%; border-bottom-style: Solid; border-bottom-
width: 2px; border-bottom-color: #ffffff;">
<p>Pas des pales :</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name="Bloc_données_1".Pas'
size="10" value=":"="Bloc_données_1".Pas :' type="text"> deg.</p>
</td>

</tr>
<tr style="height: 2%;">
<td colspan="2">
<input type="submit" value="Soumettre les paramètres et valeurs de
priorité manuelle">
</td>
</tr>
</form>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;"><p>Freinage :</p></td>
<td>
<form method="POST" action="">

```

```
<p> <input name='"Bloc_données_1".Freinage' size="10"
value':"Bloc_données_1".Freinage:' type="text"> %</p>
</form>
</td>
</tr>
<tr><td></td></tr>

</table>
</body>
</html>
```

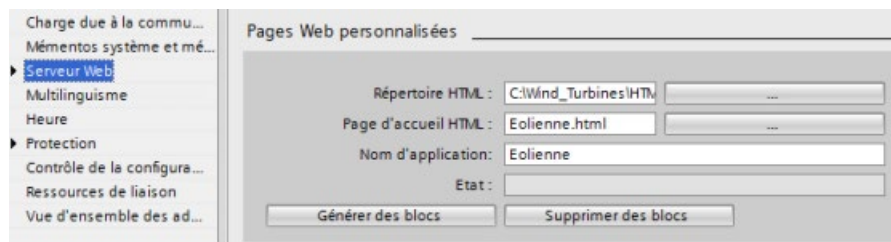
Eolienne.css

```
BODY {
  background-image: url('./Wind_turbine.jpg');
  background-position: 0% 0%;
  background-repeat: no-repeat;
  background-size: cover;
}
H2 {
  font-family: Arial;
  font-weight: bold;
  font-size: 14.0pt;
  color: #FFFFFF;
  margin-top: 0px;
  margin-bottom: 10px;
}
P {
  font-family: Arial;
  font-weight: bold;
  color: #FFFFFF;
  font-size: 12.0pt;
  margin-top: 0px;
  margin-bottom: 0px;
}
TD.Text {
  font-family: Arial;
  font-weight: bold;
  color: #FFFFFF;
  font-size: 12.0pt;
  margin-top: 0px;
  margin-bottom: 0px;
}
```

12.7.9.7 Configuration de l'exemple de page Web dans STEP 7

Pour inclure la page HTML de contrôle d'éolienne à distance en tant que page Web personnalisée pour le S7-1200, vous configurez les données concernant la page HTML dans STEP 7 et vous créez des blocs de données à partir de la page HTML.

Accédez aux propriétés CPU du S7-1200 qui commande l'éolienne et entrez les informations de configuration dans les propriétés des pages Web personnalisées du serveur Web :



Champs de configuration

- HTML directory (répertoire HTML) : Ce champ indique le nom de chemin complet du dossier dans lequel se trouve la page par défaut (ou page d'accueil) sur l'ordinateur. Le bouton "..." vous permet de naviguer jusqu'au dossier dont vous avez besoin.
- Default HTML page (page HTML par défaut) : Ce champ indique le nom de fichier de la page par défaut ou page d'accueil de l'application HTML. Le bouton "..." vous permet de sélectionner le fichier dont vous avez besoin. Dans cet exemple, Eolienne.html est la page HTML par défaut. L'exemple de contrôle d'éolienne à distance est constitué d'une seule page mais, dans d'autres applications personnalisées, la page par défaut peut appeler d'autres pages au moyen de liens sur la page par défaut. Dans le code HTML, la page par défaut doit référencer les autres pages par rapport au dossier source HTML.
- Application name (nom de l'application) : Ce champ optionnel contient le nom que le navigateur Web reporte dans le champ d'adresse quand il affiche la page. Il s'agit de "Contrôle d'éolienne à distance" dans cet exemple, mais ce pourrait être n'importe quel nom.

Aucun autre champ ne nécessite de configuration.

Étapes finales

Pour utiliser la page de contrôle d'éolienne à distance telle que configurée, générez les blocs, programmez l'instruction WWW (Page 1099) avec, comme paramètre d'entrée, le numéro du DB de commande généré, chargez les blocs de programme dans la CPU et faites passer la CPU à l'état MARCHE.

Plus tard, lorsqu'un opérateur accédera aux pages Web standard du S7-1200 qui commande l'éolienne, la page de contrôle d'éolienne à distance sera accessible via le lien "Pages utilisateur" dans la barre de navigation. Cette page offre maintenant les moyens pour réaliser un contrôle-commande de l'éolienne.

12.7.10 Configuration de pages Web personnalisées dans plusieurs langues

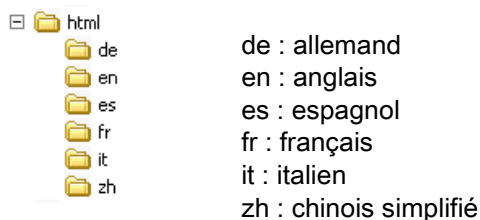
Le serveur Web vous offre la possibilité de configurer des pages Web personnalisées dans les langues suivantes :

- allemand (de)
- anglais (en)
- espagnol (es)
- français (fr)
- italien (it)
- Chinois simplifié (zh)

Pour ce faire, vous configurez vos pages HTML dans une structure de dossiers (Page 1115) qui correspond aux différentes langues et vous définissez un cookie spécifique nommé "siemens_automation_language" depuis vos pages (Page 1115). Le serveur Web réagit à ce cookie et commute sur la page par défaut dans le dossier de langue correspondant.

12.7.10.1 Création de la structure de dossiers

Pour fournir des pages Web personnalisées dans plusieurs langues, vous créez une structure de dossiers sous votre répertoire HTML. Les noms de dossier à deux lettres sont spécifiques et doivent être attribués comme suit :



Vous pouvez inclure au même niveau tout autre dossier dont vos pages ont besoin, par exemple des dossiers pour les images ou les scripts.

Vous pouvez créer n'importe quel sous-ensemble des dossiers de langue. Vous n'êtes pas obligé d'inclure la totalité des six langues. A l'intérieur des dossiers de langue, vous créez et programmez vos pages HTML dans la langue appropriée.

12.7.10.2 Programmation du changement de langue

Le serveur Web commute entre les différentes langues grâce à l'utilisation d'un cookie appelé "siemens_automation_language". Il s'agit d'un cookie défini et activé dans les pages HTML, que le serveur Web interprète pour afficher une page dans la langue appropriée provenant du dossier de langue de même nom. La page HTML doit inclure du code JavaScript pour définir ce cookie à l'un des identificateurs de langue prédéfinis : "de", "en", "es", "fr", "it" ou "zh".

Par exemple, si la page HTML définit le cookie à "de", le serveur Web commute dans le dossier "de" et affiche la page ayant le nom de page HTML par défaut tel que défini dans la configuration STEP 7 (Page 1119).

Exemple

L'exemple suivant utilise une page HTML par défaut nommée "langswitch.html" dans chaque dossier de langue. Le répertoire HTML contient également un dossier nommé "script". Le dossier script comprend un fichier JavaScript de nom "lang.js". Chaque page langswitch.html utilise ce JavaScript pour définir le cookie de langue, "siemens_automation_language".

HTML pour "langswitch.html" dans le dossier "en"

L'en-tête de la page HTML définit la langue à "anglais", le jeu de caractères à UTF-8, ainsi que le chemin allant au fichier JavaScript lang.js.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Language switching english page</title>
<script type="text/javascript" src="script/lang.js" ></script>
```

Le corps du fichier utilise une liste de sélection permettant à l'utilisateur de choisir entre l'allemand et l'anglais. L'anglais ("en") est la langue sélectionnée par défaut. Lorsque l'utilisateur change de langue, la page appelle la fonction JavaScript DoLocalLanguageChange() avec la valeur de l'option sélectionnée.

```
<!-- Language Selection -->
<table>
  <tr>
    <td align="right" valign="top" nowrap>
      <!-- change language immediately on selection change -->
      <select name="Language"
        onchange="DoLocalLanguageChange(this)"
        size="1">
        <option value="de" >German</option>
        <option value="en" selected >English</option>
      </select>
    </td>
  </tr>
</table><!-- Language Selection End-->
```


HTML pour "langswitch.html" dans le dossier "de"

L'en-tête pour la page langswitch.html allemande est identique à celui de l'anglais, si ce n'est que la langue définie est l'allemand.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="de"><meta http-
equiv="Content-Type" content="text/html; charset=utf-8">
<title>Sprachumschaltung Deutsche Seite</title>
<script type="text/javascript" src="script/lang.js" ></script>
</head>
```

Le code HTML dans la page allemande est identique à celui de la page anglaise, si ce n'est que la langue sélectionnée par défaut est l'allemand ("de").

```
<!-- Language Selection -->
<table>
  <tr>
    <td align="right" valign="top" nowrap>
      <!-- change language immediately on change of the selection -
->
      <select name="Language"
        onchange="DoLocalLanguageChange(this)"
        <size="1">
          <option value="de" selected >Deutsch</option>
          <option value="en" >Englisch</option>
        </select>
      </td>
    </tr>
  </table><!-- Language Selection End-->
```

JavaScript "lang.js" dans le dossier "script"

La fonction "DoLocalLanguageChange()" se trouve dans le fichier lang.js. Cette fonction appelle la fonction "SetLangCookie()", puis recharge la fenêtre qui affiche la page HTML.

La fonction "SetLangCookie()" construit une affectation qui affecte la valeur de la liste sélectionnée au cookie "siemens_automation_language" du document. Elle définit également le chemin vers l'application de sorte que la page commutée, et non la page demandeuse, reçoit la valeur du cookie.

En option, la page peut définir une valeur d'expiration pour le cookie dans la section mise en commentaire.

```
function DoLocalLanguageChange(oSelect) {
    SetLangCookie(oSelect.value);
    top.window.location.reload();
}
function SetLangCookie(value) {
    var strval = "siemens_automation_language=";
    // Ceci est le cookie par lequel le serveur Web
    // détecte la langue désirée
    // Ce nom est requis par le serveur Web.
    strval = strval + value;
    strval = strval + "; path=/ ";
    // Définissez le chemin de l'application sinon le chemin
    // serait pris pour la page ayant posé la requête
    // et cette page ne recevrait pas le cookie.
    /* OPTIONAL
    Utilisez une expiration si ce cookie doit vivre plus
longtemps
    que la session en cours du navigateur :
    var now = new Date();
    var endttime = new Date(now.getTime() + expiration);
    strval = strval + "; expires=" +
        endttime.toGMTString() + ";";
    */
    document.cookie = strval;
}
```

Remarque

Si votre implémentation de page Web personnalisée comprend des fichiers HTML à l'intérieur des dossiers de langage spécifique (en, de, par exemple) et également des fichiers HTML qui ne sont pas situés dans les dossiers de langage spécifique, remarquez que vous ne pouvez pas définir de types enum avec la commande AWP_Enum_Def dans les fichiers aux deux emplacements. Si vous utilisez des enums, vous devez les définir soit à l'intérieur des fichiers dans les dossiers de langage spécifique ou à l'intérieur des fichiers situés en dehors des dossiers de langage spécifique. Vous ne pouvez pas effectuer des déclarations d'enum dans les fichiers aux deux emplacements.

12.7.10.3 Configuration de STEP 7 pour qu'il utilise une structure de page multilingue

La procédure pour configurer des pages Web personnalisées multilingues est similaire à la procédure générale de configuration de pages Web personnalisées (Page 1097). Toutefois, lorsque vous avez créé des dossiers pour les langues, vous définissez votre répertoire HTML comme étant le dossier qui contient les dossiers de langue individuels et non comme étant l'un des dossiers de langue.

Lorsque vous sélectionnez la page HTML par défaut, vous naviguez jusqu'au dossier de langue et vous y sélectionnez la page HTML qui doit servir de page d'accueil. Ensuite, lorsque vous générez les blocs et les chargez dans la CPU, le serveur Web affiche la page d'accueil dans le dossier de langue que vous avez configuré.

Par exemple, si la structure de dossiers présentée ici était dans C:\, la définition du répertoire HTML serait C:\html et, si l'anglais devait être la page d'affichage initiale, vous navigueriez jusqu'à en\langswitch.html pour y définir la page HTML par défaut.



12.7.11 Commande avancée de pages Web personnalisées

Lorsque vous générez les blocs de données pour vos pages Web personnalisées, STEP 7 crée un DB de commande qu'il utilise pour gérer l'interaction avec les pages personnalisées et leur affichage. STEP 7 crée également un jeu de DB de fragment qui représentent les pages individuelles. Dans les circonstances normales, vous n'avez pas besoin de connaître la structure du DB de commande ni de savoir comment le manipuler.

Si vous voulez activer et désactiver une application Web ou manipuler des fragments manuels individuels, vous utilisez les variables du DB de commande et l'instruction WWW.

Structure du DB de commande

Le DB de commande est une structure de données étendue à laquelle vous pouvez accéder lors de la programmation de votre programme utilisateur STEP 7. Seules certaines des variables du bloc de données de commande sont décrites ici.

Structure Commandstate

"Commandstate" est une structure qui contient des commandes globales et des états globaux pour le serveur Web.

Commandes globales dans la structure "Commandstate"

Les commandes globales s'appliquent au serveur Web en général. Vous pouvez désactiver le serveur Web ou le redémarrer à partir des paramètres du DB de commande.

Variable du bloc	Type de données	Description
init	BOOL	Evaluer le DB de commande et initialiser l'application Web
deactivate	BOOL	Désactiver l'application Web

États globaux dans la structure Commandstate

Les états globaux s'appliquent au serveur Web en général et contiennent des informations d'état sur l'application Web.

Variable du bloc	Type de données	Description
initializing	BOOL	L'application Web lit le DB de commande.
error	BOOL	Initialisation de l'application Web impossible
deactivating	BOOL	L'application Web est en cours de désactivation.
deactivated	BOOL	L'application Web est désactivée.
initialized	BOOL	L'application Web est initialisée.
last_error	INT	Dernière erreur renvoyée par un appel de l'instruction WWW (Page 1099) lorsque le code renvoyé par WWW est 16#0010 : 16#0001 : la structure du DB de fragment est incohérente 16#0002 : le nom de l'application existe déjà 16#0003 : pas de ressources (mémoire) 16#0004 : la structure du DB de commande est incohérente 16#0005 : DB de fragment non disponible 16#0006 : DB de fragment pas pour AWP 16#0007 : les données de l'énumération sont incohérentes 16#000D : conflit de taille du DB de commande

Table de requête

La table de requête est un tableau de structures contenant des commandes et des états s'appliquant à des DB de fragment individuels. Si vous avez créé des fragments de type "manuel" avec la commande AWP_Start_Fragment (Page 1091), le programme utilisateur STEP 7 doit gérer ces pages par le biais du DB de commande. Les états de requête sont en lecture seule et fournissent des informations sur le fragment en cours. Vous utilisez les commandes de requête pour commander le fragment en cours.

Variable du bloc	Type de données	Description
requesttab	ARRAY [1 .. 4] OF STRUCT	Tableau de structures pour la commande de DB de fragment individuels. Le serveur Web peut traiter jusqu'à quatre fragments à un moment donné. L'indice de tableau pour un fragment particulier est arbitraire lorsque le serveur Web traite plusieurs fragments ou des fragments provenant de plusieurs sessions de navigateur.

Éléments de la structure requesttab

Variable du bloc	Type de données	Description
page_index	UINT	Numéro de la page Web en cours
fragment_index	UINT	Numéro du fragment en cours. Peut être défini à un fragment différent.
// Commandes de requête		
continue	BOOL	Activer la page/le fragment en cours pour l'émission et poursuivre avec le fragment suivant.
repeat	BOOL	Activer la page/le fragment en cours pour la ré-émission et poursuivre avec le même fragment.
abort	BOOL	Fermer la connexion http sans émission.
finish	BOOL	Envoyer ce fragment ; la page est complète, ne pas traiter de fragments supplémentaires.
// Etats de requête		Les états de requête sont en lecture seule.
idle	BOOL	Rien à faire, mais actif
waiting	BOOL	Le fragment attend d'être activé.
sending	BOOL	Le fragment émet.
aborting	BOOL	L'utilisateur a annulé la requête en cours.

Fonctionnement

A chaque fois que votre programme modifie le DB de commande, il doit appeler l'instruction WWW avec, comme paramètre, le numéro du DB de commande modifié. Les commandes globales et les commandes de requête prennent effet lorsque le programme utilisateur STEP 7 exécute l'instruction WWW (Page 1099).

Le programme utilisateur STEP 7 peut définir l'indice `fragment_index` explicitement, le serveur Web traitant alors le fragment spécifié avec une commande de requête. Sinon, le serveur Web traite le fragment en cours pour la page en cours lorsque l'instruction WWW s'exécute.

Les techniques possibles d'utilisation de "fragment_index" incluent :

- Traiter le fragment en cours : Laissez "fragment_index" inchangé et activez la commande "continue".
- Sauter le fragment en cours : Définissez "fragment_index" à 0 et activez la commande "continue".
- Remplacer le fragment en cours par un autre fragment : Définissez "fragment_index" au nouvel ID de fragment et activez la commande "continue".

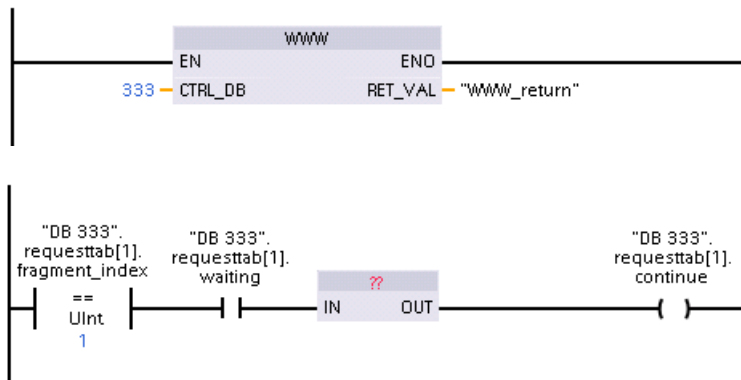
Pour vérifier les états globaux ou les états de requête qui auraient pu changer, le programme utilisateur STEP 7 doit appeler l'instruction WWW pour évaluer les valeurs en cours de ces états. Une façon de faire typique serait d'appeler l'instruction WWW périodiquement jusqu'à ce qu'un état spécifique apparaisse.

Remarque

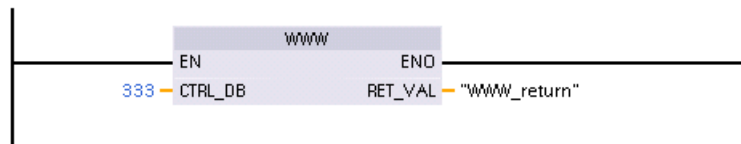
Si le programme utilisateur STEP 7 active plus d'une commande de requête, l'instruction WWW n'en traite qu'une dans cet ordre : abort, finish, repeat, continue. L'instruction WWW efface toutes les commandes de requête après le traitement.

Exemples

L'exemple suivant montre un programme utilisateur STEP 7 qui vérifie si un fragment d'ID égal à 1 est en état d'attente, après un appel antérieur de l'instruction WWW. Il peut également attendre que d'autres conditions spécifiques de l'application apparaissent. Puis il exécute tout traitement nécessaire pour le fragment, par exemple définir des variables du bloc de données, effectuer des calculs ou d'autres tâches spécifiques de l'application. Ensuite, il met à 1 le memento de "continue" afin que le serveur Web exécute ce fragment.



Lorsque le programme appelle l'instruction WWW avec ce DB de commande modifié, la page Web personnalisée avec ce fragment peut être affichée dans le navigateur Web.



Notez qu'il s'agit d'un exemple simplifié ; le fragment à contrôler pourrait se trouver dans n'importe laquelle des quatre structures requesttab du tableau. Votre programme doit traiter la totalité des quatre structures requesttab.

12.8 Contraintes

Les considérations informatiques suivantes peuvent influencer sur votre utilisation du serveur Web :

- En règle générale, vous devez utiliser l'adresse IP de la CPU ou l'adresse IP d'un routeur sans fil avec un numéro de port pour accéder aux pages Web standard ou aux pages Web personnalisées. Si votre navigateur Web n'autorise pas la connexion directe à une adresse IP, contactez votre administrateur informatique. Si votre réglementation locale prend en charge le service DNS, vous pouvez vous connecter à l'adresse IP via une entrée DNS désignant cette adresse.
- Les pare-feu, les réglages de proxy et autres restrictions spécifiques du site peuvent également restreindre l'accès à la CPU. Contactez votre administrateur informatique pour résoudre ces problèmes.
- Les pages Web standard utilisent des scripts JavaScript et des cookies. Si les scripts JavaScript ou les cookies sont désactivés par les réglages de votre navigateur Web, activez-les. Si vous ne pouvez pas les activer, cela limitera certaines fonctionnalités (Page 1125). L'utilisation de scripts JavaScript et de cookies dans les pages Web personnalisées est facultative. S'ils sont utilisés, vous devez les activer dans votre navigateur.
- Le serveur Web prend en charge le protocole SSL (Secure Sockets Layer). Vous pouvez accéder aux pages Web standard et aux pages Web personnalisées avec une URL `http://ww.xx.yy.zz` ou `https://ww.xx.yy.zz`, "ww.xx.yy.zz" représentant l'adresse IP de la CPU.
- Siemens fournit un certificat de sécurité pour sécuriser l'accès au serveur Web. Vous pouvez télécharger et importer ce certificat dans les options Internet de votre navigateur Web (Page 1126) à partir de la page Web standard d'introduction (Page 1054). Si vous décidez de ne pas importer ce certificat, vous recevrez une invitation à vérifier la sécurité à chaque fois que vous accéderez au serveur Web avec `https://`.

Nombre de connexions

Le serveur Web prend en charge 30 connexions actives au plus. Diverses actions consomment ces 30 connexions, selon le navigateur Web que vous utilisez et le nombre d'objets différents par page (fichiers .css, images, fichiers .html supplémentaires). Certaines connexions persistent tant que le serveur Web affiche la page alors que d'autres ne persistent pas après la connexion initiale.

Si, par exemple, vous utilisez certaines versions de Mozilla Firefox qui prennent en charge six connexions persistantes au maximum, vous pouvez utiliser cinq instances de navigateur ou d'onglet de navigateur avant que le serveur Web ne se mette à couper des connexions. Si une page n'utilise pas les six connexions, vous pouvez avoir des instances de navigateur ou d'onglet de navigateur supplémentaires.

N'oubliez pas que le nombre de connexions actives peut affecter les performances des pages.

Remarque

Déconnectez-vous avant de fermer le serveur Web

Si vous êtes connecté au serveur Web, assurez-vous de vous déconnecter avant de fermer le navigateur web. Le serveur Web prend en charge au plus 7 connexions simultanées.

12.8.1 Utilisation de JavaScript

Les pages Web standard utilisent des pages HTML, des JavaScripts et des cookies. Si votre site limite l'utilisation des JavaScripts et des cookies, activez-les afin que les pages fonctionnent correctement. Si vous ne pouvez pas activer JavaScript pour votre navigateur Web, les pages Web standard ne pourront pas fonctionner. Envisagez alors d'utiliser les pages de base qui n'utilisent pas JavaScript.

Voir aussi

Disposition des pages Web standard (Page 1048)

12.8.2 Restriction des fonctionnalités lorsque les options Internet n'autorisent pas les cookies

Les restrictions suivantes s'appliquent si vous désactivez les cookies dans votre navigateur Web :

- Vous ne pouvez pas ouvrir de session.
- Vous ne pouvez pas modifier le paramétrage de la langue.
- Vous ne pouvez pas passer du temps UTC au temps API. En l'absence de cookies, toutes les heures sont en temps UTC.

12.8.3 Règles de saisie des noms de variable et des valeurs

Tenez compte des conventions suivantes lors de l'utilisation des pages standard État des variables (Page 1068) et Tables de visualisation (Page 1069) :

- Pour modifier toute la valeur d'une variable DTL, par exemple, "Data_block_1_.DTL_tag, utilisez la syntaxe DTL suivante pour la valeur de forçage : DTL#YYYY-MM-DD-HH-MM-SS[.ssssssss]
- En cas d'utilisation de la notation exponentielle pour saisir une valeur de type de données Real ou LReal :
 - Pour entrer la valeur d'un réel (Real ou LReal) avec un exposant positif (tel que +3.402823e+25), utilisez l'un des formats suivants :
 - +3.402823e25
 - +3.402823e+25
 - Pour entrer la valeur d'un réel (Real ou LReal) avec un exposant négatif (tel que +3.402823e-25), entrez la valeur comme suit :
 - +3.402823e-25
 - Veillez à ce que la mantisse de la valeur réelle en notation exponentielle comprenne un séparateur décimal. En l'absence de séparateur décimal, vous obtiendrez une valeur entière inattendue. Ainsi, entrez -1.0e8 et non -1e8.
- Les valeurs LReal ne peuvent comporter que 15 chiffres (quelle que soit la position du séparateur décimal). La saisie de plus de 15 chiffres entraîne une erreur d'arrondi.

Limitations concernant les pages État des variables et Table de visualisation :

- Le nombre maximum de caractères pour l'URL est de 2083. Vous pouvez voir l'URL représentant la page en cours dans la barre d'adresse de votre navigateur.
- Pour le format d'affichage des caractères, si les valeurs en cours de la CPU ne sont pas des caractères ASCII valides tels qu'interprétés par le navigateur, la page affiche le caractère précédé d'un signe dollar (\$).

12.8.4 Importation du certificat de sécurité Siemens

Vous pouvez télécharger le certificat de sécurité Siemens dans vos options Internet. Avec le certificat, vous ne devez pas fournir de vérification de sécurité lorsque vous entrez <https://ww.xx.yy.zz> dans votre navigateur Web, "ww.xx.yy.zz" étant l'adresse IP de l'appareil. Vous n'avez pas besoin de télécharger et d'installer le certificat si vous utilisez une URL <http://> et non une URL <https://>.

Téléchargement du certificat

Utilisez le lien de téléchargement du certificat dans la page Introduction (Page 1054) pour télécharger le certificat de sécurité Siemens sur votre PC. La procédure varie selon le navigateur Web que vous utilisez.

Importation du certificat dans Internet Explorer

1. Cliquez sur le lien "Télécharger le certificat" dans la page d'introduction.
2. Dans la boîte de dialogue suivante, cliquez sur "Ouvrir" pour ouvrir le fichier.
3. Dans la boîte de dialogue "Certificat", cliquez sur le bouton "Installer le certificat" pour lancer l'assistant d'importation de certificat.
4. Cliquez sur "Suivant" dans la boîte de dialogue "Assistant Importation de certificat" pour définir le magasin de certificats.
5. Sélectionnez "Placer tous les certificats dans le magasin suivant" et cliquez sur le bouton "Parcourir".
6. Dans la boîte de dialogue "Sélectionner un magasin de certificats", sélectionnez "Autorités de certification racine tierce partie" et cliquez sur OK.
7. Cliquez sur "Suivant" puis sur "Terminer" pour achever l'Assistant Importation de certificat.

Importation du certificat dans Mozilla Firefox

1. Cliquez sur le lien "download certificate" dans la page Intro.
2. Lorsqu'un message vous le demande, cliquez sur "OK" pour accorder votre confiance à "S7-1200 Controller Family".

Dans les versions plus anciennes de Mozilla Firefox, vous devez enregistrer le fichier et exécuter l'assistant après avoir cliqué sur le lien de téléchargement du certificat :

1. Cliquez sur "Enregistrer le fichier" dans la boîte de dialogue "Ouverture de MiniWebCA_Cer.crt". Une boîte de dialogue "Téléchargements" apparaît.
2. Dans la boîte de dialogue "Téléchargements", double-cliquez sur "MiniWebCA_Cer.crt". Si vous avez tenté le téléchargement plus d'une fois, plusieurs copies sont montrées. Il suffit que vous double-cliquiez sur l'une quelconque des entrées "MiniWebCA_Cer.crt".
3. Cliquez sur "OK" si un message vous demande d'ouvrir un fichier exécutable.
4. Cliquez sur "Ouvrir" dans la boîte de dialogue "Ouverture de fichier - Avertissement de sécurité" si elle apparaît. Une boîte de dialogue "Certificate" s'ouvre.
5. Dans cette boîte de dialogue, cliquez sur le bouton "Install Certificate".
6. Suivez les instructions de l'assistant d'importation de certificat afin d'importer le certificat en laissant le système d'exploitation choisir automatiquement l'emplacement de stockage du certificat.
7. Si la boîte de dialogue "Avertissement de sécurité" apparaît, cliquez sur "Oui" pour confirmer l'installation du certificat.

Autres navigateurs

Suivez les conventions de votre navigateur Web pour importer et installer le certificat Siemens.

Une fois le certificat de sécurité Siemens "S7-1200 Controller Family" installé dans les options Internet pour le contenu de votre navigateur Web, vous ne devrez plus acquiescer de message de sécurité lorsque vous accéderez au serveur Web par <https://ww.xx.yy.zz>.

Remarque

Le certificat de sécurité reste tel quel malgré les redémarrages de la CPU, cependant, si vous modifiez l'adresse IP de l'appareil, vous devez télécharger un nouveau certificat si vous utilisez un navigateur autre qu'Internet Explorer ou Mozilla Firefox.

12.8.5 Importation de journaux en format CSV dans des versions non anglaises/américaines de Microsoft Excel

Les fichiers journaux sont en format CSV (valeurs séparées par des virgules). Vous pouvez ouvrir ces fichiers directement dans Excel à partir de la page Journaux lorsque votre système utilise la version anglaise ou américaine d'Excel. Toutefois, ce format n'est pas très représenté dans d'autres pays, car les virgules sont utilisées dans la notation numérique.

Pour les versions d'Excel autres que la version anglaise/américaine, procédez comme suit pour ouvrir un fichier journal que vous avez enregistré :

1. Ouvrez Excel et créez un classeur vide.
2. Dans le menu "Données > Données externes", sélectionnez la commande "Importer des données".
3. Naviguez jusqu'au fichier journal que vous voulez ouvrir et sélectionnez-le. L'Assistant Importation de texte est lancé.
4. Dans l'Assistant Importation de texte, changez l'option par défaut pour "Type de données d'origine" de "Largeur fixe" à "Délimité".
5. Cliquez sur le bouton Suivant.
6. Dans la boîte de dialogue Etape 2, cochez la case "Virgule" pour changer le type de séparateur de "Tabulation" à "Virgule".
7. Cliquez sur le bouton Suivant.
8. Dans la boîte de dialogue Etape 3, vous pouvez facultativement changer le format des données de MJA (mois/jour/année) à un autre format.
9. Exécutez les étapes restantes de l'Assistant Importation de texte pour importer le fichier.

Processeur de communication et Modbus TCP

13.1 Utilisation des interfaces de communication série

Deux modules de communication (CM) et un Communication Board (CB) fournissent l'interface pour les communications PtP :

- CM 1241 RS232 (Page 1635)
- CM 1241 RS422/485 (Page 1636)
- CB 1241 RS485 (Page 1633)

Vous pouvez connecter jusqu'à trois CM (de tout type) plus une CB pour un total de quatre interfaces de communication. Installez le CM à gauche de la CPU ou d'un autre CM. Installez la CB sur le devant de la CPU. Pour obtenir des informations sur l'installation et la désinstallation de modules, veuillez vous référer aux instructions d'installation (Page 68).

Les interfaces de communication série ont les caractéristiques suivantes :

- Elles disposent d'un port isolé
- Elles prennent en charge les protocoles point à point
- Elles sont configurées et programmées au moyen des instructions de communication point à point
- Affichent l'activité de transmission et réception au moyen de DEL
- Affiche une DEL de diagnostic (CM uniquement)
- Sont alimentés par la CPU : connexion à une alimentation externe inutile

Reportez-vous aux caractéristiques techniques pour les interfaces de communication (Page 1622).

DEL de signalisation

Les modules de communication comportent trois DEL de signalisation :

- DEL de diagnostic (DIAG) : Cette DEL clignote en rouge jusqu'à ce que la CPU y accède. Une fois la CPU sous tension, elle recherche les CM et y accède. La DEL de diagnostic commence à clignoter en vert. Cela signifie que la CPU a accédé au CM, mais ne lui a pas encore fourni la configuration. La CPU charge la configuration dans les CM configurés lorsque le programme est chargé dans la CPU. Après un chargement dans la CPU, la DEL de diagnostic du module de communication devrait émettre une lumière verte continue.
- DEL de transmission (Tx) : La DEL de transmission s'allume lorsque des données sont émises depuis le port de communication.
- DEL de réception (Rx) : Cette DEL s'allume lorsque le port de communication reçoit des données.

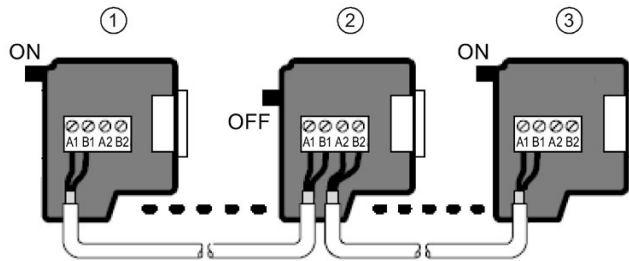
Le Communication Board présente des DEL de transmission (TxD) et de réception (RxD). Il n'y a pas de DEL de diagnostic.

13.2 Polarisation et terminaison d'un connecteur de réseau RS485

Siemens propose un connecteur de réseau RS485 (Page 1653) vous permettant de connecter aisément plusieurs appareils à un réseau RS485. Ce connecteur comporte deux jeux de bornes vous permettant de connecter les câbles de réseau entrants et sortants. Ce connecteur présente également des commutateurs pour la polarisation et la terminaison sélectives du réseau.

Remarque

La terminaison et la polarisation ne concernent que les deux extrémités du réseau RS485. Il n'y a pas de terminaison ni de polarisation des appareils situés entre les deux appareils d'extrémité. Blindage de câble nu : 12 mm environ doivent être en contact avec les guides métalliques de tous les emplacements.



- ① Position du commutateur = ON : terminaison et polarisation
- ② Position du commutateur = OFF : ni terminaison ni polarisation
- ③ Position du commutateur = ON : terminaison et polarisation

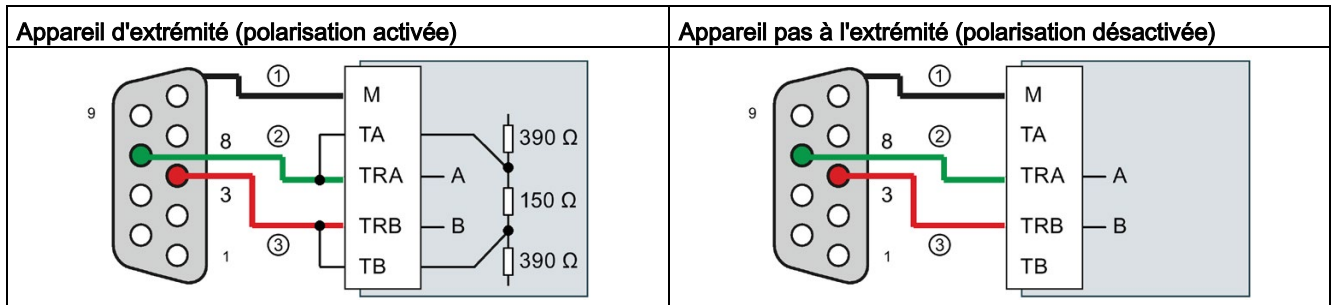
Tableau 13- 1 Terminaison et polarisation pour le connecteur RS485

Appareil d'extrémité (polarisation activée)	Appareil pas à l'extrémité (polarisation désactivée)

- ① Numéro de broche
- ② Connecteur de réseau
- ③ Blindage du câble

Le CB 1241 comporte des résistances internes pour la terminaison et la polarisation du réseau. Pour activer la terminaison et la polarisation de la liaison, raccordez TRA à TA et TRB à TB pour inclure les résistances internes dans le circuit. Le CB 1241 n'a pas de connecteur à 9 broches. Le tableau suivant montre les connexions à un connecteur à 9 broches sur le partenaire de communication.

Tableau 13- 2 Terminaison et polarisation pour le CB 1241



- ① Connectez M au blindage du câble
- ② A = TxD/RxD - (fil vert / broche 8)
- ③ B = TxD/RxD + (fil rouge / broche 3)

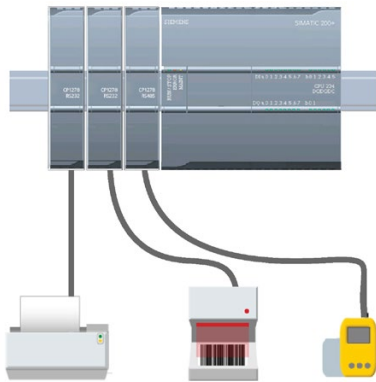
13.3 Communication point à point (PtP)

La CPU prend en charge la communication point à point (PtP) suivante pour les protocoles série en mode caractères :

- PtP Freeport (Page 1132)
- PtP 3964(R) (Page 1135)
- USS (Page 1199)
- Modbus (Page 1221)

13.3.1 PtP, communication Freeport

Le protocole PtP avec Freeport (communication librement programmable) offre une liberté et une souplesse maximales, mais requiert une implémentation importante dans le programme utilisateur.



La communication point à point offre de nombreuses possibilités :

- Possibilité d'envoyer des informations directement à un périphérique externe tel qu'une imprimante
- Possibilité de recevoir des informations d'autres appareils, tels que lecteurs de codes à barres, lecteurs RFID, systèmes de caméra ou de vision tiers et nombreux autres types d'appareils
- Possibilité d'échanger des informations, avec émission et réception de données, avec d'autres appareils tels que des appareils GPS, des systèmes de caméra ou de vision tiers, des modems radio et bien d'autres

Ce type de communication point à point est une communication série qui utilise des UART standard pour prendre en charge des débits et parités variés. Les modules de communication RS232 et RS422/485 (CM 1241) et le Communication Board RS485 (CB 1241) fournissent les interfaces électriques pour l'exécution de la communication point à point.

PtP Freeport sur PROFIBUS ou PROFINET

PtP vous permet d'utiliser un châssis de périphérie décentralisée PROFINET ou PROFIBUS pour communiquer avec divers appareils (lecteurs RFID, dispositifs GPS et autres) :

- PROFINET (Page 830) : Vous connectez l'interface Ethernet de la CPU S7-1200 à un coupleur PROFINET. Les modules de communication PtP dans le châssis avec le coupleur peuvent ensuite fournir des communications série avec les appareils PtP.
- PROFIBUS (Page 1002) : Vous insérez un module de communication PROFIBUS dans le côté gauche du châssis avec la CPU S7-1200. Vous connectez le module de communication PROFIBUS à un châssis contenant un coupleur PROFIBUS. Les modules de communication PtP dans le châssis avec le coupleur peuvent ensuite fournir des communications série avec les appareils PtP.

Pour cette raison, le S7-1200 accepte deux jeux d'instructions PtP :

- Instructions d'héritage point à point (Page 1281) : Ces instructions existaient avec la version V4.0 du S7-1200 et fonctionnent uniquement avec des communications série à l'aide d'un module de communication CM 1241 ou d'un communication board CB 1241.
- Instructions point à point (Page 1155) : Ces instructions offrent toutes les fonctionnalités des anciennes instructions et prennent également en charge les modules de communication PtP via la périphérie décentralisée PROFINET et PROFIBUS. Les instructions point à point vous permettent d'accéder aux modules de communication par le biais du châssis de périphérie décentralisée.

Les modules CM 1241 S7-1200 doivent avoir au moins la version V2.1 du firmware pour utiliser les instructions point à point. Ces modules sont restreints au châssis local sur le côté gauche de la CPU S7-1200. Vous pouvez également utiliser les instructions point à point avec un CB 1241.

13.3 Communication point à point (PtP)

La communication via la périphérie décentralisée utilise les modules suivants :

Station	Module	Numéro d'article	Interface
ET 200MP	CM PtP RS232 BA	6ES7540-1AD00-0AA0	RS232
	CM PtP RS232 HF	6ES7541-1AD00-0AB0	RS232
	CM PtP RS422/485 BA	6ES7540-1AB00-0AA0	RS422/RS485
	CM PtP RS422/485 HF	6ES7541-1AB00-0AB0	RS422/RS485
ET 200SP	CM PtP	6ES7137-6AA00-0BA0	RS232 et RS422/RS485

Remarque

Vous pouvez utiliser les instructions point à point pour accéder à un Communication Board, à des modules série locaux (ou sur le côté gauche), à des modules série via PROFINET et à des modules série via PROFIBUS. STEP 7 fournit les instructions d'héritage point à point uniquement pour prendre en charge les programmes existants. Toutefois ces instructions fonctionnent toujours avec les CPU S7-1200 actuelles. Vous n'avez pas besoin de convertir les programmes antérieurs d'un jeu d'instructions à l'autre.

Remarque

Conditions requises en matière de version de firmware de module CM pour la synchronisation de l'heure et la communication PtP

Si vous avez validé "La CPU synchronise les modules de l'appareil" dans les propriétés de Synchronisation de l'heure (Page 189) pour l'interface Profinet dans la configuration de l'appareil, mettez à jour les versions de firmware des modules de communication connectés avec les dernières versions disponibles. La validation de la synchronisation de l'heure par module pour les modules de communication avec des versions de firmware anciennes peut provoquer des problèmes ou des erreurs de communication.

13.3.2 Communication 3964(R)

La CPU S7-1200 prend en charge le protocole 3964(R) pour permettre la communication entre un module CM 1241 RS232 ou un module CM 1241 (RS422/485) et un partenaire de communication qui utilise le protocole 3964(R). Contrairement à la communication point à point décrite plus haut où vous définissez des caractéristiques d'émission et de réception spécifiques pour les messages, le protocole 3964(R) impose un protocole strict au moyen des caractères de commande suivants :

- STX Start of text
Début de la suite de caractères à émettre
- DLE Data Link Escape
Commutation de la transmission de données
- ETX End of Text
Fin de la suite de caractères à transmettre
- BCC Block check character
Caractère de contrôle de bloc
- NAK Negative Acknowledge
Acquittement négatif

Pour une description complète du protocole, voir le chapitre décrivant les principes de transmission de données série dans le manuel Manuel S7-300 Installation et paramétrage du couplage point-à-point CP 341

(<https://support.industry.siemens.com/cs/fr/fr/view/1117397>).

Configuration du module de communication

Pour communiquer avec un partenaire au moyen du protocole 3964(R), vous devez inclure l'un des modules de communication suivants dans votre configuration d'appareil dans STEP 7 :

- CM 1241 (RS232)
- CM 1241 (RS422/485)

Le firmware du module CM doit être de version V2.2.0 ou plus.

Dans le module de communication, vous devez configurer les ports de communication (Page 1137), la priorité et les paramètres du protocole (Page 1153).

Communication avec un partenaire avec le protocole 3964(R)

Lorsque vous configurez un module de communication pour le protocole 3964(R), vous utilisez les instructions d'émission et de réception point à point standard pour transférer des données entre la CPU et son partenaire de communication.

Le module de communication insère les données figurant dans le paramètre BUFFER de l'instruction d'émission dans le protocole 3964(R) et envoie les données au partenaire de communication.

Il reçoit les données du partenaire de communication au moyen du protocole 3964(R), supprime les informations de protocole et renvoie les données dans le paramètre BUFFER de l'instruction de réception.

Reportez-vous aux instructions point à point suivantes :

- Send_P2P (Envoyer les données du tampon d'émission) (Page 1173)
- Receive_P2P (Activer la réception des messages) (Page 1177)

Vous pouvez également utiliser les anciennes instructions d'émission et de réception point à point :

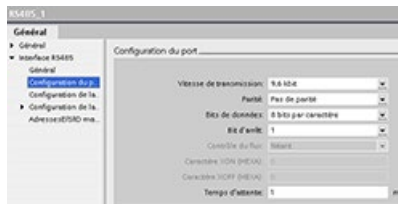
- SEND_PTP (Envoyer les données du tampon d'émission) (Page 1291)
- RCV_PTP (Valider la réception des messages) (Page 1293)

13.3.3 Configuration de la communication PtP Freeport

Vous pouvez utiliser l'une des méthodes suivantes pour configurer les interfaces de communication pour la communication PtP Freeport :

- Utilisez la configuration des appareils dans STEP 7 pour configurer les paramètres des ports (débit et parité), les paramètres d'émission et les paramètres de réception. La CPU mémorise les paramètres de configuration d'appareil. Elle les applique après une mise hors tension puis sous tension, ainsi qu'après un passage de l'état MARCHE à l'état ARRET.
- Utilisez les instructions Port_Config (Page 1158), Send_Config (Page 1162) et Receive_Config (Page 1164) pour définir les paramètres. Les paramétrages des ports définis par les instructions sont valables tant que la CPU reste à l'état MARCHE. Ils reprennent les valeurs de la configuration d'appareil après un passage à l'état ARRET ou une mise hors tension et sous tension.

Une fois les unités matérielles configurées (Page 159), vous paramétrez les interfaces de communication en sélectionnant l'un des CM dans le châssis ou le CB le cas échéant.



L'onglet "Propriétés" de la fenêtre d'inspection affiche les paramètres du CM ou CB choisi. Sélectionnez "Configuration du port" pour éditer les paramètres suivants :

- Vitesse de transmission
- Parité
- Bits de données par caractère
- Nombre de bits d'arrêt
- Contrôle de flux (RS232 seulement)
- Temps d'attente

Pour le CM 1241 RS232 et le CB RS485 (à l'exception du contrôle de flux (Page 1139), pris en charge uniquement par le CM 1241 RS232), les paramètres de configuration de port sont identiques que vous configuriez un module de communication RS232 ou RS485 ou un Communication Board RS485. Les valeurs des paramètres peuvent différer.

13.3 Communication point à point (PtP)

Pour le CM 1241 RS422/485, vous avez des options supplémentaires pour la configuration de port comme illustré ci-dessous. Le mode 422 du module CM 1241 RS422/485 prend également en charge le contrôle de flux logiciel.



Sélectionnez "Configuration du port" pour éditer les paramètres RS422/485 suivants :

- "Mode de fonctionnement" :
 - Duplex intégral (RS 422) mode sur 4 fils (couplage point à point)
 - Duplex intégral (RS422) mode sur 4 fils (maître multipoint)
 - Duplex intégral (RS422) mode sur 4 fils (esclave multipoint)
 - Semi-duplex (RS485) mode sur 2 fils
- "Préconfiguration de la ligne de réception" :
 - Aucune
 - Polarisation directe (signal R(A) 0V, signal R(B) 5V)

Le programme utilisateur STEP 7 peut également configurer le port ou en modifier la configuration existante à l'aide de l'instruction Port_Config (Page 1158). La rubrique des instructions contient des informations détaillées sur le mode de fonctionnement et la préconfiguration de la ligne ainsi que d'autres paramètres.

Paramètre	Définition
Vitesse de transmission	La valeur par défaut pour la vitesse de transmission est de 9,6 Kbits par seconde. Les choix autorisés sont : 300 bauds, 600 bauds, 1,2 Kbits, 2,4 Kbits, 4,8 Kbits, 9,6 Kbits, 19,2 Kbits, 38,4 Kbits, 57,6 Kbits, 76,8 Kbits et 115,2 Kbits
Parité	La valeur par défaut pour la parité est "Pas de parité". Les choix autorisés sont : Sans parité, parité paire, parité impaire, parité marque (bit de parité toujours à 1) et parité espace (bit de parité toujours à 0)
Bits de données par caractère	Nombre de bits de données dans un caractère. Les choix autorisés sont 7 ou 8.
Nombre de bits d'arrêt	Il peut y avoir un ou deux bits d'arrêt. Le nombre par défaut de bits d'arrêt est 1.
Contrôle du flux	Vous pouvez choisir soit le contrôle de flux matériel soit le contrôle de flux logiciel (Page 1139) pour le module de communication RS232. Si vous sélectionnez le contrôle de flux matériel, vous pouvez choisir si le signal RTS est toujours activé ou si RTS est commuté. Si vous sélectionnez le contrôle de flux logiciel, vous pouvez définir les caractères XON et XOFF. Les interfaces de communication RS485 n'acceptent pas le contrôle du flux. Le mode 422 du module CM 1241 RS422/485 prend en charge le contrôle de flux logiciel.
Temps d'attente	Le temps d'attente indique la durée pendant laquelle le CM ou le CB attend de recevoir CTS après avoir affirmé RTS ou de recevoir un caractère XON après avoir reçu un caractère XOFF, selon le type de contrôle de flux. Si le temps d'attente expire avant que l'interface de communication n'ait reçu le CTS ou XON attendu, le CM ou le CB interrompt la transmission et renvoie une erreur au programme utilisateur. Vous indiquez le temps d'attente en millisecondes. La plage va de 0 à 65535 millisecondes.

Paramètre	Définition
Mode de fonctionnement	Ce paramètre permet de sélectionner le mode de fonctionnement RS422 ou RS485 et les configurations réseau.
Préconfiguration de la ligne de réception	Ce paramètre permet de sélectionner les options de polarisation. Les valeurs valides sont Aucune, Polarisation directe et Polarisation inverse. La polarisation inverse permet la détection de rupture de fil.

13.3.3.1 Gestion du contrôle de flux

"Contrôle de flux" fait référence à un mécanisme pour équilibrer l'envoi et la réception de transmissions de données de manière à éviter la perte de données. Le contrôle de flux garantit qu'un émetteur n'envoie pas plus d'informations que le récepteur ne peut en gérer. Le contrôle de flux peut être matériel ou logiciel. Le CM RS232 prend en charge le contrôle de flux matériel et le contrôle de flux logiciel. Le CM et le CB RS485 n'acceptent pas le contrôle du flux. Le mode 422 du module CM 1241 RS422/485 prend en charge le contrôle de flux logiciel. Vous indiquez le type de contrôle de flux lorsque vous configurez le port (Page 1137) ou avec l'instruction PORT_CFG (Page 1282).

Le contrôle de flux matériel opère via les signaux de communication Demande pour émettre (RTS) et Prêt à émettre (CTS). Sur le CM RS232, le signal RTS est envoyé à partir de la broche 7 et le signal CTS est reçu par le biais de la broche 8. Le CM RS232 est un équipement terminal de données (DTE) qui affirme RTS en sortie et surveille CTS en entrée.

Contrôle de flux matériel : commutation RTS

Si vous activez le contrôle de flux matériel à commutation RTS pour un CM RS232, le module active le signal RTS pour envoyer des données. Il écoute le signal CTS pour déterminer si l'appareil récepteur peut accepter des données. Lorsque le signal CTS est actif, le module peut envoyer des données tant que le signal CTS reste actif. L'émission doit être interrompue si le signal CTS devient inactif.

L'émission reprend lorsque le signal CTS redevient actif. Si le signal CTS n'est pas activé pendant le temps d'attente configuré, le module interrompt la transmission et renvoie une erreur au programme utilisateur. Vous indiquez le temps d'attente dans la configuration du port (Page 1137).

Le contrôle de flux à commutation RTS est utile pour les appareils qui ont besoin d'un signal indiquant que l'émission est active, par exemple pour un modem radio qui utilise RTS en tant que signal "clé" pour exciter l'émetteur radio. Le contrôle de flux à commutation RTS ne fonctionne pas avec les modems téléphoniques standard. Utilisez l'option RTS toujours activé pour les modems téléphoniques.

Contrôle de flux matériel : RTS toujours activé

En mode RTS toujours activé, le CM 1241 active RTS par défaut. Un appareil tel qu'un modem téléphonique écoute le signal RTS provenant du CM et utilise ce signal en tant que signal "prêt à émettre". Le modem n'émet vers le CM que lorsque RTS est actif, c'est-à-dire lorsque le modem téléphonique voit un CTS actif. Si RTS est inactif, le modem téléphonique n'émet pas vers le CM.

Pour permettre au modem d'envoyer des données au CM à tout moment, configurez le contrôle de flux matériel "RTS toujours activé". Le CM active ainsi le signal RTS tout le temps. Le CM n'inactive pas RTS même si le module ne peut pas accepter de caractères. L'appareil émetteur doit s'assurer qu'il ne fait pas déborder le tampon de réception du CM.

Utilisation des signaux Terminal de données prêt (DTR) et Modem prêt (DSR)

Le CM active DTR pour les deux types de contrôle de flux matériel. Le module n'émet que lorsque le signal DSR devient actif. L'état de DSR n'est évalué qu'au début de l'opération d'émission. Si DSR devient inactif une fois l'émission commencée, cette dernière n'est pas suspendue.

Contrôle de flux logiciel

Le contrôle de flux logiciel utilise des caractères spéciaux dans les messages pour fournir le contrôle de flux. Vous configurez les caractères hexadécimaux qui représentent XON et XOFF.

XOFF indique qu'une émission doit s'arrêter. XON indique qu'une émission peut reprendre. XOFF et XON doivent être des caractères différents.

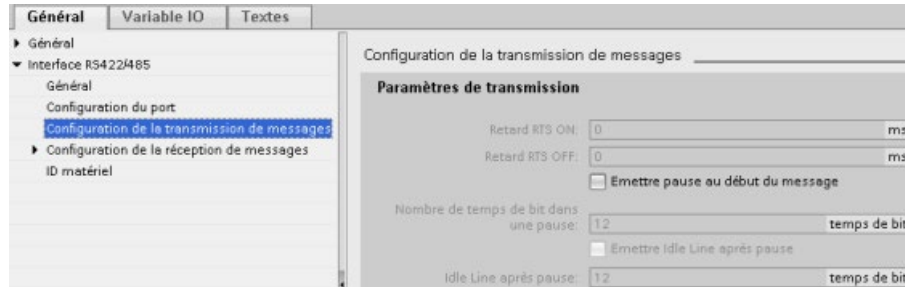
Lorsque l'émetteur reçoit un caractère XOFF du récepteur, il arrête d'émettre. L'émission reprend lorsque l'émetteur reçoit un caractère XON. S'il ne reçoit pas de caractère XON pendant le temps d'attente indiqué dans la configuration du port (Page 1137), le CM arrête l'émission et renvoie une erreur au programme utilisateur.

Le contrôle de flux logiciel requiert une communication duplex intégral puisque le récepteur doit être capable d'envoyer XOFF à l'émetteur alors qu'une transmission est en cours. Le contrôle de flux logiciel n'est possible qu'avec les messages qui contiennent uniquement des caractères ASCII. Les protocoles binaires ne peuvent pas utiliser le contrôle de flux logiciel.

Avant que la CPU ne puisse entamer la communication PtP Freeport, vous devez configurer les paramètres pour l'émission et la réception des messages. Ces paramètres conditionnent la façon dont la communication fonctionne lorsque des messages sont envoyés à un appareil cible ou reçu d'un appareil cible.

13.3.3.2 Configuration des paramètres d'émission

Dans la configuration d'appareil de la CPU, vous configurez la manière dont une interface de communication envoie des données en paramétrant la propriété "Configuration de la transmission de messages" pour l'interface sélectionnée.



Vous pouvez également configurer ou modifier dynamiquement les paramètres d'émission de messages à partir du programme utilisateur à l'aide de l'instruction Send_Config (Page 1162).

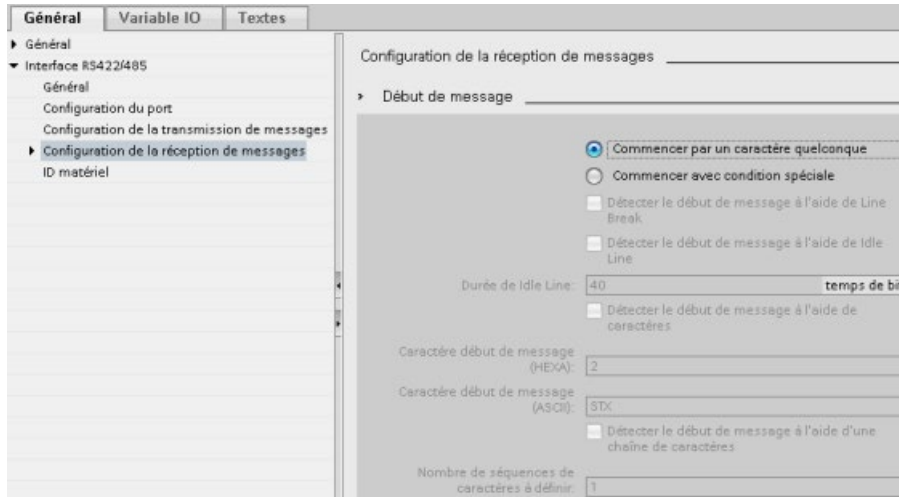
Remarque

Les valeurs de paramètres définies par l'instruction Send_Config dans le programme utilisateur ont priorité sur les propriétés de "Configuration de la transmission de messages". Notez que la CPU ne conserve pas les paramètres définis par l'instruction Send_Config en cas de mise hors tension.

Paramètre	Définition
Retard RTS ON	Indique le temps à attendre après l'activation de RTS avant de déclencher l'émission. La plage va de 0 à 65535 ms, 0 étant la valeur par défaut. Ce paramètre n'est valable que lorsque le contrôle de flux matériel est activé dans la configuration du port (Page 1137). CTS est évalué après expiration du temps de retard RTS activé. Ce paramètre ne s'applique qu'aux modules RS232.
Retard RTS OFF	Indique le temps à attendre avant de désactiver RTS après l'achèvement de l'émission. La plage va de 0 à 65535 ms, 0 étant la valeur par défaut. Ce paramètre n'est valable que lorsque le contrôle de flux matériel est activé dans la configuration du port (Page 1137). Ce paramètre ne s'applique qu'aux modules RS232.
Emettre Pause au début des messages	Indique qu'une pause sera envoyée au début de chaque message après expiration du retard RTS ON (si configuré) et activation de CTS.
Nombre de Bit Times dans une pause	Vous indiquez le nombre de temps bit qui constituent une pause où la ligne est maintenue en état de repos. La valeur par défaut est 12 et la valeur maximale est 65535, avec une limite de huit secondes.
Emettre Idle Line après Pause	Indique qu'une ligne inactive sera envoyée avant le début du message. Elle sera envoyée après la pause si une pause est configurée. Le paramètre "Idle Line après Pause" indique le nombre de temps bit qui constituent une ligne inactive où la ligne est maintenue en état de travail. La valeur par défaut est 12 et la valeur maximale est 65535, avec une limite de huit secondes.
Idle Line après Pause	

13.3.3.3 Configuration des paramètres de réception

Dans la configuration d'appareil de la CPU, vous définissez comment l'interface de communication reçoit des données et comment elle reconnaît le début et la fin d'un message. Vous indiquez ces paramètres dans les propriétés de "Configuration de la réception de messages" pour l'interface sélectionnée.



Vous pouvez également configurer ou modifier dynamiquement les paramètres de réception de messages à partir du programme utilisateur à l'aide de l'instruction `Receive_Config` instruction (Page 1164).

Remarque

Les valeurs de paramètres définies par l'instruction `Receive_Config` dans le programme utilisateur ont priorité sur les propriétés de "Configuration de la réception de messages". Notez que la CPU ne conserve pas les paramètres définis par l'instruction `RCV_CFG` en cas de mise hors tension ou de passage à l'état ARRET.

Conditions de début de message

Vous pouvez déterminer comment l'interface de communication reconnaît le début d'un message. Les caractères de début et les caractères contenant le message sont inscrits dans la mémoire tampon de réception jusqu'à apparition d'une condition de fin configurée.

Vous pouvez indiquer plusieurs conditions de début. Si vous définissez plusieurs conditions de début, toutes les conditions de début doivent être satisfaites pour que le message soit considéré comme commencé. Par exemple, si vous configurez un temps de ligne inactive et un caractère de début spécifique, le CM ou le CB vérifie d'abord que la condition concernant le temps de ligne inactive est satisfaite, puis il recherche le caractère de début indiqué. S'il reçoit un caractère autre que le caractère de début indiqué, le CM ou le CB relance la recherche de début de message en surveillant à nouveau le temps de ligne inactive.

Paramètre	Définition
Commencer par un caractère quelconque	La condition Caractère quelconque indique que tout caractère reçu avec succès signale le début d'un message. Ce caractère est le premier caractère dans le message.
Pause	La condition Pause indique qu'une opération de réception de message commence après réception d'un caractère de pause.
Ligne inactive	<p>La condition Ligne inactive indique qu'une réception de message commence lorsque la ligne de réception a été inactive ou au repos pendant le nombre de temps bit indiqué. Lorsque cette condition apparaît, le message commence.</p> <p>① Caractères ② Redémarre la temporisation de ligne inactive ③ Une ligne inactive est détectée et la réception de message commence</p>
Condition spéciale : Détecter le début de message à l'aide d'un caractère unique	Indique qu'un caractère particulier signale le début d'un message. Ce caractère est alors le premier caractère dans le message. Tout caractère reçu avant ce caractère spécifique est rejeté. Le caractère STX est pris par défaut.
Condition spéciale : Détecter le début de message à l'aide d'une chaîne de caractères	<p>Indique qu'une séquence de caractères particulière parmi quatre séquences configurées au plus signale le début d'un message. Vous pouvez indiquer jusqu'à cinq caractères pour chaque séquence. Pour chaque position de caractère, vous indiquez soit un caractère hexadécimal spécifique soit que le caractère n'est pas pris en compte dans la comparaison de séquence (caractère joker). Le dernier caractère spécifique d'une séquence met fin à cette séquence de début.</p> <p>Les séquences entrantes sont comparées aux conditions de début configurées jusqu'à ce qu'une condition de début soit satisfaite. Une fois la séquence de début satisfaite, l'évaluation des conditions de fin commence.</p> <p>Vous pouvez configurer jusqu'à quatre séquences de caractères spécifiques. Vous utilisez une condition de début multi-séquence quand différentes séquences de caractères peuvent indiquer le début d'un message. Si l'une quelconque des séquences de caractères est détectée, le message commence.</p>

13.3 Communication point à point (PtP)

L'ordre de vérification des conditions de début est défini comme suit :

- Ligne inactive
- Pause
- Caractères ou séquences de caractères

Si plusieurs conditions de début ont été définies et que l'une de ces conditions n'est pas satisfaite, le CM ou le CB relance la vérification en commençant par la première condition requise. Une fois que le CM ou le CB a établi que les conditions de début ont été satisfaites, il commence l'évaluation des conditions de fin.

Exemple de configuration : Début de message à l'apparition d'une séquence de caractères parmi deux

Considérons la configuration de condition de début de message suivante :

Détecter le début de message à l'aide d'une chaîne de caractères

Nombre de séquences de caractères à définir:

Séquence de 5 caractères pour début de message

Début de message séquence 1

Contrôler ce caractère : 1

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 2

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 3

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 4

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 5

Caractère (HEXA) :

Caractère (ASCII) :

Début de message séquence 2

Contrôler ce caractère : 1

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 2

Caractère (HEXA) : 6A

Caractère (ASCII) : j

Contrôler ce caractère : 3

Caractère (HEXA) : 6A

Caractère (ASCII) : j

Contrôler ce caractère : 4

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 5

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Dans cette configuration, la condition de début est satisfaite lorsque l'un ou l'autre des motifs suivants est détecté :

- Une séquence de cinq caractères est reçue : le premier caractère est 0x6A, le cinquième caractère est 0x1C et les positions 2, 3 et 4 peuvent contenir n'importe quel caractère. L'évaluation des conditions de fin commence une fois le cinquième caractère reçu.
- Deux caractères 0x6A consécutifs, précédés de n'importe quel caractère, sont reçus. Dans ce cas, l'évaluation des conditions de fin commence après la réception du deuxième 0x6A (3 caractères). Le caractère précédant le premier 0x6A est inclus dans la condition de début.

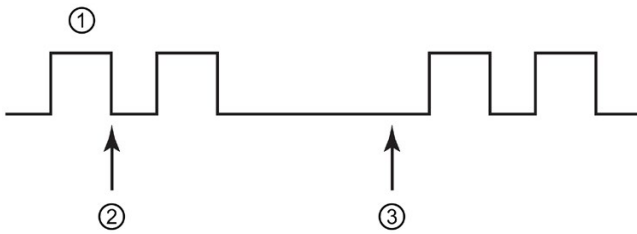
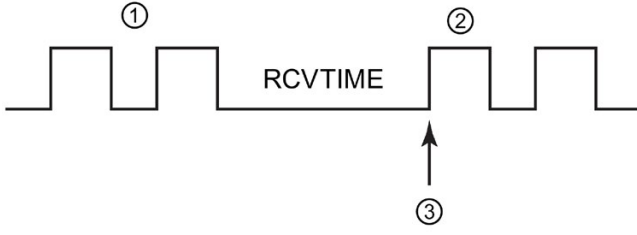
Exemples de séquences qui satisfont à cette condition de début :

- <tout caractère> 6A 6A
- 6A 12 14 18 1C
- 6A 44 A5 D2 1C

Conditions de fin de message

Vous déterminez également comment l'interface de communication reconnaît la fin d'un message. Vous pouvez configurer plusieurs conditions de fin de message. Si l'une quelconque des conditions configurées apparaît, le message s'arrête.

Par exemple, vous pouvez indiquer une condition de fin avec un dépassement du délai d'attente de message de 300 millisecondes, un dépassement du délai inter-caractères de 40 temps bit et une longueur maximale de 50 octets. La fin de message se produira si la réception du message prend plus de 300 millisecondes ou si le délai entre deux caractères quelconques dépasse 40 temps bit ou si 50 octets sont reçus.

Paramètre	Définition
<p>Détecter la fin de message à l'aide du dépassement de temps message</p>	<p>La fin de message se produit lorsque la durée configurée d'attente de la fin du message expire. Le délai d'attente de message commence à s'écouler lorsqu'une condition de début a été satisfaite. La valeur par défaut est de 200 ms, la plage étant comprise entre 0 et 65535 ms.</p>  <p>① Caractères reçus ② Condition de début de message satisfaite : la temporisation de message commence ③ La temporisation de message expire et termine le message</p>
<p>Détecter la fin de message à l'aide du dépassement de temps de réponse</p>	<p>La fin de message se produit lorsque la durée configurée d'attente d'une réponse expire avant qu'une séquence de début valide ne soit reçue. Le délai d'attente de réponse commence à s'écouler lorsqu'une émission s'achève et que le CM ou le CB commence l'opération de réception. Le délai d'attente de réponse par défaut est de 200 ms, la plage étant comprise entre 0 et 65535 ms. Si aucun caractère n'est reçu pendant le délai d'attente de réponse RCVTIME, une erreur est renvoyée à l'instruction RCV_PTP correspondante. Le dépassement du délai d'attente de la réponse ne définit pas une condition de fin spécifique. Il indique seulement qu'un caractère doit être reçu avec succès pendant le temps mentionné. Vous devez configurer une autre condition de fin pour signaler la fin réelle d'un message.</p>  <p>① Caractères émis ② Caractères reçus ③ Le premier caractère devrait être reçu maintenant.</p>

Paramètre	Définition
Détecter la fin de message à l'aide d'un dépassement du délai inter-caractères	<p>La fin de message se produit lorsque la durée maximale configurée entre deux caractères consécutifs quelconques d'un message expire. La valeur par défaut du délai inter-caractères est 12 temps bit et la valeur maximale est 65535 temps bit, avec un maximum de huit secondes.</p> <p>① Caractères reçus ② Redémarre la temporisation inter-caractères ③ La temporisation intercaractère expire et met fin au message.</p>
Détecter la fin de message à l'aide d'un nombre défini de caractères reçus	<p>La fin de message se produit lorsque le nombre défini de caractères a été reçu. La plage autorisée pour la longueur définie va de 1 à 4096.</p> <p>Notez que pour S7-1200, cette condition de fin de message ne s'applique qu'aux CPU V4.0 ou plus.</p>
Détecter la fin de message à l'aide de la longueur maximale	<p>La fin de message se produit lorsque le nombre configuré maximum de caractères a été reçu. La plage autorisée pour la longueur maximale va de 1 à 1024.</p> <p>Cette condition peut être utilisée pour empêcher le débordement de la mémoire tampon de messages. Lorsque cette condition de fin est combinée à des conditions de fin par dépassement de délai et qu'il y a dépassement de délai, tous les caractères reçus valides sont fournis même si la longueur maximale n'a pas été atteinte. Cela permet la prise en charge de protocoles de longueur variable lorsque seule la longueur maximale est connue.</p>
Lire la longueur dans le message	<p>Le message lui-même indique sa longueur. La fin de message se produit lorsqu'un message de la longueur indiquée a été reçu. La méthode pour indiquer et interpréter la longueur du message est décrite ci-après.</p>
Détecter la fin de message à l'aide d'un caractère	<p>La fin de message se produit lorsque le caractère indiqué est reçu.</p>
Détecter la fin de message à l'aide d'une chaîne de caractères	<p>La fin de message se produit lorsque la séquence de caractères indiquée est reçue. Vous pouvez indiquer une séquence de cinq caractères au maximum. Pour chaque position de caractère, vous indiquez soit un caractère hexadécimal spécifique soit que le caractère n'est pas pris en compte dans la comparaison de séquence.</p> <p>Les caractères en tête qui sont des caractères ignorés ne font pas partie de la condition de fin. Les caractères à droite qui sont des caractères ignorés font partie de la condition de fin.</p>

Exemple de configuration : Fin de message à l'apparition d'une séquence de caractères

Considérons la configuration de condition de fin de message suivante :

Détecter le fin de message à l'aide d'une chaîne de caractères

Séquence de 5 caractères pour fin de message

Contrôler ce caractère : 1

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 2

Caractère (HEXA) : 6A

Caractère (ASCII) : |

Contrôler ce caractère : 3

Caractère (HEXA) : 6A

Caractère (ASCII) : |

Contrôler ce caractère : 4

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 5

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Dans ce cas, la condition de fin est satisfaite lorsque deux caractères 0x6A consécutifs sont reçus et suivis de deux caractères quelconques. Le caractère précédant le motif 0x6A 0x6A ne fait pas partie de la séquence de caractères de fin. Les deux caractères suivant le motif 0x6A 0x6A sont nécessaires pour achever la séquence de caractères de fin. Les valeurs reçues aux positions 4 et 5 sont indifférentes mais doivent être reçues pour satisfaire la condition de fin.

Remarque

Si vous souhaitez que votre séquence de caractères indique la fin du message, placez la séquence dans les dernières positions de caractère. Dans l'exemple ci-dessus, si vous aviez voulu que 0x6A 0x6A termine le message sans caractère à droite, vous auriez configuré 0x6A dans les positions de caractère 4 et 5.

Indication de la longueur de message à l'intérieur du message

Lorsque vous sélectionnez la condition spéciale pour laquelle la longueur du message est incluse dans le message, vous devez fournir trois paramètres qui renseignent sur la longueur du message.

La structure réelle du message varie selon le protocole utilisé. Les trois paramètres sont les suivants :

- n : position de caractère (base 1) dans le message qui commence l'indication de longueur
- Taille du champ de longueur : nombre d'octets (un, deux ou quatre) de l'indication de longueur
- Longueur m : nombre de caractères suivant l'indication de longueur qui ne sont pas inclus dans le décompte de longueur

Les caractères de fin n'ont pas besoin d'être contigus. La valeur de "Longueur m" peut servir à indiquer la longueur d'un champ de total de contrôle dont la taille n'est pas incluse dans le champ de longueur.

Ces champs apparaissent dans la Configuration de la réception de messages des propriétés d'appareil :

Exemple 1 : Considérons un message structuré selon le protocole suivant :

STX	Lg (n)	Caractères 3 à 14 comptés par la longueur											
		ADR	PKE		INDEX		PWD		STW		HSW		BCC
1	2	3	4	5	6	7	8	9	10	11	12	13	14
STX	0x0C	xx	xxxx		xxxx		xxxx		xxxx		xxxx		xx

Configurez les paramètres de longueur pour la réception de ce message comme suit :

- n = 2 (l'indication de longueur du message commence au niveau de l'octet 2)
- Taille du champ de longueur = 1 (la longueur du message est définie dans un octet)
- Longueur m = 0 (il n'y a pas d'autres caractères après l'indication de longueur qui ne sont pas inclus dans le décompte de longueur ; douze caractères suivent l'indication de longueur)

13.3 Communication point à point (PtP)

Dans cet exemple, les caractères 3 à 14 inclus sont les caractères comptés par Lg (n).

Exemple 2 : Considérons un autre message structuré selon le protocole suivant :

SD1	Lg (n)	Lg (n)	SD2	Caractères 5 à 10 comptés par la longueur						FCS	ED
				DA	SA	FA	Unité de données=3 octets				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

Configurez les paramètres de longueur pour la réception de ce message comme suit :

- n = 3 (l'indication de longueur du message commence au niveau de l'octet 3)
- Taille du champ de longueur = 1 (la longueur du message est définie dans un octet)
- Longueur m = 3 (il y a trois caractères après l'indication de longueur qui ne sont pas comptés dans la longueur. Dans le protocole de cet exemple, les caractères SD2, FCS et ED ne sont pas comptés dans le décompte de longueur. Les six autres caractères sont comptés dans le décompte de longueur ; ainsi le nombre total de caractères suivant l'indication de longueur est neuf.)

Dans cet exemple, les caractères 5 à 10 inclus sont les caractères comptés par Lg (n).

13.3.4 Configuration de la communication 3964(R)

13.3.4.1 Configuration des ports de communication 3964(R)

Vous pouvez utiliser l'une des méthodes suivantes pour configurer les interfaces de communication pour la communication 3964(R) :

- Utilisez la configuration d'appareil dans STEP 7 pour configurer les paramètres des ports. La CPU enregistre les paramètres de configuration d'appareil et les applique après une mise hors tension puis sous tension.
- Utilisez l'instruction Port_Config (Page 1158) pour définir les paramètres des ports. Les paramètres de ports définis par ces instructions sont valables tant que la CPU reste à l'état MARCHE. Ils reprennent les valeurs de la configuration d'appareil après une mise hors tension et sous tension.

Une fois les interfaces de communication ajoutées à la configuration d'appareil (Page 164), vous les paramétrez en sélectionnant l'un des modules de communication dans le châssis.



L'onglet "Propriétés" de la fenêtre d'inspection affiche les paramètres du module de communication sélectionné. Sélectionnez "Configuration du port" pour éditer les paramètres suivants :

- Protocole : 3964(R)
- Mode de fonctionnement (module CM 1241 (RS422/485) uniquement)
- Préconfiguration de la ligne de réception (module CM 1241 (RS422/485) uniquement)
- Rupture de fil (module CM 1241 (RS422/485) uniquement)
- Vitesse de transmission
- Parité
- Bits de données
- Bits d'arrêt

13.3 Communication point à point (PtP)

Paramètre	Définition
Protocole	3964R ou Freeport. Sélectionnez 3964R pour configurer le port pour la communication 3964(R).
Mode de fonctionnement*	Duplex intégral (RS422) mode 4 fils liaison point à point (activé)
Préconfiguration de la ligne de réception*	Activez l'une des options suivantes : <ul style="list-style-type: none"> • Aucune • Polarisation avec $R(A) > R(B) \geq 0V$ • Polarisation avec $R(B) > R(A) \geq 0V$
Rupture de fil*	Activez l'une des options suivantes : <ul style="list-style-type: none"> • Pas de test de rupture de fil • Activer le test de rupture de fil
Vitesse de transmission	La valeur par défaut pour la vitesse de transmission est de 9,6 Kbits par seconde. Les choix autorisés sont : 300 bauds, 600 bauds, 1,2 Kbits, 2,4 Kbits, 4,8 Kbits, 9,6 Kbits, 19,2 Kbits, 38,4 Kbits, 57,6 Kbits, 76,8 Kbits et 115,2 Kbits
Parité	La valeur par défaut pour la parité est "Pas de parité". Les choix autorisés sont : Sans parité, parité paire, parité impaire, parité marque (bit de parité toujours à 1), parité espace (bit de parité toujours à 0) et parité quelconque (bit de parité à 0 pour l'émission, erreurs de parité non prises en compte à la réception).
Bits de données par caractère	Nombre de bits de données dans un caractère. Les choix autorisés sont 7 ou 8.
Nombre de bits d'arrêt	Il peut y avoir un ou deux bits d'arrêt. Par défaut, il y a un bit d'arrêt

* Module CM 1241 (RS422/485) uniquement

13.3.4.2 Configuration de la priorité et des paramètres du protocole 3964(R)

Vous pouvez utiliser l'une des méthodes suivantes pour configurer les interfaces de communication pour la communication 3964(R) :

- Dans la configuration d'appareil de l'interface de communication, cliquez sur "Configuration 3964(R)" pour définir la priorité et configurer les paramètres du protocole. La CPU enregistre les paramètres de configuration d'appareil et les applique après une mise hors tension puis sous tension.
- Utilisez l'instruction P3964_Config (Page 1170) pour définir la priorité et configurer les paramètres du protocole. Les valeurs définies par ces instructions sont valables tant que la CPU reste à l'état MARCHE. Elles reviennent aux valeurs de la configuration d'appareil après une mise hors tension et sous tension.



L'onglet "Propriétés" de la fenêtre d'inspection affiche les paramètres du module de communication sélectionné. Sélectionnez "Configuration 3964(R)" pour éditer les paramètres suivants :

- Priorité (haute ou basse)
- Paramètres du protocole
 - Avec contrôle de bloc 3964(R)
 - Utiliser les valeurs par défaut

Tentatives d'établissement de liaison

Tentatives de transmission

Délai inter-caractère

Délai d'acquiescement

13.3 Communication point à point (PtP)

Paramètre	Définition
Priorité	Haute ou basse : la priorité du CM sera définie à haute ou basse, celle du partenaire de communication devant être définie à l'inverse.
Avec contrôle de bloc 3964(R)	Si cette option est sélectionnée, la communication 3964(R) sécurise la transmission en incluant un caractère de contrôle de bloc (BCC). Dans le cas contraire, la sécurité de la transmission n'inclut pas de caractère de contrôle de bloc.
Utiliser les valeurs par défaut	Si cette option est sélectionnée, la communication 3964(R) utilise les valeurs par défaut pour les paramètres suivants du protocole : <ul style="list-style-type: none"> • Tentatives d'établissement de liaison • Tentatives de transmission • Délai inter-caractère • Délai d'acquittement Si cette option n'est pas sélectionnée, vous pouvez configurer des valeurs pour chacun de ces paramètres.
Tentatives d'établissement de liaison	Nombre de tentatives de connexion (valeur par défaut : 6 tentatives de connexion) 1 à 255
Tentatives de transmission	Nombre de tentatives de transmission (valeur par défaut : 6 tentatives de transmission) 1 à 255
Délai inter-caractère	Paramétrage du délai inter-caractère (dépend de la vitesse de transmission définie) (valeur par défaut : 220 ms) 1 ms à 65 535 ms
Délai d'acquittement	Paramétrage du temps de retard à l'acquittement (dépend de la vitesse de transmission définie) (valeur par défaut : 2 000 ms lorsque le contrôle de bloc est activé ; 550 ms lorsque le contrôle de bloc n'est pas activé) 1 ms à 65 535 ms

Remarque

A l'exception de la priorité, les paramètres du protocole doivent avoir les mêmes valeurs dans le module de communication et dans le partenaire de communication.

13.3.5 Instructions point à point

13.3.5.1 Paramètres communs pour les instructions point à point

Tableau 13- 3 Paramètres d'entrée communs pour les instructions point à point

Paramètre	Description
REQ	<p>De nombreuses instructions point à point utilisent l'entrée REQ pour déclencher l'opération en présence d'une transition du niveau bas au niveau haut. L'entrée REQ doit être au niveau haut (VRAI) pour une exécution de l'instruction, mais REQ peut rester à VRAI aussi longtemps que désiré. L'instruction ne déclenche pas de nouvelle opération tant qu'elle n'a pas été exécutée avec l'entrée REQ à FAUX de manière à ce qu'elle puisse réinitialiser l'historique de l'entrée REQ. Cela est nécessaire pour que l'instruction puisse à nouveau détecter une transition du niveau bas au niveau haut afin de déclencher l'opération suivante.</p> <p>Lorsque vous insérez une instruction point à point dans votre programme, STEP 7 vous demande d'identifier le DB d'instance. Utilisez un DB unique pour chaque appel de l'instruction point à point. Cela garantit que chaque instruction gère correctement les entrées telles que REQ.</p>
PORT	<p>Une adresse de port est affectée pendant la configuration de l'appareil de communication. Après la configuration, un nom symbolique de port par défaut peut être sélectionné dans la liste déroulante d'aide pour les paramètres. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes" de la table de variables API.</p>
Résolution en temps bit	<p>Plusieurs paramètres sont indiqués en nombre de temps bit à la vitesse de transmission configurée. Indiquer le paramètre en temps bit permet d'avoir un paramètre indépendant de la vitesse de transmission. Tous les paramètres en unités temps bit peuvent avoir une valeur maximale de 65535. Toutefois, la quantité de temps maximale que peut mesurer un CM ou un CB est de huit secondes.</p>

13.3 Communication point à point (PtP)

Les paramètres de sortie DONE, NDR, ERROR et STATUS des instructions point à point indiquent l'état d'achèvement de l'exécution des opérations point à point.

Tableau 13- 4 Paramètres de sortie DONE, NDR, ERROR et STATUS

Paramètre	Type de données	Valeur par défaut	Description
DONE	Bool	FAUX	A VRAI pour un cycle pour indiquer que la dernière demande s'est achevée sans erreur. FAUX sinon.
NDR	Bool	FAUX	A VRAI pour un cycle pour indiquer que la dernière action demandée s'est achevée sans erreur et que de nouvelles données ont été reçues. FAUX sinon.
ERROR	Bool	FAUX	A VRAI pour un cycle pour indiquer que la dernière demande s'est achevée avec des erreurs, le code d'erreur correspondant étant contenu dans STATUS. FAUX sinon.
STATUS	Word	0	Etat du résultat : <ul style="list-style-type: none"> • Si le bit DONE ou NDR est à 1, STATUS est mis à 0 ou reçoit un code informatif. • Si le bit ERROR est à 1, STATUS aura la valeur du code d'erreur. • Si aucun des bits ci-dessus n'est à 1, l'instruction renvoie des résultats d'état qui décrivent l'état en cours de la fonction. STATUS conserve sa valeur pendant la durée de l'exécution de la fonction.

Remarque

Notez que DONE, NDR et ERROR sont à 1 pour un cycle uniquement. La logique de votre programme doit sauvegarder les valeurs temporaires de l'état de sortie dans des verrous de données de manière à ce que vous puissiez détecter des changements d'état lors des cycles suivants du programme.

Tableau 13- 5 Codes d'erreur communs

STATUS (W#16#....)	Description
0000	Pas d'erreur
7000	La fonction n'est pas occupée.
7001	La fonction est occupée avec le premier appel.
7002	La fonction est occupée avec les appels suivants (interrogations après le premier appel).
8x3A	Pointeur illicite dans le paramètre x
8070	Toute la mémoire d'instance interne est occupée, trop d'instructions concurrentes en cours.
8080	Numéro de port illicite
8081	Dépassement du délai d'attente, erreur de module ou autre erreur interne
8082	Echec du paramétrage car un paramétrage est en cours en arrière-plan.
8083	Débordement de la mémoire tampon : Le CM ou le CB a renvoyé un message reçu dont la longueur est supérieure à la longueur autorisée.
8090	Erreur interne : Longueur de message erronée, sous-module erroné ou message illicite Contactez l'assistance client.
8091	Erreur interne : Version erronée dans le message de paramétrage Contactez l'assistance client.
8092	Erreur interne : Longueur d'enregistrement erronée dans le message de paramétrage Contactez l'assistance client.

Tableau 13- 6 Classes d'erreurs communes

Description de la classe	Classes d'erreurs	Description
Configuration du port	16#81Ax	Utilisé pour définir les erreurs de configuration de port communes
Configuration d'émission	16#81Bx	Utilisé pour définir les erreurs de configuration d'émission communes
Configuration de réception	16#81Cx 16#82Cx	Utilisé pour définir les erreurs de configuration de réception communes
Exécution de l'émission	16#81Dx	Utilisé pour définir les erreurs communes apparaissant à l'exécution de l'émission
Exécution de la réception	16#81Ex	Utilisé pour définir les erreurs communes apparaissant à l'exécution de la réception
Gestion des signaux	16#81Fx	Utilisé pour définir les erreurs communes associées à toute la gestion des signaux
Erreurs de pointeur	16#8p01 à 16#8p51	Utilisé pour les erreurs de pointeur ANY lorsque "p" correspond au numéro du paramètre de l'instruction
Erreurs de protocole intégrées	16#848x 16#858x	Utilisé pour les erreurs de protocole intégrées

13.3.5.2 Port_Config (Configurer des paramètres de communication dynamiquement)

Tableau 13- 7 Instruction Port_Config (Configuration de port)

CONT/LOG	SCL	Description
<pre> *Port_Config_DB* Port_Config - EN ENO - - REQ DONE - - PORT ERROR - - PROTOCOL STATUS - - BAUD - PARITY - DATABITS - STOPBITS - FLOWCTRL - XONCHAR - XOFFCHAR - WAITTIME - MODE - LINE_PRE - BRK_DET </pre>	<pre> "Port_Config_DB" (REQ:=_bool_in_, PORT:=_word_in_, PROTOCOL:=_uint_in_, BAUD:=_uint_in_, PARITY:=_uint_in_, DATABITS:=_uint_in_, STOPBITS:=_uint_in_, FLOWCTRL:=_uint_in_, XONCHAR:=_char_in_, XOFFCHAR:=_char_in_, WAITTIME:=_uint_in_, MODE:=_uint_in_, LINE_PRE:=_uint_in_, BRK_DET:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>Port_Config vous permet de modifier les paramètres du port, tels que la vitesse de transmission, à partir de votre programme.</p> <p>Vous pouvez définir la configuration statique initiale du port dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction Port_Config dans votre programme pour modifier la configuration.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

La CPU ne sauvegarde pas de manière permanente les valeurs que vous paramétrez avec l'instruction Port_Config. La CPU restaure les paramètres configurés dans la configuration des appareils lorsque la CPU passe de l'état MARCHE à l'état ARRÊT et après une mise hors tension puis sous tension. Reportez-vous à Configuration des ports de communication (Page 1137) et à Gestion du contrôle de flux (Page 1139) pour plus d'informations.

Tableau 13- 8 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
PROTOCOL	IN	UInt	0 = Protocole Freeport (valeur par défaut) 1 = Protocole 3964(R)
BAUD	IN	UInt	Vitesse de transmission du port (valeur par défaut : 6) : 1 = 300 bauds, 2 = 600 bauds, 3 = 1 200 bauds, 4 = 2 400 bauds, 5 = 4 800 bauds, 6 = 9 600 bauds, 7 = 19 200 bauds, 8 = 38 400 bauds, 9 = 57 600 bauds, 10 = 76 800 bauds, 11 = 115 200 bauds
PARITY	IN	UInt	Parité du port (valeur par défaut : 1) : 1 = Sans parité, 2 = Parité paire, 3 = Parité impaire, 4 = Parité marque, 5 = Parité espace
DATABITS	IN	UInt	Bits par caractère (valeur par défaut) : 1) : 1 = 8 bits de données, 2 = 7 bits de données
STOPBITS	IN	UInt	Bits d'arrêt (valeur par défaut : 1) : 1 = 1 bit d'arrêt, 2 = 2 bits d'arrêt
FLOWCTRL*	IN	UInt	Contrôle du flux (valeur par défaut : 1) : 1 = Pas de contrôle de flux, 2 = XON/XOFF, 3 = RTS matériel toujours activé, 4 = RTS matériel commuté
XONCHAR ¹	IN	Char	Indique le caractère qui est utilisé comme caractère XON. Ce sera typiquement le caractère DC1 (16#11). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 16#11)
XOFFCHAR ¹	IN	Char	Indique le caractère qui est utilisé comme caractère XOFF. Ce sera typiquement le caractère DC3 (16#13). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 16#13)
WAITTIME ¹	IN	UInt	Indique combien de temps attendre un caractère XON après avoir reçu un caractère XOFF ou combien de temps attendre le signal CTS après avoir activé RTS (0 à 65535 ms). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 2 000)
MODE ²	IN	UInt	Spécifie la sélection du mode de fonctionnement du module. <ul style="list-style-type: none"> • 0 = Duplex intégral (RS232) • 1 = Duplex intégral (RS422) mode 4 fils (point à point), émetteur toujours activé • 2 = Duplex intégral (RS422) mode 4 fils (maître multipoint), émetteur toujours activé • 3 = Duplex intégral (RS422) mode 4 fils (esclave multipoint), émetteur activé pendant l'émission • 4 = Semi-duplex (RS485) mode deux fils

13.3 Communication point à point (PtP)

Paramètre et type		Type de données	Description
LINE_PRE	IN	UInt	Spécifie la condition de ligne inactive (idle). Pour les modules RS422 et RS485, la condition de ligne inactive est établie en appliquant une tension de polarisation aux signaux R(A) et R(B). Les sélections possibles sont les suivantes : <ul style="list-style-type: none"> • 0 = Sans polarisation (pas de préréglage) (valeur par défaut) • 1 = Polarisation avec $R(A) > R(B) \geq 0V$; RS422 uniquement • 2 = Polarisation avec $R(B) > R(A) \geq 0V$; RS422 et RS485
BRK_DET	IN	UInt	Active/désactive la détection de rupture des câbles de communication. Avec l'activation de la détection de rupture de câble, le module indique un défaut lorsque le câble de communication n'est pas relié au module. En mode point à point RS422, la détection de rupture de câble est possible uniquement lorsque le préréglage de la ligne de réception est utilisé avec une polarisation appliquée de sorte que $R(A) > R(B) \geq 0V$. <ul style="list-style-type: none"> • 0 = Pas de test de rupture de fil (valeur par défaut) • 1 = Test de rupture de fil activé
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

¹ Non applicable lorsque Protocole = 1 (protocole 3964(R))

² Seuls les modes 0 et 1 sont valides lorsque Protocole = 1 (protocole 3964(R)) selon que votre module de communication est un module RS232 ou un module RS422.

Tableau 13- 9 Codes d'erreur

STATUS (W#16#....)	Description
81A0	Le protocole spécifique n'existe pas.
81A1	La vitesse de transmission spécifique n'existe pas.
81A2	La parité spécifique n'existe pas.
81A3	Le nombre spécifique de bits de données n'existe pas.
81A4	Le nombre spécifique de bits d'arrêt n'existe pas.
80A5	Le type spécifique de contrôle de flux n'existe pas.
81A6	Le temps d'attente est 0 et le contrôle de flux est activé.
81A7	XON et XOFF sont des valeurs illicites (par exemple, la même valeur).
81A8	Erreur dans l'en-tête du bloc (par exemple, type de bloc erroné ou longueur de bloc erronée)
81A9	Reconfiguration refusée car une configuration est en cours
81AA	Mode de fonctionnement du RS422/RS485 invalide
81AB	Préréglage invalide de la ligne de réception pour la détection des ruptures
81AC	Gestion des ruptures RS232 invalide
8280	Acquittement négatif pendant la lecture du module
8281	Acquittement négatif pendant l'écriture du module
8282	Le module ou l'esclave DP n'est pas disponible.

13.3.5.3 Send_Config (Configurer dynamiquement les paramètres de transmission série)

Tableau 13- 10 Instruction Send_Config (Configuration d'émission)

CONT/LOG	SCL	Description
<pre> *Send_Config_DB* Send_Config - EN ENO - - REQ DONE - - PORT ERROR - - RTSONDLY STATUS - - RTSOFFDLY - BREAK - IDLELINE - USR_END - APP_END </pre>	<pre> "Send_Config_DB" (REQ:=_bool_in_, PORT:=_word_in_, RTSONDLY:=_uint_in_, RTSOFFDLY:=_uint_in_, BREAK:=_uint_in_, IDLELINE:=_uint_in_, USR_END:=_string_in_, APP_END:=_string_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>Send_Config permet la configuration dynamique de paramètres d'émission série pour un port de communication point à point. Tout message mis en file d'attente à l'intérieur d'un CM ou d'un CB est rejeté lorsque Send_Config est exécuté.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Vous pouvez définir la configuration statique initiale du port dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction Send_Config dans votre programme pour modifier la configuration.

La CPU ne sauvegarde pas de manière permanente les valeurs que vous paramétrez avec l'instruction Send_Config. La CPU restaure les paramètres configurés dans la configuration des appareils lorsque la CPU passe de l'état MARCHE à l'état ARRET et après une mise hors tension puis sous tension. Voir Configuration des paramètres d'émission (Page 1141).

Tableau 13- 11 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
RTSONDLY	IN	UInt	Nombre de millisecondes à attendre après activation de RTS avant toute émission de données d'émission. Ce paramètre n'est valable que si le contrôle de flux matériel est activé. La plage valide va de 0 à 65535 ms. La valeur 0 désactive la fonction. (valeur par défaut : 0)
RTSOFFDLY	IN	UInt	Nombre de millisecondes à attendre avant de désactiver RTS après l'émission des données d'émission. Ce paramètre n'est valable que si le contrôle de flux matériel est activé. La plage valide va de 0 à 65535 ms. La valeur 0 désactive la fonction. (valeur par défaut : 0)
BREAK	IN	UInt	Ce paramètre indique qu'une pause du nombre indiqué de temps bit sera envoyée au début de chaque message. La valeur maximale est 65535 temps bit, avec un maximum de huit secondes. La valeur 0 désactive la fonction. (valeur par défaut : 12)
IDLELINE	IN	UInt	Ce paramètre indique que la ligne restera inactive pendant le nombre indiqué de temps bit avant le début de chaque message. La valeur maximale est 65535 temps bit, avec un maximum de huit secondes. La valeur 0 désactive la fonction. (valeur par défaut : 0)
USR_END*	IN	STRING[2]	Spécifie le nombre et les caractères compris dans le délimiteur de fin. Le délimiteur de fin est intégré au tampon d'émission (caractères uniquement) et marque la fin du message émis (les caractères sont émis jusqu'à ce que le délimiteur de fin soit rencontré). Le délimiteur de fin est ajouté à la fin du message. <ul style="list-style-type: none"> • STRING[2,0,xx,yy] – Le délimiteur de fin n'est pas utilisé (par défaut) • STRING[2,1,xx,yy] – Le délimiteur de fin est un caractère unique • STRING[2,2,xx,yy] – Le délimiteur de fin se compose de deux caractères USR_END ou APP_END doit avoir une longueur égale à zéro.
APP_END*	IN	STRING[5]	Spécifie le nombre et les caractères à ajouter au message émis (seuls les caractères sont ajoutés). <p>STRING[5,0,aa,bb,cc,dd,ee] – Le caractère de fin n'est pas utilisé (par défaut)</p> <ul style="list-style-type: none"> • STRING[5,1,aa,bb,cc,dd,ee] – Emettre un caractère de fin • STRING[5,2,aa,bb,cc,dd,ee] – Emettre deux caractères de fin • STRING[5,3,aa,bb,cc,dd,ee] – Emettre trois caractères de fin • STRING[5,4,aa,bb,cc,dd,ee] – Emettre quatre caractères de fin • STRING[5,5,aa,bb,cc,dd,ee] – Emettre cinq caractères de fin
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.

13.3 Communication point à point (PtP)

Paramètre et type		Type de données	Description
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

* Pas accepté pour le CM et le CB 1241 ; vous devez utiliser une chaîne vide ("") pour le paramètre.

Tableau 13- 12 Codes d'erreur

STATUS (W#16#...)	Description
81B0	Configuration d'alarme d'émission non autorisée. Contactez l'assistance client.
81B1	Le temps de pause est supérieur à la valeur maximale autorisée.
81B2	Le temps d'inactivité est supérieur à la valeur maximale autorisée.
81B3	Erreur dans l'en-tête du bloc, par exemple, mauvais type de bloc ou mauvaise longueur de bloc
81B4	Reconfiguration refusée car une configuration est en cours
81B5	Le nombre de délimiteurs de fin spécifié est supérieur à deux ou le nombre de caractères de fin est supérieur à cinq
81B6	Configuration d'émission refusée lorsqu'elle est configurée pour des protocoles intégrés de firmware
8280	Acquittement négatif pendant la lecture du module
8281	Acquittement négatif pendant l'écriture du module
8282	Le module ou l'esclave DP n'est pas disponible

13.3.5.4 Receive_Config (Configurer dynamiquement les paramètres de réception série)

Tableau 13- 13 Instruction Receive_Config (Configuration de réception)

CONT / LOG	SCL	Description
<pre> *Receive_ Config_DB* Receive_Config - EN ENO - - REQ DONE - - PORT ERROR - Receive_ STATUS - - Conditions </pre>	<pre> "Receive_Config_DB" (REQ:=_bool_in_, PORT:=_uint_in_, Re- ceive_Conditions:=_struct_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>Receive_Config procède à la configuration dynamique de paramètres de réception série pour un port de communication point à point. Cette instruction configure les conditions qui signalent le début et la fin d'un message reçu. Tout message mis en file d'attente à l'intérieur d'un CM ou d'un CB est rejeté lorsque Receive_Config est exécuté.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Vous pouvez définir la configuration statique initiale du port de communication dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction Receive_Config dans votre programme pour modifier la configuration.

La CPU ne sauvegarde pas de manière permanente les valeurs que vous paramétrez avec l'instruction Receive_Config. La CPU restaure les paramètres configurés dans la configuration des appareils lorsque la CPU passe de l'état MARCHE à l'état ARRET et après une mise hors tension puis sous tension. Pour plus d'informations, se référer à la rubrique "Configuration des paramètres de réception (Page 1142)".

Tableau 13- 14 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
CONDITIONS	IN	CONDITIONS	La structure de données Conditions indique les conditions de début et de fin de message comme décrit ci-dessous.
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Conditions de début pour l'instruction Receive_P2P

L'instruction Receive_P2P utilise la configuration indiquée par l'instruction Receive_Config pour déterminer le début et la fin de messages de communication point à point. Le début d'un message est déterminé par les conditions de début. Le début d'un message peut être déterminé par une condition de début ou par une combinaison de conditions de début. Si plusieurs conditions de début sont spécifiées, toutes les conditions doivent être satisfaites avant que le message ne commence.

Reportez-vous à la rubrique "Configuration des paramètres de réception (Page 1142)" pour une description des conditions de début de message.

Structure pour le paramètre CONDITIONS, partie 1 (conditions de début)

Tableau 13- 15 Structure CONDITIONS pour les conditions de début

Paramètre et type		Type de données	Description
STARTCOND	IN	UInt	Définit la condition de début (valeur par défaut : 1) <ul style="list-style-type: none"> • 01H : Caractère de début • 02H : Caractère quelconque • 04H : Pause • 08H : Ligne inactive • 10H : Séquence 1 • 20H : Séquence 2 • 40H : Séquence 3 • 80H : Séquence 4
IDLETIME	IN	UInt	Nombre de temps bit requis pour un délai de ligne inactive. (valeur par défaut : 40). Utilisé uniquement avec une condition de ligne inactive. 0 à 65535
STARTCHAR	IN	Byte	Caractère de début utilisé avec la condition caractère de début (valeur par défaut : B#16#2)
STRSEQ1CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 1 : (valeur par défaut : B#16#0) Il s'agit des bits d'activation pour chaque caractère dans la séquence de début. <ul style="list-style-type: none"> • 01H : Caractère 1 • 02H : Caractère 2 • 04H : Caractère 3 • 08H : Caractère 4 • 10H : Caractère 5 Désactiver le bit associé à un caractère signifie que n'importe quel caractère conviendra dans cette position de la séquence.
STRSEQ1	IN	Char[5]	Caractères de début de la séquence 1 (5 caractères). Valeur par défaut : 0
STRSEQ2CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 2. Valeur par défaut : B#16#0
STRSEQ2	IN	Char[5]	Caractères de début de la séquence 2 (5 caractères). Valeur par défaut : 0
STRSEQ3CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 3. Valeur par défaut : B#16#0
STRSEQ3	IN	Char[5]	Caractères de début de la séquence 3 (5 caractères). Valeur par défaut : 0
STRSEQ4CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 4. Valeur par défaut : B#16#0
STRSEQ4	IN	Char[5]	Caractères de début de la séquence 4 (5 caractères). Valeur par défaut : 0

Exemple

Considérons le message reçu suivant codé en hexadécimal : "68 10 aa 68 bb 10 aa 16" et les séquences de début configurées présentées dans le tableau ci-après. Les séquences de début commencent à être évaluées lorsque la réception du premier caractère 68H a abouti. Après réception du quatrième caractère (le second 68H), la condition de début 1 est satisfaite. Une fois les conditions de début satisfaites, l'évaluation des conditions de fin commence.

Le traitement de la séquence de début peut être interrompu en raison de diverses erreurs de parité, de trame ou de dépassement de délai inter-caractère. Si ces erreurs se produisent, aucun message n'est reçu puisque la condition de début n'était pas satisfaite.

Tableau 13- 16 Conditions de début

Condition de début	Premier caractère	Premier caractère +1	Premier caractère +2	Premier caractère +3	Premier caractère +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Conditions de fin pour l'instruction Receive_P2P

La fin d'un message est déterminée par l'indication de conditions de fin. La fin d'un message est déterminée par la première occurrence d'une ou de plusieurs conditions de fin configurées. Le paragraphe "Conditions de fin de message" dans la rubrique "Configuration des paramètres de réception (Page 1142)" décrit les conditions de fin que vous pouvez configurer dans l'instruction Receive_Config.

Vous pouvez configurer les conditions de fin soit dans les propriétés de l'interface de communication dans la configuration des appareils, soit dans l'instruction Receive_Config. Les paramètres de réception (conditions de début et conditions de fin) reprennent leur valeur de la configuration des appareils à chaque fois que la CPU passe de l'état ARRET à l'état MARCHE. Si le programme utilisateur STEP 7 exécute Receive_Config, les paramétrages sont modifiés pour être conformes aux conditions de Receive_Config.

Structure pour le paramètre CONDITIONS, partie 2 (conditions de fin)

Tableau 13- 17 Structure CONDITIONS pour les conditions de fin

Paramètre	Type de paramètre	Type de données	Description
ENDCOND	IN	UInt 0	Ce paramètre indique la condition de fin de message : <ul style="list-style-type: none"> • 01H : Temps de réponse • 02H : Temps de message • 04H : Délai inter-caractères • 08H : Longueur maximale • 10H : N + Lg + M • 20H : Séquence
MAXLEN	IN	UInt 1	Longueur de message maximale. Utilisé uniquement lorsque la condition de fin longueur maximale est sélectionnée. 1 à 1024 octets
N	IN	UInt 0	Position d'octet du champ de longueur à l'intérieur du message. Utilisé uniquement avec la condition de fin N + Lg + M. 1 à 1022 octets
LENGTHSIZE	IN	UInt 0	Taille du champ d'octet (1, 2 ou 4 octets). Utilisé uniquement avec la condition de fin N + Lg + M.
LENGTHM	IN	UInt 0	Indiquez le nombre de caractères suivant le champ de longueur qui ne sont pas inclus dans la valeur du champ de longueur. Utilisé uniquement avec la condition de fin N + Lg + M. 0 à 255 octets
RCVTIME	IN	UInt 200	Indiquez combien de temps attendre le premier caractère à recevoir. Il sera mis fin à l'opération de réception avec une erreur si aucune réception de caractère n'aboutit pendant le temps indiqué. Utilisé uniquement avec la condition temps de réponse. (0 à 65535 temps bit, avec un maximum de 8 secondes) Ce paramètre n'est pas une condition de fin de message puisque l'évaluation s'achève lors de la réception du premier caractère d'une réponse. Ce n'est une condition de fin que dans la mesure où il met fin à une opération de réception car aucune réponse n'est reçue alors qu'une réponse est attendue. Vous devez sélectionner une condition de fin distincte.
MSGTIME	IN	UInt 200	Indiquez combien de temps attendre pour que le message entier soit complètement reçu une fois le premier caractère reçu. Ce paramètre est utilisé uniquement lorsque la condition dépassement du délai d'attente de message est sélectionnée. (0 à 65535 millisecondes)
CHARGAP	IN	UInt 12	Indiquez le nombre de temps bit entre caractères. Si le nombre de temps bit entre caractères dépasse la valeur indiquée, la condition de fin est satisfaite. Utilisé uniquement avec la condition délai inter-caractères. (0 à 65535 temps bit, avec un maximum de 8 secondes)

Paramètre	Type de paramètre	Type de données	Description
ENDSEQ1CTL	IN	Byte B#16#0	Contrôle ignorer/comparer pour chaque caractère de la séquence 1 : Il s'agit des bits d'activation pour chaque caractère dans la séquence de fin. Le caractère 1 est le bit 0, le caractère 2 le bit 1, ..., le caractère 5 le bit 4. Désactiver le bit associé à un caractère signifie que n'importe quel caractère conviendra dans cette position de la séquence.
ENDSEQ1	IN	Char[5] 0	Caractères de début de la séquence 1 (5 caractères)

Tableau 13- 18 Codes d'erreur

STATUS (W#16#....)	Description
81C0	Condition de début illicite sélectionnée
81C1	Condition de fin illicite sélectionnée, pas de condition de fin sélectionnée
81C2	Alarme de réception activée ce qui n'est pas possible
81C3	Condition de fin longueur maximale activée et longueur maximale est 0 ou > 1024.
81C4	Longueur calculée activée et N >= 1023
81C5	Longueur calculée activée et longueur différente de 1, 2 ou 4
81C6	Longueur calculée activée et M > 255
81C7	Longueur calculée activée et longueur calculée > 1024
81C8	Dépassement du délai d'attente de la réponse activé et délai d'attente de réponse est zéro.
81C9	Dépassement du délai inter-caractères activé et délai est 0.
81CA	Dépassement du délai de ligne inactive activé et délai est 0.
81CB	Séquence de fin activée mais tous les caractères sont indifférents.
81CC	Séquence de début (l'une quelconque parmi 4) activée mais tous les caractères sont indifférents.
81CD	Erreur de sélection de la protection contre l'écrasement pour un message de réception invalide
81CE	Gestion de la mémoire tampon des messages de réception invalides concernant l'erreur de sélection de la transition d'ARRET à MARCHE
81CF	Erreur dans l'en-tête du bloc, par exemple, type de bloc erroné ou longueur de bloc erronée
8281	Acquittement négatif pendant l'écriture du module
8282	Le module ou l'esclave DP n'est pas disponible
82C0	Reconfiguration refusée car une configuration est en cours
82C1	La valeur indiquée pour le nombre de messages que le module peut mettre en mémoire tampon est supérieure à la valeur maximale autorisée.
82C2	Configuration de réception refusée lorsqu'elle est configurée pour des protocoles intégrés de firmware
8351	Type de données non autorisé dans ce pointeur Variant

13.3.5.5 P3964_Config (Configurer le protocole 3964(R))

Tableau 13- 19 Instruction P3964_Config (Configurer le protocole 3964(R))

CONT/LOG	SCL	Description
<pre> *P3964_Config_ DB* P3964_Config - EN ENO - - REQ DONE - - PORT ERROR - - BCC STATUS - - Priority - CharacterDelay - Time - AcknDelayTime - BuildupAttempt - s - RepetitionAtte - mpts </pre>	<pre> "P3964_Config_DB" (REQ:=_bool_in_, PORT:=_uint_in_, BCC:=_usint_in_, Priority:=_usint_in_, CharacterDelayTime:=_uint_in_, AcknDelayTime:=_uint_in_, BuildupAttempts:=_usint_in_, RepetitionAttempts:=_usint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>L'instruction P3964_Config permet de modifier la priorité et les paramètres du protocole à l'exécution.</p> <p>Vous pouvez définir la configuration statique initiale du port dans les propriétés de la configuration d'appareil ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction P3964_Config dans votre programme pour modifier la configuration.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

La CPU ne sauvegarde pas de manière permanente les valeurs que vous paramétrez avec l'instruction P3964_Config. Elle restaure les paramètres configurés dans la configuration d'appareil après une mise hors tension puis sous tension de la CPU. Pour plus d'informations, voir Configuration de la priorité et des paramètres du protocole 3964(R) (Page 1153).

Tableau 13- 20 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Active la modification de configuration en présence d'un front montant à cette entrée (valeur par défaut : faux)
PORT	IN	UInt	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété "ID matériel" de la configuration d'appareil. Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
BCC	IN	USInt	Active/désactive l'utilisation du contrôle de bloc. <ul style="list-style-type: none"> • 0 = sans contrôle de bloc • 1 = avec contrôle de bloc
Priority	IN	UInt	Sélection de la priorité <ul style="list-style-type: none"> • 0 = priorité basse • 1 = priorité haute La priorité du CM doit être l'opposé de la priorité du partenaire de communication.
CharacterDelayTime	IN	UInt	Paramétrage du délai inter-caractère (dépend de la vitesse de transmission définie) (valeur par défaut : 220 ms) 1 ms à 65 535 ms
AcknDelayTime	IN	UInt	Paramétrage du temps de retard à l'acquittement (dépend de la vitesse de transmission définie) (valeur par défaut : 2 000 ms) 1 ms à 65 535 ms
BuildupAttempts	IN	UInt	Nombre de tentatives de connexion (valeur par défaut : 6 tentatives de connexion) 1 à 255
RepetitionAttempts	IN	UInt	Nombre de tentatives de transmission (valeur par défaut : 6 tentatives de transmission) 1 à 255
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

13.3 Communication point à point (PtP)

Tableau 13- 21 Codes d'erreur

STATUS (W#16#...)	Description
16#8380	Erreur de paramétrage : valeur invalide pour "délai inter-caractère"
16#8381	Erreur de paramétrage : valeur invalide pour "délai de réponse"
16#8382	Erreur de paramétrage : valeur invalide pour "priorité"
16#8383	Erreur de paramétrage : valeur invalide pour "contrôle de bloc"
16#8384	Erreur de paramétrage : valeur invalide pour "tentatives de connexion"
16#8385	Erreur de paramétrage : valeur invalide pour "tentatives de transmission"
16#8386	Erreur à l'exécution : nombre de tentatives de connexion dépassé
16#8387	Erreur à l'exécution : Nombre de tentatives de transmission dépassé
16#8388	Erreur à l'exécution : erreur sur le "caractère de contrôle de bloc" La valeur calculée en interne pour le caractère de contrôle de bloc ne correspond pas au caractère de contrôle de bloc reçu par le partenaire à l'extrémité de la liaison.
16#8389	Erreur à l'exécution : caractère invalide reçu alors qu'on attendait la libération du tampon de réception
16#838A	Erreur à l'exécution : erreur logique pendant la réception Après la réception de DLE, un caractère autre que DLE ou ETX a été reçu.
16#838B	Erreur à l'exécution : délai inter-caractère dépassé
16#838C	Erreur à l'exécution : le temps d'attente pour la libération du tampon de réception a démarré.
16#838D	Erreur à l'exécution : la répétition de télégramme n'a pas démarré dans les 4 s après NAK.
16#838E	Erreur à l'exécution : un ou plusieurs caractères (autres que NAK ou STX) ont été reçus au repos.
16#838F	Erreur à l'exécution : conflit d'initialisation ; les deux partenaires sont réglés à "priorité haute".
16#8391	Erreur de paramétrage : données de configuration 3964 rejetées parce que la communication programmable est activée.

13.3.5.6 Send_P2P (Envoyer les données du tampon d'émission)

Tableau 13- 22 Instruction Send_P2P (Emission de données point à point)

CONT / LOG	SCL	Description
<pre> *Send_P2P_DB* Send_P2P - EN ENO - - REQ DONE - - PORT ERROR - - BUFFER STATUS - - LENGTH </pre>	<pre> "Send_P2P_DB" (REQ:=_bool_in_, PORT:=_word_in_, BUFFER:=_variant_in_, LENGTH:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>Send_P2P déclenche l'émission des données et transfère la mémoire tampon affectée vers l'interface de communication. Le programme de la CPU se poursuit pendant que le CM ou le CB envoie les données à la vitesse de transmission définie. Une seule opération d'émission peut être en attente à un moment donné. Le CM ou le CB renvoie une erreur si une seconde instruction Send_P2P est exécutée alors qu'il est déjà en train d'envoyer un message.</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 23 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de l'émission demandée en cas de front montant de cette entrée de validation d'émission. Cela déclenche le transfert du contenu de la mémoire tampon vers l'interface de communication point à point. (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
BUFFER	IN	Variant	Ce paramètre pointe sur l'adresse de début de la mémoire tampon d'émission. (valeur par défaut : 0) Remarque : Les données booléennes et les tableaux booléens ne sont pas acceptés.
LENGTH	IN	UInt	Longueur de trame émise en octets (valeur par défaut : 0) Utilisez toujours une longueur de 0 lorsque vous envoyez une structure complexe. Avec une longueur de 0, l'instruction transmet la trame entière.
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Tant qu'une opération d'émission est en cours, les sorties DONE et ERROR sont à FALSE. Lorsqu'une opération d'émission est achevée, soit la sortie DONE soit la sortie ERROR est mise à TRUE pour montrer l'état de l'opération d'émission. Lorsque DONE ou ERROR est TRUE, la sortie STATUS est valide.

13.3 Communication point à point (PtP)

L'instruction renvoie un état de 16#7001 si l'interface de communication accepte les données d'émission. Les exécutions suivantes de Send_P2P renvoient 16#7002 si le CM ou le CB est toujours occupé à émettre. Une fois l'opération d'émission achevée, le CM ou le CB renvoie l'état 16#0000 pour l'opération d'émission (si aucune erreur ne s'est produite). Les exécutions suivantes de Send_P2P avec REQ au niveau bas renvoient l'état 16#7000 (pas occupé).

Les schémas suivants montrent la relation des valeurs de sortie à REQ. On part du principe que l'instruction est appelée périodiquement pour vérifier l'état du processus d'émission. Dans le schéma ci-après, on suppose que l'instruction est appelée à chaque cycle (représenté par les valeurs de STATUS).

REQ							
DONE							
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

Le schéma suivant montre que les paramètres DONE et STATUS ne sont valables que pour un cycle si la ligne REQ présente une impulsion (pendant un cycle) pour déclencher l'opération d'émission.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

Le schéma suivant montre la relation des paramètres DONE, ERROR et STATUS en présence d'une erreur.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

Les valeurs de DONE, ERROR et STATUS ne sont valables que jusqu'à l'exécution suivante de Send_P2P avec le même DB d'instance.

Tableau 13- 24 Codes d'erreur

STATUS (W#16#....)	Description
81D0	Nouvelle demande alors que l'émetteur est actif
81D1	Emission interrompue car pas de CTS pendant le temps d'attente
81D2	Emission interrompue car pas de DSR en provenance de l'équipement DCE
81D3	Emission interrompue en raison d'un débordement de la file d'attente (émission de plus de 1024 octets)
81D5	Signal de polarisation inverse (rupture de fil)
81D6	Demande d'émission refusée car le délimiteur de fin n'a pas été trouvé dans la mémoire tampon d'émission
81D7	Erreur interne / erreur dans la synchronisation entre FB et CM
81D8	Tentative d'émission refusée car le port n'a pas été configuré
81DF	<p>CM a réinitialisé l'interface sur le FB pour l'une des raisons suivantes</p> <ul style="list-style-type: none"> • Le module a redémarré (mise hors tension puis sous tension) • La CPU a atteint un point d'arrêt • Le module a été reparamétré <p>Dans chaque cas, le module indique ce code dans le paramètre Status. Le module remet Status et Error à zéro après le premier enregistrement reçu pour SEND_P2P.</p>
8281	Acquittement négatif pendant l'écriture du module
8282	Le module ou l'esclave DP n'est pas disponible
8301	ID de syntaxe illicite dans n'importe quel pointeur ANY
8322	Erreur de longueur de plage lors de la lecture d'un paramètre
8324	Erreur de plage lors de la lecture d'un paramètre
8328	Erreur d'alignement lors de la lecture d'un paramètre
8332	Le paramètre contient un numéro de DB qui est plus élevé que le numéro maximum autorisé (erreur de numéro de DB).
833A	Le DB pour le paramètre BUFFER n'existe pas.

Remarque

Paramétrage de la longueur d'enregistrement maximum pour la communication Profibus

Lorsque vous utilisez un module maître Profibus CM1243-5 pour piloter un appareil Profibus ET 200SP ou ET 200MP qui utilise un module point à point RS232, RS422 ou RS485, vous devez paramétrer explicitement la variable de bloc de données "max_record_len" sur 240 comme défini ci-dessous :

Paramétrer "max_record_len" dans le DB d'instance (par exemple "Send_P2P_DB".max_record_len) sur 240 après avoir exécuté une quelconque instruction de configuration telle que Port_Config, Send_Config ou Receive_Config.

Paramétrer explicitement max_record_len est seulement nécessaire avec la communication Profibus ; la communication Profinet utilise déjà une valeur max_record_len valide.

Interaction entre les paramètres LENGTH et BUFFER

La taille de données minimale pouvant être envoyée par l'instruction SEND_P2P est un octet. Le paramètre BUFFER détermine la taille des données à envoyer. Vous ne pouvez pas utiliser le type de données Bool ou des tableaux de Bool pour le paramètre BUFFER.

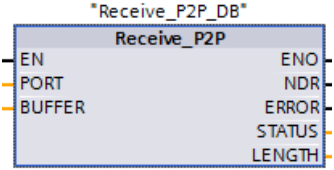
Vous pouvez toujours définir le paramètre LENGTH à 0 pour vous assurer que SEND_P2P envoie la structure de données entière représentée par le paramètre BUFFER. Si vous ne voulez envoyer qu'une partie des données dans le paramètre BUFFER, vous pouvez définir LENGTH comme suit :

Tableau 13- 25 Paramètres LENGTH et BUFFER

LENGTH	BUFFER	Description
= 0	Non utilisé	L'ensemble des données tel que défini dans le paramètre BUFFER est envoyé. Vous n'avez pas besoin d'indiquer le nombre d'octets émis lorsque LENGTH = 0.
> 0	Type de données simple	La valeur LENGTH doit contenir le nombre d'octets de ce type de données. Pour une valeur Word par exemple, LENGTH doit être égal à deux. Pour une valeur Dword ou Real, LENGTH doit être égal à quatre. Sinon, rien n'est transféré et l'erreur 8088H est renvoyée.
	Structure	La valeur LENGTH peut contenir un nombre d'octets inférieur à la longueur totale en octets de la structure, auquel cas l'instruction envoie seulement les n premiers octets de la structure à partir du BUFFER, avec n = LENGTH. Mais comme l'organisation interne des octets d'une structure ne peut pas toujours être déterminée, vous risquez d'obtenir des résultats inattendus. Dans ce cas, donnez la valeur 0 au paramètre LENGTH pour envoyer la structure entière.
	Tableau	La valeur LENGTH doit contenir un nombre d'octets qui est inférieur ou égal à la longueur totale en octets du tableau et qui doit être un multiple du nombre d'octets de l'élément de données. Par exemple, le paramètre LENGTH doit être un multiple de deux pour un tableau de mots (Word) et un multiple de quatre pour un tableau de réels (Real). Lorsque LENGTH est précisé, l'instruction transfère le nombre d'éléments de tableau qui correspondent à la valeur LENGTH en octets. Par exemple, si votre paramètre BUFFER contient un tableau de 15 doubles mots (Dword), soit 60 octets au total, et que vous indiquez une longueur LENGTH de 20, les cinq premiers doubles mots (Dword) du tableau seront transférés. La valeur de LENGTH doit être un multiple du nombre d'octets de l'élément de données. Sinon, STATUS prend la valeur 8088H, ERROR prend la valeur 1 et rien n'est envoyé.
	String	Le paramètre LENGTH contient le nombre de caractères à envoyer. Seuls les caractères de la chaîne (String) sont envoyés. Les octets de longueur maximale et réelle de la chaîne (String) ne sont pas émis.

13.3.5.7 Receive_P2P (Activer la réception des messages)

Tableau 13- 26 Instruction Receive_P2P (Réception point à point)

CONT / LOG	SCL	Description
	<pre>"Receive_P2P_DB" (PORT:=_word_in_, BUFFER:=_variant_in_, NDR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, LENGTH=>_uint_out_);</pre>	<p>Receive_P2P vérifie la présence de messages reçus dans le CM ou le CB. Si un message est disponible, il est transféré du CM ou du CB à la CPU. Une valeur STATUS appropriée est renvoyée en cas d'erreur.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 27 Types de données pour les paramètres

Paramètre et type		Type de données	Description
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
BUFFER	IN	Variant	Ce paramètre pointe sur l'adresse de début de la mémoire tampon de réception. Cette mémoire tampon doit être suffisamment grande pour recevoir la longueur de message maximale. Les données booléennes et les tableaux booléens ne sont pas acceptés. (valeur par défaut : 0)
NDR	OUT	Bool	TRUE pour un cycle lorsque de nouvelles données sont prêtes et que l'opération s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque l'opération s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)
LENGTH	OUT	UInt	Longueur du message renvoyé en octets (valeur par défaut : 0)

La valeur STATUS est valide si soit NDR soit ERROR est TRUE. La valeur STATUS fournit la raison de l'arrêt de l'opération de réception dans le CM ou le CB. Il s'agira typiquement d'une valeur positive indiquant que l'opération de réception a réussi et que le processus de réception s'est terminé normalement. Si la valeur STATUS est négative (le bit de poids fort de la valeur hexadécimale est à 1), cela signifie que l'opération de réception a été interrompue en raison d'une situation d'erreur, telles des erreurs de parité, de trame ou de débordement.

13.3 Communication point à point (PtP)

Chaque interface de communication point à point peut mettre jusqu'à 1024 octets en mémoire tampon. Il peut s'agir d'un seul grand message ou de plusieurs petits messages. S'il y a plus d'un message disponible dans le CM ou le CB, l'instruction Receive_P2P renvoie le message disponible le plus ancien. Une nouvelle exécution de l'instruction Receive_P2P renverra le message le plus ancien suivant.

Tableau 13- 28 Codes d'erreur

STATUS (W#16#...)	Description
0000	Pas de mémoire tampon
0094	Message interrompu car la longueur de caractères maximum a été reçue
0095	Message interrompu en raison de l'expiration du délai de message
0096	Message interrompu en raison de l'expiration du délai intercaractère
0097	Message interrompu en raison de l'expiration du délai de réponse
0098	Message interrompu car la condition de longueur "N+LEN+M" était satisfaite
0099	Message interrompu car la séquence de fin était satisfaite
8085	Le paramètre LENGTH a la valeur 0 ou est supérieur à 1 Ko.
8088	Le paramètre LENGTH ou la longueur reçue est plus long que la zone spécifiée dans BUFFER, ou la longueur reçue est plus longue que la zone spécifiée dans BUFFER.
8090	Message de configuration incorrect, longueur de message erronée, sous-module erroné, message illicite
81E0	Message interrompu car la mémoire tampon de réception est pleine
81E1	Message interrompu en raison d'une erreur de parité
81E2	Message interrompu en raison d'une erreur de trame
81E3	Message interrompu en raison d'une erreur de débordement
81E4	Message interrompu car la longueur calculée dépasse la taille de la mémoire tampon
81E5	Signal de polarisation inverse (rupture de fil)
81E6	La file d'attente des messages est pleine. Cette erreur est signalée sans données. Si cela se produit, le module bascule entre un transfert sans erreur et cette erreur.
81E7	Erreur interne, erreur dans la synchronisation entre l'instruction et CM : défini quand une erreur de séquence est détectée
81E8	Message interrompu, expiration du délai inter-caractère avant que la fin du critère de message ait été satisfaite
81E9	Erreur Modbus CRC détectée (Utilisée uniquement par des modules qui acceptent la génération/vérification CRC pour le protocole Modbus)
81EA	Le télégramme Modbus est trop court (Utilisé uniquement par des modules qui acceptent la génération/vérification CRC pour le protocole Modbus)
81EB	Message interrompu, car la taille maximum de message est dépassée
8201	ID de syntaxe illicite dans n'importe quel pointeur ANY
8223	Erreur de longueur de plage lors de l'écriture d'un paramètre. Le paramètre est situé soit entièrement soit en partie en dehors de la plage d'une adresse ou la longueur d'une plage binaire n'est pas un multiple de 8 avec un pointeur ANY.
8225	Erreur de plage lors de l'écriture d'un paramètre. Le paramètre se situe dans une plage qui est illicite pour la fonction système.
8229	Erreur d'alignement lors de l'écriture d'un paramètre. Le paramètre référencé se situe dans une adresse binaire qui n'est pas égale à 0.
8230	Le paramètre se situe dans un DB global en lecture seule

STATUS (W#16#...)	Description
8231	Le paramètre se situe dans un DB d'instance en lecture seule.
8232	Le paramètre contient un numéro de DB qui est plus élevé que le numéro de bloc maximum autorisé (erreur de numéro de DB).
823A	Le DB pour le paramètre BUFFER n'existe pas.
8280	Acquittement négatif pendant la lecture du module
8282	Le module ou l'esclave DP n'est pas disponible

13.3.5.8 Receive_Reset (Effacer le tampon de réception)

Tableau 13- 29 Instruction Receive_Reset (Réinitialiser récepteur)

CONT / LOG	SCL	Description
<pre> *Receive_Reset_ DB* Receive_Reset - EN ENO - - REQ DONE - - PORT ERROR - STATUS - </pre>	<pre> "Receive_Reset_DB" (REQ:=_bool_in_, PORT:=_word_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	Receive_Reset efface les mémoires tampons de réception dans le CM ou le CB.

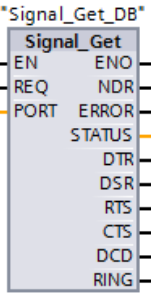
1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 30 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la réinitialisation du récepteur en cas de front montant de cette entrée de validation (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
DONE	OUT	Bool	Cette sortie est TRUE pour un cycle lorsque la dernière requête s'est achevée sans erreur.
ERROR	OUT	Bool	Cette sortie est TRUE lorsque la dernière requête s'est achevée avec des erreurs. Dans ce cas, la sortie STATUS contient en outre les codes d'erreur correspondants.
STATUS	OUT	Word	Code d'erreur (valeur par défaut : 0)

13.3.5.9 Signal_Get (Lire les signaux RS-232)

Tableau 13- 31 Instruction Signal_Get (Lire signaux RS232)

CONT / LOG	SCL	Description
<p>*Signal_Get_DB*</p> 	<pre>"Signal_Get_DB" (REQ:=_bool_in_, PORT:=_uint_in_, NDR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DTR=>_bool_out_, DSR=>_bool_out_, RTS=>_bool_out_, CTS=>_bool_out_, DCD=>_bool_out_, RING=>_bool_out_);</pre>	<p>Signal_Get lit l'état en cours des signaux de communication RS232. Cette fonction n'est valable que pour le module de communication RS232.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 32 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Lecture des valeurs d'état de signal RS232 en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
NDR	OUT	Bool	TRUE pour un cycle lorsque de nouvelles données sont prêtes et que l'opération s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque l'opération s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)
DTR	OUT	Bool	Terminal de données prêt, module prêt (sortie). Valeur par défaut : False
DSR	OUT	Bool	Modem prêt, partenaire de communication prêt (entrée). Valeur par défaut : False
RTS	OUT	Bool	Demande pour émettre, module prêt à émettre (sortie). Valeur par défaut : False
CTS	OUT	Bool	Prêt à émettre, le partenaire de communication peut recevoir les données (entrée). Valeur par défaut : False
DCD	OUT	Bool	Détection de porteuse, réception du niveau de signal (toujours faux, non pris en charge)
RING	OUT	Bool	Indicateur d'appel, signale un appel entrant (toujours faux, non pris en charge)

Tableau 13- 33 Codes d'erreur

STATUS (W#16#....)	Description
81F0	Le CM ou le CB est un module RS485 et il n'y a pas de signaux disponibles.
81F4	Erreur dans l'en-tête du bloc, par exemple, type de bloc erroné ou longueur de bloc erronée
8280	Acquittement négatif pendant la lecture du module
8282	Le module ou l'esclave DP n'est pas disponible

13.3.5.10 Signal_Set (Régler les signaux RS-232)

Tableau 13- 34 Instruction Signal_Set (Activer signaux RS232)

CONT / LOG	SCL	Description
<pre> *Signal_Set_DB* Signal_Set - EN ENO - - REQ DONE - - PORT ERROR - - SIGNAL STATUS - - RTS - DTR - DSR </pre>	<pre> "Signal_Set_DB" (REQ:=_bool_in_, PORT:=_word_in_, SIGNAL:=_byte_in_, RTS:=_bool_in_, DTR:=_bool_in_, DSR:=_bool_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>Signal_Set active l'état des signaux de communication RS232.</p> <p>Cette fonction n'est valable que pour le module de communication RS232.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

13.3 Communication point à point (PtP)

Tableau 13- 35 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Lancement de l'opération d'activation des signaux RS232 en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
SIGNAL	IN	Byte	Sélection du signal à activer : (plusieurs autorisés). Valeur par défaut : 0 <ul style="list-style-type: none"> • 01H = Activer RTS • 02H = Activer DTR • 04H = Activer DSR
RTS	IN	Bool	Valeur à définir Demande pour émettre, module prêt à émettre (true ou false), valeur par défaut : False
DTR	IN	Bool	Valeur à définir Terminal de données prêt, module prêt (true ou false). Valeur par défaut : False
DSR	IN	Bool	Modem prêt (valable uniquement pour les interfaces de type DCE), non utilisé.
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Tableau 13- 36 Codes d'erreur

STATUS (W#16#....)	Description
81F0	Le CM ou le CB est un module RS485 et il n'y a pas de signaux pouvant être activés.
81F1	Les signaux ne peuvent être activés en raison d'un contrôle de flux matériel.
81F2	DSR ne peut pas être activé car le module est un équipement terminal de données (DTE).
81F3	DTR ne peut pas être activé car le module est un équipement de transmission de données (DCE).
81F4	Erreur dans l'en-tête du bloc, par exemple, type de bloc erroné ou longueur de bloc erronée
8280	Acquittement négatif pendant la lecture du module
8281	Acquittement négatif pendant l'écriture du module
8282	Le module ou l'esclave DP n'est pas disponible

13.3.5.11 Get_Features

Tableau 13- 37 Instruction Get_Features (Lire des fonctionnalités avancées)

CONT / LOG	SCL	Description
<pre> *Get_Features_ DB* Get_Features - EN ENO - - REQ NDR - - PORT ERROR - STATUS - MODBUS_CRC - DIAG_ALARM - SUPPLY_VOLT - </pre>	<pre> "Get_Features_DB" (REQ:=_bool_in_, PORT:=_word_in_, NDR:=_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MODBUS_CRC=>_bool_out_, DIAG_ALARM=>_bool_out_, SUPPLY_VOLT=>_bool_out_); </pre>	<p>Get_Features effectue la lecture des capacités de fonctionnalités avancées d'un moduleur.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Utilisez l'instruction Get_Features pour lire les capacités de fonctionnalités avancées d'un module.

13.3 Communication point à point (PtP)

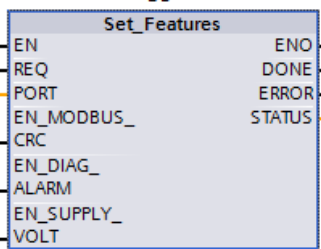
Tableau 13- 38 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
NDR	OUT	Bool	Indique que les nouvelles données sont prêtes.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)
MODBUS_CRC*	OUT	Bool	Génération et vérification de MODBUS CRC
DIAG_ALARM*	OUT	Bool	Génération d'alarme de diagnostic
SUPPLY_VOLT*	OUT	Bool	Un diagnostic pour la tension d'alimentation manquante L+ est disponible

*Get_Features renvoie VRAI (1) si la fonctionnalité est disponible, FAUX (0) si la fonctionnalité n'est pas disponible

13.3.5.12 Set_Features

Tableau 13- 39 Instruction Set_Features (Lire des fonctionnalités avancées)

CONT / LOG	SCL	Description
<p>*Set_Features_DB*</p> 	<pre>"Set_Features_DB" (REQ:=_bool_in_, PORT:=_word_in_, EN_MODBUS_CRC:=_bool_in_, EN_DIAG_ALARM:=_bool_in_, EN_SUPPLY_VOLT:=_bool_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	<p>Set_Features active les fonctionnalités avancées acceptées par un module.</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Utilisez l'instruction Set_Features pour lire les capacités de fonctionnalités avancées d'un module.

Tableau 13- 40 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
EN_MODBUS_CRC	IN	Bool	Activer la génération et la vérification de MODBUS CRC : <ul style="list-style-type: none"> • 0: Calcul CRC désactivé (par défaut) • 1: Calcul CRC activé Remarque : Seuls les modules CM V2.1, CPU V4.1 avec CB, et PtP CM pour la périphérie décentralisée supportent ce paramètre.
EN_DIAG_ALARM	IN	Bool	Activer la génération d'alarme de diagnostic : <ul style="list-style-type: none"> • 0: Alarme de diagnostic désactivée • 1: Alarme de diagnostic activée (par défaut)
EN_SUPPLY_VOLT	IN	Bool	Activer le diagnostic pour la tension d'alimentation manquante L+ : <ul style="list-style-type: none"> • 0: Diagnostic de la tension d'alimentation désactivé (par défaut) • 1: Diagnostic de la tension d'alimentation activé
DONE	OUT	Bool	Indique que la définition des fonctionnalités est terminée
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

13.3.6 Programmation de la communication point à point

STEP 7 fournit des instructions avancées qui permettent au programme utilisateur d'exécuter la communication point à point avec un protocole conçu et mis en œuvre dans le programme utilisateur. Ces instructions se subdivisent en deux catégories :

- Instructions de configuration
- Instructions de communication

Instructions de configuration

Avant que votre programme utilisateur ne puisse exécuter une communication point à point, vous devez configurer le port de l'interface de communication ainsi que les paramètres pour l'émission et la réception de données.

Vous pouvez effectuer la configuration du port et des messages pour chaque CM ou CB via la configuration d'appareil ou par le biais de ces instructions dans votre programme utilisateur :

- Port_Config (Page 1158)
- Send_Config (Page 1162)
- Receive_Config (Page 1164)

Instructions de communication

Les instructions de communication point à point permettent au programme utilisateur d'envoyer des messages aux interfaces de communication et d'en recevoir. Pour plus d'informations sur le transfert de données à l'aide de ces instructions, reportez-vous au paragraphe sur la cohérence des données (Page 204).

Toutes les fonctions point à point opèrent de manière asynchrone. Le programme utilisateur peut utiliser une architecture d'interrogation pour déterminer l'état des émissions et des réceptions. Send_P2P et Receive_P2P peuvent s'exécuter simultanément. Les modules de communication et le Communication Board mettent les messages d'émission et de réception en mémoire tampon selon les besoins, jusqu'à une taille de mémoire tampon maximale de 1024 octets.

Les CM et le CB envoient des messages aux appareils point à point réels et en reçoivent. Le protocole du message est dans une mémoire tampon qui est reçue d'un port de communication spécifique ou lui est envoyée. La mémoire tampon et le port sont des paramètres des instructions d'émission et de réception :

- Send_P2P (Page 1173)
- Receive_P2P (Page 1177)

D'autres instructions permettent de réinitialiser la mémoire tampon de réception ainsi que de lire et d'activer des signaux RS232 spécifiques :

- Receive_Reset (Page 1179)
- Signal_Get (Page 1180)
- Signal_Set (Page 1181)

13.3.6.1 Architecture d'interrogation

Le programme utilisateur STEP 7 doit appeler les instructions point à point du S7-1200 cycliquement/périodiquement pour vérifier si des messages ont été reçus. L'interrogation de l'émission indique au programme utilisateur quand l'émission s'est achevée.

Architecture d'interrogation : maître

La séquence typique pour un maître se présente comme suit :

1. Une instruction Send_P2P (Page 1173) déclenche une transmission au CM ou au CB.
2. L'instruction Send_P2P s'exécute lors des cycles suivants pour interroger l'état d'achèvement de l'émission.
3. Lorsque l'instruction Send_P2P indique que l'émission est achevée, le code utilisateur peut préparer la réception de la réponse.
4. L'instruction Receive_P2P (Page 1177) s'exécute de manière répétée pour vérifier s'il y a une réponse. Lorsque le CM ou le CB a collecté un message de réponse, l'instruction Receive_P2P copie la réponse dans la CPU et signale que de nouvelles données ont été reçues.
5. Le programme utilisateur peut traiter la réponse.
6. Retournez à l'étape 1 et répétez le cycle.

Architecture d'interrogation : esclave

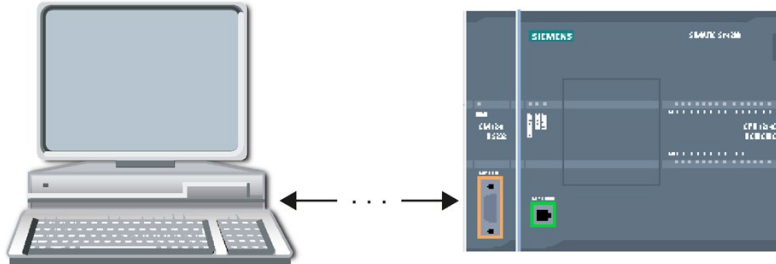
La séquence typique pour un esclave se présente comme suit :

1. Le programme utilisateur exécute l'instruction Receive_P2P à chaque cycle.
2. Lorsque le CM ou le CB a reçu une demande, l'instruction Receive_P2P signale que de nouvelles données sont prêtes et la demande est copiée dans la CPU.
3. Le programme utilisateur honore la demande et génère une réponse.
4. Utilisez une instruction Send_P2P pour renvoyer la réponse au maître.
5. Exécutez Send_P2P de manière répétée pour être sûr que l'émission a lieu.
6. Retournez à l'étape 1 et répétez le cycle.

L'esclave doit appeler Receive_P2P suffisamment souvent pour recevoir une émission du maître avant que le délai d'attente de réponse n'expire chez le maître. A cet effet, le programme utilisateur peut appeler RCV_PTP à partir d'un OB d'alarme cyclique dont la période est suffisante pour recevoir une émission du maître avant que le délai d'attente n'expire. Si vous définissez la période de l'OB de manière à permettre deux exécutions pendant le délai d'attente du maître, le programme utilisateur peut recevoir les émissions sans en manquer aucune.

13.3.7 Exemple : Communication point à point

Dans cet exemple, une CPU S7-1200 communique avec un PC muni d'un émulateur de terminal par le biais d'un module CM 1241 RS232. La configuration point à point et le programme STEP 7 dans cet exemple illustrent comment la CPU peut recevoir un message du PC et renvoyer le message en écho au PC.



Vous devez connecter l'interface de communication du module CM 1241 RS232 à l'interface RS232 du PC qui est normalement COM1. Comme ces deux ports sont des équipements terminaux de données (DTE), vous devez permuter les broches d'émission et de réception (2 et 3) lorsque vous connectez les ports, ce que vous faites à l'aide de l'une des deux méthodes suivantes :

- Utilisez un adaptateur de modem nul pour permuter les broches 2 et 3 avec un câble RS232 standard.
- Utilisez un câble de modem nul dont les broches 2 et 3 sont déjà permutées. Vous pouvez habituellement identifier un câble de modem nul par ses deux extrémités à connecteur D 9 broches femelle.

13.3.7.1 Configuration du module de communication

Vous pouvez configurer le CM 1241 dans la configuration des appareils de STEP 7 ou par le biais d'instructions dans le programme utilisateur. Dans cet exemple, nous utilisons la configuration des appareils.

- Configuration du port : Cliquez sur le port de communication du module CM dans la configuration des appareils et configurez le port comme illustré :



Configuration du port

Mode de fonctionnement

Duplex intégral (RS422) mode sur 4 fils (couplage point à point)

Duplex intégral (RS422) mode sur 4 fils (maître multipoint)

Duplex intégral (RS422) mode sur 4 fils (esclave multipoint)

Semi-duplex (RS485) mode sur 2 fils

Préconfiguration de la ligne de réception

Néant

Tension à l'état passant (signal R(A) 0V, signal R(B) 5V)

Vitesse de transmission : 9,6 kbit

Parité : Pas de parité

Bits de données : 8 bits par caractère

Bit d'arrêt : 1

Contrôle du flux : Néant

Caractère XON (HEXA) : 0

(ASCII) : NUL

Caractère XOFF (HEXA) : 0

(ASCII) : NUL

Temps d'attente : 1 ms

Remarque

Les paramètres pour "Mode de fonctionnement" et "Préconfiguration de la ligne de réception" valent uniquement pour le module CM 1241 (RS422/RS485). Ces paramètres de configuration de port n'existent pas pour les autres modules CM 1241. Reportez-vous à Configuration de RS422 et RS485 (Page 1192).

13.3 Communication point à point (PtP)

- Configuration de l'émission de messages : Acceptez les valeurs par défaut pour la configuration de l'émission de messages. Aucune pause ne doit être envoyée en début de message.
- Configuration du début de la réception de message : Configurez le CM 1241 pour qu'il commence la réception de message lorsque la ligne de communication a été inactive pendant au moins 50 temps bit (environ 5 millisecondes à 9600 bauds = $50 * 1/9600$) :

Début de message

Commencer par un caractère quelconque

Commencer avec condition spéciale

Détecter le début de message à l'aide de Line Break

Détecter le début de message à l'aide de Idle Line

Durée de Idle Line : 50 temps de bit

Détecter le début de message à l'aide de caractères

Caractère début de message (HEX) : 2

Caractère début de message (ASCII) : STX

Détecter le début de message à l'aide d'une chaîne de caractères

Nombre de séquences de caractères à définir : 1

- Configuration de la fin de la réception de message : Configurez le CM 1241 pour qu'il mette fin à un message lorsqu'il a reçu 100 octets au maximum ou un caractère de changement de ligne (10 décimal ou a hexadécimal). La séquence de fin autorise jusqu'à cinq caractères de fin à la suite. Le cinquième caractère de la séquence est le caractère de changement de ligne. Les quatre caractères précédents de la séquence sont des caractères indifférents. Le CM 1241 n'évalue pas les caractères indifférents mais recherche un caractère de changement de ligne précédé de zéro ou de plusieurs caractères indifférents pour indiquer la fin de message.

Fin de message

Définir les conditions de fin de message

Détecter la fin de message à l'aide du dépassement de temps message

Dépassement de temps message : 200 ms

Détecter la fin de message à l'aide du dépassement de temps de réponse

Dépassement du temps de réponse : 200 ms

Détecter la fin de message à l'aide d'un dépassement du délai inter-caractères

Dépassement du délai inter-caractères : 12 temps de bit

Détecter la fin de message à l'aide de la longueur maximale

Longueur maximale du message : 100 bytes

Lire la longueur dans le message

Décalage du champ de longueur dans le message : 1 bytes

Taille du champ de longueur : 1 bytes

Le champ de longueur suivant les données ne se réfère pas ... 0 bytes

Détecter la fin de message à l'aide d'une chaîne de caractères

Séquence de 5 caractères pour fin de message

Contrôler ce caractère : 1

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 2

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 3

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 4

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 5

Caractère (HEXA) : A

Caractère (ASCII) : LF

13.3.7.2 Modes de fonctionnement RS422 et RS485

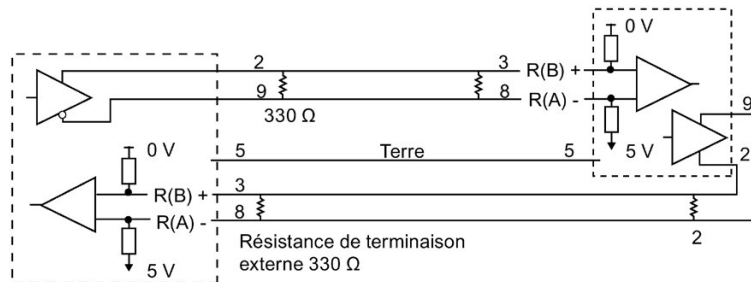
Configuration de RS422

Pour le mode RS422, il existe trois modes de fonctionnement selon la configuration de votre réseau. Sélectionnez l'un de ces modes de fonctionnement en fonction des appareils dans votre réseau. Les différents choix pour Préconfiguration de la ligne de réception font référence aux cas détaillés plus loin.

- Duplex intégral (RS 422) mode sur 4 fils (couplage point à point) : sélectionnez cette option lorsqu'il y a plus de deux appareils connectés à votre réseau. Pour Préconfiguration de la ligne de réception :
 - Sélectionnez Aucune lorsque vous fournissez la polarisation et la terminaison (cas 3).
 - Sélectionnez Polarisation directe pour utiliser la polarisation et la terminaison internes (cas 2).
 - Sélectionnez Polarisation inverse pour utiliser la polarisation et la terminaison internes et activer la détection de rupture de fil pour les deux appareils (cas 1).
- Duplex intégral (RS422) mode sur 4 fils (maître multipoint) : sélectionnez cette option pour le maître lorsque vous avez un réseau avec un maître et plusieurs esclaves. Pour Préconfiguration de la ligne de réception :
 - Sélectionnez Aucune lorsque vous fournissez la polarisation et la terminaison (cas 3).
 - Sélectionnez Polarisation directe pour utiliser la polarisation et la terminaison internes (cas 2).
 - La détection de rupture de fil n'est pas possible dans ce mode.
- Duplex intégral (RS422) mode sur 4 fils (esclave multipoint) : sélectionnez cette option pour tous les esclaves lorsque vous avez un réseau avec un maître et plusieurs esclaves. Pour Préconfiguration de la ligne de réception :
 - Sélectionnez Aucune lorsque vous fournissez la polarisation et la terminaison (cas 3).
 - Sélectionnez Polarisation directe pour utiliser la polarisation et la terminaison internes (cas 2).
 - Sélectionnez Polarisation inverse pour utiliser la polarisation et la terminaison internes et activer la détection de rupture de fil pour les esclaves (cas 1).

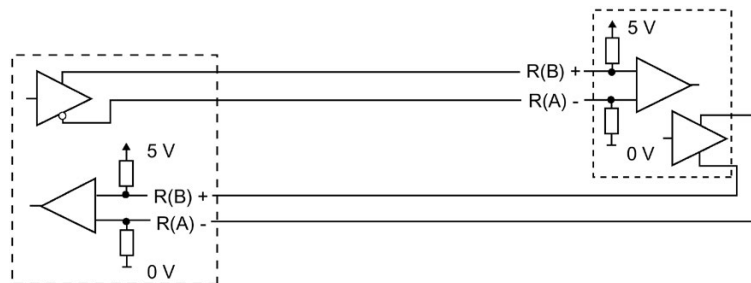
Cas 1 : RS422 avec détection de rupture de fil

- Mode de fonctionnement : RS422
- Préconfiguration de la ligne de réception : polarisation inverse (polarisé avec $R(A) > R(B) > 0V$)
- Rupture de câble : détection de rupture de câble activée (émetteur toujours actif)



Cas 2 : RS422 sans détection de rupture de câble, polarisation directe

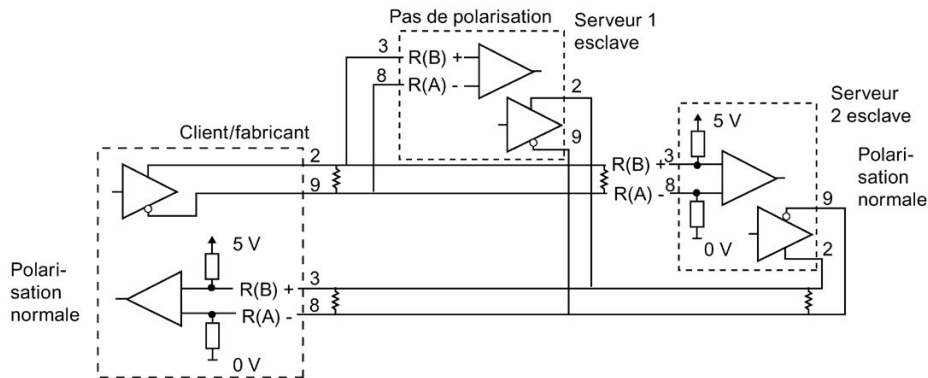
- Mode de fonctionnement : RS422
- Préconfiguration de la ligne de réception : Tension à l'état passant (polarisé avec $R(A) > R(B) > 0V$)
- Rupture de câble : pas de détection de rupture de câble (émetteur actif uniquement lors de l'émission)



Cas 3 : RS422 : sans détection de rupture de câble, pas de polarisation

- Mode de fonctionnement : RS422
- Préconfiguration de la ligne de réception : Pas de polarisation
- Rupture de câble : pas de détection de rupture de câble (émetteur actif uniquement lors de l'émission)

La polarisation et la terminaison sont ajoutées par l'utilisateur aux noeuds d'extrémité du réseau.



Configuration de RS485

Il n'y a qu'un mode de fonctionnement pour le mode RS485. Les différents choix pour Préconfiguration de la ligne de réception font référence aux cas détaillés plus loin.

- Semi-duplex (RS485) mode sur 2 fils. Pour Préconfiguration de la ligne de réception :
 - Sélectionnez Aucune lorsque vous fournissez la polarisation et la terminaison (cas 5).
 - Sélectionnez Polarisation directe pour utiliser la polarisation et la terminaison internes (cas 4).

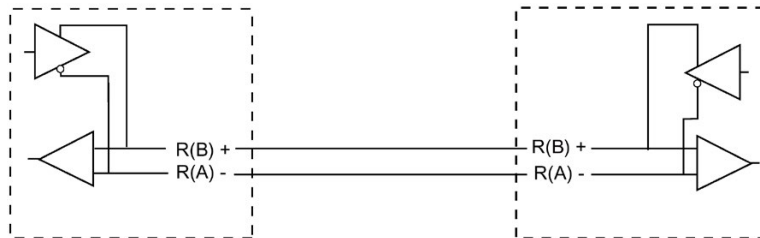
Cas 4 : RS485 : polarisation directe

- Mode de fonctionnement : RS485
- Préconfiguration de la ligne de réception : Tension à l'état passant (polarisé avec $R(A) > R(B) > 0 V$)



Cas 5 : RS485 : pas de polarisation (polarisation externe)

- Mode de fonctionnement : RS485
- Préconfiguration de la ligne de réception : pas de polarisation (polarisation externe nécessaire)

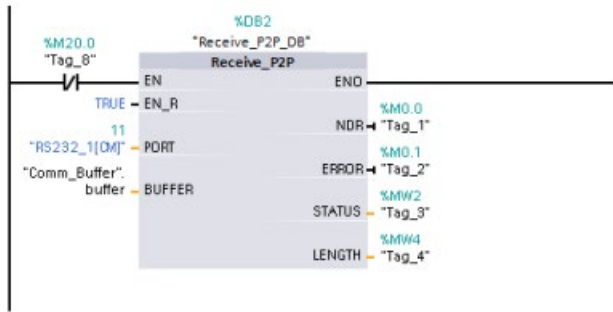


13.3.7.3 Programmation du programme STEP 7

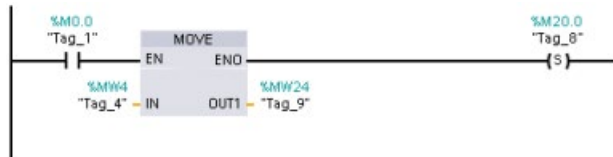
L'exemple de programme utilise un bloc de données global pour la mémoire tampon de communication, une instruction RCV_PTP (Page 1293) pour recevoir des données de l'émulateur de terminal, ainsi qu'une instruction SEND_PTP (Page 1291) pour renvoyer la mémoire tampon en écho à l'émulateur de terminal. Pour programmer l'exemple, ajoutez la configuration du bloc de données et programmez l'OB 1 principal comme décrit ci-dessous.

Bloc de données global "Comm_Buffer" : Créez un bloc de données global (DB) et nommez-le "Comm_Buffer". Dans le bloc de données, créez une valeur appelée "buffer" et ayant le type de données "array [0 .. 99] of byte".

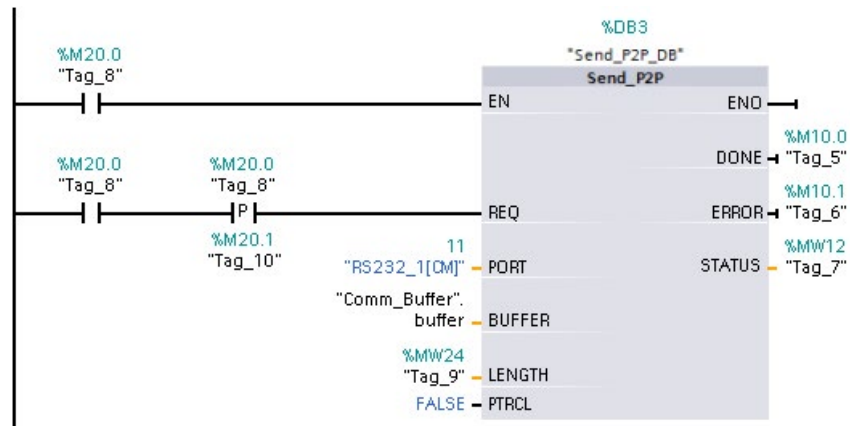
Réseau 1 : Activez l'instruction RCV_PTP à chaque fois que SEND_PTP n'est pas actif. Tag_8 en MW20.0 signale lorsque l'émission est achevée dans le réseau 4 et lorsque le module de communication est ainsi prêt à recevoir un message.



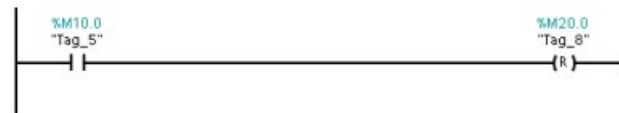
Réseau 2 : Utilisez la valeur NDR (Tag_1 en M0.0) définie par l'instruction RCV_PTP pour copier le nombre d'octets reçus et pour mettre un memento à 1 (Tag_8 en M20.0) afin de déclencher l'instruction SEND_PTP.



Réseau 3 : Activez l'instruction SEND_PTP lorsque le memento M20.0 est à 1. Utilisez également ce memento pour mettre l'entrée REQ à VRAI pour un cycle. L'entrée REQ indique à l'instruction SEND_PTP qu'une nouvelle demande doit être émise. L'entrée REQ ne doit être à VRAI que pendant une exécution de SEND_PTP. L'instruction SEND_PTP est exécutée à chaque cycle jusqu'à ce que l'émission s'achève. L'émission est achevée lorsque le dernier octet du message a été émis depuis le CM 1241. Lorsque l'émission est achevée, la sortie DONE (Tag_5 en M10.0) est mise à VRAI pour une exécution de SEND_PTP.



Réseau 4 : Surveillez la sortie DONE de SEND_PTP et remettez le memento d'émission à 0 (Tag_8 en M20.0) lorsque l'opération d'émission est achevée. Lorsque le memento d'émission est à 0, l'instruction RCV_PTP dans le réseau 1 est activée afin de recevoir le message suivant.



13.3.7.4 Configuration de l'émulateur de terminal

Vous devez configurer l'émulateur de terminal afin qu'il prenne en charge l'exemple de programme. Vous pouvez utiliser presque tout émulateur de terminal sur votre PC, HyperTerminal par exemple. Assurez-vous que l'émulateur de terminal est en mode déconnecté avant d'éditer les paramètres comme suit :

1. Réglez l'émulateur de terminal afin qu'il utilise le port RS232 sur le PC (COM1 normalement).
2. Configurez le port à 9600 bauds, 8 bits de données, sans parité, 1 bit d'arrêt et sans contrôle de flux.
3. Modifiez les paramètres de l'émulateur de terminal afin qu'il émule un terminal ANSI.
4. Réglez la configuration ASCII de l'émulateur de terminal pour l'envoi d'un changement de ligne après chaque ligne (lorsque l'utilisateur appuie sur la touche Entrée).
5. Renvoyez les caractères localement de sorte que l'émulateur de terminal affiche ce qui est tapé.

13.3.7.5 Exécution de l'exemple

Procédez comme suit pour exécuter l'exemple de programme :

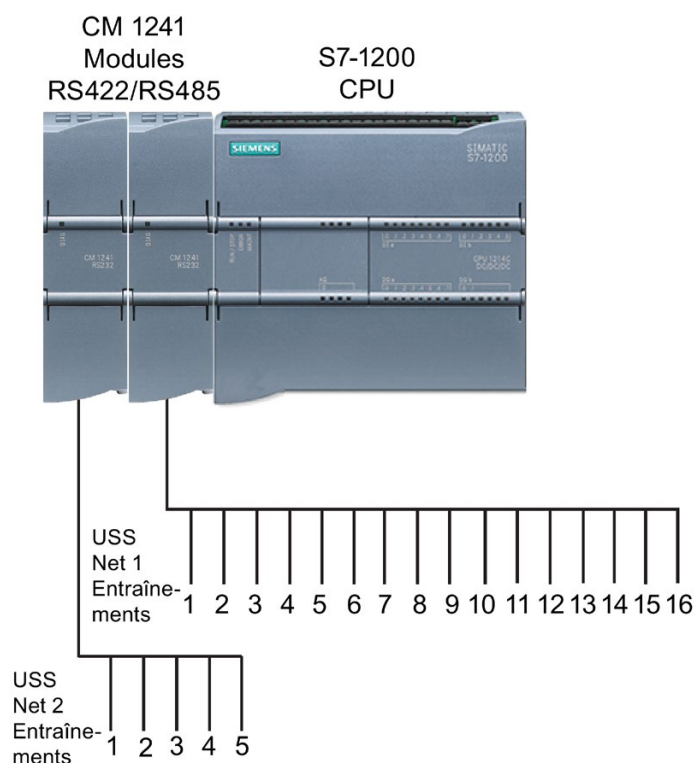
1. Chargez le programme STEP 7 dans la CPU et assurez-vous qu'elle est à l'état MARCHE.
2. Cliquez sur le bouton "connecter" de l'émulateur de terminal pour appliquer les modifications de configuration et ouvrir une session de terminal vers le CM 1241.
3. Tapez des caractères sur le PC et appuyez sur Entrée.

L'émulateur de terminal envoie les caractères au CM 1241 et à la CPU. Le programme de la CPU renvoie alors les caractères en écho à l'émulateur de terminal.

13.4 Communication USS (interface série universelle)

Les instructions USS commandent le fonctionnement d'entraînements de moteur qui prennent en charge le protocole USS (interface série universelle). Vous pouvez utiliser les instructions USS pour communiquer avec plusieurs entraînements via des liaisons RS485 vers des modules de communication CM 1241 RS485 ou un Communication Board CB 1241 RS485. Il est possible d'installer jusqu'à trois modules CM 1241 RS422/RS485 et un CB 1241 RS485 dans une CPU S7-1200. Chaque port RS485 peut faire fonctionner seize entraînements au plus.

Le protocole USS utilise un réseau maître-esclave pour communiquer par le biais d'un bus série. Le maître utilise un paramètre adresse pour envoyer un message à un esclave sélectionné. Un esclave ne peut lui jamais émettre s'il n'y a pas d'abord été invité par un maître. Un transfert de message direct entre les esclaves individuels est impossible. La communication USS opère en mode semi-duplex. L'illustration suivante de USS montre un schéma de réseau pour un exemple d'application d'entraînement.



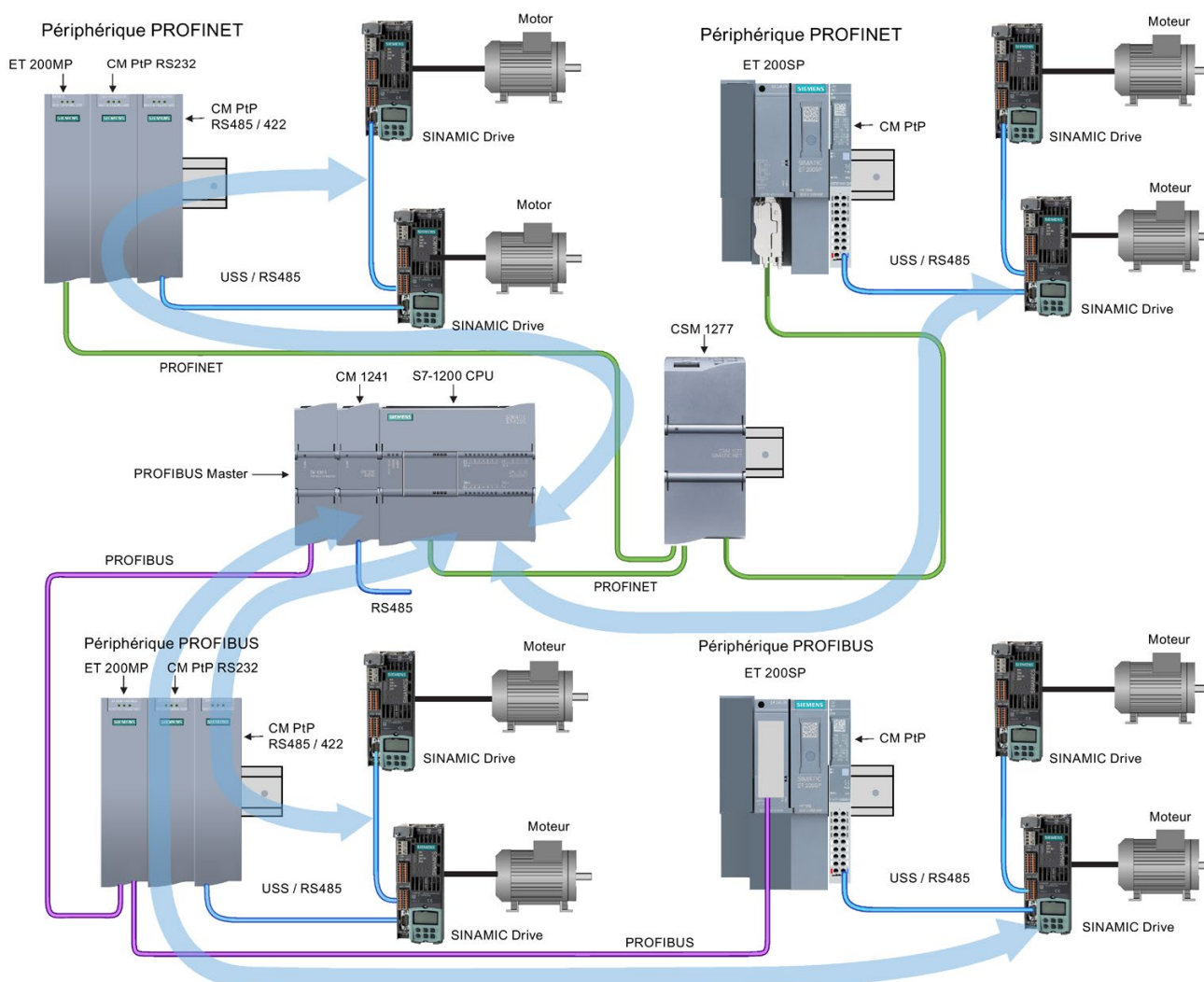
Communications USS à travers PROFIBUS ou PROFINET

À partir de la version V4.1 de la CPU S7-1200 avec STEP 7 V13 SP1, la CPU étend les fonctionnalités de USS à l'utilisation d'un châssis de périphérie décentralisée PROFINET ou PROFIBUS pour communiquer avec divers appareils (lecteurs RFID, dispositifs GPS et autres) :

- PROFIBUS (Page 830) : Vous connectez l'interface Ethernet de la CPU S7-1200 à un coupleur PROFIBUS. Les modules de communication PtP dans le châssis avec le coupleur peuvent ensuite fournir des communications série avec les appareils PtP.
- PROFIBUS (Page 1002) : Vous insérez un module de communication PROFIBUS dans le côté gauche du châssis avec la CPU S7-1200. Vous connectez le module de communication PROFIBUS à un châssis contenant un coupleur PROFIBUS. Les modules de communication PtP dans le châssis avec le coupleur peuvent ensuite fournir des communications série avec les appareils PtP.

Pour cette raison, le S7-1200 accepte deux jeux d'instructions PtP :

- Instructions d'héritage USS (Page 1305) : Ces instructions USS existaient avec la version V4.0 du S7-1200 et fonctionnent uniquement avec des communications série à l'aide d'un module de communication CM 1241 ou d'un communication board CB 1241.
- Instructions USS (Page 1206) : Ces instructions USS offrent toutes les fonctionnalités des instructions d'héritage, plus la capacité à se connecter à la périphérie décentralisée PROFINET et PROFIBUS. Ces instructions USS vous permettent de configurer les communications entre les modules de communication PtP dans le châssis de périphérie décentralisée et les appareils PtP. Les modules CM 1241 S7-1200 doivent posséder au moins la version V2.1 du firmware pour utiliser ces instructions USS.



Les flèches bleues indiquent le flux de la communication bidirectionnelle entre les appareils.

Remarque

Avec la version V4.1 du S7-1200, vous pouvez utiliser les instructions point à point pour tous les types de communication point à point : série, série sur PROFINET, et série sur PROFIBUS. STEP 7 fournit les instructions d'héritage point à point uniquement pour prendre en charge les programmes existants. Les instructions d'héritage fonctionnent toujours avec toutes les CPU S7-1200. Vous n'avez pas besoin de convertir les programmes antérieurs d'un jeu d'instructions à l'autre.

13.4.1 Sélection de la version des instructions USS

Deux versions des instructions USS sont disponibles dans STEP 7 :

- La version 2.0 (instructions d'héritage) était à l'origine disponible dans STEP 7 Basic/Professional V13.
- La version 2.1 est disponible dans STEP 7 Basic/Professional V13 SP1 ou ultérieur.

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

Vous ne pouvez pas utiliser les deux versions des instructions avec le même module, mais deux modules différents peuvent utiliser différentes versions des instructions.



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.

		V2.1
USS communication		
USS_Port_Scan	Communication via US...	V2.1
USS_Drive_Control	Data exchange with th...	V2.0
USS_Read_Param	Read data from drive	V2.1
USS_Write_Param	Change data in drive	V1.4

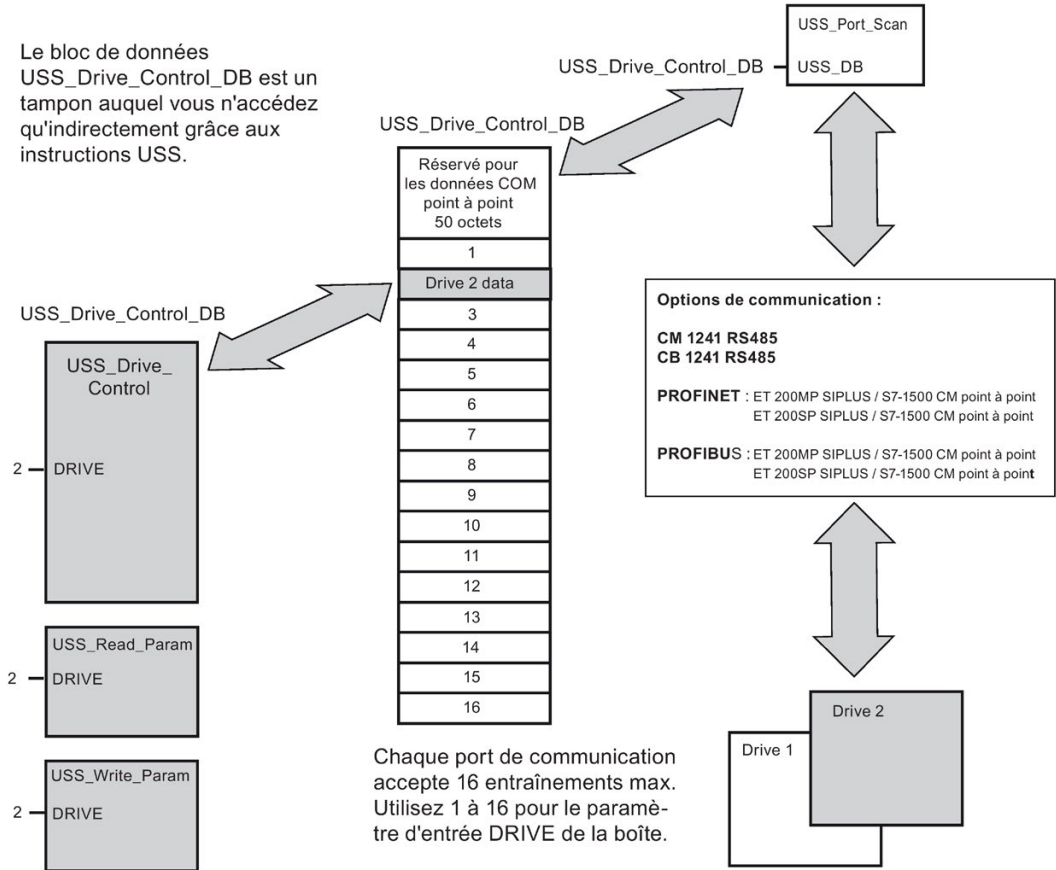
Pour changer la version des instructions USS, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction USS dans votre programme, une nouvelle instance de FB ou FC, en fonction de l'instruction USS sélectionnée, est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB ou FC dans l'arborescence du projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction USS dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB ou FC USS dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction USS.

13.4.2 Conditions requises pour l'utilisation du protocole USS

Les quatre instructions USS utilisent deux blocs fonctionnels (FB) et deux fonctions (FC) pour prendre en charge le protocole USS. Un bloc de données d'instance USS_Port_Scan est utilisé pour chaque réseau USS. Le bloc de données d'instance USS_Port_Scan contient de la mémoire temporaire et des mémoires tampons pour tous les entraînements de ce réseau USS. Les instructions USS partagent les informations dans ce bloc de données.



Tous les entraînements (16 au maximum) connectés à un même port RS485 font partie du même réseau USS. Tous les entraînements connectés à un port RS485 différent font partie d'un réseau USS différent. Chaque réseau USS est géré à l'aide d'un bloc de données unique. Toutes les instructions associées à un même réseau USS doivent partager ce bloc de données. Cela inclut toutes les instructions USS_Drive_Control, USS_Port_Scan, USS_Read_Param et USS_Write_Param utilisées pour commander tous les entraînements dans un même réseau USS.

13.4 Communication USS (interface série universelle)

L'instruction USS_Drive_Control est un bloc fonctionnel (FB). Lorsque vous placez l'instruction USS_Drive_Control dans l'éditeur de programmes, la boîte de dialogue "Options d'appel" s'affiche et vous demande d'affecter un DB à ce FB. S'il s'agit de la première instruction USS_Drive_Control dans ce programme pour ce réseau USS, vous pouvez accepter le DB indiqué par défaut (ou modifier son nom si vous le désirez) et ce nouveau DB est alors créé pour vous. En revanche, s'il ne s'agit pas de la première instruction USS_Drive_Control pour cette voie, vous devez utiliser la liste déroulante dans la boîte de dialogue "Options d'appel" pour sélectionner le nom de DB précédemment affecté à ce réseau USS.

L'instruction USS_Port_Scan est un bloc fonctionnel (FB) et gère la communication effective entre la CPU et les entraînements à travers le port de communication RS485 point à point. Chaque appel de ce FB gère une communication avec un entraînement. Votre programme doit appeler ce FB suffisamment rapidement pour éviter un dépassement du délai de communication par les entraînements. Vous pouvez appeler ce FB fonctionnel dans un OB de cycle de programme principal ou dans n'importe quel OB d'alarme.

Les instructions USS_Read_Param et USS_Write_Param sont toutes deux des Fonctions (FC). Il n'y a pas d'affectation de DB lorsque vous placez ces FC dans l'éditeur. En revanche, vous devez affecter le DB approprié à l'entrée "USS_DB" de ces instructions. Double-cliquez sur le champ de paramètre, puis cliquez sur l'icône d'aide pour voir les noms de DB disponibles.

Typiquement, vous appellerez le FB USS_Port_Scan dans un OB d'alarme cyclique. Définissez la période de cet OB d'alarme cyclique à environ la moitié de l'intervalle d'appel minimum. (Une communication à 1200 bauds devrait, par exemple, utiliser une période d'au plus 350 ms).

Le FB USS_Drive_Control permet à votre programme d'accéder à un entraînement spécifié dans le réseau USS. Ses entrées et sorties constituent les états et les commandes pour l'entraînement. S'il y a 16 entraînements dans le réseau, votre programme doit comporter au moins 16 appels de USS_Drive_Control, un pour chaque entraînement. Vous devez appeler ces blocs à la fréquence nécessaire pour commander le fonctionnement de l'entraînement.

Vous ne pouvez appeler le FB USS_Drive_Control que dans un OB de cycle de programme principal.

 **PRUDENCE**

Considérations sur l'appel d'instructions USS dans des OB

Appelez USS_Drive_Control, USS_Read_Param et USS_Write_Param uniquement à partir d'un OB de cycle de programme. Le FB USS_Port_Scan peut être appelé dans n'importe quel OB, généralement un OB d'alarme cyclique.

N'utilisez pas les instructions USS_Drive_Control, USS_Read_Param et USS_Write_Param dans un OB de priorité supérieure à celle de l'instruction USS_Port_Scan correspondante. Ainsi, ne placez pas USS_Port_Scan dans l'OB principal et USS_Read_Param dans un OB d'alarme cyclique. L'interruption possible de USS_Port_Scan peut produire des erreurs inattendues qui peuvent entraîner des blessures.

Les FC USS_Read_Param et USS_Write_Param lisent et écrivent les paramètres de fonctionnement de l'entraînement distant. Ces paramètres pilotent le fonctionnement interne de l'entraînement. Vous trouverez la définition de ces paramètres dans le manuel de l'entraînement. Votre programme peut contenir autant de ces fonctions que nécessaire, mais une seule demande de lecture ou d'écriture peut être active par entraînement à un moment donné. Vous ne pouvez appeler les FC USS_Read_Param et USS_Write_Param que dans un OB de cycle de programme principal.

Calcul du temps requis pour communiquer avec l'entraînement

La communication avec l'entraînement est asynchrone par rapport au cycle S7-1200. Typiquement, le S7-1200 exécute plusieurs cycles avant qu'une transaction de communication avec un entraînement s'achève.

L'intervalle USS_Port_Scan est le temps requis pour une transaction avec un entraînement. Le tableau ci-dessous montre l'intervalle USS_Port_Scan minimum pour chaque vitesse de transmission. Appeler le FB USS_Port_Scan plus fréquemment que l'intervalle USS_Port_Scan n'augmente pas le nombre de transactions. Le délai d'attente de l'entraînement est la quantité de temps qui pourrait être utilisée pour une transaction si des erreurs de communication entraînaient 3 tentatives pour achever la transaction. Par défaut, la bibliothèque du protocole USS exécute automatiquement jusqu'à 2 nouvelles tentatives pour chaque transaction.

Tableau 13- 41 Calcul des exigences en temps

Débit	Intervalle d'appel USS_Port_Scan minimum calculé (millisecondes)	Délai d'attente de message par entraînement (millisecondes)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

13.4.3 Opérations USS

13.4.3.1 USS_Port_Scan (Éditer la communication via le réseau USS)

Tableau 13- 42 Instruction USS_Port_Scan

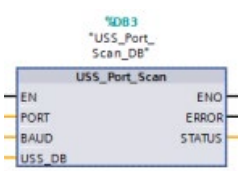
CONT/LOG	SCL	Description
	<pre>USS_Port_Scan(PORT:=_uint_in_, BAUD:=_dint_in_, ERROR=>_bool_out_, STATUS=>_word_out_, USS_DB:=_fbtref_inout_);</pre>	<p>L'instruction USS_Port_Scan gère la communication par le biais d'un réseau USS.</p>

Tableau 13- 43 Types de données pour les paramètres

Paramètre et type		Type de données	Description
PORT	IN	Port	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
BAUD	IN	DInt	Vitesse de transmission à utiliser pour la communication USS
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_Drive_Control dans votre programme.
ERROR	OUT	Bool	Lorsqu'elle est vraie, cette sortie signale qu'une erreur s'est produite et que la sortie STATUS est valide.
STATUS	OUT	Word	La valeur STATUS de la demande indique le résultat du cycle ou de l'initialisation. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

Typiquement, il n'y a qu'une instruction USS_Port_Scan par port de communication point à point dans le programme, et chaque appel de ce bloc fonctionnel (FB) gère une transmission vers ou depuis un entraînement unique. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données d'instance.

Votre programme doit exécuter la fonction USS_Port_Scan suffisamment souvent pour empêcher les dépassements de délai des entraînements. USS_Port_Scan est généralement appelé dans un OB d'alarme cyclique pour éviter les dépassements de délai des entraînements et mettre les mises à jour les plus récentes des données USS à disposition des appels USS_Drive_Control.

Remarque

Lors de l'utilisation de la bibliothèque de protocole USS et de l'instruction USS_Port_Scan avec un CB 1241, vous devez mettre la variable de bloc de données LINE_PRE à 0 (pas d'état initial). Si la valeur par défaut "2" est utilisée pour la variable de bloc de données LINE_PRE, le code d'erreur 16#81AB est renvoyé par l'instruction USS_Port_Scan. La variable de bloc de données LINE_PRE se trouve dans le bloc de données associé à l'instruction USS_Port_Scan (habituellement sous le nom de USS_Port_Scan_DB).

Vérifiez que la valeur de départ de LINE_PRE est bien changée en 0 (zéro).

13.4.3.2 USS_Drive_Control (Échanger des données avec l'entraînement)

Tableau 13- 44 Instruction USS_Drive_Control

CONT/LOG	SCL	Description
	<pre>"USS_Drive_Control_DB" (RUN:= _bool_in_, OFF2:= _bool_in_, OFF3:= _bool_in_, F_ACK:= _bool_in_, DIR:= _bool_in_, DRIVE:= _usint_in_, PZD_LEN:= _usint_in_, SPEED_SP:= _real_in_, CTRL3:= _word_in_, CTRL4:= _word_in_, CTRL5:= _word_in_, CTRL6:= _word_in_, CTRL7:= _word_in_, CTRL8:= _word_in_, NDR=> _bool_out_, ERROR=> _bool_out_, STATUS=> _word_out_, RUN_EN=> _bool_out_, D_DIR=> _bool_out_, INHIBIT=> _bool_out_, FAULT=> _bool_out_, SPEED=> _real_out_, STATUS1=> _word_out_, STATUS3=> _word_out_, STATUS4=> _word_out_, STATUS5=> _word_out_, STATUS6=> _word_out_, STATUS7=> _word_out_, STATUS8=> _word_out_);</pre>	<p>L'instruction USS_Drive_Control échange des données avec un entraînement en créant des messages de demande et en interprétant les messages de réponse de l'entraînement. Il faut utiliser un bloc fonctionnel distinct pour chaque entraînement, mais toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données d'instance. Vous devez créer le nom du DB lorsque vous insérez la première instruction USS_Drive_Control ; vous faites ensuite appel au DB créé par la première utilisation de l'instruction.</p> <p>STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.</p>

¹ CONT et LOG : Affichez le détail de la boîte pour visualiser tous les paramètres en cliquant au bas de la boîte. Les connecteurs de paramètre qui sont grisés sont facultatifs ; l'affectation de paramètre n'y est pas exigée.

Tableau 13- 45 Types de données pour les paramètres

Paramètre et type		Type de données	Description
RUN	IN	Bool	Bit de démarrage d'entraînement. Lorsqu'elle est vraie, cette entrée permet le fonctionnement de l'entraînement à la vitesse prédéfinie. Lorsque RUN passe à FALSE pendant le fonctionnement d'un entraînement, le moteur décélère jusqu'à l'arrêt complet. Ce comportement diffère de l'arrêt électrique (OFF2) ou du freinage du moteur (OFF3).
OFF2	IN	Bool	Bit d'arrêt électrique. Lorsqu'il est faux, ce bit permet à l'entraînement de tourner en roue libre jusqu'à l'arrêt sans freinage.
OFF3	IN	Bool	Bit d'arrêt rapide. Lorsqu'il est faux, ce bit provoque un arrêt rapide par freinage de l'entraînement plutôt que par passage en roue libre jusqu'à l'arrêt.
F_ACK	IN	Bool	Bit d'acquiescement de défaut. Ce bit est mis à 1 pour réinitialiser le bit de défaut d'un entraînement. Le bit est mis à 1 après que le défaut a été corrigé afin de signaler à l'entraînement qu'il n'a plus besoin de signaler le défaut précédent.
DIR	IN	Bool	Commande du sens de l'entraînement. Ce bit est mis à 1 pour indiquer que le sens est la marche avant (pour SPEED_SP positif).
DRIVE	IN	USInt	Adresse de l'entraînement. Cette entrée est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PZD_LEN	IN	USInt	Longueur en mots. Il s'agit du nombre de mots des données PZD. Les valeurs valides sont 2, 4, 6 ou 8 mots. La valeur par défaut est 2.
SPEED_SP	IN	Real	Consigne de vitesse. Il s'agit de la vitesse de l'entraînement sous forme de pourcentage de la fréquence configurée. Une valeur positive correspond à une marche avant (lorsque DIR est vrai). La plage valide va de 200,00 à -200,00.
CTRL3	IN	Word	Mot de commande 3. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL4	IN	Word	Mot de commande 4. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL5	IN	Word	Mot de commande 5. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL6	IN	Word	Mot de commande 6. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL7	IN	Word	Mot de commande 7. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL8	IN	Word	Mot de commande 8. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
NDR	OUT	Bool	Nouvelles données prêtes (New data ready) : Lorsqu'il est vrai, ce bit signale que les sorties contiennent des données pour une nouvelle demande de communication.

13.4 Communication USS (interface série universelle)

Paramètre et type		Type de données	Description
ERROR	OUT	Bool	Une erreur s'est produite. Lorsqu'elle est vraie, cette sortie signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_Drive_Control.
STATUS	OUT	Word	La valeur STATUS de la demande indique le résultat du cycle. Il ne s'agit pas d'un mot d'état provenant de l'entraînement.
RUN_EN	OUT	Bool	Marche activée. Ce bit signale si l'entraînement fonctionne ou non.
D_DIR	OUT	Bool	Sens de l'entraînement. Ce bit signale si l'entraînement fonctionne en marche avant ou non.
INHIBIT	OUT	Bool	Entraînement inhibé. Ce bit indique l'état du bit d'inhibition dans l'entraînement.
FAULT	OUT	Bool	Défaut de l'entraînement. Ce bit signale que l'entraînement a enregistré un défaut. Vous devez résoudre le problème puis mettre le bit F_ACK à 1 pour effacer ce bit lorsqu'il a été mis à 1.
SPEED	OUT	Real	Vitesse en cours de l'entraînement (valeur mise à l'échelle du mot d'état d'entraînement 2). Il s'agit de la valeur de la vitesse de l'entraînement sous forme de pourcentage de la vitesse configurée.
STATUS1	OUT	Word	Mot d'état 1 de l'entraînement. Cette valeur contient des bits d'état fixes d'un entraînement.
STATUS3	OUT	Word	Mot d'état 3 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS4	OUT	Word	Mot d'état 4 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS5	OUT	Word	Mot d'état 5 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS6	OUT	Word	Mot d'état 6 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS7	OUT	Word	Mot d'état 7 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS8	OUT	Word	Mot d'état 8 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.

Lors de l'exécution de la première instruction USS_Drive_Control, l'entraînement indiqué par l'adresse USS (paramètre DRIVE) est initialisé dans le DB d'instance. Après cette initialisation, les exécutions suivantes de USS_Port_Scan peuvent commencer la communication avec l'entraînement à ce numéro d'entraînement.

La modification du numéro d'entraînement nécessite un passage de la CPU de ARRET à MARCHE afin d'initialiser le DB d'instance. Les paramètres d'entrée sont écrits dans la mémoire tampon USS de message à émettre et les sorties sont lues d'une mémoire tampon de réponse valide "précédente" s'il en existe une. Il n'y a pas de transmission de données pendant l'exécution de USS_Drive_Control. Les entraînements communiquent lorsque USS_Port_Scan est exécuté. USS_Drive_Control ne fait que configurer les messages à envoyer et interpréter les données éventuelles reçues d'une précédente demande.

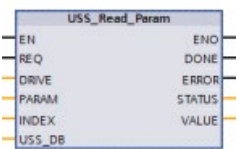
Vous pouvez commander le sens de rotation de l'entraînement via l'entrée DIR (Bool) ou via le signe (positif ou négatif) de l'entrée SPEED_SP (Real). Le tableau suivant montre comment ces entrées déterminent ensemble le sens de l'entraînement, dans l'hypothèse où le moteur est câblé pour une rotation avant.

Tableau 13- 46 Interaction entre les paramètres SPEED_SP et DIR

SPEED_SP	DIR	Sens de rotation de l'entraînement
Valeur > 0	0	Arrière
Valeur > 0	1	Avant
Valeur < 0	0	Direct
Valeur < 0	1	Arrière

13.4.3.3 USS_Read_Param (Lire des paramètres de l'entraînement)

Tableau 13- 47 Instruction USS_Read_Param

CONT/LOG	SCL	Description
	<pre>USS_Read_Param(REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, INDEX:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, VALUE=>_variant_out_, USS_DB:=_fbtref_inout);</pre>	<p>L'instruction USS_Read_Param lit un paramètre provenant de l'entraînement. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données. Il faut appeler USS_Read_Param dans un OB de cycle de programme principal.</p>

13.4 Communication USS (interface série universelle)

Tableau 13- 48 Types de données pour les paramètres

Type de paramètre		Type de données	Description
REQ	IN	Bool	Envoi de la demande. Lorsqu'elle est vraie, l'entrée REQ indique qu'une nouvelle demande de lecture est désirée. Il n'en est pas tenu compte si la demande pour ce paramètre est déjà en attente.
DRIVE	IN	USInt	Adresse de l'entraînement. DRIVE est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PARAM	IN	UInt	Numéro du paramètre. PARAM désigne le paramètre d'entraînement à écrire. La plage de ce paramètre est comprise entre 0 et 2047. Sur certains entraînements, l'octet de poids fort peut accéder à des valeurs PARAM supérieures à 2047. Vous trouverez plus de détails sur la manière d'accéder à une plage étendue dans le manuel de l'entraînement.
INDEX	IN	UInt	Indice du paramètre. INDEX désigne l'indice du paramètre d'entraînement à écrire. Valeur de 16 bits dans laquelle l'octet de poids faible est la valeur d'indice réelle avec une plage de 0 à 255. L'octet de poids fort peut être utilisé par l'entraînement et lui est spécifique. Consultez le manuel de l'entraînement pour plus de détails.
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_Drive_Control dans votre programme.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDIInt, Real	Il s'agit de la valeur du paramètre qui a été lue et qui est uniquement valide lorsque le bit DONE est vrai.
DONE ¹	OUT	Bool	S'il est vrai, ce bit signale que la sortie VALUE contient la valeur du paramètre lue suite à la demande précédente. Ce bit est mis à 1 lorsque USS_Drive_Control voit les données de réponse de lecture provenant de l'entraînement. Ce bit est mis à 0 lorsque vous demandez des données de réponse à l'aide d'une autre interrogation USS_Read_Param ou lors du deuxième des deux appels suivants de USS_Drive_Control.
ERROR	OUT	Bool	Une erreur s'est produite. Lorsqu'elle est vraie, la sortie ERROR signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_Port_Scan.
STATUS	OUT	Word	STATUS indique le résultat de la demande de lecture. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

¹ Le bit DONE indique que des données valides ont été lues dans l'entraînement de moteur référencé et transmises à la CPU. Il ne signifie pas que la bibliothèque USS est capable de lire immédiatement un autre paramètre. Une demande PKW vide doit être envoyée à l'entraînement et doit également être acquittée par l'instruction avant que la voie de paramètre pour l'entraînement spécifique ne soit mise à disposition. L'appel immédiat d'une FC USS_Read_Param ou USS_Write_Param pour l'entraînement spécifié provoque une erreur "0x818A".

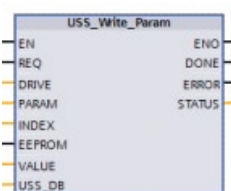
13.4.3.4 USS_Write_Param (Modifier des paramètres dans l'entraînement)

Remarque

Opérations d'écriture en EEPROM (pour l'EEPROM à l'intérieur d'un entraînement USS)

N'utilisez pas la fonction d'écriture permanente en EEPROM de manière abusive. Réduisez au minimum le nombre d'opérations d'écriture en EEPROM pour allonger la durée de vie de cette dernière.

Tableau 13- 49 Instruction USS_Write_Param

CONT/LOG	SCL	Description
	<pre>USS_Write_Param(REQ:=_bool_in_ _, DRIVE:=_usint_in_, PARAM:=_uint_in_, INDEX:=_uint_in_, EEPROM:=_bool_in_, VALUE:=_variant_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, USS_DB:=_fbtref_inout_);</pre>	<p>L'instruction USS_Write_Param modifie un paramètre dans l'entraînement. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données.</p> <p>Il faut appeler USS_Write_Param dans un OB de cycle de programme principal.</p>

13.4 Communication USS (interface série universelle)

Tableau 13- 50 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Envoi de la demande. Lorsqu'elle est vraie, l'entrée REQ indique qu'une nouvelle demande d'écriture est désirée. Il n'en est pas tenu compte si la demande pour ce paramètre est déjà en attente.
DRIVE	IN	USInt	Adresse de l'entraînement. DRIVE est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PARAM	IN	UInt	Numéro du paramètre. PARAM désigne le paramètre d'entraînement à écrire. La plage de ce paramètre est comprise entre 0 et 2047. Sur certains entraînements, l'octet de poids fort peut accéder à des valeurs PARAM supérieures à 2047. Vous trouverez plus de détails sur la manière d'accéder à une plage étendue dans le manuel de l'entraînement.
INDEX	IN	UInt	Indice du paramètre. INDEX désigne l'indice du paramètre d'entraînement à écrire. Valeur de 16 bits dans laquelle l'octet de poids faible est la valeur d'indice réelle avec une plage de 0 à 255. L'octet de poids fort peut être utilisé par l'entraînement et lui est spécifique. Consultez le manuel de l'entraînement pour plus de détails.
EEPROM	IN	Bool	Stockage dans l'EEPROM de l'entraînement. Lorsque cette entrée est vraie, une transaction d'écriture dans le paramètre de l'entraînement est stockée dans l'EEPROM de l'entraînement. Lorsqu'elle est fausse, l'écriture est temporaire et ne sera pas conservée en cas de mise hors tension puis sous tension de l'entraînement.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDIInt, Real	Valeur du paramètre qui doit être écrite. Elle doit être valide lors de la transition de REQ.
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_Drive_Control dans votre programme.
DONE ¹	OUT	Bool	S'il est vrai, le bit DONE signale que l'entrée VALUE a été écrite dans l'entraînement. Ce bit est mis à 1 lorsque USS_Drive_Control voit les données de réponse d'écriture provenant de l'entraînement. Ce bit est mis à 0 lorsque vous demandez les données de réponse à l'aide d'une autre interrogation USS_Drive_Control ou lors du deuxième des deux appels suivants de USS_Drive_Control.
ERROR	OUT	Bool	Lorsqu'elle est vraie, la sortie ERROR signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_Port_Scan.
STATUS	OUT	Word	STATUS indique le résultat de la demande d'écriture. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

¹ Le bit DONE indique que des données valides ont été lues dans l'entraînement de moteur référencé et transmises à la CPU. Il ne signifie pas que la bibliothèque USS est capable de lire immédiatement un autre paramètre. Une demande PKW vide doit être envoyée à l'entraînement et doit également être acquittée par l'instruction avant que la voie de paramètre pour l'entraînement spécifique ne soit mise à disposition. L'appel immédiat d'une FC USS_Read_Param ou USS_Write_Param pour l'entraînement spécifié provoque une erreur "0x818A".

13.4.4 Codes d'état USS

Les codes d'état des instructions USS sont renvoyés dans la sortie STATUS des fonctions USS.

Tableau 13- 51 Codes STATUS ¹

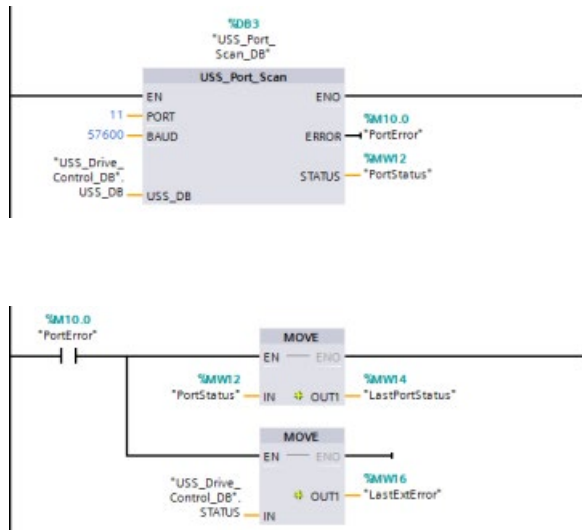
STATUS (W#16#...)	Description
0000	Pas d'erreur
8180	La longueur de la réponse de l'entraînement ne correspondait pas aux caractères reçus de l'entraînement. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8181	Le paramètre VALUE n'avait pas le type de données Word, Real ou DWord.
8182	L'utilisateur a fourni un mot pour une valeur de paramètre et a reçu un double mot ou un réel dans la réponse provenant de l'entraînement.
8183	L'utilisateur a fourni un double mot ou un réel pour une valeur de paramètre et a reçu un mot dans la réponse provenant de l'entraînement.
8184	Le télégramme de réponse de l'entraînement contenait un total de contrôle erroné. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8185	Adresse d'entraînement illicite (plage d'adresses d'entraînement valide : 1 à 16)
8186	La consigne de vitesse est hors de la plage valide (plage de consignes de vitesse valide : -200% à 200%).
8187	Un entraînement de numéro erroné a répondu à la demande envoyée. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8188	Longueur de mot PZD illicite (plage valide = 2, 4, 6 ou 8 mots)
8189	Vitesse de transmission illicite
818A	La voie de demande de paramètre est utilisée par une autre demande pour cet entraînement.
818B	L'entraînement n'a pas répondu aux demandes et nouvelles tentatives. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
818C	L'entraînement a renvoyé une erreur étendue à une opération de demande de paramètre. Consultez la description des erreurs étendues après ce tableau.
818D	L'entraînement a renvoyé une erreur d'accès illicite à une opération de demande de paramètre. Consultez le manuel de l'entraînement pour savoir pourquoi l'accès au paramètre est éventuellement limité.
818E	L'entraînement n'a pas été initialisé. Ce code d'erreur est renvoyé à USS_Read_Param ou USS_Write_Param lorsque USS_Drive_Control pour cet entraînement n'a pas été appelé au moins une fois. Cela empêche l'initialisation au premier cycle de USS_Drive_Control d'écraser une demande de lecture ou d'écriture de paramètre en attente puisque l'entraînement est initialisé en tant que nouvelle entrée. Pour corriger cette erreur, appelez USS_Drive_Control pour ce numéro d'entraînement.
80Ax-80Fx	Erreurs spécifiques renvoyées par les FB de communication point à point appelés par la bibliothèque USS. Ces valeurs de codes d'erreur ne sont pas modifiées par la bibliothèque USS et sont définies dans les descriptions des instructions point à point.

¹ En plus des erreurs d'instructions USS présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point (Page 1155) sous-jacentes.

Des informations supplémentaires sont fournies dans la variable "USS_Extended_Error" du DB d'instance de USS_DRV pour certains codes STATUS. Pour les codes hexadécimaux STATUS 8180, 8184, 8187 et 818B, USS_Extended_Error contient le numéro de l'entraînement où l'erreur de communication s'est produite. Pour le code hexadécimal STATUS 818C, USS_Extended_Error contient un code d'erreur d'entraînement provenant de l'entraînement lié à l'utilisation d'une instruction USS_Read_Param ou USS_Write_Param.

Exemple : Compte rendu d'erreurs de communication

Les erreurs de communication (STATUS = 16#818B) sont signalées uniquement dans l'instruction USS_Port_Scan et non dans l'instruction USS_Drive_Control. Par exemple, si le réseau n'est pas muni de résistances de terminaison correctes, un entraînement peut passer en MARCHE mais l'instruction USS_Drive_Control montrera uniquement des "0" pour les paramètres de sortie. Dans ce cas, vous pouvez uniquement détecter l'erreur de communication dans l'instruction USS_Port_Scan. Comme cette erreur n'est visible que pendant un cycle, vous devez ajouter une logique de capture comme illustré dans l'exemple suivant. Dans cet exemple, lorsque le bit d'erreur de l'instruction USS_Port_Scan est à VRAI, les valeurs de STATUS et de USS_Extended_Error sont sauvegardées en mémoire M. Le numéro de l'entraînement est placé dans la variable USS_Extended_Error lorsque la valeur hexadécimale du code STATUS est 8180, 8184, 8187 ou 818B.



Réseau 1 Les valeurs de l'état du port "PortStatus" et du code d'erreur étendue "USS_Drive_Control_DB".USS_Extended_Error ne sont valables que pendant un cycle du programme. Il faut les capturer en vue d'un traitement ultérieur.

Réseau 2 Le contact "PortError" déclenche la mémorisation de la valeur "PortStatus" dans "LastPortStatus" et de la valeur "USS_Drive_Control_DB".USS_Extended_Error dans "LastExtError".

Accès en lecture et écriture aux paramètres internes d'un entraînement

Les entraînements USS acceptent l'accès en lecture et écriture aux paramètres internes d'un entraînement. Cette caractéristique permet la commande et la configuration à distance de l'entraînement. Les opérations d'accès aux paramètres d'entraînement peuvent échouer suite à des erreurs telles que des valeurs hors plage ou des demandes illicites pour le mode en cours d'un entraînement. L'entraînement génère un code d'erreur qui est renvoyé dans la variable "USS_Extended_Error". Ce code d'erreur ne vaut que pour la dernière exécution d'une instruction USS_Read_Param ou USS_Write_Param. Le code d'erreur d'entraînement est placé dans la variable "USS_Extended_Error" lorsque la valeur de STATUS code est 818C hexadécimal. La valeur du code d'erreur de USS_Extended_Error dépend du modèle d'entraînement. Le manuel de l'entraînement contient une description des codes des erreurs étendues pour les opérations de lecture et d'écriture de paramètres.

13.4.5 Exigences générales pour la configuration d'un entraînement USS

Les exigences générales pour la configuration d'un entraînement USS comprennent les points suivants :

- Les entraînements doivent être paramétrés pour utiliser 4 mots PKW.
- Les entraînements peuvent être configurés pour 2, 4, 6 ou 8 mots PZD.
- Le nombre de mots PZD dans l'entraînement doit correspondre à l'entrée PZD_LEN dans l'instruction USS_Drive_Control de cet entraînement.
- La vitesse de transmission dans tous les entraînements doit coïncider avec l'entrée BAUD dans l'instruction USS_Port_Scan.
- La commande à distance doit être activée pour l'entraînement.
- La consigne de fréquence à USS sur la liaison COM doit être sélectionnée pour l'entraînement.
- L'adresse de l'entraînement doit être définie entre 1 et 16 et correspondre à l'entrée DRIVE dans le bloc USS_Drive_Control pour cet entraînement.
- La commande du sens de l'entraînement doit utiliser la polarité de la consigne de l'entraînement.
- Le réseau RS485 doit être muni de résistances de terminaison appropriées.

13.4.6 Exemple : Configuration et connexion générales à un entraînement USS

Connexion d'un entraînement MicroMaster

Ces informations sur les entraînements MicroMaster SIEMENS sont données à titre d'exemple. Pour les autres entraînements, consultez les instructions de configuration dans le manuel de l'entraînement.

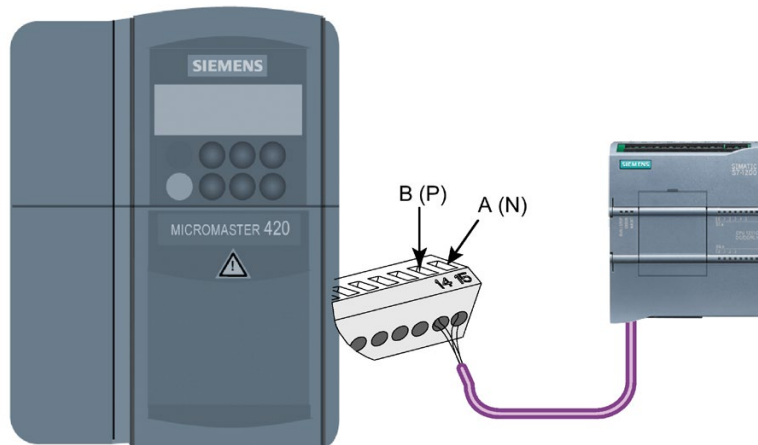
Pour effectuer la connexion à un entraînement MicroMaster série 4 (MM4), insérez les extrémités du câble RS485 dans les deux bornes sans vis à bride gainée fournies pour l'exploitation USS. Vous pouvez utiliser un câble et des connecteurs PROFIBUS standard pour connecter le S7-1200.

! PRUDENCE

Le fait d'interconnecter des équipements ayant des potentiels de référence différents peut provoquer des flux de courant indésirables via le câble de liaison.

Ces courants indésirables peuvent entraîner des erreurs de programmation ou endommager l'équipement. Assurez-vous que tous les matériels que vous allez connecter avec un câble de communication partagent un même potentiel de référence ou qu'ils sont isolés afin d'empêcher les flux de courant indésirables. Le blindage doit être relié à la terre du châssis ou à la broche 1 du connecteur à 9 broches. Nous vous recommandons de relier la borne de câblage 2-0 V de l'entraînement MicroMaster à la terre du châssis.

Il faut insérer les deux fils à l'extrémité opposée du câble RS485 dans les borniers de l'entraînement MM4. Pour brancher le câble sur un entraînement MM4, vous devez enlever les volets de l'entraînement afin d'accéder aux borniers. Vous trouverez dans le guide de l'utilisateur de l'entraînement MM4 plus d'informations sur la façon d'ôter les volets de protection de votre entraînement spécifique.



Les connexions des borniers sont désignées numériquement. Avec un connecteur PROFIBUS côté S7-1200, connectez la borne A du câble à la borne 15 d'un entraînement MM420 ou à la borne 30 d'un entraînement MM440. Connectez la borne B du connecteur de câble à la borne 14 de l'entraînement MM420 ou à la borne 29 de l'entraînement MM440.

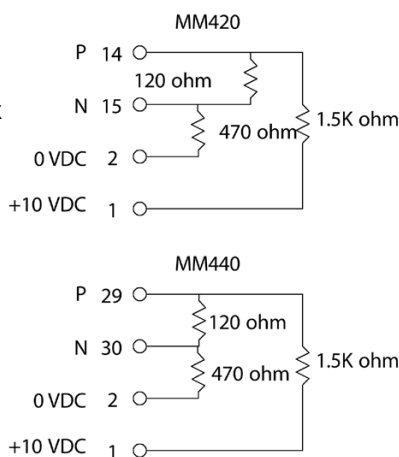
Si le S7-1200 constitue un nœud de terminaison dans le réseau ou s'il s'agit d'une liaison point à point, il faut utiliser les bornes A1 et B1 – et non les bornes A2 et B2 – du connecteur, puisqu'elles permettent de régler les paramétrages de terminaison (par exemple, avec le type de connecteur DP 6ES7972-0BA40-0X40).

! PRUDENCE

Remplacez correctement les volets de protection des entraînements avant la mise sous tension

Veillez à replacer correctement les volets de protection des entraînements avant de mettre l'unité sous tension.

Si l'entraînement est configuré comme le nœud de terminaison dans le réseau, des résistances de terminaison et de polarisation doivent également être raccordées aux bornes appropriées. Ce schéma montre des exemples de raccordements d'entraînement MM4 nécessaires pour la terminaison et la polarisation.



Configuration de l'entraînement MicroMaster série 4

Avant de connecter un entraînement au S7-1200, assurez-vous que l'entraînement a les paramètres système suivants. Servez-vous du clavier de l'entraînement pour définir les paramètres.

1. Restaurez les paramètres d'usine de l'entraînement (facultatif).	P0010=30 P0970=1
Si vous sautez l'étape 1, assurez-vous que ces paramètres sont définis aux valeurs indiquées.	Longueur PZD USS = P2012 indice 0=(2, 4, 6 ou 8) Longueur PKW USS = P2013 indice 0=4
2. Validez l'accès en lecture/écriture à tous les paramètres (mode expert).	P0003=3
3. Vérifiez les paramètres du moteur pour votre entraînement. Ces paramètres varient selon le ou les moteurs utilisés. Vous devez d'abord définir le paramètre P010 à 1 (mode de mise en marche rapide) pour pouvoir définir les paramètres P304, P305, P307, P310 et P311. Remettez le paramètre P010 à 0 une fois le paramétrage achevé. En effet, les paramètres P304, P305, P307, P310 et P311 peuvent être modifiés uniquement en mode de mise en marche rapide.	P0304=Tension nominale du moteur (V) P0305=Courant nominal du moteur (A) P0307=Puissance nominale du moteur (W) P0310=Fréquence nominale du moteur (Hz) P0311=Vitesse nominale du moteur
4. Définissez le mode de commande à local ou à distant.	P0700 indice 0=5
5. Définissez la sélection de la consigne de fréquence à USS sur la liaison COM.	P1000 indice 0=5
6. Temps d'accélération (facultatif) Il s'agit de la durée en secondes nécessaire au moteur pour accélérer jusqu'à la fréquence maximale.	P1120=(0 à 650,00)
7. Temps de décélération (facultatif) Il s'agit de la durée en secondes nécessaire au moteur pour décélérer jusqu'à l'arrêt complet.	P1121=(0 à 650,00)
8. Définissez la fréquence de référence de la liaison série.	P2000=(1 à 650 Hz)
9. Définissez la normalisation USS.	P2009 indice 0=0
10. Définissez le débit de l'interface série RS485 :	P2010 indice 0= 4 (2400 bauds) 5 (4800 bauds) 6 (9600 bauds) 7 (19200 bauds) 8 (38400 bauds) 9 (57600 bauds) 12 (115200 bauds)
11. Entrez l'adresse de l'esclave. Chaque entraînement (31 au maximum) peut être commandé par l'intermédiaire du bus.	P2011 indice 0=(0 à 31)
12. Définissez le délai d'attente de la liaison série. Il s'agit de la durée maximale autorisée entre deux télégrammes de données entrants. Cette fonction sert à désactiver l'inverseur en cas d'échec de communication. Le décompte commence une fois un télégramme de données valable reçu. Si un autre télégramme de données n'est pas reçu pendant l'intervalle de temps indiqué, l'inverseur se déclenche et affiche le code d'erreur F0070. Mettre la valeur à zéro désactive la commande.	P2014 indice 0=(0 à 65 535 ms) 0=délai d'attente désactivé
13. Transférez les données de la mémoire vive en EEPROM.	P0971=1 (démarrer le transfert) : enregistrer en EEPROM les modifications apportées aux valeurs des paramètres.

13.5 Communication Modbus

13.5.1 Vue d'ensemble de la communication Modbus RTU et Modbus TCP

Codes de fonction Modbus

- Une CPU fonctionnant en tant que maître Modbus RTU (ou client Modbus TCP) peut lire/écrire à la fois des données et des états d'E/S dans un esclave Modbus RTU distant (ou un serveur Modbus TCP). Les données distantes peuvent être lues et traitées dans la logique de votre programme.
- Une CPU fonctionnant en tant qu'esclave Modbus RTU (ou serveur Modbus TCP) permet à un appareil superviseur de lire/écrire à la fois des données et des états d'E/S dans la mémoire CPU. Un maître RTU (ou client Modbus TCP) peut écrire de nouvelles valeurs dans la mémoire CPU esclave/serveur disponible pour votre logique de programme.


 ATTENTION
Si un pirate parvient à accéder physiquement aux réseaux, il peut lire et écrire les données.
TIA Portal, la CPU et les IHM (à l'exception de celles utilisant GET/PUT) utilisent une communication sécurisée qui protège contre les attaques Replay et par l'homme du milieu. Une fois que la communication est activée, l'échange des messages signés s'effectue en texte clair, ce qui permet au pirate de lire les données, mais qui protège contre l'accès aux données en écriture. C'est le TIA Portal et non le processus de communication qui crypte les données des blocs avec protection Know-how.
Toutes les autres formes de communication (échange I/O via PROFIBUS, PROFINET, AS-i, ou autre bus I/O, GET/PUT, blocs T et modules de communication (CM)) n'ont pas de fonction de sécurité. Vous devez les protéger en limitant leur accès physique. Si un pirate parvient à accéder physiquement aux réseaux par le biais de ces modes de communication, il peut lire et écrire les données.
Pour obtenir des informations et des recommandations de sécurité, veuillez vous reporter à nos "Directives d'exploitation pour la sécurité industrielle" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf).

Tableau 13- 52 Fonctions de lecture de données : Lecture d'E/S distantes et de données de programme

Code de fonction Modbus	Fonctions de lecture dans l'esclave (serveur) : adressage standard
01	Lecture de bits de sortie : 1 à 2000 bits par demande
02	Lecture de bits d'entrée : 1 à 2000 bits par demande
03	Lecture de registres de maintien : 1 à 125 mots par demande
04	Lecture de mots d'entrée : 1 à 125 mots par demande

13.5 Communication Modbus

Tableau 13- 53 Fonctions d'écriture de données : Ecriture d'E/S distantes et modification de données de programme

Code de fonction Modbus	Fonctions d'écriture dans l'esclave (serveur) : adressage standard
05	Ecriture d'un bit de sortie : 1 bit par demande
06	Ecriture d'un registre de maintien : 1 mot par demande
15	Ecriture d'un ou de plusieurs bits de sortie : 1 à 1968 bits par demande
16	Ecriture d'un ou de plusieurs registres de maintien : 1 à 123 mots par demande

- Les fonctions Modbus de codes 08 à 11 fournissent des informations de diagnostic sur la communication avec l'esclave.
- La fonction Modbus de code 0 diffuse un message à tous les esclaves (sans réponse de ces derniers). La fonction de diffusion générale n'est pas disponible pour Modbus TCP, car la communication est orientée liaison.

Tableau 13- 54 Adresses de stations de réseau Modbus

Station	Adresse	
Station RTU	Adresse de station standard	1 à 247
	Adresse de station étendue	1 à 65535
Station TCP	Adresse de station	Adresse IP et numéro de port

Adresses de mémoire Modbus

Le nombre réel d'adresses de mémoire Modbus disponibles dépend du modèle de la CPU, de la taille de la mémoire de travail et de la part de mémoire CPU utilisée par d'autres données de programme. Le tableau suivant indique la valeur nominale de la plage d'adresses.

Tableau 13- 55 Adresses de mémoire Modbus

Station	Plage d'adresses	
Station RTU	Adresse de mémoire standard	10K
	Adresse de mémoire étendue	64K
Station TCP	Adresse de mémoire standard	10K

Communication Modbus RTU

Modbus RTU (Remote Terminal Unit : unité terminale distante) est un protocole de communication de réseau standard qui utilise la liaison électrique RS232 ou RS485 pour le transfert de données série entre dispositifs réseau Modbus. Vous pouvez ajouter des ports de réseau point à point (PtP) à une CPU avec un CM RS232 ou RS485 ou un CB RS485.

Modbus RTU utilise un réseau maître/esclave où toutes les communications sont déclenchées par un maître unique et où les esclaves peuvent uniquement répondre à une demande du maître. Le maître envoie une demande à une adresse d'esclave et seule cette adresse d'esclave répond à la commande.

Communication Modbus TCP

Modbus TCP (Transmission Control Protocol : protocole de gestion de transmission) est un protocole de communication de réseau standard qui utilise le connecteur PROFINET sur la CPU pour la communication TCP/IP. Aucun module matériel de communication supplémentaire n'est nécessaire.

Modbus TCP utilise des liaisons de la communication ouverte (OUC) comme voie de communication Modbus. Plusieurs liaisons client-serveur peuvent exister en plus de la liaison entre STEP 7 et la CPU. Les liaisons mixtes client et serveur sont prises en charge jusqu'au nombre maximum de liaisons autorisé par le modèle de CPU (Page 827).

Chaque liaison MB_SERVER doit utiliser un DB d'instance et un numéro de port IP uniques. Une seule liaison est acceptée par port IP. Chaque instruction MB_SERVER (avec son DB d'instance et son port IP uniques) doit être exécutée individuellement pour chaque liaison.

Un client Modbus TCP (maître) doit gérer la liaison client-serveur à l'aide du paramètre DISCONNECT. Les actions de base Modbus client sont présentées ci-dessous.

1. Etablir une liaison vers l'adresse IP et le numéro de port IP d'un serveur particulier (esclave)
2. Déclencher l'émission client d'un message Modbus et recevoir les réponses du serveur
3. Si désiré, déclencher la déconnexion du client et du serveur pour permettre la liaison avec un serveur différent

Instructions Modbus RTU dans votre programme

- Modbus_Comm_Load: Une exécution de Modbus_Comm_Load sert à configurer les paramètres de port point à point, tels que la vitesse de transmission, la parité et le contrôle de flux. Une fois qu'un port CPU est configuré pour le protocole Modbus RTU, il peut être utilisé uniquement par les instructions Modbus_Master ou Modbus_Slave.
- Modbus_Master: L'instruction Modbus_Master permet à la CPU d'agir en tant que maître Modbus RTU et de communiquer avec un ou plusieurs esclaves Modbus.
- Modbus_Slave: L'instruction Modbus_Slave permet à la CPU d'agir en tant qu'esclave Modbus RTU et de communiquer avec un maître Modbus.

Instructions Modbus TCP dans votre programme

- MB_CLIENT: Permet d'établir une liaison TCP client-serveur, d'envoyer un message de commande, de recevoir la réponse et de gérer la déconnexion du serveur.
- MB_SERVER: Permet de se connecter à un client Modbus TCP sur demande, de recevoir un message Modbus et d'envoyer la réponse.

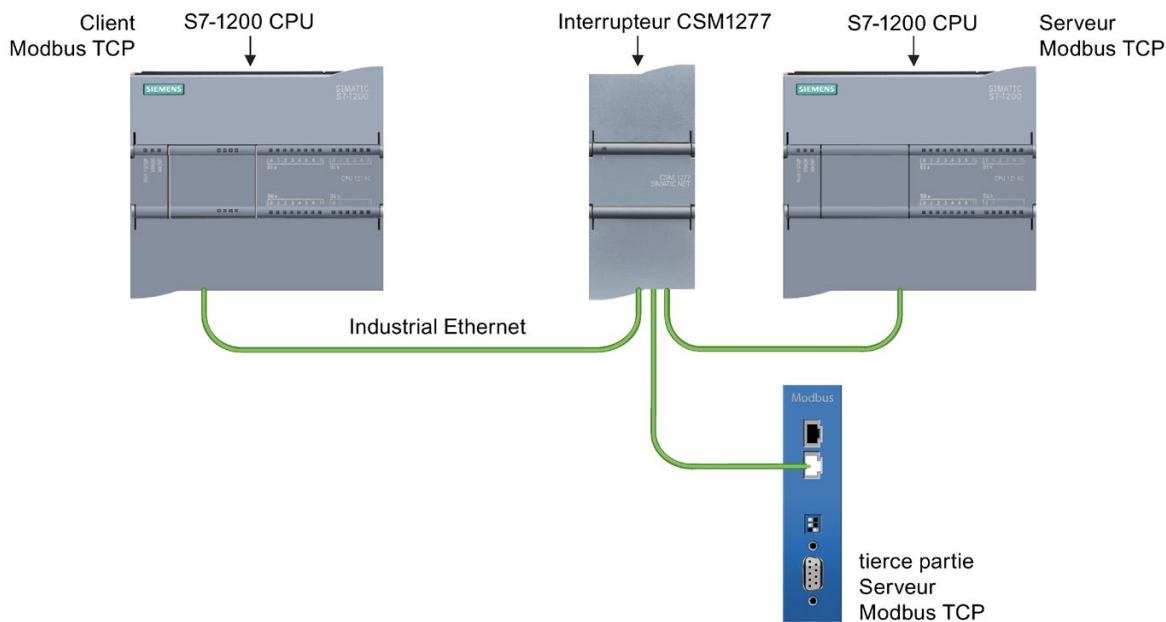
13.5.2 Modbus TCP

13.5.2.1 Vue d'ensemble

À partir de la version V4.1 de la CPU S7-1200 avec STEP 7 V13 SP1, la CPU étend les fonctionnalités de Modbus TCP à l'utilisation d'instructions T-block améliorées.

Pour cette raison, le S7-1200 accepte deux jeux d'instructions PtP :

- Instructions d'héritage Modbus TCP (Page 1317) : Ces instructions Modbus TCP existaient avec la version V4.0 du S7-1200.
- Instructions Modbus TCP (Page 1225) : Ces instructions Modbus TCP offrent toutes les fonctionnalités des instructions d'héritage.



13.5.2.2 Sélection de la version des instructions Modbus TCP

Trois versions d'instructions Modbus TCP sont disponibles dans STEP 7 :

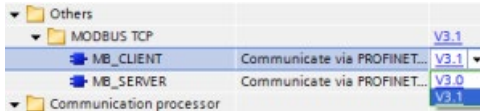
- Ancienne version 3.0 : compatible avec toutes les versions de CPU et de CP
- Ancienne version 3.1 : compatible avec toutes les versions de CPU et de CP
- Version 4.1 : compatible avec les CPU de version V4.0 et ultérieure et les CM de version V2.1 et ultérieure

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

N'utilisez pas à la fois des instructions de version 3.0 et 3.1 dans le même programme CPU. Les instructions Modbus TCP dans votre programme doivent avoir le même numéro de version principale (1.x, 2.y ou V.z). Toutefois, les instructions individuelles à l'intérieur d'un groupe de version principale peuvent avoir des versions secondaires différentes (1.x).



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.



Pour changer la version des instructions Modbus TCP, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction Modbus TCP dans votre programme, une nouvelle instance de FB est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB dans l'arborescence de projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction Modbus TCP dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB Modbus TCP dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction Modbus TCP.

13.5.2.3 Instructions Modbus TCP

Instruction MB_CLIENT (Communiquer à l'aide de PROFINET en tant que client Modbus TCP)

Tableau 13- 56 Instruction MB_CLIENT

CONT/LOG	SCL	Description
<p>"MB_CLIENT_DB"</p>	<pre>"MB_CLIENT_DB" (REQ:=_bool_in_, DISCONNECT:=_bool_in_, MB_MODE:=_usint_in_, MB_DATA_ADDR:=_udint_in_, MB_DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_DATA_PTR:=_variant_inout_, CONNECT:=_variant_inout_);</pre>	<p>MB_CLIENT permet de communiquer en tant que client Modbus TCP par l'intermédiaire du port PROFINET sur la CPU S7-1200. Aucun module matériel de communication supplémentaire n'est nécessaire.</p> <p>MB_CLIENT permet d'établir une liaison client-serveur, d'envoyer une demande de fonction Modbus, de recevoir une réponse et de gérer la déconnexion d'un serveur Modbus TCP.</p>

Tableau 13- 57 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	In	Bool	FALSE = Aucune demande de communication Modbus TRUE = Demande de communication avec un serveur Modbus TCP
DISCONNECT	IN	Bool	Le paramètre DISCONNECT permet à votre programme de gérer la connexion et la déconnexion à un serveur Modbus. Si DISCONNECT est égal à 0 et qu'il n'existe pas de liaison, MB_CLIENT tente d'établir une liaison à l'adresse et au numéro de port IP affectés. Si DISCONNECT est égal à 1 et qu'une liaison existe, une opération de déconnexion est tentée. Aucune autre opération n'est tentée lorsque cette entrée est activée.
MB_MODE	IN	USInt	Sélection du mode : Définit le type de demande (lecture, écriture ou diagnostic). Voir le tableau des fonctions Modbus ci-après pour plus de détails.
MB_DATA_ADDR	IN	UDInt	Adresse de début Modbus : Définit l'adresse de début des données auxquelles on accédera via MB_CLIENT. Voir le tableau des fonctions Modbus ci-après pour les adresses valides.
MB_DATA_LEN	IN	UInt	Longueur des données Modbus : Définit le nombre de bits ou mots auxquels accéder dans cette demande. Voir le tableau des fonctions Modbus ci-après pour les longueurs valides.
MB_DATA_PTR	IN_OUT	VARIANT	Pointeur désignant le registre de données Modbus : Le registre stocke temporairement les données allant vers un serveur Modbus ou venant d'un serveur Modbus. Le pointeur doit désigner un DB global non optimisé ou une adresse en mémoire M.
CONNECT	IN_OUT	VARIANT	Référence à la structure d'un DB qui contient des paramètres de liaison de type de données système "TCON_IP_v4".
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • 0 : Pas d'opération MB_CLIENT en cours • 1 : Opération MB_CLIENT en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque l'exécution de MB_CLIENT s'est achevée avec une erreur. Le code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

Remarque

Version de firmware de la CPU requise

Les instructions Modbus TCP décrites dans ce paragraphe du manuel nécessitent une version de firmware V4.1 ou ultérieure.

Paramètre REQ

FALSE = Aucune demande de communication Modbus

TRUE = Demande de communication avec un serveur Modbus TCP

Si aucune instance de MB_CLIENT n'est active et que le paramètre DISCONNECT est égal à 0, une nouvelle demande Modbus est lancée si REQ est égal à 1. Si la liaison n'est pas déjà établie, une nouvelle liaison le sera.

Si la même instance de MB_CLIENT est à nouveau exécutée avec DISCONNECT=0 et REQ=1, et ce avant l'achèvement de la demande en cours, il n'y aura pas d'émission Modbus suivante. Toutefois, dès que la demande en cours aura été achevée, une nouvelle demande pourra être traitée si MB_CLIENT est exécuté avec l'entrée REQ à 1.

Lorsque la demande de communication MB_CLIENT en cours est achevée, le bit DONE est TRUE pour un cycle. Le bit DONE peut être utilisé comme porte temporelle pour ordonner plusieurs demandes MB_CLIENT.

Remarque

Cohérence des données d'entrée pendant le traitement de MB_CLIENT

Lorsqu'un client Modbus a déclenché une opération Modbus, l'état de toutes les entrées est sauvegardé en interne, puis comparé à chaque appel successif. Cette comparaison sert à déterminer si cet appel particulier était à l'origine de la demande client active. Plusieurs appels MB_CLIENT peuvent être effectués à l'aide d'un DB d'instance commun.

Il est important que les entrées ne soient pas modifiées tant qu'une opération MB_CLIENT est traitée activement. Si cette règle n'est pas respectée, un appel MB_CLIENT ne peut pas déterminer l'instance active.

Sélection de la fonction de communication Modbus via les paramètres MB_MODE et MB_DATA_ADDR

L'instruction MB_CLIENT utilise une entrée MB_MODE et non un code de fonction. MB_DATA_ADDR définit l'adresse Modbus de début des données distantes.

La combinaison de MB_MODE et MB_DATA_ADDR détermine le code de fonction qui est utilisé dans le message Modbus réel. Le tableau suivant montre la correspondance entre le paramètre MB_MODE, MB_DATA_ADDR et la fonction Modbus :

13.5 Communication Modbus

Tableau 13- 58 Fonctions Modbus

MB_MODE	Fonction Modbus	Longueur des données	Opération et données	MB_DATA_ADDR
0	01	1 à 2000	Lecture de bits de sortie : 1 à 2000 bits par demande	1 à 9999
101	01	1 à 2000	Lecture de bits de sortie : 1 à 2000 bits par demande	00000 à 65535
0	02	1 à 2000	Lecture de bits d'entrée : 1 à 2000 bits par demande	10001 à 19999
102	02	1 à 2000	Lecture de bits d'entrée : 1 à 2000 bits par demande	00000 à 65535
0	03	1 à 125	Lecture de registres de maintien : 1 à 125 mots par demande	40001 à 49999 ou 400001 à 465535
103	03	1 à 125	Lecture de registres de maintien : 1 à 125 mots par demande	00000 à 65535
0	04	1 à 125	Lecture de mots d'entrée : 1 à 125 mots par demande	30001 à 39999
104	04	1 à 125	Lecture de mots d'entrée : 1 à 125 mots par demande	00000 à 65535
1	05	1	Écriture d'un bit de sortie : Un bit par demande	1 à 9999
105	05	1	Ecriture d'un bit de sortie : Un bit par demande	00000 à 65535
1	06	1	Écriture d'un registre de maintien : 1 mot par demande	40001 à 49999 ou 400001 à 465535
106	06	1	Ecriture d'un registre de maintien : 1 mot par demande	00000 à 65535
1	15	2 à 1968	Ecriture de plusieurs bits de sortie : 2 à 1968 bits par demande	1 à 9999
1	16	2 à 123	Ecriture de plusieurs registres de maintien : 2 à 123 mots par demande	40001 à 49999 ou 400001 à 465535
2	15	1 à 1968	Écriture d'un ou de plusieurs bits de sortie : 1 à 1968 bits par demande	1 à 9999
2	15	1 à 1968	Ecriture d'un ou de plusieurs bits de sortie : 1 à 1968 bits par demande	1 à 9999
2	15	1 à 1968	Écriture d'un ou de plusieurs bits de sortie : 1 à 1968 bits par demande	1 à 9999
115	15	1 à 1968	Écriture d'un ou de plusieurs bits de sortie : 1 à 1968 bits par demande	00000 à 65535
2	16	1 à 123	Écriture d'un ou de plusieurs registres de maintien : 1 à 123 mots par demande	40001 à 49999 ou 400001 à 465535
116	16	1 à 123	Ecriture d'un ou de plusieurs registres de maintien : 1 à 123 mots par demande	00000 à 65535

MB_MODE	Fonction Modbus	Longueur des données	Opération et données	MB_DATA_ADDR
11	11	0	Lecture du mot d'état de communication serveur et du compteur d'événements. Le mot d'état indique si l'esclave est occupé (0 = non occupé, 0xFFFF = occupé). Le compteur d'événements est incrémenté à chaque fois qu'un message s'achève avec succès. Les deux paramètres MB_DATA_ADDR et MB_DATA_LEN de MB_CLIENT ne sont pas pris en compte pour cette fonction.	
80	08	1	Contrôle de l'état du serveur à l'aide du code de diagnostic 0x0000 (test de retour, le serveur renvoie la demande en écho) 1 mot par demande	
81	08	1	Réinitialisation du compteur d'événements du serveur à l'aide du code de diagnostic 0x000A 1 mot par demande	
3 à 10, 12 à 79, 82 à 100, 107 à 114, 117 à 255			Réservé	

Remarque

MB_DATA_PTR définit une mémoire tampon pour stocker les données lues/écrites dans un serveur Modbus TCP

La mémoire tampon de données peut se situer dans un DB global non optimisé ou à une adresse de mémoire M.

Si la mémoire tampon est en mémoire M, utilisez le format de pointeur Any, à savoir P#"adresse bit" "type de données" "longueur" (par exemple, P#M1000.0 WORD 500).

Définition d'une mémoire tampon de communication avec le paramètre MB_DATA_PTR

- Fonctions de communication MB_CLIENT :
 - Lecture et écriture de données d'un bit dans des adresses de serveur Modbus (00001 à 09999)
 - Lecture de données d'un bit dans des adresses de serveur Modbus (10001 à 19999)
 - Lecture de données mots de 16 bits dans des adresses de serveur Modbus (30001 à 39999) et (40001 à 49999)
 - Ecriture de données mots de 16 bits dans des adresses de serveur Modbus (40001 à 49999)
- Les données mots ou bits sont transférées vers/depuis la mémoire tampon de DB ou de mémoire M définie par MB_DATA_PTR.
- Si vous avez défini un DB comme mémoire tampon avec MB_DATA_PTR, vous devez affecter un type à tous les éléments de données du DB.
 - Le type de données Bool de 1 bit représente une adresse de bit Modbus.
 - Les types de données de 16 bits (mot unique), tels que WORD, UInt et Int, représentent une adresse de mot Modbus.
 - Les types de données de 32 bits (double mot), tels que DWORD, DInt et Real, représentent deux adresses de mot Modbus.
- MB_DATA_PTR permet d'indiquer des éléments de DB complexes, tels que :
 - Tableaux
 - structures nommées où chaque élément est unique
 - structures complexes nommées où chaque élément a un nom unique et un type de données de 16 ou 32 bits
- Il n'est pas obligatoire que les zones de données MB_DATA_PTR soient dans le même bloc de données global (ou la même zone de mémoire M). Vous pouvez indiquer un bloc de données pour les lectures Modbus, un autre bloc de données pour les écritures Modbus ou un bloc de données pour chaque MB_CLIENT.

Définition de données utilisées pour établir une liaison PROFINET avec le paramètre CONNECT

Vous devez utiliser un bloc de données global et enregistrer les données de connexion requises avant de référencer ce DB dans le paramètre CONNECT.

1. Créez un nouveau bloc de données global ou utilisez un bloc de données global existant pour enregistrer les données CONNECT. Vous pouvez utiliser un bloc de données pour enregistrer plusieurs structures de données TCON_IP_v4. Chaque liaison de type client ou serveur Modbus TCP utilise une structure de données TCON_IP_v4. Vous référencez les données de connexion au paramètre CONNECT.
2. Attribuez un nom évocateur au DB et à une variable statique. Par exemple, appelez le bloc de données "Liaisons Modbus" et une variable statique "TCPActive_1" (c.-à-d. liaison client Modbus TCP 1).
3. Dans l'éditeur de bloc de données, définissez le type de données système "TCON_IP_v4" dans la colonne Type de données, par exemple variable statique "TCPActive_1".
4. Affichez le détail de la structure TCON_IP_v4 de manière à pouvoir modifier les paramètres de liaison, comme illustré dans l'image suivante.
5. Modifiez les données dans la structure TCON_IP_v4 d'une liaison MB_CLIENT.
6. Entrez la référence à la structure de DB pour le paramètre CONNECT de MB_CLIENT. Par exemple, "Liaisons Modbus".TCPActive_1.

Modbus connections				
	Nom	Type de données	Valeur de départ	Commentaire
1	Static			
2	TCPActive_1	TCON_IP_v4		
3	Interfaceld	HW_ANY	64	HW-identifler of IE-interface submodule
4	ID	CONN_OUC	1	connection reference / identifiier
5	ConnectionType	Byte	16#0B	type of conection: 11=TCP/IP, 19=UDP (17=TC...
6	ActiveEstablished	Bool	True	active/passive connection establishment
7	RemoteAddress	IP_V4		remote IP address (IPv4)
8	ADDR	array [1..4] of Byte		IPv4 address
9	ADDR[1]	Byte	192	
10	ADDR[2]	Byte	168	
11	ADDR[3]	Byte	2	
12	ADDR[4]	Byte	241	
13	RemotePort	UInt	502	remote UDP/TCP port number
14	LocalPort	UInt	0	local UDP/TCP port number

Modifier des données du DB TCP_IP_v4 pour chaque liaison MB_CLIENT

- **Interfaceld** : Dans la fenêtre de configuration des appareils, cliquez sur l'image du port PROFINET de la CPU. Ensuite, cliquez sur l'onglet Général des propriétés et utilisez l'identifiant de matériel qui s'y trouve.
- **ID** : Entrez une ID de liaison comprise entre 1 et 4095. La communication Modbus TCP est réalisée à l'aide d'instructions TCON, TDISCON, TSEND et TRCV sous-jacentes pour OUC (communication ouverte).
- **ConnectionType** : Pour TCP/IP, sélectionnez 16#0B (nombre décimal = 11) comme type de liaison par défaut.
- **ActiveEstablished** : Cette valeur doit être mise à 1 ou TRUE. La liaison dans laquelle MB_CLIENT déclenche la communication Modbus est active.

- **RemoteAddress** : Entrez l'adresse IP du serveur Modbus TCP cible dans les quatre éléments de tableau ADDR. Par exemple, entrez 192.168.2.241, comme dans l'image précédente.
- **RemotePort** : Valeur par défaut : 502. C'est le numéro de port IP du serveur Modbus avec lequel MB_CLIENT tente d'établir la liaison et de communiquer. Certains serveurs Modbus tiers nécessitent un numéro de port différent.
- **LocalPort** : Cette valeur doit être mise à 0 pour une liaison MB_CLIENT.

Liaisons client multiples

Un client Modbus TCP peut accepter des liaisons simultanées jusqu'au nombre maximum de liaisons de communication ouverte autorisé par l'API. Le nombre total de liaisons pour un API, clients et serveurs Modbus TCP compris, ne doit pas dépasser le nombre maximum de liaisons de communication ouverte prises en charge (Page 827).

Les liaisons client individuelles simultanées doivent respecter les règles suivantes :

- Chaque liaison MB_CLIENT doit utiliser un DB d'instance unique.
- Chaque liaison MB_CLIENT doit indiquer une adresse IP de serveur unique.
- Chaque liaison MB_CLIENT doit indiquer un ID de liaison unique.
- Des numéros de port IP uniques peuvent ou non être obligatoires selon la configuration du serveur.

Un ID de liaison différent doit être utilisé pour chaque DB d'instance. En résumé, le DB d'instance et l'ID de liaison forment une paire qui doit être unique pour chaque liaison.

Tableau 13- 59 Bloc de données d'instance MB_CLIENT : Variables statiques accessibles à l'utilisateur

Variable	Type de données	Valeur par défaut	Description
Blocked_Proc_Timeout	Real	3.0	Durée (en secondes) pendant laquelle attendre une instance de client Modbus bloquée avant d'annuler l'état ACTIF de cette instance. Cela peut par exemple se produire lorsqu'une demande client a été émise et que l'application arrête d'exécuter la fonction client avant l'achèvement complet de la demande. La limite maximale du S7-1200 est de 55 secondes.
MB_Unit_ID	Word	255	Identificateur d'unité Modbus : On accède à un serveur Modbus TCP à l'aide de son adresse IP. Le paramètre MB_UNIT_ID n'est donc pas utilisé pour l'adressage Modbus TCP. Le paramètre MB_UNIT_ID correspond à l'adresse d'esclave dans le protocole Modbus RTU. Si un serveur Modbus TCP est utilisé comme passerelle vers un protocole Modbus RTU, on peut utiliser MB_UNIT_ID pour identifier l'esclave connecté sur le réseau série. On utiliserait MB_UNIT_ID pour transmettre la demande à l'adresse d'esclave Modbus RTU correcte. Certains appareils Modbus TCP peuvent exiger du paramètre MB_UNIT_ID d'être à l'intérieur d'une plage de valeurs restreinte.
RCV_TIMEOUT	Real	2.0	Durée en secondes pendant laquelle MB_CLIENT attend qu'un serveur réponde à une demande
Connected	Bool	0	Indique si la liaison au serveur affecté est établie ou coupée : 1=liaison établie, 0=liaison coupée

Tableau 13- 60 Erreurs de protocole MB_CLIENT :

STATUS (W#16#)	Code réponse envoyé au client Modbus (B#16#)	Erreurs de protocole Modbus
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou accès en dehors des limites de la zone d'adresses de MB_HOLD_REG
8384	03	Erreur de valeur de données
8385	03	Code de diagnostic de données non pris en charge (code de fonction 08)

13.5 Communication Modbus

Tableau 13- 61 Codes d'erreur d'exécution de MB_CLIENT ¹

STATUS (W#16#)	Erreurs de paramètres MB_CLIENT
7001	MB_CLIENT attend une réponse du serveur Modbus à une demande de connexion ou de déconnexion sur le port TCP affecté. Ce code n'est renvoyé qu'à la première exécution d'une opération de connexion ou de déconnexion.
7002	MB_CLIENT attend une réponse du serveur Modbus à une demande de connexion ou de déconnexion pour le port TCP affecté. Ce code est renvoyé pour les exécutions suivantes en attendant l'achèvement d'une opération de connexion ou de déconnexion.
7003	Une opération de déconnexion s'est achevée avec succès (valable pendant un seul cycle API).
80C8	Le serveur n'a pas répondu dans le temps imparti. MB_CLIENT doit recevoir une réponse avec l'ID de transaction initialement transmis dans le temps imparti ; sinon, cette erreur est renvoyée. Vérifiez la liaison au serveur Modbus. Cette erreur n'est renvoyée qu'après la réalisation de nouvelles tentatives (le cas échéant).
8188	Mode invalide
8189	Adresse de données invalide
818A	Longueur de données invalide
818B	Pointeur invalide sur la zone DATA_PTR. Il peut s'agir de la combinaison de MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Le pointeur DATA_PTR désigne une zone DB optimisée (ce doit être une zone DB non optimisée ou une zone de mémoire M).
8200	Le port est occupé à traiter une demande Modbus existante.
8380	La trame Modbus reçue est incorrecte ou trop peu d'octets ont été reçus.
8387	Le paramètre ID de liaison défini est différent de l'ID utilisé pour les demandes précédentes. Un seul ID de liaison peut être utilisé dans chaque DB d'instance MB_CLIENT. Ce code est également renvoyé comme erreur interne si l'ID de protocole Modbus TCP reçu d'un serveur est différent de 0.
8388	Un serveur Modbus a renvoyé une quantité de données différente de celle demandée. Ce code s'applique aux fonctions Modbus 15 et 16 uniquement.

¹ En plus des erreurs MB_CLIENT présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication par blocs T sous-jacentes (TCON, TDISCON, TSEND et TRCV).

Instruction MB_SERVER (Communiquer à l'aide de PROFINET en tant que serveur Modbus TCP)

Tableau 13- 62 Instruction MB_SERVER

CONT/LOG	SCL	Description
	<pre>"MB_SERVER_DB" (DISCONNECT:=_bool_in_, CONNECT:=_variant_in_, NDR=>_bool_out_, DR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	<p>MB_SERVER permet de communiquer en tant que serveur Modbus TCP par l'intermédiaire du port PROFINET sur la CPU S7-1200. Aucun module matériel de communication supplémentaire n'est nécessaire.</p> <p>MB_SERVER permet d'accepter une demande de liaison à un client Modbus TCP, de recevoir une demande de fonction Modbus et d'envoyer un message de réponse.</p>

Tableau 13- 63 Types de données pour les paramètres

Paramètre et type		Type de données	Description
DISCONNECT	IN	Bool	MB_SERVER tente d'établir une liaison "passive" à un dispositif partenaire. Cela signifie que le serveur est passivement à l'écoute d'une demande de connexion TCP provenant de toute adresse IP demandeuse. Si DISCONNECT est égal à 0 et qu'il n'existe pas de liaison, une liaison passive peut être initiée. Si DISCONNECT est égal à 1 et qu'une liaison existe, une opération de déconnexion est lancée. Ce paramètre permet à votre programme de gérer le moment où une liaison est acceptée. Aucune autre opération n'est tentée lorsque cette entrée est activée.
CONNECT	IN	Variant	Référence à la structure d'un DB qui contient des paramètres de liaison de type de données système "TCON_IP_v4".
MB_HOLD_REG	IN_OUT	Variant	Pointeur désignant le registre de maintien Modbus pour MB_SERVER. Le registre de maintien doit être soit un DB global non optimisé, soit une adresse en mémoire M. Cette zone de mémoire sert à stocker les données auxquelles un client Modbus a le droit d'accéder à l'aide des fonctions Modbus de registre 3 (lecture), 6 (écriture), 16 (écriture) et 23 (lecture/écriture).
NDR	OUT	Bool	Nouvelles données prêtes (New Data Ready) : 0 = Pas de nouvelles données, 1 = Indique que de nouvelles données ont été écrites par un client Modbus
DR	OUT	Bool	Données lues : 0 = Pas de données lues, 1 = Indique que des données ont été lues par un client Modbus
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque l'exécution de MB_SERVER s'est achevée avec une erreur. Le code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

Remarque

Version de firmware de la CPU requise

Les instructions Modbus TCP décrites dans ce paragraphe du manuel nécessitent une version de firmware V4.1 ou ultérieure.

Remarque

Utilisation de la fonction 23 avec l'instruction MB_SERVER

L'instruction MB_SERVER prend en charge l'utilisation du code de fonction 23 pour l'écriture et la lecture dans un registre de maintien en une demande unique. Toutefois, l'instruction MB_CLIENT NE prend PAS en charge cette fonction et renvoie un code d'erreur. Vous devez également noter que, si la demande contient à la fois lecture et écriture, l'instruction traite l'opération d'écriture avant l'opération de lecture.

Définition de données utilisées pour établir une liaison PROFINET avec le paramètre CONNECT

Vous devez utiliser un bloc de données global et enregistrer les données de connexion requises avant de référencer ce DB dans le paramètre CONNECT.

1. Créez un nouveau bloc de données global ou utilisez un bloc de données global existant pour enregistrer les données CONNECT. Vous pouvez utiliser un bloc de données pour enregistrer plusieurs structures de données TCON_IP_v4. Chaque liaison Modbus TCP client ou serveur utilise une structure de données TCON_IP_v4. Vous référencez les données de connexion au paramètre CONNECT.
2. Attribuez un nom évocateur au DB et à une variable statique. Par exemple, appelez le bloc de données "Liaisons Modbus" et une variable statique "TCPpassive_1" (c.-à-d. liaison serveur Modbus TCP 1).
3. Dans l'éditeur de bloc de données, définissez le type de données système "TCON_IP_v4" dans la colonne Type de données, par exemple variable statique "TCPactive_1".
4. Affichez le détail de la structure TCON_IP_v4 de manière à pouvoir modifier les paramètres de liaison, comme illustré dans l'image suivante.
5. Modifiez les données dans la structure TCON_IP_v4 d'une liaison MB_SERVER.
6. Entrez la référence à la structure de DB pour le paramètre CONNECT de MB_SERVER. Par exemple, "Liaisons Modbus".TCPpassive_1.

Modbus connections				
	Nom	Type de données	Valeur de départ	Commentaire
1	Static			
2	TCPpassive_1	TCON_IP_v4		
3	InterfaceId	HW_ANY	64	HW-identfier of IE-interface submodule
4	ID	CONN_OUC	1	connection reference / identfier
5	ConnectionType	Byte	16#0B	type of connetion: 11=TCP/IP, 19=UDP (17=TC...
6	ActiveEstablished	Bool	False	active/passive connection establishment
7	RemoteAddress	IP_v4		remote IP address (IPv4)
8	ADDR	array [1..4] of Byte		IPv4 address
9	ADDR[1]	Byte	192	
10	ADDR[2]	Byte	168	
11	ADDR[3]	Byte	2	
12	ADDR[4]	Byte	241	
13	RemotePort	UInt	0	remote UDP/TCP port number
14	LocalPort	UInt	502	local UDP/TCP port number

Modifier des données du DB TCP_IP_v4 pour chaque liaison MB_SERVER

- **InterfaceID** : Dans la fenêtre de configuration des appareils, cliquez sur l'image du port PROFINET de la CPU. Ensuite, cliquez sur l'onglet Général des propriétés et utilisez l'identifiant de matériel qui s'y trouve.
- **ID** : Entrez un nombre compris entre 1 et 4095, unique pour cette liaison. La communication Modbus TCP est réalisée à l'aide d'instructions TCON, TDISCON, TSEND et TRCV sous-jacentes pour OUC (communication ouverte). Jusqu'à huit liaisons de communication ouverte simultanées sont autorisées.
- **ConnectionType** : Pour TCP/IP, sélectionnez 16#0B (valeur décimale = 11) comme type de liaison par défaut.
- **ActiveEstablished** : Cette valeur doit être mise à 0 ou FALSE. La liaison est passive en ce que MB_SERVER est dans l'attente d'une demande de liaison provenant d'un client Modbus.
- **RemoteAddress** : Deux options coexistent :
 - Utilisez 0.0.0.0 et MB_CLIENT répondra à toute demande de liaison Modbus provenant d'un client TCP.
 - Entrez l'adresse IP d'un client Modbus TCP cible et MB_CLIENT ne répondra qu'aux demandes provenant de l'adresse IP client concernée. Par exemple, entrez 192.168.2.241, comme dans l'image précédente.
- **RemotePort** : Cette valeur doit être mise à 0 pour une liaison MB_SERVER.
- **LocalPort** : Valeur par défaut : 502. C'est le numéro de port IP du client Modbus avec lequel MB_SERVER tente d'établir la liaison et de communiquer. Certains clients Modbus tiers nécessitent un numéro de port différent.

Adresses Modbus et mémoire image

MB_SERVER permet à des fonctions Modbus entrantes de codes 1, 2, 4, 5 et 15 de lire/d'écrire des bits/mots directement dans la mémoire image des entrées/sorties. Pour les fonctions de transfert de données de codes 3, 6 et 16, le paramètre MB_HOLD_REG doit être défini avec un type de données supérieur à un octet. Le tableau suivant montre la correspondance entre les adresses Modbus et la mémoire image dans la CPU.

Tableau 13- 64 Correspondance entre adresses Modbus et mémoire image

Fonctions Modbus					S7-1200		
Codes	Fonction	Zone de données	Plage d'adresses			Zone de données	Adresse de la CPU
01	Lecture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
02	Lecture de bits	Entrée	10001	à	18192	Mémoire image des entrées	I0.0 à I1023.7
04	Lecture de mots	Entrée	30001	à	30512	Mémoire image des entrées	IW0 à IW1022
05	Ecriture d'un bit	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
15	Ecriture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7

Les fonctions de message Modbus entrantes de codes 3, 6 et 16 lisent/écrivent des mots dans un registre de maintien Modbus qui peut être en mémoire M ou un bloc de données. Le type de registre de maintien est indiqué par le paramètre MB_HOLD_REG.

Remarque

Paramétrage de MB_HOLD_REG

Les registres de maintien Modbus définis comme tableaux de mots, entiers, caractères longs, entiers non signés, octets, entiers courts, entiers courts non signés, caractères, doubles mots, entiers doubles, entiers doubles non signés ou réels peuvent être placés dans n'importe quelle zone de mémoire.

Vous devez placer les registres de maintien Modbus définis comme structures dans des DB non optimisés.

Pour un registre de maintien Modbus en mémoire M, utilisez le format de pointeur Any, c'est-à-dire P#"adresse bit" "type de données" "longueur" (par exemple, P#M1000.0 WORD 500).

Le tableau suivant montre des exemples de correspondance entre adresses Modbus et registre de maintien pour les codes de fonction Modbus 03 (lecture de mots), 06 (écriture d'un mot) et 16 (écriture de mots). La limite supérieure effective des adresses de DB dépend des limites de mémoire de travail et de mémoire M maximum pour chaque modèle de CPU.

Tableau 13- 65 Exemples de correspondance entre adresse Modbus et adresse de mémoire CPU

Adresse Modbus	Exemples pour le paramètre MB_HOLD_REG		
	P#M100.0 Word 5	P#DB10.DBx0.0 Word 5	"Recette".ingredient
40001	MW100	DB10.DBW0	"Recette".ingredient[1]
40002	MW102	DB10.DBW2	"Recette".ingredient[2]
40003	MW104	DB10.DBW4	"Recette".ingredient[3]
40004	MW106	DB10.DBW6	"Recette".ingredient[4]
40005	MW108	DB10.DBW8	"Recette".ingredient[5]

Liaisons serveur multiples

Plusieurs liaisons serveur peuvent être créées. Un API unique peut établir des liaisons simultanées à plusieurs clients Modbus TCP.

Un serveur Modbus TCP peut accepter des liaisons simultanées jusqu'au nombre maximum de liaisons de communication ouverte autorisé par l'API. Le nombre total de liaisons pour un API, clients et serveurs Modbus TCP compris, ne doit pas dépasser le nombre maximum de liaisons de communication ouverte prises en charge (Page 827). Les liaisons Modbus TCP peuvent être partagées entre liaisons de type client et serveur.

Les liaisons serveur individuelles simultanées doivent respecter les règles suivantes :

- Chaque liaison MB_SERVER doit utiliser un DB d'instance unique.
- Chaque liaison MB_SERVER doit affecter un numéro de port IP unique. Une seule liaison est acceptée par port.
- Chaque liaison MB_SERVER doit affecter un ID de liaison unique.
- L'instruction MB_SERVER doit être appelée individuellement pour chaque liaison (avec son DB d'instance respectif).

L'ID de liaison doit être unique pour chaque liaison individuelle. Un ID de liaison unique doit être utilisé pour chaque DB d'instance individuel. Le DB d'instance et l'ID de liaison forment une paire qui doit être unique pour chaque liaison.

13.5 Communication Modbus

Tableau 13- 66 Codes de fonction de diagnostic Modbus

Fonctions de diagnostic Modbus pour MB_SERVER		
Codes	Sous-fonction	Description
08	0x0000	Renvoi d'un test d'écho des données de requête : MB_SERVER renvoie en écho à un client Modbus un mot de données reçu.
08	0x000A	Effacement du compteur d'événements de communication : MB_SERVER efface le compteur d'événements de communication qui est utilisé pour la fonction Modbus 11.
11		Lecture du compteur d'événements de communication : MB_SERVER utilise un compteur d'événements de communication interne pour enregistrer le nombre de demandes de lecture et d'écriture Modbus envoyées au serveur Modbus qui ont abouti. Le compteur ne s'incrémente pas pour les demandes des fonctions 8 et 11 ni pour les demandes entraînant une erreur de communication. La fonction de diffusion générale n'est pas disponible pour Modbus TCP, car une seule liaison client-serveur existe à un moment donné.

Variables de bloc de données (DB) de l'instruction MB_SERVER

Ce tableau montre les variables statiques publiques qui sont sauvegardées dans le bloc de données d'instance pour MB_SERVER pouvant être utilisées dans votre programme.

Tableau 13- 67 Variables statiques publiques de MB_SERVER

Variable	Type de données	Valeur par défaut	Description
HR_Start_Offset	Word	0	Définit l'adresse de début du registre de maintien Modbus.
Request_Count	Word	0	Nombre de toutes les demandes reçues par ce serveur
Server_Message_Count	Word	0	Nombre de demandes reçues pour ce serveur spécifique
Xmt_Rcv_Count	Word	0	Nombre d'émissions ou de réceptions pendant lesquelles une erreur s'est produite. Cette variable est également incrémentée si un message reçu est un message Modbus invalide.
Exception_Count	Word	0	Erreurs Modbus spécifiques nécessitant le renvoi d'une exception
Success_Count	Word	0	Nombre de demandes reçues pour ce serveur spécifique qui ne contient pas d'erreurs de protocole.
Connected	Bool	0	Indique si la liaison au client affecté est établie ou coupée : 1=liaison établie, 0=liaison coupée
QB_Start	UInt	0	Adresse de début des octets de sortie dans lesquels la CPU peut écrire (QB0 à QB65535)
QB_Count	UInt	65535	Nombre d'octets dans lesquels un appareil distant peut écrire. Si QB_Count est égal à 0, un appareil distant ne peut pas écrire dans les sorties. Exemple : pour que l'écriture soit autorisée uniquement dans les octets QB10 à QB17, QB_Start = 10 et QB_Count = 8.
QB_Read_Start	UInt	0	Adresse de début des octets de sortie que la CPU peut lire (QB0 à QB65535)
QB_Read_Count	UInt	65535	Nombre d'octets de sortie qu'un appareil distant peut lire. Si QB_Count est égal à 0, un appareil distant ne peut pas lire dans les sorties. Exemple : pour que la lecture soit autorisée uniquement dans les octets QB10 à QB17, QB_Start = 10 et QB_Count = 8.

Variable	Type de données	Valeur par défaut	Description
IB_Read_Start	UInt	0	Adresse de début des octets d'entrée que la CPU peut lire (IB0 à IB65535)
IB_Read_Count	UInt	65535	Nombre d'octets d'entrée qu'un appareil distant peut lire. Si IB_Count est égal à 0, un appareil distant ne peut pas lire dans les entrées. Exemple : pour que la lecture soit autorisée uniquement dans les octets IB10 à IB17, IB_Start = 10 et IB_Count = 8.
NDR_immediate	Bool	FALSE	Même signification que le paramètre NDR (New Data Ready, nouvelles données prêtes). MB_SERVER actualise "NDR_immediate" dans le même appel qui traite une demande d'écriture Modbus TCP.
DR_immediate	Bool	FALSE	Même signification que le paramètre DR (Data Read, données lues). MB_SERVER actualise "DR_immediate" dans le même appel qui traite une demande d'écriture Modbus TCP.

Votre programme peut écrire des données dans la variable HR_Start_Offset et commander le fonctionnement du serveur Modbus. Les autres variables peuvent être lues pour visualiser l'état de Modbus.

Les versions suivantes sont requises pour la disponibilité des variables de bloc de données (DB) de l'instruction MB_SERVER :

Tableau 13- 68 Versions requises pour la disponibilité des variables de DB de l'instruction MB_SERVER : instruction, TIA Portal et CPU S7-1200

Version de l'instruction MB_SERVER	Version de TIA Portal	Version du firmware (FW) de la CPU S7-1200	Variables du bloc de données
4.0	V14 SP1	CPU FW V4.0 ou plus	QB_Start
			QB_Count
5.0	V15.1	CPU FW V4.2 ou plus	QB_Start
			QB_Count
			QB_Read_Start
			QB_Read_Count
			IB_Read_Start
			IB_Read_Count
			NDR_immediate
DR_immediate			

HR_Start_Offset

Les adresses de registre de maintien Modbus commencent à 40001, ce qui correspond à l'adresse de début du registre de maintien dans la mémoire API. Vous pouvez toutefois utiliser la variable "HR_Start_Offset" afin que le registre de maintien Modbus commence à une adresse autre que 40001.

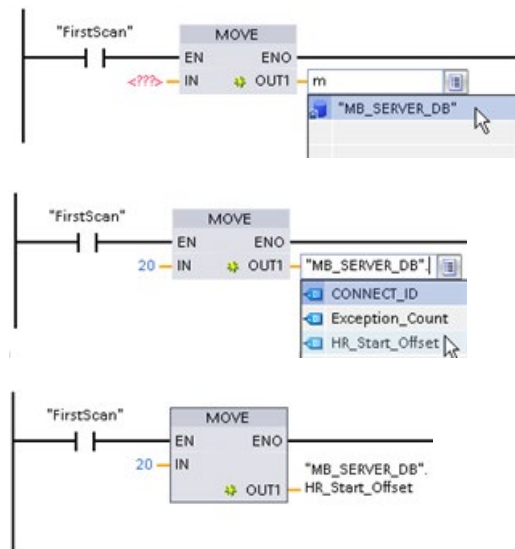
Par exemple, si le registre de maintien commence à MW100 et s'il a une longueur de 100 mots. Un décalage de 20 indique une adresse de début de registre située à 40021 au lieu de 40001. Toute adresse inférieure à 40021 ou supérieure à 40119 entraînera une erreur d'adressage.

Tableau 13- 69 Exemple d'adressage de registre de maintien Modbus

HR_Start_Offset	Adresse	Minimum	Maximum
0	Adresse Modbus (mot)	40001	40099
	Adresse S7-1200	MW100	MW298
20	Adresse Modbus (mot)	40021	40119
	Adresse S7-1200	MW100	MW298

HR_Start_Offset est une donnée mot dans le bloc de données d'instance MB_SERVER, qui indique l'adresse de début du registre de maintien Modbus. Vous pouvez définir cette variable statique publique à l'aide de la liste déroulante d'aide pour les paramètres après avoir inséré MB_SERVER dans votre programme.

Par exemple, une fois que vous avez inséré MB_SERVER dans un réseau CONT, vous pouvez aller à un réseau précédent et affecter HR_Start_Offset. L'adresse de début doit être affectée avant l'exécution de MB_SERVER.



Entrée d'une variable pour le serveur Modbus à l'aide du nom de DB par défaut :

1. Placez le curseur dans le champ de paramètre et tapez la lettre m.
2. Sélectionnez "MB_SERVER_DB" dans la liste déroulante des noms de DB.
3. Sélectionnez "MB_SERVER_DB.HR_Start_Offset" dans la liste déroulante des variables de DB.

Accès aux zones de données des blocs de données au lieu de l'accès direct aux adresses Modbus

À partir de la version V5.0 de l'instruction MB_SERVER et de la version V4.2 du firmware de la CPU S7-1200, vous pouvez accéder à des zones de données dans les DB au lieu d'accéder directement aux mémoires images et aux registres de maintien. Pour ce faire, vous devez décocher les cases des options "Sauvegarder uniquement dans la mémoire de chargement" et "Accès au bloc optimisé" dans la page des propriétés "Attributs" du DB global.

Si, à l'arrivée d'une demande Modbus, vous n'avez pas défini de zone de données pour le type de données Modbus du code de fonction correspondant, la CPU traite la demande comme dans les versions précédentes de l'instruction : vous accédez directement aux mémoires images et aux registres de maintien.

Si vous avez défini une zone de données pour le type de données Modbus du code de fonction, l'instruction MB_SERVER lit ou écrit dans cette zone de données. C'est le type de tâche qui détermine s'il s'agit d'une opération de lecture ou d'écriture.

Une demande Modbus individuelle permet de lire ou d'écrire dans une zone de données uniquement. Ainsi, si vous voulez lire des registres de maintien qui s'étendent sur plusieurs zones de données, vous devez exécuter plusieurs demandes Modbus.

Voici les règles pour la définition des zones de données :

- Vous pouvez définir huit zones de données au maximum dans différents DB ; chaque DB ne doit contenir qu'une seule zone de données. Une demande Modbus individuelle ne peut lire ou écrire que dans une zone de données exactement. Chaque zone de données correspond à une zone d'adresses Modbus. Vous définissez les zones de données dans la variable statique "Data_Area_Array" du DB d'instance.
- Si vous voulez utiliser moins de huit zones de données, vous devez placer les zones de données requises l'une derrière l'autre sans intervalle entre elles. La première entrée vide dans les zones de données met fin à la recherche de zone de données pendant le traitement. Si, par exemple, vous définissez les éléments de tableau 1, 2, 4 et 5, "Data_Area_Array" reconnaît uniquement les éléments 1 et 2 puisque l'élément 3 est vide.

13.5 Communication Modbus

- Le tableau Data_Area_Array est constitué de huit éléments : Data_Area_Array[1] à Data_Area_Array[8].
- Chaque élément Data_Area_Array[x], 1 <= x <= 8, est un UDT de type MB_DataArea structuré de la manière suivante :

Paramètre	Type de données	Signification
data_type	UInt	<p>Identifiant pour le type de données Modbus qui est mappé sur cette zone de données :</p> <ul style="list-style-type: none"> • 0 : identifiant pour un élément de tableau vide ou une zone de données inutilisée. Les valeurs de "db", "start" et "length" sont alors sans objet. • 1 : sortie dans la mémoire image (utilisation avec les codes de fonction 1, 5 et 15) • 2 : entrée dans la mémoire image (utilisation avec le code de fonction 2) • 3 : registre de maintien (utilisation avec les codes de fonction 3, 6 et 16) • 4 : registre d'entrée (utilisation avec le code de fonction 4) <p>Remarque : Si vous avez défini une zone de données pour un type de données Modbus, l'instruction MB_SERVER ne peut plus accéder directement à ce type de données Modbus. Si l'adresse d'une demande Modbus pour un tel type de données ne correspond pas à une zone de données définie, la valeur W#16#8383 est renvoyée dans le paramètre STATUS.</p>
db	UInt	<p>Numéro du bloc de données dans lequel le registre ou les bits Modbus définis par la suite sont mappés.</p> <p>Le numéro de DB doit être unique dans les zones de données. Le même numéro de DB ne doit pas être défini dans plusieurs zones de données.</p> <p>Vous devez décocher les cases des options "Sauvegarder uniquement dans la mémoire de chargement" et "Accès au bloc optimisé" dans la page des propriétés "Attributs" du DB global.</p> <p>Les zones de données commencent également à l'adresse d'octet 0 du DB.</p> <p>Valeurs autorisées : 1 à 60999</p>
start	UInt	<p>Première adresse Modbus mappée sur le bloc de données à partir de l'adresse 0.0</p> <p>Valeurs autorisées : 0 à 65535</p>
length	UInt	<p>Nombre de bits (pour les valeurs 1 et 2 de "data_type") ou nombre de registres (pour les valeurs 3 et 4 de "data_type")</p> <p>Les zones d'adresses Modbus d'un type de données Modbus précis ne doivent pas se chevaucher.</p> <p>Valeurs autorisées : 1 à 65535</p>

Exemples de définition de zones de données :

- Premier exemple : data_type = 3, db = 1, start = 10, length = 6

La CPU mappe les registres de maintien (data_type = 3) dans le bloc de données 1 (db = 1), plaçant l'adresse Modbus 10 (start = 10) dans le mot de données 0 et la dernière adresse Modbus valide 15 (length = 6) dans le mot de données 5.

- Deuxième exemple : data_type = 2, db = 15, start = 1700, length = 112

La CPU mappe les entrées (data_type = 2) dans le bloc de données 15 (db = 15), plaçant l'adresse Modbus 1700 (start = 1700) dans le mot de données 0 et la dernière adresse Modbus valide 1811 (length = 112) dans le mot de données 111.

Codes d'erreur

Tableau 13- 70 Codes d'erreur d'exécution de MB_SERVER ¹

STATUS (W#16#)	Code réponse envoyé au serveur Modbus(B#16#)	Erreurs de protocole Modbus
7001		MB_SERVER attend qu'un client Modbus se connecte au port TCP affecté. Ce code est envoyé à la première exécution d'une opération de connexion ou de déconnexion.
7002		MB_SERVER attend qu'un client Modbus se connecte au port TCP affecté. Ce code est retourné pour les exécutions suivantes en attendant l'achèvement d'une opération de connexion ou de déconnexion.
7003		Une opération de déconnexion s'est achevée avec succès (valable pendant un seul cycle API).
8187		MB_HOLD_REG est invalide : il désigne peut-être un DB optimisé ou une zone inférieure à 2 octets.
818C		Le pointeur MB_HOLD_REG désigne une zone DB non-optimisée (ce doit être une zone de DB global non optimisée ou une zone de mémoire M) ou le délai d'attente de processus bloqué dépasse la limite de 55 secondes. (spécifique au S7-1200)
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou accès en dehors des limites de la zone d'adresses de MB_HOLD_REG
8384	03	Erreur de valeur de données
8385	03	Code de diagnostic de données non pris en charge (code de fonction 08)

¹ En plus des erreurs MB_SERVER présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication par blocs T sous-jacentes (TCON, TDISCON, TSEND et TRCV).

13.5.2.4 Exemples Modbus TCP

Exemple : Liaisons TCP multiples MB_SERVER

Vous pouvez avoir plusieurs liaisons serveur Modbus TCP. Pour ce faire, vous devez exécuter MB_SERVER de manière indépendante pour chaque liaison. Chaque liaison doit utiliser un DB d'instance, un ID de liaison et un port IP qui lui soient propres. Le S7-1200 n'admet qu'une seule liaison par port IP.

Pour des performances optimales, il est préférable d'exécuter MB_SERVER à chaque cycle de programme pour chaque liaison.

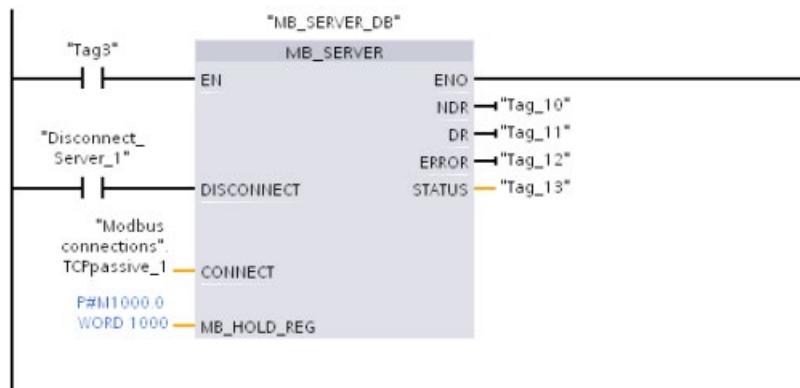
Le paramètre CONNECT utilise le type de données système TCP_IP_v4. Par exemple, ces structures de données se trouvent dans un DB appelé "Liaisons Modbus". Le bloc de données "Liaisons Modbus" contient deux structures TCP_IP_v4 "TCPpassive_1" (pour liaison 1) et "TCP_passive_2" (pour liaison 2). Les propriétés de liaison ID et LocalPort décrites dans les commentaires réseau sont des éléments de données stockés dans la structure de données CONNECT.

Les données CONNECT TCP_IP_v4 contiennent également une adresse IP dans le tableau ADDR RemoteAddress. Les affectations d'adresse IP à l'intérieur de TCPpassive_1 et TCP_passive_2 n'affectent pas l'établissement de liaisons serveur TCP, mais déterminent les clients Modbus TCP autorisés à communiquer via les liaisons avec chaque MB_SERVER. MB_SERVER est passivement à l'écoute d'un message provenant du client Modbus et il compare l'adresse IP du message entrant avec l'adresse IP stockée dans le tableau ADDR RemoteAddress correspondant.

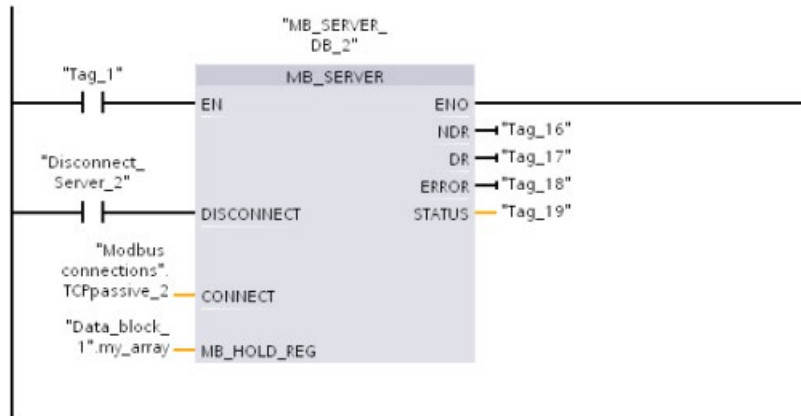
Trois adresses IP MB_SERVER sont possibles pour les deux instructions MB_SERVER :

- **Adresse IP = 0.0.0.0**
Chaque MB_SERVER répondra à tout client Modbus TCP qui utilise une adresse IP.
- **Adresse IP = Adresse IP identique dans TCPpassive_1 et TCPpassive_2**
Les deux liaisons MB_SERVER ne répondront qu'aux demandes de clients Modbus provenant de cette adresse IP.
- **Adresse IP = Adresse IP différente dans TCPpassive_1 et TCPpassive_2**
Chaque MB_SERVER ne répondra qu'aux clients Modbus provenant de l'adresse IP stockée dans les données TCP_IP_v4 correspondantes.

Réseau 1 : Liaison #1, DB d'instance = "MB_SERVER_DB", dans "Liaisons Modbus.TCPpassive_1" (ID = 1 et LocalPort = 502)



Réseau 2 : Liaison #2, DB d'instance = "MB_SERVER_DB_1", dans "Liaisons Modbus.TCPpassive_2" (ID = 2 et LocalPort = 503)



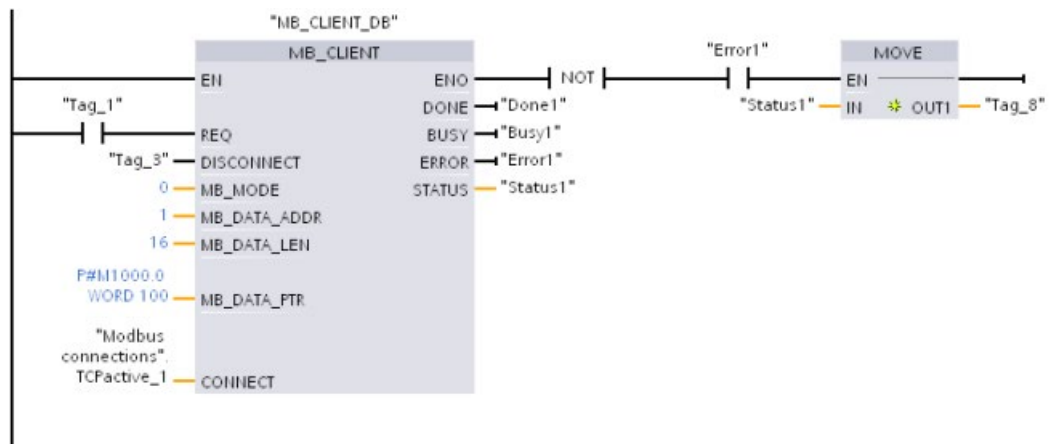
Exemple : MB_CLIENT 1 : Plusieurs demandes avec une liaison TCP commune

Vous pouvez envoyer plusieurs demandes client Modbus par la même liaison. Pour ce faire, utilisez les mêmes DB d'instance, ID de liaison et numéro de port.

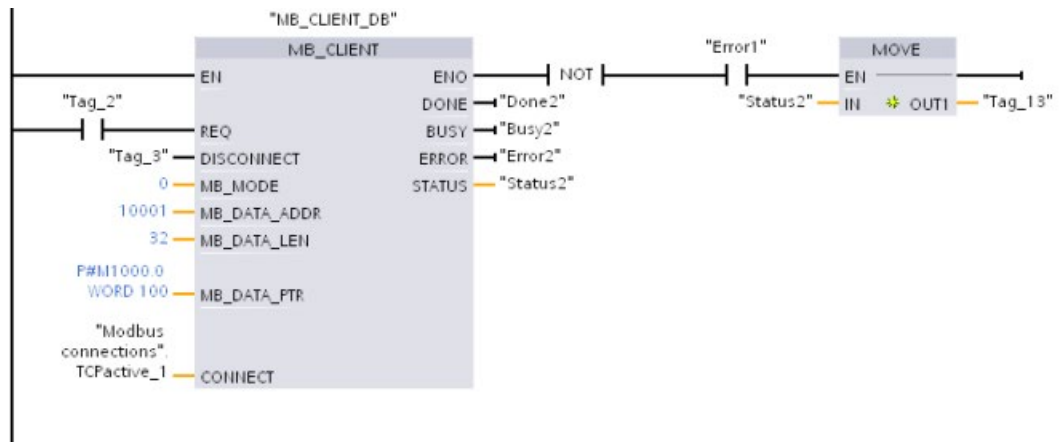
Comme les deux boîtes MB_CLIENT utilisent la même structure de données TCON_IP_v4 au paramètre CONNECT ("Modbus_connections".TCPActive_1), l'ID de liaison, le numéro de port et l'adresse IP sont identiques. Les données d'adresse IP CONNECT indiquent l'adresse IP du serveur Modbus TCP cible.

Un seul MB_CLIENT peut être actif à un instant donné. Une fois qu'un client a achevé son exécution, le client suivant peut commencer à s'exécuter. C'est la logique de votre programme qui gère l'ordre d'exécution. L'exemple montre deux clients qui lisent des données distantes d'un client Modbus unique et transfèrent les données dans la CPU du client Modbus (mémoire M commence à M1000.0). Une erreur renvoyée est détectée, ce qui est facultatif.

Réseau 1 : Fonction Modbus 1 - Lecture de 16 bits de sortie d'un serveur Modbus TCP avec l'adresse IP indiquée dans "Liaisons Modbus".TCPActive_1.



Réseau 2 : Fonction Modbus 2 - Lecture de 32 bits d'entrée d'un serveur Modbus TCP avec l'adresse IP indiquée dans "Liaisons Modbus".TCPActive_1.



Exemple : MB_CLIENT 2 : Plusieurs demandes avec des liaisons TCP différentes

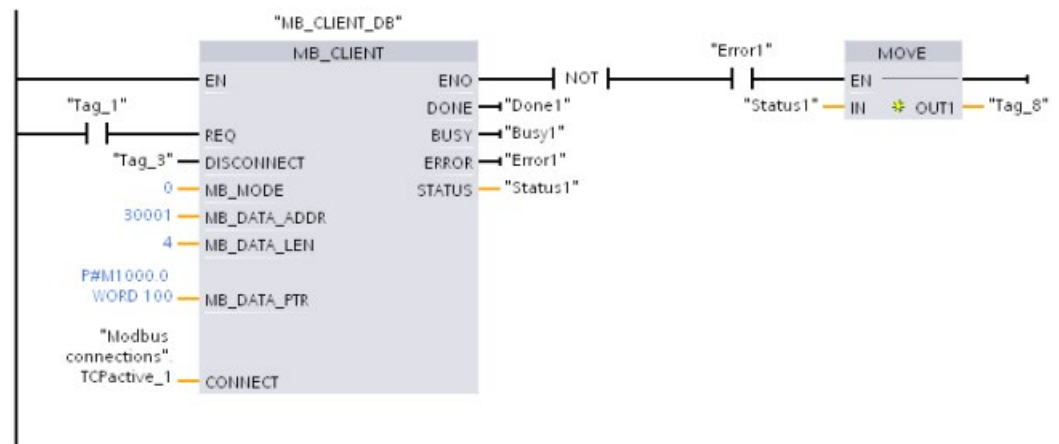
Vous pouvez envoyer plusieurs demandes client Modbus TCP par le biais de liaisons différentes. Pour ce faire, vous devez utiliser des DB d'instance et des ID de liaison différents.

Le numéro de RemotePort (port IP) doit être différent si les liaisons sont établies vers le même serveur Modbus. Si les liaisons concernent des serveurs différents, aucune restriction ne s'applique au numéro de port IP.

L'exemple montre deux clients Modbus TCP qui transfèrent des données distantes de deux serveurs Modbus TCP différents dans la même zone de mémoire de la CPU locale, qui commence à l'adresse M1000.0. En outre, une erreur renvoyée est détectée, ce qui est facultatif.

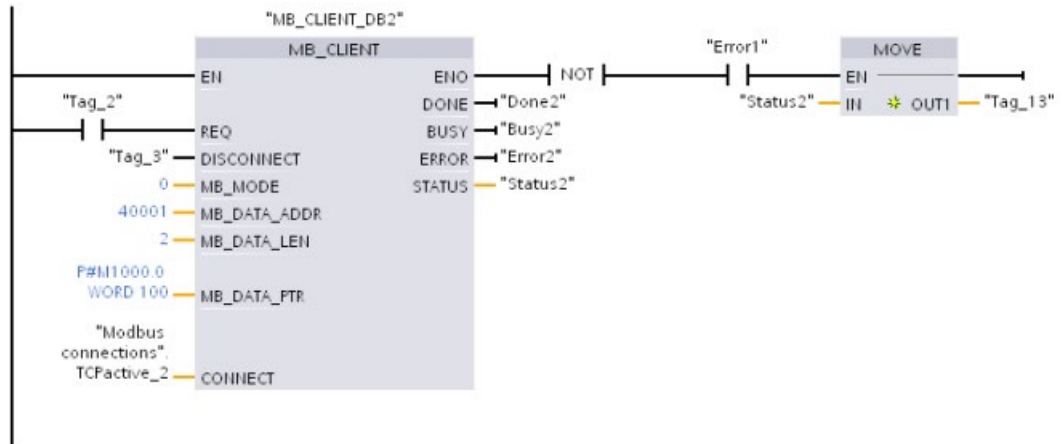
Réseau 1 : Fonction Modbus 4 - Lecture de mots de mémoire image des entrées à partir d'un serveur Modbus TCP

Paramètre CONNECT = "Liaisons Modbus".TCPActive_1 : Connection ID = 1, RemoteAddress = 192.168.2.241, RemotePort = 502



Réseau 2 : Fonction Modbus 3 - Lecture de mots d'un registre de maintien d'un serveur Modbus TCP

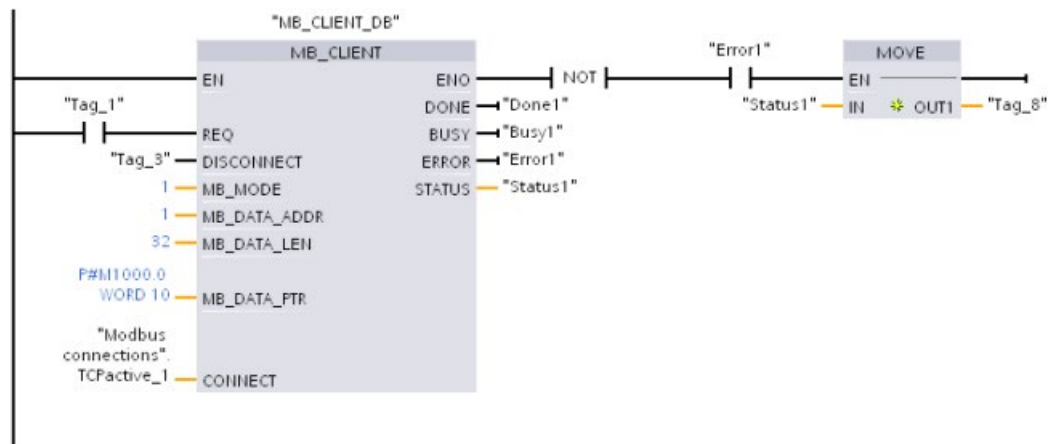
Paramètre CONNECT = "Liaisons Modbus".TCPActive_2 : Connection ID = 2,
RemoteAddress = 192.168.2.242, **RemotePort** = 502



Exemple : MB_CLIENT 3 : Demande d'écriture dans la mémoire image des sorties

Cet exemple montre une demande client Modbus pour le transfert de bits de la mémoire de la CPU locale (commence à M1000.0) dans un serveur Modbus TCP distant.

Réseau 1 : Fonction Modbus 15 : Ecriture de bits de sortie dans un serveur Modbus

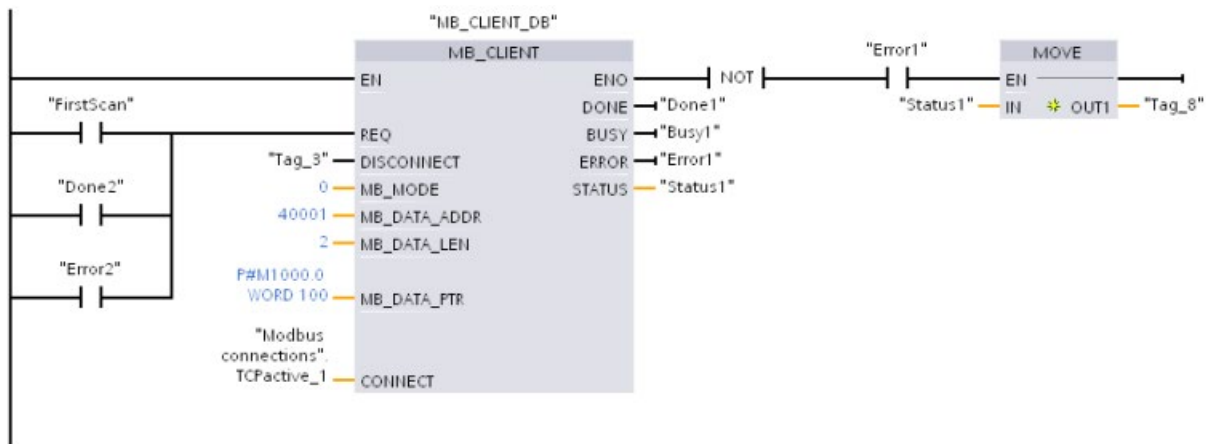


Exemple : MB_CLIENT 4 : Coordination de plusieurs demandes

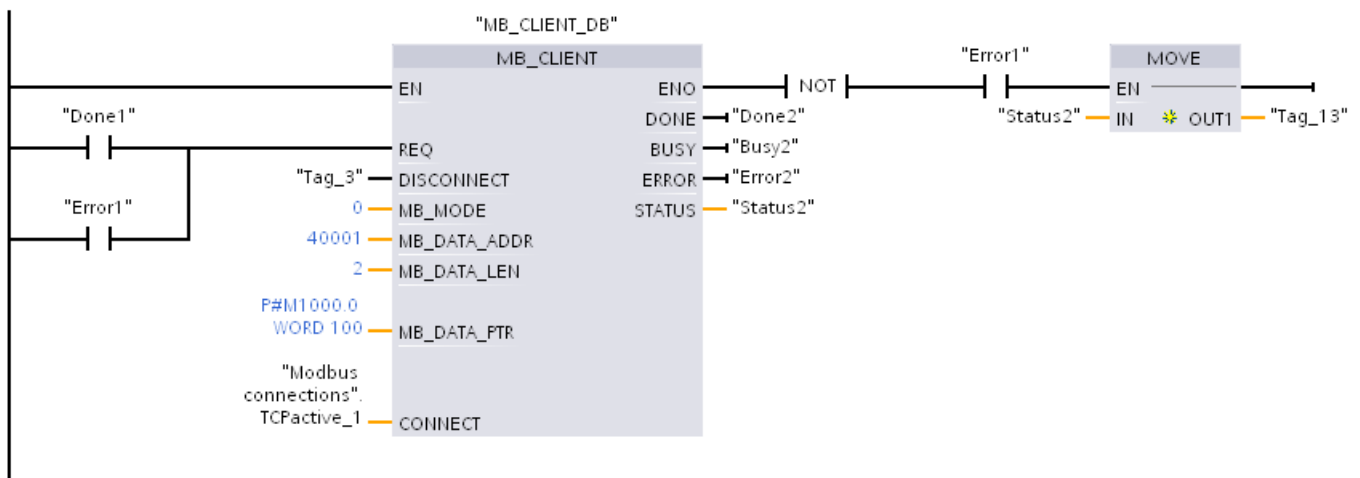
Vous devez veiller à ce que chaque demande Modbus TCP individuelle achève son exécution. La logique de votre programme doit gérer l'ordre d'exécution. L'exemple ci-dessous montre comment les sorties de la première et de la deuxième demande client peuvent être utilisées pour gérer l'ordre d'exécution.

L'exemple montre deux clients qui utilisent les mêmes données de liaison CONNECT (utilisées à des moments différents). Les clients transfèrent les données du registre de maintien du même serveur Modbus TCP distant dans la même adresse M de mémoire de la CPU locale. En outre, une erreur renvoyée est détectée, ce qui est facultatif.

Réseau 1 : Fonction Modbus 3 - Lecture de mots d'un registre de maintien du serveur Modbus TCP



Réseau 2 : Fonction Modbus 3 - Lecture de mots d'un registre de maintien du serveur Modbus TCP



13.5.3 Modbus RTU

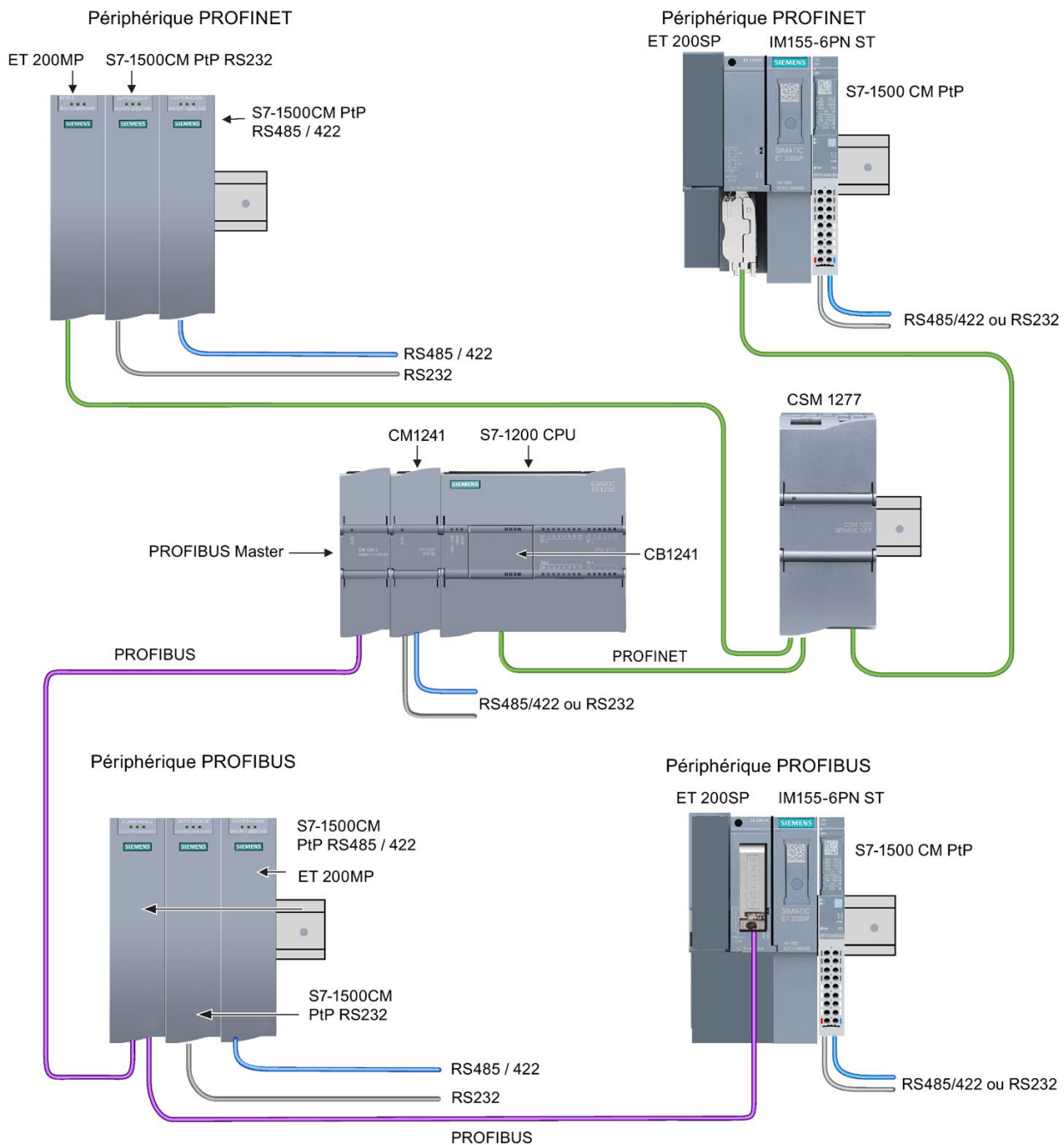
13.5.3.1 Vue d'ensemble

À partir de la version V4.1 de la CPU S7-1200 avec STEP 7 V13 SP1, la CPU étend les fonctionnalités de Modbus RTU à l'utilisation d'un châssis de périphérie décentralisée PROFINET ou PROFIBUS pour communiquer avec divers appareils (lecteurs RFID, dispositifs GPS et autres) :

- PROFINET (Page 830) : Vous connectez l'interface Ethernet de la CPU S7-1200 à un coupleur PROFINET. Les modules de communication PtP dans le châssis avec le coupleur peuvent ensuite fournir des communications série avec les appareils PtP.
- PROFIBUS (Page 1002) : Vous insérez un module de communication PROFIBUS dans le côté gauche du châssis avec la CPU S7-1200. Vous connectez le module de communication PROFIBUS à un châssis contenant un coupleur PROFIBUS. Les modules de communication PtP dans le châssis avec le coupleur peuvent ensuite fournir des communications série avec les appareils PtP.

Pour cette raison, le S7-1200 accepte deux jeux d'instructions PtP :

- Instructions d'héritage Modbus RTU (Page 1337) : Ces instructions Modbus RTU existaient avec la version V4.0 du S7-1200 et fonctionnent uniquement avec des communications série à l'aide d'un module de communication CM 1241 ou d'un communication board CB 1241.
- Instructions Modbus RTU (Page 1254) : Ces instructions Modbus RTU offrent toutes les fonctionnalités des instructions d'héritage, plus la capacité à se connecter à la périphérie décentralisée PROFINET et PROFIBUS. Ces instructions Modbus RTU vous permettent de configurer les communications entre les modules de communication PtP dans le châssis de périphérie décentralisée et les appareils PtP. Les modules CM 1241 S7-1200 doivent posséder au moins la version V2.1 du firmware pour utiliser ces instructions Modbus RTU.



Remarque

Avec la version V4.1 du S7-1200, vous pouvez utiliser les instructions point à point pour tous les types de communication point à point : série, série sur PROFINET, et série sur PROFIBUS. STEP 7 fournit les instructions d'héritage point à point uniquement pour prendre en charge les programmes existants. Cependant, les instructions d'héritage fonctionnent toujours avec les CPU V4.1 ainsi qu'avec les CPU V4.0 et antérieures. Vous n'avez pas besoin de convertir les programmes antérieurs d'un jeu d'instructions à l'autre.

13.5.3.2 Sélection de la version des instructions Modbus RTU

Trois versions d'instructions Modbus RTU sont disponibles dans STEP 7 :

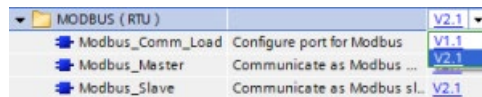
- Version 1.1 : compatible avec les CPU de version V4.0 et ultérieure et les CM de version V2.1 et ultérieure
- Version 2.1 : compatible avec les CPU de version V4.0 et ultérieure et les CM de version V2.1 et ultérieure
- Version 3.0 : compatible avec les CPU de version V4.0 et ultérieure et les CM de version V2.1 et ultérieure

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

Vous ne pouvez pas utiliser les deux versions des instructions avec le même module, mais deux modules différents peuvent utiliser différentes versions des instructions. N'utilisez pas à la fois des instructions de version 1.x et 2.y dans le même programme CPU. Les instructions Modbus RTU dans votre programme doivent avoir le même numéro de version principale (1.x, 2.y ou V.z). Toutefois, les instructions individuelles à l'intérieur d'un groupe de version principale peuvent avoir des versions secondaires différentes (1.x).



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.



Pour changer la version des instructions Modbus RTU, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction Modbus RTU dans votre programme, une nouvelle instance de FB est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB dans l'arborescence de projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction Modbus RTU dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB Modbus RTU dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction Modbus RTU.

13.5.3.3 Nombre maximum d'esclaves Modbus pris en charge

L'adressage Modbus prend en charge 247 esclaves maximum (numéros d'esclave compris entre 1 et 247). Chaque segment de réseau Modbus peut avoir 32 périphériques maximum, sur la base des capacités de charge et de commande de l'interface RS485. Lorsque la limite de 32 périphériques est atteinte, il faut utiliser un répéteur pour étendre le réseau au segment suivant. Sept répéteurs sont requis pour prendre en charge les 247 esclaves connectés à un maître pour RS485.

Les répéteurs Siemens ne fonctionnent qu'avec PROFIBUS ; ils sont conçus pour surveiller le jeton circulant PROFIBUS. Vous ne pouvez pas utiliser les répéteurs Siemens avec d'autres protocoles. Par conséquent, vous avez besoin de répéteurs d'autres fabricants pour Modbus.

Les retards d'acquiescement de Modbus sont longs par défaut ; l'utilisation de plusieurs répéteurs ne crée pas de problème de retard. Le maître Modbus ignore fait qu'un esclave est lent à répondre ou que plusieurs répéteurs retardent la réponse.

13.5.3.4 Instructions Modbus RTU

Instruction Modbus_Comm_Load (Configurer SIPLUS I/O ou un port sur le module PtP pour Modbus RTU)

Tableau 13- 71 Instruction Modbus_Comm_Load

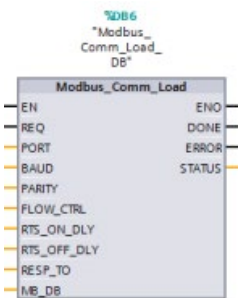
CONT/LOG	SCL	Description
	<pre>"Modbus_Comm_Load_DB" (REQ:=_bool_in_, PORT:=_uint_in_, BAUD:=_udint_in_, PARITY:=_uint_in_, FLOW_CTRL:=_uint_in_, RTS_ON_DLY:=_uint_in_, RTS_OFF_DLY:=_uint_in_, RESP_TO:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_DB:=_fbtref_inout_);</pre>	<p>L'instruction Modbus_Comm_Load configure SIPLUS I/O ou un port point à point pour la communication via le protocole Modbus RTU.</p> <p>Options matérielles pour le port Modbus RTU : Installez jusqu'à trois CM (RS485 ou RS232), ainsi qu'un CB (RS485).</p> <p>Options SIPLUS I/O Modbus RTU : Installez ET 200MP S7-1500CM PtP (RS485 / 422 ou RS232) ou ET 200SP S7-1500 CM PtP (RS485 / 422 ou RS232)</p> <p>Un bloc de données d'instance est automatiquement affecté lorsque vous insérez l'instruction Modbus_Comm_Load dans votre programme.</p>

Tableau 13- 72 Types de données pour les paramètres

Paramètre et type		Type de données	Description
EN	IN	Bool	Remarque : L'instruction Modbus RTU Modbus_Comm_Load utilise les instructions RDREC et WRREC pour initialiser le module PTP. Toutefois, l'instruction RDREC/WRREC fonctionne de manière asynchrone, ce qui signifie qu'il faut plusieurs cycles pour achever l'instruction. Vous devez donc faire en sorte que le paramètre EN de l'instruction Modbus_Comm_Load reste à vrai jusqu'à l'achèvement de l'instruction RDREC/WRREC.
REQ	IN	Bool	Un front montant déclenche l'opération. (version 2.0 uniquement)
PORT	IN	Port	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponibles au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
BAUD	IN	UDInt	Sélection du débit : 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Toutes les autres valeurs sont invalides.
PARITY	IN	UInt	Sélection de la parité : <ul style="list-style-type: none"> • 0 = Pas de parité • 1 = Parité impaire • 2 = Parité paire
FLOW_CTRL ¹	IN	UInt	Sélection du contrôle de flux : <ul style="list-style-type: none"> • 0 (valeur par défaut) = Pas de contrôle de flux • 1 = Contrôle de flux matériel avec RTS toujours activé (ne s'applique pas aux ports RS485) • 2 = Contrôle de flux matériel avec RTS commuté
RTS_ON_DLY ¹	IN	UInt	Sélection du retard RTS activé : <ul style="list-style-type: none"> • 0 = (valeur par défaut) Pas de délai à partir de RTS actif jusqu'à ce que le premier caractère du message soit émis • 1 à 65535 = Délai en millisecondes à partir de RTS actif jusqu'à ce que le premier caractère du message soit émis (ne s'applique pas aux ports RS485). Les retards RTS sont appliqués indépendamment de la sélection FLOW_CTRL.
RTS_OFF_DLY ¹	IN	UInt	Sélection du retard RTS désactivé : <ul style="list-style-type: none"> • 0 = (valeur par défaut) Pas de délai à partir du dernier caractère émis jusqu'à ce que RTS soit désactivé • 1 à 65535 = Délai en millisecondes à partir du dernier caractère émis jusqu'à ce que RTS soit désactivé (ne s'applique pas aux ports RS485). Les retards RTS sont appliqués indépendamment de la sélection FLOW_CTRL.

13.5 Communication Modbus

Paramètre et type		Type de données	Description
RESP_TO ¹	IN	UInt	Dépassement du délai d'attente de la réponse Temps en millisecondes accordé par Modbus_Master pour que l'esclave réponde. Si l'esclave ne répond pas pendant cette durée, Modbus_Master renouvellera la demande ou mettra fin à la demande avec une erreur lorsque le nombre de nouvelles tentatives indiqué aura été envoyé. 5 ms à 65535 ms (valeur par défaut = 1000 ms).
MB_DB	IN	Variant	Référence au bloc de données d'instance utilisé par les instructions Modbus_Master ou Modbus_Slave. Une fois que vous avez inséré Modbus_Master ou Modbus_Slave dans votre programme, l'identificateur de DB apparaît dans la liste déroulante d'aide pour les paramètres disponibles au niveau du connecteur MB_DB de la boîte.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur. (version 2.0 uniquement)
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

¹ Paramètres facultatifs pour Modbus_Comm_Load (V 2.x ou ultérieur). Cliquez sur la flèche en bas de la boîte CONT/LOG pour agrandir la boîte et inclure ces paramètres.

Modbus_Comm_Load est exécuté pour configurer un port pour le protocole Modbus RTU. Une fois qu'un port est configuré pour le protocole Modbus RTU, il ne peut être utilisé que par les instructions Modbus_Master ou Modbus_Slave.

Il faut exécuter Modbus_Comm_Load une fois pour chaque port de communication utilisé pour la communication Modbus à configurer. Affectez un DB d'instance Modbus_Comm_Load unique pour chaque port que vous utilisez. Vous pouvez installer trois modules de communication (RS232 ou RS485) au maximum et un Communication Board (RS485) dans la CPU. Appelez Modbus_Comm_Load dans un OB de démarrage et exécutez-le une fois ou déclenchez l'appel via le memento système Premier cycle (Page 113) pour l'exécuter une fois. Ne réexécutez Modbus_Comm_Load que si des paramètres de communication tels que la vitesse de transmission ou la parité doivent changer.

Si vous utilisez la bibliothèque Modbus avec un module dans un châssis décentralisé, l'instruction Modbus_Comm_Load doit être exécutée dans un programme d'alarme cyclique (par exemple, une fois par seconde ou une fois toutes les 10 secondes). Si l'alimentation du châssis décentralisé est perdue ou si le module est débroché au moment de la restauration du fonctionnement du module, seul l'ensemble de paramètres HWConfig est envoyé au module PtP. Toutes les demandes initiées par le délai Modbus_Master, et le Modbus_Slave restent sans réponse (aucune réponse à aucun message). L'exécution cyclique de l'instruction Modbus_Comm_Load résout ces problèmes.

Un bloc de données d'instance est affecté à Modbus_Master ou Modbus_Slave lorsque vous insérez ces instructions dans votre programme. Ce bloc de données d'instance est référencé lorsque vous définissez le paramètre MB_DB pour l'instruction Modbus_Comm_Load.

Variables du bloc de données d'instance de Modbus_Comm_Load

Le tableau suivant montre les variables statiques publiques sauvegardées dans le DB d'instance de Modbus_Comm_Load qui peuvent être utilisées dans votre programme :

Tableau 13- 73 Variables statiques du DB d'instance de Modbus_Comm_Load

Variable	Type de données	Valeur par défaut	Description
ICHAR_GAP	Word	0	Délai inter-caractère maximum. Ce paramètre indiqué en millisecondes augmente la période prévue entre les caractères reçus. Le nombre correspondant de temps bit pour ce paramètre est ajouté à la valeur Modbus par défaut de 35 temps bit (3,5 temps caractère).
RETRIES	Word	2	Nombre de nouvelles tentatives exécutées par le maître avant le renvoi du code d'erreur 0x80C8 pour "Pas de réponse"
EN_SUPPLY_VOLT	Bool	0	Activer le diagnostic pour la tension d'alimentation manquante L+.
MODE	USInt	0	Mode de fonctionnement Les modes de fonctionnement valides sont les suivants : <ul style="list-style-type: none"> • 0 = duplex intégral (RS232) • 1 = duplex intégral (RS422) mode 4 fils (point à point) • 2 = duplex intégral (RS422) mode 4 fils (maître multipoint, CM PtP (ET 200SP)) • 3 = duplex intégral (RS422) mode 4 fils (esclave multipoint, CM PtP (ET 200SP)) • 4 = semi-duplex (RS485) mode deux fils (voir remarque ci-dessous)
LINE_PRE	USInt	0	Préconfiguration de la ligne de réception Les états initiaux valides sont les suivants : <ul style="list-style-type: none"> • 0 = état initial "non" (voir remarque ci-dessous) • 1 = signal R(A) = 5 V CC, signal R(B) = 0 V CC (détection de rupture) : la détection de rupture est possible avec cet état initial. Peut seulement être sélectionné avec : "duplex intégral (RS422) mode 4 fils (couplage point à point)" et "duplex intégral (RS422) mode 4 fils (esclave multipoint)". • 2 = signal R(A) = 0 V CC, signal R(B) = 5 V CC : ce réglage par défaut correspond à l'état inactif (pas d'opération d'émission active). Aucune détection de rupture n'est possible avec cet état initial.
BRK_DET	USInt	0	Détection de rupture Les sélections valides sont les suivantes : <ul style="list-style-type: none"> • 0 = détection de rupture désactivée • 1 = détection de rupture activée

Variable	Type de données	Valeur par défaut	Description
EN_DIAG_ALARM	Bool	0	Activer l'alarme de diagnostic : <ul style="list-style-type: none"> • 0 = non activée • 1 = activée
STOP_BITS	USInt	1	Nombre de bits d'arrêt : <ul style="list-style-type: none"> • 1 = 1 bit d'arrêt • 2 = 2 bits d'arrêt • 0, 3 à 255 = réservés

Remarque

Réglage requis pour l'utilisation de câbles PROFIBUS avec CM 1241 pour RS485

Tableau 13- 74 Codes d'erreur d'exécution de Modbus_Comm_Load ¹

STATUS (W#16#)	Description
0000	Pas d'erreur
8180	Valeur d'ID de port invalide (port/identificateur matériel erroné pour le module de communication)
8181	Valeur de vitesse de transmission invalide
8182	Valeur de parité invalide
8183	Valeur de contrôle de flux invalide
8184	Valeur invalide du délai de réponse (délai de réponse inférieur au minimum qui est de 5 ms)
8185	Le paramètre MB_DB n'est pas un bloc de données d'instance d'une instruction Modbus_Master ou Modbus_Slave.

¹ En plus des erreurs Modbus_Comm_Load présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

Instruction Modbus_Master (Communiquer à l'aide de SIPLUS I/O ou du port PtP en tant que maître Modbus RTU)

Tableau 13- 75 Instruction Modbus_Master

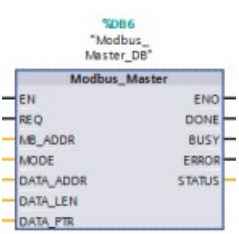
CONT/LOG	SCL	Description
	<pre>"Modbus_Master_DB" (REQ:=_bool_in_, MB_ADDR:=_uint_in_, MODE:=_usint_in_, DATA_ADDR:=_udint_in_, DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA_PTR:=_variant_inout_);</pre>	<p>L'instruction Modbus_Master permet la communication en tant que maître DB à l'aide d'un port configuré lors d'une exécution précédente de l'instruction Modbus_Comm_Load. Un bloc de données d'instance est automatiquement affecté lorsque vous insérez l'instruction Modbus_Master dans votre programme. Ce bloc de données d'instance Modbus_Master est utilisé lorsque vous définissez le paramètre MB_DB pour l'instruction Modbus_Comm_Load.</p>

Tableau 13- 76 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	0 = Pas de demande 1 = Demande d'envoi de données à un esclave Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Adresse de station Modbus RTU. Plage d'adressage standard (1 à 247) Plage d'adressage étendue (1 à 65535) La valeur 0 est réservée à la diffusion générale d'un message à tous les esclaves Modbus. Les codes de fonction Modbus 05, 06, 15 et 16 sont les seuls codes de fonction acceptés pour la diffusion générale.
MODE	IN	USInt	Sélection du mode : Indique le type de demande (lecture, écriture ou diagnostic). Voir le tableau des fonctions Modbus ci-après pour plus de détails.
DATA_ADDR	IN	UDInt	Adresse de début dans l'esclave. Indique l'adresse de début des données auxquelles accéder dans l'esclave Modbus. Voir les adresses valides dans le tableau des fonctions Modbus ci-après.
DATA_LEN	IN	UInt	Longueur des données. Indique le nombre de bits ou mots auxquels accéder dans cette demande. Voir les longueurs valides dans le tableau des fonctions Modbus ci-après.
DATA_PTR	IN_OUT	Variant	Pointeur de données. Désigne l'adresse M ou DB (type de DB non optimisé) pour les données en cours d'écriture ou de lecture.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 – Pas d'opération Modbus_Master en cours 1 – Opération Modbus_Master en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

Règles de communication Modbus_Master

- Modbus_Comm_Load doit être exécuté pour configurer un port avant qu'une instruction Modbus_Master ne puisse communiquer avec ce port.
- Si un port doit être utilisé pour déclencher des demandes du maître Modbus, il ne doit pas être utilisé par Modbus_Slave. Une ou plusieurs instances d'exécution de Modbus_Master peuvent être utilisées avec ce port, mais toutes les exécutions de Modbus_Master doivent utiliser le même DB d'instance Modbus_Master pour ce port.
- Les instructions Modbus n'utilisent pas d'événements d'alarme de communication pour piloter le processus de communication. Votre programme doit interroger les conditions émission et réception achevées de l'instruction Modbus_Master.
- Nous vous conseillons d'appeler toutes les exécutions de Modbus_Master pour un port donné dans un OB de cycle de programme. Les instructions Modbus_Master ne peuvent s'exécuter que dans un niveau d'exécution, cycle de programme ou alarme cyclique/temporisée. Elles ne peuvent pas s'exécuter dans les deux niveaux de priorité d'exécution. L'interruption d'une instruction Modbus_Master par une autre instruction Modbus_Master dans un niveau de priorité d'exécution supérieur entraîne un fonctionnement incorrect. Les instructions Modbus_Master ne doivent pas s'exécuter dans les niveaux de priorité mise en route, diagnostic ou erreur de temps.
- Une fois qu'une instruction Modbus_Master déclenche une transmission, cette instance doit s'exécuter en continu avec l'entrée EN activée jusqu'à ce qu'un état DONE égal à 1 ou qu'un état ERROR égal à 1 soit renvoyé. Une instance particulière de Modbus_Master est considérée comme active jusqu'à ce que l'un de ces deux événements se produise. Tant que l'instance initiale est active, tout appel à une autre instance avec l'entrée REQ activée entraîne une erreur. Si l'exécution continue de l'instance initiale est interrompue, l'état de la demande reste actif pendant une durée indiquée par la variable statique "Blocked_Proc_Timeout". A l'expiration de cette durée, l'instruction Modbus_Master suivante appelée avec l'entrée REQ activée devient l'instance active. Cela empêche une instance Modbus_Master unique de monopoliser un port ou d'en verrouiller l'accès. Si l'instance active initiale n'est pas activée pendant la durée indiquée par la variable statique "Blocked_Proc_Timeout", l'exécution suivante par cette instance (avec REQ non activé) annulera l'état actif. Si REQ est activé, cette exécution déclenche une nouvelle demande Modbus_Master comme si aucune autre instance n'était active.

Paramètre REQ

0 = Pas de demande ; 1 = Demande d'envoi de données à un esclave Modbus

Vous pouvez commander cette entrée à l'aide d'un contact déclenché sur niveau ou sur front. Dès que cette entrée est activée, un automate fini est démarré pour garantir qu'aucune autre instruction Modbus_Master utilisant le même DB d'instance n'a le droit d'émettre de demande, et ce jusqu'à ce que la demande en cours soit achevée. Tous les autres états d'entrée sont capturés et conservés en interne pendant la demande en cours jusqu'à ce que la réponse soit reçue ou qu'une erreur soit détectée.

Si la même instance de Modbus_Master est à nouveau exécutée avec l'entrée REQ à 1 avant l'achèvement de la demande en cours, il n'y aura pas d'émissions suivantes. Toutefois, dès que la demande aura été achevée, une nouvelle demande sera émise si Modbus_Master est à nouveau exécuté avec l'entrée REQ à 1.

Sélection du type de fonction Modbus via les paramètres DATA_ADDR et MODE

DATA_ADDR (adresse Modbus de début dans l'esclave) : Indique l'adresse de début des données auxquelles accéder dans l'esclave Modbus.

L'instruction Modbus_Master utilise une entrée MODE et non une entrée Code de fonction. La combinaison de MODE et de l'adresse Modbus déterminent le code de fonction qui est utilisé dans le message Modbus réel. Le tableau suivant montre la correspondance entre le paramètre MODE, le code de fonction Modbus et la plage d'adresses Modbus.

Tableau 13- 77 Fonctions Modbus

MODE	Fonction Modbus	Longueur des données	Opération et données	Adresse Modbus
0	01	1 à 2000 1 à 1992 ¹	Lecture de bits de sortie : 1 à 1992 (ou 2000) bits par demande	1 à 9999
0	02	1 à 2000 1 à 1992 ¹	Lecture de bits d'entrée : 1 à 1992 (ou 2000) bits par demande	10001 à 19999
0	03	1 à 125 1 à 124 ¹	Lecture de registres de maintien : 1 à 124 (ou 125) mots par demande	40001 à 49999 ou 400001 à 465535
0	04	1 à 125 1 à 124 ¹	Lecture de mots d'entrée : 1 à 124 (ou 125) mots par demande	30001 à 39999
104	04	1 à 125 1 à 124 ¹	Lecture de mots d'entrée : 1 à (124 ou 125) mots par demande	00000 à 65535
1	05	1	Ecriture d'un bit de sortie : Un bit par demande	1 à 9999
1	06	1	Ecriture d'un registre de maintien : 1 mot par demande	40001 à 49999 ou 400001 à 465535
1	15	2 à 1968 2 à 1960 ¹	Ecriture de plusieurs bits de sortie : 2 à 1960 (ou 1968) bits par demande	1 à 9999
1	16	2 à 123 2 à 122 ¹	Ecriture de plusieurs registres de maintien : 2 à 122 (ou 123) mots par demande	40001 à 49999 ou 400001 à 465535
2	15	1 à 1968 2 à 1960 ¹	Ecriture d'un ou de plusieurs bits de sortie : 1 à 1960 (ou 1968) bits par demande	1 à 9999
2	16	1 à 123 1 à 122 ¹	Ecriture d'un ou de plusieurs registres de maintien : 1 à 122 (ou 123) mots par demande	40001 à 49999 ou 400001 à 465535
11	11	0	Lecture du mot d'état de communication esclave et du compteur d'événements. Le mot d'état indique si l'esclave est occupé (0 : non occupé, 0xFFFF : occupé). Le compteur d'événements est incrémenté à chaque fois qu'un message s'achève avec succès. Les deux opérandes DATA_ADDR et DATA_LEN de l'instruction Modbus_Master ne sont pas pris en compte pour cette fonction.	
80	08	1	Contrôle de l'état de l'esclave à l'aide du code de diagnostic de données 0x0000 (test de retour : l'esclave renvoie la demande en écho) 1 mot par demande	

MODE	Fonction Modbus	Longueur des données	Opération et données	Adresse Modbus
81	08	1	Réinitialisation du compteur d'événements de l'esclave à l'aide du code de diagnostic de données 0x000A 1 mot par demande	
3 à 10, 12 à 79, 82 à 255			Réservé	

- ¹ En mode "Adressage étendu", les longueurs maximales de données sont réduites d'un octet ou d'un mot selon le type de données utilisé par la fonction.

Paramètre DATA_PTR

Le paramètre DATA_PTR pointe sur l'adresse M ou DB à lire ou dans laquelle écrire. Si vous utilisez un bloc de données, vous devez créer un bloc de données global qui fournit de la mémoire pour les lectures et écritures dans les esclaves Modbus.

Remarque

Le bloc de données DATA_PTR doit autoriser l'adressage direct

Le bloc de données doit autoriser à la fois l'adressage direct (absolu) et l'adressage symbolique. Vous devez donc sélectionner l'attribut d'accès "standard" lorsque vous créez le bloc de données.

À partir de la version V4.0 de l'instruction Modbus_Master, vous pouvez activer l'attribut de bloc de données "Accès au bloc optimisé". Vous pouvez utiliser un élément individuel ou un tableau d'éléments en mémoire optimisée avec les types de données suivants uniquement : Bool, Byte, Char, Word, Int, DWord, Dint, Real, USInt, UInt, UDIInt, SInt ou WChar.

Structures de bloc de données pour le paramètre DATA_PTR

- Ces types de données sont valides pour les **lectures de mot** des adresses Modbus 30001 à 39999, 40001 à 49999 et 400001 à 465536 et également pour les **écritures de mot** dans les adresses Modbus 40001 à 49999 et 400001 à 465536.
 - Tableau standard de types de données WORD, UINT ou INT
 - Structure WORD, UINT ou INT nommée dans laquelle chaque élément a un nom unique et un type de données de 16 bits
 - Structure complexe nommée dans laquelle chaque élément a un nom unique et un type de données de 16 ou 32 bits
- Pour les **lectures et écritures de bit** des adresses Modbus 00001 à 09999 et les lectures de bits des adresses 10001 à 19999
 - Tableau standard de types de données BOOL
 - Structure booléenne nommée de variables booléennes à nom unique.
- Bien que cela ne soit pas obligatoire, il est recommandé que chaque instruction Modbus_Master ait sa propre zone de mémoire distincte. La raison en est que le risque d'altération des données est plus important si plusieurs instructions Modbus_Master lisent et écrivent dans la même zone de mémoire.
- Il n'est pas obligatoire que les zones de données DATA_PTR soient dans le même bloc de données global. Vous pouvez créer un bloc de données à zones multiples pour les lectures Modbus, un bloc de données pour les écritures Modbus ou un bloc de données pour chaque station esclave.

Variables de bloc de données (DB) de l'instruction Modbus_Master

Le tableau suivant montre les variables statiques publiques sauvegardées dans le DB d'instance de Modbus_Master que vous pouvez utiliser dans votre programme :

Tableau 13- 78 Variables statiques du DB d'instance de Modbus_Master

Variable	Type de données	Valeur par défaut	Description
Blocked_Proc_Timeout	Real	3.0	Durée (en secondes) d'attente d'une instance Modbus_Master bloquée avant que l'état ACTIF de cette instance soit annulé. Cela peut, par exemple, se produire lorsqu'une demande Modbus_Master a été émise et que le programme arrête d'appeler la fonction Modbus_Master avant l'achèvement complet de la demande. La valeur de temps doit être supérieure à 0 et inférieure à 55 secondes ou une erreur se produit.
Extended_Addresssing	Bool	FALSE	Configure l'adressage d'esclave à un ou deux octets : <ul style="list-style-type: none"> • FALSE = adresse à un octet ; 0 à 247 • TRUE = adresse à deux octets (correspond à un adressage étendu) ; 0 à 65535
MB_DB	MB_BASE	-	Le paramètre MB_DB de l'instruction Modbus_Comm_Load doit être connecté au paramètre MB_DB de l'instruction Modbus_Master.

13.5 Communication Modbus

Votre programme peut écrire des valeurs dans les variables Blocked_Proc_Timeout et Extended_Addressing pour commander le fonctionnement du Modbus_Master. Vous trouverez dans la description de HR_Start_Offset (Page 1266) et Extended_Addressing (Page 1266) à la rubrique Modbus_Slave un exemple d'utilisation de ces variables dans l'éditeur de programmes ainsi que des détails sur l'adressage étendu Modbus.

Codes d'erreur

Tableau 13- 79 Codes d'erreur d'exécution de Modbus_Master (erreurs de communication et de configuration) ¹

STATUS (W#16#)	Description
0000	Pas d'erreur
80C8	Dépassement de délai chez l'esclave. L'esclave spécifié n'a pas répondu dans le délai spécifié. Veuillez vérifier la vitesse de transmission, la parité, et le câblage de l'esclave. Cette erreur n'est signalée qu'uniquement après la réalisation de nouvelles tentatives configurées.
80C9	L'instruction Modbus_Master a expiré pour l'une des raisons suivantes : <ul style="list-style-type: none"> • L'instruction est en attente de réponse de la part du module qui est utilisé pour les communications. • La valeur Blocked_Proc_Timeout est paramétrée comme trop petite. Cette erreur est signalée si un périphérique à périphérie décentralisée PROFIBUS ou PROFINET renvoie l'un des éléments suivants : <ul style="list-style-type: none"> • Une interruption de puissance ou de communication • Un événement de brochage/enfichage concernant un module de communication Dans ces circonstances, la configuration matérielle à partir de l'API est chargée à nouveau, et Modbus_Comm_Load doit être exécutée à nouveau pour configurer le module de communication de manière appropriée.
80D1	Le récepteur a émis une demande de contrôle de flux pour suspendre l'émission active et n'a jamais réactivé la transmission durant le temps d'attente indiqué. Cette erreur est également générée pendant le contrôle de flux matériel lorsque le récepteur n'affirme pas Prêt à émettre (CTS) pendant le temps d'attente indiqué.
80D2	La demande d'émission a été annulée, car aucun signal DSR (modem prêt) n'a été reçu de l'équipement de transmission de données (DCE).
80E0	Le message a été interrompu car la mémoire tampon de réception est pleine.
80E1	Le message a été interrompu en raison d'une erreur de parité.
80E2	Le message a été interrompu en raison d'une erreur de trame.
80E3	Le message a été interrompu en raison d'une erreur de débordement.
80E4	Le message a été interrompu, car la longueur indiquée dépasse la taille de mémoire tampon totale.
8180	Valeur d'ID de port invalide ou erreur avec l'instruction Modbus_Comm_Load
8186	Adresse de station Modbus invalide
8188	Mode invalide indiqué pour une demande de diffusion générale
8189	Valeur d'adresse de données invalide
818A	Valeur de longueur de données invalide
818B	Pointeur invalide vers la source/destination de données locale : taille incorrecte

STATUS (W#16#)	Description
818C	Pointeur invalide pour DATA_PTR ou Blocked_Proc_Timeout invalide. La zone de données doit être l'un des éléments suivants : <ul style="list-style-type: none"> • DB classique • Tableau de types de données simples dans un DB symbolique ou rémanent • Mémoire M
8200	Le port est occupé par le traitement d'une demande d'émission.
8280	Reconnaissance négative lors de la lecture du module. Vérifiez l'entrée au niveau du paramètre PORT. Cela peut être dû à la perte subie par un module de périphérie décentralisée PROFIBUS ou PROFINET, causée soit par une puissance de station, une perte de communication ou un débrogage de module.
8281	Reconnaissance négative lors de l'écriture au module. Vérifiez l'entrée au niveau du paramètre PORT. Cela peut être dû à la perte subie par un module de périphérie décentralisée PROFIBUS ou PROFINET, causée soit par une puissance de station, une perte de communication ou un débrogage de module.

Tableau 13- 80 Codes d'erreur d'exécution de Modbus_Master (erreurs de protocole Modbus) ¹

STATUS (W#16#)	Code de réponse de l'esclave	Erreurs de protocole Modbus
8380	-	Erreur CRC
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou adresse en dehors de la plage valide de la zone DATA_PTR
8384	Supérieur à 03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non prise en charge (code de fonction 08)
8386	-	Le code de fonction dans la réponse ne correspond pas au code dans la demande.
8387	-	Ce n'est pas le bon esclave qui a répondu.
8388	-	La réponse de l'esclave à une demande d'écriture est incorrecte. La demande d'écriture renvoyée par l'esclave ne correspond pas à ce que le maître a effectivement envoyé.

¹ En plus des erreurs Modbus_Master présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

Remarque

Paramétrage de la longueur d'enregistrement maximum pour la communication Profibus

Lorsque vous utilisez un module maître Profibus CM1243-5 pour piloter un appareil Profibus ET 200SP ou ET 200MP qui utilise un module point à point RS232, RS422 ou RS485, vous devez paramétrer explicitement la variable de bloc de données "max_record_len" sur 240 comme défini ci-dessous :

Paramétrer "max_record_len" dans la partie Send_P2P du DB d'instance (par exemple, "Modbus_Master_DB".Send_P2P.max_record_len) sur 240 après avoir exécuté Modbus_Comm_Load.

Paramétrer explicitement max_record_len est seulement nécessaire avec la communication Profibus ; la communication Profinet utilise déjà une valeur max_record_len valide.

Instruction Modbus_Slave (Communiquer à l'aide de SIPLUS I/O ou du port PtP en tant qu'esclave Modbus RTU)

Tableau 13- 81 Instruction Modbus_Slave


CONT/LOG	SCL	Description
	<pre>"Modbus_Slave_DB" (MB_ADDR:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	<p>L'instruction Modbus_Slave permet à votre programme de communiquer de l'une des deux manières suivantes :</p> <ul style="list-style-type: none"> • Comme un esclave Modbus RTU à travers un port PtP sur le CM (RS485 ou RS232) et le CB (RS485) • Comme un esclave Modbus RTU à travers un port PtP à travers des options SIPLUS I/O Modbus RTU : <ul style="list-style-type: none"> – Installez ET 200MP S7-1500CM PtP (RS485 / 422 ou RS232). – Installez ET 200SP S7-1500 CM PtP (RS485 / 422 ou RS232). <p>Lorsqu'un maître RTU Modbus distant émet une demande, votre programme utilisateur répond par l'exécution de Modbus_Slave. STEP 7 crée automatiquement un DB d'instance lorsque vous insérez l'instruction. Utilisez ce nom Modbus_Slave_DB lorsque vous indiquez le paramètre MB_DB pour l'instruction Modbus_Comm_Load.</p>

Tableau 13- 82 Types de données pour les paramètres

Paramètre et type		Type de données	Description
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Adresse de station de l'esclave Modbus : Plage d'adressage standard (1 à 247) Plage d'adressage étendue (0 à 65535)
MB_HOLD_REG	IN_OUT	Variante	Pointeur désignant le DB de registre de maintien Modbus : Le registre de maintien Modbus peut être un memento ou un bloc de données.
NDR	OUT	Bool	Nouvelles données prêtes (New Data Ready) : <ul style="list-style-type: none"> 0 = Pas de nouvelles données 1 = Indique que des nouvelles données ont été écrites par le maître Modbus.
DR	OUT	Bool	Données lues : <ul style="list-style-type: none"> 0 = Pas de données lues 1 = Indique que des données ont été lues par le maître Modbus.
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. Si l'exécution s'achève avec une erreur, la valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

Les fonctions de communication Modbus de codes 1, 2, 4, 5 et 15 peuvent lire et écrire des bits et des mots directement dans la mémoire image des entrées et la mémoire image des sorties de la CPU. Pour ces codes de fonctions, le paramètre MB_HOLD_REG doit être défini avec un type de données supérieur à un octet. Le tableau suivant montre en exemple la correspondance entre les adresses Modbus et la mémoire image dans la CPU.

Tableau 13- 83 Correspondance entre adresses Modbus et mémoire image

Fonctions Modbus					S7-1200		
Codes	Fonction	Zone de données	Plage d'adresses			Zone de données	Adresse de la CPU
01	Lecture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
02	Lecture de bits	Entrée	10001	à	18192	Mémoire image des entrées	I0.0 à I1023.7
04	Lecture de mots	Entrée	30001	à	30512	Mémoire image des entrées	IW0 à IW1022
05	Ecriture d'un bit	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
15	Ecriture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7

13.5 Communication Modbus

Les fonctions de communication Modbus de codes 3, 6 et 16 utilisent un registre de maintien Modbus qui peut être une zone d'adresse en mémoire M ou un bloc de données. Le type de registre de maintien est indiqué par le paramètre MB_HOLD_REG dans l'instruction Modbus_Slave.

Remarque

Type de bloc de données MB_HOLD_REG

Un bloc de données faisant office de registre de maintien Modbus doit autoriser à la fois l'adressage direct (absolu) et l'adressage symbolique. Vous devez donc sélectionner l'attribut d'accès "standard" lorsque vous créez le bloc de données.

À partir de la version V4.0 de l'instruction Modbus_Slave, vous pouvez activer l'attribut de bloc de données "Accès au bloc optimisé". Vous pouvez utiliser un élément individuel ou un tableau d'éléments en mémoire optimisée avec les types de données suivants uniquement : Bool, Byte, Char, Word, Int, DWord, Dint, Real, USInt, UInt, UDim, SInt ou WChar.

Le tableau suivant montre des exemples de correspondance entre adresses Modbus et registre de maintien pour les codes de fonction Modbus 03 (lecture de mots), 06 (écriture d'un mot) et 16 (écriture de mots). La limite supérieure effective des adresses de DB dépend des limites de mémoire de travail et de mémoire M maximum pour chaque modèle de CPU.

Tableau 13- 84 Correspondance entre adresses Modbus et mémoire CPU

Adresse du maître Modbus	Exemples de paramètre MB_HOLD_REG				
	MW100	DB10.DBw0	MW120	DB10.DBW50	"Recette".ingredient
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recette".ingredient[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Recette".ingredient[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Recette".ingredient[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Recette".ingredient[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Recette".ingredient[5]

Tableau 13- 85 Fonctions de diagnostic

Fonctions de diagnostic Modbus Modbus_Slave du S7-1200		
Codes	Sous-fonction	Description
08	0000H	Renvoi d'un test d'écho des données de requête : <ul style="list-style-type: none"> • Avant STEP 7 V15.1, Modbus_Slave renvoie en écho à un maître Modbus un mot de données reçu. • À partir de STEP 7 V15.1, l'instruction Modbus_Slave de version V4.1 ou ultérieure renvoie en écho un ou plusieurs mots de données reçus.
08	000AH	Effacement du compteur d'événements de communication : Le Modbus_Slave efface le compteur d'événements de communication qui est utilisé pour la fonction Modbus 11.
11		Lecture du compteur d'événements de communication : Le Modbus_Slave utilise un compteur d'événements de communication interne pour enregistrer le nombre de demandes de lecture et d'écriture Modbus envoyées au Modbus_Slave qui ont abouti. Le compteur ne s'incrémente pas pour les fonctions 8 et 11 ni pour les demandes à diffusion générale. Il ne s'incrémente pas non plus pour toute demande entraînant une erreur de communication (erreurs de parité ou de CRC, par exemple).

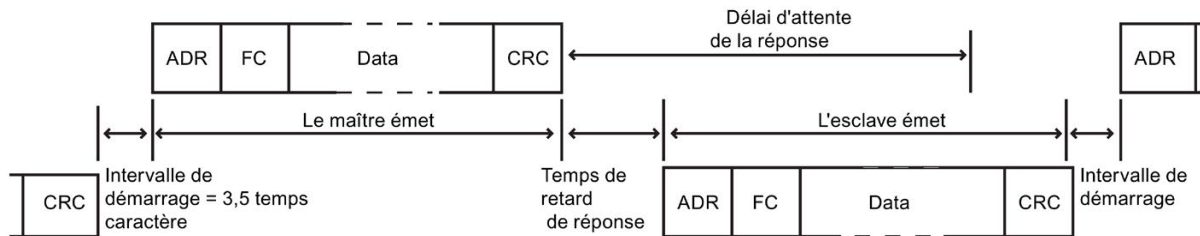
L'instruction Modbus_Slave accepte les demandes d'écriture à diffusion générale provenant de n'importe quel maître Modbus à condition que la requête demande l'accès à des adresses valides. Modbus_Slave génère un code d'erreur 0x8188 pour les codes de fonction non pris en charge en diffusion générale.

Règles de communication Modbus_Slave

- Modbus_Comm_Load doit être exécuté pour configurer un port avant qu'une instruction Modbus_Slave ne puisse communiquer par le biais de ce port.
- Si un port doit répondre comme esclave à un Modbus_Master, ne programmez pas ce port avec l'instruction Modbus_Master.
- Une seule instance de Modbus_Slave peut être utilisée avec un port donné, faute de quoi le comportement pourrait devenir imprévisible.
- Les instructions Modbus n'utilisent pas d'événements d'alarme de communication pour piloter le processus de communication. Votre programme doit piloter le processus de communication en interrogeant les conditions émission et réception achevées de l'instruction Modbus_Slave.
- L'instruction Modbus_Slave doit s'exécuter périodiquement à une fréquence permettant de répondre opportunément à des demandes entrantes provenant d'un Modbus_Master. Il est recommandé d'exécuter Modbus_Slave à chaque cycle dans un OB de cycle de programme. Il est possible d'exécuter Modbus_Slave dans un OB d'alarme cyclique, mais ce n'est pas recommandé en raison du risque que des retards excessifs dans le programme d'alarme bloquent temporairement l'exécution d'autres programmes d'alarme.

Synchronisation des signaux Modbus

Modbus_Slave doit être exécuté périodiquement pour recevoir chaque demande provenant du Modbus_Master et pour y répondre de manière appropriée. La fréquence d'exécution de Modbus_Slave dépend du délai d'attente de la réponse dans le Modbus_Master. Ceci est illustré dans le schéma suivant.



Le délai d'attente de la réponse RESP_TO est la durée pendant laquelle un Modbus_Master attend le début d'une réponse d'un Modbus_Slave. Ce temps n'est pas défini par le protocole Modbus mais constitue un paramètre de chaque Modbus_Master. La fréquence d'exécution (intervalle de temps entre une exécution et la suivante) de Modbus_Slave doit se baser sur les paramètres spécifiques de votre Modbus_Master. Vous devriez exécuter Modbus_Slave au minimum deux fois pendant le délai d'attente de la réponse du Modbus_Master.

Variables de bloc de données (DB) de l'instruction Modbus_Slave

Le tableau suivant montre les variables statiques publiques sauvegardées dans le DB d'instance de Modbus_Slave que vous pouvez utiliser dans votre programme :

Tableau 13- 86 Variables statiques du DB d'instance de Modbus_Slave

Variable	Type de données	Valeur par défaut	Description
HR_Start_Offset	Word	0	Définit l'adresse de début du registre de maintien Modbus (valeur par défaut = 0).
Extended_Addresssing	Bool	FALSE	Configure l'adressage d'esclave à un ou deux octets : <ul style="list-style-type: none"> • FALSE = adresse à un octet • TRUE = adresse à deux octets
Request_Count	Word	0	Total de l'ensemble des demandes reçues par cet esclave
Slave_Message_Count	Word	0	Nombre de demandes reçues pour cet esclave spécifique
Bad_CRC_Count	Word	0	Nombre de demandes reçues comportant une erreur CRC
Broadcast_Count	Word	0	Nombre de demandes de diffusion générale reçues
Exception_Count	Word	0	Erreurs spécifiques à Modbus nécessitant un acquittement avec le renvoi d'une exception au maître
Success_Count	Word	0	Nombre de demandes reçues pour cet esclave spécifique ne contenant pas d'erreurs de protocole
MB_DB	MB_BASE	-	Le paramètre MB_DB de l'instruction Modbus_Comm_Load doit être connecté au paramètre MB_DB de l'instruction Modbus_Slave.
QB_Start	UInt	0	Adresse de début des octets de sortie dans lesquels la CPU peut écrire (QB0 à QB65535)
QB_Count	UInt	65535	Nombre d'octets dans lesquels un appareil distant peut écrire. Si QB_Count est égal à 0, un appareil distant ne peut pas écrire dans les sorties. Exemple : pour que l'écriture soit autorisée uniquement dans les octets QB10 à QB17, QB_Start = 10 et QB_Count = 8.
QB_Read_Start	UInt	0	Adresse de début des octets de sortie que la CPU peut lire (QB0 à QB65535)
QB_Read_Count	UInt	65535	Nombre d'octets de sortie qu'un appareil distant peut lire. Si QB_Count est égal à 0, un appareil distant ne peut pas lire dans les sorties. Exemple : pour que la lecture soit autorisée uniquement dans les octets QB10 à QB17, QB_Start = 10 et QB_Count = 8.
IB_Read_Start	UInt	0	Adresse de début des octets d'entrée que la CPU peut lire (IB0 à IB65535)
IB_Read_Count	UInt	65535	Nombre d'octets d'entrée qu'un appareil distant peut lire. Si IB_Count est égal à 0, un appareil distant ne peut pas lire dans les entrées. Exemple : pour que la lecture soit autorisée uniquement dans les octets IB10 à IB17, IB_Start = 10 et IB_Count = 8.

13.5 Communication Modbus

Votre programme peut écrire des valeurs dans les variables HR_Start_Offset et Extended_Addressing pour commander le fonctionnement de l'esclave Modbus. Vous pouvez lire les autres variables pour surveiller l'état de Modbus.

Les versions suivantes sont requises pour la disponibilité des variables de bloc de données (DB) de l'instruction Modbus_Slave :

Tableau 13- 87 Versions requises pour la disponibilité des variables de DB de l'instruction Modbus_Slave : instruction, TIA Portal et CPU S7-1200

Version de l'instruction Modbus_Slave	Version de TIA Portal	Version du firmware (FW) de la CPU S7-1200	Variables du bloc de données
3.0	V14 SP1	CPU FW V4.0 ou plus	QB_Start
			QB_Count
4.0	V15.1	CPU FW V4.2 ou plus	QB_Start
			QB_Count
			QB_Read_Start
			QB_Read_Count
			IB_Read_Start
			IB_Read_Count

HR_Start_Offset

Les adresses de registre de maintien Modbus commencent à 40001 ou 400001. Cela correspond à l'adresse de début du registre de maintien dans la mémoire API. Vous pouvez toutefois configurer la variable "HR_Start_Offset" afin que le registre de maintien Modbus commence à une valeur autre que 40001 ou 400001.

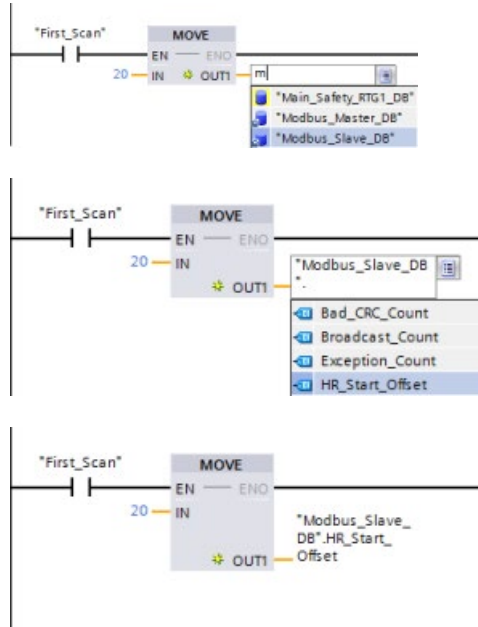
Par exemple, si le registre de maintien est configuré pour commencer à MW100 et s'il a une longueur de 100 mots. Un décalage de 20 spécifie une adresse de début de registre située à 40021 au lieu de 40001. Toute adresse inférieure à 40021 et supérieure à 400119 entraînera une erreur d'adressage.

Tableau 13- 88 Exemple d'adressage de registre de maintien Modbus

HR_Start_Offset	Adresse	Minimum	Maximum
0	Adresse Modbus (mot)	40001	40099
	Adresse S7-1200	MW100	MW298
20	Adresse Modbus (mot)	40021	40119
	Adresse S7-1200	MW100	MW298

HR_Start_Offset est une valeur de mot qui indique l'adresse de début du registre de maintien Modbus et est sauvegardée dans le bloc de données d'instance de Modbus_Slave. Vous pouvez définir la valeur de cette variable statique publique à l'aide de la liste déroulante d'aide pour les paramètres lorsque vous avez inséré Modbus_Slave dans votre programme.

Par exemple, une fois Modbus_Slave inséré dans un réseau CONT, vous pouvez aller à un réseau précédent et définir la valeur HR_Start_Offset. La valeur doit être affectée avant l'exécution de Modbus_Slave.



Entrée d'une variable pour l'esclave Modbus à l'aide du nom de DB par défaut :

1. Placez le curseur dans le champ de paramètre et tapez la lettre m.
2. Sélectionnez "Modbus_Slave_DB" dans la liste déroulante.
3. Placez le curseur à droite du nom de DB (après le guillemet) et entrez un point.
4. Sélectionnez "Modbus_Slave_DB.HR_Start_Offset" dans la liste déroulante.

Extended Addressing

Vous accédez à la variable Extended Addressing de la même manière qu'à la variable HR_Start_Offset décrite ci-avant, si ce n'est que Extended Addressing est une valeur booléenne. Une valeur booléenne doit être écrite par une bobine de sortie et non par une boîte MOVE.

L'adressage d'esclave Modbus peut être configuré à un octet (la norme pour Modbus) ou à deux octets. On utilise l'adressage étendu pour accéder à plus de 247 unités dans un réseau unique. La sélection de l'adressage étendu vous permet d'accéder à 64000 adresses au maximum. Une trame pour la fonction Modbus 1 est présentée ci-dessous comme exemple.

Tableau 13- 89 Adresse d'esclave à un octet (octet 0)

Fonction 1	Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	
Demande	Adr. es-clave	Code F	Adresse de début		Longueur des bobines		
Réponse valide	Adr. es-clave	Code F	Longueur	Données de bobine			
Réponse erronée	Adr. es-clave	0x81	Code E				

Tableau 13- 90 Adresse d'esclave à deux octets (octet 0 et octet 1)

	Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6
Demande	Adresse d'esclave		Code F	Adresse de début		Longueur des bobines	
Réponse valide	Adresse d'esclave		Code F	Longueur	Données de bobine		
Réponse erro- née	Adresse d'esclave		0x81	Code E			

Accès aux zones de données des blocs de données au lieu de l'accès direct aux adresses Modbus

À partir de la version V4.0 de l'instruction Modbus_Slave et de la version V4.2 du firmware de la CPU S7-1200, vous pouvez accéder à des zones de données dans les DB au lieu d'accéder directement aux mémoires images et aux registres de maintien. Pour ce faire, vous devez décocher les cases des options "Sauvegarder uniquement dans la mémoire de chargement" et "Accès au bloc optimisé" dans la page des propriétés "Attributs" du DB global.

Si, à l'arrivée d'une demande Modbus, vous n'avez pas défini de zone de données pour le type de données Modbus du code de fonction correspondant, la CPU traite la demande comme dans les versions précédentes de l'instruction : vous accédez directement aux mémoires images et aux registres de maintien.

Si vous avez défini une zone de données pour le type de données Modbus du code de fonction, l'instruction Modbus_Slave lit ou écrit dans cette zone de données. C'est le type de tâche qui détermine s'il s'agit d'une opération de lecture ou d'écriture.

Une demande Modbus individuelle permet de lire ou d'écrire dans une zone de données uniquement. Ainsi, si vous voulez lire des registres de maintien qui s'étendent sur plusieurs zones de données, vous devez exécuter plusieurs demandes Modbus.

Voici les règles pour la définition des zones de données :

- Vous pouvez définir huit zones de données au maximum dans différents DB ; chaque DB ne doit contenir qu'une seule zone de données. Une demande Modbus individuelle ne peut lire ou écrire que dans une zone de données exactement. Chaque zone de données correspond à une zone d'adresses Modbus. Vous définissez les zones de données dans la variable statique "Data_Area_Array" du DB d'instance.
- Si vous voulez utiliser moins de huit zones de données, vous devez placer les zones de données requises l'une derrière l'autre sans intervalle entre elles. La première entrée vide dans les zones de données met fin à la recherche de zone de données pendant le traitement. Si, par exemple, vous définissez les éléments de tableau 1, 2, 4 et 5, "Data_Area_Array" reconnaît uniquement les éléments 1 et 2 puisque l'élément 3 est vide.
- Le tableau Data_Area_Array est constitué de huit éléments : Data_Area_Array[1] à Data_Area_Array[8].
- Chaque élément Data_Area_Array[x], 1 <= x <= 8, est un UDT de type MB_DataArea structuré de la manière suivante :

Paramètre	Type de données	Signification
data_type	UInt	<p>Identifiant pour le type de données Modbus qui est mappé sur cette zone de données :</p> <ul style="list-style-type: none"> 0 : identifiant pour un élément de tableau vide ou une zone de données inutilisée. Les valeurs de "db", "start" et "length" sont alors sans objet. 1 : sortie dans la mémoire image (utilisation avec les codes de fonction 1, 5 et 15) 2 : entrée dans la mémoire image (utilisation avec le code de fonction 2) 3 : registre de maintien (utilisation avec les codes de fonction 3, 6 et 16) 4 : registre d'entrée (utilisation avec le code de fonction 4) <p>Remarque : Si vous avez défini une zone de données pour un type de données Modbus, l'instruction Modbus_Slave ne peut plus accéder directement à ce type de données Modbus. Si l'adresse d'une demande Modbus pour un tel type de données ne correspond pas à une zone de données définie, la valeur W#16#8383 est renvoyée dans le paramètre STATUS.</p>
db	UInt	<p>Numéro du bloc de données dans lequel le registre ou les bits Modbus définis par la suite sont mappés.</p> <p>Le numéro de DB doit être unique dans les zones de données. Le même numéro de DB ne doit pas être défini dans plusieurs zones de données.</p> <p>Vous devez décocher les cases des options "Sauvegarder uniquement dans la mémoire de chargement" et "Accès au bloc optimisé" dans la page des propriétés "Attributs" du DB global.</p> <p>Les zones de données commencent également à l'adresse d'octet 0 du DB.</p> <p>Valeurs autorisées : 1 à 60999</p>
start	UInt	<p>Première adresse Modbus mappée sur le bloc de données à partir de l'adresse 0.0</p> <p>Valeurs autorisées : 0 à 65535</p>
length	UInt	<p>Nombre de bits (pour les valeurs 1 et 2 de "data_type") ou nombre de registres (pour les valeurs 3 et 4 de "data_type")</p> <p>Les zones d'adresses Modbus d'un type de données Modbus précis ne doivent pas se chevaucher.</p> <p>Valeurs autorisées : 1 à 65535</p>

- Premier exemple : data_type = 3, db = 1, start = 10, length = 6

La CPU mappe les registres de maintien (data_type = 3) dans le bloc de données 1 (db = 1), plaçant l'adresse Modbus 10 (start = 10) dans le mot de données 0 et la dernière adresse Modbus valide 15 (length = 6) dans le mot de données 5.

- Deuxième exemple : data_type = 2, db = 15, start = 1700, length = 112

La CPU mappe les entrées (data_type = 2) dans le bloc de données 15 (db = 15), plaçant l'adresse Modbus 1700 (start = 1700) dans le mot de données 0 et la dernière adresse Modbus valide 1811 (length = 112) dans le mot de données 111.

Codes d'erreur

Tableau 13- 91 Codes d'erreur d'exécution de Modbus_Slave (erreurs de communication et de configuration) ¹

STATUS (W#16#)	Description
80D1	Le récepteur a émis une demande de contrôle de flux pour suspendre l'émission active et n'a jamais réactivé la transmission durant le temps d'attente indiqué. Cette erreur est également générée pendant le contrôle de flux matériel lorsque le récepteur n'affirme pas Prêt à émettre (CTS) pendant le temps d'attente indiqué.
80D2	La demande d'émission a été annulée, car aucun signal DSR (modem prêt) n'a été reçu de l'équipement de transmission de données (DCE).
80E0	Le message a été interrompu car la mémoire tampon de réception est pleine.
80E1	Le message a été interrompu en raison d'une erreur de parité.
80E2	Le message a été interrompu en raison d'une erreur de trame.
80E3	Le message a été interrompu en raison d'une erreur de débordement.
80E4	Le message a été interrompu, car la longueur indiquée dépasse la taille de mémoire tampon totale.
8180	Valeur d'ID de port invalide ou erreur avec l'instruction Modbus_Comm_Load
8186	Adresse de station Modbus invalide
8187	Pointeur invalide vers le DB MB_HOLD_REG : La zone est trop petite.
818C	Pointeur MB_HOLD_REG invalide. La zone de données doit être l'un des éléments suivants : <ul style="list-style-type: none"> • DB classique • Tableau de types de données simples dans un DB symbolique ou rémanent • Mémoire M

Tableau 13- 92 Codes d'erreur d'exécution de Modbus_Slave (erreurs de protocole Modbus) ¹

STATUS (W#16#)	Code de réponse de l'es-clave	Erreurs de protocole Modbus
8380	Pas de réponse	Erreur CRC
8381	01	Code de fonction non pris en charge ou non pris en charge dans des messages à diffusion générale
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou adresse en dehors de la plage valide de la zone DATA_PTR
8384	03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non prise en charge (code de fonction 08)

¹ En plus des erreurs Modbus_Slave présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

Remarque

Paramétrage de la longueur d'enregistrement maximum pour la communication PROFIBUS

Lorsque vous utilisez un module maître PROFIBUS CM1243-5 pour piloter un appareil PROFIBUS ET 200SP ou ET 200MP qui utilise un module point à point RS232, RS422 ou RS485, vous devez paramétrer explicitement la variable de bloc de données "max_record_len" à 240 comme défini ci-dessous :

Paramétrer "max_record_len" dans la partie Send_P2P du DB d'instance (par exemple, "Modbus_Slave_DB".Send_P2P.max_record_len) sur 240 après avoir exécuté Modbus_Comm_Load.

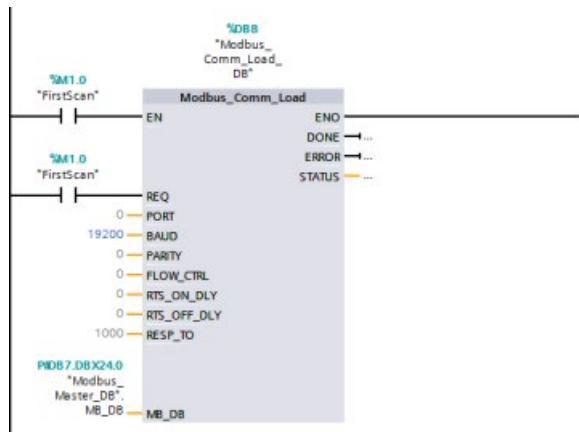
Paramétrer explicitement max_record_len est seulement nécessaire avec la communication PROFIBUS ; la communication PROFINET utilise déjà une valeur max_record_len valide.

13.5.3.5 Exemples Modbus RTU

Exemple : Programme maître Modbus RTU

Modbus_Comm_Load est initialisé au démarrage à l'aide du memento Premier cycle. Cette manière d'exécuter Modbus_Comm_Load ne doit être utilisée que lorsque la configuration du port série n'est pas modifiée pendant l'exécution.

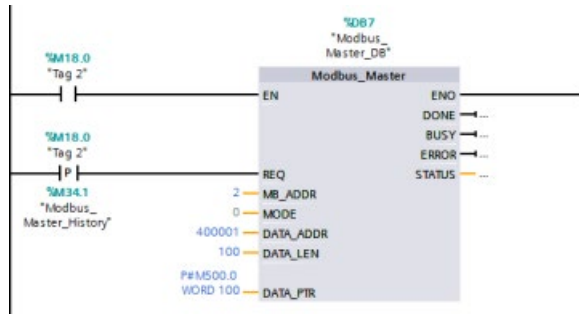
Réseau 1 : Configurer/initialiser le port de communication du module RS485 une seule fois pendant le premier cycle.



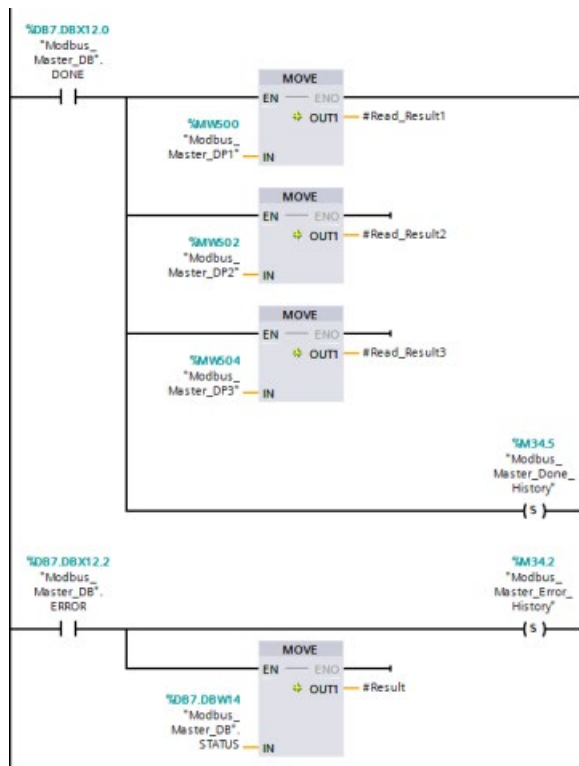
Une instruction Modbus_Master est utilisée dans l'OB de cycle de programme pour communiquer avec un esclave unique. Il est possible d'utiliser d'autres instructions Modbus_Master dans l'OB de cycle de programme pour communiquer avec d'autres esclaves ou de réutiliser un FB Modbus_Master pour communiquer avec des esclaves supplémentaires.

13.5 Communication Modbus

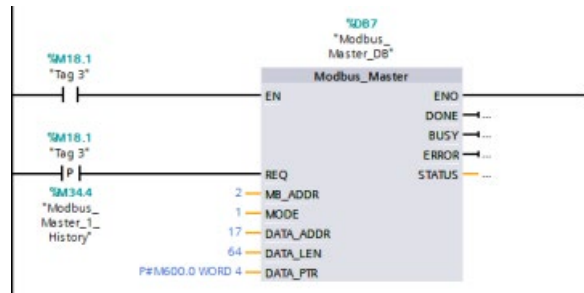
Réseau 2 : Lire 100 mots de données de registre de maintien de l'adresse 400001 sur l'esclave 2 vers l'adresse de mémoire MW500-MW698.



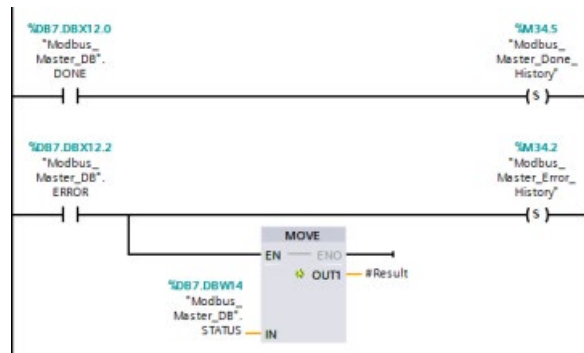
Réseau 3 : Déplacer les 3 premiers mots des données de registre de maintien lues à un autre emplacement et mettre un bit d'historique DONE à 1. Ce réseau met également à 1 un bit d'historique ERROR et sauvegarde le mot STATUS à un autre emplacement en cas d'erreur.



Réseau 4 : Écrire 64 bits de données de MW600-MW607 dans les adresses de bits de sortie 00017 à 00081 sur l'esclave 2.



Réseau 5 : Mettre un bit d'historique DONE à 1 lorsque l'écriture est achevée. En cas d'erreur, le programme met à 1 un bit d'historique ERROR et sauvegarde le code STATUS.

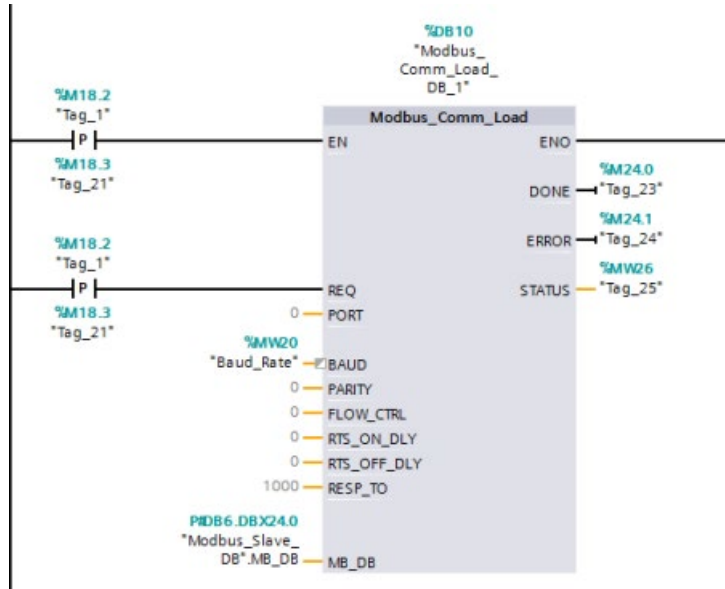


Exemple : Programme esclave Modbus RTU

MB_COMM_LOAD présenté ci-dessous est initialisé à chaque fois que "Tag_1" est activé.

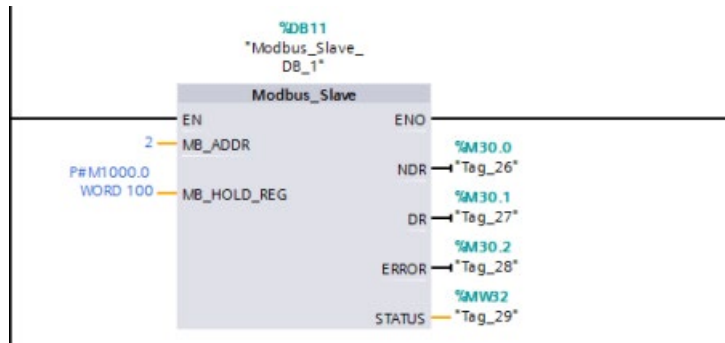
Cette manière d'exécuter MB_COMM_LOAD ne doit être utilisée que lorsque la configuration du port série est modifiée à l'exécution, en raison d'une configuration IHM.

Réseau 1 : Initialiser les paramètres de module RS485 à chaque fois qu'ils sont modifiés par un appareil IHM.



L'instruction MB_SLAVE montrée ci-dessous est insérée dans un OB cyclique qui est exécuté toutes les 10 ms. Cela n'assure certes pas la réponse la plus rapide possible de l'esclave, mais cela garantit de bonnes performances à 9600 bauds pour des messages courts (20 octets au plus dans la demande).

Réseau 2 : Vérifier la présence de demandes maître Modbus à chaque cycle. Le registre de maintien Modbus est configuré pour 100 mots en commençant à MW1000.



13.6 Communication PtP d'héritage (CM/CB 1241 uniquement)

Avant la sortie de STEP 7 V13 SP1 et des CPU S7-1200 V4.1, les instructions de communication point à point existaient avec des noms différents et, dans certains cas, avec des interfaces légèrement différentes. Les concepts généraux concernant la communication point à point (Page 1132), ainsi que le port (Page 1137) et la configuration des paramètres (Page 1155), s'appliquent aux deux jeux d'instructions. Reportez-vous aux instructions point à point d'héritage individuelles pour obtenir des informations relatives à la programmation.


Tableau 13- 93 Classes d'erreurs communes

Description de la classe	Classes d'erreurs	Description
Configuration du port	80Ax	Utilisé pour définir les erreurs de configuration de port communes
Configuration d'émission	80Bx	Utilisé pour définir les erreurs de configuration d'émission communes
Configuration de réception	80Cx	Utilisé pour définir les erreurs de configuration de réception communes
Exécution de l'émission	80Dx	Utilisé pour définir les erreurs communes apparaissant à l'exécution de l'émission
Exécution de la réception	80Ex	Utilisé pour définir les erreurs communes apparaissant à l'exécution de la réception
Gestion des signaux	80Fx	Utilisé pour définir les erreurs communes associées à toute la gestion des signaux

13.6.1 Instructions d'héritage point à point

13.6.1.1 PORT_CFG (Configurer des paramètres de communication dynamiquement)

Tableau 13- 94 Instruction PORT_CFG (Configuration de port)

CONT/LOG	SCL	Description
<p>"PORT_CFG_ DB"</p> 	<pre>"PORT_CFG_DB" (REQ:=_bool_in_, PORT:=_uint_in_, PROTOCOL:=_uint_in_, BAUD:=_uint_in_, PARITY:=_uint_in_, DATABITS:=_uint_in_, STOPBITS:=_uint_in_, FLOWCTRL:=_uint_in_, XONCHAR:=_char_in_, XOFFCHAR:=_char_in_, WAITTIME:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	<p>PORT_CFG vous permet de modifier les paramètres du port, tels que la vitesse de transmission, à partir de votre programme.</p> <p>Vous pouvez définir la configuration statique initiale du port dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction PORT_CFG dans votre programme pour modifier la configuration.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Les modifications de configuration de PORT_CFG ne sont pas stockées de façon permanente dans la CPU. Les paramètres configurés dans la configuration des appareils sont restaurés lorsque la CPU passe de l'état MARCHE à l'état ARRET et après une mise hors tension puis sous tension. Reportez-vous à Configuration des ports de communication (Page 1137) et à Gestion du contrôle de flux (Page 1139) pour plus d'informations.

Tableau 13- 95 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
PROTOCOL	IN	UInt	0 : protocole de communication point à point (valeur par défaut) 1..n : définition future de protocoles spécifiques
BAUD	IN	UInt	Vitesse de transmission du port (valeur par défaut : 6) : 1 = 300 bauds, 2 = 600 bauds, 3 = 1200 bauds, 4 = 2400 bauds, 5 = 4800 bauds, 6 = 9600 bauds, 7 = 19200 bauds, 8 = 38400 bauds, 9 = 57600 bauds, 10 = 76800 bauds, 11 = 115200 bauds
PARITY	IN	UInt	Parité du port (valeur par défaut : 1) : 1 = Sans parité, 2 = Parité paire, 3 = Parité impaire, 4 = Parité marque, 5 = Parité espace
DATABITS	IN	UInt	Bits par caractère (valeur par défaut : 1) : 1 = 8 bits de données, 2 = 7 bits de données
STOPBITS	IN	UInt	Bits d'arrêt (valeur par défaut : 1) : 1 = 1 bit d'arrêt, 2 = 2 bits d'arrêt
FLOWCTRL	IN	UInt	Contrôle du flux (valeur par défaut : 1) : 1 = Pas de contrôle de flux, 2 = XON/XOFF, 3 = RTS matériel toujours activé, 4 = RTS matériel commuté
XONCHAR	IN	Char	Indique le caractère qui est utilisé comme caractère XON. Ce sera typiquement le caractère DC1 (16#11). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 16#11)
XOFFCHAR	IN	Char	Indique le caractère qui est utilisé comme caractère XOFF. Ce sera typiquement le caractère DC3 (16#3). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 16#13)
XWAITIME	IN	UInt	Indique combien de temps attendre un caractère XON après avoir reçu un caractère XOFF ou combien de temps attendre le signal CTS après avoir activé RTS (0 à 65535 ms). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 2000)
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)


13.6 Communication PtP d'héritage (CM/CB 1241 uniquement)

Tableau 13- 96 Codes d'erreur

STATUS (W#16#....)	Description
80A0	Le protocole spécifique n'existe pas.
80A1	La vitesse de transmission spécifique n'existe pas.
80A2	La parité spécifique n'existe pas.
80A3	Le nombre spécifique de bits de données n'existe pas.
80A4	Le nombre spécifique de bits d'arrêt n'existe pas.
80A5	Le type spécifique de contrôle de flux n'existe pas.
80A6	Le temps d'attente est 0 et le contrôle de flux est activé.
80A7	XON et XOFF sont des valeurs illicites (par exemple, la même valeur).

13.6.1.2 SEND_CFG (Configurer dynamiquement les paramètres de transmission série)

Tableau 13- 97 Instruction SEND_CFG (Configuration d'émission)

CONT/LOG	SCL	Description
	<pre>"SEND_CFG_DB" (REQ:=_bool_in_, PORT:=_uint_in_, RTSONDLY:=_uint_in_, RTSOFFDLY:=_uint_in_, BREAK:=_uint_in_, IDLELINE:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	<p>SEND_CFG permet la configuration dynamique de paramètres d'émission série pour un port de communication point à point. Tout message mis en file d'attente à l'intérieur d'un CM ou d'un CB est rejeté lorsque SEND_CFG est exécuté.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Vous pouvez définir la configuration statique initiale du port dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction SEND_CFG dans votre programme pour modifier la configuration.

Les modifications de configuration de SEND_CFG ne sont pas stockées de façon permanente dans la CPU. Les paramètres configurés dans la configuration des appareils sont restaurés lorsque la CPU passe de l'état MARCHE à l'état ARRET et après une mise hors tension puis sous tension. Voir Configuration des paramètres d'émission.

Tableau 13- 98 Types de données pour les paramètres

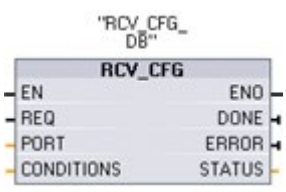
Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : faux)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
RTSONDLY	IN	UInt	Nombre de millisecondes à attendre après activation de RTS avant toute émission de données d'émission. Ce paramètre n'est valable que si le contrôle de flux matériel est activé. La plage valide va de 0 à 65 535 ms. La valeur 0 désactive la fonction. (valeur par défaut : 0)
RTSOFFDLY	IN	UInt	Nombre de millisecondes à attendre avant de désactiver RTS après l'émission des données d'émission. Ce paramètre n'est valable que si le contrôle de flux matériel est activé. La plage valide va de 0 à 65 535 ms. La valeur 0 désactive la fonction. (valeur par défaut : 0)
BREAK	IN	UInt	Ce paramètre indique qu'une pause du nombre indiqué de temps bit sera envoyée au début de chaque message. La valeur maximale est 65535 temps bit, avec un maximum de huit secondes. La valeur 0 désactive la fonction. (valeur par défaut : 12)
IDLELINE	IN	UInt	Ce paramètre indique que la ligne restera inactive pendant le nombre indiqué de temps bit avant le début de chaque message. La valeur maximale est 65535 temps bit, avec un maximum de huit secondes. La valeur 0 désactive la fonction. (valeur par défaut : 12)
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Tableau 13- 99 Codes d'erreur

STATUS (W#16#....)	Description
80B0	Configuration d'alarme d'émission non autorisée.
80B1	Le temps de pause est supérieur à la valeur maximale autorisée.
80B2	Le temps d'inactivité est supérieur à la valeur maximale autorisée.

13.6.1.3 RCV_CFG (Configurer dynamiquement les paramètres de réception série)

Tableau 13- 100 Instruction RCV_CFG (Configuration de réception)

CONT/LOG	SCL	Description
	<pre>"RCV_CFG_DB" (REQ:=_bool_in_, PORT:=_uint_in_, CONDITIONS:=_struct_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	RCV_CFG procède à la configuration dynamique de paramètres de réception série pour un port de communication point à point. Cette instruction configure les conditions qui signalent le début et la fin d'un message reçu. Tout message mis en file d'attente à l'intérieur d'un CM ou d'un CB est rejeté lorsque RCV_CFG est exécuté.

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Vous pouvez définir la configuration statique initiale du port de communication dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction RCV_CFG dans votre programme pour modifier la configuration.

Les modifications de configuration de RCV_CFG ne sont pas stockées de façon permanente dans la CPU. Les paramètres configurés dans la configuration des appareils sont restaurés lorsque la CPU passe de l'état MARCHE à l'état ARRÊT et après une mise hors tension puis sous tension. Reportez-vous à Configuration des paramètres de réception (Page 1142) pour plus d'informations.

Tableau 13- 101 Types de données pour les paramètres

Paramètre et type	Type de données	Description	
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : faux)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
CONDITIONS	IN	CONDITIONS	La structure de données Conditions indique les conditions de début et de fin de message comme décrit ci-dessous.
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Conditions de début pour l'instruction RCV_PTP

L'instruction RCV_PTP utilise la configuration indiquée par l'instruction RCV_CFG pour déterminer le début et la fin de messages de communication point à point. Le début d'un message est déterminé par les conditions de début. Le début d'un message peut être déterminé par une condition de début ou par une combinaison de conditions de début. Si plusieurs conditions de début sont spécifiées, toutes les conditions doivent être satisfaites avant que le message ne commence.

Reportez-vous à la rubrique "Configuration des paramètres de réception (Page 1142)" pour une description des conditions de début de message.

Structure pour le paramètre CONDITIONS, partie 1 (conditions de début)

Tableau 13- 102 Structure CONDITIONS pour les conditions de début (START)

Paramètre et type		Type de données	Description
STARTCOND	IN	UInt	Définit la condition de début (valeur par défaut : 1) <ul style="list-style-type: none"> • 01H : Caractère de début • 02H : Caractère quelconque • 04H : Pause • 08H : Ligne inactive • 10H : Séquence 1 • 20H : Séquence 2 • 40H : Séquence 3 • 80H : Séquence 4
IDLETIME	IN	UInt	Nombre de temps bit requis pour un délai de ligne inactive. (valeur par défaut : 40). Utilisé uniquement avec une condition de ligne inactive. 0 à 65535
STARTCHAR	IN	Byte	Caractère de début utilisé avec la condition caractère de début (valeur par défaut : B#16#2)
SEQ[1].CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 1 : (valeur par défaut : B#16#0) Il s'agit des bits d'activation pour chaque caractère dans la séquence de début. <ul style="list-style-type: none"> • 01H : Caractère 1 • 02H : Caractère 2 • 04H : Caractère 3 • 08H : Caractère 4 • 10H : Caractère 5 Désactiver le bit associé à un caractère signifie que n'importe quel caractère conviendra dans cette position de la séquence.
SEQ[1].STR	IN	Char[5]	Caractères de début de la séquence 1 (5 caractères). Valeur par défaut : 0
SEQ[2].CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 2. Valeur par défaut : B#16#0

Paramètre et type		Type de données	Description
SEQ[2].STR	IN	Char[5]	Caractères de début de la séquence 2 (5 caractères). Valeur par défaut : 0
SEQ[3].CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 3. Valeur par défaut : B#16#0
SEQ[3].STR	IN	Char[5]	Caractères de début de la séquence 3 (5 caractères). Valeur par défaut : 0
SEQ[4].CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 4. Valeur par défaut : B#16#0
SEQ[4].STR	IN	Char[5]	Caractères de début de la séquence 4 (5 caractères). Valeur par défaut : 0

Exemple

Considérons le message reçu suivant codé en hexadécimal : "68 10 aa 68 bb 10 aa 16" et les séquences de début configurées présentées dans le tableau ci-après. Les séquences de début commencent à être évaluées lorsque la réception du premier caractère 68H a abouti. Après réception du quatrième caractère (le second 68H), la condition de début 1 est satisfaite. Une fois les conditions de début satisfaites, l'évaluation des conditions de fin commence.

Le traitement de la séquence de début peut être interrompu en raison de diverses erreurs de parité, de trame ou de dépassement de délai inter-caractère. Si ces erreurs se produisent, aucun message n'est reçu puisque la condition de début n'était pas satisfaite.

Tableau 13- 103 Conditions de début

Condition de début	Premier caractère	Premier caractère +1	Premier caractère +2	Premier caractère +3	Premier caractère +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Conditions de fin pour l'instruction RCV_PTP

La fin d'un message est déterminée par l'indication de conditions de fin. La fin d'un message est déterminée par la première occurrence d'une ou de plusieurs conditions de fin configurées. Le paragraphe "Conditions de fin de message" dans la rubrique "Configuration des paramètres de réception (Page 1142)" décrit les conditions de fin que vous pouvez configurer dans l'instruction RCV_CFG.

Vous pouvez configurer les conditions de fin soit dans les propriétés de l'interface de communication dans la configuration des appareils, soit dans l'instruction RCV_CFG. Les paramètres de réception (conditions de début et conditions de fin) reprennent leur valeur de la configuration des appareils à chaque fois que la CPU passe de l'état ARRET à l'état MARCHE. Si le programme utilisateur STEP 7 exécute RCV_CFG, les paramétrages sont modifiés pour être conformes aux conditions de RCV_CFG.

Structure pour le paramètre CONDITIONS, partie 2 (conditions de fin)

Tableau 13- 104 Structure CONDITIONS pour les conditions de fin (END)

Paramètre	Type de paramètre	Type de données	Description
ENDCOND	IN	UInt 0	Ce paramètre indique la condition de fin de message : <ul style="list-style-type: none"> • 01H : Temps de réponse • 02H : Temps de message • 04H : Délai inter-caractères • 08H : Longueur maximale • 10H : N + Lg + M • 20H : Séquence
MAXLEN	IN	UInt 1	Longueur de message maximale. Utilisé uniquement lorsque la condition de fin longueur maximale est sélectionnée. 1 à 1024 octets
N	IN	UInt 0	Position d'octet du champ de longueur à l'intérieur du message. Utilisé uniquement avec la condition de fin N + Lg + M. 1 à 1022 octets
LENGTHSIZE	IN	UInt 0	Taille du champ de longueur (1, 2 ou 4 octets). Utilisé uniquement avec la condition de fin N + Lg + M.
LENGTHM	IN	UInt 0	Indiquez le nombre de caractères suivant le champ de longueur qui ne sont pas inclus dans la valeur du champ de longueur. Utilisé uniquement avec la condition de fin N + Lg + M. 0 à 255 octets
RCVTIME	IN	UInt 200	Indiquez combien de temps attendre le premier caractère à recevoir. Il sera mis fin à l'opération de réception avec une erreur si aucune réception de caractère n'aboutit pendant le temps indiqué. Utilisé uniquement avec la condition temps de réponse. (0 à 65535 temps bit, avec un maximum de 8 secondes) Ce paramètre n'est pas une condition de fin de message puisque l'évaluation s'achève lors de la réception du premier caractère d'une réponse. Ce n'est une condition de fin que dans la mesure où il met fin à une opération de réception car aucune réponse n'est reçue alors qu'une réponse est attendue. Vous devez sélectionner une condition de fin distincte.
MSGTIME	IN	UInt 200	Indiquez combien de temps attendre pour que le message entier soit complètement reçu une fois le premier caractère reçu. Ce paramètre est utilisé uniquement lorsque la condition dépassement du délai d'attente de message est sélectionnée. (0 à 65535 millisecondes)
CHARGAP	IN	UInt 12	Indiquez le nombre de temps bit entre caractères. Si le nombre de temps bit entre caractères dépasse la valeur indiquée, la condition de fin est satisfaite. Utilisé uniquement avec la condition délai inter-caractères. (0 à 65535 temps bit, avec un maximum de 8 secondes)

13.6 Communication PtP d'héritage (CM/CB 1241 uniquement)

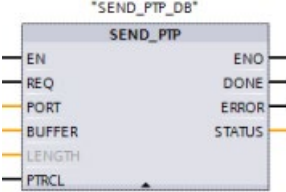
Paramètre	Type de paramètre	Type de données	Description
SEQ.CTL	IN	Byte B#16#0	Contrôle ignorer/comparer pour chaque caractère de la séquence 1 : Il s'agit des bits d'activation pour chaque caractère dans la séquence de fin. Le caractère 1 est le bit 0, le caractère 2 le bit 1, ..., le caractère 5 le bit 4. Désactiver le bit associé à un caractère signifie que n'importe quel caractère conviendra dans cette position de la séquence.
SEQ.STR	IN	Char[5] 0	Caractères de début de la séquence 1 (5 caractères)

Tableau 13- 105 Codes d'erreur

STATUS (W#16#...)	Description
80C0	Condition de début illicite sélectionnée
80C1	Condition de fin illicite sélectionnée, pas de condition de fin sélectionnée
80C2	Alarme de réception activée ce qui n'est pas possible
80C3	Condition de fin longueur maximale activée et longueur maximale est 0 ou > 1024.
80C4	Longueur calculée activée et N >= 1023
80C5	Longueur calculée activée et longueur différente de 1, 2 ou 4
80C6	Longueur calculée activée et M > 255
80C7	Longueur calculée activée et longueur calculée > 1024
80C8	Dépassement du délai d'attente de la réponse activé et délai d'attente de réponse est zéro.
80C9	Dépassement du délai inter-caractères activé et délai est 0.
80CA	Dépassement du délai de ligne inactive activé et délai est 0.
80CB	Séquence de fin activée mais tous les caractères sont indifférents.
80CC	Séquence de début (l'une quelconque parmi 4) activée mais tous les caractères sont indifférents.

13.6.1.4 SEND_PTP (Envoyer les données du tampon d'émission)

Tableau 13- 106 Instruction SEND_PTP (Emission de données point à point)

CONT/LOG	SCL	Description
	<pre>"SEND_PTP_DB" (REQ:=_bool_in_, PORT:=_uint_in_, BUFFER:=_variant_in_, LENGTH:=_uint_in_, PTRCL:=_bool_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	<p>SEND_PTP déclenche l'émission des données et transfère la mémoire tampon affectée vers l'interface de communication. Le programme de la CPU se poursuit pendant que le CM ou le CB envoie les données à la vitesse de transmission définie. Une seule opération d'émission peut être en attente à un moment donné. Le CM ou le CB renvoie une erreur si une seconde instruction SEND_PTP est exécutée alors qu'il est déjà en train d'envoyer un message.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 107 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de l'émission demandée en cas de front montant de cette entrée de validation d'émission. Cela déclenche le transfert du contenu de la mémoire tampon vers l'interface de communication point à point. (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
BUFFER	IN	Variant	Ce paramètre pointe sur l'adresse de début de la mémoire tampon d'émission. (valeur par défaut : 0) Remarque : Les données booléennes et les tableaux booléens ne sont pas acceptés.
LENGTH ¹	IN	UInt	Longueur de trame émise en octets (valeur par défaut : 0) Utilisez toujours une longueur de 0 lorsque vous envoyez une structure complexe.
PTRCL	IN	Bool	Réservé pour une utilisation future
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

¹ Paramètre optionnel : Cliquez sur la flèche en bas de la boîte CONT/LOG pour agrandir la boîte et inclure ce paramètre.

13.6 Communication PtP d'héritage (CM/CB 1241 uniquement)

Tant qu'une opération d'émission est en cours, les sorties DONE et ERROR sont à FALSE. Lorsqu'une opération d'émission est achevée, soit la sortie DONE soit la sortie ERROR est mise à TRUE pour montrer l'état de l'opération d'émission. Lorsque DONE ou ERROR est TRUE, la sortie STATUS est valide.

L'instruction renvoie un état de 16#7001 si l'interface de communication accepte les données d'émission. Les exécutions suivantes de SEND_PTP renvoient 16#7002 si le CM ou le CB est toujours occupé à émettre. Une fois l'opération d'émission achevée, le CM ou le CB renvoie l'état 16#0000 pour l'opération d'émission (si aucune erreur ne s'est produite). Les exécutions suivantes de SEND_PTP avec REQ au niveau bas renvoient l'état 16#7000 (pas occupé).

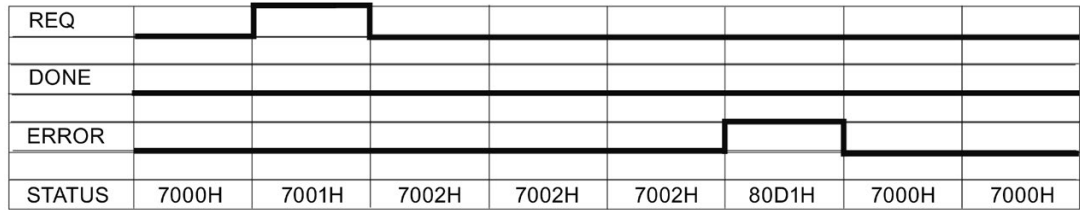
Les schémas suivants montrent la relation des valeurs de sortie à REQ. On part du principe que l'instruction est appelée périodiquement pour vérifier l'état du processus d'émission. Dans le schéma ci-après, on suppose que l'instruction est appelée à chaque cycle (représenté par les valeurs de STATUS).

REQ	[Impulsion]						
DONE						[Impulsion]	[Impulsion]
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

Le schéma suivant montre que les paramètres DONE et STATUS ne sont valables que pour un cycle si la ligne REQ présente une impulsion (pendant un cycle) pour déclencher l'opération d'émission.

REQ	[Impulsion]							
DONE						[Impulsion]	[Impulsion]	
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

Le schéma suivant montre la relation des paramètres DONE, ERROR et STATUS en présence d'une erreur.



Les valeurs de DONE, ERROR et STATUS ne sont valables que jusqu'à l'exécution suivante de SENT_PTP avec le même DB d'instance.

Tableau 13- 108 Codes d'erreur

STATUS (W#16#....)	Description
80D0	Nouvelle demande alors que l'émetteur est actif
80D1	Emission interrompue car pas de CTS pendant le temps d'attente
80D2	Emission interrompue car pas de DSR en provenance de l'équipement DCE
80D3	Emission interrompue en raison d'un débordement de la file d'attente (émission de plus de 1024 octets)
80D5	Signal de polarisation inverse (rupture de fil)
833A	Le DB pour le paramètre BUFFER n'existe pas.

13.6.1.5 RCV_PTP (Valider la réception des messages)

Tableau 13- 109 Instruction RCV_PTP (Réception point à point)

CONT/LOG	SCL	Description
<p>"RCV_PTP_DB"</p>	<pre>"RCV_PTP_DB" (EN_R:= _bool_in_, PORT:= _uint_in_, BUFFER:= _variant_in_, NDR=> _bool_out_, ERROR=> _bool_out_, STATUS=> _word_out_, LENGTH=> _uint_out_);</pre>	<p>RCV_PTP vérifie la présence de messages reçus dans le CM ou le CB. Si un message est disponible, il est transféré du CM ou du CB à la CPU. Une valeur STATUS appropriée est renvoyée en cas d'erreur.</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 110 Types de données pour les paramètres

Paramètre et type		Type de données	Description
EN_R	IN	Bool	Lorsque cette entrée est à TRUE et qu'un message est disponible, ce dernier est transféré du CM ou du CB dans la mémoire tampon BUFFER. Lorsque EN_R est FALSE, la présence de messages est vérifiée dans le CM ou le CB et les sorties NDR, ERROR et STATUS sont actualisées, mais le message n'est pas transféré dans la mémoire tampon BUFFER. (valeur par défaut : 0)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
BUFFER	IN	Variant	Ce paramètre pointe sur l'adresse de début de la mémoire tampon de réception. Cette mémoire tampon doit être suffisamment grande pour recevoir la longueur de message maximale. Les données booléennes et les tableaux booléens ne sont pas acceptés. (valeur par défaut : 0)
NDR	OUT	Bool	TRUE pour un cycle lorsque de nouvelles données sont prêtes et que l'opération s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque l'opération s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)
LENGTH	OUT	UInt	Longueur du message renvoyé en octets (valeur par défaut : 0)

Remarquez la corrélation suivante entre l'entrée EN_R et la mémoire tampon des messages de l'instruction RCV_PTP :

L'entrée EN_R commande la copie d'un message reçu vers la BUFFER.

Lorsque l'entrée EN_R est TRUE et qu'un message est disponible, la CPU transfère le message de CM ou CB à la BUFFER et met à jour les sorties NDR, ERROR, STATUS, et LENGTH.

Lorsque EN_R est FALSE, la CPU vérifie la présence de messages dans CM ou CB et met à jour les sorties NDR, ERROR, et STATUS, mais ne transfère pas le message à la BUFFER. (À noter que la valeur par défaut d'EN_R est FALSE.)

La pratique recommandée consiste à définir EN_R sur TRUE et à commander l'exécution de l'instruction RCV_PTP avec l'entrée EN.

La valeur STATUS est valide si soit NDR soit ERROR est TRUE. La valeur STATUS fournit la raison de l'arrêt de l'opération de réception dans le CM ou le CB. Il s'agira typiquement d'une valeur positive indiquant que l'opération de réception a réussi et que le processus de réception s'est terminé normalement. Si la valeur STATUS est négative (le bit de poids fort de la valeur hexadécimale est à 1), cela signifie que l'opération de réception a été interrompue en raison d'une situation d'erreur, telles des erreurs de parité, de trame ou de débordement.

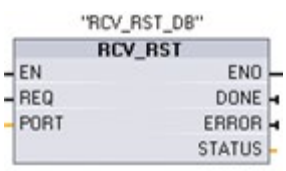
Chaque interface de communication point à point peut mettre jusqu'à 1024 octets en mémoire tampon. Il peut s'agir d'un seul grand message ou de plusieurs petits messages. S'il y a plus d'un message disponible dans le CM ou le CB, l'instruction RCV_PTP renvoie le message disponible le plus ancien. Une nouvelle exécution de l'instruction RCV_PTP renverra le message le plus ancien suivant.

Tableau 13- 111 Codes d'erreur

STATUS (W#16#...)	Description
0000	Pas de mémoire tampon
0094	Message interrompu car la longueur de caractères maximum a été reçue
0095	Message interrompu en raison de l'expiration du délai de message
0096	Message interrompu en raison de l'expiration du délai intercaractère
0097	Message interrompu en raison de l'expiration du délai de réponse
0098	Message interrompu car la condition de longueur "N+LEN+M" était satisfaite
0099	Message interrompu car la séquence de fin était satisfaite
80E0	Message interrompu car la mémoire tampon de réception est pleine
80E1	Message interrompu en raison d'une erreur de parité
80E2	Message interrompu en raison d'une erreur de trame
80E3	Message interrompu en raison d'une erreur de débordement
80E4	Message interrompu car la longueur calculée dépasse la taille de la mémoire tampon
80E5	Signal de polarisation inverse (rupture de fil)
833A	Le DB pour le paramètre BUFFER n'existe pas.

13.6.1.6 RCV_RST (Effacer le tampon de réception)

Tableau 13- 112 Instruction RCV_RST (Réinitialiser récepteur)

CONT/LOG	SCL	Description
	<pre>"RCV_RST_DB" (REQ:= _bool_in_, PORT:= _uint_in_, DONE=> _bool_out_, ERROR=> _bool_out_, STATUS=> _word_out_);</pre>	RCV_RST efface les mémoires tampons de réception dans le CM ou le CB.

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 113 Types de données pour les paramètres


Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la réinitialisation du récepteur en cas de front montant de cette entrée de validation (valeur par défaut : faux)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
DONE	OUT	Bool	Cette sortie est TRUE pour un cycle lorsque la dernière requête s'est achevée sans erreur.
ERROR	OUT	Bool	Cette sortie est TRUE lorsque la dernière requête s'est achevée avec des erreurs. Dans ce cas, la sortie STATUS contient en outre les codes d'erreur correspondants.
STATUS	OUT	Word	Code d'erreur (valeur par défaut : 0) Les codes d'état de la communication sont présentés au paragraphe Paramètres communs pour les instructions point à point (Page 1155).

Remarque

Vous pouvez utiliser l'instruction RCV_RST pour garantir que les mémoires tampons de messages sont vides après une erreur de communication ou après la modification d'un paramètre de communication tel que le débit. L'exécution de RCV_RST provoque l'effacement de toutes les mémoires tampons de messages internes par le module. Une fois les mémoires tampons de messages effacées, vous pouvez être sûr que, lorsque votre programme exécutera une nouvelle instruction de réception, les messages qu'il renverra seront bien des nouveaux messages et non d'anciens messages antérieurs à l'appel de RCV_RST.

13.6.1.7 SGN_GET (Interroger les signaux RS-232)

Tableau 13- 114 Instruction SGN_GET (Lire signaux RS232)

CONT/LOG	SCL	Description
	<pre>"SGN_GET_DB" (REQ:=_bool_in_, PORT:=_uint_in_, NDR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DTR=>_bool_out_, DSR=>_bool_out_, RTS=>_bool_out_, CTS=>_bool_out_, DCD=>_bool_out_, RING=>_bool_out_);</pre>	<p>SGN_GET lit l'état en cours des signaux de communication RS232.</p> <p>Cette fonction n'est valable que pour le module de communication RS232.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 115 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Lecture des valeurs d'état de signal RS232 en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
NDR	OUT	Bool	TRUE pour un cycle lorsque de nouvelles données sont prêtes et que l'opération s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque l'opération s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)
DTR	OUT	Bool	Terminal de données prêt, module prêt (sortie). Valeur par défaut : False
DSR	OUT	Bool	Modem prêt, partenaire de communication prêt (entrée). Valeur par défaut : False
RTS	OUT	Bool	Demande pour émettre, module prêt à émettre (sortie). Valeur par défaut : False
CTS	OUT	Bool	Prêt à émettre, le partenaire de communication peut recevoir les données (entrée). Valeur par défaut : False
DCD	OUT	Bool	Détection de porteuse, réception du niveau de signal (toujours faux, non pris en charge)
RING	OUT	Bool	Indicateur d'appel, signale un appel entrant (toujours faux, non pris en charge)

13.6 Communication PtP d'héritage (CM/CB 1241 uniquement)

Tableau 13- 116 Codes d'erreur

STATUS (W#16#....)	Description
80F0	Le CM ou le CB est un module RS485 et il n'y a pas de signaux disponibles.

13.6.1.8 SGN_SET (Régler les signaux RS-232)

Tableau 13- 117 Instruction SGN_SET (Activer signaux RS232)

CONT/LOG	SCL	Description
	<pre>"SGN_SET_DB" (REQ:=_bool_in_, PORT:=_uint_in_, SIGNAL:=_byte_in_, RTS:=_bool_in_, DTR:=_bool_in_, DSR:=_bool_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	<p>SGN_SET active l'état des signaux de communication RS232.</p> <p>Cette fonction n'est valable que pour le module de communication RS232.</p>

¹ STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 13- 118 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Lancement de l'opération d'activation des signaux RS232 en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
SIGNAL	IN	Byte	Sélectionne le signal à régler (choix multiple possible). Valeur par défaut : 0 <ul style="list-style-type: none"> • 01H = Activer RTS • 02H = Activer DTR • 04H = Activer DSR
RTS	IN	Bool	Valeur à définir Demande pour émettre, module prêt à émettre (true ou false), valeur par défaut : False
DTR	IN	Bool	Valeur à définir Terminal de données prêt, module prêt (true ou false). Valeur par défaut : False
DSR	IN	Bool	Modem prêt (valable uniquement pour les interfaces de type DCE), non utilisé.
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

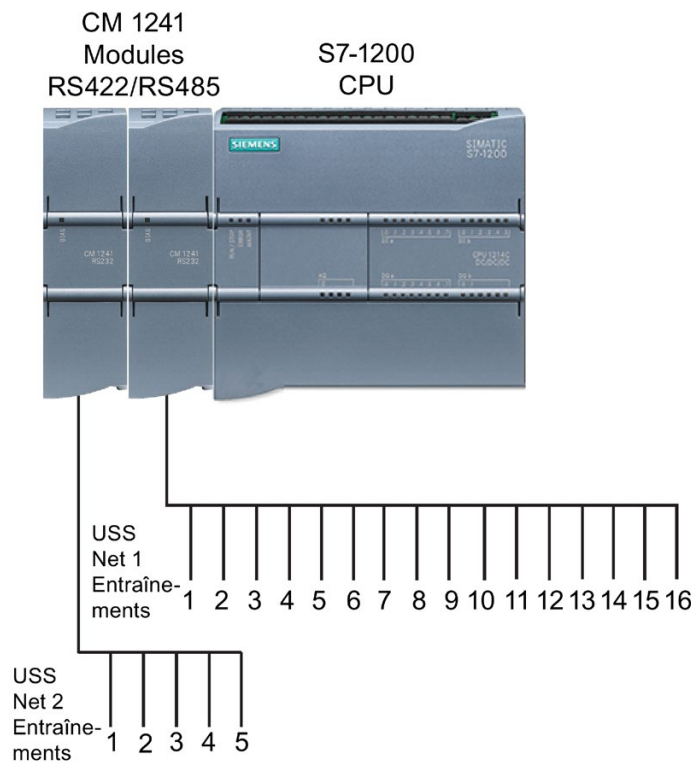
Tableau 13- 119 Codes d'erreur

STATUS (W#16#....)	Description
80F0	Le CM ou le CB est un module RS485 et il n'y a pas de signaux pouvant être activés.
80F1	Les signaux ne peuvent être activés en raison d'un contrôle de flux matériel.
80F2	DSR ne peut pas être activé car le module est un équipement terminal de données (DTE).
80F3	DTR ne peut pas être activé car le module est un équipement de transmission de données (DCE).

13.7 Communication d'héritage USS (CM/CB 1241 uniquement)

Les instructions USS commandent le fonctionnement d'entraînements de moteur qui prennent en charge le protocole USS (interface série universelle). Vous pouvez utiliser les instructions USS pour communiquer avec plusieurs entraînements via des liaisons RS485 vers des modules de communication CM 1241 RS485 ou un Communication Board CB 1241 RS485. Il est possible d'installer jusqu'à trois modules CM 1241 RS422/RS485 et un CB 1241 RS485 dans une CPU S7-1200. Chaque port RS485 peut faire fonctionner seize entraînements au plus.

Le protocole USS utilise un réseau maître-esclave pour communiquer par le biais d'un bus série. Le maître utilise un paramètre adresse pour envoyer un message à un esclave sélectionné. Un esclave ne peut lui jamais émettre s'il n'y a pas d'abord été invité par un maître. Un transfert de message direct entre les esclaves individuels est impossible. La communication USS opère en mode semi-duplex. L'illustration suivante de USS montre un schéma de réseau pour un exemple d'application d'entraînement.



Avant la sortie de STEP 7 V13 SP1 et des CPU S7-1200 V4.1, les instructions de communication USS existaient avec des noms différents et, dans certains cas, avec des interfaces légèrement différentes. Les concepts généraux s'appliquent aux deux jeux d'instructions. Reportez-vous aux instructions d'héritage individuelles USS pour obtenir des informations relatives à la programmation.

13.7.1 Sélection de la version des instructions USS

Deux versions des instructions USS sont disponibles dans STEP 7 :

- La version 2.0 était initialement destinée à STEP 7 Basic/Professional V13.
- La version 2.1 est disponible dans STEP 7 Basic/Professional V13 SP1 ou ultérieur.

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

Vous ne pouvez pas utiliser les deux versions des instructions avec le même module, mais deux modules différents peuvent utiliser différentes versions des instructions.



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.

USS		V1.1
USS_PORT	Edit communication via US...	V1.1
USS_DRV	Swap data with drive	V1.1
USS_RPM	Readout parameters from t...	V1.1
USS_WPM	Change parameters in the d...	V1.1

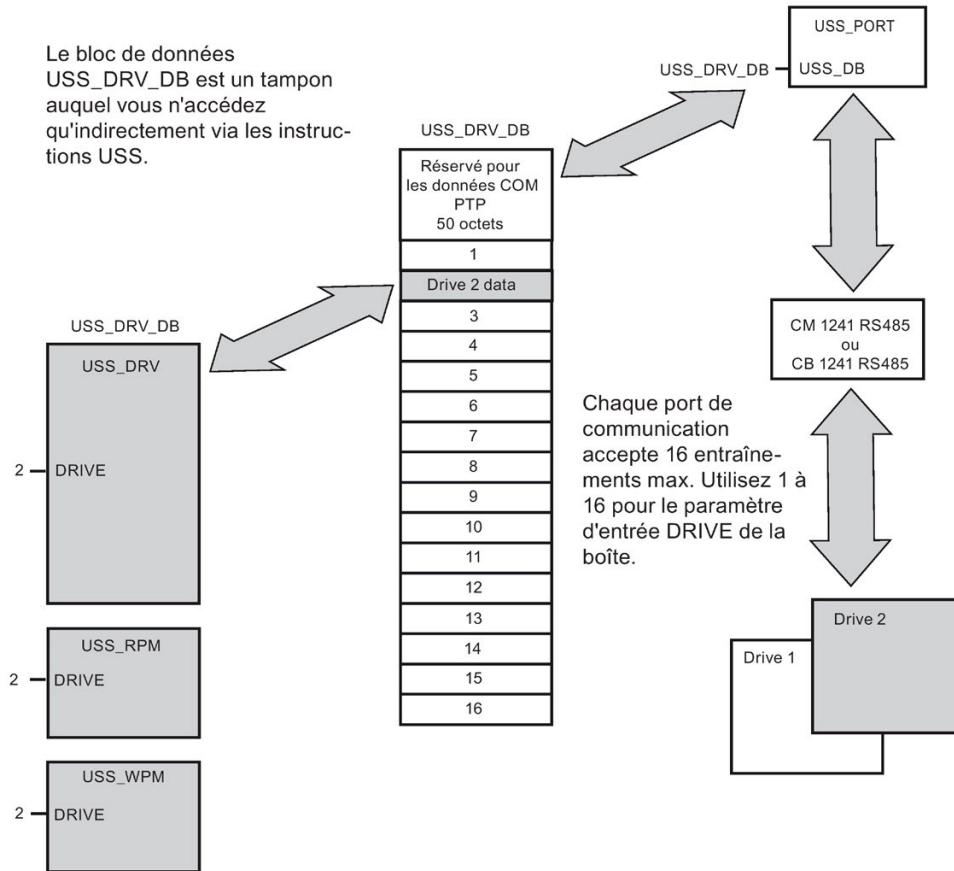
Pour changer la version des instructions USS, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction USS dans votre programme, une nouvelle instance de FB ou FC, en fonction de l'instruction USS sélectionnée, est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB ou FC dans l'arborescence du projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction USS dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB ou FC USS dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction USS.

13.7.2 Conditions requises pour l'utilisation du protocole USS

Les quatre instructions USS utilisent un FB et trois FC pour la prise en charge du protocole USS. Un bloc de données d'instance USS_PORT est utilisé pour chaque réseau USS. Il contient de la mémoire temporaire et des mémoires tampons pour tous les entraînements de ce réseau USS. Les instructions USS partagent les informations dans ce bloc de données.



Tous les entraînements (16 au maximum) connectés à un même port RS485 font partie du même réseau USS. Tous les entraînements connectés à un port RS485 différent font partie d'un réseau USS différent. Chaque réseau USS est géré à l'aide d'un bloc de données unique. Toutes les instructions associées à un même réseau USS doivent partager ce bloc de données. Cela inclut toutes les instructions USS_DRV, USS_PORT, USS_RPM et USS_WPM utilisées pour commander tous les entraînements dans un même réseau USS.

L'instruction USS_DRV est un bloc fonctionnel (FB). Lorsque vous placez l'instruction USS_DRV dans l'éditeur de programmes, la boîte de dialogue "Options d'appel" s'affiche et vous demande d'affecter un DB à ce FB. S'il s'agit de la première instruction USS_DRV dans ce programme pour ce réseau USS, vous pouvez accepter le DB indiqué par défaut (ou modifier son nom si vous le désirez) et ce nouveau DB est alors créé pour vous. En revanche, s'il ne s'agit pas de la première instruction USS_DRV pour cette voie, vous devez utiliser la liste déroulante dans la boîte de dialogue "Options d'appel" pour sélectionner le nom de DB précédemment affecté à ce réseau USS.

Les instructions USS_PORT, USS_RPM et USS_WPM sont des fonctions (FC). Il n'y a pas d'affectation de DB lorsque vous placez ces FC dans l'éditeur. En revanche, vous devez affecter le DB approprié à l'entrée "USS_DB" de ces instructions. Double-cliquez sur le champ de paramètre, puis cliquez sur l'icône d'aide pour voir les noms de DB disponibles.

La fonction USS_PORT gère la communication effective entre la CPU et les entraînements par le biais du port de communication RS485 point à point. Chaque appel de cette fonction gère une communication avec un entraînement. Votre programme doit appeler cette fonction suffisamment rapidement pour éviter un dépassement du délai de communication par les entraînements. Vous pouvez appeler cette fonction dans un OB de cycle de programme principal ou dans n'importe quel OB d'alarme.

Typiquement, vous appellerez la fonction USS_PORT dans un OB d'alarme cyclique. Définissez la période de cet OB d'alarme cyclique à environ la moitié de l'intervalle d'appel minimum. Une communication à 1200 bauds devrait, par exemple, utiliser une période d'au plus 350 ms).

Le bloc fonctionnel USS_DRV permet à votre programme d'accéder à un entraînement spécifié dans le réseau USS. Ses entrées et sorties constituent les états et les commandes pour l'entraînement. S'il y a 16 entraînements dans le réseau, votre programme doit comporter au moins 16 appels de USS_DRV, un pour chaque entraînement. Vous devez appeler ces blocs à la fréquence nécessaire pour commander le fonctionnement de l'entraînement.

Vous ne pouvez appeler le bloc fonctionnel USS_DRV que dans un OB de cycle de programme principal.

PRUDENCE

Considérations sur l'appel d'instructions USS dans des OB

Appelez USS_DRV, USS_RPM et USS_WPM uniquement à partir d'un OB de cycle de programme. La fonction USS_PORT peut être appelée dans n'importe quel OB, généralement un OB d'alarme cyclique.

N'utilisez pas les instructions USS_DRV, USS_RPM ou USS_WPM dans un OB de priorité supérieure à celle de l'instruction USS_PORT correspondante. Ainsi, ne placez pas USS_PORT dans l'OB de cycle de programme principal et USS_RPM dans un OB d'alarme cyclique. L'interruption possible de USS_PORT peut produire des erreurs inattendues qui peuvent entraîner des blessures.

Les fonctions USS_RPM et USS_WPM lisent et écrivent les paramètres de fonctionnement de l'entraînement distant. Ces paramètres pilotent le fonctionnement interne de l'entraînement. Vous trouverez la définition de ces paramètres dans le manuel de l'entraînement. Votre programme peut contenir autant de ces fonctions que nécessaire, mais une seule demande de lecture ou d'écriture peut être active par entraînement à un moment donné. Vous ne pouvez appeler les fonctions USS_RPM et USS_WPM que dans un OB de cycle de programme principal.

Calcul du temps requis pour communiquer avec l'entraînement

La communication avec l'entraînement est asynchrone par rapport au cycle S7-1200. Typiquement, le S7-1200 exécute plusieurs cycles avant qu'une transaction de communication avec un entraînement s'achève.

L'intervalle USS_PORT est le temps requis pour une transaction avec un entraînement. Le tableau ci-dessous montre l'intervalle USS_PORT minimum pour chaque vitesse de transmission. Appeler la fonction USS_PORT plus fréquemment que l'intervalle USS_PORT n'augmente pas le nombre de transactions. Le délai d'attente de l'entraînement est la quantité de temps qui pourrait être utilisée pour une transaction si des erreurs de communication entraînaient 3 tentatives pour achever la transaction. Par défaut, la bibliothèque du protocole USS exécute automatiquement jusqu'à 2 nouvelles tentatives pour chaque transaction.

Tableau 13- 120 Calcul des exigences en temps

Débit	Intervalle d'appel USS_PORT minimum calculé (millisecondes)	Délai d'attente de message par entraînement (millisecondes)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

13.7.3 Instructions d'héritage USS

13.7.3.1 Instruction USS_PORT (Editer communication via réseau USS)

Tableau 13- 121 Instruction USS_PORT

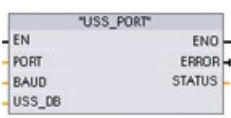
CONT/LOG	SCL	Description
	<pre>USS_PORT (PORT:= _uint_in_, BAUD:= _dint_in_, ERROR=> _bool_out_, STATUS=> _word_out_, USS_DB:= fbtref inout);</pre>	<p>L'instruction USS_PORT gère la communication par le biais d'un réseau USS.</p>

Tableau 13- 122 Types de données pour les paramètres



Paramètre et type		Type de données	Description
PORT	IN	Port	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
BAUD	IN	DInt	Vitesse de transmission à utiliser pour la communication USS
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_DRV dans votre programme.
ERROR	OUT	Bool	Lorsqu'elle est vraie, cette sortie signale qu'une erreur s'est produite et que la sortie STATUS est valide.
STATUS	OUT	Word	La valeur STATUS de la demande indique le résultat du cycle ou de l'initialisation. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

Typiquement, il n'y a qu'une instruction USS_PORT par port de communication point à point dans le programme et chaque appel de cette fonction gère une transmission vers ou depuis un entraînement unique. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données d'instance.

Votre programme doit exécuter la fonction USS_PORT suffisamment souvent pour empêcher les dépassements de délai des entraînements. USS_PORT est généralement appelé dans un OB d'alarme cyclique pour éviter les dépassements de délai des entraînements et mettre les mises à jour les plus récentes des données USS à disposition des appels USS_DRV.

13.7.3.2 Instruction USS_DRV (Données permutées avec l'entraînement)

Tableau 13- 123 Instruction USS_DRV

CONT/LOG	SCL	Description
<p>Vue par défaut</p>  <p>Vue étendue</p> 	<pre>"USS_DRV_DB" (RUN:= _bool_in_, OFF2:= _bool_in_, OFF3:= _bool_in_, F_ACK:= _bool_in_, DIR:= _bool_in_, DRIVE:= _usint_in_, PZD_LEN:= _usint_in_, SPEED_SP:= _real_in_, CTRL3:= _word_in_, CTRL4:= _word_in_, CTRL5:= _word_in_, CTRL6:= _word_in_, CTRL7:= _word_in_, CTRL8:= _word_in_, NDR=> _bool_out_, ERROR=> _bool_out_, STATUS=> _word_out_, RUN_EN=> _bool_out_, D_DIR=> _bool_out_, INHIBIT=> _bool_out_, FAULT=> _bool_out_, SPEED=> _real_out_, STATUS1=> _word_out_, STATUS3=> _word_out_, STATUS4=> _word_out_, STATUS5=> _word_out_, STATUS6=> _word_out_, STATUS7=> _word_out_, STATUS8=> _word_out_);</pre>	<p>L'instruction USS_DRV échange des données avec un entraînement en créant des messages de demande et en interprétant les messages de réponse de l'entraînement. Il faut utiliser un bloc fonctionnel distinct pour chaque entraînement, mais toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données d'instance. Vous devez créer le nom du DB lorsque vous insérez la première instruction USS_DRV ; vous faites ensuite appel au DB créé par la première utilisation de l'instruction.</p> <p>STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.</p>

¹ CONT et LOG : Affichez le détail de la boîte pour visualiser tous les paramètres en cliquant au bas de la boîte. Les connecteurs de paramètre qui sont grisés sont facultatifs ; l'affectation de paramètre n'y est pas exigée.

Tableau 13- 124 Types de données pour les paramètres

Paramètre et type		Type de données	Description
RUN	IN	Bool	Bit de démarrage d'entraînement. Lorsqu'elle est vraie, cette entrée permet le fonctionnement de l'entraînement à la vitesse prédéfinie. Lorsque RUN passe à FALSE pendant le fonctionnement d'un entraînement, le moteur décélère jusqu'à l'arrêt complet. Ce comportement diffère de l'arrêt électrique (OFF2) ou du freinage du moteur (OFF3).
OFF2	IN	Bool	Bit d'arrêt électrique. Lorsqu'il est faux, ce bit permet à l'entraînement de tourner en roue libre jusqu'à l'arrêt sans freinage.
OFF3	IN	Bool	Bit d'arrêt rapide. Lorsqu'il est faux, ce bit provoque un arrêt rapide par freinage de l'entraînement plutôt que par passage en roue libre jusqu'à l'arrêt.
F_ACK	IN	Bool	Bit d'acquiescement de défaut. Ce bit est mis à 1 pour réinitialiser le bit de défaut d'un entraînement. Le bit est mis à 1 après que le défaut a été corrigé afin de signaler à l'entraînement qu'il n'a plus besoin de signaler le défaut précédent.
DIR	IN	Bool	Commande du sens de l'entraînement. Ce bit est mis à 1 pour indiquer que le sens est la marche avant (pour SPEED_SP positif).
DRIVE	IN	USInt	Adresse de l'entraînement. Cette entrée est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PZD_LEN	IN	USInt	Longueur en mots. Il s'agit du nombre de mots des données PZD. Les valeurs valides sont 2, 4, 6 ou 8 mots. La valeur par défaut est 2.
SPEED_SP	IN	Real	Consigne de vitesse. Il s'agit de la vitesse de l'entraînement sous forme de pourcentage de la fréquence configurée. Une valeur positive correspond à une marche avant (lorsque DIR est vrai). La plage valide va de 200,00 à -200,00.
CTRL3	IN	Word	Mot de commande 3. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL4	IN	Word	Mot de commande 4. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL5	IN	Word	Mot de commande 5. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL6	IN	Word	Mot de commande 6. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL7	IN	Word	Mot de commande 7. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
CTRL8	IN	Word	Mot de commande 8. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement (paramètre facultatif).
NDR	OUT	Bool	Nouvelles données prêtes (New data ready) : Lorsqu'il est vrai, ce bit signale que les sorties contiennent des données pour une nouvelle demande de communication.

13.7 Communication d'héritage USS (CM/CB 1241 uniquement)

Paramètre et type		Type de données	Description
ERROR	OUT	Bool	Une erreur s'est produite. Lorsqu'elle est vraie, cette sortie signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_PORT.
STATUS	OUT	Word	La valeur STATUS de la demande indique le résultat du cycle. Il ne s'agit pas d'un mot d'état provenant de l'entraînement.
RUN_EN	OUT	Bool	Marche activée. Ce bit signale si l'entraînement fonctionne ou non.
D_DIR	OUT	Bool	Sens de l'entraînement. Ce bit signale si l'entraînement fonctionne en marche avant ou non.
INHIBIT	OUT	Bool	Entraînement inhibé. Ce bit indique l'état du bit d'inhibition dans l'entraînement.
FAULT	OUT	Bool	Défaut de l'entraînement. Ce bit signale que l'entraînement a enregistré un défaut. Vous devez résoudre le problème puis mettre le bit F_ACK à 1 pour effacer ce bit lorsqu'il a été mis à 1.
SPEED	OUT	Real	Vitesse en cours de l'entraînement (valeur mise à l'échelle du mot d'état d'entraînement 2). Il s'agit de la valeur de la vitesse de l'entraînement sous forme de pourcentage de la vitesse configurée.
STATUS1	OUT	Word	Mot d'état 1 de l'entraînement. Cette valeur contient des bits d'état fixes d'un entraînement.
STATUS3	OUT	Word	Mot d'état 3 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS4	OUT	Word	Mot d'état 4 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS5	OUT	Word	Mot d'état 5 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS6	OUT	Word	Mot d'état 6 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS7	OUT	Word	Mot d'état 7 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS8	OUT	Word	Mot d'état 8 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.

Lors de l'exécution de la première instruction USS_DRV, l'entraînement indiqué par l'adresse USS (paramètre DRIVE) est initialisé dans le DB d'instance. Après cette initialisation, les exécutions suivantes de USS_PORT peuvent commencer la communication avec l'entraînement à ce numéro d'entraînement.

La modification du numéro d'entraînement nécessite un passage de la CPU de ARRET à MARCHE afin d'initialiser le DB d'instance. Les paramètres d'entrée sont écrits dans la mémoire tampon USS de message à émettre et les sorties sont lues d'une mémoire tampon de réponse valide "précédente" s'il en existe une. Il n'y a pas de transmission de données pendant l'exécution de USS_DRV. Les entraînements communiquent lorsque USS_PORT est exécuté. USS_DRV ne fait que configurer les messages à envoyer et interpréter les données éventuelles reçues d'une précédente demande.

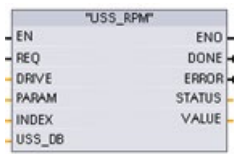
Vous pouvez commander le sens de rotation de l'entraînement via l'entrée DIR (Bool) ou via le signe (positif ou négatif) de l'entrée SPEED_SP (Real). Le tableau suivant montre comment ces entrées déterminent ensemble le sens de l'entraînement, dans l'hypothèse où le moteur est câblé pour une rotation avant.

Tableau 13- 125 Interaction entre les paramètres SPEED_SP et DIR

SPEED_SP	DIR	Sens de rotation de l'entraînement
Valeur > 0	0	Arrière
Valeur > 0	1	Avant
Valeur < 0	0	Avant
Valeur < 0	1	Arrière

13.7.3.3 Instruction USS_RPM (Lire des paramètres de l'entraînement)

Tableau 13- 126 Instruction USS_RPM

CONT/LOG	SCL	Description
 <p>The diagram shows a block titled "USS_RPM" with the following connections: Inputs (left side): EN, REQ, DRIVE, PARAM, INDEX, USS_DB. Outputs (right side): END, DONE, ERROR, STATUS, VALUE.</p>	<pre>USS_RPM(REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, INDEX:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, VALUE=>_variant_out_, USS_DB:=_fbtref_inout_);</pre>	<p>L'instruction USS_RPM lit un paramètre provenant de l'entraînement. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données. Il faut appeler USS_RPM dans un OB de cycle de programme principal.</p>

13.7 Communication d'héritage USS (CM/CB 1241 uniquement)

Tableau 13- 127 Types de données pour les paramètres

Type de paramètre		Type de données	Description
REQ	IN	Bool	Envoi de la demande. Lorsqu'elle est vraie, l'entrée REQ indique qu'une nouvelle demande de lecture est désirée. Il n'en est pas tenu compte si la demande pour ce paramètre est déjà en attente.
DRIVE	IN	USInt	Adresse de l'entraînement. DRIVE est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PARAM	IN	UInt	Numéro du paramètre. PARAM désigne le paramètre d'entraînement à écrire. La plage de ce paramètre est comprise entre 0 et 2047. Sur certains entraînements, l'octet de poids fort peut accéder à des valeurs PARAM supérieures à 2047. Vous trouverez plus de détails sur la manière d'accéder à une plage étendue dans le manuel de l'entraînement.
INDEX	IN	UInt	Indice du paramètre. INDEX désigne l'indice du paramètre d'entraînement à écrire. Valeur de 16 bits dans laquelle l'octet de poids faible est la valeur d'indice réelle avec une plage de 0 à 255. L'octet de poids fort peut être utilisé par l'entraînement et lui est spécifique. Consultez le manuel de l'entraînement pour plus de détails.
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_DRV dans votre programme.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDIInt, Real	Il s'agit de la valeur du paramètre qui a été lue et qui est uniquement valide lorsque le bit DONE est vrai.
DONE ¹	OUT	Bool	S'il est vrai, ce bit signale que la sortie VALUE contient la valeur du paramètre lue suite à la demande précédente. Cet bit est mis à 1 lorsque USS_DRV voit les données de réponse de lecture provenant de l'entraînement. Ce bit est mis à 0 lorsque vous demandez des données de réponse via une autre interrogation USS_RPM ou lors du deuxième des deux appels suivants de USS_DRV.
ERROR	OUT	Bool	Une erreur s'est produite. Lorsqu'elle est vraie, la sortie ERROR signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_PORT.
STATUS	OUT	Word	STATUS indique le résultat de la demande de lecture. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

¹ Le bit DONE indique que des données valides ont été lues dans l'entraînement de moteur référencé et transmises à la CPU. Il ne signifie pas que la bibliothèque USS est capable de lire immédiatement un autre paramètre. Une demande PKW vide doit être envoyée à l'entraînement et doit également être acquittée par l'instruction avant que la voie de paramètre pour l'entraînement spécifique ne soit mise à disposition. L'appel immédiat d'une instruction USS_RPM ou USS_WPM pour l'entraînement spécifié provoque une erreur 0x818A.

13.7.3.4 Instruction USS_WPM (Modifier les paramètres dans l'entraînement)

Remarque

Opérations d'écriture en EEPROM (pour l'EEPROM à l'intérieur d'un entraînement USS)

N'utilisez pas la fonction d'écriture permanente en EEPROM de manière abusive. Réduisez au minimum le nombre d'opérations d'écriture en EEPROM pour allonger la durée de vie de cette dernière.

Tableau 13- 128 Instruction USS_WPM

CONT/LOG	SCL	Description
	<pre>USS_WPM(REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, INDEX:=_uint_in_, EEPROM:=_bool_in_, VALUE:=_variant_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, USS DB:= fbtref inout);</pre>	<p>L'instruction USS_WPM modifie un paramètre dans l'entraînement. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données.</p> <p>Il faut appeler USS_WPM dans un OB de cycle de programme principal.</p>

13.7 Communication d'héritage USS (CM/CB 1241 uniquement)

Tableau 13- 129 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Envoi de la demande. Lorsqu'elle est vraie, l'entrée REQ indique qu'une nouvelle demande d'écriture est désirée. Il n'en est pas tenu compte si la demande pour ce paramètre est déjà en attente.
DRIVE	IN	USInt	Adresse de l'entraînement. DRIVE est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PARAM	IN	UInt	Numéro du paramètre. PARAM désigne le paramètre d'entraînement à écrire. La plage de ce paramètre est comprise entre 0 et 2047. Sur certains entraînements, l'octet de poids fort peut accéder à des valeurs PARAM supérieures à 2047. Vous trouverez plus de détails sur la manière d'accéder à une plage étendue dans le manuel de l'entraînement.
INDEX	IN	UInt	Indice du paramètre. INDEX désigne l'indice du paramètre d'entraînement à écrire. Valeur de 16 bits dans laquelle l'octet de poids faible est la valeur d'indice réelle avec une plage de 0 à 255. L'octet de poids fort peut être utilisé par l'entraînement et lui est spécifique. Consultez le manuel de l'entraînement pour plus de détails.
EEPROM	IN	Bool	Stockage dans l'EEPROM de l'entraînement. Lorsque cette entrée est vraie, une transaction d'écriture dans le paramètre de l'entraînement est stockée dans l'EEPROM de l'entraînement. Lorsqu'elle est fausse, l'écriture est temporaire et ne sera pas conservée en cas de mise hors tension puis sous tension de l'entraînement.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDIInt, Real	Valeur du paramètre qui doit être écrite. Elle doit être valide lors de la transition de REQ.
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_DRV dans votre programme.
DONE ¹	OUT	Bool	S'il est vrai, le bit DONE signale que l'entrée VALUE a été écrite dans l'entraînement. Ce bit est mis à 1 lorsque USS_DRV voit les données de réponse d'écriture provenant de l'entraînement. Ce bit est mis à 0 lorsque vous demandez les données de réponse via une autre interrogation USS_WPM ou lors du deuxième des deux appels suivants de USS_DRV.
ERROR	OUT	Bool	Lorsqu'elle est vraie, la sortie ERROR signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_PORT.
STATUS	OUT	Word	STATUS indique le résultat de la demande d'écriture. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

¹ Le bit DONE indique que des données valides ont été lues dans l'entraînement de moteur référencé et transmises à la CPU. Il ne signifie pas que la bibliothèque USS est capable de lire immédiatement un autre paramètre. Une demande PKW vide doit être envoyée à l'entraînement et doit également être acquittée par l'instruction avant que la voie de paramètre pour l'entraînement spécifique ne soit mise à disposition. L'appel immédiat d'une instruction USS_RPM ou USS_WPM pour l'entraînement spécifié provoque une erreur 0x818A.

13.7.4 Codes d'état USS d'héritage

Les codes d'état des instructions USS sont renvoyés dans la sortie STATUS des fonctions USS.

Tableau 13- 130 Codes STATUS ¹

STATUS (W#16#...)	Description
0000	Pas d'erreur
8180	La longueur de la réponse de l'entraînement ne correspondait pas aux caractères reçus de l'entraînement. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8181	Le paramètre VALUE n'avait pas le type de données Word, Real ou DWord.
8182	L'utilisateur a fourni un mot pour une valeur de paramètre et a reçu un double mot ou un réel dans la réponse provenant de l'entraînement.
8183	L'utilisateur a fourni un double mot ou un réel pour une valeur de paramètre et a reçu un mot dans la réponse provenant de l'entraînement.
8184	Le télégramme de réponse de l'entraînement contenait un total de contrôle erroné. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8185	Adresse d'entraînement illicite (plage d'adresses d'entraînement valide : 1 à 16)
8186	La consigne de vitesse est hors de la plage valide (plage de consignes de vitesse valide : -200% à 200%).
8187	Un entraînement de numéro erroné a répondu à la demande envoyée. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8188	Longueur de mot PZD illicite (plage valide = 2, 4, 6 ou 8 mots)
8189	Vitesse de transmission illicite
818A	La voie de demande de paramètre est utilisée par une autre demande pour cet entraînement.
818B	L'entraînement n'a pas répondu aux demandes et nouvelles tentatives. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
818C	L'entraînement a renvoyé une erreur étendue à une opération de demande de paramètre. Consultez la description des erreurs étendues après ce tableau.
818D	L'entraînement a renvoyé une erreur d'accès illicite à une opération de demande de paramètre. Consultez le manuel de l'entraînement pour savoir pourquoi l'accès au paramètre est éventuellement limité.
818E	L'entraînement n'a pas été initialisé. Ce code d'erreur est renvoyé à USS_RPM ou USS_WPM lorsque USS_DRV pour cet entraînement n'a pas été appelé au moins une fois. Cela empêche l'initialisation au premier cycle de USS_DRV d'écraser une demande de lecture ou d'écriture de paramètre en attente puisque l'entraînement est initialisé en tant que nouvelle entrée. Pour corriger cette erreur, appelez USS_DRV pour ce numéro d'entraînement.
80Ax-80Fx	Erreurs spécifiques renvoyées par les FB de communication point à point appelés par la bibliothèque USS. Ces valeurs de codes d'erreur ne sont pas modifiées par la bibliothèque USS et sont définies dans les descriptions des instructions point à point.

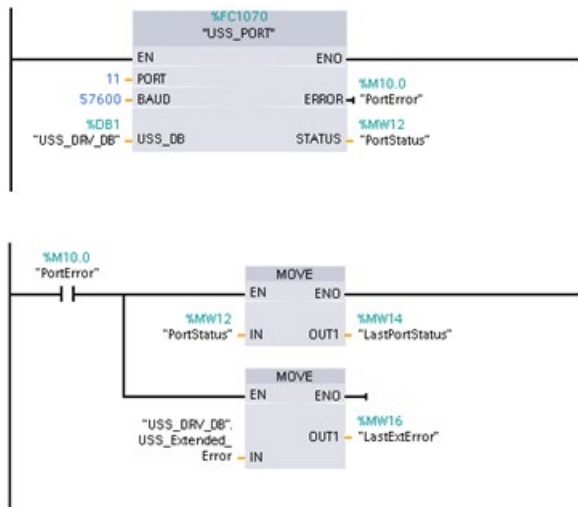
¹ En plus des erreurs d'instructions USS présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

13.7 Communication d'héritage USS (CM/CB 1241 uniquement)

Des informations supplémentaires sont fournies dans la variable "USS_Extended_Error" du DB d'instance de USS_DRV pour certains codes d'état STATUS. Pour les codes hexadécimaux STATUS 8180, 8184, 8187 et 818B, USS_Extended_Error contient le numéro de l'entraînement où l'erreur de communication s'est produite. Pour le code hexadécimal STATUS 818C, USS_Extended_Error contient un code d'erreur d'entraînement provenant de l'entraînement lié à l'utilisation d'une instruction USS_RPM ou USS_WPM.

Exemple : compte rendu d'erreurs de communication

Les erreurs de communication (STATUS = 16#818B) sont signalées uniquement dans l'instruction USS_PORT et non dans l'instruction USS_DRV. Par exemple, si le réseau n'est pas muni de résistances de terminaison correctes, un entraînement peut passer en MARCHE mais l'instruction USS_DRV montrera uniquement des 0 pour les paramètres de sortie. Dans ce cas, vous pouvez uniquement détecter l'erreur de communication dans l'instruction USS_PORT. Comme cette erreur n'est visible que pendant un cycle, vous devez ajouter une logique de capture comme illustré dans l'exemple suivant. Dans cet exemple, lorsque le bit d'erreur de l'instruction USS_PORT est à VRAI, les valeurs de STATUS et de USS_Extended_Error sont sauvegardées en mémoire M. Le numéro de l'entraînement est placé dans la variable USS_Extended_Error lorsque la valeur hexadécimale du code STATUS est 8180, 8184, 8187 ou 818B.



Réseau 1 Les valeurs de l'état du port "PortStatus" et du code d'erreur étendue "USS_DRV_DB".USS_Extended_Error ne sont valables que pendant un cycle du programme. Il faut les capturer en vue d'un traitement ultérieur.

Réseau 2 Le contact "PortError" déclenche la mémorisation de la valeur "PortStatus" dans "LastPortStatus" et de la valeur "USS_DRV_DB".USS_Extended_Error dans "LastExtError".

Accès en lecture et en écriture aux paramètres internes d'un entraînement

Les entraînements USS acceptent l'accès en lecture et écriture aux paramètres internes d'un entraînement. Cette caractéristique permet la commande et la configuration à distance de l'entraînement. Les opérations d'accès aux paramètres d'entraînement peuvent échouer suite à des erreurs telles que des valeurs hors plage ou des demandes illicites pour le mode en cours d'un entraînement. L'entraînement génère un code d'erreur qui est renvoyé dans la variable "USS_Extended_Error". Ce code d'erreur ne vaut que pour la dernière exécution d'une instruction USS_RPM ou USS_WPM. Le code d'erreur d'entraînement est placé dans la variable "USS_Extended_Error" lorsque la valeur de STATUS code est 818C hexadécimal. La valeur du code d'erreur de "USS_Extended_Error" dépend du modèle d'entraînement. Le manuel de l'entraînement contient une description des codes des erreurs étendues pour les opérations de lecture et d'écriture de paramètres.

13.7.5 Exigences générales pour la configuration d'un entraînement USS d'héritage

Les exigences générales pour la configuration d'un entraînement USS d'héritage comprennent les points suivants :

- Les entraînements doivent être paramétrés pour utiliser 4 mots PKW.
- Les entraînements peuvent être configurés pour 2, 4, 6 ou 8 mots PZD.
- Le nombre de mots PZD dans l'entraînement doit correspondre à l'entrée PZD_LEN dans l'instruction USS_DRV de cet entraînement.
- La vitesse de transmission dans tous les entraînements doit coïncider avec l'entrée BAUD dans l'instruction USS_PORT .
- La commande à distance doit être activée pour l'entraînement.
- La consigne de fréquence à USS sur la liaison COM doit être sélectionnée pour l'entraînement.
- L'adresse de l'entraînement doit être définie entre 1 et 16 et correspondre à l'entrée DRIVE dans le bloc USS_DRV pour cet entraînement.
- La commande du sens de l'entraînement doit utiliser la polarité de la consigne de l'entraînement.
- Le réseau RS485 doit être muni de résistances de terminaison appropriées.

La configuration et la connexion générales d'un entraînement USS est la même pour les instructions USS (V4.1) et les instructions USS d'héritage (V4.0 et antérieure). Reportez-vous à l'exemple : configuration et connexion générales d'un entraînement USS (Page 1218) pour plus d'informations.

13.8 Communication d'héritage Modbus TCP

13.8.1 Vue d'ensemble

Avant la sortie de STEP 7 V13 SP1 et des CPU S7-1200 V4.1, les instructions de communication Modbus TCP existaient avec des noms différents et, dans certains cas, avec des interfaces légèrement différentes. Les concepts généraux s'appliquent aux deux jeux d'instructions. Reportez-vous aux instructions d'héritage individuelles Modbus TCP pour obtenir des informations relatives à la programmation.

13.8.2 Sélection de la version des instructions Modbus TCP

Trois versions d'instructions Modbus TCP sont disponibles dans STEP 7 :

- Ancienne version 3.0 : compatible avec toutes les versions de CPU et de CP
- Ancienne version 3.1 : compatible avec toutes les versions de CPU et de CP
- Version 4.1 : compatible avec les CPU de version V4.0 et ultérieure et les CM de version V2.1 et ultérieure

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

N'utilisez pas à la fois des instructions de version 3.0 et 3.1 dans le même programme CPU. Les instructions Modbus TCP dans votre programme doivent avoir le même numéro de version principale (1.x, 2.y ou V.z). Toutefois, les instructions individuelles à l'intérieur d'un groupe de version principale peuvent avoir des versions secondaires différentes (1.x).



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.



Pour changer la version des instructions Modbus TCP, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction Modbus TCP dans votre programme, une nouvelle instance de FB est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB dans l'arborescence de projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction Modbus TCP dans un programme, vous devez vérifier les propriétés de l'arborescence du projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB Modbus TCP dans l'arborescence du projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction Modbus TCP.

13.8.3 Instructions d'héritage Modbus TCP

13.8.3.1 MB_CLIENT (Communiquer comme client Modbus TCP via PROFINET)

Tableau 13- 131 Instruction MB_CLIENT


CONT/LOG	SCL	Description
<p>"MB_CLIENT_DB"</p> 	<pre>"MB_CLIENT_DB" (REQ:=_bool_in_, DISCONNECT:=_bool_in_, CONNECT_ID:=_uint_in_, IP_OCTET_1:=_byte_in_, IP_OCTET_2:=_byte_in_, IP_OCTET_3:=_byte_in_, IP_OCTET_4:=_byte_in_, IP_PORT:=_uint_in_, MB_MODE:=_usint_in_, MB_DATA_ADDR:=_udint_in_, MB_DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_DATA_PTR:=_variant_inout_);</pre>	<p>MB_CLIENT permet de communiquer en tant que client Modbus TCP par l'intermédiaire du connecteur PROFINET sur la CPU S7-1200. Aucun module matériel de communication supplémentaire n'est nécessaire.</p> <p>MB_CLIENT permet d'établir une liaison client-serveur, d'envoyer une demande de fonction Modbus, de recevoir une réponse et de gérer la déconnexion d'un serveur Modbus TCP.</p>

Tableau 13- 132 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	In	Bool	FALSE = Aucune demande de communication Modbus TRUE = Demande de communication avec un serveur Modbus TCP
DISCONNECT	IN	Bool	Le paramètre DISCONNECT permet à votre programme de gérer la connexion et la déconnexion à un serveur Modbus. Si DISCONNECT est égal à 0 et qu'il n'existe pas de liaison, MB_CLIENT tente d'établir une liaison à l'adresse et au numéro de port IP affectés. Si DISCONNECT est égal à 1 et qu'une liaison existe, une opération de déconnexion est tentée. Aucune autre opération n'est tentée lorsque cette entrée est activée.
CONNECT_ID	IN	UInt	Le paramètre CONNECT_ID doit identifier de manière unique chaque liaison dans l'API. Chaque instance unique de l'instruction MB_CLIENT ou MB_SERVER doit contenir un paramètre CONNECT_ID unique.
IP_OCTET_1	IN	USInt	Adresse IP du serveur Modbus TCP : Octet 1 8 bits de l'adresse IPv4 de 32 bits du serveur Modbus TCP auquel le client se connectera et avec lequel il communiquera à l'aide du protocole Modbus TCP.
IP_OCTET_2	IN	USInt	Adresse IP du serveur Modbus TCP : Octet 2
IP_OCTET_3	IN	USInt	Adresse IP du serveur Modbus TCP : Octet 3
IP_OCTET_4	IN	USInt	Adresse IP du serveur Modbus TCP : Octet 4
IP_PORT	IN	UInt	Valeur par défaut = 502 : Numéro de port IP du serveur auquel le client tentera de se connecter dans le but de communiquer à l'aide du protocole TCP/IP.
MB_MODE	IN	USInt	Sélection du mode : Définit le type de demande (lecture, écriture ou diagnostic). Voir le tableau des fonctions Modbus ci-après pour plus de détails.
MB_DATA_ADDR	IN	UDInt	Adresse de début Modbus : Définit l'adresse de début des données auxquelles on accédera via MB_CLIENT. Voir les adresses valides dans le tableau des fonctions Modbus ci-après.
MB_DATA_LEN	IN	UInt	Longueur des données Modbus : Définit le nombre de bits ou mots auxquels accéder dans cette demande. Voir les longueurs valides dans le tableau des fonctions Modbus ci-après.
MB_DATA_PTR	IN_OUT	Variant	Pointeur désignant le registre de données Modbus : Le registre stocke temporairement les données allant vers un serveur Modbus ou venant d'un serveur Modbus. Le pointeur doit désigner un DB global non optimisé ou une adresse en mémoire M.
DONE	OUT	Bool	Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • 0 : Pas d'opération MB_CLIENT en cours • 1 : Opération MB_CLIENT en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque l'exécution de MB_CLIENT s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

Paramètre REQ

FALSE = Aucune demande de communication Modbus

TRUE = Demande de communication avec un serveur Modbus TCP

Si aucune instance de MB_CLIENT n'est active et que le paramètre DISCONNECT est égal à 0, une nouvelle demande Modbus sera lancée si REQ est égal à 1. Si la liaison n'est pas déjà établie, une nouvelle liaison le sera.

Si la même instance de MB_CLIENT est à nouveau exécutée avec DISCONNECT=0 et REQ=1, et ce avant l'achèvement de la demande en cours, il n'y aura pas d'émission Modbus suivante. Toutefois, dès que la demande en cours aura été achevée, une nouvelle demande pourra être traitée si MB_CLIENT est exécuté avec l'entrée REQ à 1.

Lorsque la demande de communication MB_CLIENT en cours est achevée, le bit DONE est TRUE pour un cycle. Le bit DONE peut être utilisé comme porte temporelle pour ordonner plusieurs demandes MB_CLIENT.

Remarque

Cohérence des données d'entrée pendant le traitement de MB_CLIENT

Lorsqu'un client Modbus a déclenché une opération Modbus, l'état de toutes les entrées est sauvegardé en interne, puis comparé à chaque appel successif. Cette comparaison sert à déterminer si cet appel particulier était à l'origine de la demande client active. Plusieurs appels MB_CLIENT peuvent être effectués à l'aide d'un DB d'instance commun.

En conséquence, il est important que les entrées ne soient pas modifiées tant qu'une opération MB_CLIENT est traitée activement. Si cette règle n'est pas respectée, un appel MB_CLIENT ne peut pas déterminer qu'il est l'instance active.

Sélection de la fonction de communication Modbus via les paramètres MB_MODE et MB_DATA_ADDR

MB_DATA_ADDR définit l'adresse Modbus de début des données auxquelles accéder. L'instruction MB_CLIENT utilise une entrée MB_MODE et non une entrée de code de fonction.

La combinaison des valeurs MB_MODE et MB_DATA_ADDR détermine le code de fonction qui est utilisé dans le message Modbus réel. Le tableau suivant montre la correspondance entre le paramètre MB_MODE, la fonction Modbus et la plage d'adresses Modbus.

Tableau 13- 133 Fonctions Modbus

MB_MODE	Fonction Modbus	Longueur des données	Opération et données	MB_DATA_ADDR
0	01	1 à 2000	Lecture de bits de sortie : 1 à 2000 bits par demande	1 à 9999
0	02	1 à 2000	Lecture de bits d'entrée : 1 à 2000 bits par demande	10001 à 19999
0	03	1 à 125	Lecture de registres de maintien : 1 à 125 mots par demande	40001 à 49999 ou 400001 à 465535
0	04	1 à 125	Lecture de mots d'entrée : 1 à 125 mots par demande	30001 à 39999
1	05	1	Ecriture d'un bit de sortie : 1 bit par demande	1 à 9999
1	06	1	Ecriture d'un registre de maintien : 1 mot par demande	40001 à 49999 ou 400001 à 465535
1	15	2 à 1968	Ecriture de plusieurs bits de sortie : 2 à 1968 bits par demande	1 à 9999
1	16	2 à 123	Ecriture de plusieurs registres de maintien : 2 à 123 mots par demande	40001 à 49999 ou 400001 à 465535
2	15	1 à 1968	Ecriture d'un ou de plusieurs bits de sortie : 1 à 1968 bits par demande	1 à 9999
2	16	1 à 123	Ecriture d'un ou de plusieurs registres de maintien : 1 à 123 mots par demande	40001 à 49999 ou 400001 à 465535
11	11	0	Lecture du mot d'état de communication serveur et du compteur d'événements. Le mot d'état indique si le serveur est occupé (0 : non occupé, 0xFFFF : occupé). Le compteur d'événements est incrémenté à chaque fois qu'un message s'achève avec succès. Les deux paramètres MB_DATA_ADDR et MB_DATA_LEN de MB_CLIENT ne sont pas pris en compte pour cette fonction.	
80	08	1	Contrôle de l'état du serveur à l'aide du code de diagnostic de données 0x0000 (test de retour : le serveur renvoie la demande en écho) 1 mot par demande	

MB_MODE	Fonction Modbus	Longueur des données	Opération et données	MB_DATA_ADDR
81	08	1	Réinitialisation du compteur d'événements du serveur à l'aide du code de diagnostic de données 0x000A 1 mot par demande	
3 à 10, 12 à 79, 82 à 255			Réservé	

Remarque

MB_DATA_PTR définit une mémoire tampon pour stocker les données lues/écrites dans un serveur Modbus TCP

La mémoire tampon de données peut être dans un DB global non optimisé ou à une adresse de mémoire M.

Si la mémoire tampon est en mémoire M, utilisez le format de pointeur standard Any, à savoir P#"adresse bit" "type de données" "longueur" (par exemple, P#M1000.0 WORD 500).

Définition d'une mémoire tampon de communication avec MB_DATA_PTR

- Fonctions de communication MB_CLIENT :
 - Lecture et écriture de données d'un bit dans des adresses de serveur Modbus (00001 à 09999)
 - Lecture de données d'un bit dans des adresses de serveur Modbus (10001 à 19999)
 - Lecture de données mots de 16 bits dans des adresses de serveur Modbus (30001 à 39999) et (40001 à 49999)
 - Ecriture de données mots de 16 bits dans des adresses de serveur Modbus (40001 à 49999)
- Les données mots ou bits sont transférées vers/depuis la mémoire tampon de DB ou de mémoire M définie par MB_DATA_PTR.
- Si vous avez défini un DB comme mémoire tampon avec MB_DATA_PTR, vous devez affecter un type à tous les éléments de données du DB.
 - Le type de données Bool de 1 bit représente une adresse de bit Modbus.
 - Les types de données de 16 bits (mot unique), tels que WORD, UInt et Int, représentent une adresse de mot Modbus.
 - Les types de données de 32 bits (double mot), tels que DWORD, DInt et Real, représentent deux adresses de mot Modbus.

- MB_DATA_PTR permet d'indiquer des éléments de DB complexes, tels que :
 - tableaux standard
 - structures nommées où chaque élément est unique
 - structures complexes nommées où chaque élément a un nom unique et un type de données de 16 ou 32 bits
- Il n'est pas obligatoire que les zones de données MB_DATA_PTR soient dans le même bloc de données global (ou la même zone de mémoire M). Vous pouvez indiquer un bloc de données pour les lectures Modbus, un autre bloc de données pour les écritures Modbus ou un bloc de données pour chaque station MB_CLIENT.

Liaisons client multiples

Un client Modbus TCP peut accepter des liaisons simultanées jusqu'au nombre maximum de liaisons de communication ouverte autorisé par l'API. Le nombre total de liaisons pour un API, clients et serveurs Modbus TCP compris, ne doit pas dépasser le nombre maximum de liaisons de communication ouverte prises en charge (Page 827). Les liaisons Modbus TCP peuvent être partagées entre liaisons de type client et/ou serveur.

Les liaisons client individuelles doivent respecter les règles suivantes :

- Chaque liaison MB_CLIENT doit utiliser un DB d'instance unique.
- Chaque liaison MB_CLIENT doit indiquer une adresse IP de serveur unique.
- Chaque liaison MB_CLIENT doit indiquer un ID de liaison unique.
- Des numéros de port IP uniques peuvent ou non être obligatoires selon la configuration du serveur.

L'ID de liaison doit être unique pour chaque liaison individuelle. Cela signifie qu'un ID de liaison unique ne doit être utilisé qu'avec le DB d'instance individuel correspondant. En résumé, le DB d'instance et l'ID de liaison forment une paire qui doit être unique pour chaque liaison.

Tableau 13- 134 Variables statiques accessibles à l'utilisateur du bloc de données d'instance MB_CLIENT

Variable	Type de données	Valeur par défaut	Description
Blocked_Proc_Timeout	Real	3.0	Durée (en secondes) pendant laquelle attendre une instance de client Modbus bloquée avant d'annuler l'état ACTIF de cette instance. Cela peut, par exemple, se produire lorsqu'une demande client a été émise et que l'application arrête d'exécuter la fonction client avant l'achèvement complet de la demande. La limite maximale du S7-1200 est de 55 secondes.
MB_Unit_ID	Word	255	Identificateur d'unité Modbus : On accède à un serveur Modbus TCP à l'aide de son adresse IP. Le paramètre MB_UNIT_ID n'est donc pas utilisé pour l'adressage Modbus TCP. Le paramètre MB_UNIT_ID correspond à l'adresse d'esclave dans le protocole Modbus RTU. Si un serveur Modbus TCP est utilisé comme passerelle vers un protocole Modbus RTU, on peut utiliser MB_UNIT_ID pour identifier l'esclave connecté sur le réseau série. On utiliserait MB_UNIT_ID pour transmettre la demande à l'adresse d'esclave Modbus RTU correcte. Certains appareils Modbus TCP peuvent nécessiter le paramètre MB_UNIT_ID afin d'être initialisés à l'intérieur d'une plage de valeurs restreinte.
RCV_TIMEOUT	Real	2.0	Durée en secondes pendant laquelle MB_CLIENT attend qu'un serveur réponde à une demande
Connected	Bool	0	Indique si la liaison au serveur affecté est établie ou coupée : 1=liaison établie, 0=liaison coupée

Tableau 13- 135 Erreurs de protocole MB_CLIENT :

STATUS (W#16#)	Code réponse envoyé au client Modbus (B#16#)	Erreurs de protocole Modbus
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou accès en dehors des limites de la zone d'adresses de MB_HOLD_REG
8384	03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non pris en charge (code de fonction 08)

13.8 Communication d'héritage Modbus TCP

Tableau 13- 136 Codes d'erreur d'exécution de MB_CLIENT ¹

STATUS (W#16#)	Erreurs de paramètres MB_CLIENT
7001	MB_CLIENT attend une réponse du serveur Modbus à une demande de connexion ou de déconnexion sur le port TCP affecté. Ce code n'est envoyé qu'à la première exécution d'une opération de connexion ou de déconnexion.
7002	MB_CLIENT attend une réponse du serveur Modbus à une demande de connexion ou de déconnexion pour le port TCP affecté. Ce code est envoyé pour les exécutions suivantes en attendant l'achèvement d'une opération de connexion ou de déconnexion.
7003	Une opération de déconnexion s'est achevée avec succès (valable pendant un seul cycle API).
80C8	Le serveur n'a pas répondu dans le temps imparti. MB_CLIENT doit recevoir une réponse avec l'ID de transaction initialement transmis dans le temps imparti ; sinon, cette erreur est renvoyée. Vérifiez la liaison au serveur Modbus. Cette erreur n'est signalée qu'après que le nombre configuré de nouvelles tentatives (le cas échéant) a été tenté.
8188	Valeur de mode invalide
8189	Valeur d'adresse de données invalide
818A	Valeur de longueur de données invalide
818B	Pointeur invalide sur la zone DATA_PTR. Il peut s'agir de la combinaison de MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Pointeur vers une zone DATA_PTR optimisée (ce doit être une zone DB non optimisée ou une zone de mémoire M).
8200	Le port est occupé à traiter une demande Modbus existante.
8380	La trame Modbus reçue est mal formée ou trop peu d'octets ont été reçus.
8387	Le paramètre ID de liaison défini est différent de l'ID utilisé pour les demandes précédentes. Un seul ID de liaison peut être utilisé dans chaque DB d'instance MB_CLIENT. Ce code est également utilisé comme erreur interne si l'ID de protocole Modbus TCP reçu d'un serveur est différent de 0.
8388	Un serveur Modbus a renvoyé une quantité de données différente de celle demandée. Ce code s'applique aux fonctions Modbus 15 et 16 uniquement.

¹ En plus des erreurs MB_CLIENT présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication par blocs T sous-jacentes (TCON, TDISCON, TSEND et TRCV (Page 885)).

13.8.3.2 MB_SERVER (Communiquer comme serveur Modbus TCP via PROFINET)

L'instruction MB_SERVER permet de communiquer en tant que serveur Modbus TCP par l'intermédiaire du connecteur PROFINET sur la CPU S7-1200. L'instruction "MB_SERVER" traite les demandes de connexion d'un client Modbus TCP, reçoit et traite les demandes Modbus, et envoie les réponses.

Pour utiliser l'instruction, vous n'avez pas besoin d'un module matériel supplémentaire.

IMPORTANT

Notes relatives à la sécurité

A noter que chaque client du réseau bénéficie d'un accès en lecture et en écriture aux entrées et sorties de la mémoire image ainsi qu'au bloc de données ou à la zone des mémentos définie par le registre de maintien Modbus.

Il existe la possibilité de limiter l'accès à une adresse IP pour empêcher les opérations de lecture et d'écriture non autorisées. A noter, toutefois, que l'adresse partagée peut également être utilisée pour un accès non autorisé.

Tableau 13- 137 Instruction MB_SERVER

CONT/LOG	SCL	Description
	<pre>"MB_SERVER_DB" (DISCONNECT:=_bool_in_, CONNECT_ID:=_uint_in_, IP_PORT:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	<p>MB_SERVER permet de communiquer en tant que serveur Modbus TCP par l'intermédiaire du connecteur PROFINET sur la CPU S7-1200. Aucun module matériel de communication supplémentaire n'est nécessaire.</p> <p>MB_SERVER peut accepter une demande de connexion à un client Modbus TCP, recevoir une demande de fonction Modbus et envoyer un message de réponse.</p>

Tableau 13- 138 Types de données pour les paramètres

Paramètre et type		Type de données	Description
DISCONNECT	IN	Bool	MB_SERVER tente d'établir une liaison "passive" à un dispositif partenaire. Cela signifie que le serveur est passivement à l'écoute d'une demande de connexion TCP provenant de toute adresse IP demandeuse. Si DISCONNECT est égal à 0 et qu'il n'existe pas de liaison, une liaison passive peut être établie. Si DISCONNECT est égal à 1 et qu'une liaison existe, une opération de déconnexion est lancée. Cela permet à votre programme de gérer le moment où une liaison est acceptée. Aucune autre opération n'est tentée lorsque cette entrée est activée.
CONNECT_ID	IN	UInt	Le paramètre CONNECT_ID identifie de manière unique chaque liaison dans l'API. Chaque instance unique de l'instruction MB_CLIENT ou MB_SERVER doit contenir un paramètre CONNECT_ID unique.
IP_PORT	IN	UInt	Valeur par défaut = 502 : Numéro qui identifie le port IP qui sera surveillé dans l'attente d'une demande de connexion provenant d'un client Modbus. Les numéros de port TCP suivants ne sont pas autorisés pour une liaison passive MB_SERVER : 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 et 34964.
MB_HOLD_REG	IN_OUT	Variant	Pointeur désignant le registre de maintien Modbus pour MB_SERVER. Le registre de maintien peut être soit un DB global non optimisé, soit une adresse en mémoire M. Cette zone de mémoire sert à stocker les valeurs auxquelles un client Modbus a le droit d'accéder à l'aide des fonctions Modbus de registre 3 (lecture), 6 (écriture) et 16 (écriture).
NDR	OUT	Bool	Nouvelles données prêtes (New Data Ready) : 0 = Pas de nouvelles données, 1 = Indique que de nouvelles données ont été écrites par un client Modbus
DR	OUT	Bool	Données lues : 0 = Pas de données lues, 1 = Indique que des données ont été lues par un client Modbus
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque l'exécution de MB_SERVER s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

MB_SERVER permet à des fonctions Modbus entrantes de codes 1, 2, 4, 5 et 15 de lire ou d'écrire des bits et des mots directement dans la mémoire image des entrées et la mémoire image des sorties de la CPU S7-1200. Pour les fonctions de transfert de données de codes 3, 6 et 16, le paramètre MB_HOLD_REG doit être défini avec un type de données supérieur à un octet. Le tableau suivant montre la correspondance entre les adresses Modbus et la mémoire image dans la CPU.

Tableau 13- 139 Correspondance entre adresses Modbus et mémoire image

Fonctions Modbus					S7-1200		
Codes	Fonction	Zone de données	Plage d'adresses			Zone de données	Adresse de la CPU
01	Lecture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
02	Lecture de bits	Entrée	10001	à	18192	Mémoire image des entrées	I0.0 à I1023.7
04	Lecture de mots	Entrée	30001	à	30512	Mémoire image des entrées	IW0 à IW1022
05	Écriture d'un bit	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
15	Écriture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7

Les fonctions de message Modbus entrantes de codes 3, 6 et 16 lisent ou écrivent des mots dans un registre de maintien Modbus qui peut être une plage d'adresses en mémoire M ou un bloc de données. Le type de registre de maintien est indiqué par le paramètre MB_HOLD_REG.

Remarque

Paramétrage de MB_HOLD_REG

Le registre de maintien Modbus peut se situer dans un DB global non optimisé ou à une adresse de mémoire M.

Si le registre de maintien Modbus est en mémoire M, utilisez le format de pointeur standard Any, à savoir P#"adresse bit" "type de données" "longueur" (par exemple, P#M1000.0 WORD 500).

Le tableau suivant montre des exemples de correspondance entre adresses Modbus et registre de maintien pour les codes de fonction Modbus 03 (lecture de mots), 06 (écriture d'un mot) et 16 (écriture de mots). La limite supérieure effective des adresses de DB dépend des limites de mémoire de travail et de mémoire M maximum pour chaque modèle de CPU.

Tableau 13- 140 Exemples de correspondance entre adresse Modbus et adresse de mémoire CPU

Adresse Modbus	Exemples pour le paramètre MB_HOLD_REG		
	P#M100.0 Word 5	P#DB10.DBx0.0 Word 5	"Recette".ingredient
40001	MW100	DB10.DBW0	"Recette".ingredient[1]
40002	MW102	DB10.DBW2	"Recette".ingredient[2]
40003	MW104	DB10.DBW4	"Recette".ingredient[3]
40004	MW106	DB10.DBW6	"Recette".ingredient[4]
40005	MW108	DB10.DBW8	"Recette".ingredient[5]

Liaisons serveur multiples

Plusieurs liaisons serveur peuvent être créées. Cela permet à un API unique d'établir des liaisons simultanées à plusieurs clients Modbus TCP.

Un serveur Modbus TCP peut accepter des liaisons simultanées jusqu'au nombre maximum de liaisons de communication ouverte autorisé par l'API. Le nombre total de liaisons pour un API, clients et serveurs Modbus TCP compris, ne doit pas dépasser le nombre maximum de liaisons de communication ouverte prises en charge (Page 827). Les liaisons Modbus TCP peuvent être partagées entre liaisons de type client et/ou serveur.

Les liaisons serveur individuelles doivent respecter les règles suivantes :

- Chaque liaison MB_SERVER doit utiliser un DB d'instance unique.
- Chaque liaison MB_SERVER doit être établie avec un numéro de port IP unique. Une seule liaison est acceptée par port.
- Chaque liaison MB_SERVER doit utiliser un ID de liaison unique.
- L'instruction MB_SERVER doit être appelée individuellement pour chaque liaison (avec son DB d'instance respectif).

L'ID de liaison doit être unique pour chaque liaison individuelle. Cela signifie qu'un ID de liaison unique ne doit être utilisé qu'avec le DB d'instance individuel correspondant. En résumé, le DB d'instance et l'ID de liaison forment une paire qui doit être unique pour chaque liaison.

Tableau 13- 141 Codes de fonction de diagnostic Modbus

Fonctions de diagnostic Modbus pour MB_SERVER		
Codes	Sous-fonction	Description
08	0x0000	Renvoi d'un test d'écho des données de requête : MB_SERVER renvoie en écho à un client Modbus un mot de données reçu.
08	0x000A	Effacement du compteur d'événements de communication : MB_SERVER efface le compteur d'événements de communication qui est utilisé pour la fonction Modbus 11.
11		Lecture du compteur d'événements de communication : MB_SERVER utilise un compteur d'événements de communication interne pour enregistrer le nombre de demandes de lecture et d'écriture Modbus envoyées au serveur Modbus qui ont abouti. Le compteur ne s'incrémente pas pour les fonctions 8 et 11. Il ne s'incrémente pas non plus pour toute demande entraînant une erreur de communication. La fonction de diffusion générale n'est pas disponible pour Modbus TCP, car une seule liaison client-serveur existe à un moment donné.

Variables MB_SERVER

Ce tableau montre les variables statiques publiques sauvegardées dans le bloc de données d'instance pour MB_SERVER pouvant être utilisées dans votre programme.

Tableau 13- 142 Variables statiques publiques pour MB_SERVER

Variable	Type de données	Valeur par défaut	Description
HR_Start_Offset	Word	0	Définit l'adresse de début du registre de maintien Modbus.
Request_Count	Word	0	Nombre de toutes les demandes reçues par ce serveur
Server_Message_Count	Word	0	Nombre de demandes reçues pour ce serveur spécifique
Xmt_Rcv_Count	Word	0	Nombre d'émissions ou de réceptions pendant lesquelles une erreur s'est produite. Cette variable est également incrémentée si un message reçu est un message Modbus invalide.
Exception_Count	Word	0	Erreurs Modbus spécifiques nécessitant le renvoi d'une exception
Success_Count	Word	0	Nombre de demandes reçues pour ce serveur spécifique qui ne contient pas d'erreurs de protocole.
Connected	Bool	0	Indique si la liaison au client affecté est établie ou coupée : 1=liaison établie, 0=liaison coupée

Votre programme peut écrire des valeurs dans la variable HR_Start_Offset et commander le fonctionnement du serveur Modbus. Les autres variables peuvent être lues pour visualiser l'état de Modbus.

HR_Start_Offset

Les adresses de registre de maintien Modbus commencent à 40001, ce qui correspond à l'adresse de début du registre de maintien dans la mémoire API. Vous pouvez toutefois configurer la variable "HR_Start_Offset" afin que le registre de maintien Modbus commence à une valeur autre que 40001.

Par exemple, si le registre de maintien est configuré pour commencer à MW100 et s'il a une longueur de 100 mots. Un décalage de 20 indique une adresse de début de registre située à 40021 au lieu de 40001. Toute adresse inférieure à 40021 et supérieure à 40119 entraînera une erreur d'adressage.

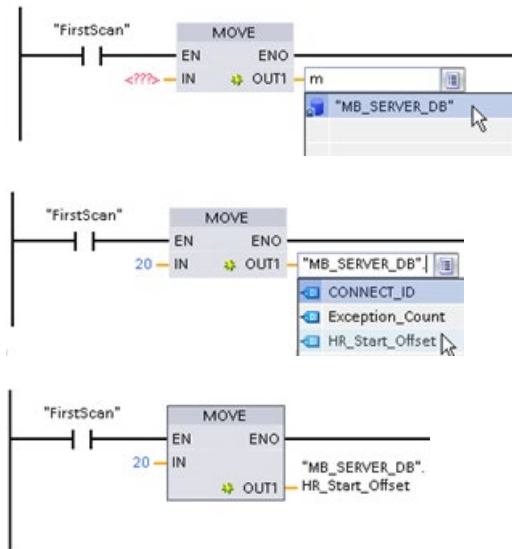
Tableau 13- 143 Exemple d'adressage de registre de maintien Modbus

HR_Start_Offset	Adresse	Minimum	Maximum
0	Adresse Modbus (mot)	40001	40099
	Adresse S7-1200	MW100	MW298
20	Adresse Modbus (mot)	40021	40119
	Adresse S7-1200	MW100	MW298

HR_Start_Offset est une valeur de mot qui indique l'adresse de début du registre de maintien Modbus et est sauvegardée dans le bloc de données d'instance de MB_SERVER. Vous pouvez définir la valeur de cette variable statique publique à l'aide de la liste déroulante d'aide pour les paramètres lorsque vous avez inséré MB_SERVER dans votre programme.

13.8 Communication d'héritage Modbus TCP

Par exemple, une fois MB_SERVER inséré dans un réseau CONT, vous pouvez aller à un réseau précédent et définir la valeur HR_Start_Offset. La valeur doit être affectée avant l'exécution de MB_SERVER.



Entrée d'une variable pour le serveur Modbus à l'aide du nom de DB par défaut :

1. Placez le curseur dans le champ de paramètre et tapez la lettre m.
2. Sélectionnez "MB_SERVER_DB" dans la liste déroulante des noms de DB.
3. Sélectionnez "MB_SERVER_DB.HR_Start_Offset" dans la liste déroulante des variables de DB.

Tableau 13- 144 Codes d'erreur d'exécution de MB_SERVER ¹

STATUS (W#16#)	Code réponse envoyé au serveur Modbus (B#16#)	Erreurs de protocole Modbus
7001		MB_SERVER attend qu'un client Modbus se connecte au port TCP affecté. Ce code est envoyé à la première exécution d'une opération de connexion ou de déconnexion.
7002		MB_SERVER attend qu'un client Modbus se connecte au port TCP affecté. Ce code est envoyé pour les exécutions suivantes en attendant l'achèvement d'une opération de connexion ou de déconnexion.
7003		Une opération de déconnexion s'est achevée avec succès (valable pendant un seul cycle API).
8187		Pointeur invalide sur MB_HOLD_REG : la zone est trop petite.
818C		Pointeur désignant une zone MB_HOLD_REG optimisée (ce doit être une zone DB non optimisée ou une zone de mémoire M) ou le délai d'attente de processus bloqué dépasse la limite de 55 secondes. (spécifique au S7-1200)
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou accès en dehors des limites de la zone d'adresses de MB_HOLD_REG
8384	03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non pris en charge (code de fonction 08)

¹ En plus des erreurs MB_SERVER présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication par blocs T sous-jacentes (TCON, TDISCON, TSEND et TRCV (Page 885)).

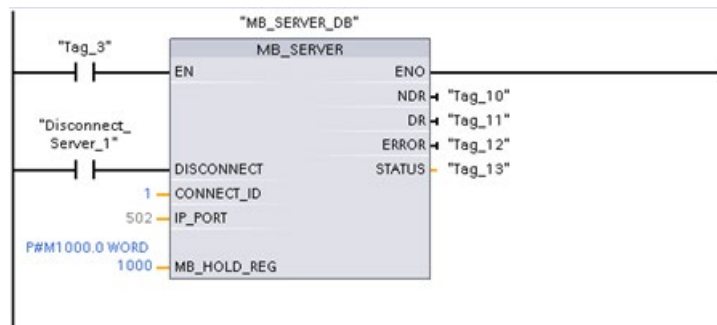
13.8.4 Exemples d'héritage Modbus TCP

13.8.4.1 Exemple : Liaisons TCP multiples MB_SERVER d'héritage

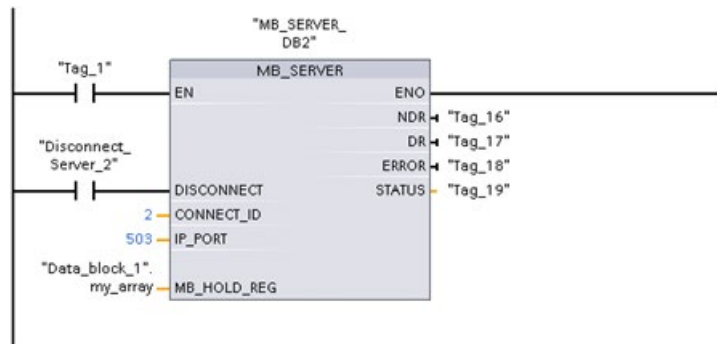
Vous pouvez avoir plusieurs liaisons serveur Modbus TCP. Pour ce faire, vous devez exécuter MB_SERVER de manière indépendante pour chaque liaison. Chaque liaison doit utiliser un DB d'instance, un ID de liaison et un port IP qui lui soient propres. Le S7-1200 n'admet qu'une seule liaison par port IP.

Pour des performances optimales, il est préférable d'exécuter MB_SERVER à chaque cycle de programme pour chaque liaison.

Réseau 1 : Liaison #1 avec port IP indépendant, ID de liaison et DB d'instance en propre



Réseau 2 : Liaison #2 avec port IP indépendant, ID de liaison et DB d'instance en propre



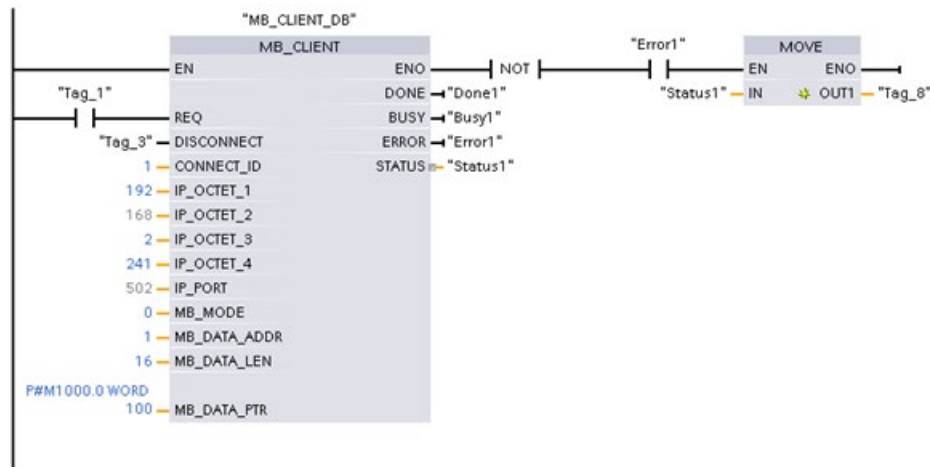
13.8.4.2 Exemple : MB_CLIENT 1 d'héritage : Plusieurs demandes avec une liaison TCP commune

Vous pouvez envoyer plusieurs demandes client Modbus par la même liaison. Pour ce faire, utilisez les mêmes DB d'instance, ID de liaison et numéro de port.

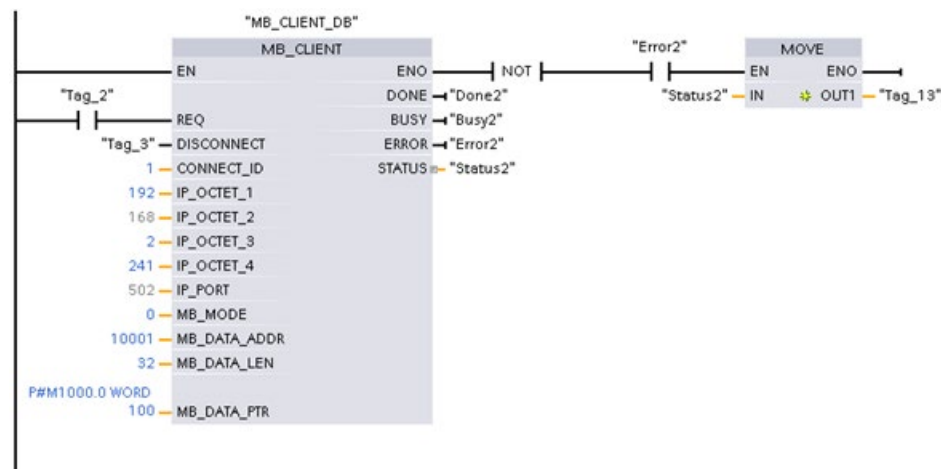
Un seul client peut être actif à un instant donné. Une fois qu'un client a achevé son exécution, le client suivant commence à s'exécuter. C'est votre programme qui gère l'ordre d'exécution.

L'exemple montre deux clients qui écrivent dans la même zone de mémoire. En outre, une erreur renvoyée est détectée, ce qui est facultatif.

Réseau 1 : Fonction Modbus 1 : Lecture de 16 bits de la mémoire image des sorties



Réseau 2 : Fonction Modbus 2 : Lecture de 32 bits de la mémoire image des entrées



13.8.4.3 Exemple : MB_CLIENT 2 d'héritage : Plusieurs demandes avec des liaisons TCP différentes

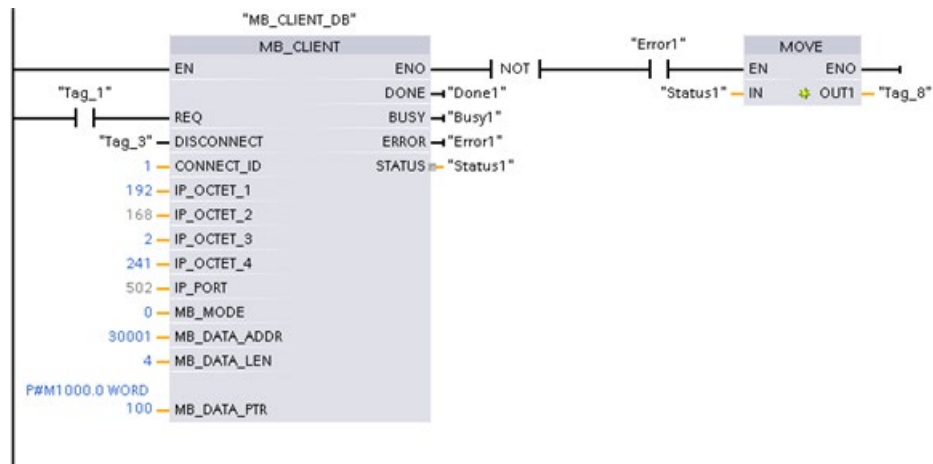
Vous pouvez envoyer plusieurs demandes client Modbus par le biais de liaisons différentes. Pour ce faire, vous devez utiliser des DB d'instance, des adresses IP et des ID de liaison différents.

Le numéro de port doit être différent si les liaisons sont établies vers le même serveur Modbus. Si les liaisons concernent des serveurs différents, aucune restriction ne s'applique au numéro de port.

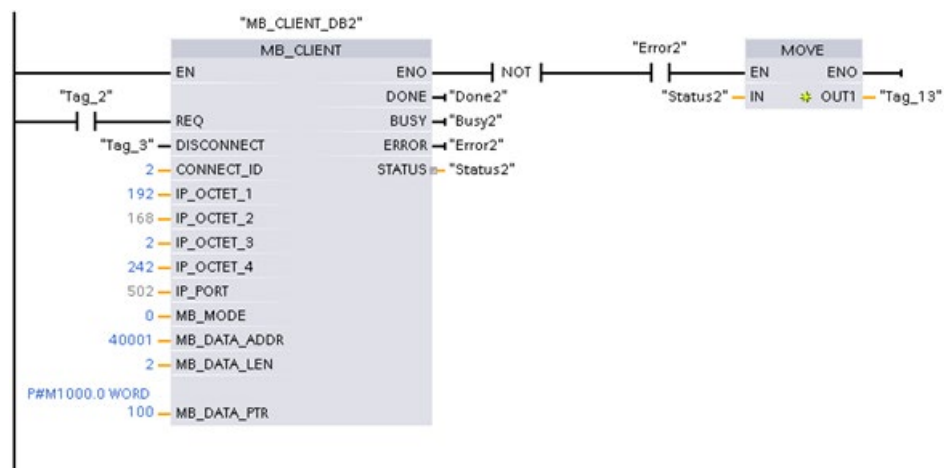
L'exemple montre deux clients qui écrivent dans la même zone de mémoire. En outre, une erreur renvoyée est détectée, ce qui est facultatif.

Réseau 1 :

Fonction Modbus 4 : Lecture de mots d'entrée (dans la mémoire du S7-1200)



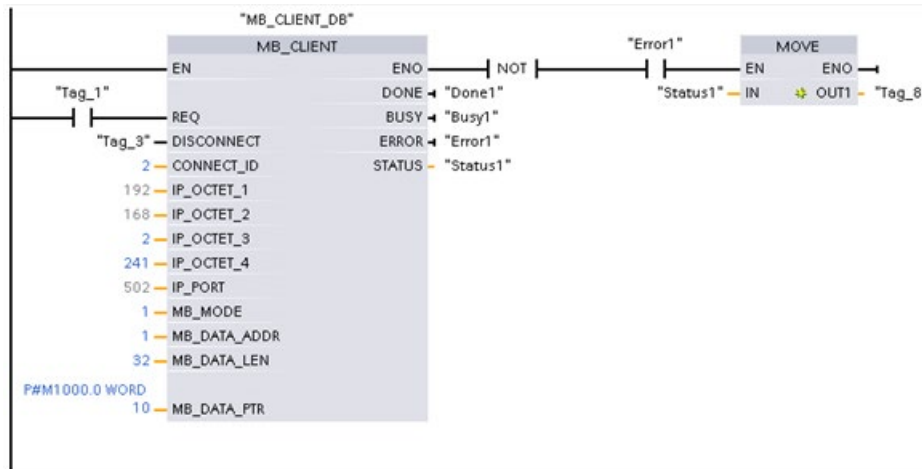
Réseau 2 : Fonction Modbus 3 - Lecture de mots d'un registre de maintien d'un serveur Modbus TCP



13.8.4.4 Exemple : MB_CLIENT 3 d'héritage : Demande d'écriture dans la mémoire image des sorties

Cet exemple montre une demande client Modbus pour l'écriture dans la mémoire image des sorties du S7-1200.

Réseau 1 : Fonction Modbus 15 : Ecriture d'octets dans la mémoire image des sorties du S7-1200

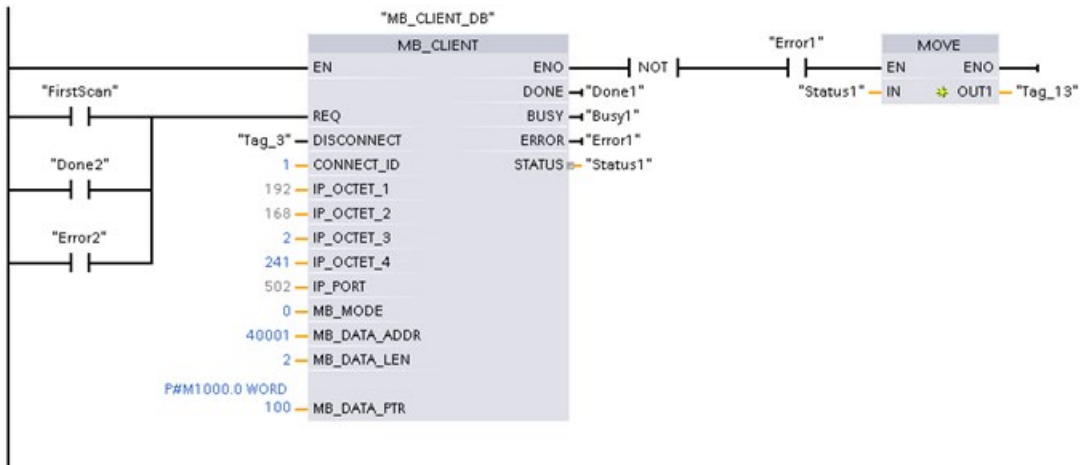


13.8.4.5 Exemple : MB_CLIENT 4 d'héritage : Coordination de plusieurs demandes

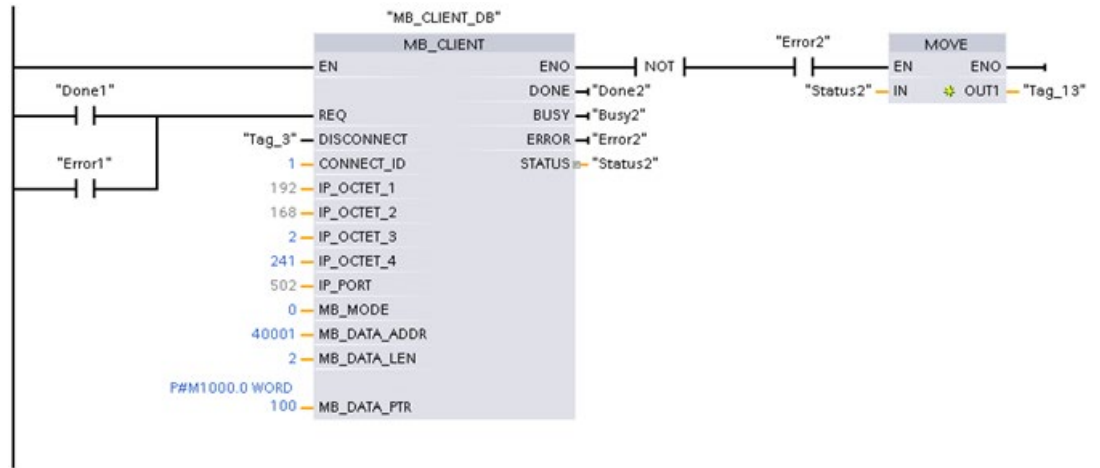
Vous devez veiller à ce que chaque demande Modbus TCP individuelle achève son exécution. Cette coordination doit être assurée par votre programme. L'exemple ci-dessous montre comment les sorties de la première et de la deuxième demande client peuvent être utilisées pour coordonner l'exécution.

L'exemple montre deux clients qui écrivent dans la même zone de mémoire. En outre, une erreur renvoyée est détectée, ce qui est facultatif.

Réseau 1 : Fonction Modbus 3 : Lecture de mots d'un registre de maintien



Réseau 2 : Fonction Modbus 3 : Lecture de mots d'un registre de maintien



13.9 Communication d'héritage Modbus RTU (CM/CB 1241 uniquement)

13.9.1 Vue d'ensemble

Avant la sortie de STEP 7 V13 SP1 et des CPU S7-1200 V4.1, les instructions de communication Modbus RTU existaient avec des noms différents et, dans certains cas, avec des interfaces légèrement différentes. Les concepts généraux s'appliquent aux deux jeux d'instructions. Reportez-vous aux instructions d'héritage individuelles Modbus RTU pour obtenir des informations relatives à la programmation.

13.9.2 Sélection de la version des instructions Modbus RTU

Deux versions d'anciennes instructions Modbus RTU sont disponibles dans STEP 7 :

- Ancienne version 1.3 : compatible avec toutes les versions de CPU et de CP
- Ancienne version 2.2 : compatible avec toutes les versions de CPU et de CP

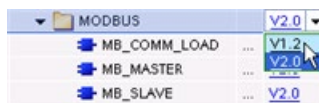
(Remarque : dans la version 2.2, des paramètres REQ et DONE sont ajoutés à MB_COMM_LOAD. En outre, le paramètre MB_ADDR pour MB_MASTER et MB_SLAVE autorise maintenant les valeurs UInt pour l'adressage étendu)

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

Vous ne pouvez pas utiliser les deux versions des instructions avec le même module, mais deux modules différents peuvent utiliser différentes versions des instructions. Les instructions Modbus RTU dans votre programme doivent avoir le même numéro de version principale (1.x, 2.y, ou V.z). Toutefois, les instructions individuelles à l'intérieur d'un groupe de version principale peuvent avoir des versions secondaires différentes (1.x).



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.



Pour changer la version des instructions Modbus, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

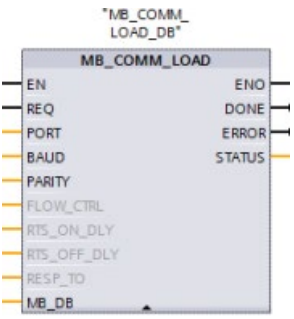
Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction Modbus dans votre programme, une nouvelle instance de FB est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB dans l'arborescence de projet sous PLC_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction Modbus dans un programme, vous devez vérifier les propriétés de l'arborescence de projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB Modbus dans l'arborescence de projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction Modbus.

13.9.3 Instructions d'héritage Modbus RTU

13.9.3.1 MB_COMM_LOAD (Configurer le port sur le module PtP pour Modbus RTU)

Tableau 13- 145 Instruction MB_COMM_LOAD

CONT/LOG	SCL	Description
	<pre>"MB_COMM_LOAD_DB" (REQ:=_bool_in_, PORT:=_uint_in_, BAUD:=_uint_in_, PARITY:=_uint_in_, FLOW_CTRL:=_uint_in_, RTS_ON_DLY:=_uint_in_, RTS_OFF_DLY:=_uint_in_, RESP_TO:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_DB:=_fbtref_inout_);</pre>	<p>L'instruction MB_COMM_LOAD configure un port point à point pour la communication via le protocole Modbus RTU. Options matérielles pour le port Modbus : Installez jusqu'à trois CM (RS485 ou RS232), ainsi qu'un CB (RS485). Un bloc de données d'instance est automatiquement affecté lorsque vous insérez l'instruction MB_COMM_LOAD dans votre programme.</p>

13.9 Communication d'héritage Modbus RTU (CM/CB 1241 uniquement)

Tableau 13- 146 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (version 2.0 uniquement)
PORT	IN	Port	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
BAUD	IN	UDInt	Sélection du débit : 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Toutes les autres valeurs sont invalides.
PARITY	IN	UInt	Sélection de la parité : <ul style="list-style-type: none"> • 0 = Pas de parité • 1 = Parité impaire • 2 = Parité paire
FLOW_CTRL ¹	IN	UInt	Sélection du contrôle de flux : <ul style="list-style-type: none"> • 0 (valeur par défaut) = Pas de contrôle de flux • 1 = Contrôle de flux matériel avec RTS toujours activé (ne s'applique pas aux ports RS485) • 2 = Contrôle de flux matériel avec RTS commuté
RTS_ON_DLY ¹	IN	UInt	Sélection du retard RTS activé : <ul style="list-style-type: none"> • 0 = (valeur par défaut) Pas de délai à partir de RTS actif jusqu'à ce que le premier caractère du message soit émis • 1 à 65535 = Délai en millisecondes à partir de RTS actif jusqu'à ce que le premier caractère du message soit émis (ne s'applique pas aux ports RS485). Les retards RTS sont appliqués indépendamment de la sélection FLOW_CTRL.
RTS_OFF_DLY ¹	IN	UInt	Sélection du retard RTS désactivé : <ul style="list-style-type: none"> • 0 = (valeur par défaut) Pas de délai à partir du dernier caractère émis jusqu'à ce que RTS soit désactivé • 1 à 65535 = Délai en millisecondes à partir du dernier caractère émis jusqu'à ce que RTS soit désactivé (ne s'applique pas aux ports RS485). Les retards RTS sont appliqués indépendamment de la sélection FLOW_CTRL.
RESP_TO ¹	IN	UInt	Dépassement du délai d'attente de la réponse Temps en millisecondes accordé par MB_MASTER pour que l'esclave réponde. Si l'esclave ne répond pas pendant cette durée, MB_MASTER renouvellera la demande ou mettra fin à la demande avec une erreur lorsque le nombre de nouvelles tentatives indiqué aura été envoyé. 5 ms à 65535 ms (valeur par défaut = 1000 ms).

Paramètre et type		Type de données	Description
MB_DB	IN	Variant	Référence au bloc de données d'instance utilisé par les instructions MB_MASTER ou MB_SLAVE. Une fois que vous avez inséré MB_MASTER ou MB_SLAVE dans votre programme, l'identificateur de DB apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur MB_DB de la boîte.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur. (version 2.0 uniquement)
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

¹ Paramètres facultatifs pour MB_COMM_LOAD (V 2.x ou ultérieur). Cliquez sur la flèche en bas de la boîte CONT/LOG pour agrandir la boîte et inclure ces paramètres.

MB_COMM_LOAD est exécuté pour configurer un port pour le protocole Modbus RTU. Une fois qu'un port est configuré pour le protocole Modbus RTU, il ne peut être utilisé que par les instructions MB_MASTER ou MB_SLAVE.

Il faut exécuter MB_COMM_LOAD une fois pour chaque port de communication utilisé pour la communication Modbus à configurer. Affectez un DB d'instance MB_COMM_LOAD unique pour chaque port que vous utilisez. Vous pouvez installer trois modules de communication (RS232 ou RS485) au maximum et un Communication Board (RS485) dans la CPU. Appelez MB_COMM_LOAD dans un OB de démarrage et exécutez-le une fois ou déclenchez l'appel via le memento système Premier cycle (Page 113) pour l'exécuter une fois. Ne réexécutez MB_COMM_LOAD que si des paramètres de communication tels que la vitesse de transmission ou la parité doivent changer.

Un bloc de données d'instance est affecté à MB_MASTER ou MB_SLAVE lorsque vous insérez ces instructions dans votre programme. Ce bloc de données d'instance est référencé lorsque vous définissez le paramètre MB_DB pour l'instruction MB_COMM_LOAD.

Variables du bloc de données MB_COMM_LOAD

Le tableau suivant montre les variables statiques publiques sauvegardées dans le DB d'instance pour MB_COMM_LOAD pouvant être utilisées dans votre programme.

Tableau 13- 147 Variables statiques dans le DB d'instance

Variable	Type de données	Description
ICHAR_GAP	UInt	Retard pour le délai inter-caractères. Ce paramètre indiqué en millisecondes sert à augmenter la durée escomptée entre caractères reçus. Le nombre correspondant de temps bit pour ce paramètre est ajouté à la valeur Modbus par défaut de 35 temps bit (3,5 temps caractère).
RETRIES	UInt	Nombre de nouvelles tentatives que tentera le maître avant de renvoyer le code d'erreur Pas de réponse 0x80C8.
STOP_BITS	USInt	Nombre de bits d'arrêt utilisés pour la trame de chaque caractère. Les valeurs correctes sont 1 et 2.

Tableau 13- 148 Codes d'erreur d'exécution de MB_COMM_LOAD ¹

STATUS (W#16#)	Description
0000	Pas d'erreur
8180	Valeur d'ID de port invalide (port/identificateur matériel erroné pour le module de communication)
8181	Valeur de vitesse de transmission invalide
8182	Valeur de parité invalide
8183	Valeur de contrôle de flux invalide
8184	Valeur invalide du délai de réponse (délai de réponse inférieur au minimum qui est de 5 ms)
8185	Le paramètre MB_DB n'est pas un bloc de données d'instance d'une instruction MB_MASTER ou MB_SLAVE.

¹ En plus des erreurs MB_COMM_LOAD présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

13.9.3.2 MB_MASTER (Communiquer via le port PtP en tant que maître Modbus RTU)

Tableau 13- 149 Instruction MB_MASTER

CONT/LOG	SCL	Description
	<pre>"MB_MASTER_DB" (REQ:=_bool_in_, MB_ADDR:=_uint_in_, MODE:=_usint_in_, DATA_ADDR:=_udint_in_, DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA_PTR:=_variant_inout_);</pre>	<p>L'instruction MB_MASTER permet la communication en tant que maître DB à l'aide d'un port configuré lors d'une exécution précédente de l'instruction MB_COMM_LOAD. Un bloc de données d'instance est automatiquement affecté lorsque vous insérez l'instruction MB_MASTER dans votre programme. Ce bloc de données d'instance MB_MASTER est utilisé lorsque vous définissez le paramètre MB_DB pour l'instruction MB_COMM_LOAD.</p>

Tableau 13- 150 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	0 = Pas de demande 1 = Demande d'envoi de données à un esclave Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Adresse de station Modbus RTU. Plage d'adressage standard (1 à 247) Plage d'adressage étendue (1 à 65535) La valeur 0 est réservée à la diffusion générale d'un message à tous les esclaves Modbus. Les codes de fonction Modbus 05, 06, 15 et 16 sont les seuls codes de fonction acceptés pour la diffusion générale.
MODE	IN	USInt	Sélection du mode : Indique le type de demande (lecture, écriture ou diagnostic). Voir le tableau des fonctions Modbus ci-après pour plus de détails.
DATA_ADDR	IN	UDInt	Adresse de début dans l'esclave. Indique l'adresse de début des données auxquelles accéder dans l'esclave Modbus. Voir les adresses valides dans le tableau des fonctions Modbus ci-après.
DATA_LEN	IN	UInt	Longueur des données. Indique le nombre de bits ou mots auxquels accéder dans cette demande. Voir les longueurs valides dans le tableau des fonctions Modbus ci-après.
DATA_PTR	IN	Variant	Pointeur de données. Désigne l'adresse M ou DB (type de DB non optimisé) pour les données en cours d'écriture ou de lecture.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 = Pas d'opération MB_MASTER en cours 1 = Opération MB_MASTER en cours
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

Règles de communication pour le maître Modbus

- MB_COMM_LOAD doit être exécuté pour configurer un port avant qu'une instruction MB_MASTER ne puisse communiquer avec ce port.
- Si un port doit être utilisé pour déclencher des demandes du maître Modbus, il ne doit pas être utilisé par MB_SLAVE. Une ou plusieurs instances d'exécution de MB_MASTER peuvent être utilisées avec ce port, mais toutes les exécutions de MB_MASTER doivent utiliser le même DB d'instance MB_MASTER pour ce port.
- Les instructions Modbus n'utilisent pas d'événements d'alarme de communication pour piloter le processus de communication. Votre programme doit interroger les conditions émission et réception achevées de l'instruction MB_MASTER.
- Nous vous conseillons d'appeler toutes les exécutions de MB_MASTER pour un port donné dans un OB de cycle de programme. En effet, les instructions maître Modbus ne peuvent s'exécuter que dans un niveau d'exécution, cycle de programme ou alarme cyclique/temporisée. Elles ne peuvent pas s'exécuter dans les deux niveaux de priorité d'exécution. L'interruption d'une instruction maître Modbus par une autre instruction maître Modbus dans un niveau de priorité d'exécution supérieur entraîne un fonctionnement incorrect. Les instructions maître Modbus ne doivent pas s'exécuter dans les niveaux de priorité mise en route, diagnostic ou erreur de temps.
- Une fois qu'une instruction maître déclenche une transmission, cette instance doit s'exécuter en continu avec l'entrée EN activée jusqu'à ce qu'un état DONE égal à 1 ou qu'un état ERROR égal à 1 soit renvoyé. Une instance particulière de MB_MASTER est considérée comme active jusqu'à ce que l'un de ces deux événements se produise. Tant que l'instance initiale est active, tout appel à une autre instance avec l'entrée REQ activée entraîne une erreur. Si l'exécution continue de l'instance initiale est interrompue, l'état de la demande reste actif pendant une durée indiquée par la variable statique "Blocked_Proc_Timeout". A l'expiration de cette durée, l'instruction maître suivante appelée avec l'entrée REQ activée devient l'instance active. Cela empêche une instance maître Modbus unique de monopoliser un port ou d'en verrouiller l'accès. Si l'instance active initiale n'est pas activée pendant la durée indiquée par la variable statique "Blocked_Proc_Timeout", l'exécution suivante par cette instance (avec REQ non activé) annulera l'état actif. Si REQ est activé, cette exécution déclenche une nouvelle demande maître comme si aucune autre instance n'était active.

Paramètre REQ

0 = Pas de demande ; 1 = Demande d'envoi de données à un esclave Modbus

Vous pouvez commander cette entrée à l'aide d'un contact déclenché sur niveau ou sur front. Dès que cette entrée est activée, un automate fini est démarré pour garantir qu'aucune autre instruction MB_MASTER utilisant le même DB d'instance n'a le droit d'émettre de demande, et ce jusqu'à ce que la demande en cours soit achevée. Tous les autres états d'entrée sont capturés et conservés en interne pendant la demande en cours jusqu'à ce que la réponse soit reçue ou qu'une erreur soit détectée.

Si la même instance de MB_MASTER est à nouveau exécutée avec l'entrée REQ à 1 avant l'achèvement de la demande en cours, il n'y aura pas d'émissions suivantes. Toutefois, dès que la demande aura été achevée, une nouvelle demande sera émise si MB_MASTER est à nouveau exécuté avec l'entrée REQ à 1.

Sélection du type de fonction Modbus via les paramètres DATA_ADDR et MODE

DATA_ADDR (adresse Modbus de début dans l'esclave) : Indique l'adresse de début des données auxquelles accéder dans l'esclave Modbus.

L'instruction MB_MASTER utilise une entrée MODE et non une entrée Code de fonction. La combinaison de MODE et de l'adresse Modbus déterminent le code de fonction qui est utilisé dans le message Modbus réel. Le tableau suivant montre la correspondance entre le paramètre MODE, le code de fonction Modbus et la plage d'adresses Modbus.

Tableau 13- 151 Fonctions Modbus

MODE	Fonction Modbus	Longueur des données	Opération et données	Adresse Modbus
0	01	1 à 2000 1 à 1992 ¹	Lecture de bits de sortie : 1 à 1992 (ou 2000) bits par demande	1 à 9999
0	02	1 à 2000 1 à 1992 ¹	Lecture de bits d'entrée : 1 à 1992 (ou 2000) bits par demande	10001 à 19999
0	03	1 à 125 1 à 124 ¹	Lecture de registres de maintien : 1 à 124 (ou 125) mots par demande	40001 à 49999 ou 400001 à 465535
0	04	1 à 125 1 à 124 ¹	Lecture de mots d'entrée : 1 à 124 (ou 125) mots par demande	30001 à 39999
1	05	1	Ecriture d'un bit de sortie : Un bit par demande	1 à 9999
1	06	1	Ecriture d'un registre de maintien : 1 mot par demande	40001 à 49999 ou 400001 à 465535
1	15	2 à 1968 2 à 1960 ¹	Ecriture de plusieurs bits de sortie : 2 à 1960 (ou 1968) bits par demande	1 à 9999
1	16	2 à 123 2 à 122 ¹	Ecriture de plusieurs registres de maintien : 2 à 122 (ou 123) mots par demande	40001 à 49999 ou 400001 à 465535
2	15	1 à 1968 2 à 1960 ¹	Ecriture d'un ou de plusieurs bits de sortie : 1 à 1960 (ou 1968) bits par demande	1 à 9999
2	16	1 à 123 1 à 122 ¹	Ecriture d'un ou de plusieurs registres de maintien : 1 à 122 (ou 123) mots par demande	40001 à 49999 ou 400001 à 465535
11	11	0	Lecture du mot d'état de communication esclave et du compteur d'événements. Le mot d'état indique si l'esclave est occupé (0 : non occupé, 0xFFFF : occupé). Le compteur d'événements est incrémenté à chaque fois qu'un message s'achève avec succès. Les deux opérandes DATA_ADDR et DATA_LEN de MB_MASTER ne sont pas pris en compte pour cette fonction.	
80	08	1	Contrôle de l'état de l'esclave à l'aide du code de diagnostic de données 0x0000 (test de retour : l'esclave renvoie la demande en écho) 1 mot par demande	

MODE	Fonction Modbus	Longueur des données	Opération et données	Adresse Modbus
81	08	1	Réinitialisation du compteur d'événements de l'esclave à l'aide du code de diagnostic de données 0x000A 1 mot par demande	
3 à 10, 12 à 79, 82 à 255			Réservé	

¹ En mode "Adressage étendu", les longueurs maximales de données sont réduites d'un octet ou d'un mot selon le type de données utilisé par la fonction.

Paramètre DATA_PTR

Le paramètre DATA_PTR pointe sur l'adresse M ou DB à lire ou dans laquelle écrire. Si vous utilisez un bloc de données, vous devez créer un bloc de données global qui fournit de la mémoire pour les lectures et écritures dans les esclaves Modbus.

Remarque

Le bloc de données DATA_PTR doit autoriser l'adressage direct

Le bloc de données doit autoriser à la fois l'adressage direct (absolu) et l'adressage symbolique. Vous devez donc sélectionner l'attribut d'accès "standard" lorsque vous créez le bloc de données.

Structures de bloc de données pour le paramètre DATA_PTR

- Ces types de données sont valides pour les **lectures de mot** des adresses Modbus 30001 à 39999, 40001 à 49999 et 400001 à 465536 et également pour les **écritures de mot** dans les adresses Modbus 40001 à 49999 et 400001 à 465536.
 - Tableau standard de types de données WORD, UINT ou INT
 - Structure WORD, UINT ou INT nommée dans laquelle chaque élément a un nom unique et un type de données de 16 bits
 - Structure complexe nommée dans laquelle chaque élément a un nom unique et un type de données de 16 ou 32 bits
- Pour les **lectures et écritures de bit** des adresses Modbus 00001 à 09999 et les lectures de bits des adresses 10001 à 19999
 - Tableau standard de types de données BOOL
 - Structure booléenne nommée de variables booléennes à nom unique.

- Bien que cela ne soit pas obligatoire, il est recommandé que chaque instruction MB_MASTER ait sa propre zone de mémoire distincte. La raison en est que le risque d'altération des données est plus important si plusieurs instructions MB_MASTER lisent et écrivent dans la même zone de mémoire.
- Il n'est pas obligatoire que les zones de données DATA_PTR soient dans le même bloc de données global. Vous pouvez créer un bloc de données à zones multiples pour les lectures Modbus, un bloc de données pour les écritures Modbus ou un bloc de données pour chaque station esclave.

Variables du bloc de données maître Modbus

Le tableau suivant montre les variables statiques publiques sauvegardées dans le DB d'instance pour MB_MASTER pouvant être utilisées dans votre programme.

Tableau 13- 152 Variables statiques dans le DB d'instance

Variable	Type de données	Valeur initiale	Description
Blocked_Proc_Timeout	Real	3,0	Durée (en secondes) pendant laquelle attendre une instance de maître Modbus bloquée avant d'annuler l'état ACTIF de cette instance. Cela peut, par exemple, se produire lorsqu'une demande maître a été émise et que le programme arrête d'appeler la fonction maître avant l'achèvement complet de la demande. La valeur de temps doit être supérieure à 0 et inférieure à 55 secondes ou une erreur se produit. La valeur par défaut est 0,5 seconde.
Extended Addressing	Bool	False	Configure l'adressage d'esclave à un ou deux octets. La valeur par défaut est 0. (0=adresse à un octet, 1=adresse à deux octets)

Votre programme peut écrire des valeurs dans les variables Blocked_Proc_Timeout et Extended_Addresssing pour commander le fonctionnement du maître Modbus. Vous trouverez dans la description de HR_Start_Offset et Extended_Addresssing au paragraphe MB_SLAVE un exemple d'utilisation de ces variables dans l'éditeur de programmes et des détails sur l'adressage étendu Modbus (Page 1347).

Codes d'erreur

Tableau 13- 153 Codes d'erreur d'exécution de MB_MASTER (erreurs de communication et de configuration) ¹

STATUS (W#16#)	Description
0000	Pas d'erreur
80C8	Dépassement de délai chez l'esclave. Vérifiez la vitesse de transmission, la parité et le câblage de l'esclave.
80D1	Le récepteur a émis une demande de contrôle de flux pour suspendre l'émission active et n'a jamais réactivé la transmission durant le temps d'attente indiqué. Cette erreur est également générée pendant le contrôle de flux matériel lorsque le récepteur n'affirme pas Prêt à émettre (CTS) pendant le temps d'attente indiqué.
80D2	La demande d'émission a été annulée, car aucun signal DSR (modem prêt) n'a été reçu de l'équipement de transmission de données (DCE).
80E0	Le message a été interrompu car la mémoire tampon de réception est pleine.
80E1	Le message a été interrompu en raison d'une erreur de parité.
80E2	Le message a été interrompu en raison d'une erreur de trame.
80E3	Le message a été interrompu en raison d'une erreur de débordement.
80E4	Le message a été interrompu, car la longueur indiquée dépasse la taille de mémoire tampon totale.
8180	Valeur d'ID de port invalide ou erreur avec l'instruction MB_COMM_LOAD
8186	Adresse de station Modbus invalide
8188	Mode invalide indiqué pour une demande de diffusion générale
8189	Valeur d'adresse de données invalide
818A	Valeur de longueur de données invalide
818B	Pointeur invalide vers la source/destination de données locale : taille incorrecte
818C	Pointeur invalide pour DATA_PTR ou Blocked_Proc_Timeout invalide : La zone de données doit être un DB (permettant l'adressage symbolique et absolu) ou se situer en mémoire M.
8200	Le port est occupé par le traitement d'une demande d'émission.

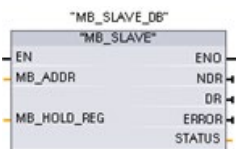
Tableau 13- 154 Codes d'erreur d'exécution de MB_MASTER (erreurs de protocole Modbus) ¹

STATUS (W#16#)	Code de réponse de l'esclave	Erreurs de protocole Modbus
8380	-	Erreur CRC
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou adresse en dehors de la plage valide de la zone DATA_PTR
8384	Supérieur à 03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non pris en charge (code de fonction 08)
8386	-	Le code de fonction dans la réponse ne correspond pas au code dans la demande.
8387	-	Ce n'est pas le bon esclave qui a répondu.
8388	-	La réponse de l'esclave à une demande d'écriture est incorrecte. La demande d'écriture renvoyée par l'esclave ne correspond pas à ce que le maître a effectivement envoyé.

¹ En plus des erreurs MB_MASTER présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

13.9.3.3 MB_SLAVE (Communiquer via le port PtP en tant qu'esclave Modbus RTU)

Tableau 13- 155 Instruction MB_SLAVE

CONT/LOG	SCL	Description
	<pre>"MB_SLAVE_DB" (MB_ADDR:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	<p>L'instruction MB_SLAVE permet à votre programme de communiquer en tant qu'esclave Modbus par le biais d'un port point à point sur le CM (RS485 ou RS232) et le CB (RS485). Lorsqu'un maître RTU Modbus distant émet une demande, votre programme utilisateur répond par l'exécution de MB_SLAVE. STEP 7 crée automatiquement un DB d'instance lorsque vous insérez l'instruction. Utilisez le nom MB_SLAVE_DB lorsque vous indiquez le paramètre MB_DB pour l'instruction MB_COMM_LOAD.</p>

13.9 Communication d'héritage Modbus RTU (CM/CB 1241 uniquement)

Tableau 13- 156 Types de données pour les paramètres

Paramètre et type		Type de données	Description
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Adresse de station de l'esclave Modbus : Plage d'adressage standard (1 à 247) Plage d'adressage étendue (0 à 65535)
MB_HOLD_REG	IN	Variant	Pointeur désignant le DB de registre de maintien Modbus : Le registre de maintien Modbus peut être un memento ou un bloc de données.
NDR	OUT	Bool	Nouvelles données prêtes (New Data Ready) : <ul style="list-style-type: none"> 0 = Pas de nouvelles données 1 = Indique que des nouvelles données ont été écrites par le maître Modbus.
DR	OUT	Bool	Données lues : <ul style="list-style-type: none"> 0 = Pas de données lues 1 = Indique que des données ont été lues par le maître Modbus.
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. Si l'exécution s'achève avec une erreur, la valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

Les fonctions de communication Modbus de codes 1, 2, 4, 5 et 15 peuvent lire et écrire des bits et des mots directement dans la mémoire image des entrées et la mémoire image des sorties de la CPU. Pour ces fonctions, le paramètre MB_HOLD_REG doit être défini avec un type de données supérieur à un octet. Le tableau suivant montre en exemple la correspondance entre les adresses Modbus et la mémoire image dans la CPU.

Tableau 13- 157 Correspondance entre adresses Modbus et mémoire image

Fonctions Modbus					S7-1200		
Codes	Fonction	Zone de données	Plage d'adresses			Zone de données	Adresse de la CPU
01	Lecture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
02	Lecture de bits	Entrée	10001	à	18192	Mémoire image des entrées	I0.0 à I1023.7
04	Lecture de mots	Entrée	30001	à	30512	Mémoire image des entrées	IW0 à IW1022
05	Ecriture d'un bit	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
15	Ecriture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7

Les fonctions de communication Modbus de codes 3, 6 et 16 utilisent un registre de maintien Modbus qui peut être une zone d'adresse en mémoire M ou un bloc de données. Le type de registre de maintien est indiqué par le paramètre MB_HOLD_REG dans l'instruction MB_SLAVE.

Remarque

Bloc de données MB_HOLD_REG

Un bloc de données faisant office de registre de maintien Modbus doit autoriser à la fois l'adressage direct (absolu) et l'adressage symbolique. Vous devez donc sélectionner l'attribut d'accès "standard" lorsque vous créez le bloc de données.

Le tableau suivant montre des exemples de correspondance entre adresses Modbus et registre de maintien pour les codes de fonction Modbus 03 (lecture de mots), 06 (écriture d'un mot) et 16 (écriture de mots). La limite supérieure effective des adresses de DB dépend des limites de mémoire de travail et de mémoire M maximum pour chaque modèle de CPU.

Tableau 13- 158 Correspondance entre adresses Modbus et mémoire CPU

Adresse du maître Modbus	Exemples pour le paramètre MB_HOLD_REG				
	MW100	DB10.DBw0	MW120	DB10.DBW50	"Recette".ingredient
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recette".ingredient[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Recette".ingredient[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Recette".ingredient[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Recette".ingredient[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Recette".ingredient[5]

Tableau 13- 159 Fonctions de diagnostic

Fonctions de diagnostic Modbus MB_SLAVE du S7-1200		
Codes	Sous-fonction	Description
08	0000H	Renvoi d'un test d'écho des données de requête : MB_SLAVE renvoie en écho à un maître Modbus un mot de données reçu.
08	000AH	Effacement du compteur d'événements de communication : MB_SLAVE effacera le compteur d'événements de communication qui est utilisé pour la fonction Modbus 11.
11		Lecture du compteur d'événements de communication : MB_SLAVE utilise un compteur d'événements de communication interne pour enregistrer le nombre de demandes de lecture et d'écriture Modbus envoyées à l'esclave Modbus qui ont abouti. Le compteur ne s'incrémente pas pour les fonctions 8 et 11 ni pour les demandes à diffusion générale. Il ne s'incrémente pas non plus pour toute demande entraînant une erreur de communication (erreurs de parité ou de CRC, par exemple).

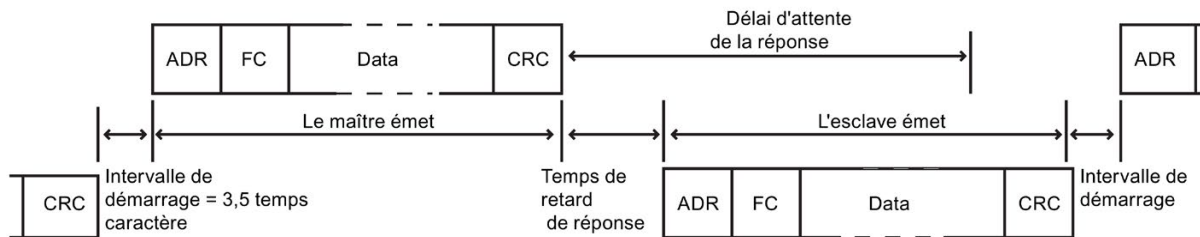
L'instruction MB_SLAVE accepte les demandes d'écriture à diffusion générale provenant de n'importe quel maître Modbus à condition que la requête demande l'accès à des adresses valides. MB_SLAVE génère un code d'erreur 0x8188 pour les codes de fonction non pris en charge en diffusion générale.

Règles de communication pour l'esclave Modbus

- MB_COMM_LOAD doit être exécuté pour configurer un port avant qu'une instruction MB_SLAVE ne puisse communiquer par le biais de ce port.
- Si un port doit répondre en tant qu'esclave à un maître Modbus, ne programmez pas ce port avec l'instruction MB_MASTER.
- Une seule instance de MB_SLAVE peut être utilisée avec un port donné, faute de quoi le comportement pourrait devenir imprévisible.
- Les instructions Modbus n'utilisent pas d'événements d'alarme de communication pour piloter le processus de communication. Votre programme doit piloter le processus de communication en interrogeant les conditions émission et réception achevées de l'instruction MB_SLAVE.
- L'instruction MB_SLAVE doit s'exécuter périodiquement à une fréquence permettant de répondre opportunément à des demandes entrantes provenant d'un maître Modbus. Il est recommandé d'exécuter MB_SLAVE à chaque cycle dans un OB de cycle de programme. Il est possible d'exécuter MB_SLAVE dans un OB d'alarme cyclique, mais ce n'est pas conseillé en raison du risque que des retards excessifs dans le programme d'alarme bloquent temporairement l'exécution d'autres programmes d'alarme.

Synchronisation des signaux Modbus

MB_SLAVE doit être exécuté périodiquement pour recevoir chaque demande provenant du maître Modbus et pour y répondre de manière appropriée. La fréquence d'exécution de MB_SLAVE dépend du délai d'attente de la réponse dans le maître Modbus. Ceci est illustré dans le schéma suivant.



Le délai d'attente de la réponse RESP_TO est la durée pendant laquelle un maître Modbus attend le début d'une réponse d'un esclave Modbus. Ce temps n'est pas défini par le protocole Modbus mais constitue un paramètre de chaque maître Modbus. La fréquence d'exécution (intervalle de temps entre une exécution et la suivante) de MB_SLAVE doit se baser sur les paramètres spécifiques de votre maître Modbus. Vous devriez exécuter MB_SLAVE au minimum deux fois pendant le délai d'attente de la réponse du maître Modbus.

Variables pour l'esclave Modbus

Ce tableau montre les variables statiques publiques sauvegardées dans le bloc de données d'instance pour MB_SLAVE pouvant être utilisées dans votre programme.

Tableau 13- 160 Variables pour l'esclave Modbus

Variable	Type de données	Description
Request_Count	Word	Nombre de toutes les demandes reçues par cet esclave
Slave_Message_Count	Word	Nombre de demandes reçues pour cet esclave spécifique
Bad_CRC_Count	Word	Nombre de demandes reçues comportant une erreur CRC
Broadcast_Count	Word	Nombre de demandes à diffusion générale reçues
Exception_Count	Word	Erreurs Modbus spécifiques nécessitant le renvoi d'une exception
Success_Count	Word	Nombre de demandes reçues pour cet esclave spécifique ne contenant pas d'erreurs de protocole
HR_Start_Offset	Word	Indique l'adresse de début du registre de maintien Modbus (valeur par défaut = 0).
Extended_Addresssing	Bool	Configure l'adressage d'esclave à un ou deux octets. (0=adresse à un octet, 1=adresse à deux octets ; valeur par défaut = 0)

Votre programme peut écrire des valeurs dans les variables HR_Start_Offset et Extended_Addresssing pour commander le fonctionnement de l'esclave Modbus. Les autres variables peuvent être lues pour visualiser l'état de Modbus.

HR_Start_Offset

Les adresses de registre de maintien Modbus commencent à 40001 ou 400001, ce qui correspond à l'adresse de début du registre de maintien dans la mémoire API. Vous pouvez toutefois configurer la variable "HR_Start_Offset" afin que le registre de maintien Modbus commence à une valeur autre que 40001 ou 400001.

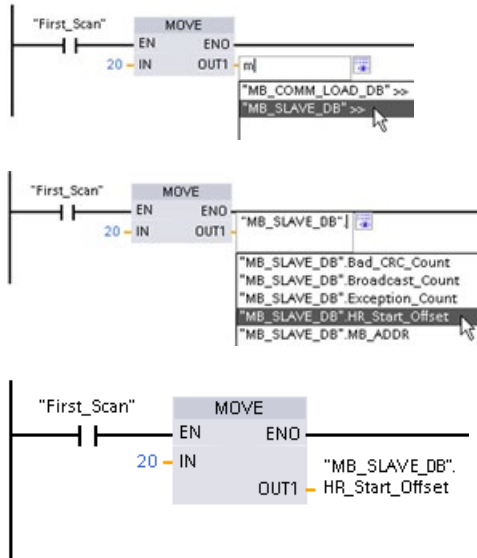
Par exemple, si le registre de maintien est configuré pour commencer à MW100 et s'il a une longueur de 100 mots. Un décalage de 20 indique une adresse de début de registre située à 40021 au lieu de 40001. Toute adresse inférieure à 40021 et supérieure à 400119 entraînera une erreur d'adressage.

Tableau 13- 161 Exemple d'adressage de registre de maintien Modbus

HR_Start_Offset	Adresse	Minimum	Maximum
0	Adresse Modbus (mot)	40001	40099
	Adresse S7-1200	MW100	MW298
20	Adresse Modbus (mot)	40021	40119
	Adresse S7-1200	MW100	MW298

HR_Start_Offset est une valeur de mot qui indique l'adresse de début du registre de maintien Modbus et est sauvegardée dans le bloc de données d'instance de MB_SLAVE. Vous pouvez définir la valeur de cette variable statique publique à l'aide de la liste déroulante d'aide pour les paramètres lorsque vous avez inséré MB_SLAVE dans votre programme.

Par exemple, une fois MB_SLAVE inséré dans un réseau CONT, vous pouvez aller à un réseau précédent et définir la valeur HR_Start_Offset. La valeur doit être affectée avant l'exécution de MB_SLAVE.



Entrée d'une variable pour l'esclave Modbus à l'aide du nom de DB par défaut :

1. Placez le curseur dans le champ de paramètre et tapez la lettre m.
2. Sélectionnez "MB_SLAVE_DB" dans la liste déroulante.
3. Placez le curseur à droite du nom de DB (après le guillemet) et entrez un point.
4. Sélectionnez "MB_SLAVE_DB.HR_Start_Offset" dans la liste déroulante.

Extended Addressing

Vous accédez à la variable Extended Addressing de la même manière qu'à la variable HR_Start_Offset décrite ci-avant, si ce n'est que Extended Addressing est une valeur booléenne. Une valeur booléenne doit être écrite par une bobine de sortie et non par une boîte MOVE.

L'adressage d'esclave Modbus peut être configuré à un octet (la norme pour Modbus) ou à deux octets. On utilise l'adressage étendu pour accéder à plus de 247 unités dans un réseau unique. La sélection de l'adressage étendu vous permet d'accéder à 64000 adresses au maximum. Une trame pour la fonction Modbus 1 est présentée ci-dessous comme exemple.

Tableau 13- 162 Adresse d'esclave à un octet (octet 0)

Fonction 1	Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	
Demande	Adr. es-clave	Code F	Adresse de début		Longueur des bobines		
Réponse valide	Adr. es-clave	Code F	Longueur	Données de bobine			
Réponse erronée	Adr. es-clave	0x81	Code E				

Tableau 13- 163 Adresse d'esclave à deux octets (octet 0 et octet 1)

	Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6
Demande	Adresse d'esclave		Code F	Adresse de début		Longueur des bobines	
Réponse valide	Adresse d'esclave		Code F	Longueur	Données de bobine		
Réponse erronée	Adresse d'esclave		0x81	Code E			

Codes d'erreur

Tableau 13- 164 Codes d'erreur d'exécution de MB_SLAVE (erreurs de communication et de configuration) ¹

STATUS (W#16#)	Description
80D1	Le récepteur a émis une demande de contrôle de flux pour suspendre l'émission active et n'a jamais réactivé la transmission durant le temps d'attente indiqué. Cette erreur est également générée pendant le contrôle de flux matériel lorsque le récepteur n'affirme pas Prêt à émettre (CTS) pendant le temps d'attente indiqué.
80D2	La demande d'émission a été annulée, car aucun signal DSR (modem prêt) n'a été reçu de l'équipement de transmission de données (DCE).
80E0	Le message a été interrompu car la mémoire tampon de réception est pleine.
80E1	Le message a été interrompu en raison d'une erreur de parité.
80E2	Le message a été interrompu en raison d'une erreur de trame.
80E3	Le message a été interrompu en raison d'une erreur de débordement.
80E4	Le message a été interrompu, car la longueur indiquée dépasse la taille de mémoire tampon totale.
8180	Valeur d'ID de port invalide ou erreur avec l'instruction MB_COMM_LOAD
8186	Adresse de station Modbus invalide
8187	Pointeur invalide vers le DB MB_HOLD_REG : La zone est trop petite.
818C	Pointeur MB_HOLD_REG invalide sur mémoire M ou DB (le DB doit autoriser à la fois l'adressage symbolique et l'adressage absolu)

Tableau 13- 165 Codes d'erreur d'exécution de MB_SLAVE (erreurs de protocole Modbus) ¹

STATUS (W#16#)	Code de réponse de l'esclave	Erreurs de protocole Modbus
8380	Pas de réponse	Erreur CRC
8381	01	Code de fonction non pris en charge ou non pris en charge dans des messages à diffusion générale
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou adresse en dehors de la plage valide de la zone DATA_PTR
8384	03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non pris en charge (code de fonction 08)

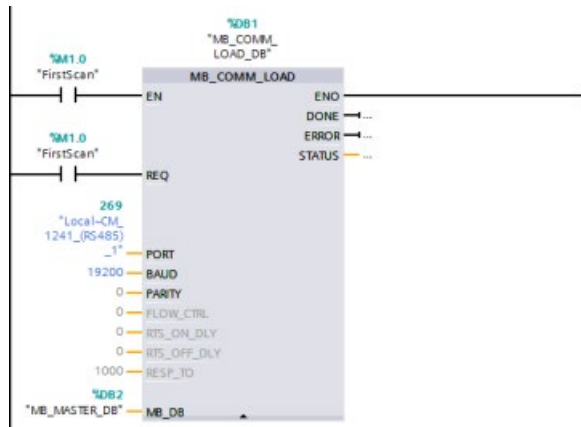
¹ En plus des erreurs MB_SLAVE présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

13.9.4 Exemples d'instructions Modbus RTU héritées

13.9.4.1 Exemple : Programme maître Modbus RTU d'héritage

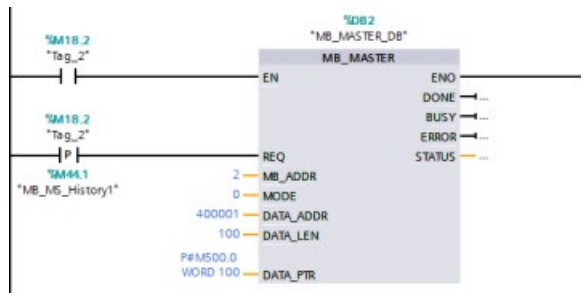
MB_COMM_LOAD est initialisé au démarrage à l'aide du memento Premier cycle. Cette manière d'exécuter MB_COMM_LOAD ne doit être utilisée que lorsque la configuration du port série n'est pas modifiée pendant l'exécution.

Réseau 1 : Configurer/initialiser le port de communication du module RS485 une seule fois pendant le premier cycle.



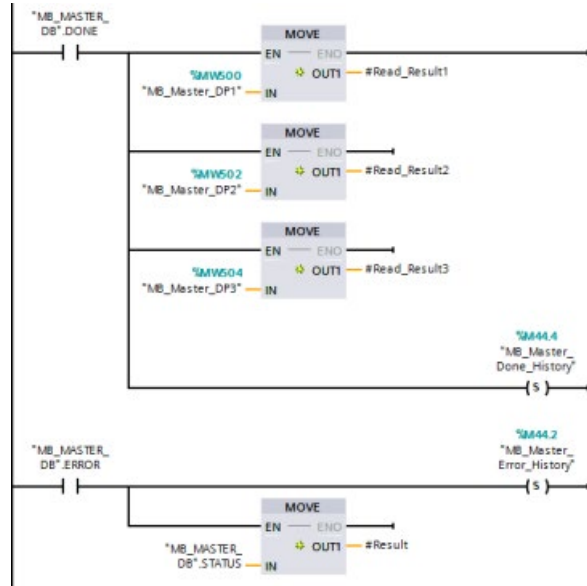
Une instruction MB_MASTER est utilisée dans l'OB de cycle de programme pour communiquer avec un esclave unique. Il est possible d'utiliser d'autres instructions MB_MASTER dans l'OB de cycle de programme pour communiquer avec d'autres esclaves ou de réutiliser un FB MB_MASTER pour communiquer avec des esclaves supplémentaires.

Réseau 2 : Lire 100 mots de données de registre de maintien de l'adresse 400001 sur l'esclave 2 vers l'adresse de mémoire MW500-MW698.

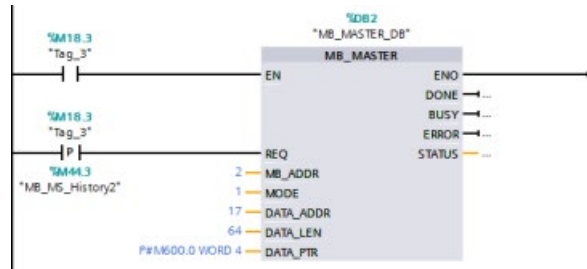


13.9 Communication d'héritage Modbus RTU (CM/CB 1241 uniquement)

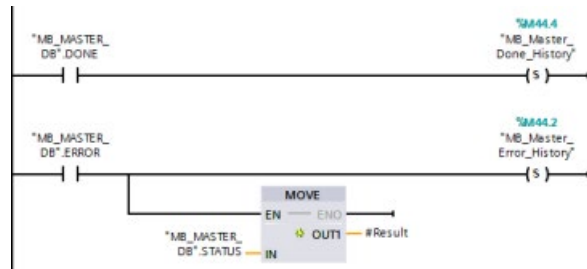
Réseau 3 : Déplacer les 3 premiers mots des données de registre de maintien lues à un autre emplacement et mettre un bit d'historique DONE à 1. Ce réseau met également à 1 un bit d'historique ERROR et sauvegarde le mot STATUS à un autre emplacement en cas d'erreur.



Réseau 4 : Écrire 64 bits de données de MW600-MW607 dans les adresses de bits de sortie 00017 à 00081 sur l'esclave 2.



Réseau 5 : Mettre un bit d'historique DONE à 1 lorsque l'écriture est achevée. En cas d'erreur, le programme met à 1 un bit d'historique ERROR et sauvegarde le code STATUS.

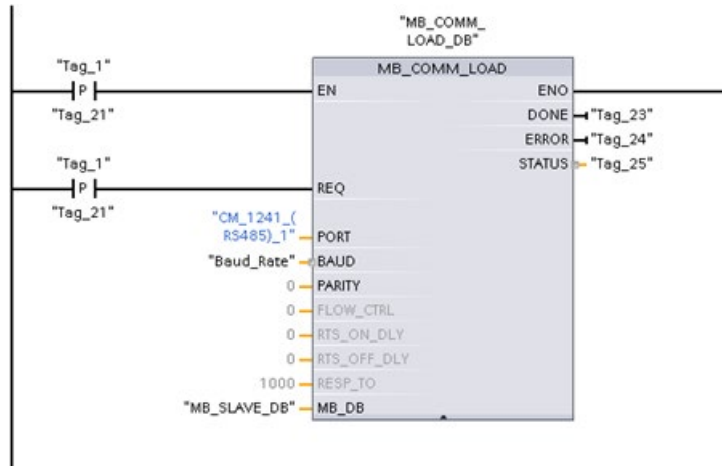


13.9.4.2 Exemple : Programme esclave Modbus RTU d'héritage

MB_COMM_LOAD présenté ci-dessous est initialisé à chaque fois que "Tag_1" est activé.

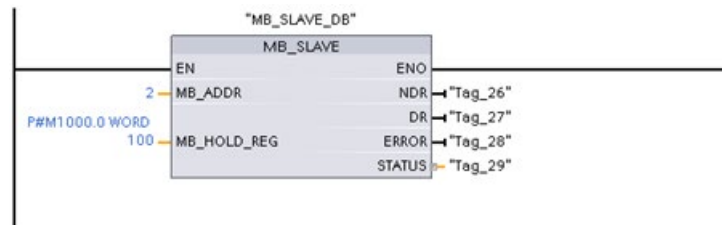
Cette manière d'exécuter MB_COMM_LOAD ne doit être utilisée que lorsque la configuration du port série est modifiée à l'exécution, en raison d'une configuration IHM.

Réseau 1 : Initialiser les paramètres de module RS485 à chaque fois qu'ils sont modifiés par un appareil IHM.



L'instruction MB_SLAVE montrée ci-dessous est insérée dans un OB cyclique qui est exécuté toutes les 10 ms. Cela n'assure certes pas la réponse la plus rapide possible de l'esclave, mais cela garantit de bonnes performances à 9600 bauds pour des messages courts (20 octets au plus dans la demande).

Réseau 2 : Vérifier la présence de demandes maître Modbus à chaque cycle. Le registre de maintien Modbus est configuré pour 100 mots en commençant à MW1000.



13.10 Industrial Remote Communication (IRC)

13.10.1 Présentation de CP de commande à distance

La communication industrielle à distance Industrial Remote Communication offre un accès sécurisé et économique à des installations, des machines et des applications fortement disséminées de toutes tailles. Industrial Remote Communications comprend les moyens de communication suivants par le biais de modules CP :

- **TeleControl** : La téléconduite consiste à relier des stations de processus (RTU : remote terminal units) géographiquement dispersées à un ou plusieurs systèmes de conduite de processus centraux pour la conduite et la supervision. Divers composants de transmission dans la gamme des réseaux distants (Remote Networks) prennent en charge la communication à distance par différents réseaux de communication publics ou privés. Des protocoles de téléconduite spéciaux exécutent l'échange de données du processus, cycliquement ou par déclenchement événementiel, ce qui permet un pilotage efficace de l'ensemble du processus.
- **TeleService** : la télémaintenance consiste en l'échange de données avec des installations techniques distantes (machines, installations, ordinateurs, etc.) à des fins de détection de défauts, de diagnostic, de maintenance, de réparation et d'optimisation.
- Applications supplémentaires pour la communication à distance, par exemple la surveillance, les applications pour réseau électrique intelligent (smart grid) et la surveillance d'état

CP de commande à distance pour le S7-1200

Les processeurs de communication suivants, dont beaucoup fournissent également accès au serveur Web (Page 1047) S7-1200, sont disponibles pour les applications TeleControl :

- **CP 1243-1 :**
 - Numéro d'article 6GK7 243-1BX30-0XE0
 - Processeur de communication pour connecter le SIMATIC S7-1200 via l'infrastructure publique (par exemple DSL) à un centre de commande avec TeleControl Server Basic (TCSB version V3).
 - Avec l'aide de la technologie VPN et le pare-feu, le CP permet un accès protégé au S7-1200.
 - Vous pouvez utiliser le CP comme interface Ethernet supplémentaire de la CPU pour la communication S7.
 - Vous communiquez entre le CP et la CPU en utilisant des données configurables qui accèdent aux variables API.
- **CP 1243-1 DNP3 :**
 - Numéro d'article 6GK7 243-1JX30-0XE0
 - Processeur de communication pour connecter le SIMATIC S7-1200 aux centres de commande en utilisant le protocole DNP3.
 - Vous communiquez entre le CP et la CPU en utilisant des données configurables qui accèdent aux variables API.
- **CP 1243-1 IEC :**
 - Numéro d'article 6GK7 243-1PX30-0XE0
 - Processeur de communication pour connecter le SIMATIC S7-1200 aux centres de commande en utilisant le protocole IEC 60870-5.
 - Vous communiquez entre le CP et la CPU en utilisant des données configurables qui accèdent aux variables API.
- **CP 1243-1 PCC :**
 - Numéro d'article : 6GK7 243-1HX30-0XE0
 - Processeur de communication pour connecter le SIMATIC S7-1200 aux centres de commande par la communication PCC (Plant Cloud Communication).
 - Vous communiquez entre le CP et la CPU à l'aide de données configurables qui accèdent aux variables API.
- **CP 1242-7 :**
 - Numéro d'article : 6GK7 242-7KX31-0XE0
 - Processeur de communication pour connecter le SIMATIC S7-1200 à un centre de commande avec TeleControl Server Basic en utilisant le réseau mobile sans fil (GPRS) et l'infrastructure publique (DSL)

- **CP 1242-7 GPRS V2 :**
 - Numéro d'article 6GK7 242-7KX31-0XE0
 - Processeur de communication pour connecter le SIMATIC S7-1200 à un centre de commande avec TeleControl Server Basic (TCSB version v3) en utilisant le réseau mobile sans fil (GPRS) et l'infrastructure publique (DSL)
 - Avec l'aide de la technologie VPN et le pare-feu, le CP permet un accès protégé au S7-1200.
 - Vous pouvez utiliser le CP comme interface Ethernet supplémentaire de la CPU pour la communication S7.
 - Vous communiquez entre le CP et la CPU en utilisant des données configurables qui accèdent aux variables API.
- **CP 1243-7 LTE-xx :**
 - Processeur de communication pour connecter le SIMATIC S7-1200 à un centre de commande avec TeleControl Server Basic (TCSB version v3) en utilisant le réseau mobile sans fil (GPRS) et l'infrastructure publique (DSL)
 - Assistance concernant les spécifications de réseau mobile sans fil suivantes : GSM/GPRS, UMTS (G3), LTE
 - Pour répondre aux demandes de pays avec des spécifications de réseau mobile sans fil différentes, le CP est disponible en deux variantes :
 - CP 1243-7 LTE-US :
 - Norme nord-américaine
 - Numéro d'article 6GK7 243-7SX30-0XE0
 - CP 1243-7 LTE-EU :
 - Norme en Europe occidentale
 - Numéro d'article 6GK7 243-7KX30-0XE0
 - Avec l'aide de la technologie VPN et le pare-feu, le CP permet un accès protégé au S7-1200.
 - Vous pouvez utiliser le CP comme interface Ethernet supplémentaire de la CPU pour la communication S7.
 - Vous communiquez entre le CP et la CPU en utilisant des données configurables qui accèdent aux variables API.
- **CP 1243-8 IRC :**
 - Numéro d'article : 6GK7 242-8RX30-0XE0
 - Processeur de communication pour connecter le SIMATIC S7-1200 à un réseau ST7, configuration de points de données et VPN

Remarque

Vous devez posséder le logiciel TeleControl Server Basic pour les applications TeleControl pour les CP autres que le CP 1243-1.

Communication sûre

Le protocole SINAUT ST7 qui a fait ses preuves ou le protocole DNP3 ou CEI 60870-5 standardisé ajoutent la sécurité à la communication Industrial Remote Communication (http://w3app.siemens.com/mcms/infocenter/dokumentcenter/sc/ic/InfocenterLanguagePacks/Netzwerksicherheit/6ZB5530-1AP02-0BA4_BR_Network_Security_en_112015.pdf). La solution TeleControl fournit des mesures complètes pour empêcher la falsification et la perte de données. Chaque module de transmission comporte une vaste mémoire pour plusieurs milliers de trames de données, ce qui permet de surmonter les temps d'arrêt dans la liaison de transmission. Des solutions VPN spécialisées protègent les réseaux spéciaux basés sur IP.

Le processeur de communication CP 1243-1 relie de façon sûre l'automate programmable S7-1200 SIMATIC aux réseaux Ethernet. Grâce à son pare-feu intégré (Stateful Inspection) et aux fonctions de sécurité (IPsec) du protocole VPN, ce processeur de communication vous aide à protéger les stations S7-1200 et les réseaux de niveau inférieur de tout accès non autorisé et à protéger les transmissions de données des manipulations et de l'espionnage grâce au cryptage. Ce CP peut en outre être utilisé pour intégrer la station S7-1200 dans le logiciel de poste de conduite TeleControl Server Basic à l'aide de réseaux distants basés sur IP.

13.10.2 Connexion à un réseau GSM

Communication WAN sur IP via GPRS

Le S7-1200 peut être connecté aux réseaux GSM grâce au processeur de communication CP 1242-7. Le CP 1242-7 permet une communication WAN depuis des stations distantes avec un centre de commande et une communication inter-stations.

Une communication inter-stations est possible uniquement via un réseau GSM. Pour une communication entre une station distante et une salle de commande, le centre de commande doit disposer d'un PC avec accès à Internet.

Le CP 1242-7 accepte les services de communication suivants via le réseau GSM :

- GPRS (General Packet Radio Service)

Le service orienté paquets de transmission de données "GPRS" est géré via le réseau GSM.

- (SMS) Service de messages courts

Le CP 1242-7 peut recevoir et envoyer des messages SMS. Le partenaire de communication peut être un téléphone portable ou un S7-1200.

Le CP 1242-7 est adapté pour une utilisation dans le secteur industriel partout dans le monde et accepte les bandes de fréquence suivantes :

- 850 MHz
- 900 MHz
- 1 800 MHz
- 1 900 MHz

Conditions requises

Le matériel utilisé dans les stations ou le centre de commande dépend de l'application spécifique.

- Pour une communication avec ou via une salle de commande centrale, le centre de commande doit disposer d'un PC avec un accès à Internet.
- Excepté le matériel de la station, une station S7-1200 distante avec un CP 1242-7 doit satisfaire aux exigences suivantes pour pouvoir communiquer via le réseau GSM :
 - Contrat avec un fournisseur de réseau GSM approprié
 - Si le système GPRS est utilisé, le contrat doit permettre l'utilisation du service GPRS.
 - Si une communication directe entre stations via le réseau GSM est nécessaire, le fournisseur de réseau GSM doit affecter une adresse IP fixe aux CP. Dans ce cas, la communication entre stations n'est pas réalisée via le centre de commande.
 - Carte SIM appartenant au contrat
 - La carte SIM est insérée dans le CP 1242-7.
 - Disponibilité locale d'un réseau GSM dans la plage de la station

13.10.3 Applications du CP 1242-7

Le CP 1242-7 peut être utilisé pour les applications suivantes :

Applications Telecontrol

- Envoi de messages par SMS

Via the CP 1242-7, la CPU d'un S7-1200 distant peut recevoir des messages SMS depuis le réseau GSM ou envoyer des messages par SMS à un téléphone portable configuré ou un S7-1200.

- Communication avec un centre de commande

Les stations S7-1200 distantes communiquent via le réseau GSM et l'Internet avec un serveur telecontrol dans la station maître. Pour le transfert de données à l'aide du système GPRS, l'application "TELECONTROL SERVER BASIC" est installée sur le serveur telecontrol dans la station maître. Le serveur telecontrol communique avec un système de commande central de niveau supérieur à l'aide de la fonction du serveur OPC intégrée.

- Communication entre stations S7-1200 via un réseau GSM

La communication entre stations distantes avec un CP 1242-7 peut être gérée de deux manières différentes :

- Communication inter-stations via une station maître

Dans cette configuration, une connexion sécurisée permanente entre stations S7-1200 qui communiquent les unes avec les autres et le serveur telecontrol est établie dans la station maître. La communication entre les stations est réalisée via le serveur telecontrol. Le CP 1242-7 fonctionne en mode "Telecontrol".

- Communication directe entre les stations

Pour une communication directe entre stations sans détournement via la station maître, des cartes SIM avec une adresse IP fixe qui permettent aux stations de s'adresser les unes aux autres directement sont utilisées. Les services de communication et fonctions de sécurité possibles (par exemple VPN) dépendent de ce qui est proposé par le fournisseur de réseau. Le CP 1242-7 fonctionne en mode "GPRS direct".

TeleService via GPRS

Une connexion TeleService peut être établie entre une station d'ingénierie avec STEP 7 et une station S7-1200 distante avec un CP 1242-7 via le réseau GSM et l'Internet. La connexion a lieu entre la station d'ingénierie via un serveur telecontrol ou une passerelle TeleService qui agit comme un intermédiaire redirigeant les trames et établissant l'autorisation. Ces PC utilisent les fonctions de l'application "TELECONTROL SERVER BASIC".

Vous pouvez utiliser la connexion TeleService pour les raisons suivantes :

- Téléchargement des données de configuration ou du programme depuis le projet STEP 7 vers la station
- Interrogation des données de diagnostic sur la station

13.10.4 Autres caractéristiques du CP 1242-7

Autres services et fonctions du CP 1242-7

- Synchronisation horaire du CP via l'Internet

Vous pouvez régler l'heure sur le CP de la manière suivante :

- En mode "Telecontrol", l'heure du jour est transférée par le serveur telecontrol. Le CP l'utilise pour régler sa propre heure.
- En mode "GPRS direct", le CP peut demander l'heure à l'aide du SNTP.

Pour synchroniser l'heure de la CPU, vous pouvez lire l'heure actuelle depuis la CPU à l'aide d'un bloc.

- Mise en mémoire tampon provisoire de messages à envoyer en cas de problèmes de connexion
- Disponibilité accrue grâce à la possibilité de se connecter à un serveur telecontrol de remplacement
- Journalisation du volume de données

Les volumes de données transférés sont journalisés et peuvent être évalués à des fins spécifiques.

13.10.5 Informations supplémentaires

Les manuels des CP, ainsi que la documentation associée et les informations produit donnent des informations détaillées :

- CP 1242-7 (<http://support.automation.siemens.com/WW/view/fr/45605894>)
- CP 1243-7 LTE (<https://support.industry.siemens.com/cs/ww/fr/ps/15924>)
- CP 1243-1 DNP3 (<https://support.industry.siemens.com/cs/ww/fr/ps/15938>)
- CP 1243-8 IRC (<https://support.industry.siemens.com/cs/ww/fr/ps/21162>)
- CP 1243-1 IEC (<https://support.industry.siemens.com/cs/ww/fr/ps/15942>)
- Mises à jour du firmware si disponibles (<https://support.industry.siemens.com/cs/ww/fr/view/109482530/en>)

13.10.6 Accessoires

Antenne GSM/GPRS ANT794-4MR

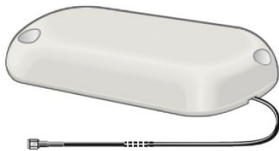
Les antennes suivantes sont disponibles pour une utilisation dans des réseaux GSM/GPRS et peuvent être installées à la fois en intérieur et en extérieur :

- Antenne quadribande ANT794-4MR
<http://support.automation.siemens.com/WW/view/fr/23119005>



Nom abrégé	N° de référence	Explication
ANT794-4MR	6NH9 860-1AA00	Antenne Quadband (900, 1800/1900 MHz, UMTS) ; étanche pour les utilisations en intérieur et en extérieur ; câble de raccordement de 5 m relié de façon permanente à l'antenne ; connecteur SMA, comprenant équerre de montage, vis et prises murales

- Antenne en nappe ANT794-3M



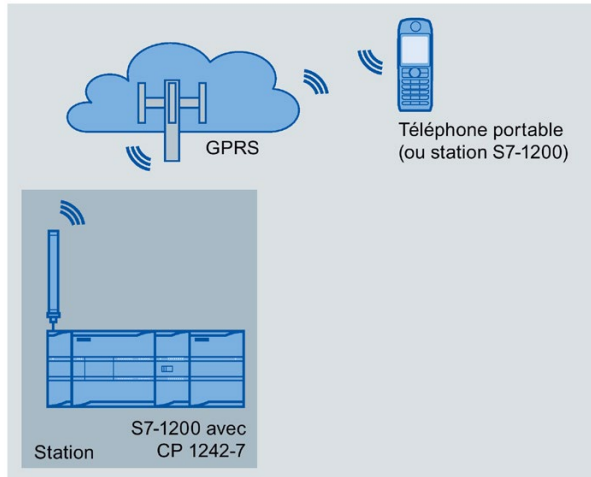
Nom abrégé	N° de référence	Explication
ANT794-3M	6NH9 870-1AA00	Antenne en nappe (900, 1800/1900 MHz) ; étanche pour les utilisations en intérieur et en extérieur ; câble de raccordement de 1,2 m relié de façon permanente à l'antenne ; connecteur SMA, comprenant disque adhésif, montage possible avec vis

Les antennes doivent être commandées séparément.

13.10.7 Exemples de configuration pour telecontrol

Vous trouverez ci-dessous des exemples de configuration pour les stations dotées d'un CP 1242-7.

Envoi de messages par SMS



Un SIMATIC S7-1200 avec un CP 1242-7 peut envoyer des messages par SMS à un téléphone portable ou une station S7-1200 configurée.

Telecontrol par un center de commande

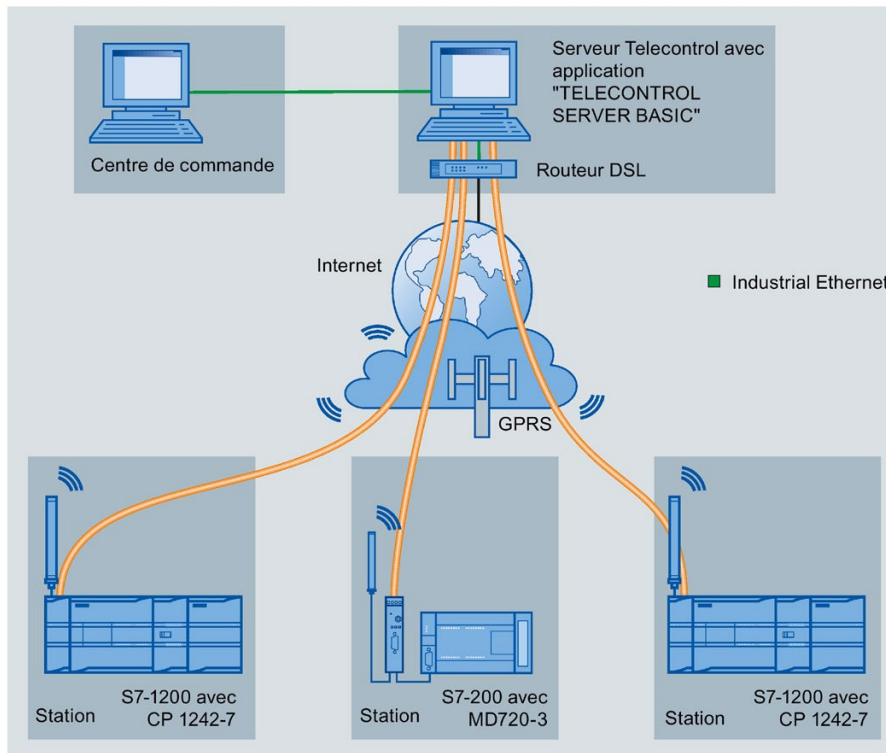


Figure 13-1 Communication entre des stations S7-1200 et un centre de commande

Dans les applications telecontrol, les stations SIMATIC S7-1200 dotées d'un CP 1242-7 communiquent avec un centre de commande via le réseau GSM et l'Internet. L'application "TELECONTROL SERVER BASIC" (TCSB) est installée sur le serveur telecontrol dans la station maître. Cela entraîne les cas d'utilisation suivants :

- Communication Telecontrol entre station et centre de commande

Dans ce cas d'utilisation, les données provenant du terrain sont envoyées par les stations au serveur telecontrol dans la station maître via le réseau GSM et Internet. Le serveur telecontrol est utilisé pour surveiller les stations distantes.
- Communication entre une station et une salle de commande avec un client OPC

Comme dans le premier cas, les stations communiquent avec le serveur telecontrol. A l'aide de son serveur OPC intégré, le serveur telecontrol échange des données avec le client OPC de la salle de commande.

Le client OPC et le serveur telecontrol peuvent se trouver sur un ordinateur unique, par exemple quand TCSB est installé sur l'ordinateur d'un centre de commande avec WinCC.
- Communication inter-stations via un centre de commande

Une communication inter-stations est possible avec des stations S7 équipées d'un CP 1242-7.

Pour permettre une communication inter-station, le serveur telecontrol transmet les messages de la station d'envoi à la station de réception.

Communication directe entre stations

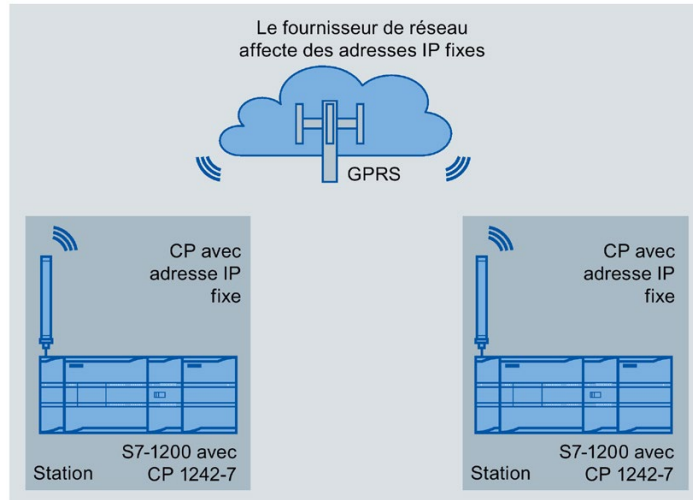


Figure 13-2 Communication directe entre deux stations S7-1200

Dans cette configuration, deux stations SIMATIC S7-1200 communiquent directement l'une avec l'autre à l'aide du CP 1242-7 via le réseau GSM. Chaque CP 1242-7 possède une adresse IP fixe. Le service correspondant du fournisseur de réseau GSM doit permettre cela.

TeleService via GPRS

Dans TeleService via GPRS, une station d'ingénierie sur laquelle STEP 7 est installé communique via le réseau GSM et l'Internet avec le CP 1242-7 dans le S7-1200.

Puisqu'un pare-feu est normalement fermé pour les demandes de connexions provenant de l'extérieur, une station de sectionnement est requise entre la station distante et la station d'ingénierie. Cette station de sectionnement peut prendre la forme d'un serveur telecontrol ou, en l'absence de serveur telecontrol dans la configuration, d'une passerelle TeleService.

TeleService avec serveur telecontrol

La connexion a lieu via le serveur telecontrol.

- La station d'ingénierie et le serveur telecontrol sont connectés via l'Intranet (LAN) ou Internet.
- Le serveur telecontrol et la station distante sont connectés via l'Internet et via le réseau GSM.

La station d'ingénierie et le serveur telecontrol peuvent également être le même ordinateur ; en d'autres termes, STEP 7 et TCSB sont installés sur le même ordinateur.

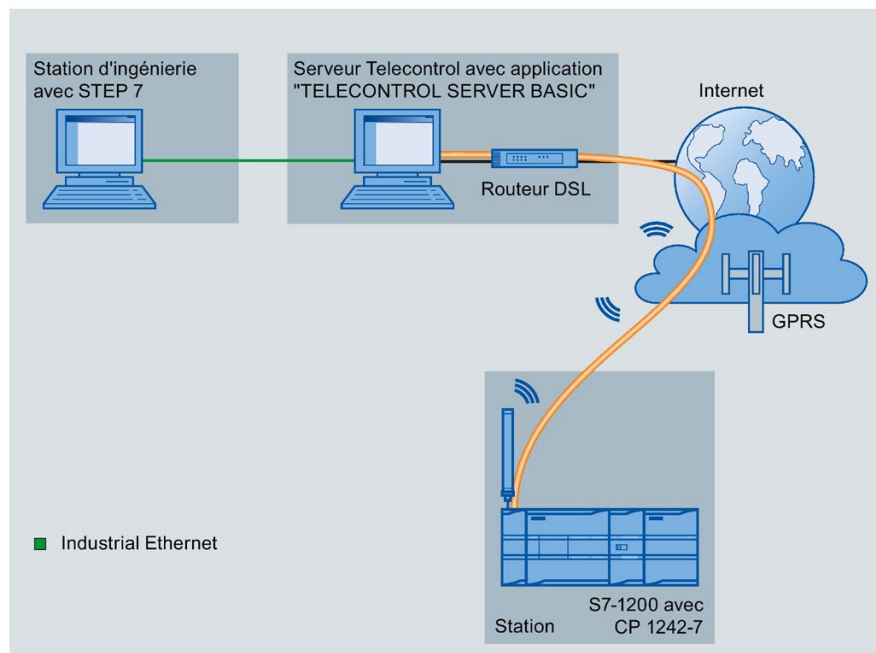


Figure 13-3 TeleService via GPRS dans une configuration avec serveur telecontrol

TeleService sans serveur telecontrol

La connexion a lieu via la passerelle TeleService.

La connexion entre la station d'ingénierie et la passerelle TeleService peut être locale via un réseau LAN ou via l'Internet.

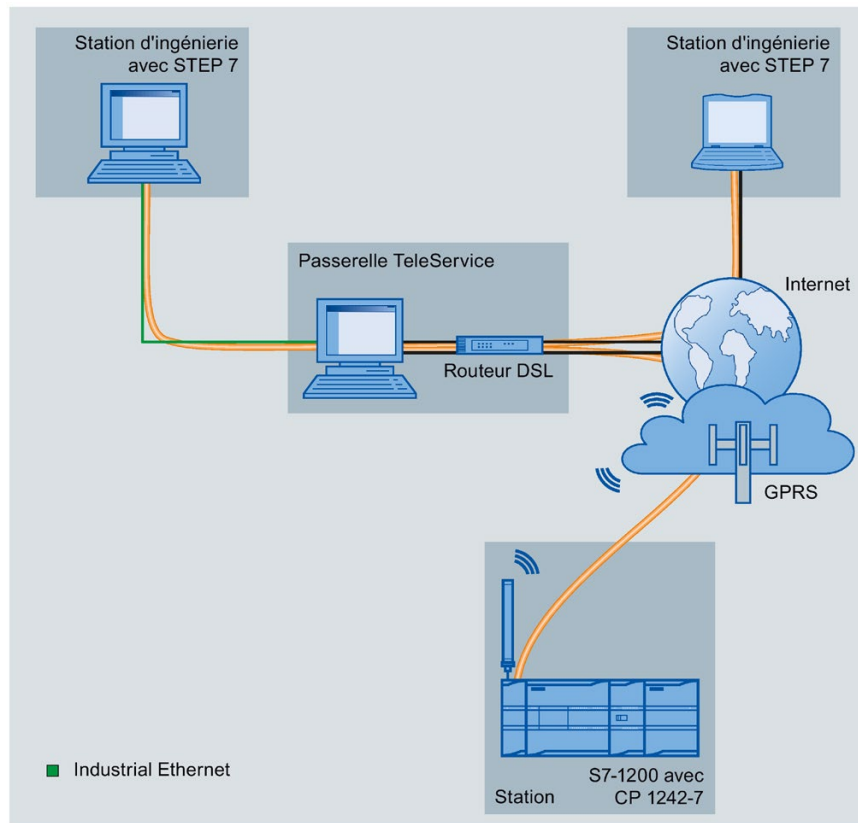


Figure 13-4 TeleService via GPRS dans une configuration avec passerelle TeleService

14.1 Instruction TM_MAIL (Envoyer e-mail)

Tableau 14- 1 Instruction TM_MAIL

CONT/LOG	SCL	Description																		
<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">"TM_MAIL_DB"</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">TM_MAIL</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">EN</td> <td style="text-align: center;">ENO</td> </tr> <tr> <td style="text-align: center;">REQ</td> <td style="text-align: center;">BUSY</td> </tr> <tr> <td style="text-align: center;">ID</td> <td style="text-align: center;">DONE</td> </tr> <tr> <td style="text-align: center;">TO_S</td> <td style="text-align: center;">ERROR</td> </tr> <tr> <td style="text-align: center;">CC</td> <td style="text-align: center;">STATUS</td> </tr> <tr> <td style="text-align: center;">SUBJECT</td> <td></td> </tr> <tr> <td style="text-align: center;">TEXT</td> <td></td> </tr> <tr> <td style="text-align: center;">ATTACHMENT</td> <td></td> </tr> </tbody> </table> </div>	TM_MAIL		EN	ENO	REQ	BUSY	ID	DONE	TO_S	ERROR	CC	STATUS	SUBJECT		TEXT		ATTACHMENT		<pre>"TM_MAIL_DB" (REQ:=_bool_in_, ID:=_int_in_, TO_S:=_string_in_, CC:=_string_in_, SUBJECT:=_string_in_, TEXT:=_string_in_, ATTACHMENT:=_variant_in_, BUSY=>_bool_out_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_,);</pre>	<p>L'instruction TM_MAIL envoie un courrier électronique à l'aide du protocole SMTP (Simple Mail Transfer Protocol) sur TCP/IP par le biais de la liaison CPU Industrial Ethernet. En l'absence de connectivité Internet basée sur Ethernet, on peut utiliser un adaptateur TeleService optionnel pour se connecter aux lignes téléphoniques terrestres. TM_MAIL s'exécute de manière asynchrone et la tâche s'étend sur plusieurs appels TM_MAIL. Vous devez affecter un DB d'instance lorsque vous appelez TM_MAIL. Ce DB d'instance ne doit pas être défini comme rémanent. Ainsi, le DB d'instance est initialisé lors du passage de la CPU de l'état ARRÊT à l'état MARCHE et une nouvelle opération TM_MAIL peut être déclenchée.</p>
TM_MAIL																				
EN	ENO																			
REQ	BUSY																			
ID	DONE																			
TO_S	ERROR																			
CC	STATUS																			
SUBJECT																				
TEXT																				
ATTACHMENT																				

¹ STEP 7 crée automatiquement le DB d'instance lorsque vous insérez l'instruction.

L'envoi du courriel démarre en présence d'un front montant (de 0 à 1) dans le paramètre d'entrée REQ. Le tableau suivant montre les relations entre BUSY, DONE et ERROR. Vous pouvez surveiller l'avancement de l'exécution de TM_MAIL et détecter l'achèvement de l'instruction en évaluant ces paramètres lors d'appels successifs.

Les paramètres de sortie DONE, ERROR, STATUS et SFC_STATUS ne sont valables que pendant un cycle lorsque l'état du paramètre de sortie BUSY passe de 1 à 0. La logique de votre programme doit sauvegarder les valeurs temporaires de l'état de sortie de manière à ce que vous puissiez détecter des changements d'état lors des cycles d'exécution suivants du programme.

Remarque

TM_MAIL envoie un courriel via TCP/IP au moyen de l'interface Ethernet de la CPU. Utilisez l'instruction Instruction TMAIL_C (Envoyer un courriel à l'aide de l'interface Ethernet de la CPU) (Page 913) pour envoyer un courriel via une interface CP (avec ou sans SSL).

14.1 Instruction TM_MAIL (Envoyer e-mail)

Tableau 14- 2 Interactions entre les paramètres Done, Busy et Error

DONE	BUSY	ERROR	Description
Sans objet	1	Sans objet	La tâche est en cours.
1	0	0	La tâche a été achevée avec succès.
0	0	1	La tâche s'est terminée avec une erreur. Vous trouverez la cause de l'erreur dans le paramètre STATUS.
0	0	0	Il n'y a pas de tâche en cours.

Si la CPU passe à l'état ARRET alors que TM_MAIL est actif, la liaison de communication au serveur de courriel est coupée. La liaison de communication au serveur de courriel est également interrompue si des problèmes surviennent dans la communication CPU sur le bus Industrial Ethernet. Dans ces cas, l'opération d'envoi est suspendue et le courriel n'atteint pas son destinataire.

IMPORTANT

Modification des programmes utilisateur

La suppression et le remplacement de blocs de programme, les appels de TM_MAIL ou les appels des DB d'instance de TM_MAIL peuvent rompre le lien entre les blocs de programme. Si vous ne parvenez pas à conserver le lien entre les blocs de programme, les fonctions de communication TCP/IP peuvent prendre un état indéfini, pouvant entraîner des dommages matériels importants. Vous devez exécuter un démarrage à chaud ou à froid de la CPU après avoir transféré un bloc de programme modifié.

Pour éviter de rompre le lien entre les blocs de programme, ne modifiez les parties de votre programme utilisateur qui affectent directement les appels de TM_MAIL que lorsque :

- la CPU est à l'état ARRET,
- aucun courriel n'est envoyé (REQ et BUSY = 0).

Cohérence des données

Le paramètre d'entrée ADDR_MAIL_SERVER est lu au démarrage de l'opération. Une nouvelle valeur ne prend pas effet tant que l'opération en cours n'est pas achevée et qu'une nouvelle opération TM_MAIL n'est pas déclenchée.

En revanche, les paramètres WATCH_DOG_TIME, TO_S, CC, FROM, SUBJECT, TEXT, ATTACHMENT, USERNAME et PASSWORD sont lus pendant l'exécution de TM_MAIL et ne peuvent être modifiés qu'une fois la tâche achevée (BUSY = 0).

Connexion de numérotation : Configuration des paramètres IE de l'adaptateur TS Adapter

Vous devez configurer les paramètres IE de l'adaptateur TeleService pour les appels sortants afin d'établir la liaison au serveur de numérotation de votre fournisseur de services Internet. Si vous activez l'attribut d'appel sur demande, la liaison est établie uniquement lorsqu'un courriel est envoyé. En cas de connexion avec un modem analogique, il faut plus de temps pour établir la liaison (environ une minute en plus). Vous devez inclure ce temps supplémentaire dans la valeur WATCH_DOG_TIME.

Tableau 14- 3Types de données pour les paramètres

Paramètre et type		Types de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération.
ID	IN	Int	Identificateur de la liaison. Voir le paramètre ID des instructions TCON, TDISCON, TSEND et TRCV. Il faut utiliser un nombre qui n'est pas utilisé pour d'autres instances de cette instruction dans le programme utilisateur.
TO_S	IN	String	Adresses des destinataires. Données STRING de 240 caractères au maximum.
CC	IN	String	Adresses de copie conforme (facultatives). Données STRING de 240 caractères au maximum.
SUBJECT	IN	String	Objet du courriel. Données STRING de 240 caractères au maximum.
TEXT	IN	String	Texte du courriel (facultatif). Données STRING de 240 caractères au maximum. Si ce paramètre est une chaîne vide, le courriel sera envoyé sans texte.
ATTACHMENT	IN	Variant	Pointeur désignant les données des pièces jointes. Données octet, mot ou double mot de 65 534 octets au maximum. Si vous n'affectez pas de valeur à ce paramètre, le courriel sera envoyé sans pièce jointe.
DONE	OUT	Bool	<ul style="list-style-type: none"> 0 : Tâche pas encore commencée ou encore en cours d'exécution 1 : Tâche exécutée sans erreur
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 : Pas d'opération en cours 1 : Opération en cours
ERROR	OUT	Bool	Le bit ERROR est égal à 1 pendant un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans la sortie STATUS ne vaut que pendant le cycle où ERROR est égal à 1.
STATUS	OUT	Word	Valeur ou information d'erreur renvoyée par l'instruction TM_MAIL

14.1 Instruction TM_MAIL (Envoyer e-mail)

Paramètre et type		Types de données	Description
ADDR_MAIL_SERVER	¹ Static	DWord	Adresse IP du serveur de courriel. Vous devez définir chaque fragment d'adresse IP en tant qu'octet de deux caractères hexadécimaux de 4 bits. Si un fragment d'adresse est égal à la valeur décimale 10 qui vaut A en hexadécimal, vous devez entrer "0A" pour cet octet. Par exemple : pour une adresse IP = 192.168.0.10 ADDR_MAIL_SERVER = DW#16#C0A8000A, car : <ul style="list-style-type: none"> • 192 = 16#C0, • 168 = 16#A8 • 0 = 16#00 • 10 = 16#0A
WATCH_DOG_TIME	¹ Static	Time	Temps maximal alloué à TM_MAIL pour terminer le process SMTP complet, de l'établissement de la liaison SMTP à la fin de la transmission SMTP. Si ce temps est dépassé, l'exécution de TM_MAIL s'achève avec une erreur. Le temps de retard effectif jusqu'à ce que TM_MAIL s'achève et que l'erreur soit émise peut dépasser WATCH_DOG_TIME en raison du temps supplémentaire requis pour l'opération de déconnexion. Définissez le temps à 2 minutes pour commencer. Cette valeur peut être bien inférieure pour une liaison téléphonique RNIS.
USERNAME	¹ Static	String	Nom d'utilisateur du compte de messagerie. Données STRING de 180 caractères au maximum.
PASSWORD	¹ Static	String	Mot de passe du serveur de messagerie. Données STRING de 180 caractères au maximum.
FROM	¹ Static	String	Adresse de l'émetteur. Chaîne STRING avec 240 caractères au maximum.
SFC_STATUS	¹ Static	Word	Code de condition d'exécution des blocs de communication appelés

¹ Les valeurs de ces paramètres ne sont pas modifiées à chaque appel de TM_MAIL. Elles sont définies dans le bloc de données d'instance de TM_MAIL et ne sont référencées qu'une fois, à savoir lors du premier appel de TM_MAIL.

Authentification SMTP

TM_MAIL prend en charge la méthode d'authentification SMTP AUTH LOGIN. Pour plus d'informations sur cette méthode d'authentification, consultez le manuel de votre serveur de courriel ou le site Web de votre fournisseur de services Internet.

La méthode d'authentification AUTH LOGIN utilise les paramètres USERNAME et PASSWORD de TM_MAIL pour se connecter au serveur de courriel. Le nom d'utilisateur et le mot de passe doivent être préalablement configurés dans un compte de messagerie sur un serveur de courriel.

Si aucune valeur n'est affectée au paramètre USERNAME, la méthode d'authentification AUTH LOGIN n'est pas utilisée et le courriel est envoyé sans authentification.

Paramètres TO_S:, CC: et FROM:

Les paramètres TO_S:, CC: et FROM: sont des chaînes, comme illustré dans les exemples suivants :

TO: <wenna@mydomain.com>, <ruby@mydomain.com> ,

CC: <admin@mydomain.com>, <judy@mydomain.com> ,

FROM: <admin@mydomain.com>

Il faut respecter les règles suivantes lors de l'entrée de ces chaînes de caractères :

- Il faut entrer les caractères "TO:", "CC:" et "FROM:" sans oublier les deux-points.
- Il faut insérer un espace et le signe "<" avant chaque adresse. Il doit, par exemple, y avoir un espace entre "TO:" et <adresse électronique>.
- Il faut terminer chaque adresse par le signe ">".
- Il faut entrer une virgule "," après chaque adresse électronique pour les adresses TO_S: et CC:. Par exemple, la virgule est obligatoire après l'unique adresse électronique dans "TO: <email address>,".
- L'entrée FROM: ne doit contenir qu'une seule adresse électronique, sans virgule à la fin.

En raison du mode d'exécution et de l'utilisation de la mémoire, la syntaxe des données TO_S:, CC: et FROM: de TM_MAIL n'est pas vérifiée. Si les règles de format précédentes ne sont pas exactement respectées, la transaction avec le serveur de courriel SMTP échouera.

Paramètres STATUS et SFC_STATUS

Les codes d'erreur d'exécution renvoyés par TM_MAIL peuvent être classés comme suit :

- W#16#0000 : l'exécution de TM_MAIL s'est achevée avec succès.
- W#16#7xxx : état de l'exécution de TM_MAIL
- W#16#8xxx : erreur lors d'un appel interne adressé à un appareil de communication ou au serveur de messagerie.

Le tableau suivant présente les codes d'erreur d'exécution de TM_MAIL à l'exception des codes d'erreur des modules de communication appelés en interne.

Remarque

Exigences concernant le serveur de courriel

TM_MAIL ne peut communiquer qu'avec un serveur de courriel utilisant SMTP via le port 25. Le numéro de port affecté ne peut pas être modifié.

La plupart des services informatiques et des serveurs de courriel externes bloquent maintenant le port 25 pour empêcher un PC infecté par un virus de devenir un générateur de courriels indésirables.

Vous pouvez vous connecter à un serveur de courriel interne via SMTP et laisser le serveur interne gérer les améliorations de sécurité actuelles qui sont nécessaires pour transmettre le courriel à un serveur de courriel externe via Internet.

Exemple : Configuration de serveur de messagerie interne

Si vous utilisez Microsoft Exchange comme serveur de courriel interne, vous pouvez configurer le serveur pour qu'il autorise l'accès SMTP depuis l'adresse IP affectée à l'API S7-1200. Configurez la console de gestion Exchange : Configuration du serveur > Transport Hub > Connecteurs de réception > Relais IP. L'onglet Réseau comprend une boîte intitulée "Recevoir du courrier à partir de serveurs distants ayant ces adresses IP". C'est là que vous ajouterez l'adresse IP de l'API qui exécute l'instruction TM_MAIL. Aucune authentification n'est requise pour ce type de connexion avec un serveur Microsoft Exchange interne.

Configuration du serveur de courriel

TM_MAIL peut utiliser uniquement un serveur de courriel qui autorise la communication via le port 25, SMTP et l'authentification AUTH LOGIN (facultatif).

Configurez un compte de serveur de courriel compatible afin qu'il accepte l'ouverture de session SMTP à distance. Puis éditez le DB d'instance de TM_MAIL pour y ajouter les chaînes de caractères USERNAME et PASSWORD qui sont utilisées pour authentifier la connexion avec votre compte de messagerie.

Tableau 14- 4 Codes d'erreur

STATUS (W#16#...)	SFC_STATUS (W#16#...)	Description
0000	-	L'opération TM_MAIL s'est achevée sans erreur. Ce code STATUS nul ne garantit pas qu'un courriel a effectivement été envoyé (voir le premier point de la remarque suivant ce tableau).
7001	-	TM_MAIL est actif (BUSY = 1).
7002	7002	TM_MAIL est actif (BUSY = 1).
8xxx	xxxx	L'opération TM_MAIL s'est achevée avec une erreur dans les appels d'instruction de communication internes. Pour plus d'informations sur le paramètre SFC_STATUS, consultez les descriptions du paramètre STATUS des instructions de communication ouverte PROFINET sous-jacentes.
8010	xxxx	La connexion a échoué. Pour plus d'informations sur le paramètre SFC_STATUS, consultez la description du paramètre STATUS de l'instruction TCON.
8011	xxxx	Erreur lors de l'envoi des données. Pour plus d'informations sur le paramètre SFC_STATUS, consultez la description du paramètre STATUS de l'instruction TSEND.
8012	xxxx	Erreur lors de la réception des données. Pour plus d'informations sur le paramètre SFC_STATUS, consultez la description du paramètre STATUS de l'instruction TRCV.
8013	xxxx	La connexion a échoué. Pour plus d'informations sur l'évaluation du paramètre SFC_STATUS, consultez les descriptions du paramètre STATUS des instructions TCON et TDISCON.
8014	-	La connexion a échoué. Vous avez peut-être entré une adresse IP de serveur de courriel incorrecte (ADDR_MAIL_SERVER) ou indiqué trop peu de temps (WATCH_DOG_TIME) pour la connexion. Il est également possible que la CPU n'ait pas de liaison au réseau ou que la configuration de la CPU soit incorrecte.

STATUS (W#16#...)	SFC_STATUS (W#16#...)	Description
8015	-	Pointeur invalide pour le paramètre ATTACHMENT : Utilisez un pointeur Variant avec indication de type de données et longueur. Exemple : "P#DB.DBX0.0" est incorrect et "P#DB.DBX0.0 octet 256" est correct.
82xx, 84xx, 85xx	-	Le message d'erreur provient du serveur de courriel et correspond au code d'erreur "8" du protocole SMTP. Voir le deuxième point de la remarque suivant ce tableau.
8450	-	Pas d'exécution de l'opération : boîte aux lettres électronique non disponible. Réessayez ultérieurement.
8451	-	Annulation de l'opération : Erreur locale lors du traitement. Réessayez ultérieurement.
8500	-	Erreur de syntaxe de commande : peut-être le serveur de courriel n'accepte-t-il pas la procédure d'authentification LOGIN. Vérifiez les paramètres de TM_MAIL. Essayez d'envoyer un courriel sans authentification. Tentez de remplacer le paramètre USERNAME par une chaîne vide.
8501	-	Erreur de syntaxe : paramètre ou argument incorrect. Vous avez peut-être tapé une adresse incorrecte dans les paramètres TO_S ou CC.
8502	-	Commande inconnue ou non implémentée. Vérifiez vos entrées, en particulier le paramètre FROM. Il est peut-être incomplet ou vous avez omis les caractères "@" ou ".".
8535	-	Authentification SMTP inachevée. Vous avez peut-être entré un nom d'utilisateur ou un mot de passe incorrect.
8550	-	Impossible d'atteindre le serveur de courriel ou vous n'avez pas de droits d'accès. Vous avez peut-être entré un nom d'utilisateur ou un mot de passe incorrect ou votre serveur de courriel n'accepte pas l'accès par ouverture de session. Cette erreur peut également être due à une entrée erronée du nom de domaine après le signe "@" dans les paramètres TO_S ou CC.
8552	-	Annulation de l'opération : dépassement de la taille de mémoire allouée. Réessayez ultérieurement.
8554	-	L'émission a échoué. Réessayez ultérieurement.

Remarque

Possibilité d'erreurs d'envoi de courriel non signalées

- L'entrée incorrecte d'une adresse de destinataire ne génère pas d'erreur STATUS pour TM_MAIL. Dans ce cas, il n'y a aucune garantie que les autres destinataires (dont les adresses électroniques ont été correctement indiquées) recevront le courriel.
 - Vous trouverez plus d'informations sur les codes d'erreur SMTP sur Internet ou dans la documentation d'erreur du serveur de courriel. Vous pouvez également lire le dernier message d'erreur en provenance du serveur de courriel. Le message d'erreur est stocké dans le paramètre buffer1 du DB d'instance de TM_MAIL.
-

Outils en ligne et de diagnostic

15.1 DEL d'état

La CPU et les modules d'E/S utilisent des DEL pour fournir des informations sur l'état de fonctionnement du module ou des E/S.

DEL d'état sur une CPU

La CPU fournit les indicateurs d'état suivants :

- STOP/RUN
 - Jaune continu indique l'état ARRET.
 - Vert continu indique l'état MARCHÉ.
 - Vert et jaune clignotant en alternance indiquent que la CPU est à l'état de fonctionnement MISE EN ROUTE.
- ERROR
 - Un clignotant rouge indique une erreur, comme une erreur interne à la CPU, une erreur dans la carte mémoire ou une erreur de configuration (modules mal assortis)
 - Un clignotement rouge pendant trois secondes signale une erreur qui n'est pas permanente. Ce sera, par exemple, le cas si l'horloge temps réel est réinitialisée à l'heure par défaut en raison d'une coupure de courant.
 - Etat défectueux :
 - Rouge continu signale un matériel défectueux.
 - Toutes les DEL clignotent si le firmware détecte un défaut.
- La DEL MAINT (Maintenance) clignote dès que vous insérez une carte mémoire. La CPU passe alors à l'état ARRET. Une fois que la CPU est passée à l'état ARRET, exécutez l'une des actions suivantes pour déclencher l'évaluation de la carte mémoire :
 - Faites passer la CPU à l'état MARCHÉ.
 - Effectuez un effacement général (MRES).
 - Mettez la CPU hors tension puis sous tension.

Vous pouvez également utiliser l'instruction LED (Page 461) pour déterminer l'état des DEL.

Tableau 15- 1DEL d'état pour une CPU

Description	STOP/RUN Jaune / Vert	ERROR Rouge	MAINT Jaune
Hors tension	Eteint	Eteint	Eteint
Démarrage, auto-test ou actualisation du firmware	Clignotant (jaune et vert en alternance)	-	Eteint
Etat ARRET	Allumé (jaune)	-	-
Etat MARCHE	Allumé (vert)	-	-
Enlever la carte mémoire	Allumé (jaune)	-	Clignotant
Erreur	Allumé (soit jaune soit vert)	Clignotant	-
Maintenance requise <ul style="list-style-type: none"> E/S forcées Remplacer la pile (si le Battery Board est installé) 	Allumé (soit jaune soit vert)	-	Allumé
Matériel défectueux	Allumé (jaune)	Allumé	Eteint
Test des DEL ou firmware CPU défectueux	Clignotant (jaune et vert en alternance)	Clignotant	Clignotant
Version de configuration de la CPU inconnue ou incompatible	Allumé (jaune)	Clignotant	Clignotant

Remarque**Erreur "Version de configuration de la CPU inconnue ou incompatible"**

Tenter de télécharger un programme S7-1200 V3.0 dans une CPU de S7-1200 V4.0 provoque une erreur de CPU et elle affiche un message d'erreur correspondant dans la mémoire tampon de diagnostic. Si vous obtenez cet état en utilisant une version invalide de carte transfert de programme (Page 147), retirez la carte, effectuez une transition de ARRET à MARCHE, un effacement général de la mémoire (MRES) ou mettez la CPU hors tension puis sous tension. Si vous obtenez cet état en téléchargeant un programme invalide, restaurez les réglages d'usine de la CPU (Page 1387). Après avoir supprimé la condition d'erreur de la CPU, vous pouvez télécharger un programme de CPU V4.0 valide.

La CPU fournit également deux DEL qui indiquent l'état de la communication PROFINET. Ouvrez le cache du bornier inférieur pour voir les DEL PROFINET.

- Link (vert) s'allume pour signaler qu'une connexion a été établie avec succès.
- Rx/Tx (jaune) s'allume pour signaler une activité de transmission.

La CPU et chaque module d'entrées-sorties TOR (SM) fournissent une DEL I/O Channel pour chacune des entrées et sorties TOR. La DEL I/O Channel (verte) s'allume ou s'éteint pour indiquer l'état de l'entrée ou sortie correspondante.

Comportement du S7-1200 à la suite d'une erreur fatale

Si le firmware de la CPU détecte une erreur fatale, il tente un redémarrage en mode secours et, s'il réussit, signale le mode secours en faisant clignoter en continu les DEL STOP/RUN, ERROR et MAINT. Le programme utilisateur et la configuration matérielle ne sont pas chargés après le redémarrage en mode secours.

Si la CPU mène à bien le redémarrage en mode secours, elle exécute les actions suivantes :

- Elle donne la valeur 0 aux sorties de la CPU et du Signal Board.
- Elle donne aux sorties des modules d'entrées-sorties du châssis central et de la périphérie décentralisée la valeur définie pour "Réaction à l'arrêt de la CPU" dans la configuration d'appareil des sorties TOR du module.

Si le redémarrage en mode secours échoue (en raison d'une erreur matérielle par exemple), les DEL STOP et ERROR sont allumées et la DEL MAINT est éteinte.

ATTENTION

Le fonctionnement dans un état défectueux ne peut pas être garanti

Les appareils de commande peuvent présenter des défaillances dans des situations non sûres et provoquer un fonctionnement inattendu des appareils pilotés pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Vous devez donc utiliser une fonction d'arrêt d'urgence, des dispositifs de sécurité électromécaniques ou d'autres sécurités redondantes qui soient indépendants de l'API.

DEL d'état sur un SM

En outre, chaque module SM TOR fournit une DEL DIAG qui indique l'état du module :

- Vert indique que le module est opérationnel.
- Rouge indique que le module est défectueux ou n'est pas opérationnel.

Chaque module d'entrées-sorties analogiques fournit une DEL I/O Channel pour chacune des entrées et sorties analogiques.

- Vert indique que la voie a été configurée et est active.
- Rouge signale une situation d'erreur de l'entrée ou sortie analogique correspondante.

En outre, chaque module SM analogique fournit une DEL DIAG qui indique l'état du module :

- Vert indique que le module est opérationnel.
- Rouge indique que le module est défectueux ou n'est pas opérationnel.

Le module d'entrées-sorties détecte la présence ou l'absence de courant vers le module (alimentation côté site, si requis).

Tableau 15- 2DEL d'état pour un module d'entrées-sorties (SM)

Description	DIAG (rouge / vert)	I/O Channel (rouge / vert)
L'alimentation côté champ est coupée*.	Rouge clignotant	Rouge clignotant
Non configuré ou mise à jour en cours	Vert clignotant	Eteint
Module configuré sans erreur	Allumé (vert)	Allumé (vert)
Situation d'erreur	Rouge clignotant	-
Erreur d'E/S (avec diagnostic activé)	-	Rouge clignotant
Erreur d'E/S (avec diagnostic désactivé)	-	Allumé (vert)

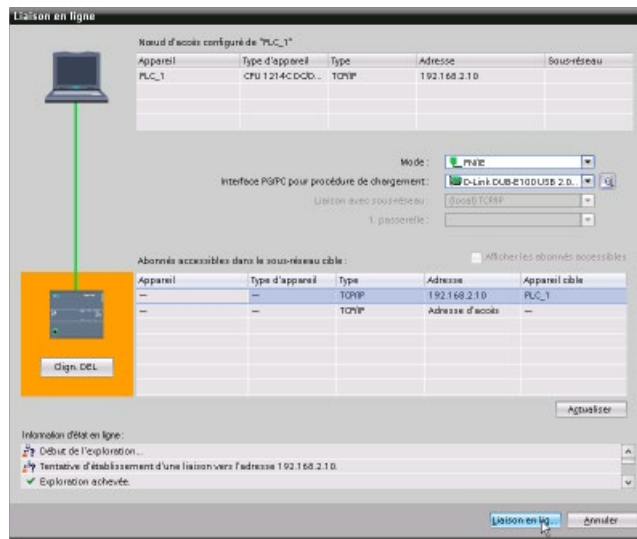
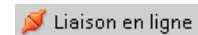
* Cet état n'est pris en charge que sur les modules d'entrées-sorties analogiques.

15.2 Passage en ligne et connexion à une CPU

Vous devez établir une liaison en ligne entre la console de programmation et la CPU pour charger des programmes et des données techniques de projet, ainsi que pour des activités telles que les suivantes :

- Test des programmes utilisateur
- Affichage et changement de l'état de fonctionnement de la CPU (Page 1392)
- Affichage et réglage de la date et de l'heure de la CPU (Page 1386)
- Affichage des informations sur les modules
- Comparaison et synchronisation (Page 1394) hors ligne par rapport à des blocs de programme en ligne
- Chargement de blocs de programme depuis la CPU et dans la CPU
- Affichage du diagnostic et de la mémoire tampon de diagnostic (Page 1393)
- Utilisation d'une table de visualisation (Page 1400) pour tester le programme utilisateur par visualisation et forçage de valeurs
- Utilisation d'une table de forçage permanent pour forcer des valeurs de manière permanente dans la CPU (Page 1403)

Pour établir une liaison en ligne à une CPU configurée, cliquez sur la CPU dans l'arborescence du projet et cliquez sur le bouton "Liaison en ligne" dans la vue du projet :



Si c'est la première fois que vous allez en ligne avec cette CPU, vous devez sélectionner le type d'interface PG/PC et l'interface PG/PC spécifique dans la boîte de dialogue Liaison en ligne avant d'établir une liaison en ligne à une CPU trouvée sur cette interface.

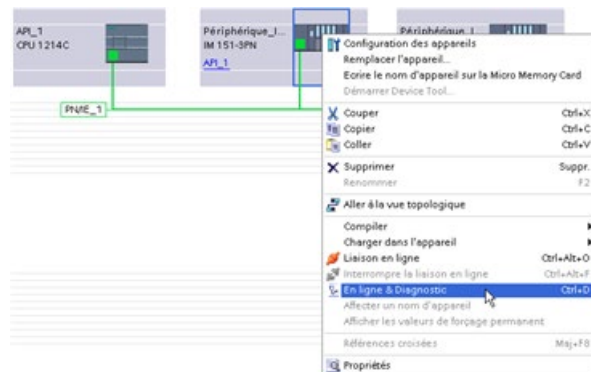
Vous avez maintenant connecté votre console de programmation à la CPU. Les cadres de couleur orange correspondent à une liaison en ligne. Vous pouvez désormais utiliser les outils en ligne et de diagnostic dans l'arborescence du projet et dans la Task Card Outils en ligne.

15.3 Affectation d'un nom à un périphérique PROFINET IO en ligne

Vous devez affecter un nom aux appareils sur votre réseau PROFINET pour pouvoir vous connecter à la CPU. Servez-vous de l'éditeur "Appareils & réseaux" pour affecter un nom à vos appareils PROFINET s'ils n'en ont pas déjà un ou s'il faut le modifier.

Vous devez définir, pour chaque périphérique PROFINET IO, le même nom dans le projet STEP 7 et, à l'aide de l'outil "En ligne & diagnostic", dans la mémoire de configuration de périphérique PROFINET IO (par exemple, mémoire de configuration de coupleur ET200S). Si un nom manque ou ne coïncide pas à l'un ou l'autre endroit, le mode d'échange de données PROFINET IO ne fonctionnera pas.

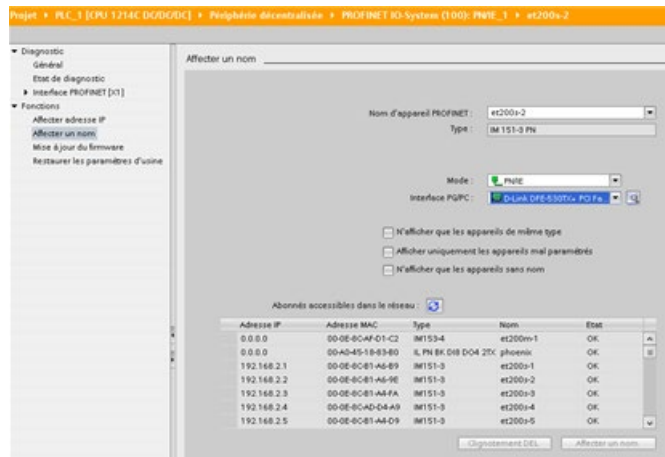
1. Dans l'éditeur "Appareils & réseaux", cliquez avec le bouton droit de la souris sur le périphérique PROFINET IO concerné et sélectionnez "En ligne & diagnostic".



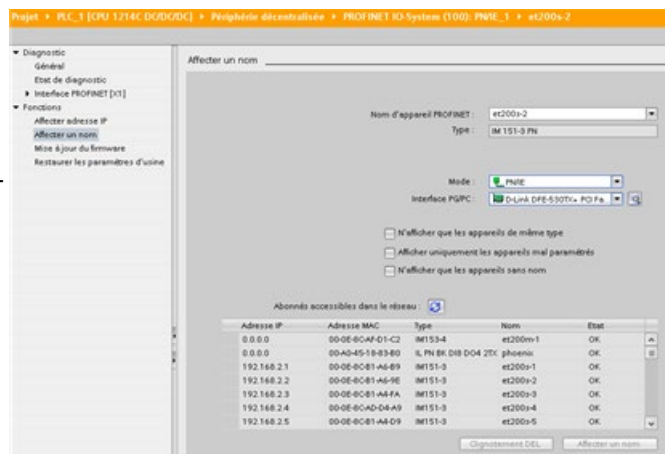
2. Faites les sélections de menu suivantes dans la boîte de dialogue "En ligne & Diagnostic" :

- "Fonctions"
- "Affecter le nom d'appareil PROFINET"

Cliquez sur le bouton "Actualiser la liste" pour afficher tous les appareils PROFINET IO dans le réseau.



3. Dans la liste affichée, cliquez sur le périphérique PROFINET IO concerné et cliquez sur le bouton "Affecter un nom" pour écrire le nom dans la mémoire de configuration de périphérique PROFINET IO.



15.4 Réglage de l'adresse IP et de l'heure

Vous pouvez régler l'adresse IP (Page 838) et l'heure dans la CPU en ligne. Après avoir accédé à "En ligne et Diagnostics" depuis l'arborescence du Projet pour une CPU en ligne, vous pouvez afficher ou modifier l'adresse IP. Vous pouvez également afficher ou régler les paramètres de date et d'heure de la CPU en ligne.



Remarque

Cette fonction est disponible uniquement pour une CPU qui soit dispose uniquement d'une adresse MAC (pas encore d'adresse IP attribuée), soit a été réinitialisée aux réglages d'usine.

15.5 Restauration des réglages d'usine

Vous pouvez restaurer les réglages d'usine d'origine d'un S7-1200 dans les conditions suivantes :

- La CPU dispose d'une liaison en ligne.
- La CPU est à l'état ARRET.

Remarque

Si la CPU est à l'état MARCHE et que vous lancez l'opération de restauration, vous pourrez la faire passer à l'état ARRET après acquittement d'un message de confirmation.

Marche à suivre

Procédez comme suit pour restaurer les réglages d'usine d'une CPU :

1. Ouvrez la vue En ligne & Diagnostic de la CPU.
2. Sélectionnez "Restaurer les réglages d'usine" dans le dossier "Fonctions".
3. Cochez la case "Conserver l'adresse IP" si vous voulez conserver l'adresse IP ou "Effacer l'adresse IP" si vous voulez effacer l'adresse IP.
4. Cliquez sur le bouton "Restaurer".
5. Acquittez le message de confirmation en cliquant sur "OK".

Résultat

Le module passe à l'état ARRET si nécessaire et les réglages d'usine sont restaurés. La CPU effectue les actions suivantes :

si une carte mémoire est installée dans la CPU	si aucune carte mémoire n'est installée dans la CPU
<ul style="list-style-type: none"> • Efface la mémoire tampon de diagnostic • Réinitialise l'heure • Restaure la mémoire de travail de la carte mémoire • Met toutes les zones d'opérandes aux valeurs initiales configurées • Met tous les paramètres aux valeurs configurées • Conserve ou efface l'adresse IP en fonction de la sélection effectuée. (L'adresse MAC est fixe et n'est jamais modifiée)¹. • Supprime l'enregistrement des données de commande (Page 165), le cas échéant 	<ul style="list-style-type: none"> • Efface la mémoire tampon de diagnostic • Réinitialise l'heure • Efface la mémoire de travail et la mémoire de chargement interne • Met toutes les zones d'opérandes aux valeurs initiales configurées • Met tous les paramètres aux valeurs configurées • Conserve ou efface l'adresse IP en fonction de la sélection effectuée. (L'adresse MAC est fixe et n'est jamais modifiée)¹. • Supprimer l'enregistrement des données de commande, le cas échéant

¹ Si vous avez sélectionné "Conserver l'adresse IP", la CPU règle l'adresse IP, le masque de sous-réseau et l'adresse du routeur (le cas échéant) selon les paramètres de votre configuration matérielle, à moins que vous n'ayez modifié ces valeurs dans le programme utilisateur ou un autre outil, auquel cas la CPU restaure les valeurs modifiées.

15.6 Mise à jour du firmware

Vous pouvez actualiser le firmware d'une CPU connectée à partir des outils En ligne et diagnostic de STEP 7 de l'une des deux manières suivantes :

- mise à jour à partir de la CPU dans le projet,
- mise à jour à partir des abonnés accessibles dans le navigateur du projet.

Mise à jour du firmware d'une CPU dans votre projet

Procédez comme suit pour mettre à jour le firmware :

1. Ouvrez la CPU dans le navigateur de projet qui correspond à la CPU connectée.
2. Ouvrez la vue En ligne & Diagnostic de la CPU connectée.
3. Sélectionnez "Mise à jour du firmware" dans le dossier "Fonctions".
4. Dans la section "Outil de mise à jour du firmware", cliquez sur le bouton Parcourir et naviguez jusqu'à l'emplacement contenant le fichier de mise à jour du firmware. Il peut s'agir d'un emplacement sur votre disque dur où vous avez téléchargé un fichier de mise à jour du firmware S7-1200 (<http://support.automation.siemens.com/WW/view/fr/34612486/133100>) à partir du site Web Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/fr>).
5. Sélectionnez un fichier qui est compatible avec votre module. Le tableau présente les modules compatibles correspondant au fichier sélectionné.
6. Cliquez sur le bouton "Démarrer mise à jour". Suivez les boîtes de dialogue, si nécessaire, pour changer l'état de fonctionnement de votre CPU.

STEP 7 affiche des boîtes de dialogue d'avancement pendant qu'il charge la mise à jour du logiciel. Lorsqu'il a terminé, il vous demande de démarrer le module avec le nouveau firmware.

Remarque

Si vous décidez de ne pas démarrer le module avec le nouveau firmware, le firmware précédent reste actif jusqu'à ce que vous réinitialisiez le module, par exemple par mise hors tension puis sous tension. Le nouveau firmware ne devient actif qu'après réinitialisation du module.

Mise à jour du firmware à partir des abonnés accessibles

Procédez comme suit pour effectuer une mise à jour du firmware pour un ou plusieurs abonnés accessibles :

1. Ouvrez "Accès en ligne" dans le navigateur du projet.
2. Ouvrez l'interface de communication à laquelle votre CPU est connectée.
3. Double-cliquez sur "Mettre à jour les abonnés accessibles" et attendez que STEP 7 affiche les appareils en ligne.
4. Ouvrez la CPU que vous voulez mettre à jour et double-cliquez sur "En ligne & diagnostic".
5. Développez "Mise à jour du firmware" dans le dossier "Fonctions". L'automate programmable ainsi que les modules locaux pour l'automate sont affichés. Sélectionnez soit "API" soit "Modules locaux" et exécutez la mise à jour du firmware à partir de la section "Outil de mise à jour du firmware" comme décrit ci-dessus.

Vous pouvez également mettre à jour le firmware au moyen de l'une des méthodes suivantes :

- Utilisation d'une Carte mémoire SIMATIC (Page 153)
- Utilisation de la page Web standard "Information sur les modules" du serveur Web (Page 1060)
- Utilisation de SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/fr/view/98161300>)

15.7 Formatage d'une carte mémoire SIMATIC à partir de STEP 7

Vous pouvez formater la carte mémoire dans une CPU connectée à partir des outils En ligne et diagnostic de STEP 7. Procédez comme suit :

1. Assurez-vous que la CPU est à l'état ARRÊT. Notez que, si la CPU est à l'état MARCHE et que vous démarrez une opération de formatage, STEP 7 vous demande de l'autoriser à mettre la CPU à l'état ARRÊT.
2. Insérez une carte mémoire dans la CPU connectée.
3. Ouvrez En ligne & diagnostic pour la CPU connectée à partir de la CPU dans le projet ou à partir des abonnés accessibles dans Accès en ligne dans le navigateur du projet.
4. Si la CPU n'est pas en ligne, sélectionnez "Liaison en ligne" pour la CPU connectée.
5. Sélectionnez "Formater la carte mémoire" dans le menu "Fonction".
6. Cliquez sur "Formater".
7. Répondez au message de confirmation par "Oui".

STEP 7 formate alors la carte mémoire et affiche un message dans la fenêtre d'information lorsqu'il a terminé. La CPU est à l'état ARRÊT à l'achèvement de l'opération de formatage et les voyants STOP et MAINT clignotent. Vous ne pouvez pas passer à l'état MARCHE à ce stade. Vous devez d'abord procéder à l'une des opérations suivantes :

- Retirer la carte mémoire et redémarrer la CPU : Si la mémoire de chargement interne de la CPU contient un programme, la CPU démarre avec le programme.
- Redémarrer la CPU sans retirer la carte mémoire : Si la mémoire de chargement interne de la CPU contient un programme, la CPU le copie dans la carte mémoire et démarre avec ce programme. Si la mémoire de chargement interne de la CPU ne contient pas de programme, la CPU change la carte mémoire en une carte programme (Page 150) et attend un chargement.

Remarque

Formater une carte mémoire n'a pas d'effet sur le contenu de la mémoire de chargement interne.

Si la CPU utilisait la mémoire de chargement interne lorsque vous avez inséré la carte mémoire et que vous n'avez pas redémarré la CPU entre l'insertion de la carte et l'exécution de l'opération de formatage, la CPU conserve le contenu de la mémoire de chargement interne.

15.8 Panneau de commande CPU de la CPU en ligne



Le "Panneau de commande CPU" affiche l'état de fonctionnement (ARRET ou MARCHE) de la CPU en ligne. Il montre également si la CPU a rencontré une erreur ou si des valeurs sont en cours de forçage.

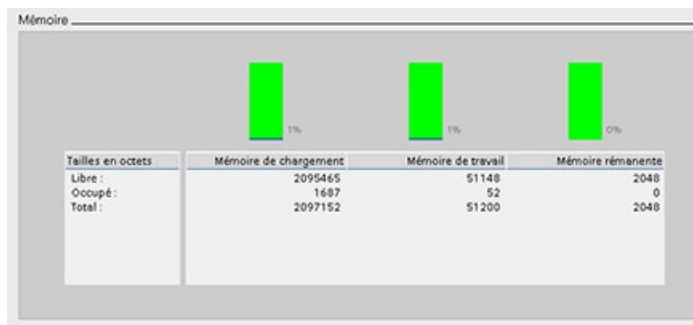
Utilisez le panneau de commande CPU de la Task Card Outils en ligne pour changer l'état de fonctionnement d'une CPU en ligne. La Task Card Outils en ligne est accessible dès que la CPU est en ligne.

15.9 Surveillance du temps de cycle et de l'utilisation de la mémoire

Vous pouvez surveiller le temps de cycle et l'utilisation de la mémoire d'une CPU en ligne.

Après connexion à la CPU en ligne, ouvrez la Task Card Outils en ligne pour visualiser les mesures suivantes :

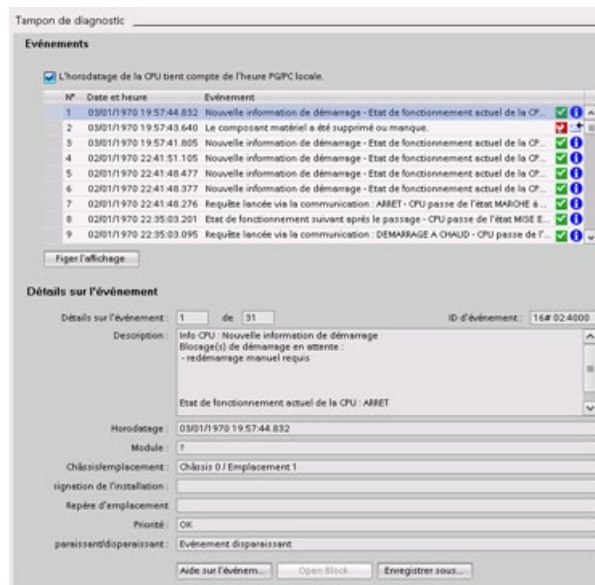
- Temps de cycle
- Utilisation de la mémoire



15.10 Affichage des événements de diagnostic dans la CPU

Servez-vous de la mémoire tampon de diagnostic pour visualiser les activités récentes dans la CPU. La mémoire tampon de diagnostic est accessible à partir de "En ligne & diagnostic" pour une CPU en ligne dans l'arborescence du projet. Elle contient les entrées suivantes :

- Événements de diagnostic
- Changements d'état de fonctionnement de la CPU (passages à l'état ARRET ou MARCHE)



La première entrée contient l'événement le plus récent. Chaque entrée dans la mémoire tampon de diagnostic contient la date et l'heure auxquelles l'événement a été consigné, ainsi qu'une description.

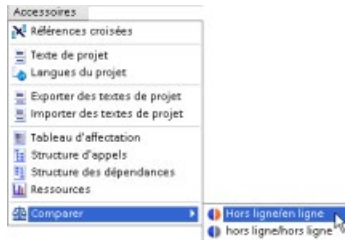
Le nombre maximum d'entrées dépend de la CPU. 50 entrées au maximum sont acceptées.

Seuls les dix événements les plus récents dans la mémoire tampon de diagnostic sont stockés de façon permanente. La restauration des réglages d'usine dans la CPU réinitialise la mémoire tampon de diagnostic en y effaçant les entrées.

Vous pouvez également utiliser l'instruction GET_DIAG (Page 491) pour collecter les informations de diagnostic.

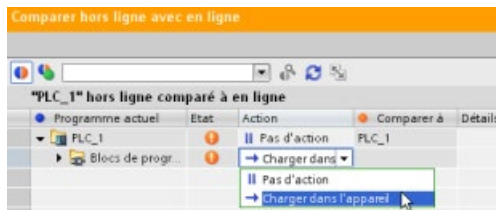
15.11 Comparaison de CPU hors ligne et en ligne

Vous pouvez comparer les blocs de code dans une CPU en ligne avec les blocs de code de votre projet. Si les blocs de code de votre projet ne correspondent pas à ceux de la CPU en ligne, l'éditeur de comparaison vous permet de synchroniser votre projet avec la CPU en ligne soit en chargeant les blocs de code de votre projet dans la CPU, soit en effaçant les blocs du projet qui n'existent pas dans la CPU en ligne.



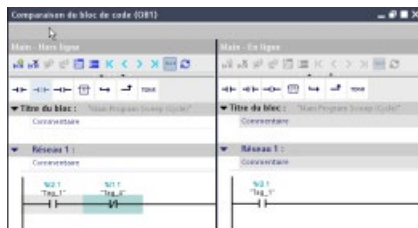
Sélectionnez la CPU dans votre projet.

Utilisez la commande "Comparer hors ligne/en ligne" pour ouvrir l'éditeur "Comparer" (accédez à la commande soit depuis le menu "Outils", soit en cliquant avec le bouton droit de la souris sur la CPU dans votre projet).



Cliquez dans la colonne "Action" d'un objet et choisissez de supprimer l'objet, de ne rien faire ou de charger l'objet dans l'appareil.

Cliquez sur le bouton "Synchroniser" pour charger les blocs de code.



Cliquez avec le bouton droit de la souris sur un objet dans la colonne "Comparer à" et cliquez sur le bouton "Démarrer la comparaison détaillée" pour afficher les blocs de code côte à côte.

La comparaison détaillée met en évidence les différences entre les blocs de code de la CPU en ligne et les blocs de code de la CPU dans votre projet.

Remarque

Accès en lecture requis sur les CPU protégées pour les opérations de comparaison hors ligne/en ligne

Pour STEP 7 V14 ou versions supérieures, le niveau de sécurité "Accès IHM" est insuffisant pour effectuer les opérations de comparaison hors ligne/en ligne. Vous devez disposer de "Accès en lecture" ou de "Accès complet" pour effectuer des opérations de comparaison hors ligne/en ligne.


Voir aussi Protection d'accès pour la CPU (Page 218)

15.12 Exécution d'une comparaison hors ligne/en ligne de la topologie

Depuis la vue synoptique de la topologie dans STEP 7, vous pouvez comparer la topologie hors ligne configurée et la topologie réelle en ligne.






Marche à suivre

Afin de déterminer les différences entre une topologie configurée et une topologie réelle, procédez comme suit :



1. Affichez le tableau synoptique de la topologie dans la vue topologique.
2. Dans la barre d'outils de la vue d'ensemble de la topologie, cliquez sur le bouton "Comparaison hors ligne/en ligne" : 


Résultat

STEP 7 supprime les colonnes "Station partenaire", "Interface partenaire" et "Données de câble" dans le tableau synoptique de la topologie et insère les colonnes "Etat" et "Action" pour la comparaison. Pour chaque appareil ou port dans la vue d'ensemble de la topologie, la colonne Etat affiche le résultat de la comparaison sous la forme suivante :

Icône	Signification
	Différente topologie dans au moins un composant subordonné
	Typologie identique
	Informations sur la topologie disponibles hors ligne uniquement ou appareil désactivé
	Informations sur la topologie disponibles en ligne uniquement
	Typologie différente
	L'appareil ne supporte pas de fonctions topologiques

Pour chaque port ou appareil comparé, la colonne Action offre les choix suivants :

Icône	Signification
	Aucune action n'est possible
	Reprise de la connexion déterminée en ligne

Pour répéter la comparaison, cliquez sur le bouton  dans la barre d'outils de la vue d'ensemble de la topologie.

Pour plus d'informations sur la vue topologique, la vue d'ensemble de la topologie et les comparaisons hors ligne/en ligne de la topologie, reportez-vous système d'information de STEP 7. Vous trouverez également des informations additionnelles dans le manuel PROFINET avec STEP 7 V13

(<https://support.industry.siemens.com/cs/ww/fr/view/49948856>).

15.13 Visualisation et forçage de valeurs dans la CPU

STEP 7 fournit des outils en ligne pour la surveillance de la CPU :

- Vous pouvez visualiser les valeurs en cours des variables. La fonction de visualisation ne modifie pas la séquence du programme. Elle vous donne des informations sur la séquence du programme et les données du programme dans la CPU.
- Vous pouvez également utiliser d'autres fonctions pour commander la séquence et les données du programme utilisateur :
 - Vous pouvez forcer la valeur des variables dans la CPU en ligne pour voir comment le programme utilisateur réagit.
 - Vous pouvez forcer de manière permanente une sortie de périphérie (Q0.1:P ou "Demarr":P, par exemple) à une valeur spécifique.
 - Vous pouvez débloquer les sorties à l'état ARRET.

Remarque

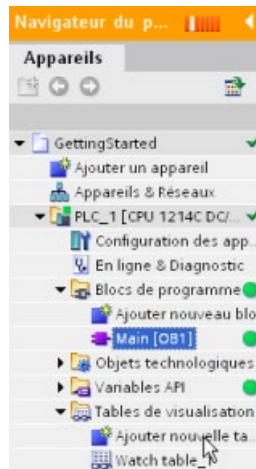
Soyez toujours prudent lorsque vous utilisez des fonctions de forçage. Ces fonctions peuvent avoir une influence grave sur l'exécution du programme utilisateur/système.

Tableau 15- 3 Fonctionnalités en ligne des éditeurs STEP 7

Editeur	Visualisation	Forçage	Forçage permanent
Table de visualisation	Oui	Oui	Non
Table de forçage permanent	Oui	Non	Oui
Editeur de programmes	Oui	Oui	Non
Table des variables	Oui	Non	Non
Editeur de DB	Oui	Non	Non

15.13.1 Passage en ligne pour visualiser les valeurs dans la CPU

Pour visualiser les variables, une liaison en ligne doit être établie à la CPU. Il vous suffit de cliquer sur le bouton "Liaison en ligne" de la barre d'outils.



Une fois que vous êtes connecté à la CPU, STEP 7 affiche les entêtes des zones de travail en orange.

L'arborescence du projet affiche une comparaison du projet hors ligne et de la CPU en ligne. Un cercle vert signifie que la CPU et le projet sont synchronisés, c'est-à-dire que tous deux ont la même configuration et le même programme utilisateur.

Les tables de variables montrent les variables. Les tables de visualisation peuvent également montrer les variables, ainsi que les adresses directes.

	Nom	Adresse	Format d'affichage	Valeur visualisati...	Valeur de forçage
1	*On*	%I0.0	Bool		
2	*Off*	%I0.1	Bool		
3	*Run*	%Q0.0	Bool		

Pour visualiser l'exécution du programme utilisateur et afficher les valeurs des variables, cliquez sur le bouton "Visualiser tout" de la barre d'outils.

	Nom	Adresse	Format d'affichage	Valeur visualisati...	Valeur de forçage
1	*On*	%I0.0	Bool	<input type="checkbox"/> FALSE	
2	*Off*	%I0.1	Bool	<input type="checkbox"/> FALSE	
3	*Run*	%Q0.0	Bool	<input type="checkbox"/> FALSE	

Le champ "Valeur de visualisation" indique la valeur de chaque variable.

15.13.2 Affichage de l'état dans l'éditeur de programme

Vous pouvez visualiser l'état d'un maximum de 50 variables dans les éditeurs de programme CONT et LOG. Utilisez la barre des éditeurs pour afficher l'éditeur CONT. La barre des éditeurs vous permet de basculer entre les éditeurs ouverts sans avoir à les ouvrir ou les fermer.

Dans la barre d'outils de l'éditeur de programme, cliquez sur le bouton "Marche/arrêt visualisation" pour afficher le statut de votre programme utilisateur.



Le réseau dans l'éditeur de programme affiche le flux de courant en vert.

Vous pouvez également faire un clic droit sur l'opération ou le paramètre pour modifier la valeur pour l'opération.

15.13.3 Acquisition d'un instantané des valeurs en ligne d'un DB en vue de la restauration



Vous pouvez acquérir un instantané des valeurs en cours des variables de blocs de données dans une CPU en ligne pour une utilisation ultérieure.

Prenez note des conditions suivantes :

- Une liaison en ligne doit être établie à la CPU.
- Le DB doit être ouvert dans STEP 7.


Acquisition d'un instantané

Procédez comme suit pour l'acquisition d'un instantané :

1. Dans l'éditeur de DB, cliquez sur le bouton "Visualiser tout" :  La colonne "Valeur de visualisation" affiche les valeurs de données en cours.
2. Cliquez sur le bouton  pour acquérir un instantané des valeurs en cours et les afficher dans la colonne "Instantané".

Vous pourrez utiliser cet instantané ultérieurement pour actualiser les valeurs en cours de la CPU ou pour remplacer les valeurs initiales.

Copie des valeurs d'instantané dans la CPU


Pour copier les valeurs d'instantané dans les valeurs en cours des variables de blocs de données dans la CPU, cliquez sur le bouton suivant : 

La CPU en ligne charge les valeurs d'instantané dans les valeurs en cours. La colonne "Valeur de visualisation" montre les valeurs en cours dans la CPU. Il est possible que le cycle modifie ensuite les valeurs dans la CPU à des valeurs différentes de celles de l'instantané mais, au moment de la copie, la CPU effectue un chargement cohérent des valeurs d'instantané.

Remarque

Soyez conscient du fait que, si votre instantané contient des informations d'état, des valeurs de temporisation ou des informations calculées, la CPU restaure ces valeurs telles qu'elles étaient au moment où vous avez pris l'instantané.

Copie des valeurs d'instantané dans les valeurs initiales

Pour copier les valeurs d'instantané dans les valeurs initiales des variables de blocs de données, cliquez sur le bouton suivant : 

Lorsque vous aurez compilé le DB et que vous l'aurez chargé dans la CPU, le DB utilisera les nouvelles valeurs initiales après le passage de la CPU à l'état MARCHE.

Copie de valeurs d'instantané ou de valeurs de visualisation individuelles dans les valeurs initiales

L'éditeur de bloc de données permet de copier des valeurs individuelles et de les coller à la place des valeurs initiales. Il suffit de cliquer avec le bouton droit de la souris dans n'importe quelle colonne de valeur et de sélectionner Copier pour la placer dans le presse-papiers Windows. Vous pouvez alors cliquer avec le bouton droit de la souris sur n'importe quelle valeur initiale et sélectionner Coller pour remplacer cette valeur par celle figurant dans le presse-papiers.

Lorsque vous aurez compilé le DB et que vous l'aurez chargé dans la CPU, le DB utilisera les nouvelles valeurs initiales après le passage de la CPU à l'état MARCHE.

15.13.4 Utilisation d'une table de visualisation pour visualiser et forcer des valeurs dans la CPU

Une table de visualisation vous permet d'exécuter des fonctions de visualisation et de forçage sur des données pendant que la CPU exécute votre programme. Il peut s'agir de données de la mémoire image (I ou Q), de mémentos, de DB ou d'entrées physiques (I_:P) en fonction de la fonction de visualisation ou de forçage. Vous ne pouvez pas visualiser les sorties physiques (Q_:P) avec précision, car la fonction de visualisation peut uniquement afficher la dernière valeur écrite depuis la mémoire Q et ne lit pas la valeur réelle dans les sorties physiques.

La fonction de visualisation ne modifie pas la séquence du programme. Elle vous donne des informations sur la séquence du programme et les données du programme dans la CPU.

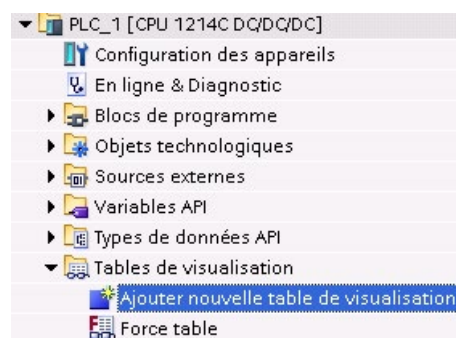
Les fonctions de forçage permettent à l'utilisateur de gérer la séquence et les données du programme. Vous devez être prudent lorsque vous utilisez des fonctions de forçage. Ces fonctions peuvent avoir une influence grave sur l'exécution du programme utilisateur/système. Les trois fonctions de forçage sont Forçage, Forçage permanent et Débloquer les sorties en STOP.

Vous pouvez exécuter les fonctions en ligne suivantes avec la table de visualisation :

- Visualisation de l'état des variables
- Forçage des valeurs de variables individuelles

Vous pouvez choisir quand vous voulez visualiser ou forcer la variable :

- Au début du cycle : Lecture ou écriture de la valeur au début du cycle
- En fin de cycle : Lecture ou écriture de la valeur à la fin du cycle
- Au passage à l'arrêt



Pour créer une table de visualisation :

1. Double-cliquez sur "Ajouter nouvelle table de visualisation" pour ouvrir une nouvelle table de visualisation.
2. Entrez un nom de variable pour ajouter une variable à la table de visualisation.

Les options suivantes sont disponibles pour la visualisation des variables :

- Visualiser tout : Cette commande lance la visualisation des variables visibles dans la table de visualisation active.
- Visualiser immédiatement : Cette commande lance la visualisation des variables visibles dans la table de visualisation active. La table de visualisation visualise les variables immédiatement et une seule fois.

Les options suivantes sont disponibles pour le forçage des variables :

- "Forcer à 0" met la valeur d'une adresse sélectionnée à "0".
- "Forcer à 1" met la valeur d'une adresse sélectionnée à "1".
- "Forçage immédiat" modifie immédiatement la valeur des adresses sélectionnées pour un cycle.
- "Forcer avec déclenchement" modifie les valeurs des adresses sélectionnées.

Cette fonction ne fournit pas de retour pour indiquer que les adresses sélectionnées ont été effectivement forcées. Si vous avez besoin d'un retour, utilisez la fonction "Forçage immédiat".

- "Débloquer sorties de périphérie" désactive le verrouillage des sorties et est disponible uniquement lorsque la CPU est à l'état ARRET.

Pour visualiser les variables, une liaison en ligne doit être établie à la CPU.

	Nom	Adresse	Format d'affich...	Valeur visualisation	Visualisation avec décl...	Forçage avec décl...	Valeur de forçage
1	"Start"	%I0.0	Bool		Permanent	Permanent	<input type="checkbox"/>
2	"Stop"	%I0.1	Bool		Permanent	Permanent	<input type="checkbox"/>
3	"Running"	%M0.0	Bool		Permanent	Permanent	<input type="checkbox"/>

Vous sélectionnez les diverses fonctions à l'aide des boutons en haut de la table de visualisation.

Entrez le nom de la variable à visualiser et sélectionnez un format d'affichage dans la liste de sélection déroulante. Lorsqu'une liaison en ligne est établie avec la CPU, cliquez sur le bouton "Visualiser" pour afficher la valeur en cours de la donnée dans le champ "Valeur de visualisation".

15.13.4.1 Utilisation d'un déclenchement lors de la visualisation ou du forçage de variables API

Le déclenchement détermine à quel point du cycle l'adresse sélectionnée doit être visualisée ou forcée.

Tableau 15- 4 Types de déclenchement

Déclenchement	Description
Permanent	Collecte les données en continu.
Début du cycle	Permanent : Collecte les données en continu au début du cycle après que la CPU a lu les entrées.
	Unique : Collecte les données une fois au début du cycle après que la CPU a lu les entrées.
Fin du cycle	Permanent : Collecte les données en continu à la fin du cycle avant que la CPU n'écrive dans les sorties.
	Unique : Collecte les données une fois à la fin du cycle avant que la CPU n'écrive dans les sorties.
Commutation en STOP	Permanent : Collecte les données en continu lorsque la CPU passe à l'état ARRET.
	Unique : Collecte les données une fois après que la CPU est passée à l'état ARRET.

Pour forcer une variable API lors d'un événement de déclenchement donné, sélectionnez soit le début soit la fin du cycle.

- Forçage d'une sortie : Le meilleur événement de déclenchement pour forcer une sortie est la fin du cycle, juste avant que la CPU n'écrive dans les sorties.

Visualisez la valeur des sorties au début du cycle pour déterminer quelle valeur est écrite dans les sorties physiques. Visualisez également les sorties avant que la CPU n'écrive les valeurs dans les sorties physiques pour contrôler la logique du programme et effectuer une comparaison avec le comportement réel des E/S.

- Forçage d'une entrée : Le meilleur événement de déclenchement pour forcer une entrée est le début du cycle, juste après que la CPU a lu les entrées et avant que le programme utilisateur n'utilise les valeurs d'entrée.

Si vous pensez que des valeurs changent pendant le cycle, vous voudrez peut-être visualiser la valeur des entrées à la fin du cycle pour vous assurer que la valeur de l'entrée à la fin du cycle n'a pas changé depuis le début du cycle. S'il y a une différence dans les valeurs, c'est peut-être que votre programme utilisateur écrit par erreur dans des entrées.

Pour diagnostiquer pourquoi la CPU est passée à l'état ARRET, utilisez le déclenchement "Commutation en STOP" pour capturer les dernières valeurs du processus.

15.13.4.2 Déblocage des sorties à l'état ARRET

La table de visualisation vous permet d'écrire dans les sorties lorsque la CPU est à l'état ARRET. Cette fonctionnalité vous permet de contrôler le câblage des sorties et de vérifier que le câble connecté à une broche de sortie déclenche un signal haut ou bas vers la borne de l'appareil de processus auquel il est connecté.

ATTENTION

Risques liés à l'écriture dans des sorties physiques à l'état ARRET

Même si la CPU est à l'état ARRET, débloquent une sortie physique peut activer le point du processus auquel elle est connectée, entraînant éventuellement un fonctionnement imprévisible de l'équipement. Or, un fonctionnement inattendu de l'équipement peut entraîner la mort ou des blessures graves.

Avant d'écrire dans une sortie à partir de la table de visualisation, assurez-vous que la modification de la sortie physique ne peut pas provoquer un fonctionnement imprévisible de l'équipement. Respectez toujours les mesures de sécurité nécessaires pour votre équipement.

Vous pouvez changer l'état des sorties à l'état ARRET lorsque les sorties sont débloquées. Si les sorties sont verrouillées, vous ne pouvez pas les modifier à l'état ARRET. Procédez comme suit pour permettre la passage des sorties à l'état ARRET à partir de la table de visualisation :

1. Sélectionnez la commande "Mode avancé" dans le menu "En ligne".
2. Sélectionnez l'option "Débloquer les sorties de périphérie" dans la commande "Forcer" du menu "En ligne" ou bien dans le menu contextuel qui s'affiche lorsque vous cliquez avec le bouton droit de la souris sur la ligne de la table de visualisation.

Vous ne pouvez pas activer les sorties à l'état ARRET si vous avez configuré une périphérie décentralisée. Une erreur est renvoyée si vous tentez de le faire.

Le passage de la CPU à l'état MARCHE désactive l'option "Débloquer sorties de périphérie".

Si des entrées ou sorties sont forcées de manière permanente, la CPU n'est pas autorisée à débloquer des sorties alors qu'elle est à l'état ARRET. Le forçage permanent doit d'abord être annulé.

15.13.5 Forçage permanent de valeurs dans la CPU

15.13.5.1 Utilisation de la table de forçage permanent

Une table de forçage permanent fournit une fonction de "forçage permanent" qui écrase la valeur d'une entrée ou d'une sortie avec une valeur spécifiée pour l'adresse de périphérie d'entrée ou de sortie. La CPU applique cette valeur forcée en permanence à la mémoire image des entrées avant l'exécution du programme utilisateur et à la mémoire image des sorties avant l'écriture des sorties dans les modules.

Remarque

Les valeurs de forçage permanent sont stockées dans la CPU et non dans la table de forçage permanent.

Vous ne pouvez pas forcer une entrée (adresse I) ou une sortie (adresse Q), mais vous pouvez forcer une entrée ou une sortie de périphérie. La table de forçage permanent ajoute automatiquement ":P" à l'adresse (par exemple, "On":P ou "Run":P).



	i	Nom	Adresse	Format d'affichage	Valeur visualisation	Valeur de forçage permanent	F
1		"On":P	%I0.0:P	Bool		TRUE	<input checked="" type="checkbox"/>
2		"Off":P	%I0.1:P	Bool			<input type="checkbox"/>
3		"Run":P	%Q0.1:P	Bool			<input type="checkbox"/>

Entrez la valeur pour l'entrée ou la sortie à forcer dans la cellule "Valeur de forçage permanent". Vous pouvez alors utiliser la case à cocher dans la colonne "Forçage permanent" pour activer le forçage permanent de l'entrée ou de la sortie.



Utilisez le bouton "Démarrer ou remplacer forçage permanent" pour forcer la valeur des variables de manière permanente dans la table de forçage permanent. Cliquez sur le bouton "Arrêter forçage permanent" pour arrêter le forçage des variables.

Dans la table de forçage permanent, vous pouvez visualiser l'état de la valeur forcée pour une entrée, mais vous ne pouvez pas visualiser la valeur forcée d'une sortie.

Vous pouvez également visualiser l'état des variables forcées dans l'éditeur de programmes.



Remarque

Lors du forçage permanent d'une entrée ou d'une sortie dans une table de forçage permanent, les actions de forçage permanent deviennent partie intégrante de la configuration du projet. Si vous fermez STEP 7, les éléments forcés de manière permanente restent actifs dans le programme de la CPU, et ce jusqu'à ce qu'ils soient effacés. Pour annuler le forçage permanent de ces éléments, vous devez utiliser STEP 7 pour vous connecter à la CPU en ligne, puis utiliser la table de forçage permanent pour désactiver ou arrêter la fonction de forçage permanent pour ces éléments.

15.13.5.2 Fonctionnement de la fonction de forçage permanent

Vous pouvez effectuer un forçage permanent d'entrées et de sorties en indiquant l'adresse d'entrée ou de sortie physique (I_:P ou Q_:P) dans la table de forçage permanent puis en lançant la fonction de forçage permanent.

Dans le programme, les entrées physiques lues sont remplacées par les valeurs de forçage permanent. Le programme utilise les valeurs de forçage permanent pendant le traitement. Lorsque le programme écrit dans une sortie physique, la valeur de sortie est remplacée par la valeur de forçage permanent. La valeur de forçage permanent apparaît au niveau de la sortie physique et est utilisée par le processus.

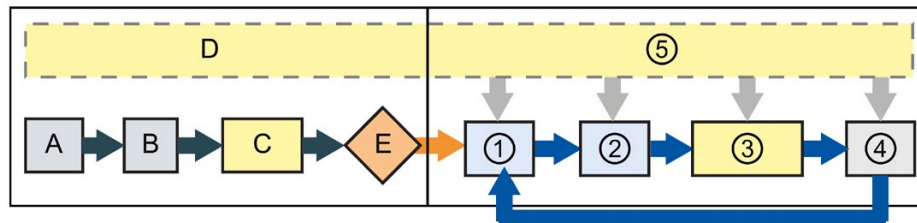
Lors du forçage permanent d'une entrée ou d'une sortie dans la table de forçage permanent, les actions de forçage deviennent partie intégrante du programme utilisateur. Même si vous fermez le logiciel de programmation, les sélections de forçage permanent restent actives dans le programme CPU qui fonctionne, jusqu'à ce que vous les annuliez en commutant en ligne avec le logiciel de programmation et en arrêtant la fonction de forçage permanent. Les programmes à entrées ou sorties forcées qui sont chargés dans une autre CPU à partir d'une carte mémoire continuent à appliquer le forçage permanent aux entrées et sorties sélectionnées dans le programme.

Si la CPU exécute le programme utilisateur à partir d'une carte mémoire protégée en écriture, vous ne pouvez pas lancer ou modifier le forçage permanent d'E/S à partir d'une table de visualisation, car il n'est pas possible d'écraser les valeurs dans le programme utilisateur protégé en écriture. Toute tentative de forçage permanent de valeurs protégées en écriture génère une erreur. Si vous utilisez une carte mémoire pour transférer un programme utilisateur, tout élément forcé en permanence sur cette carte mémoire sera transféré dans la CPU.

Remarque

Les E/S TOR affectées à HSC, PWM et PTO ne peuvent pas être forcées.

Les E/S TOR utilisées par le compteur rapide (HSC), la modulation de largeur d'impulsion (PWM) et la sortie de trains d'impulsions (PTO) sont affectées pendant la configuration des appareils. Lorsque des adresses d'E/S TOR sont affectées à ces appareils, les valeurs dans les adresses d'E/S affectées ne peuvent pas être forcées par la fonction de forçage permanent de la table de forçage permanent.



Mise en route

- A L'effacement de la zone de mémoire I n'est pas affecté par la fonction Forçage permanent.
- B L'initialisation des valeurs de sortie n'est pas affectée par la fonction Forçage permanent.
- C Pendant l'exécution des OB de démarrage, la CPU applique la valeur de forçage permanent lorsque le programme utilisateur accède à l'entrée physique.
- D L'enregistrement d'événements d'alarme dans la file d'attente n'est pas affecté.
- E La validation de l'écriture dans les sorties n'est pas affectée.

MARCHE

- ① Pendant l'écriture de la mémoire Q dans les sorties physiques, la CPU applique la valeur de forçage permanent lors de la mise à jour des sorties.
- ② Lors de la lecture des entrées physiques, la CPU applique les valeurs de forçage permanent juste avant de copier les entrées en mémoire I.
- ③ Pendant l'exécution du programme utilisateur (OB de cycle de programme), la CPU applique la valeur de forçage permanent lorsque le programme utilisateur accède à l'entrée physique ou écrit dans la sortie physique.
- ④ Le traitement des demandes de communications et le test d'auto-diagnostic ne sont pas affectés par la fonction Forçage permanent.
- ⑤ Le traitement des alarmes à n'importe quel moment du cycle n'est pas affecté.

15.14 Chargement dans la CPU à l'état MARCHE

La CPU prend en charge la fonction "Chargement dans la CPU à l'état MARCHE". Cette fonction doit vous permettre d'apporter de petites modifications au programme utilisateur en entraînant le moins de gêne possible pour le processus commandé par le programme. La réalisation de cette fonction permet toutefois des modifications importantes du programme, pouvant provoquer des perturbations ou même s'avérer dangereuses.

ATTENTION

Risques liés au chargement dans la CPU à l'état MARCHE

Lorsque vous chargez des modifications dans la CPU à l'état MARCHE, ces modifications affectent immédiatement le fonctionnement du processus. La modification du programme à l'état MARCHE peut entraîner un fonctionnement inattendu du système, pouvant causer la mort, des blessures graves ou des dégâts matériels importants.

Aussi, seules des personnes autorisées, conscientes des conséquences des modifications à l'état MARCHE sur le fonctionnement du système, sont-elles habilitées à effectuer un chargement à l'état MARCHE.

La fonction "Chargement à l'état MARCHE" permet de modifier un programme et de charger les modifications dans votre CPU sans faire passer cette dernière à l'état ARRET :

- Vous pouvez apporter des modifications mineures au processus en cours sans avoir à l'arrêter (par exemple, modifier la valeur d'un paramètre).
- Vous pouvez déboguer un programme plus rapidement grâce à cette fonction (par exemple, inverser la logique d'un contact normalement ouvert ou fermé).

Vous pouvez apporter les modifications suivantes aux variables et blocs de programme et les charger dans la CPU à l'état MARCHE :

- Créer, remplacer et effacer des fonctions (FC), blocs fonctionnels (FB) et tables de variables.
- Créer, effacer et remplacer des blocs de données (DB) et des blocs de données d'instance pour blocs fonctionnels (FB). Vous pouvez procéder à des ajouts dans des structures de DB et les charger dans la CPU à l'état MARCHE. Selon vos paramètres de configuration (Page 1412), la CPU peut conserver les valeurs des variables de DB existantes et initialiser les nouvelles variables du DB à leur valeur initiale ou bien elle peut réinitialiser toutes les variables du DB à leur valeur initiale. Vous ne pouvez pas charger un DB du serveur Web (commande ou fragment) à l'état MARCHE.
- Remplacer des blocs d'organisation (OB). Vous ne pouvez toutefois ni créer d'OB, ni en supprimer.

Vous pouvez charger simultanément jusqu'à vingt blocs à l'état MARCHE. Si vous devez charger plus de vingt blocs, vous devez faire passer la CPU à l'état ARRET.

Si vous chargez des modifications dans un processus réel (et non dans un processus simulé comme cela pourrait être le cas au cours du débogage d'un programme), il est vital de considérer toutes les conséquences en termes de sécurité sur les machines et les opérateurs des machines avant le chargement.

Remarque

Si la CPU est à l'état MARCHE et que des modifications ont été apportées au programme, STEP 7 tentera toujours, dans un premier temps, d'effectuer un chargement dans la CPU à l'état MARCHE. Si ce n'est pas ce que vous désirez, vous devez faire passer la CPU à l'état ARRÊT.

Si les modifications effectuées ne sont pas acceptées en chargement à l'état MARCHE, STEP 7 demandera à l'utilisateur de mettre la CPU à l'état ARRÊT.

15.14.1 Conditions requises pour un chargement dans la CPU à l'état MARCHE

Les conditions suivantes doivent être satisfaites pour que vous puissiez charger des modifications de votre programme dans une CPU à l'état MARCHE :

- Votre CPU doit être de version V3.0 ou plus.

Remarque

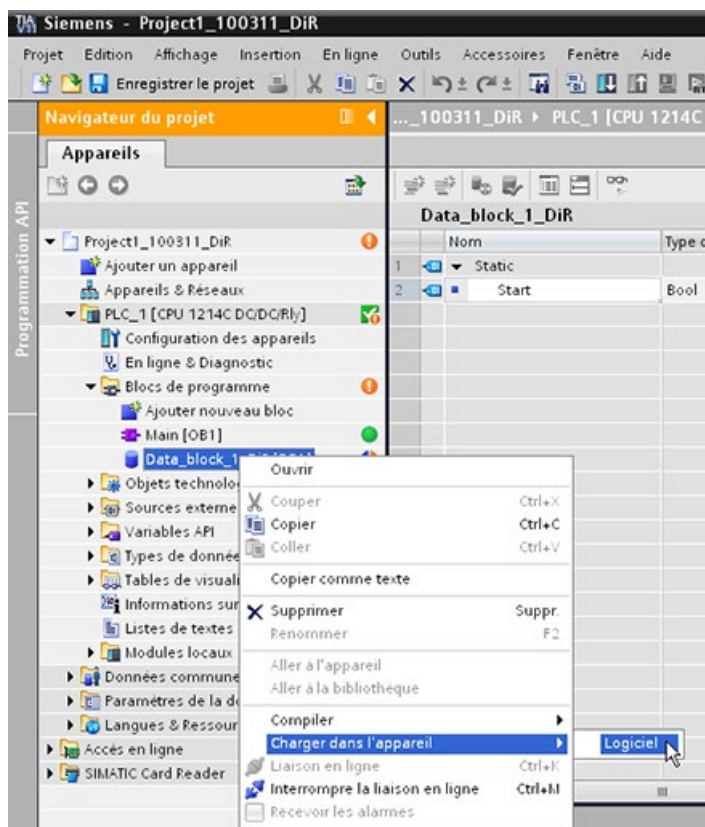
Votre CPU doit être de version V4.0 ou plus pour que vous puissiez modifier l'interface de bloc étendue à l'état MARCHE (Page 1412).

- La compilation de votre programme ne doit pas présenter d'erreurs.
- Vous devez avoir établi une communication entre la console de programmation où s'exécute STEP 7 et la CPU.

15.14.2 Modification du programme à l'état "Marche"

Pour modifier le programme en état MARCHE, vous devez d'abord vous assurer que la CPU et le programme remplissent les prérequis (Page 1407), puis procédez comme suit :

1. Sélectionnez l'une des méthodes suivantes pour charger votre programme dans la CPU à l'état MARCHE :
 - Sélectionnez la commande "Charger dans l'appareil" du menu "En ligne".
 - Cliquez sur le bouton "Charger dans l'appareil" de la barre d'outils.
 - Clic droit sur "Blocs de programme" dans le navigateur du projet et sélection de la commande "Charger dans l'appareil > Logiciel"



Si la compilation du programme s'exécute sans erreur, STEP 7 commence le chargement du programme dans la CPU.

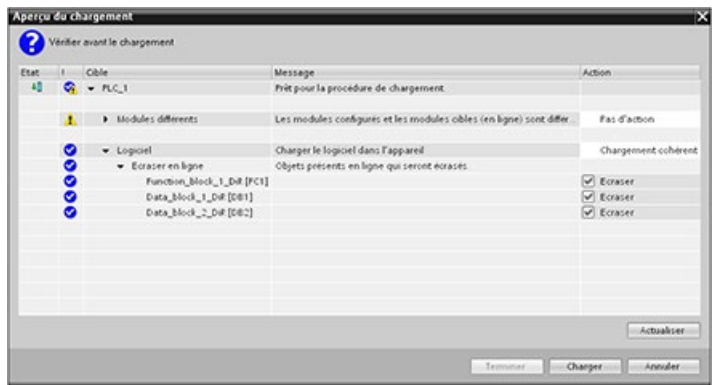
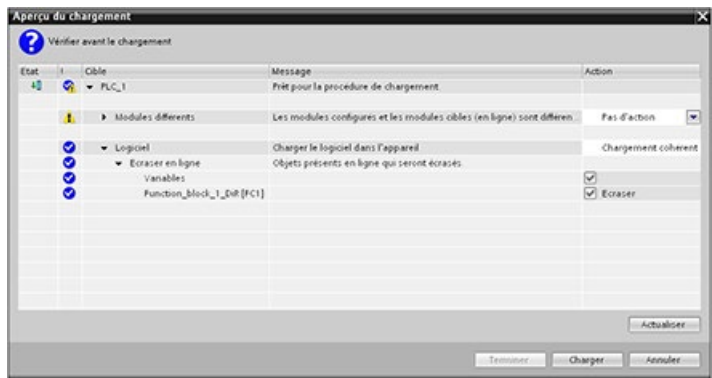
2. Lorsque STEP 7 vous demande de charger votre programme ou d'annuler l'opération, cliquez sur "Charger" pour charger le programme dans la CPU.

15.14.3 Chargement de blocs sélectionnés

Dans le dossier Blocs de programme, vous pouvez sélectionner un bloc unique ou une sélection de blocs à télécharger.

Si vous sélectionnez un bloc individuel à charger dans la CPU, la seule option disponible dans la colonne "Action" est "Chargement cohérent". Vous pouvez afficher le détail de la ligne de catégorie pour vous assurer des blocs à charger. Dans cet exemple, une petite modification a été apportée au bloc hors ligne et aucun autre bloc n'a besoin d'être chargé.

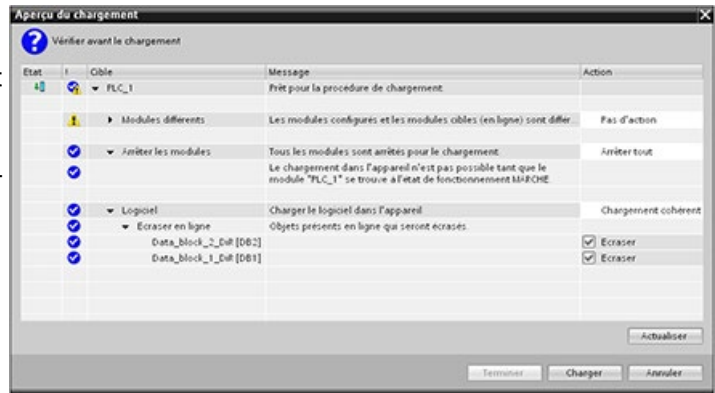
Dans cet exemple, il faut charger plus d'un bloc dans la CPU.



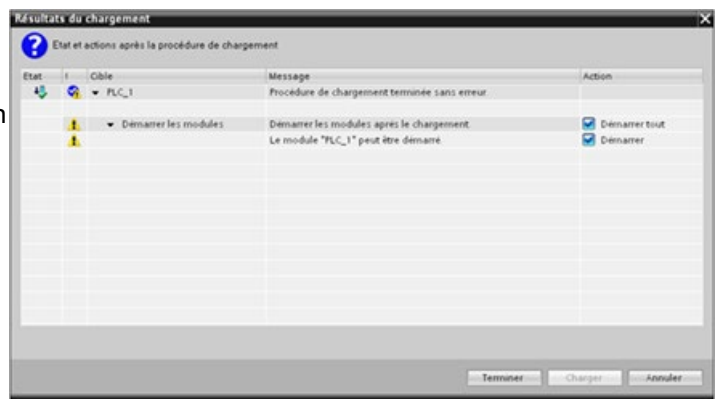
Remarque

Vous pouvez charger simultanément jusqu'à vingt blocs à l'état MARCHÉ. Si vous devez charger plus de vingt blocs, vous devez faire passer la CPU à l'état ARRÊT.

Si vous tentez un chargement à l'état MARCHE mais que le système constate que ce n'est pas possible avant le chargement en cours, la ligne de catégorie Arrêt modules apparaît dans la boîte de dialogue.

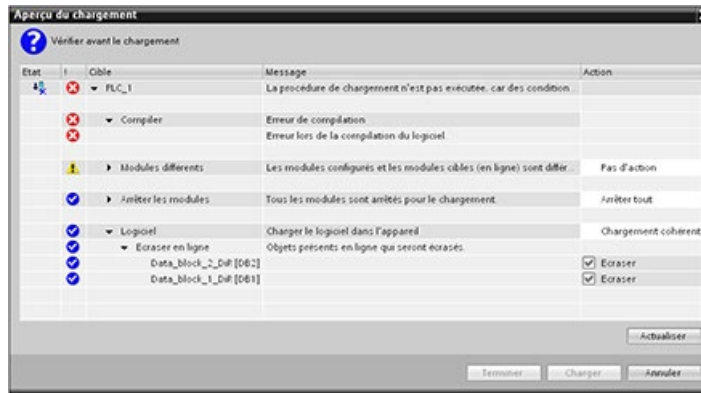


Cliquez sur le bouton "Charger" ; la boîte de dialogue "Résultats du chargement" s'affiche. Cliquez sur le bouton "Terminer" pour achever le chargement dans la CPU.



15.14.4 Chargement d'un bloc individuel sélectionné alors qu'un autre bloc présente une erreur de compilation

Si vous tentez un chargement cohérent alors qu'une erreur de compilation existe dans un autre bloc, la boîte de dialogue signale une erreur et le bouton de chargement est désactivé.



Vous devez corriger l'erreur de compilation dans l'autre bloc. Le bouton "Charger" devient alors actif.

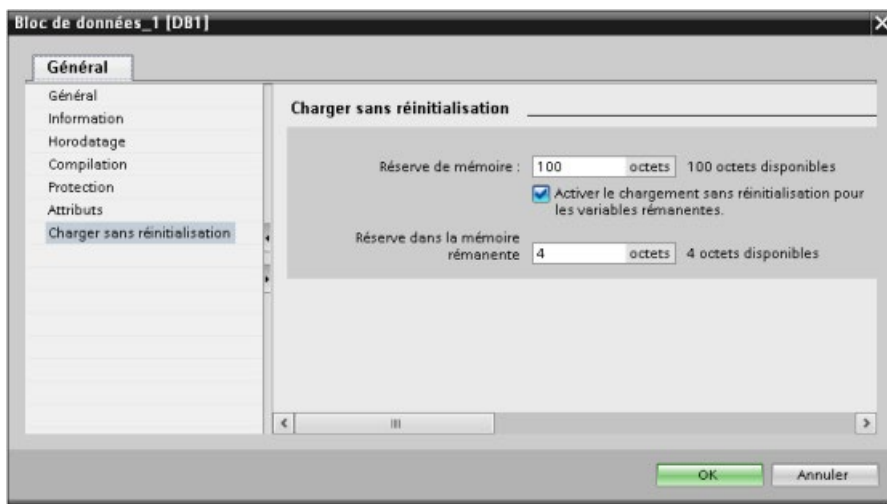


15.14.5 Forcer et télécharger des blocs existants à l'état MARCHE

La fonction de chargement à l'état Marche vous permet d'ajouter et de forcer des variables dans des blocs de données et des blocs fonctionnels et de charger les blocs modifiés dans la CPU à l'état MARCHE.


Chargement sans réinitialisation

Chaque DB et FB possède une réserve de mémoire que vous pouvez utiliser pour ajouter des variables au bloc avant de le charger par la suite à l'état MARCHE. Par défaut, la taille initiale de la réserve de mémoire est de 100 octets. Vous pouvez ajouter des variables supplémentaires à votre bloc jusqu'à concurrence de la taille de la réserve de mémoire et charger le bloc étendu dans la CPU à l'état MARCHE. Vous pouvez également augmenter la réserve de mémoire s'il vous faut plus d'espace mémoire pour les variables supplémentaires dans votre bloc. Si le nombre de variables ajoutées dépasse la taille de mémoire allouée, vous ne pouvez pas charger le bloc étendu dans la CPU à l'état MARCHE.




La fonction de "Chargement sans réinitialisation" vous permet d'étendre un bloc de données en ajoutant des variables de bloc de données et de charger le bloc étendu à l'état MARCHE. Ainsi, vous pouvez ajouter des variables à un bloc de données et le charger sans réinitialiser votre programme. La CPU conserve les valeurs de variables du bloc de données existantes et initialise les variables venant d'être ajoutées avec les valeurs de départ définies.

Procédez comme suit pour activer cette fonction pour un projet en ligne avec la CPU à l'état MARCHE :

1. Depuis le dossier Blocs de programme dans l'arborescence du projet STEP 7, ouvrez le bloc.
2. Cliquez sur le bouton à bascule "Charger sans réinitialisation" dans l'éditeur de bloc pour activer la fonction. (l'icône comporte un cadre lorsque la fonction est activée : )
3. Cliquez sur OK dans l'invite pour valider votre choix.
4. Ajoutez des variables dans l'interface de bloc et chargez le bloc à l'état MARCHE. Vous pouvez ajouter et charger autant de nouvelles variables que la taille de votre réserve de mémoire le permet.

Si vous avez ajouté plus d'octets à votre bloc que configurés dans la réserve de mémoire, STEP 7 affiche une erreur lorsque vous tentez de charger le bloc à l'état MARCHE. Vous devez éditer les propriétés du bloc pour augmenter la taille. Vous ne pouvez pas supprimer des entrées existantes ou modifier la "Réserve de mémoire" du bloc tant que la fonction "Chargement sans réinitialisation" est activée. Procédez comme suit pour désactiver la fonction "Chargement sans réinitialisation" :

1. Cliquez sur le bouton à bascule "Charger sans réinitialisation" dans l'éditeur de bloc pour désactiver la fonction. (l'icône ne comporte pas de cadre lorsque la fonction est désactivée : )
2. Cliquez sur OK dans l'invite pour valider votre choix.
3. Chargez le bloc. Dans la boîte de dialogue du chargement, vous devez sélectionner "réinitialiser" afin de charger le bloc étendu.

Le chargement réinitialise alors toutes les variables de bloc, existantes comme nouvelles, aux valeurs de départ définies.

Charger des variables de bloc rémanentes

Le chargement de variables de bloc rémanentes à l'état MARCHE nécessite l'affectation d'une réserve dans la mémoire rémanente. Procédez comme suit pour configurer cette réserve dans la mémoire rémanente :

1. Depuis le dossier Blocs de programme dans l'arborescence du projet STEP 7, cliquez avec le bouton droit de la souris sur le bloc et sélectionnez "Propriétés" dans le menu contextuel.
2. Sélectionnez la propriété "Chargement sans réinitialisation".
3. Cochez la case "Activer le chargement sans réinitialisation pour les variables rémanentes".
4. Configurez le nombre d'octets disponibles pour la réserve dans la mémoire rémanente.
5. Cliquez sur OK pour sauvegarder vos modifications.
6. Ajoutez des variables de bloc de données rémanentes dans le bloc de données et chargez-le à l'état MARCHE. Vous pouvez ajouter et charger autant de nouvelles variables de bloc rémanentes que la taille de votre réserve dans la mémoire rémanente le permet.

Si vous avez ajouté plus d'octets rémanents à votre bloc de données que le nombre configuré dans la réserve de mémoire rémanente, STEP 7 affiche une erreur lorsque vous tentez de charger le bloc à l'état MARCHE. N'ajoutez des variables de bloc rémanentes que jusqu'à concurrence de la réserve dans la mémoire rémanente si vous voulez les charger à l'état MARCHE.

Lorsque vous téléchargez les variables de bloc étendu rémanentes, les variables contiennent leurs valeurs instantanées.

Réglage de la taille de la réserve de mémoire pour les nouveaux blocs

Chaque nouveau bloc de données contient par défaut une réserve de mémoire de 100 octets. Lorsque vous créez un nouveau bloc, il dispose d'une réserve de 100 octets. Si vous souhaitez une taille de réserve de mémoire différente pour les nouveaux blocs, vous pouvez modifier le réglage dans les paramètres de programmation API :

1. Dans STEP 7, sélectionnez la commande **Outils > Paramètres** .
2. Dans la boîte de dialogue Paramètres, affichez le détail de "Programmation API" et sélectionnez "Général".
3. Dans le groupe "Chargement sans réinitialisation", entrez le nombre souhaité d'octets pour la réserve de mémoire.

Lorsque vous créez de nouveaux blocs, STEP 7 utilise le réglage de la réserve de mémoire que vous avez indiqué pour les nouveaux blocs.

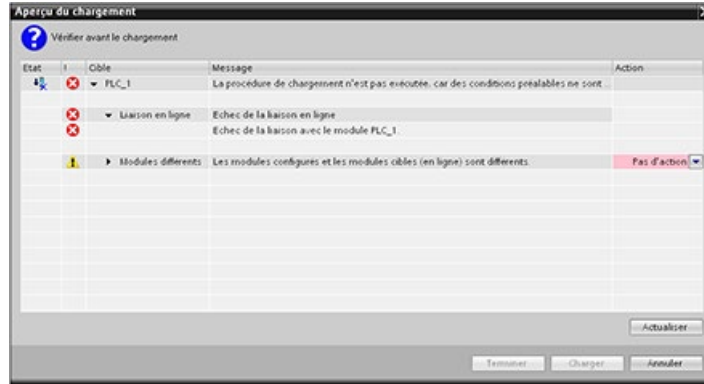
Restrictions

Les restrictions suivantes s'appliquent à l'édition et au chargement des blocs à l'état MARCHE :

- L'extension de l'interface de bloc en ajoutant de nouvelles variables, ainsi que le chargement à l'état MARCHE, n'est disponible que pour les blocs optimisés (Page 199).
- Vous ne pouvez pas modifier la structure d'un bloc ni charger le bloc modifié à l'état MARCHE sans réinitialisation. Ajouter de nouveaux éléments dans une variable Struct (Page 138), modifier des noms de variable, tailles de tableau, types de données ou un état rémanent : tout cela nécessite la réinitialisation du bloc si vous le chargez à l'état MARCHE. Les seules modifications que vous pouvez apporter à des variables de bloc existantes tout en pouvant charger le bloc à l'état MARCHE sans réinitialisation, sont les modifications apportées aux valeurs initiales (blocs de données), valeurs par défaut (blocs fonctionnels) ou aux commentaires.
- Vous ne pouvez pas charger plus de variables de bloc à l'état MARCHE que la réserve de mémoire ne peut prendre en compte.
- Vous ne pouvez pas charger plus de variables de bloc rémanentes à l'état MARCHE que la réserve dans la mémoire rémanente ne peut prendre en compte.

15.14.6 Réaction du système en cas d'échec de l'opération de chargement

En cas de défaillance de la liaison réseau pendant le chargement initial à l'état MARCHE, STEP 7 affiche la boîte de dialogue "Aperçu du chargement" suivante :



15.14.7 Considérations sur le chargement dans la CPU à l'état MARCHE

Avant de charger le programme dans la CPU à l'état MARCHE, tenez compte des conséquences d'une modification à l'état MARCHE sur le fonctionnement de la CPU dans les situations suivantes :

- Si vous avez effacé la logique de commande d'une sortie, la CPU maintient le dernier état de cette sortie jusqu'à la mise hors tension puis sous tension suivante ou jusqu'au passage suivant à l'état ARRET.
- Si vous avez effacé un compteur rapide ou des fonctions de sortie d'impulsions alors qu'ils s'exécutaient, le compteur rapide ou la sortie d'impulsions continue à s'exécuter jusqu'à la mise hors tension puis sous tension suivante ou jusqu'au passage suivant à l'état ARRET.
- Toute logique dépendant de l'état du memento "Premier cycle" n'est pas exécutée avant la mise hors tension puis sous tension suivante ou le passage suivant de l'état ARRET à l'état MARCHE. Le memento "Premier cycle" est mis à 1 uniquement par le passage à l'état MARCHE et n'est pas affecté par un chargement à l'état MARCHE.

- Les valeurs en cours des blocs de données et/ou des variables peuvent être écrasées.

Remarque

Pour que vous puissiez charger votre programme à l'état MARCHE, la CPU doit accepter les modifications à l'état MARCHE, le programme doit être compilé sans erreurs et la communication entre STEP 7 et la CPU doit fonctionner correctement.

Vous pouvez apporter les modifications suivantes aux variables et blocs de programme et les charger dans la CPU à l'état MARCHE :

- Créer, remplacer et effacer des fonctions (FC), blocs fonctionnels (FB) et tables de variables.
- Créer et effacer des blocs de données (DB). Toutefois, les modifications de structures de DB ne peuvent pas être écrasées. Les valeurs initiales des DB peuvent être écrasées. Vous ne pouvez pas charger un DB du serveur Web (commande ou fragment) à l'état MARCHE.
- Remplacer des blocs d'organisation (OB). Vous ne pouvez toutefois ni créer d'OB, ni en supprimer.

Vous pouvez charger simultanément jusqu'à vingt blocs à l'état MARCHE. Si vous devez charger plus de vingt blocs, vous devez faire passer la CPU à l'état ARRÊT.

Une fois qu'un chargement a été déclenché, vous ne pouvez pas exécuter d'autres tâches dans STEP 7 tant que ce chargement n'est pas achevé.

Instructions pouvant échouer en raison d'un chargement à l'état MARCHE

Les instructions suivantes peuvent présenter une erreur temporaire lorsque des modifications chargées à l'état MARCHE sont activées dans la CPU. L'erreur se produit lorsque l'instruction est déclenchée alors que la CPU se prépare à activer les modifications chargées. Pendant ce temps, la CPU suspend toute demande d'accès du programme utilisateur à la mémoire de chargement alors qu'elle achève l'accès en cours du programme utilisateur à la mémoire de chargement. Ceci afin que les modifications chargées puissent être activées de manière cohérente.

Instruction	Réponse pendant que l'activation est en attente
DataLogCreate	STATUS = W#16#80C0, ERROR = TRUE
DataLogOpen	STATUS = W#16#80C0, ERROR = TRUE
DataLogWrite	STATUS = W#16#80C0, ERROR = TRUE
DataLogClose	STATUS = W#16#80C0, ERROR = TRUE
DataLogNewFile	STATUS = W#16#80C0, ERROR = TRUE
DataLogClear	STATUS = W#16#80C0, ERROR = TRUE
DataLogDelete	STATUS = W#16#80C0, ERROR = TRUE
READ_DBL	RET_VAL = W#16#82C0
WRIT_DBL	RET_VAL = W#16#82C0
Create_DB	RET_VAL = W#16#80C0
Delete_DB	RET_VAL = W#16#80C0
RTM	RET_VAL = 0x80C0

Dans tous les cas, la sortie RLO de l'instruction a la valeur FAUX lorsque l'erreur se produit. Cette erreur est temporaire. Si elle se produit, il faut retenter l'instruction plus tard.

Remarque

Il ne faut pas faire de nouvelle tentative pendant l'exécution en cours de l'OB.

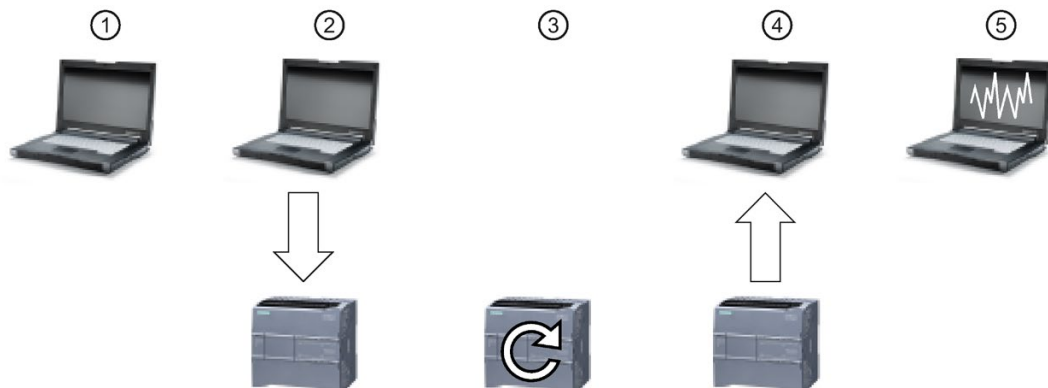
15.15 Traçage et enregistrement de données CPU en fonctions de conditions de déclenchement

STEP 7 fournit des fonctions de traçage et d'analyse logique qui vous permettent de configurer des variables que l'API doit tracer et enregistrer. Vous pouvez alors charger les données de mesure de trace enregistrées dans votre console de programmation et utiliser les outils STEP 7 pour analyser, gérer et représenter graphiquement vos données. Vous utilisez le dossier Traces dans l'arborescence de projet STEP 7 pour créer et gérer les traces.

Remarque

Les données de mesure de trace sont disponibles uniquement dans le projet STEP 7 et ne sont pas disponibles pour un traitement par d'autres outils.

La figure suivante montre les différentes étapes de la fonction de traçage :



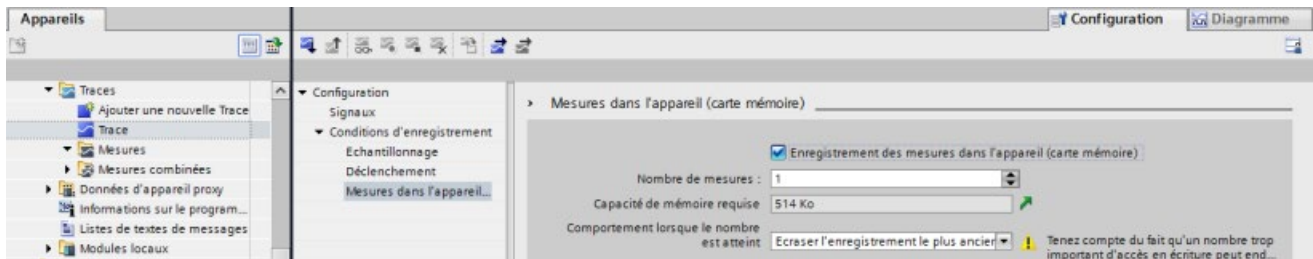
- ① Configurez la trace dans l'éditeur de trace de STEP 7. Vous pouvez configurer les options suivantes :
 - Valeurs de données à enregistrer
 - Durée de l'enregistrement
 - Fréquence d'enregistrement
 - Condition de déclenchement
- ② Transférez la configuration de la trace de STEP 7 dans l'API.
- ③ L'API exécute le programme et, lorsque la condition de déclenchement se produit, il déclenche l'enregistrement des données de trace.
- ④ Transférez les valeurs enregistrées de l'API dans STEP 7.
- ⑤ Utilisez les outils de STEP 7 pour analyser, afficher graphiquement et enregistrer les données.

Le S7-1200 prend en charge deux tâches de trace avec au maximum 16 variables acquises par événement de déclenchement. Chaque tâche trace fournit 524 288 octets de mémoire vive pour l'enregistrement des valeurs de trace et la charge système associée, par exemple les adresses de variables et les horodatages.

Enregistrement des mesures de trace dans la carte mémoire

La CPU S7-1200 peut uniquement enregistrer des mesures de trace dans la carte mémoire SIMATIC. En l'absence de carte mémoire dans votre CPU, la CPU consignera une entrée dans le tampon de diagnostic si le programme tente d'enregistrer des mesures de trace. La CPU limite l'espace assigné aux mesures de trace de sorte qu'1 Mo de mémoire de chargement externe soit toujours disponible. Si une mesure de trace devait requérir plus de mémoire qu'il n'est autorisé au maximum, la CPU n'enregistrera pas la mesure et archivera une entrée de tampon de diagnostic.

En outre, si vous sélectionnez "Écraser l'enregistrement le plus ancien" dans STEP 7, l'écriture en continu peut réduire la durée de vie de la mémoire de chargement. Lorsque vous sélectionnez "Écraser l'enregistrement le plus ancien", la CPU remplace l'enregistrement le plus ancien par le plus récent lorsqu'elle a atteint le nombre configuré de mesures de trace, puis elle poursuit la trace et l'enregistrement des mesures. Écraser les mesures les plus anciennes permet de capturer les problèmes intermittents.



La CPU prend en charge un maximum de 999 résultats de mesure de trace. Pendant qu'elle est en train d'enregistrer les mesures de trace en mémoire de chargement externe, la CPU ne vérifie pas la condition de déclenchement pour la tâche trace. Une fois qu'elle a fini d'enregistrer les mesures, elle vérifie à nouveau les conditions de déclenchement.

Accès aux exemples

Consultez le système d'information de STEP 7 pour plus de détails sur la manière de programmer une trace, de charger la configuration dans la CPU, de charger les données de traçage depuis la CPU et d'afficher les données dans l'analyseur logique. Vous trouverez ici des exemples détaillées dans le chapitre "Utilisation de fonctions en ligne et de diagnostic > Utilisation de fonctions de traçage et d'analyse logique".

De plus, le manuel en ligne "Industry Automation SINAMICS/SIMATIC, Utilisation de la fonction Trace et analyseur logique"

(<https://support.industry.siemens.com/cs/ww/fr/view/64897128>) constitue une bonne référence.

15.16 Détermination du type de condition de rupture de fil sur un module SM 1231

Comme décrit à la rubrique Plages de mesures d'entrées analogiques pour tension et courant (SB et SM) (Page 1554), le module SM 1231 renvoie une valeur d'entrée analogique de 32767 (16#7FFF) pour les situations de rupture de fil et de débordement. Si vous voulez déterminer laquelle de ces deux erreurs s'est produite, vous pouvez inclure la logique correspondante dans votre programme STEP 7. La méthode pour déterminer le type d'erreur comprend les tâches suivantes :

- Créer un OB d'alarme de diagnostic qui sera appelé pour chaque événement de diagnostic apparaissant ou disparaissant
- Inclure un appel de l'instruction RALRM
- Définir pour le paramètre AINFO un tableau d'octets qui contient les informations sur le type d'erreur
- Évaluer les octets 32 et 33 de la structure AINFO dans le DB RALRM_DB lorsque la CPU déclenche l'OB d'alarme de diagnostic

Création d'un OB d'alarme de diagnostic

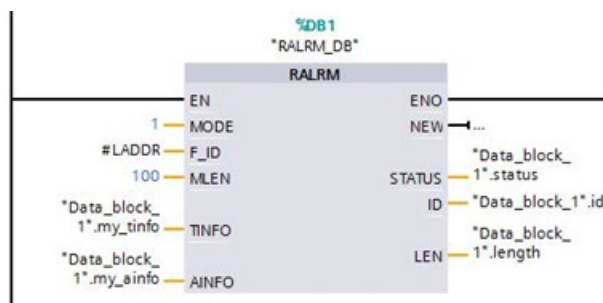
Créez un OB d'alarme de diagnostic pour pouvoir déterminer le moment où une rupture de fil s'est produite. La CPU appellera cet OB pour tout événement de diagnostic apparaissant ou disparaissant.

Lorsque la CPU appellera l'OB d'alarme de diagnostic, le paramètre d'entrée LADDR contiendra l'identification matérielle du module présentant l'erreur. Vous trouverez l'identification matérielle du module SM 1231 dans la configuration d'appareil de STEP 7 pour le module SM 1231.

Appel de l'instruction RALRM

Procédez comme suit pour programmer l'appel de l'instruction RALRM :

1. Ajoutez un appel de RALRM dans votre programme STEP 7.
2. Donnez au paramètre d'entrée F_ID la valeur de l'identification matérielle figurant dans le paramètre LADDR de l'OB d'alarme de diagnostic.
3. Utilisez un tableau d'octets pour les paramètres d'entrée TINFO et AINFO. La taille du tableau doit être de 34 octets ou plus.



















Interprétation de AINFO après la survenue d'une alarme de diagnostic

Après l'exécution de l'OB d'alarme de diagnostic, le tableau d'octets AINFO contient les données sur le diagnostic du module.

Les octets 0 à 25 constituent l'information d'en-tête. Les octets relatifs au diagnostic du module sont définis comme suit :

Octet	Description	
26 et 27	Valeur de mot 16#8000. Indique que le diagnostic est un diagnostic de type PROFINET.	
28 et 29	Mot contenant le numéro de la voie responsable de ce diagnostic	
30	Motif binaire aaabb000 qui indique le type de voie (aaa) et le type d'erreur (bb)	
	aaa	bb
	000 : réservé	00 : réservé
	001 : voie d'entrée	01 : erreur apparaissante
	010 : voie de sortie	10 : erreur disparaissante
	011 : voie d'entrée/sortie	11 : erreur disparaissante, autres erreurs présentes
31	Indication du format de données 0 : format de données libre 1 : bit 2 : deux bits 3 : quatre bits 4 : octet 5 : mot (deux octets) 6 : double mot (quatre octets) 7 : deux doubles mots (huit octets)	
32 et 33	Mot définissant le type d'erreur 16#0000 : réservé 16#0001 : court-circuit 16#0002 : sous-tension 16#0003 : surtension 16#0004 : surcharge 16#0005 : surchauffe 16#0006 : rupture de fil 16#0007 : limite haute dépassée 16#0008 : limite basse dépassée 16#0009 : erreur	

Considérons, par exemple, les octets 26 à 33 de cette structure AINFO :

29			my_ainfo[26]	Byte	16#0	16#80
30			my_ainfo[27]	Byte	16#0	16#00
31			my_ainfo[28]	Byte	16#0	16#00
32			my_ainfo[29]	Byte	16#0	16#00
33			my_ainfo[30]	Byte	16#0	16#28
34			my_ainfo[31]	Byte	16#0	16#05
35			my_ainfo[32]	Byte	16#0	16#00
36			my_ainfo[33]	Byte	16#0	16#07

- Le mot dans les octets 26 et 27 est 16#8000, ce qui indique qu'il s'agit d'un diagnostic de type PROFINET.
- Le mot dans les octets 28 et 29 indique qu'il s'agit d'un diagnostic pour la voie 0 du module.
- L'octet 30 a la valeur 16#28, ce qui correspond à 001 01 000 lorsqu'interprété comme motif binaire aaabb000. Cette valeur indique que ce diagnostic concerne une voie d'entrée et qu'il s'agit d'une erreur apparaissant.
- L'octet 31 a la valeur 5, ce qui indique une valeur de mot.
- La valeur de mot dans les octets 32 et 33 est 16#0007 signalant le dépassement de la limite haute.

L'acquisition des informations AINFO dans un événement d'alarme de diagnostic vous permet ainsi de déterminer la nature de l'événement de diagnostic.

15.17 Sauvegarde et restauration d'une CPU

15.17.1 Options de sauvegarde et de restauration

Au cours du temps, vous apporterez un certain nombre de modifications à votre système d'automatisation, par exemple en ajoutant de nouveaux appareils, en remplaçant des appareils existants ou en adaptant le programme utilisateur. Si ces modifications entraînent un comportement indésirable, vous pouvez restaurer l'état antérieur de l'installation d'automatisation si vous disposez d'une sauvegarde. STEP 7 et la CPU S7-1200 proposent plusieurs options de sauvegarde et de restauration de la configuration du matériel et du logiciel.

Options de sauvegarde

Le tableau suivant présente une vue d'ensemble des options de sauvegarde et de restauration des CPU S7 :

	Instantané des valeurs de visualisation	Chargement de l'appareil (logiciel)	Chargement de l'appareil comme nouvelle station (matériel et logiciel)	Chargement de la sauvegarde d'un appareil en ligne
Cas d'application	Restauration d'un état spécifique d'un bloc de données. Les valeurs actuelles des blocs de données sont reprises dans le projet avec leur horodatage.	Chargement dans le projet de blocs existant dans une CPU	Chargement de la configuration matérielle et du logiciel d'un appareil dans le projet	Création d'une sauvegarde complète d'une CPU sous forme de point de restauration. La copie de sauvegarde est cohérente, elle ne peut être ni ouverte, ni modifiée.
Condition	La CPU existe dans un projet. Les blocs de données doivent être identiques en ligne et hors ligne.	La CPU existe dans le projet.	L'appareil est disponible dans le catalogue du matériel de TIA Portal. Les fichiers HSP ou GSD éventuellement requis sont installés.	-
Possible à l'état	MARCHE, ARRÊT	MARCHE, ARRÊT	MARCHE, ARRÊT	ARRÊT
Possible pour les CPU F	Oui	Oui	Non	Oui
La sauvegarde peut être éditée	Oui	Oui	Oui	Non

Contenu de la sauvegarde

Le tableau ci-dessous indique quelles données peuvent être chargées et sauvegardées et par quel moyen :

	Instantané des valeurs de visualisation	Chargement de l'appareil (logiciel)	Chargement de l'appareil comme nouvelle station (matériel et logiciel)	Chargement de la sauvegarde d'un appareil en ligne
Valeurs actuelles des blocs de données	Instantané possible	Chargement possible	Chargement possible	Sauvegarde possible
Blocs logiciels	-	Chargement possible	Chargement possible	Sauvegarde possible
Variables API (noms de variables et constantes)	-	Chargement possible	Chargement possible	Sauvegarde possible
Objets technologiques	-	Chargement possible	Chargement possible	Sauvegarde possible
Configuration matérielle	-	-	Chargement possible	Sauvegarde possible
Tables de visualisation (serveur Web)	-	-	Chargement impossible	Sauvegarde possible
Données locales, mémentos, temporisations, compteurs et mémoire image	Instantané impossible	Chargement impossible	Chargement impossible	Sauvegarde possible
Archives et recettes (API)	-	-	-	Sauvegarde possible
Données générales sur la carte mémoire SIMATIC, par ex. aide pour les blocs de programme ou fichiers GSD	-	-	-	Sauvegarde possible

Considérations spécifiques pour la sauvegarde de valeurs actuelles

Le type de sauvegarde "Sauvegarde d'un appareil en ligne" sauvegarde les valeurs actuelles des variables qui sont définies comme rémanentes. Pour assurer la cohérence des données rémanentes, désactivez tous les accès en écriture aux données rémanentes pendant la sauvegarde.

Un passage de l'état ARRÊT à l'état MARCHÉ réinitialise les valeurs actuelles des données non rémanentes à leurs valeurs initiales. Une sauvegarde de la CPU contient uniquement les valeurs initiales des données non rémanentes.

15.17.2 Sauvegarde d'une CPU en ligne

Faire une sauvegarde de votre configuration peut être utile pour revenir à une configuration spécifique. Vous pourrez rétablir la configuration actuelle ultérieurement.

Conditions

Vous pouvez créer autant de sauvegardes que vous le souhaitez et disposez ainsi de configurations variées pour une CPU. Les conditions suivantes doivent être satisfaites pour la réalisation d'une sauvegarde :

- Vous avez déjà créé la CPU dans le projet STEP 7.
- Vous avez relié la CPU directement à la console de programmation/au PC au moyen de l'interface PROFINET de la CPU. Les opérations de sauvegarde et de restauration ne prennent pas en charge les interfaces PROFINET des modules de communication.
- La CPU est en ligne (en l'absence de liaison en ligne, la procédure de sauvegarde établit une liaison en ligne).
- La CPU est à l'état ARRÊT (si ce n'est pas le cas, la procédure de sauvegarde vous demande l'autorisation de faire passer la CPU à l'état ARRÊT).

Marche à suivre

Procédez comme suit pour créer une sauvegarde de la configuration actuelle d'une CPU :

1. Sélectionnez la CPU dans le navigateur du projet.
2. Sélectionnez la commande "Charger la sauvegarde de l'appareil en ligne" dans le menu "En ligne".

Si nécessaire, saisissez le mot de passe requis pour l'accès en lecture à la CPU et confirmez le passage de la CPU à l'état de fonctionnement ARRÊT.

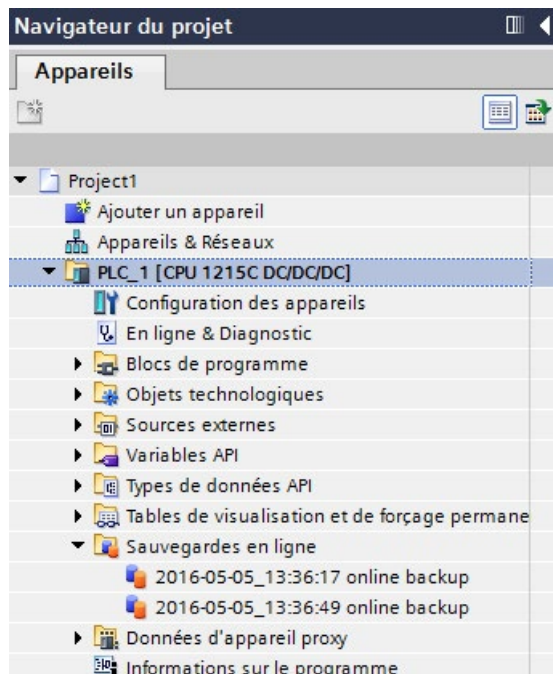
Résultat

Les sauvegardes sont désignées par le nom de la CPU, ainsi que par la date et l'heure de la sauvegarde. La sauvegarde comprend toutes les données qui sont nécessaires pour restaurer une configuration définie de la CPU. La CPU sauvegarde les données suivantes :

- le contenu de la carte mémoire en présence d'une carte mémoire
- les zones de mémoire rémanente des blocs de données, compteurs et mémentos
- d'autres contenus de mémoire rémanente, tels que les paramètres d'adresses IP

La sauvegarde contient les valeurs en cours de la CPU, mais n'inclut pas le tampon de diagnostic.

La sauvegarde se trouve dans le navigateur du projet, dans le dossier "Sauvegardes en ligne" sous la CPU. La figure suivante montre une CPU S7-1200 pour laquelle deux sauvegardes ont été créées :



Remarque

Notez que vous pouvez également sauvegarder la CPU en ligne dans l'outil SIMATIC Automation Tool (SAT) ou à partir de la page standard Sauvegarde en ligne du serveur Web (Page 1072).

Lorsque vous sauvegardez des fichiers à partir de STEP 7, STEP 7 enregistre ces fichiers à l'intérieur du projet STEP 7. Lorsque vous sauvegardez des fichiers à partir du serveur Web, votre PC ou votre appareil enregistre ces fichiers dans le dossier pris par défaut pour les téléchargements. Vous ne pouvez pas restaurer de fichiers de sauvegarde STEP 7 dans le serveur Web et vous ne pouvez pas restaurer de fichiers de sauvegarde Serveur Web dans STEP 7. Toutefois, vous pouvez enregistrer les fichiers de sauvegarde STEP 7 directement dans le dossier de téléchargement sur votre PC ou votre appareil. Dans ce cas, vous pourrez restaurer ces fichiers dans le serveur Web.

Enregistrement des fichiers de sauvegarde dans votre PC ou votre appareil

Procédez comme suit pour enregistrer un fichier de sauvegarde dans votre PC ou votre appareil :

1. Cliquez avec le bouton droit de la souris sur un fichier du dossier Sauvegardes en ligne dans le navigateur du projet.
2. Sélectionnez "Enregistrer sous" dans le menu contextuel.
3. Naviguez jusqu'au dossier où vous voulez enregistrer le fichier, par exemple le dossier pris par défaut pour les téléchargements sur votre PC ou votre appareil.
4. Cliquez sur Enregistrer.

15.17.3 Restauration d'une CPU

Si vous avez sauvegardé la configuration d'une CPU précédemment, vous pouvez transférer cette sauvegarde dans la CPU. La CPU passe à l'état ARRÊT lors de la restauration d'une sauvegarde. Lorsqu'un niveau d'accès est configuré pour la CPU, vous devez fournir le mot de passe pour l'accès en lecture à la CPU.



Restauration de sauvegardes au contenu inconnu

La restauration d'une sauvegarde au contenu inconnu peut provoquer des dégâts matériels importants ou des blessures graves en présence de dysfonctionnements ou d'erreurs dans le programme.

Assurez-vous que la sauvegarde correspond à une configuration dont le contenu est connu.

Conditions

Les conditions suivantes doivent être satisfaites pour la restauration d'une sauvegarde :

- Le projet STEP 7 comprend une configuration pour la CPU et une sauvegarde réalisée précédemment.
- La CPU est directement raccordée à la console de programmation/au PC au moyen de son interface PROFINET.
- La CPU est à l'état ARRÊT
- Si un niveau d'accès était configuré, vous devez connaître le mot de passe pour l'accès complet à la CPU.

Marche à suivre

Procédez comme suit pour restaurer une sauvegarde :

1. Ouvrez la CPU dans le navigateur du projet pour afficher les objets des niveaux hiérarchiques inférieurs.
2. Sélectionnez dans le dossier "Sauvegardes en ligne" la sauvegarde que vous souhaitez restaurer.
3. Sélectionnez la commande "Charger dans l'appareil" dans le menu "En ligne".
 - Si vous aviez déjà établi une liaison en ligne (Page 1383), la boîte de dialogue "Aperçu du chargement" s'ouvre. Cette boîte de dialogue affiche des messages et recommande des actions nécessaires pour l'opération de chargement.
 - Si vous n'aviez pas établi de liaison en ligne, la boîte de dialogue "Chargement étendu" s'ouvre et vous devez d'abord sélectionner l'interface par le biais de laquelle vous voulez établir la liaison en ligne à la CPU.
4. Vérifiez les messages dans la boîte de dialogue "Aperçu du chargement" et, le cas échéant, sélectionnez les actions requises dans la colonne "Action".
5. Cliquez sur le bouton "Charger" (ce bouton devient actif dès que le chargement est possible).
6. STEP 7 restaure la sauvegarde dans la CPU. Dans la boîte de dialogue "Résultats du chargement", vous pouvez vérifier si la procédure de chargement s'est déroulée correctement et prendre toute mesure supplémentaire qui s'avérerait nécessaire.
7. Lorsque vous avez contrôlé le contenu de la boîte de dialogue "Résultats du chargement", cliquez sur le bouton "Terminer".

Si un message vous le demande, saisissez le mot de passe pour l'accès complet à la CPU et confirmez que la CPU peut passer à l'état de fonctionnement ARRÊT.

STEP 7 restaure le contenu de la sauvegarde dans la CPU et redémarre la CPU.

Remarque

Notez que vous pouvez également restaurer une sauvegarde de la CPU à partir de la page standard Restauration en ligne du serveur Web (Page 1072).

Caractéristiques techniques

A.1 Site Web Industry Online Support de Siemens

Les informations techniques pour ces produits sont disponibles sur le site Web Industry Online Support de Siemens (<https://support.industry.siemens.com/cs/ww/fr/>).

A.2 Caractéristiques techniques d'ordre général

Conformité aux normes

La conception du système d'automatisation S7-1200 est conforme aux normes et spécifications d'essai suivantes. Les critères de test pour le système d'automatisation S7-1200 se fondent sur ces normes et spécifications d'essai.

Notez que les modèles S7-1200 ne sont peut-être pas tous certifiés pour ces normes et que l'état d'homologation peut changer sans avis. Il est de votre responsabilité de déterminer les homologations applicables en se reportant aux marquages figurant sur le produit. Veuillez contacter votre agence Siemens si vous avez besoin d'informations supplémentaires concernant la dernière liste d'homologations exactes par numéro de référence.

homologation CE



Le système d'automatisation S7-1200 satisfait aux exigences et objectifs en matière de sécurité des directives CE énumérées ci-dessous et est conforme aux normes européennes harmonisées (EN) pour les automates programmables énumérées dans les Journaux officiels de l'Union Européenne.

- Directive CE 2006/95/CE (Basse tension) "Matériel électrique destiné à être employé dans certaines limites de tension"
 - EN 61131-2 Automates programmables Spécifications et essais des équipements
- Directive CE 2004/108/CE (directive CEM) "Compatibilité électromagnétique"
 - Norme sur les émissions
EN 61000-6:+A1 : environnement industriel
 - Norme sur l'immunité
EN 61000-6-2 : environnement industriel
- Directive CE 94/9/CE (ATEX) "Appareils et systèmes de protection destinés à être utilisés en atmosphères explosibles"
 - EN 60079-0:+A11
 - EN 60079-15 : Classe de protection "n"

La déclaration de conformité CE est tenue à disposition des autorités compétentes auprès de :

Siemens AG
Sector Industry
I IA AS FA DH AMB
Postfach 1963
D-92209 Amberg
Germany

Homologation cULus



Underwriters Laboratories Inc. conformément à :

- Underwriters Laboratories, Inc. : listé UL 508 (Industrial Control Equipment)
- Canadian Standards Association : CSA C22.2 numéro 142 (Process Control Equipment)

Remarque

La gamme SIMATIC S7-1200 satisfait à la norme CSA.

La marque cULus indique que le S71200 a été examiné et certifié par les laboratoires Underwriters Laboratories (UL) comme étant conforme aux normes UL 508 et CSA 22.2 numéro 142.

Homologation FM



Factory Mutual Research (FM)

Norme d'homologation classe numéro 3600 et 3611

Homologué pour utilisation dans :

Classe I, Division 2, Groupe gaz A, B, C, D, Classe de température T3C Ta = 60 °C

Classe I, Zone 2, IIC, Classe de température T3 Ta = 60 °C

Classe canadienne I, Installation zone 2 par CEC 18-150

EXCEPTION IMPORTANTE : Voir les caractéristiques techniques pour le nombre d'entrées ou de sorties pouvant être à 1 simultanément. Certains modèles sont déclassés pour Ta = 60 °C.

ATTENTION

Le remplacement de composants peut remettre en cause la conformité à la classe I, division 2 et zone 2.

Les réparations ne doivent être effectuées que par un centre de service Siemens agréé.

Homologation IECEx

EN 60079-0 : Atmosphères explosives - Règles générales

EN60079-15 : Matériel électrique pour atmosphères explosibles

Type de protection "nA"

US/FMG/ExTR14.0013

Ex nA IIC Gc T3

Des informations sur l'indice IECEx peuvent apparaître sur le produit avec les informations sur les zones dangereuses FM.

Seuls les produits dotés d'un indice IECEx sont homologués. Veuillez contacter votre agence Siemens si vous avez besoin d'informations supplémentaires concernant la dernière liste d'homologations exactes par numéro de référence.

Les modèles de relais ne sont pas inclus dans les homologations IECEx.

Veuillez consulter le marquage spécifique du produit pour l'indice de température.

Installez les modules dans une enceinte adaptée fournissant un indice de protection minimum d'IP 54 conformément à la norme CEI 60079-15.

Homologation ATEX



L'homologation ATEX ne s'applique qu'aux modèles à courant continu. Elle ne s'applique pas aux modèles à courant alternatif et relais.

EN 60079-0 : Atmosphères explosives - Règles générales

EN 60079-15 : Matériel électrique pour atmosphères explosibles ;

Type de protection 'nA'

II 3 G Ex nA II T4 ou T3 Gc

Conditions spéciales pour une utilisation sûre :

Installez les modules dans une enceinte adaptée procurant un niveau minimum de protection de IP54 conformément à la norme EN 60529, ou dans un emplacement procurant un niveau de protection équivalent.

Les câbles et les conducteurs associés doivent être classés pour la température réelle mesurée dans des conditions nominales.

Des mesures doivent être prises pour empêcher que la tension nominale aux bornes d'alimentation ne soit dépassée par des pointes de tension supérieures à 119 V.

Australie et Nouvelle-Zélande - Marque RCM (Regulatory Compliance Mark - marque de conformité réglementaire)



Le système d'automatisation S7-1200 satisfait aux exigences des normes AS/NZS 61000.6.4 et CEI 61000-6-4 (Classe A).

Certification coréenne



Le système d'automatisation S7-1200 satisfait aux exigences de l'homologation coréenne (marque KC). Il a été défini comme équipement de classe A et est conçu pour des applications industrielles et non pour un usage domestique.

Homologation de l'Union eurasienne des douanes (Biélorussie, Kazakhstan, Fédération de Russie)



EAC (Conformité eurasienne) : Déclaration de conformité selon la Norme technique de l'Union des douanes (TR CU)

Homologation pour le domaine maritime

Les produits S7-1200 sont régulièrement soumis à des homologations d'agences spéciales pour des marchés et des applications spécifiques. Veuillez contacter votre agence Siemens si vous avez besoin d'informations supplémentaires concernant la dernière liste d'homologations exactes par numéro de référence.

Sociétés de classification :

- ABS (American Bureau of Shipping) : États-Unis d'Amérique
- BV (Bureau Veritas) : France
- DNV (Det Norske Veritas) : Norvège
- GL (Germanischer Lloyd) : Allemagne
- LRS (Lloyds Register of Shipping) : Angleterre
- Class NK (Nippon Kaiji Kyokai) : Japon
- Korean Register of Shipping : Corée
- CSS (China Classification Society) : Chine

Environnements industriels

Le système d'automatisation S7-1200 est conçu pour être utilisé dans des environnements industriels.

Tableau A- 1 Environnements industriels

Champ d'application	Exigences en matière d'émissions	Exigences en matière d'immunité
Industriel	EN 61000-6-4:2007+A1	EN 61000-6-2:2005

Remarque

Le système d'automatisation S7-1200 est destiné à une utilisation dans des zones industrielles ; une utilisation dans des zones résidentielles peut avoir un impact sur les réceptions radio ou TV. Si vous utilisez le S7-1200 dans des zones résidentielles, vous devez veiller à ce que son émission d'interférences radio respecte la valeur limite de la Classe B conformément à l'EN 55011.

Les exemples de mesures appropriées pour atteindre le niveau d'interférences RF, Classe B, comprennent :

- L'installation du S7-1200 dans une armoire de commande mise à la terre
- L'utilisation de filtres antiparasites dans les conduites d'alimentation

Veillez à ce que l'émission d'interférences radio respecte la Classe B conformément à l'EN 55011.

Une acceptation individuelle est requise (l'installation finale doit remplir toutes les conditions de sécurité de CEM d'une installation résidentielle).

Compatibilité électromagnétique

La compatibilité électromagnétique (CEM) est la capacité d'un appareil électrique à fonctionner comme prévu dans un environnement électromagnétique et à fonctionner sans émettre des niveaux d'interférence électromagnétique pouvant perturber d'autres appareils électriques à proximité.

Tableau A- 2 Immunité selon EN 61000-6-2

Compatibilité électromagnétique - Immunité selon EN 61000-6-2	
EN 61000-4-2 Décharge électrostatique	8 kV : décharge dans l'air vers toutes les surfaces 6 kV : décharge au contact vers les surfaces conductrices exposées
EN 61000-4-3 Essai d'immunité aux champs électromagnétiques rayonnés aux fréquences radioélectriques	80 à 1000 MHz, 10 V/m, 80% de modulation d'amplitude à 1 kHz 1,4 à 2,0 GHz, 3 V/m, 80% de modulation d'amplitude à 1 kHz 2,0 à 2,7 GHz, 1 V/m, 80% de modulation d'amplitude à 1 kHz
EN 61000-4-4 Salves transitoires rapides	2 kV, 5 kHz avec réseau de couplage vers système CC et CA 2 kV, 5 kHz avec blocage de couplage vers E/S
EN 6100-4-5 Immunité aux pointes de tension	Systèmes CA - état courant 2 kV, état différentiel 1 kV systèmes CC - état courant 2 kV, état différentiel 1 kV Pour des systèmes CC, reportez-vous à l'immunité aux pointes de tensions ci-dessous
EN 61000-4-6 Perturbations par conduction	150 kHz à 80 MHz, 10 V eff., 80 % de modulation d'amplitude à 1 kHz
EN 61000-4-11 Baisses de tension	Systèmes CA 0% pour 1 cycle, 40% pour 12 cycles et 70% pour 30 cycles à 60 Hz

Tableau A- 3 Émissions conduites et rayonnées selon EN 61000-6-4

Compatibilité électromagnétique - Émissions conduites et rayonnées selon EN 61000-6-4		
Émissions conduites EN 55016, classe A, groupe 1	0,15 MHz à 0,5 MHz	< 79 dB (µV) quasi-pointe ; < 66 dB (µV) moyenne
	0,5 MHz à 5 MHz	< 73 dB (µV) quasi-pointe ; < 60 dB (µV) moyenne
	5 MHz à 30 MHz	< 73 dB (µV) quasi-pointe ; < 60 dB (µV) moyenne
Émissions rayonnées EN 55016, classe A, groupe 1	30 MHz à 230 MHz	< 40 dB (µV/m) quasi-pointe, mesuré à 10 m
	230 MHz à 1 GHz	< 47 dB (µV/m) quasi-pointe, mesuré à 10 m
	1 GHz à 3 GHz	< 76 dB (µV/m) quasi-pointe, mesuré à 10 m

Immunité aux pointes de tension

Les systèmes de câblage soumis à des pointes de tension provenant d'un couplage en cas de foudre doivent être équipés d'une protection externe. On trouve une caractéristique technique pour l'évaluation d'une protection des pointes de tension de type foudre dans EN 61000-4-5, avec des limites opérationnelles établies par EN 61000-6-2. Les CPU du S7-1200 DC et les modules entrées-sorties nécessitent une protection externe pour conserver un fonctionnement sécurisé lorsqu'ils sont soumis à des tensions de pointes de tension définies par cette norme.

Certains appareils qui prennent en charge la protection nécessaire d'immunité aux pointes de tension sont répertoriés ci-après. Ces appareils n'assurent la protection que s'ils sont convenablement installés selon les recommandations du fabricant. Les appareils fabriqués par d'autres fournisseurs ayant les mêmes caractéristiques techniques ou de meilleures caractéristiques peuvent également être utilisés :

Tableau A- 4 Appareils prenant en charge la protection d'immunité aux pointes de tension

Sous-système	Appareil de protection
Alimentation +24 V CC	BLITZDUCTOR VT, BVT AVD 24, référence 918 422
Industrial Ethernet	DEHNpatch DPA M CLE RJ45B 48, référence 929 121
RS-485	BLITZDUCTOR XT, Basic Unit BXT BAS, référence 920 300
	BLITZDUCTOR XT, Module BXT ML2 BD HFS 5, référence 920 271
RS-232	BLITZDUCTOR XT, Basic Unit BXT BAS, référence 920 300
	BLITZDUCTOR XT, Module BXT ML2 BE S 12, référence 920 222
Entrées TOR +24 V CC	DEHN, Inc., Type DCO SD2 E 24, référence 917 988
Sorties TOR et alimentation de capteur +24 V CC	DEHN, Inc., Type DCO SD2 E 24, référence 917 988
E/S analogiques	DEHN, Inc., Type DCO SD2 E 12, référence 917 987
Sorties relais	Aucune nécessaire

Conditions ambiantes

Tableau A- 5 Transport et stockage

Conditions ambiantes - Transport et stockage	
EN 6006822, test Bb, chaleur sèche EN 6006821, test Ab, froid	- 40 °C à + 70 °C
EN 60068-2-30, test Db, chaleur humide saturée	25 °C à 55 °C, 95 % d'humidité
EN 60068-2-14, test Na, choc de température	-40 °C à +70 °C, temps de maintien 3 heures, 2 cycles
EN 60068-2-32, chute libre	0,3 m, 5 fois, emballage du produit
Pression atmosphérique	1 139 à 660 hPa (ce qui correspond à une altitude de - 1 000 à 3 500 m)

Tableau A- 6 Conditions ambiantes climatiques

Conditions ambiantes - Conditions climatiques	
Le système d'automatisation S7-1200 est conçu pour être utilisé dans des locaux à l'abri des intempéries. Les conditions de fonctionnement sont basées sur les exigences de la norme CEI 60721-3-3 :	
<ul style="list-style-type: none"> • Classe 3M3 (exigences mécaniques) • Classe 3K3 (exigences climatiques) 	
Plage de température ambiante (admission d'air 25 mm en dessous de l'unité)	- 20 °C à 60 °C en montage horizontal - 20 °C à 50 °C en montage vertical 95 % d'humidité sans condensation Sauf indication contraire
Pression atmosphérique	1 139 à 795 hPa (ce qui correspond à une altitude de - 1 000 à 2 000 m)
Concentration de contaminants	SO ₂ : < 0.5 ppm ; H ₂ S : < 0.1 ppm ; RH < 60 % sans condensation ISA-S71.04 niveau de gravité G1, G2, G3
EN 60068-2-14, test Nb, changement de température	0 °C à 60 °C
EN 60068-2-27 Choc mécanique	15 g, impulsion de 11 ms, 6 chocs dans chacun des 3 axes
EN 60068-2-6 Vibrations sinusoïdales	Montage sur profilé support : 3,5 mm de 5 à 9 Hz, 1 G de 8,4 à 150 Hz Montage sur panneau : 7,0 mm de 5 à 8,4 Hz, 2 G de 8,4 à 150 Hz 10 balayages par axe, 1 octave/minute

Niveau de pollution/catégorie de surtension selon CEI 61131-2

- Degré de pollution 2
- Catégorie de surtension : II

Classe de protection

- Classe de protection II selon EN 61131-2 (un conducteur de protection n'est pas requis)

Degré de protection

- IP20 Protection mécanique, EN 60529
- Protection contre le contact des doigts avec la haute tension comme testé par sonde standard. Une protection externe est nécessaire contre la poussière, la saleté, l'eau et les objets étrangers de diamètre inférieur à 12,5 mm.

Tensions nominales

Tableau A- 7 Tensions nominales

Tension nominale	Tolérance
24 V CC	20,4 V CC à 28,8 V CC
120/230 V CA	85 V CA à 264 V CA, 47 à 63 Hz

Remarque

Lorsqu'un contact mécanique active la puissance de sortie vers la CPU S7-1200 ou tout autre module d'extension TOR, il envoie un signal "1" aux sorties TOR pendant environ 50 microsecondes. Cela peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants. Vous devez en tenir compte dans votre planification, en particulier si vous utilisez des appareils qui réagissent aux impulsions de courte durée.

Protection contre la tension inverse

Un circuit de protection contre la tension inverse est fourni sur chaque paire de bornes de l'alimentation +24 V CC ou de l'alimentation d'entrée utilisateur pour les CPU, les modules d'entrées-sorties (SM) et les Signal Boards (SB). Endommager le système reste possible si l'on câble des paires de bornes différentes dans des polarités inverses.

Certains ports d'entrée d'alimentation 24 V CC dans le système S7-1200 sont interconnectés, avec un circuit logique commun connectant plusieurs bornes M. Par exemple, les circuits suivants sont interconnectés lorsqu'ils sont signalés comme "non isolés" dans les fiches techniques : l'alimentation 24 V CC de la CPU, l'alimentation de capteur de la CPU, l'entrée d'alimentation pour la bobine de relais d'un SM et l'alimentation pour une entrée analogique non isolée. Toutes les bornes M non isolées doivent être connectées au même potentiel de référence externe.

ATTENTION

Connecter des bornes M non isolées à des potentiels de référence différents provoque des flux de courant indésirables qui peuvent être à l'origine de dégâts ou d'un fonctionnement imprévisible dans l'automate et tout équipement connecté.

Le non-respect de ces conseils peut être à l'origine de dégâts ou d'un fonctionnement imprévisible pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Vérifiez toujours que toutes les bornes M non isolées dans un système S7-1200 sont connectées au même potentiel de référence.

Sorties CC

Aucun circuit de protection contre les courts-circuits n'est fourni pour les sorties CC des CPU, modules d'entrées-sorties (SM) et Signal Boards (SB).

Durée d'utilisation électrique des relais

Les données de performance types estimées à partir des exemples de tests sont représentées ci-dessous. Les performances effectives peuvent varier selon votre application spécifique. Un circuit de protection externe qui est adapté à la charge améliorera la durée de vie des contacts. Les contacts N.C. ont une durée de vie type d'environ un tiers de celle du contact N.O. dans des conditions de charge inductive et de lampe.

Un circuit protecteur externe augmentera la durée de vie des contacts.

Tableau A- 8 Données de performance types

Données pour sélectionner un actionneur				
Courant thermique continu		2 A max.		
Commutation de la capacité et de la durée de vie des contacts				
Pour une charge ohmique	Tension	Courant	Nombre de cycles de fonctionnement (type)	
	24 V CC	2,0 A	0,1 million	
	24 V CC	1,0 A	0,2 million	
	24 V CC	0,5 A	1,0 million	
	48 V CA	1,5 A	1,5 million	
	60 V CA	1,5 A	1,5 million	
	120 V C A	2,0 A	1,0 million	
	120 V C A	1,0 A	1,5 million	
	120 V C A	0,5 A	2,0 millions	
	230 V C A	2,0 A	1,0 million	
	230 V C A	1,0 A	1,5 million	
	230 V C A	0,5 A	2,0 millions	

Données pour sélectionner un actionneur			
Pour une charge inductive (selon IEC 947-5-1 DC13/AC15)	Tension	Courant	Nombre de cycles de fonctionnement (type)
	24 V CC	2,0 A	0,05 million
	24 V CC	1,0 A	0,1 million
	24 V CC	0,5 A	0,5 million
	24 V CA	1,5 A	1,0 million
	48 V CA	1,5 A	1,0 million
	60 V CA	1,5 A	1,0 million
	120 V C A	2,0 A	0,7 million
	120 V C A	1,0 A	1,0 million
	120 V C A	0,5 A	1,5 million
	230 V C A	2,0 A	0,7 million
	230 V C A	1,0 A	1,0 million
	230 V C A	0,5 A	1,5 million
Activation d'une entrée numérique		Possible	
Fréquence de commutation			
	Mécanique	max. 10 Hz	
	Pour une charge ohmique	max. 1 Hz	
	Pour une charge inductive (selon IEC 947-5-1 DC13/AC15)	max. 0,5 Hz	
	Pour une charge de lampe	max. 1Hz	

Rémanence de la mémoire interne de la CPU

- Durée de vie des données rémanentes et des données des journaux de données : 10 ans
- Données rémanentes hors tension, durée des écritures en cycles : 2 million de cycles
- Données de journaux : durée des écritures en cycles : 500 millions d'entrées dans les journaux de données

Remarque

Effet des journaux de données sur la mémoire interne de la CPU

Chaque écriture dans un journal de données utilise au minimum 2 Ko de mémoire. Si votre programme écrit fréquemment de petites quantités de données, il utilise au moins 2 Ko de mémoire à chaque écriture. Une meilleure solution consiste à accumuler les petits éléments de données dans un bloc de données (DB) et à écrire le bloc de données dans le journal de données à des intervalles moins fréquents.


Si votre programme écrit de nombreuses entrées dans le journal de données à une fréquence élevée, vous devriez envisager d'utiliser une carte mémoire SD remplaçable.

A.3 Brochages de port X1 d'interface PROFINET

La CPU S7-1200 se connecte au réseau PROFINET en utilisant une prise jack RJ45 femelle standard. Le brochage de connecteur dépend du type de CPU.

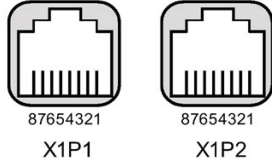
CPU à port unique

Les CPU à port unique (CPU 1211C, CPU 1212C et CPU 1214C) possèdent une configuration de broche MDI Ethernet standard, comme suit :

Broche	Nom de signal	Description	Brochage de prise jack RJ45 femelle
1	TD+	Transmission de données	
2	TD-		
3	Niveau d'exigence+	Réception de données	
4	GND	Terre	
5	GND		
6	RD-	Réception de données	
7	GND	Terre	
8	GND		

CPU à double port

Les CPU à double port (CPU 1215C et CPU1217C) possèdent une configuration de broche MDI-X Ethernet standard, comme suit :

Broche	Nom de signal	Description	Brochage de prise jack RJ45 femelle
1	Niveau d'exigence+	Réception de données	 <p>87654321 87654321 X1P1 X1P2</p>
2	RD-	Transmission de données	
3	TD+		
4	GND	Terre	
5	GND		
6	TD-	Transmission de données	
7	GND	Terre	
8	GND		

Autonégociation

Si la configuration du port permet l'autonégociation, la CPU S7-1200 détecte automatiquement le type de câble et échange les lignes de transmission/de réception, si nécessaire. Si la configuration du port ne permet pas l'autonégociation, la CPU désactive également cet échange automatique. Configuration d'un réglage d'autonégociation de port dans la boîte de dialogue des options de port de TIA Portal. Il s'agit d'une option avancée spécifique au port pour l'interface PROFINET (X1) des propriétés de la CPU. Pour plus d'informations, reportez-vous à "Configuration du port PROFINET" dans la section 11.2.3.4 : "Configuration d'une adresse IP pour une CPU dans votre projet" (Page 840).

A.4 CPU 1211C

A.4.1 Caractéristiques et fonctions générales

Tableau A- 9 Caractéristiques générales

Caractéristiques techniques	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Référence	6ES7211-1BE40-0XB0	6ES7211-1HE40-0XB0	6ES7211-1AE40-0XB0
Dimensions L x H x P (mm)	90 x 100 x 75		
Poids avec emballage	420 grammes	380 grammes	370 grammes
Dissipation de courant	10 W	8 W	
Courant disponible (bus CM)	750 mA max. (5 V CC)		
Courant disponible (24 V CC)	300 mA max. (alimentation de capteur)		
Consommation de courant entrées TOR (24 V CC)	4 mA / entrée utilisée		

Tableau A- 10 Fonctions de la CPU

Caractéristiques techniques		Description
Mémoire utilisateur (Reportez-vous à "Caractéristiques techniques générales" (Page 1429), "Conservation de la mémoire de la CPU interne".)	de travail	50 Ko
	de chargement	1 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées		6 entrées/4 sorties
E/S analogiques intégrées		2 entrées
Taille de la mémoire image		1024 octets d'entrées (I) / 1024 octets de sorties (Q)
Mémentos (M)		4096 octets
Mémoire temporaire (locale)		<ul style="list-style-type: none"> • 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus) • 6 Ko pour chacun des autres niveaux de priorité d'alarme (FB et FC inclus)
Modules d'entrées-sorties pour extension		aucun
SB, CB, BB pour extension		1 max.
Modules de communication pour extension		3 CM max.
Compteurs rapides		Jusqu'à 6 configurés pour utiliser toute entrée intégrée ou du SB. Voir "Affectation d'entrée matérielle" (Page 599) pour la CPU 1211C : affectation d'adresses HSC par défaut. 100/180 kHz (Ia.0 à Ia.5)
Sorties d'impulsions ²		Jusqu'à 4 configurées pour utiliser toute sortie intégrée ou du SB. 100 kHz (Qa.0 à Qa.3)
Entrées de capture d'impulsions		6
Alarmes temporisées		4 au total avec résolution de 1 ms
Alarmes cycliques		4 au total avec résolution de 1 ms
Alarmes sur front		6 sur front montant et 6 sur front descendant (10 et 10 avec Signal Board optionnel)
Carte mémoire		Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel		+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel		20 jours typ./12 jours min. à 40 °C (supercondensateur sans maintenance)

¹ La vitesse plus faible s'applique lorsque le HSC est configuré pour le fonctionnement en quadrature de phase.

² Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Tableau A- 11 Performances

Type d'instruction		Vitesse d'exécution	
		Adressage direct (I, Q et M)	Accès au DB
Instructions booléennes		0,08 µs/instruction	
Déplacement	Move_Bool	0,3 µs/instruction	1,17 µs/instruction
	Move_Word	0,137 µs/instruction	1,0 µs/instruction
	Move_Real	0,72 µs/instruction	1,0 µs/instruction
Opérations arithmétiques sur réels	Add_Real	1,48 µs/instruction	1,78 µs/instruction

Remarque

De nombreuses variables affectent les temps mesurés. Les temps de performance ci-dessus concernent les instructions les plus rapides dans cette catégorie et les programmes exempts d'erreur.

A.4.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1211C

Tableau A- 12Blocs, temporisations et compteurs pris en charge par la CPU 1211C

Elément		Description
Blocs	Type	OB, FB, FC, DB
	Taille	30 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	FB et FC : 1 à 65535 (FB 1 à FB 65535, par exemple) DB : 1 à 59999
	Profondeur d'imbrication	16 en cas d'appel depuis l'OB de cycle de programme ou de démarrage 6 en cas d'appel depuis un OB d'alarme associé à un événement ¹
	Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.
OB	Cycle de programme	Multiple
	Mise en route	Multiple
	Alarme temporisée	4 (1 par événement)
	Alarmes cycliques	4 (1 par événement)
	Alarmes de processus	50 (1 par événement)
	Alarmes d'erreur de temps	1
	Alarmes de diagnostic	1
	Débrochage ou enfichage de modules	1
	Défaillance du châssis ou de la station	1
	Heure	Multiple
	Etat	1
	Mettre à jour	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
MC-PostServo	1	
Temporisations	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> • SInt, USInt : 3 octets • Int, UInt : 6 octets • DInt, UDInt : 12 octets

¹ Les programmes de sécurité utilisent deux niveaux d'imbrication. Le programme utilisateur a donc une profondeur d'imbrication de quatre dans les programmes de sécurité.

Tableau A- 13Communication

Caractéristiques techniques	Description
Nombre de ports	1
Type	Ethernet
Appareil IHM	4
Console de programmation (PG)	1
Liaisons	<ul style="list-style-type: none"> • 8 liaisons pour la communication ouverte Open User Communication (active ou passive) : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV • 8 liaisons CPU à CPU (client ou serveur) pour les données GET/PUT • 6 liaisons pour l'affectation dynamique à GET/PUT ou Open User Communication
Débits	10/100 Mb/s
Isolation (signal externe à logique)	Transformateur isolé, 1 500 V CA (essai de type) ¹
Type de câble	CAT5e blindé
Interfaces	
Nombre d'interfaces PROFINET	1
Nombre d'interfaces PROFIBUS	0
Interface	
Matériel	
Nombre de ports	1
Commutateur intégré	Non
RJ-45 (Ethernet)	Oui ; X1
Protocoles	
Contrôleur PROFINET IO	Oui
Périphérique PROFINET IO	Oui
Communication SIMATIC	Oui
Communication IE ouverte	Oui
Serveur Web	Oui
Redondance des supports	Non
Contrôleur PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT	Non
MRP	Non
PROFIenergy	Oui. La CPU S7-1200 ne prend en charge que l'unité PROFIenergy (avec la fonctionnalité I-device).
Démarrage priorisé	Oui (16 périphériques PROFINET max.)
Nombre de périphériques IO raccordables max.	16

Caractéristiques techniques	Description
Nombre de périphériques IO que vous pouvez raccorder pour RT, max.	16
Dont en ligne, max.	16
Nombre de périphériques IO pouvant être activés/désactivés simultanément, max.	8
Temps d'actualisation	La valeur minimale du temps d'actualisation dépend également du composant de communication défini pour PROFINET IO, du nombre de périphériques IO et de la quantité de données utilisateur configurées.
Avec RT	
Horloge de transmission de 1 ms	1 ms à 512 ms
Périphérique PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT pris en charge	Non
MRP pris en charge	Non
PROFenergy	Oui
Appareil partagé	Oui
Nombre de contrôleurs IO avec appareil partagé, max.	2
Communication SIMATIC	
Communication S7, en tant que serveur	Oui
Communication S7, en tant que client	Oui
Données utilisateur par tâche, max.	Voir l'aide en ligne (Communication S7, taille des données utilisateur)
Communication IE ouverte	
TCP/IP	Oui
Longueur de données, max.	8 ko
Prise en charge de plusieurs liaisons passives par port	Oui
ISO-on-TCP (RFC1006)	Oui
Longueur de données, max.	8 ko
UDP	Oui
Longueur de données, max.	1 472 octets
DHCP	Non
SNMP	Oui
DCP	Oui
LLDP	Oui

¹ L'isolation du port Ethernet est conçue pour limiter les risques pendant les pannes de réseau de courte durée avec tensions dangereuses. Elle ne satisfait pas aux exigences en matière de sécurité pour l'isolation des lignes CA de routine.

Tableau A- 14Alimentation

Caractéristiques techniques		CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Plage de tension		85 à 264 V CA	20,4 V CC à 28,8 V CC	
Fréquence de ligne		47 à 63 Hz	--	
Courant d'en- trée	CPU seulement à la charge max.	60 mA à 120 V CA 30 mA à 240 V CA	300 mA à 24 V CC	300 mA à 24 V CC
	CPU avec tous les acces- soires d'extension à la charge max.	180 mA à 120 V CA 90 mA à 240 V CA	900 mA à 24 V CC	
Appel de courant (max.)		20 A à 264 V CA	12 A à 28,8 V CC	
I ² t		0,8 A ² s	0,5 A ² s	
Isolation (alimentation d'entrée à logique)		1 500 V CA	Non isolée	
Fuite à la terre, ligne CA à la terre fonc- tionnelle		0,5 mA max.	--	
Temps de retard (perte d'alimentation)		20 ms à 120 V CA 80 ms à 240 V CA	10 ms à 24 V CC	
Fusible interne, non remplaçable par l'utilisateur		3 A, 250 V, action retardée		

Tableau A- 15Alimentation de capteur

Caractéristiques techniques		CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Plage de tension		20,4 à 28,8 V CC	L+ moins 4 V CC min.	
Courant de sortie nominal (max.)		300 mA (protégé contre les courts-circuits)		
Bruit d'ondulation maximum (<10 MHz)		< 1 V crête à crête	Comme la ligne d'entrée	
Isolation (logique CPU à alimentation de capteur)		Non isolée		

A.4.3 Entrées et sorties TOR

Tableau A- 16 Entrées TOR

Caractéristiques techniques	CPU 1211C AC/DC/Relais, CPU 1211C DC/DC/Relais et CPU 1211C DC/DC/DC
Nombre d'entrées	6
Type	P/N (CEI type 1 en mode P)
Tension nominale	24 V CC à 4 mA, nominal
Tension continue admise	30 V CC max.
Tension de choc	35 V CC pour 0,5 s
Signal 1 logique (min.)	15 V CC à 2,5 mA
Signal 0 logique (max.)	5 V CC à 1 mA
Isolation (site à logique)	707 V CC (essai de type)
Groupes d'isolation	1
Temps de filtre	microsecondes : 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 millisecondes : 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V CC)	100/80 kHz (la.0 à la.5)
Nombre d'entrées simultanément à 1	6 à 60 °C horizontal, 50 °C vertical
Longueur de câble (mètres)	500 m blindé, 300 m non blindé, 50 m blindé pour entrées HSC

Tableau A- 17 Sorties TOR

Caractéristiques techniques	CPU 1211C AC/DC/Relais et CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Nombre de sorties	4	
Type	Relai, mécanique	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V CC ou 5 à 250 V CA	20,4 à 28,8 V CC
Signal 1 logique à courant max.	--	20 V CC min.
Signal 0 logique avec charge 10 k Ω	--	0,1 V CC max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 μ A max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	
Isolation (site à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)	707 V CC (essai de type)
Groupes d'isolation	1	
Tension de blocage inductive	--	L+ moins 48 V CC, dissipation 1 W

Caractéristiques techniques	CPU 1211C AC/DC/Relais et CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Fréquence de commutation maximum des relais	1 Hz	--
Retard de commutation (Qa.0 à Qa.3)	10 ms max.	1,0 µs max., de 0 à 1 3,0 µs max., de 1 à 0
Fréquence de sortie de trains d'impulsions	Non recommandé ¹	100 kHz (Qa.0 à Qa.3) ² , 2 Hz min.
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Nombre de sorties simultanément à 1	4 à 60 °C horizontal, 50 °C vertical	
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	

- ¹ Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.
- ² Selon vos câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10 % du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

A.4.4 Entrées analogiques

Tableau A- 18 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V
Plage de dépassement (mot de données)	27649 à 32511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32512 à 32767
Résolution	10 bits
Tension de tenue maximum	35 V CC
Lissage	Aucun, faible, moyen ou fort Reportez-vous au tableau sur la Réponse indicielle (ms) des entrées analogiques de la CPU (Page 1451).
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥ 100 kΩ
Isolation (site à logique)	Aucune

Caractéristiques techniques	Description
Précision (25 °C / -20 à 60 °C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

A.4.4.1 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 19 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms
Temps d'échantillonnage	4,17 ms	5 ms	25 ms

A.4.4.2 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 20 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

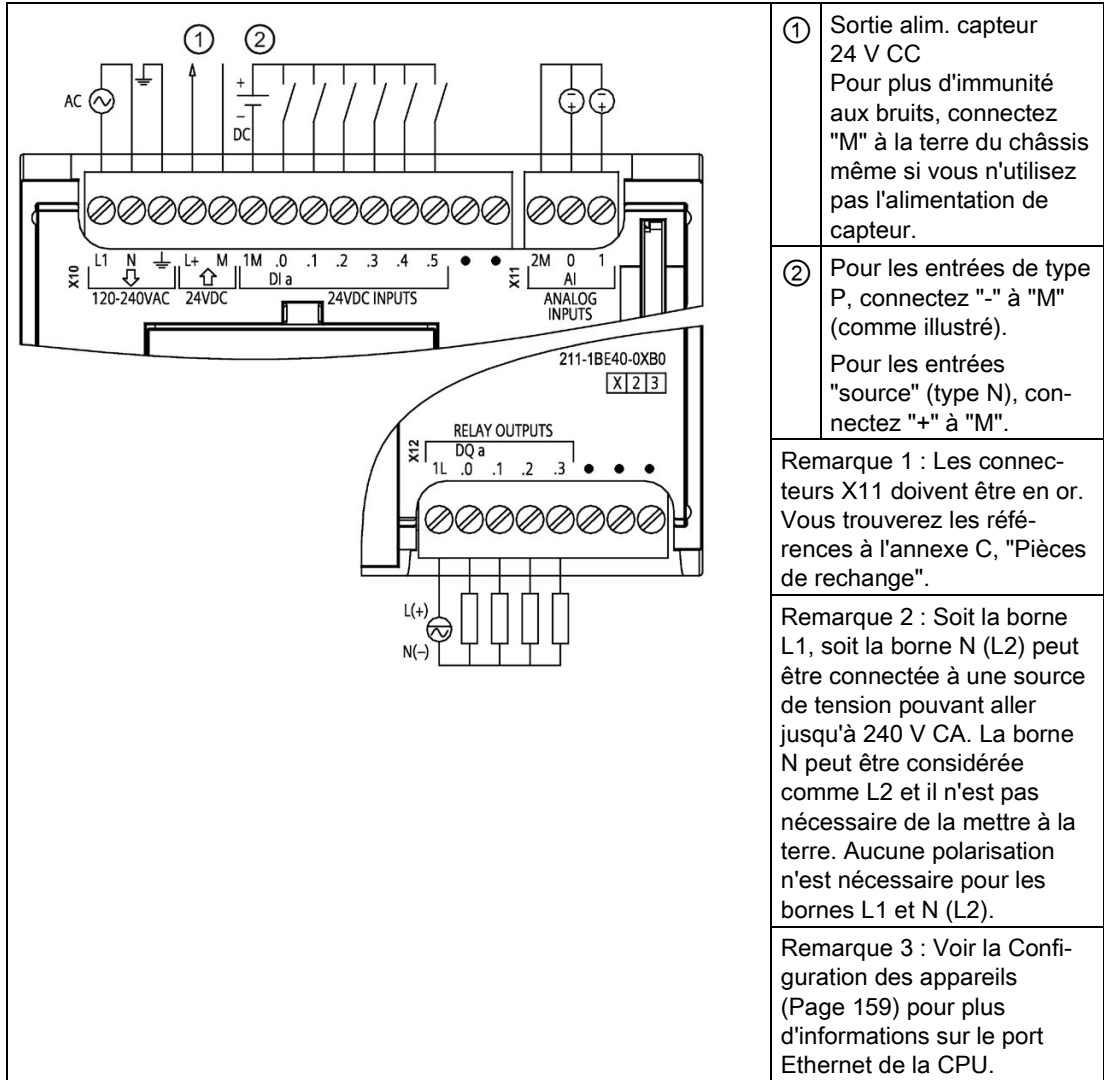
A.4.4.3 Plages de mesure des entrées analogiques pour la tension (CPU)

Tableau A- 21 Représentation des entrées analogiques pour la tension (CPU)

Technologie		Plage de mesure de tension	
Décimal	Hexadécimal	0 à 10 V	
32767	7FFF	11,852 V CC	Débordement haut
32512	7F00		
32511	7EFF	11,759 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valeurs négatives		Valeurs négatives impossibles	

A.4.5 Schémas de câblage de la CPU 1211C

Tableau A- 22CPU 1211C AC/DC/Relais (6ES7211-1BE40-0XB0)



① Sortie alim. capteur 24 V CC
 Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.

② Pour les entrées de type P, connectez "-" à "M" (comme illustré).
 Pour les entrées "source" (type N), connectez "+" à "M".

Remarque 1 : Les connecteurs X11 doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Remarque 2 : Soit la borne L1, soit la borne N (L2) peut être connectée à une source de tension pouvant aller jusqu'à 240 V CA. La borne N peut être considérée comme L2 et il n'est pas nécessaire de la mettre à la terre. Aucune polarisation n'est nécessaire pour les bornes L1 et N (L2).

Remarque 3 : Voir la Configuration des appareils (Page 159) pour plus d'informations sur le port Ethernet de la CPU.

Tableau A- 23 Brochage pour la CPU 1211C AC/DC/Relais (6ES7211-1BE40-0XB0)

Broche	X10	X11 (or)	X12
1	L1/120-240 V CA	2 M	1L
2	N/120-240 V CA	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+ / Sortie capteur 24 V CC	--	DQ a.2
5	M / Sortie capteur 24 V CC	--	DQ a.3
6	1M	--	Pas de connexion
7	DI a.0	--	Pas de connexion
8	DI a.1	--	Pas de connexion
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Pas de connexion	--	--
14	Pas de connexion	--	--

Tableau A- 24 CPU 1211C DC/DC/Relais (6ES7211-1HE40-0XB0)

① Sortie alim. capteur
24 V CC
Pour plus d'immunité aux perturbations, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.

② Pour les entrées de type P, connectez "-" à "M" (comme illustré).
Pour les entrées "source" (type N), connectez "+" à "M".

Remarque 1 : Les connecteurs X11 doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Remarque 2 : Voir la Configuration des appareils (Page 159) pour plus d'informations sur le port Ethernet de la CPU.

Tableau A- 25 Brochage pour la CPU 1211C DC/DC/Relais (6ES7211-1HE40-0XB0)

Broche	X10	X11 (or)	X12
1	L+/24 V CC	2 M	1L
2	M/24 V CC	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+/Sortie capteur 24 V CC	--	DQ a.2
5	M/Sortie capteur 24 V CC	--	DQ a.3
6	1M	--	Pas de connexion
7	DI a.0	--	Pas de connexion
8	DI a.1	--	Pas de connexion
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Pas de connexion	--	--
14	Pas de connexion	--	--

Tableau A- 26 CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)

① Sortie alim. capteur 24 V CC
Pour plus d'immunité aux perturbations, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.

② Pour les entrées de type P, connectez "-" à "M" (comme illustré).
Pour les entrées "source" (type N), connectez "+" à "M".

Remarque 1 : Les connecteurs X11 doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Remarque 2 : Voir la Configuration des appareils (Page 159) pour plus d'informations sur le port Ethernet de la CPU.

Tableau A- 27 Brochage pour la CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)

Broche	X10	X11 (or)	X12
1	L+/24 V CC	2 M	3L+
2	M/24 V CC	AI 0	3M
3	Terre fonctionnelle	AI 1	DQ a.0
4	L+/Sortie capteur 24 V CC	--	DQ a.1
5	M/Sortie capteur 24 V CC	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	Pas de connexion
8	DI a.1	--	Pas de connexion
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Pas de connexion	--	--
14	Pas de connexion	--	--

Remarque

Les entrées analogiques inutilisées doivent être court-circuitées.

A.5 CPU 1212C

A.5.1 Caractéristiques et fonctions générales

Tableau A- 28 Généralités

Caractéristiques techniques	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Référence	6ES7212-1BE40-0XB0	6ES7212-1HE40-0XB0	6ES7212-1AE40-0XB0
Dimensions L x H x P (mm)	90 x 100 x 75		
Poids avec emballage	425 grammes	385 grammes	370 grammes
Dissipation de courant	11 W	9 W	
Courant disponible (bus SM et CM)	1 000 mA max. (5 V CC)		
Courant disponible (24 V CC)	300 mA max. (alimentation de capteur)		
Consommation de courant entrées TOR (24 V CC)	4 mA / entrée utilisée		

Tableau A- 29 Fonctions de la CPU

Caractéristiques techniques		Description
Mémoire utilisateur (Reportez-vous à "Caractéristiques techniques générales (Page 1429)", "Conservation de la mémoire de la CPU interne".)	de travail	75 Ko
	de chargement	2 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées		8 entrées/6 sorties
E/S analogiques intégrées		2 entrées
Taille de la mémoire image		1024 octets d'entrées (I) / 1024 octets de sorties (Q)
Mémentos (M)		4096 octets
Mémoire temporaire (locale)		<ul style="list-style-type: none"> • 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus) • 6 Ko pour chacun des autres niveaux de priorité d'alarme (FB et FC inclus)
Modules d'entrées-sorties pour extension		2 SM max.
SB, CB, BB pour extension		1 max.
Modules de communication pour extension		3 CM max.
Compteurs rapides		<p>Jusqu'à 6 configurés pour utiliser toute entrée intégrée ou du SB. Voir "Affectation d'entrée matérielle" (Page 599) pour la CPU 1212C : affectation d'adresses HSC par défaut.</p> <ul style="list-style-type: none"> • 100/180 kHz (Ia.0 à Ia.5) • 30/120 kHz (Ia.6 à Ia.7)
Sorties d'impulsions ²		<p>Jusqu'à 4 configurées pour utiliser toute sortie intégrée ou du SB.</p> <ul style="list-style-type: none"> • 100 kHz (Qa.0 à Qa.3) • 20 kHz (Qa.4 à Qa.5)
Entrées de capture d'impulsions		8
Alarmes temporisées		4 au total avec résolution de 1 ms
Alarmes cycliques		4 au total avec résolution de 1 ms
Alarmes sur front		8 sur front montant et 8 sur front descendant (12 et 12 avec Signal Board optionnel)
Carte mémoire		Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel		+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel		20 jours typ./12 jours min. à 40 °C (supercondensateur sans maintenance)

¹ La vitesse plus faible s'applique lorsque le HSC est configuré pour le fonctionnement en quadrature de phase.

² Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Tableau A- 30 Performances

Type d'instruction		Vitesse d'exécution	
		Adressage direct (I, Q et M)	Accès au DB
Instructions booléennes		0,08 µs/instruction	
Déplacement	Move_Bool	0,3 µs/instruction	1,17 µs/instruction
	Move_Word	0,137 µs/instruction	1,0 µs/instruction
	Move_Real	0,72 µs/instruction	1,0 µs/instruction
Opérations arithmétiques sur réels	Add_Real	1,48 µs/instruction	1,78 µs/instruction

Remarque

De nombreuses variables affectent les temps mesurés. Les temps de performance ci-dessus concernent les instructions les plus rapides dans cette catégorie et les programmes exempts d'erreur.

A.5.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1212C

Tableau A- 31 Blocs, temporisations et compteurs pris en charge par la CPU 1212C

Elément		Description
Blocs	Type	OB, FB, FC, DB
	Taille	50 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	FB et FC : 1 à 65535 (FB 1 à FB 65535, par exemple) DB : 1 à 59999
	Profondeur d'imbrication	16 en cas d'appel depuis l'OB de cycle de programme ou de démarrage 6 en cas d'appel depuis un OB d'alarme associé à un événement ¹
	Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.
OB	Cycle de programme	Multiple
	Mise en route	Multiple
	Alarme temporisée	4 (1 par événement)
	Alarmes cycliques	4 (1 par événement)
	Alarmes de processus	50 (1 par événement)
	Alarmes d'erreur de temps	1
	Alarmes de diagnostic	1
	Débrochage ou enfichage de modules	1
	Défaillance du châssis ou de la station	1
	Heure	Multiple
	Etat	1
	Mettre à jour	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
MC-PostServo	1	
Temporisations	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> • SInt, USInt : 3 octets • Int, UInt : 6 octets • DInt, UDInt : 12 octets

¹ Les programmes de sécurité utilisent deux niveaux d'imbrication. Le programme utilisateur a donc une profondeur d'imbrication de quatre dans les programmes de sécurité.

Tableau A- 32Communication

Caractéristiques techniques	Description
Nombre de ports	1
Type	Ethernet
Appareil IHM	4
Console de programmation (PG)	1
Liaisons	<ul style="list-style-type: none"> 8 liaisons pour la communication ouverte Open User Communication (active ou passive) : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV 8 liaisons CPU à CPU (client ou serveur) pour les données GET/PUT 6 liaisons pour l'affectation dynamique à GET/PUT ou Open User Communication
Débits	10/100 Mb/s
Isolation (signal externe à logique)	Transformateur isolé, 1 500 V CA (essai de type) ¹
Type de câble	CAT5e blindé
Interfaces	
Nombre d'interfaces PROFINET	1
Nombre d'interfaces PROFIBUS	0
Interface	
Matériel	
Nombre de ports	1
Commutateur intégré	Non
RJ-45 (Ethernet)	Oui ; X1
Protocoles	
Contrôleur PROFINET IO	Oui
Périphérique PROFINET IO	Oui
Communication SIMATIC	Oui
Communication IE ouverte	Oui
Serveur Web	Oui
Redondance des supports	Non
Contrôleur PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT	Non
MRP	Non
PROFIenergy	Oui. La CPU S7-1200 ne prend en charge que l'unité PROFIenergy (avec la fonctionnalité I-device).
Démarrage priorisé	Oui (16 périphériques PROFINET max.)
Nombre de périphériques IO raccordables max.	16

Caractéristiques techniques	Description
Nombre de périphériques IO que vous pouvez raccorder pour RT, max.	16
Dont en ligne, max.	16
Nombre de périphériques IO pouvant être activés/désactivés simultanément, max.	8
Temps d'actualisation	La valeur minimale du temps d'actualisation dépend également du composant de communication défini pour PROFINET IO, du nombre de périphériques IO et de la quantité de données utilisateur configurées.
Avec RT	
Horloge de transmission de 1 ms	1 ms à 512 ms
Périphérique PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT pris en charge	Non
MRP pris en charge	Non
PROFenergy	Oui
Appareil partagé	Oui
Nombre de contrôleurs IO avec appareil partagé, max.	2
Communication SIMATIC	
Communication S7, en tant que serveur	Oui
Communication S7, en tant que client	Oui
Données utilisateur par tâche, max.	Voir l'aide en ligne (Communication S7, taille des données utilisateur)
Communication IE ouverte	
TCP/IP	Oui
Longueur de données, max.	8 ko
Prise en charge de plusieurs liaisons passives par port	Oui
ISO-on-TCP (RFC1006)	Oui
Longueur de données, max.	8 ko
UDP	Oui
Longueur de données, max.	1 472 octets
DHCP	Non
SNMP	Oui
DCP	Oui
LLDP	Oui

¹ L'isolation du port Ethernet est conçue pour limiter les risques pendant les pannes de réseau de courte durée avec tensions dangereuses. Elle ne satisfait pas aux exigences en matière de sécurité pour l'isolation des lignes CA de routine.

Tableau A- 33Alimentation

Caractéristiques techniques		CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Plage de tension		85 à 264 V CA	20,4 V CC à 28,8 V CC	
Fréquence de ligne		47 à 63 Hz	--	
Courant d'entrée (charge max.)	CPU uniquement	80 mA à 120 V CA 40 mA à 240 V CA	400 mA à 24 V CC	
	CPU avec tous les accessoires d'extension	240 mA à 120 V CA 120 mA à 240 V CA	1 200 mA à 24 V CC	
Appel de courant (max.)		20 A à 264 V CA	12 A à 28,8 V CC	
I ² t		0,8 A ² s	0,5 A ² s	
Isolation (alimentation d'entrée à logique)		1 500 V CA	Non isolée	
Fuite à la terre, ligne CA à la terre fonctionnelle		0,5 mA max.	--	
Temps de retard (perte d'alimentation)		20 ms à 120 V CA 80 ms à 240 V CA	10 ms à 24 V CC	
Fusible interne, non remplaçable par l'utilisateur		3 A, 250 V, action retardée		

Tableau A- 34Alimentation de capteur

Caractéristiques techniques		CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Plage de tension		20,4 à 28,8 V CC	L+ moins 4 V CC min.	
Courant de sortie nominal (max.)		300 mA (protégé contre les courts-circuits)		
Bruit d'ondulation maximum (<10 MHz)		< 1 V crête à crête	Comme la ligne d'entrée	
Isolation (logique CPU à alimentation de capteur)		Non isolée		

A.5.3 Entrées et sorties TOR

Tableau A- 35 Entrées TOR

Caractéristiques techniques	CPU 1212C AC/DC/Relais, DC/DC/Relais et DC/DC/DC
Nombre d'entrées	8
Type	P/N (CEI type 1 en mode P)
Tension nominale	24 V CC à 4 mA, nominal
Tension continue admise	30 V CC max.
Tension de choc	35 V CC pour 0,5 s
Signal 1 logique (min.)	15 V CC à 2,5 mA
Signal 0 logique (max.)	5 V CC à 1 mA
Isolation (site à logique)	707 V CC (essai de type)
Groupes d'isolation	1
Temps de filtre	microsecondes : 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 millisecondes : 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V CC)	100/80 kHz (Ia.0 à Ia.5) 30/20 kHz (Ia.6 à Ia.7)
Nombre d'entrées simultanément à 1	4 (points non adjacents) à 60 °C horizontal ou 50 °C vertical 8 à 55 °C horizontal ou 45 °C vertical
Longueur de câble (mètres)	500 m blindé, 300 m non blindé, 50 m blindé pour entrées HSC

Tableau A- 36Sorties TOR

Caractéristiques techniques	CPU 1212C AC/DC/Relais et DC/DC/Relais	CPU 1212C DC/DC/DC
Nombre de sorties	6	
Type	Relai, mécanique	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V CC ou 5 à 250 V CA	20,4 à 28,8 V CC
Signal 1 logique à courant max.	--	20 V CC min.
Signal 0 logique avec charge 10 kΩ	--	0,1 V CC max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 µA max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	
Isolation (site à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)	707 V CC (essai de type)
Groupes d'isolation	2	1
Isolation (groupe à groupe)	1 500 V CA ¹	--
Tension de blocage inductive	--	L+ moins 48 V CC, dissipation 1 W
Retard de commutation (Qa.0 à Qa.3)	10 ms max.	1,0 µs max., de 0 à 1 3,0 µs max., de 1 à 0
Retard de commutation (Qa.4 à Qa.5)	10 ms max.	5 µs max., de 0 à 1 20 µs max., de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	--
Fréquence de sortie de trains d'impulsions	Non recommandé ²	100 kHz (Qa.0 à Qa.3) ³ , 2 Hz min. 20 kHz (Qa.4 à Qa.5) ³
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	3 (points non adjacents) à 60 °C horizontal ou 50 °C vertical 6 à 55 °C horizontal ou 45 °C vertical	
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	

¹ L'isolation groupe à groupe du relais sépare la tension de ligne de la TBTS/TBTP et sépare les différentes phases jusqu'à 250 V CA ligne à la terre.

² Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

³ Selon votre câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10 % du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

A.5.4 Entrées analogiques

Tableau A- 37 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V
Plage de dépassement (mot de données)	27649 à 32511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32512 à 32767
Résolution	10 bits
Tension de tenue maximum	35 V CC
Lissage	Aucun, faible, moyen ou fort Reportez-vous au tableau sur la Réponse indicielle (ms) des entrées analogiques de la CPU (Page 1465).
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥ 100 kΩ
Isolation (site à logique)	Aucune
Précision (25 °C / -20 à 60 °C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

A.5.4.1 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 38 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Temps d'échantillonnage	4,17 ms	5 ms	25 ms

A.5.4.2 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 39 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.5.4.3 Plages de mesure des entrées analogiques pour la tension (CPU)

Tableau A- 40 Représentation des entrées analogiques pour la tension (CPU)

Technologie		Plage de mesure de tension	
Décimal	Hexadécimal	0 à 10 V	
32767	7FFF	11,852 V CC	Débordement haut
32512	7F00		
32511	7EFF	11,759 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valeurs négatives		Valeurs négatives impossibles	

A.5.5 Schémas de câblage de la CPU 1212C

Tableau A- 41CPU 1212C AC/DC/Relais (6ES7212-1BE40-0XB0)

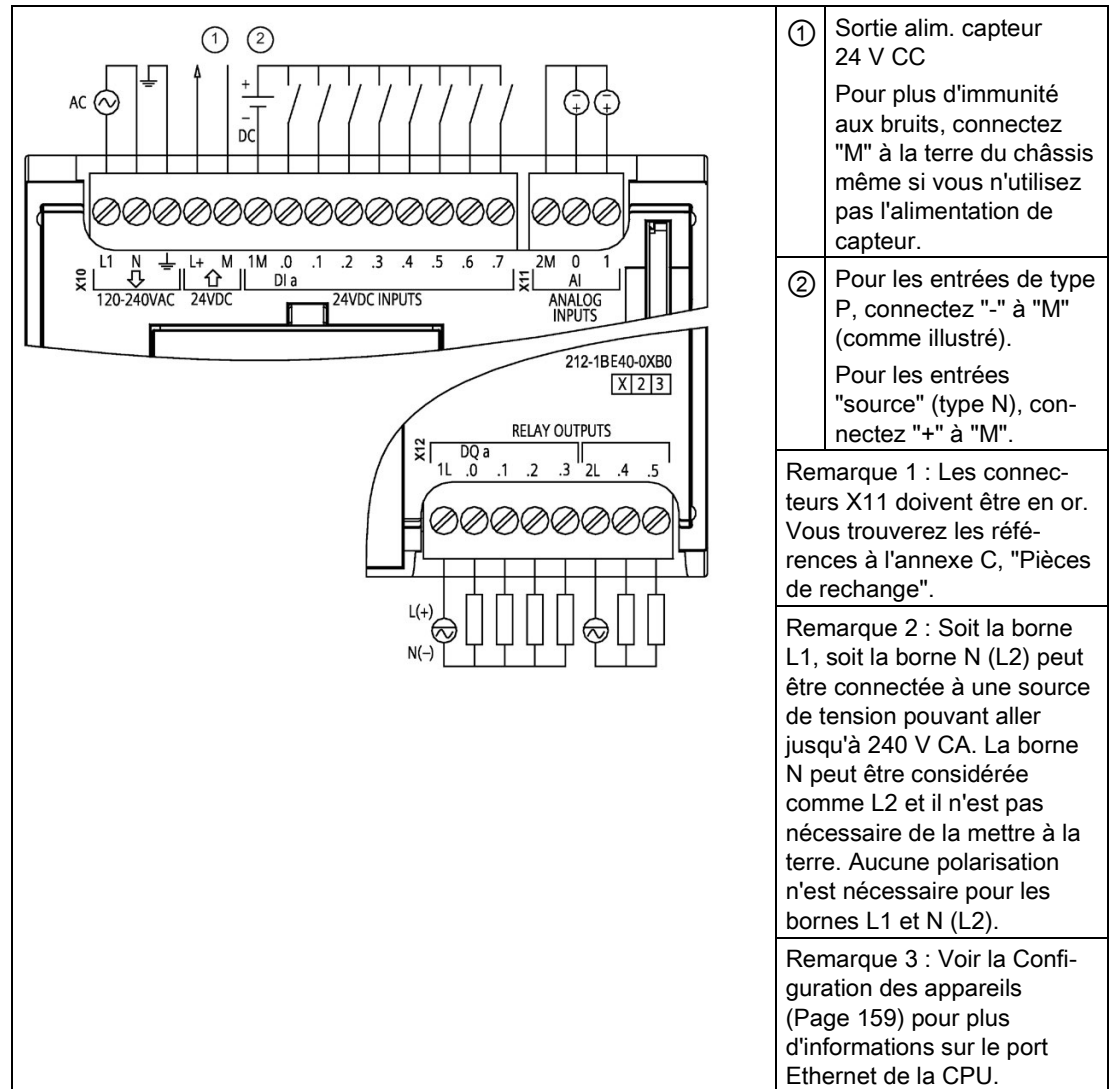


Tableau A- 42Brochage pour la CPU 1212C AC/DC/Relais (6ES7212-1BE40-0XB0)

Broche	X10	X11 (or)	X12
1	L1/120-240 V CA	2 M	1L
2	N/120-240 V CA	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+/Sortie capteur 24 V CC	--	DQ a.2
5	M/Sortie capteur 24 V CC	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Tableau A- 43CPU 1212C DC/DC/Relais (6ES7212-1HE40-0XB0)

The diagram shows the terminal block for the CPU 1212C DC/DC/Relais. It features three main sections: 24VDC INPUTS (terminals L+, M, 1M, .0-.7), ANALOG INPUTS (terminals AI 0, 1, 2M, 0, 1), and RELAY OUTPUTS (terminals DQ a, 1L, .0-.3, 2L, .4, .5). Connections for DC power and ground are shown at the top, with callouts 1 and 2 indicating specific output connections.

① Sortie alim. capteur 24 V CC
Pour plus d'immunité aux perturbations, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.

② Pour les entrées de type P, connectez "-" à "M" (comme illustré).
Pour les entrées "source" (type N), connectez "+" à "M".

Remarque 1 : Les connecteurs X11 doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Remarque 2 : Voir la Configuration des appareils (Page 159) pour plus d'informations sur le port Ethernet de la CPU.

Tableau A- 44Brochage pour la CPU 1212C DC/DC/Relais (6ES7212-1HE40-0XB0)

Broche	X10	X11 (or)	X12
1	L+/24 V CC	2 M	1L
2	M/24 V CC	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+/Sortie capteur 24 V CC	--	DQ a.2
5	M/Sortie capteur 24 V CC	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Tableau A- 45CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)

① Sortie alim. capteur
24 V CC
Pour plus d'immunité aux perturbations, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.

② Pour les entrées de type P, connectez "-" à "M" (comme illustré).
Pour les entrées "source" (type N), connectez "+" à "M".

Remarque 1 : Les connecteurs X11 doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Remarque 2 : Voir la Configuration des appareils (Page 159) pour plus d'informations sur le port Ethernet de la CPU.

Tableau A- 46 Brochage pour la CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)

Broche	X10	X11 (or)	X12
1	L+/24 V CC	2 M	3L+
2	M/24 V CC	AI 0	3M
3	Terre fonctionnelle	AI 1	DQ a.0
4	L+/Sortie capteur 24 V CC	--	DQ a.1
5	M/Sortie capteur 24 V CC	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Remarque

Les entrées analogiques inutilisées doivent être court-circuitées.

A.6 CPU 1214C

A.6.1 Caractéristiques et fonctions générales

Tableau A- 47Généralités

Caractéristiques techniques	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Référence	6ES7214-1BG40-0XB0	6ES7214-1HG40-0XB0	6ES7214-1AG40-0XB0
Dimensions L x H x P (mm)	110 x 100 x 75		
Poids avec emballage	475 grammes	435 grammes	415 grammes
Dissipation de courant	14 W	12 W	
Courant disponible (bus SM et CM)	1 600 mA max. (5 V CC)		
Courant disponible (24 V CC)	400 mA max. (alimentation de capteur)		
Consommation de courant entrées TOR (24 V CC)	4 mA / entrée utilisée		

Tableau A- 48 Fonctions de la CPU

Caractéristiques techniques		Description
Mémoire utilisateur (Reportez-vous à "Caractéristiques techniques générales", (Page 1429) "Conservation de la mémoire de la CPU interne".)	de travail	100 Ko
	de chargement	4 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées		14 entrées/10 sorties
E/S analogiques intégrées		2 entrées
Taille de la mémoire image		1024 octets d'entrées (I) / 1024 octets de sorties (Q)
Mémentos (M)		8192 octets
Mémoire temporaire (locale)		<ul style="list-style-type: none"> • 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus) • 6 Ko pour chacun des autres niveaux de priorité d'alarme (FB et FC inclus)
Modules d'entrées-sorties pour extension		8 SM max.
SB, CB, BB pour extension		1 max.
Modules de communication pour extension		3 CM max.
Compteurs rapides		<p>Jusqu'à 6 configurés pour utiliser toute entrée intégrée ou du SB. Voir "Affectation d'entrée matérielle" (Page 599) pour la CPU 1214C : affectation d'adresses HSC par défaut.</p> <ul style="list-style-type: none"> • 100/180 kHz (Ia.0 à Ia.5) • 30/120 kHz (Ia.6 à Ib.5)
Sorties d'impulsions ²		<p>Jusqu'à 4 configurées pour utiliser toute sortie intégrée ou du SB.</p> <ul style="list-style-type: none"> • 100 kHz (Qa.0 à Qa.3) • 20 kHz (Qa.4 à Qb.1)
Entrées de capture d'impulsions		14
Alarmes temporisées		4 au total avec résolution de 1 ms
Alarmes cycliques		4 au total avec résolution de 1 ms
Alarmes sur front		12 sur front montant et 12 sur front descendant (16 et 16 avec Signal Board optionnel)
Carte mémoire		Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel		+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel		20 jours typ./12 jours min. à 40 °C (supercondensateur sans maintenance)

¹ La vitesse plus faible s'applique lorsque le HSC est configuré pour le fonctionnement en quadrature de phase.

² Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Tableau A- 49 Performances

Type d'instruction		Vitesse d'exécution	
		Adressage direct (I, Q et M)	Accès au DB
Instructions booléennes		0,08 µs/instruction	
Déplacement	Move_Bool	0,3 µs/instruction	1,17 µs/instruction
	Move_Word	0,137 µs/instruction	1,0 µs/instruction
	Move_Real	0,72 µs/instruction	1,0 µs/instruction
Opérations arithmétiques sur réels	Add_Real	1,48 µs/instruction	1,78 µs/instruction

Remarque

De nombreuses variables affectent les temps mesurés. Les temps de performance ci-dessus concernent les instructions les plus rapides dans cette catégorie et les programmes exempts d'erreur.

A.6.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1214C

Tableau A- 50 Blocs, temporisations et compteurs pris en charge par la CPU 1214C

Élément	Description	
Blocs	Type	OB, FB, FC, DB
	Taille	64 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	FB et FC : 1 à 65535 (FB 1 à FB 65535, par exemple) DB : 1 à 59999
	Profondeur d'imbrication	16 en cas d'appel depuis l'OB de cycle de programme ou de démarrage 6 en cas d'appel depuis un OB d'alarme associé à un événement ¹
	Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.
OB	Cycle de programme	Multiple
	Mise en route	Multiple
	Alarmes temporisées	4 (1 par événement)
	Alarmes cycliques	4 (1 par événement)
	Alarmes de processus	50 (1 par événement)
	Alarmes d'erreur de temps	1
	Alarmes de diagnostic	1
	Débrochage/enfichage de modules	1
	Défaillance du châssis ou de la station	1

Élément	Description	
	Heure	Multiple
	Etat	1
	Mettre à jour	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Temporisations	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> • SInt, USInt : 3 octets • Int, UInt : 6 octets • DInt, UDInt : 12 octets

¹ Les programmes de sécurité utilisent deux niveaux d'imbrication. Le programme utilisateur a donc une profondeur d'imbrication de quatre dans les programmes de sécurité.

Tableau A- 51Communication

Caractéristiques techniques	Description
Nombre de ports	1
Type	Ethernet
Appareil IHM	4
Console de programmation (PG)	1
Liaisons	<ul style="list-style-type: none"> • 8 liaisons pour la communication ouverte Open User Communication (active ou passive) : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV • 8 liaisons CPU à CPU (client ou serveur) pour les données GET/PUT • 6 liaisons pour l'affectation dynamique à GET/PUT ou Open User Communication
Débits	10/100 Mb/s
Isolation (signal externe à logique)	Transformateur isolé, 1 500 V CA (essai de type) ¹
Type de câble	CAT5e blindé
Interfaces	
Nombre d'interfaces PROFINET	1
Nombre d'interfaces PROFIBUS	0

Caractéristiques techniques	Description
Interface	
Matériel	
Nombre de ports	1
Commutateur intégré	Non
RJ-45 (Ethernet)	Oui ; X1
Protocoles	
Contrôleur PROFINET IO	Oui
Périphérique PROFINET IO	Oui
Communication SIMATIC	Oui
Communication IE ouverte	Oui
Serveur Web	Oui
Redondance des supports	Non
Contrôleur PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT	Non
MRP	Non
PROFenergy	Oui. La CPU S7-1200 ne prend en charge que l'unité PROFenergy (avec la fonctionnalité I-device).
Démarrage priorisé	Oui (16 périphériques PROFINET max.)
Nombre de périphériques IO raccordables max.	16
Nombre de périphériques IO que vous pouvez raccorder pour RT, max.	16
Dont en ligne, max.	16
Nombre de périphériques IO pouvant être activés/désactivés simultanément, max.	8
Temps d'actualisation	La valeur minimale du temps d'actualisation dépend également du composant de communication défini pour PROFINET IO, du nombre de périphériques IO et de la quantité de données utilisateur configurées.
Avec RT	
Horloge de transmission de 1 ms	1 ms à 512 ms
Périphérique PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT pris en charge	Non
MRP pris en charge	Non

Caractéristiques techniques		Description
PROFenergy		Oui
Appareil partagé		Oui
Nombre de contrôleurs IO avec appareil partagé, max.		2
Communication SIMATIC		
Communication S7, en tant que serveur		Oui
Communication S7, en tant que client		Oui
Données utilisateur par tâche, max.		Voir l'aide en ligne (Communication S7, taille des données utilisateur)
Communication IE ouverte		
TCP/IP		Oui
	Longueur de données, max.	8 ko
	Prise en charge de plusieurs liaisons passives par port	Oui
ISO-on-TCP (RFC1006)		Oui
	Longueur de données, max.	8 ko
UDP		Oui
	Longueur de données, max.	1 472 octets
DHCP		Non
SNMP		Oui
DCP		Oui
LLDP		Oui

¹ L'isolation du port Ethernet est conçue pour limiter les risques pendant les pannes de réseau de courte durée avec tensions dangereuses. Elle ne satisfait pas aux exigences en matière de sécurité pour l'isolation des lignes CA de routine.

Tableau A- 52Alimentation

Caractéristiques techniques		CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Plage de tension		85 à 264 V CA	20,4 V CC à 28,8 V CC	
Fréquence de ligne		47 à 63 Hz	--	
Courant d'entrée (charge max.)	CPU uniquement	100 mA à 120 V CA 50 mA à 240 V CA	500 mA à 24 V CC	
	CPU avec tous les accessoires d'extension	300 mA à 120 V CA 150 mA à 240 V CA	1 500 mA à 24 V CC	
Appel de courant (max.)		20 A à 264 V CA	12 A à 28,8 V CC	
I ² t		0,8 A ² s	0,5 A ² s	
Isolation (alimentation d'entrée à logique)		1 500 V CA	Non isolée	
Fuite à la terre, ligne CA à la terre fonctionnelle		0,5 mA max.	-	
Temps de retard (perte d'alimentation)		20 ms à 120 V CA 80 ms à 240 V CA	10 ms à 24 V CC	
Fusible interne, non remplaçable par l'utilisateur		3 A, 250 V, action retardée		

Tableau A- 53Alimentation de capteur

Caractéristiques techniques	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Plage de tension	20,4 à 28,8 V CC	L+ moins 4 V CC min.	
Courant de sortie nominal (max.)	400 mA (protégé contre les courts-circuits)		
Bruit d'ondulation maximum (< 10 MHz)	< 1 V crête à crête	Comme la ligne d'entrée	
Isolation (logique CPU à alimentation de capteur)	Non isolée		

A.6.3 Entrées et sorties TOR

Tableau A- 54Entrées TOR

Caractéristiques techniques	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Nombre d'entrées	14		
Type	P/N (CEI type 1 en mode P)		
Tension assignée	24 V CC à 4 mA, nominal		
Tension continue admise	30 V CC max.		
Tension de choc	35 V CC pour 0,5 s		
Signal 1 logique (min.)	15 V CC à 2,5 mA		
Signal 0 logique (max.)	5 V CC à 1 mA		
Isolation (site à logique)	707 V CC (essai de type)		
Groupes d'isolation	1		
Temps de filtre	paramètres us : 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 paramètres ms : 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0		
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V CC)	100/80 kHz (Ia.0 à Ia.5) 30/20 kHz (Ia.6 à Ib.5)		
Nombre d'entrées simultanément à 1	<ul style="list-style-type: none"> • 7 (points non adjacents) à 60 °C horizontal ou 50 °C vertical • 14 à 55 °C horizontal ou 45 °C vertical 		
Longueur de câble (mètres)	500 m blindé, 300 m non blindé, 50 m blindé pour entrées HSC		

Tableau A- 55Sorties TOR

Caractéristiques techniques	CPU 1214C AC/DC/Relais et DC/DC/Relais	CPU 1214C DC/DC/DC
Nombre de sorties	10	
Type	Relai, mécanique	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V CC ou 5 à 250 V CA	20,4 à 28,8 V CC
Signal 1 logique à courant max.	--	20 V CC min.
Signal 0 logique avec charge 10 kΩ	--	0,1 V CC max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 µA max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	
Isolation (site à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)	707 V CC (essai de type)
Groupes d'isolation	2	1
Isolation (groupe à groupe)	1 500 V CA ¹	--
Tension de blocage inductive	--	L+ moins 48 V CC, dissipation 1 W
Retard de commutation (Qa.0 à Qa.3)	10 ms max.	1,0 µs max., de 0 à 1 3,0 µs max., de 1 à 0
Retard de commutation (Qa.4 à Qb.1)	10 ms max.	5 µs max., de 0 à 1 20 µs max., de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	--
Fréquence de sortie de trains d'impulsions	Non recommandé ²	100 kHz (Qa.0 à Qa.3) ³ , 2 Hz min. 20 kHz (Qa.4 à Qb.1) ³
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> • 5 (points non adjacents) à 60 °C horizontal ou 50 °C vertical • 10 à 55 °C horizontal ou 45 °C vertical 	
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	

¹ L'isolation groupe à groupe du relais sépare la tension de ligne de la TBTS/TBTP et sépare les différentes phases jusqu'à 250 V CA ligne à la terre.

² Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

³ Selon votre câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10 % du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

A.6.4 Entrées analogiques

Tableau A- 56 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V
Plage de dépassement (mot de données)	27649 à 32511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32512 à 32767
Résolution	10 bits
Tension de tenue maximum	35 V CC
Lissage	Aucun, faible, moyen ou fort Reportez-vous au tableau sur la Réponse indicielle (ms) des entrées analogiques de la CPU (Page 1479).
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥ 100 kΩ
Isolation (site à logique)	Aucune
Précision (25 °C / -20 à 60 °C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

A.6.4.1 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 57 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms
Temps d'échantillonnage	4,17 ms	5 ms	25 ms

A.6.4.2 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 58 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.6.4.3 Plages de mesure des entrées analogiques pour la tension (CPU)

Tableau A- 59 Représentation des entrées analogiques pour la tension (CPU)

Technologie		Plage de mesure de tension	
Décimal	Hexadécimal	0 à 10 V	
32767	7FFF	11,852 V CC	Débordement haut
32512	7F00		
32511	7EFF	11,759 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valeurs négatives		Valeurs négatives impossibles	

A.6.5 Schémas de câblage de la CPU 1214C

Tableau A- 60CPU 1214C AC/DC/Relais (6ES7214-1BG40-0XB0)

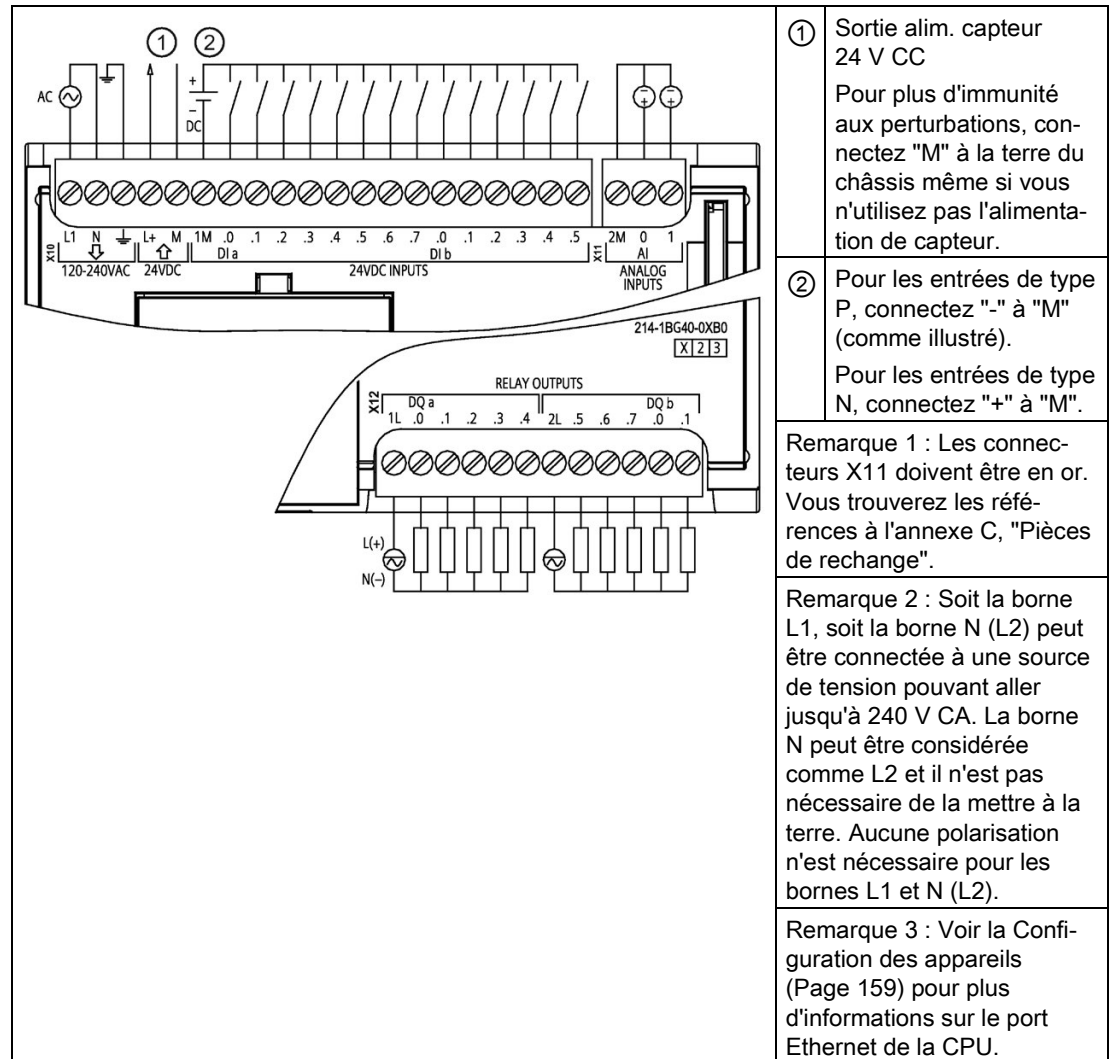


Tableau A- 61 Brochage pour la CPU 1214C AC/DC/Relais (6ES7214-1BG40-0XB0)

Broche	X10	X11 (or)	X12
1	L1/120-240 V CA	2 M	1L
2	N/120-240 V CA	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+/Sortie capteur 24 V CC	--	DQ a.2
5	M/Sortie capteur 24 V CC	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

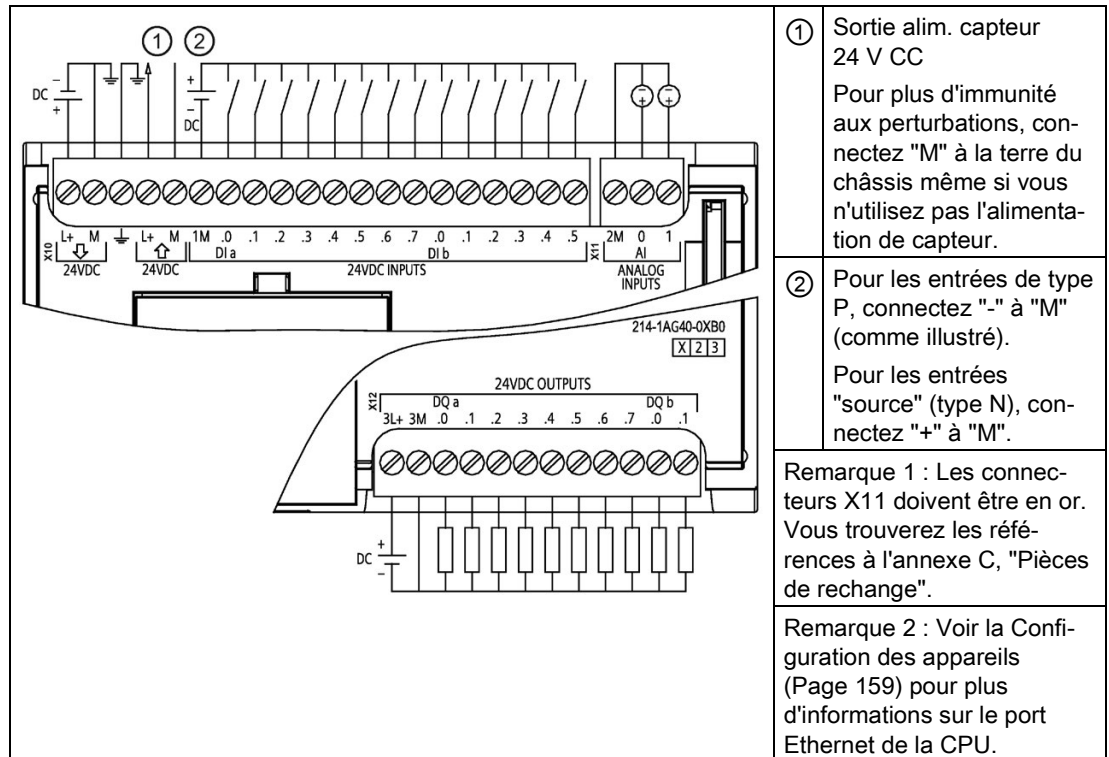
Tableau A- 62CPU 1214C DC/DC/Relais (6ES7214-1HG40-0XB0)

<p>The diagram illustrates the terminal block of the 6ES7214-1HG40-0XB0 CPU module. It shows connections for DC power (24VDC), 24VDC inputs (DI a, DI b), analog inputs (AI), and relay outputs (DQ a, DQ b). The relay outputs are connected to a load (L+) and a neutral (N-). The diagram is labeled with '214-1HG40-0XB0' and 'X123'.</p>	<p>① Sortie alim. capteur 24 V CC</p> <p>Pour plus d'immunité aux perturbations, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.</p>
	<p>② Pour les entrées de type P, connectez "-" à "M" (comme illustré).</p> <p>Pour les entrées "source" (type N), connectez "+" à "M".</p>
	<p>Remarque 1 : Les connecteurs X11 doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".</p> <p>Remarque 2 : Voir la Configuration des appareils (Page 159) pour plus d'informations sur le port Ethernet de la CPU.</p>

Tableau A- 63 Brochage pour la CPU 1214C DC/DC/Relais (6ES7214-1HG40-0XB0)

Broche	X10	X11 (or)	X12
1	L+/24 V CC	2 M	1L
2	M/24 V CC	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+/Sortie capteur 24 V CC	--	DQ a.2
5	M/Sortie capteur 24 V CC	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tableau A- 64CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)



① Sortie alim. capteur 24 V CC
 Pour plus d'immunité aux perturbations, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.

② Pour les entrées de type P, connectez "-" à "M" (comme illustré).
 Pour les entrées "source" (type N), connectez "+" à "M".

Remarque 1 : Les connecteurs X11 doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Remarque 2 : Voir la Configuration des appareils (Page 159) pour plus d'informations sur le port Ethernet de la CPU.

Tableau A- 65 Brochage pour la CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)

Broche	X10	X11 (or)	X12
1	L+/24 V CC	2 M	3L+
2	M/24 V CC	AI 0	3M
3	Terre fonctionnelle	AI 1	DQ a.0
4	L+/Sortie capteur 24 V CC	--	DQ a.1
5	M/Sortie capteur 24 V CC	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	-
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Remarque

Les entrées analogiques inutilisées doivent être court-circuitées.

A.7 CPU 1215C

A.7.1 Caractéristiques et fonctions générales

Tableau A- 66 Généralités

Caractéristiques techniques	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Référence	6ES7215-1BG40-0XB0	6ES7215-1HG40-0XB0	6ES7215-1AG40-0XB0
Dimensions L x H x P (mm)	130 x 100 x 75		
Poids avec emballage	585 grammes	550 grammes	520 grammes
Dissipation de courant	14 W	12 W	
Courant disponible (bus SM et CM)	1 600 mA max. (5 V CC)		
Courant disponible (24 V CC)	400 mA max. (alimentation de capteur)		
Consommation de courant entrées TOR (24 V CC)	4 mA / entrée utilisée		

Tableau A- 67 Fonctions de la CPU

Caractéristiques techniques	Description	
Mémoire utilisateur (Reportez-vous à "Caractéristiques techniques générales (Page 1429)", "Conservation de la mémoire de la CPU interne".)	de travail	125 Ko
	de chargement	4 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées	14 entrées/10 sorties	
E/S analogiques intégrées	2 entrées/2 sorties	
Taille de la mémoire image	1024 octets d'entrées (I) / 1024 octets de sorties (Q)	
Mémentos (M)	8192 octets	
Mémoire temporaire (locale)	<ul style="list-style-type: none"> 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus) 6 Ko pour chacun des autres niveaux de priorité d'alarme (FB et FC inclus) 	
Modules d'entrées-sorties pour extension	8 SM max.	
SB, CB, BB pour extension	1 max.	
Modules de communication pour extension	3 CM max.	

Caractéristiques techniques	Description
Compteurs rapides	Jusqu'à 6 configurés pour utiliser toute entrée intégrée ou du SB. Voir "Affectation d'entrée matérielle" (Page 599) pour la CPU 1215C : affectation d'adresses HSC par défaut. <ul style="list-style-type: none"> • 100/180 kHz (Ia.0 à Ia.5) • 30/120 kHz (Ia.6 à Ib.5)
Sorties d'impulsions ²	Jusqu'à 4 configurées pour utiliser toute sortie intégrée ou du SB <ul style="list-style-type: none"> • 100 kHz (Qa.0 à Qa.3) • 20 kHz (Qa.4 à Qb.1)
Entrées de capture d'impulsions	14
Alarmes temporisées	4 au total avec résolution de 1 ms
Alarmes cycliques	4 au total avec résolution de 1 ms
Alarmes sur front	12 sur front montant et 12 sur front descendant (16 et 16 avec Signal Board optionnel)
Carte mémoire	Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel	+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel	20 jours typ./12 jours min. à 40 °C (supercondensateur sans maintenance)

¹ La vitesse plus faible s'applique lorsque le HSC est configuré pour le fonctionnement en quadrature de phase.

² Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Tableau A- 68 Performances

Type d'instruction		Vitesse d'exécution	
		Adressage direct (I, Q et M)	Accès au DB
Instructions booléennes		0,08 µs/instruction	
Déplacement	Move_Bool	0,3 µs/instruction	1,17 µs/instruction
	Move_Word	0,137 µs/instruction	1,0 µs/instruction
	Move_Real	0,72 µs/instruction	1,0 µs/instruction
Opérations arithmétiques sur réels	Add_Real	1,48 µs/instruction	1,78 µs/instruction

Remarque

De nombreuses variables affectent les temps mesurés. Les temps de performance ci-dessus concernent les instructions les plus rapides dans cette catégorie et les programmes exempts d'erreur.

A.7.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1215C

Tableau A- 69Blocs, temporisations et compteurs pris en charge par la CPU 1215C

Elément		Description
Blocs	Type	OB, FB, FC, DB
	Taille	64 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	FB et FC : 1 à 65535 (FB 1 à FB 65535, par exemple) DB : 1 à 59999
	Profondeur d'imbrication	16 en cas d'appel depuis l'OB de cycle de programme ou de démarrage 6 en cas d'appel depuis un OB d'alarme associé à un événement ¹
	Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.
OB	Cycle de programme	Multiple
	Mise en route	Multiple
	Alarmes temporisées	4 (1 par événement)
	Alarmes cycliques	4 (1 par événement)
	Alarmes de processus	50 (1 par événement)
	Alarmes d'erreur de temps	1
	Alarmes de diagnostic	1
	Débrochage ou enfichage de modules	1
	Défaillance du châssis ou de la station	1
	Heure	Multiple
	Etat	1
	Mettre à jour	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
MC-PreServo	1	
MC-PostServo	1	
Temporisations	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> • SInt, USInt : 3 octets • Int, UInt : 6 octets • DInt, UDInt : 12 octets

¹ Les programmes de sécurité utilisent deux niveaux d'imbrication. Le programme utilisateur a donc une profondeur d'imbrication de quatre dans les programmes de sécurité.

Tableau A- 70Communication

Caractéristiques techniques	Description
Nombre de ports	2
Type	Ethernet
Appareil IHM	4
Console de programmation (PG)	1
Liaisons	<ul style="list-style-type: none"> 8 liaisons pour la communication ouverte Open User Communication (active ou passive) : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV 8 liaisons CPU à CPU (client ou serveur) pour les données GET/PUT 6 liaisons pour l'affectation dynamique à GET/PUT ou Open User Communication
Débits	10/100 Mb/s
Isolation (signal externe à logique)	Transformateur isolé, 1 500 V CA (essai de type) ¹
Type de câble	CAT5e blindé
Interfaces	
Nombre d'interfaces PROFINET	1
Nombre d'interfaces PROFIBUS	0
Interface	
Matériel	
Nombre de ports	2
Commutateur intégré	Oui
RJ-45 (Ethernet)	Oui ; X1
Protocoles	
Contrôleur PROFINET IO	Oui
Périphérique PROFINET IO	Oui
Communication SIMATIC	Oui
Communication IE ouverte	Oui
Serveur Web	Oui
Redondance des supports	Oui
Contrôleur PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT	Non
MRP	Oui en tant que client MRP
PROFIenergy	Oui. La CPU S7-1200 ne prend en charge que l'unité PROFIenergy (avec la fonctionnalité I-device).
Démarrage priorisé	Oui (16 périphériques PROFINET max.)
Nombre de périphériques IO raccordables max.	16

Caractéristiques techniques	Description
Nombre de périphériques IO que vous pouvez raccorder pour RT, max.	16
Dont en ligne, max.	16
Nombre de périphériques IO pouvant être activés/désactivés simultanément, max.	8
Temps d'actualisation	La valeur minimale du temps d'actualisation dépend également du composant de communication défini pour PROFINET IO, du nombre de périphériques IO et de la quantité de données utilisateur configurées.
Avec RT	
Horloge de transmission de 1 ms	1 ms à 512 ms
Périphérique PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT pris en charge	Non
MRP pris en charge	Oui
PROFenergy	Oui
Appareil partagé	Oui
Nombre de contrôleurs IO avec appareil partagé, max.	2
Communication SIMATIC	
Communication S7, en tant que serveur	Oui
Communication S7, en tant que client	Oui
Données utilisateur par tâche, max.	Voir l'aide en ligne (Communication S7, taille des données utilisateur)
Communication IE ouverte	
TCP/IP	Oui
Longueur de données, max.	8 ko
Prise en charge de plusieurs liaisons passives par port	Oui
ISO-on-TCP (RFC1006)	Oui
Longueur de données, max.	8 ko
UDP	Oui
Longueur de données, max.	1 472 octets
DHCP	Non
SNMP	Oui
DGP	Oui
LLDP	Oui

¹ L'isolation du port Ethernet est conçue pour limiter les risques pendant les pannes de réseau de courte durée avec tensions dangereuses. Elle ne satisfait pas aux exigences en matière de sécurité pour l'isolation des lignes CA de routine.

Tableau A- 71Alimentation

Caractéristiques techniques		CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Plage de tension		85 à 264 V CA	20,4 V CC à 28,8 V CC	
Fréquence de ligne		47 à 63 Hz	--	
Courant d'en- trée (charge max.)	CPU uniquement	100 mA à 120 V CA 50 mA à 240 V CA	500 mA à 24 V CC	
	CPU avec tous les ac- cessoires d'extension	300 mA à 120 V CA 150 mA à 240 V CA	1 500 mA à 24 V CC	
Appel de courant (max.)		20 A à 264 V CA	12 A à 28,8 V CC	
I ² t		0,8 A ² s	0,5 A ² s	
Isolation (alimentation d'entrée à logique)		1 500 V CA	Non isolée	
Fuite à la terre, ligne CA à la terre fon- ctionnelle		0,5 mA max.	-	
Temps de retard (perte d'alimentation)		20 ms à 120 V CA 80 ms à 240 V CA	10 ms à 24 V CC	
Fusible interne, non remplaçable par l'utilisateur		3 A, 250 V, action retardée		

Tableau A- 72Alimentation de capteur

Caractéristiques techniques		CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Plage de tension		20,4 à 28,8 V CC	L+ moins 4 V CC min.	
Courant de sortie nominal (max.)		400 mA (protégé contre les courts-circuits)		
Bruit d'ondulation maximum (< 10 MHz)		< 1 V crête à crête	Comme la ligne d'entrée	
Isolation (logique CPU à alimentation de capteur)		Non isolée		

A.7.3 Entrées et sorties TOR

Tableau A- 73 Entrées TOR

Caractéristiques techniques	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Nombre d'entrées	14		
Type	P/N (CEI type 1 en mode P)		
Tension assignée	24 V CC à 4 mA, nominal		
Tension continue admise	30 V CC max.		
Tension de choc	35 V CC pour 0,5 s		
Signal 1 logique (min.)	15 V CC à 2,5 mA		
Signal 0 logique (max.)	5 V CC à 1 mA		
Isolation (site à logique)	707 V CC (essai de type)		
Groupes d'isolation	1		
Temps de filtre	microsecondes : 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 millisecondes : 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0		
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V CC)	100/80 kHz (Ia.0 à Ia.5) 30/20 kHz (Ia.6 à Ib.5)		
Nombre d'entrées simultanément à 1	<ul style="list-style-type: none"> • 7 (points non adjacents) à 60 °C horizontal ou 50 °C vertical • 14 à 55 °C horizontal ou 45 °C vertical 		
Longueur de câble (mètres)	500 m blindé, 300 m non blindé, 50 m blindé pour entrées HSC		

Tableau A- 74Sorties TOR

Caractéristiques techniques	CPU 1215C AC/DC/Relais et CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Nombre de sorties	10	
Type	Relai, mécanique	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V CC ou 5 à 250 V CA	20,4 à 28,8 V CC
Signal 1 logique à courant max.	--	20 V CC min.
Signal 0 logique avec charge 10 kΩ	--	0,1 V CC max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 µA max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	
Isolation (site à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)	707 V CC (essai de type)
Groupes d'isolation	2	1
Isolation (groupe à groupe)	1 500 V CA ¹	--
Tension de blocage inductive	--	L+ moins 48 V CC, dissipation 1 W
Retard de commutation (Qa.0 à Qa.3)	10 ms max.	1,0 µs max., de 0 à 1 3,0 µs max., de 1 à 0
Retard de commutation (Qa.4 à Qb.1)	10 ms max.	5 µs max., de 0 à 1 20 µs max., de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	--
Fréquence de sortie de trains d'impulsions	Non recommandé ²	100 kHz (Qa.0 à Qa.3) ³ , 2 Hz min. 20 kHz (Qa.4 à Qb.1) ³
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> • 5 (points non adjacents) à 60 °C horizontal ou 50 °C vertical • 10 à 55 °C horizontal ou 45 °C vertical 	
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	

¹ L'isolation groupe à groupe du relai sépare la tension de ligne de la TBTS/TBTP et sépare les différentes phases jusqu'à 250 V CA ligne à la terre.

² Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

³ Selon votre câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10 % du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

A.7.4 Entrées et sorties analogiques

Tableau A- 75 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V
Plage de dépassement (mot de données)	27649 à 32511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32512 à 32767
Résolution	10 bits
Tension de tenue maximum	35 V CC
Lissage	Aucun, faible, moyen ou fort Reportez-vous au tableau sur la Réponse indicielle (ms) des entrées analogiques de la CPU (Page 1495).
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥ 100 kΩ
Isolation (site à logique)	Aucune
Précision (25 °C / -20 à 60 °C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

A.7.4.1 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 76 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms
Temps d'échantillonnage	4,17 ms	5 ms	25 ms

A.7.4.2 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 77 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.7.4.3 Plages de mesure des entrées analogiques pour la tension (CPU)

Tableau A- 78 Représentation des entrées analogiques pour la tension (CPU)

Technologie		Plage de mesure de tension	
Décimal	Hexadécimal	0 à 10 V	
32767	7FFF	11,852 V CC	Débordement haut
32512	7F00		
32511	7EFF	11,759 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valeurs négatives		Valeurs négatives impossibles	

A.7.4.4 Caractéristiques des sorties analogiques

Tableau A- 79Sorties analogiques

Caractéristiques techniques	Description
Nombre de sorties	2
Type	Courant
Plage pleine échelle	0 à 20 mA
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	20,01 à 23,52 mA
Plage de dépassement (mot de données)	27649 à 32511
Plage de débordement haut	voir la note ¹
Plage de débordement haut (mot de données)	32512 à 32767
Résolution	10 bits
Impédance de sortie	≤500 Ω max.
Isolation (site à logique)	Aucune
Précision (25 °C / - 20 à 60 °C)	3,0% / 3,5% de la pleine échelle
Temps d'établissement	2 ms
Longueur de câble (mètres)	100 m, paire torsadée blindée

¹ En cas de débordement haut, les sorties analogiques se comportent conformément aux paramétrages des propriétés de configuration d'appareil. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : "Appliquer valeur de remplacement" ou "Conserver dernière valeur".

Tableau A- 80Représentation de sorties analogiques pour courant (CPU 1215C et CPU 1217C)

Système		Plage de sortie du courant	
Décimal	Hexadécimal	0 mA à 20 mA	
32767	7FFF	Voir note 1	Débordement haut
32512	7F00	Voir note 1	
32511	7EFF	23,52 mA	Plage de dépassement haut
27649	6C01		
27648	6C00	20 mA	Plage nominale
20736	5100	15 mA	
34	22	0,0247 mA	
0	0	0 mA	
Valeurs négatives		Valeurs négatives impossibles	

¹ En cas de débordement haut, les sorties analogiques se comportent conformément aux paramétrages des propriétés de configuration d'appareil. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : "Appliquer valeur de remplacement" ou "Conserver dernière valeur".

A.7.5 Schémas de câblage de la CPU 1215C

Tableau A- 81CPU 1215C AC/DC/Relais (6ES7215-1BG40-0XB0)

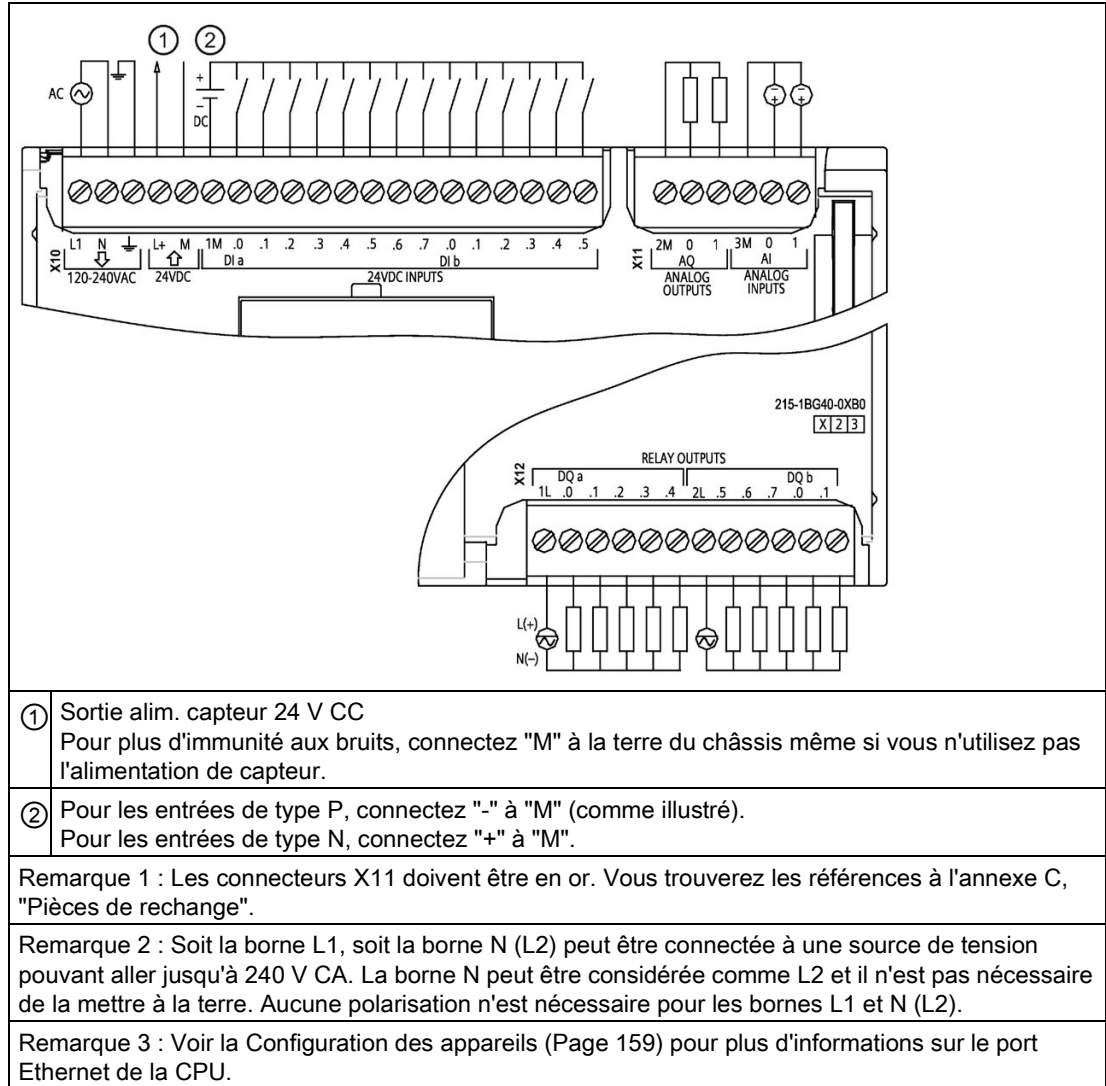


Tableau A- 82Brochage pour la CPU 1215C AC/DC/Relais (6ES7215-1BG40-0XB0)

Broche	X10	X11 (or)	X12
1	L1/120-240 V CA	2 M	1L
2	N/120 - 240 V CA	AQ 0	DQ a.0
3	Terre fonctionnelle	AQ 1	DQ a.1
4	L+/Sortie capteur 24 V CC	3M	DQ a.2
5	M/Sortie capteur 24 V CC	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tableau A- 83CPU 1215C DC/DC/Relais (6ES7215-1HG40-0XB0)

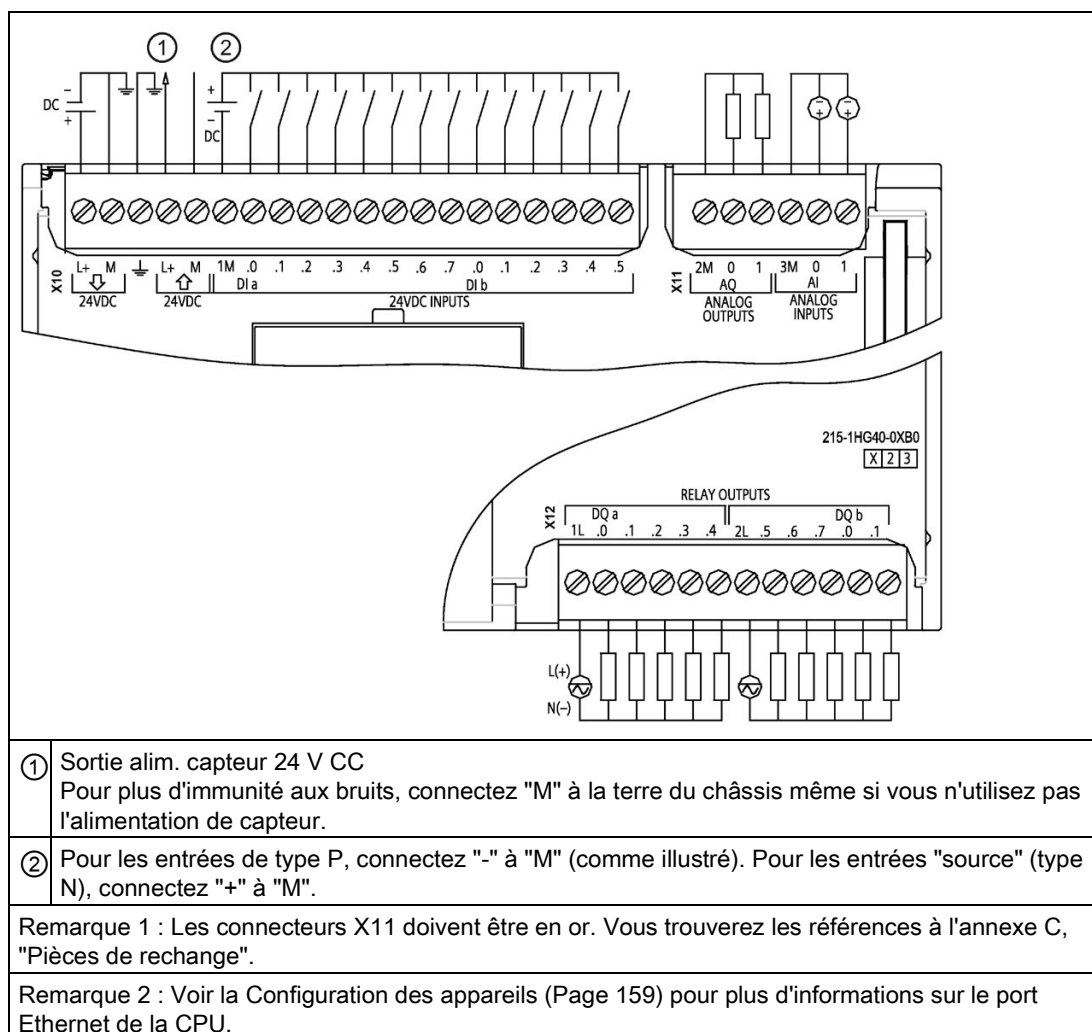


Tableau A- 84Brochage pour la CPU 1215C DC/DC/Relais (6ES7215-1HG40-0XB0)

Broche	X10	X11 (or)	X12
1	L+/24 V CC	2 M	1L
2	M/24 V CC	AQ 0	DQ a.0
3	Terre fonctionnelle	AQ 1	DQ a.1
4	L+/Sortie capteur 24 V CC	3M	DQ a.2
5	M/Sortie capteur 24 V CC	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tableau A- 85CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)

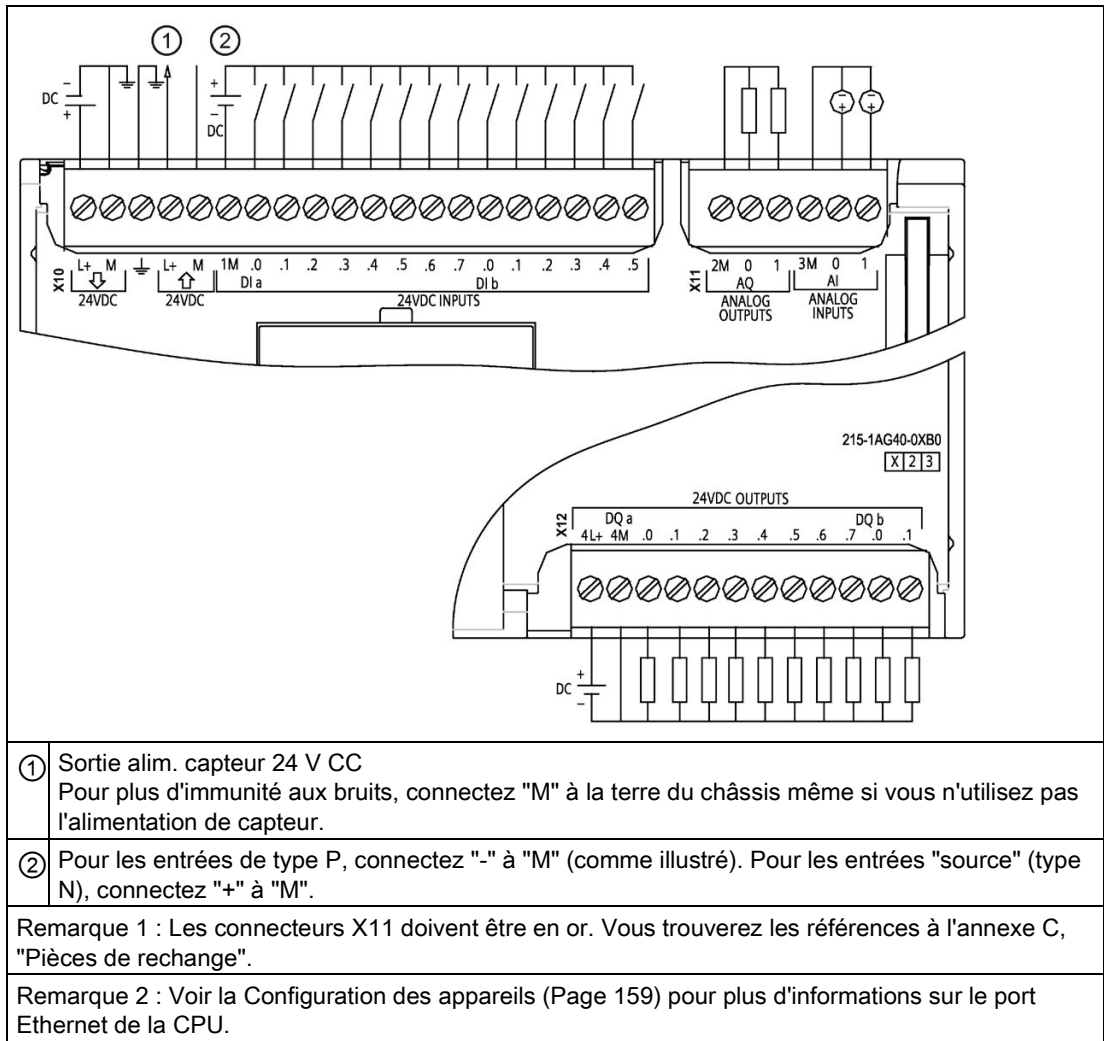


Tableau A- 86Brochage pour la CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)

Broche	X10	X11 (or)	X12
1	L1/24 V CC	2 M	4L+
2	M/24 V CC	AQ 0	4M
3	Terre fonctionnelle	AQ 1	DQ a.0
4	L+/Sortie capteur 24 V CC	3M	DQ a.1
5	M/Sortie capteur 24 V CC	AI 0	DQ a.2
6	1M	AI 1	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Remarque

Les entrées analogiques inutilisées doivent être court-circuitées.

A.8 CPU 1217C

A.8.1 Caractéristiques et fonctions générales

Tableau A- 87Généralités

Caractéristiques techniques	CPU 1217C DC/DC/DC
Référence	6ES7217-1AG40-0XB0
Dimensions L x H x P (mm)	150 x 100 x 75
Poids avec emballage	530 grammes
Dissipation de courant	12 W
Courant disponible (bus SM et CM)	1 600 mA max. (5 V CC)
Courant disponible (24 V CC)	400 mA max. (alimentation de capteur)
Consommation de courant entrées TOR (24 V CC)	4 mA / entrée utilisée

Tableau A- 88Fonctions de la CPU

Caractéristiques techniques	Description	
Mémoire utilisateur (Reportez-vous à "Caractéristiques techniques générales (Page 1429)", Conservation de la mémoire de la CPU interne".)	de travail	150 Ko
	de chargement	4 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées	14 entrées/10 sorties	
E/S analogiques intégrées	2 entrées/2 sorties	
Taille de la mémoire image	1024 octets d'entrées (I) / 1024 octets de sorties (Q)	
Mémentos (M)	8192 octets	
Mémoire temporaire (locale)	<ul style="list-style-type: none"> 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus) 6 Ko pour chacun des autres niveaux de priorité d'alarme (FB et FC inclus) 	
Modules d'entrées-sorties pour extension	8 SM max.	
SB, CB, BB pour extension	1 max.	
Modules de communication pour extension	3 CM max.	

Caractéristiques techniques	Description
Compteurs rapides	Jusqu'à 6 configurés pour utiliser toute entrée intégrée ou du SB (voir la table de configuration matérielle pour entrée TOR (DI) de la CPU 1217C) (Page 1510) <ul style="list-style-type: none"> • 1 MHz (Ib.2 à Ib.5) • 100/180 kHz (Ia.0 à Ia.5) • 30/120 kHz (Ia.6 à Ib.1)
Sorties d'impulsions	Jusqu'à 4 configurées pour utiliser toute sortie intégrée ou du SB (voir la table de configuration matérielle pour sortie TOR (DQ) de la CPU 1217C) (Page 1510) <ul style="list-style-type: none"> • 1 MHz (Qa.0 à Qa.3) • 100 kHz (Qa.4 à Qb.1)
Entrées de capture d'impulsions	14
Alarmes temporisées	4 au total avec résolution de 1 ms
Alarmes cycliques	4 au total avec résolution de 1 ms
Alarmes sur front	12 sur front montant et 12 sur front descendant (16 et 16 avec Signal Board optionnel)
Carte mémoire	Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel	+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel	20 jours typ./12 jours min. à 40 °C (supercondensateur sans maintenance)

¹ La vitesse plus faible s'applique lorsque le HSC est configuré pour le fonctionnement en quadrature de phase.

Tableau A- 89 Performances

Type d'instruction		Vitesse d'exécution	
		Adressage direct (I, Q et M)	Accès au DB
Instructions booléennes		0,08 µs/instruction	
Déplacement	Move_Bool	0,3 µs/instruction	1,17 µs/instruction
	Move_Word	0,137 µs/instruction	1,0 µs/instruction
	Move_Real	0,72 µs/instruction	1,0 µs/instruction
Opérations arithmétiques sur réels	Add_Real	1,48 µs/instruction	1,78 µs/instruction

Remarque

De nombreuses variables affectent les temps mesurés. Les temps de performance ci-dessus concernent les instructions les plus rapides dans cette catégorie et les programmes exempts d'erreur.

A.8.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1217C

Tableau A- 90Blocs, temporisations et compteurs pris en charge par la CPU 1217C

Elément		Description
Blocs	Type	OB, FB, FC, DB
	Taille	64 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	FB et FC : 1 à 65535 (FB 1 à FB 65535, par exemple) DB : 1 à 59999
	Profondeur d'imbrication	16 en cas d'appel depuis l'OB de cycle de programme ou de démarrage 6 en cas d'appel depuis un OB d'alarme associé à un événement ¹
	Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.
OB	Cycle de programme	Multiple
	Mise en route	Multiple
	Alarmes temporisées	4 (1 par événement)
	Alarmes cycliques	4 (1 par événement)
	Alarmes de processus	50 (1 par événement)
	Alarmes d'erreur de temps	1
	Alarmes de diagnostic	1
	Débrochage ou enfichage de modules	1
	Défaillance du châssis ou de la station	1
	Heure	Multiple
	Etat	1
	Mettre à jour	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
MC-PostServo	1	
Temporisations	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> • SInt, USInt : 3 octets • Int, UInt : 6 octets • DInt, UDInt : 12 octets

¹ Les programmes de sécurité utilisent deux niveaux d'imbrication. Le programme utilisateur a donc une profondeur d'imbrication de quatre dans les programmes de sécurité.

Tableau A- 91Communication

Caractéristiques techniques	Description
Nombre de ports	2
Type	Ethernet
Appareil IHM	4
Console de programmation (PG)	1
Liaisons	<ul style="list-style-type: none"> • 8 liaisons pour la communication ouverte Open User Communication (active ou passive) : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRC • 8 liaisons CPU à CPU (client ou serveur) pour les données GET/PUT • 6 liaisons pour l'affectation dynamique à GET/PUT ou Open User Communication
Débits	10/100 Mb/s
Isolation (logique de signal externe)	Transformateur isolé, 1 500 V CA (essai de type) ¹
Type de câble	CAT5e blindé
Interfaces	
Nombre d'interfaces PROFINET	1
Nombre d'interfaces PROFIBUS	0
Interface	
Matériel	
Nombre de ports	2
Commutateur intégré	Oui
RJ-45 (Ethernet)	Oui ; X1
Protocoles	
Contrôleur PROFINET IO	Oui
Périphérique PROFINET IO	Oui
Communication SIMATIC	Oui
Communication IE ouverte	Oui
Serveur Web	Oui
Redondance des supports	Oui
Contrôleur PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT	Non
MRP	Oui en tant que client MRP
PROFenergy	Oui. La CPU S7-1200 ne prend en charge que l'unité PROFenergy (avec la fonctionnalité I-device).
Démarrage priorisé	Oui (16 périphériques PROFINET max.)
Nombre de périphériques IO raccordables max.	16

Caractéristiques techniques	Description
Nombre de périphériques IO que vous pouvez raccorder pour RT, max.	16
Dont en ligne, max.	16
Nombre de périphériques IO pouvant être activés/désactivés simultanément, max.	8
Temps d'actualisation	La valeur minimale du temps d'actualisation dépend également du composant de communication défini pour PROFINET IO, du nombre de périphériques IO et de la quantité de données utilisateur configurées.
Avec RT	
Horloge de transmission de 1 ms	1 ms à 512 ms
Périphérique PROFINET IO	
Services	
Communication PG/OP	Oui
Routage S7	Oui
Mode synchrone	Non
Communication IE ouverte	Oui
IRT pris en charge	Non
MRP pris en charge	Oui
PROFenergy	Oui
Appareil partagé	Oui
Nombre de contrôleurs IO avec appareil partagé, max.	2
Communication SIMATIC	
Communication S7, en tant que serveur	Oui
Communication S7, en tant que client	Oui
Données utilisateur par tâche, max.	Voir l'aide en ligne (Communication S7, taille des données utilisateur)
Communication IE ouverte	
TCP/IP	Oui
Longueur de données, max.	8 ko
Prise en charge de plusieurs liaisons passives par port	Oui
ISO-on-TCP (RFC1006)	Oui
Longueur de données, max.	8 ko
UDP	Oui
Longueur de données, max.	1 472 octets
DHCP	Non
SNMP	Oui
DCP	Oui
LLDP	Oui

¹ L'isolation du port Ethernet est conçue pour limiter les risques pendant les pannes de réseau de courte durée avec tensions dangereuses. Elle ne satisfait pas aux exigences en matière de sécurité pour l'isolation des lignes CA de routine.

Tableau A- 92Alimentation

Caractéristiques techniques		CPU 1217C DC/DC/DC
Plage de tension		20,4 V CC à 28,8 V CC
Fréquence de ligne		--
Courant d'entrée (charge max.)	CPU uniquement	600 mA à 24 V CC
	CPU avec tous les accessoires d'extension	1 600 mA à 24 V CC
Appel de courant (max.)		12 A à 28,8 V CC
$I^2 t$		0,5 A ² s
Isolation (alimentation d'entrée à logique)		Non isolée
Temps de retard (par perte d'alimentation)		10 ms à 24 V CC
Fusible interne, non remplaçable par l'utilisateur		3 A, 250 V, action retardée

Tableau A- 93Alimentation de capteur

Caractéristiques techniques		CPU 1217C DC/DC/DC
Plage de tension		L+ moins 4 V CC min.
Courant de sortie nominal (max.)		400 mA (protégé contre les courts-circuits)
Bruit d'ondulation maximum (<10 MHz)		Comme la ligne d'entrée
Isolation (logique CPU à alimentation de capteur)		Non isolée

A.8.3 Entrées et sorties TOR

Tableau A- 94 Entrées TOR

Caractéristiques techniques	CPU 1217C DC/DC/DC
Nombre d'entrées	14 : total : 10 : P/N (CEI type 1 en mode P) 4 : Différentiel (RS422/RS485)
Type : P/N (CEI type 1 en mode P)	Ia.0 à Ia.7, Ib.0 à Ib.1
Tension nominale	24 V CC à 4 mA, nominal
Tension continue admise	30 V CC max.
Tension de choc	35 V CC pour 0,5 s
Signal 1 logique (min.)	15 V CC à 2,5 mA
Signal 0 logique (max.)	5 V CC à 1 mA
Isolation (site à logique)	707 V CC (essai de type)
Groupes d'isolation	1
Temps de filtre	microsecondes : 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 millisecondes : 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V CC)	100/80 kHz (Ia.0 à Ia.5) 30/20 kHz (Ia.6 à Ib.1)
Type : Entrée différentielle (RS422/RS485)	Ib.2 à Ib.5 (.2+ .2- à .5+ .5-)
Plage de tension en mode commun	-7 V à +12 V, 1 seconde, 3 V eff. continu (caractéristiques RS422/RS485)
Terminaison et polarisation intégrées	390 Ω à 2M sur Ib'-', 390 Ω à +5 V sur Ib'-', (pas polarisé lorsque bornes en circuit ouvert) 220 Ω entre Ib'+ et Ib'-'
Impédance d'entrée récepteur	100 Ω, terminaison et polarisation inclus
Seuil/sensibilité récepteur différentiel	+/- 0,2 V min., hystérésis typique 60 mV (caractéristiques RS422/RS485)
Isolation (site à logique)	707 V CC (essai de type)
Groupes d'isolation	1
Temps de filtre	microsecondes : 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 millisecondes : 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Fréquences d'entrée d'horloge HSC (max.)	Monophasé : 1 MHz (Ib.2 à Ib.5) Quadrature de phase : 1 MHz (Ib.2 à Ib.5)
Distorsion entrée voie-à-voie différentielle	40 ns max.

Caractéristiques techniques	CPU 1217C DC/DC/DC
Spécifications générales (toutes les entrées TOR)	
Nombre d'entrées simultanément à 1	5 entrées P/N (pas d'E/S adjacentes) et 4 entrées différentielles à 60 °C horizontal ou 50 °C vertical 14 à 55 °C horizontal ou 45 °C vertical
Longueur de câble (mètres)	500 m blindé, 300 m non blindé 50 m blindé pour entrées HSC (P/N) 50 m blindé, paire torsadée pour toutes les entrées différentielles

Tableau A- 95Table de configuration matérielle pour entrée TOR (DI) de la CPU 1217C

Entrée	Type et fréquence
DIa.0	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 100 kHz max.
DIa.1	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 100 kHz max.
DIa.2	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 100 kHz max.
DIa.3	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 100 kHz max.
DIa.4	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 100 kHz max.
DIa.5	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 100 kHz max.
DIa.6	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 30 kHz max.
DIa.7	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 30 kHz max.
DIb.0	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 30 kHz max.
DIb.1	Type : 24 V, P/N type 1 entrée Fréquence d'entrée compteur rapide : 30 kHz max.
DIb.2+ .2-	Type : Entrée différentielle RS422/RS485 Fréquence d'entrée compteur rapide : 1 MHz max.
DIb.3+ .3-	Type : Entrée différentielle RS422/RS485 Fréquence d'entrée compteur rapide : 1 MHz max.
DIb.4+ .4-	Type : Entrée différentielle RS422/RS485 Fréquence d'entrée compteur rapide : 1 MHz max.
DIb.5+ .5-	Type : Entrée différentielle RS422/RS485 Fréquence d'entrée compteur rapide : 1 MHz max.

Tableau A- 96Sorties TOR

Caractéristiques techniques	CPU 1217C DC/DC/DC
Nombre de sorties	10 au total 6 : Transistor à effet de champ MOS (mode P) 4 : Différentiel (RS422/RS485)
Type : Transistor à effet de champ MOS (mode P)	Qa.4 à Qb.1
Plage de tension	20,4 à 28,8 V CC
Signal 1 logique à courant max.	20 V CC min.
Signal 0 logique avec charge 10 kΩ	0,1 V CC max.
Courant (max.)	0,5 A
Charge de lampe	5 W
Résistance état activé	0,6 Ω max.
Courant de fuite par sortie	10 µA max.
Courant de choc	8 A pour 100 ms max.
Protection contre la surcharge	Non
Isolation (site à logique)	707 V CC (essai de type)
Groupes d'isolation	1
Tension de blocage inductive	L+ moins 48 V CC, dissipation 1 W
Retard de commutation (Qa.4 à Qb.1)	1,0 µs max., de 0 à 1 3,0 µs max., de 1 à 0
Fréquence de commutation maximum des relais	--
Fréquence de sortie de trains d'impulsions	100 kHz max. (Qa.4 à Qb.1) ¹ , 2 Hz min.
Type : Sortie différentielle (RS422/RS485)	Qa.0 à Qa.3 (.0+ 0- à .3+ .3-)
Plage de tension en mode commun	-7 V à +12 V, 1 seconde, 3 V eff. continu (caractéristiques RS422/RS485)
Tension de sortie différentielle émetteur	2 V min. à RL = 100 Ω, 1,5 V min. à RL = 54 Ω (caractéristiques RS422/RS485)
Terminaison intégrée	100 Ω entre Qa'+ et Qa'-
Impédance de sortie entraînement	100 Ω, terminaison incluse
Isolation (site à logique)	707 V CC (essai de type)
Groupes d'isolation	1
Retard de commutation (DQa.0 à DQa.3)	100 ns max.
Distorsion sortie voie-à-voie différentielle	40 ns max.
Fréquence de sortie de trains d'impulsions	1 MHz (Qa.0 à Qa.3), 2 Hz min.

Caractéristiques techniques	CPU 1217C DC/DC/DC
Spécifications générales (toutes les sorties TOR)	
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	3 transistors à effet de champ MOS (mode P) (pas d'E/S adjacentes) et 4 sorties différentielles à 60 °C horizontal ou 50 °C vertical 10 à 55 °C horizontal ou 45 °C vertical
Longueur de câble (mètres)	500 m blindé, 150 m non blindé

- ¹ Selon vos câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10% du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

Tableau A- 97Table de configuration matérielle pour sortie TOR (DQ) de la CPU 1217C

Sortie	Type et fréquence
DQa.0+ .0-	Type : Sortie différentielle RS422/RS485 Fréquence de sortie de trains d'impulsions : 1 MHz max., 2 Hz min.
DQa.1+ .1-	Type : Sortie différentielle RS422/RS485 Fréquence de sortie de trains d'impulsions : 1 MHz max., 2 Hz min.
DQa.2+ .2-	Type : Sortie différentielle RS422/RS485 Fréquence de sortie de trains d'impulsions : 1 MHz max., 2 Hz min.
DQa.3+ .3-	Type : Sortie différentielle RS422/RS485 Fréquence de sortie de trains d'impulsions : 1 MHz max., 2 Hz min.
DQa.4	Type : mode P 24 V Fréquence de sortie de trains d'impulsions : 100 kHz max., 2 Hz min.
DQa.5	Type : mode P 24 V Fréquence de sortie de trains d'impulsions : 100 kHz max., 2 Hz min.
DQa.6	Type : mode P 24 V Fréquence de sortie de trains d'impulsions : 100 kHz max., 2 Hz min.
DQa.7	Type : mode P 24 V Fréquence de sortie de trains d'impulsions : 100 kHz max., 2 Hz min.
DQb.0	Type : mode P 24 V Fréquence de sortie de trains d'impulsions : 100 kHz max., 2 Hz min.
DQb.1	Type : mode P 24 V Fréquence de sortie de trains d'impulsions : 100 kHz max., 2 Hz min.

A.8.4 Entrées et sorties analogiques

A.8.4.1 Caractéristiques des entrées analogiques

Tableau A- 98 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V
Plage de dépassement (mot de données)	27649 à 32511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32512 à 32767
Résolution	10 bits
Tension de tenue maximum	35 V CC
Lissage	Aucun, faible, moyen ou fort Reportez-vous au tableau sur la Réponse indicielle (ms) des entrées analogiques de la CPU (Page 1514).
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥ 100 kΩ
Isolation (site à logique)	Aucune
Précision (25 °C / -20 à 60 °C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

A.8.4.2 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 99 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms
Temps d'échantillonnage	4,17 ms	5 ms	25 ms

A.8.4.3 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 100 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.8.4.4 Plages de mesure des entrées analogiques pour la tension (CPU)

Tableau A- 101 Représentation des entrées analogiques pour la tension (CPU)

Technologie		Plage de mesure de tension	
Décimal	Hexadécimal	0 à 10 V	
32767	7FFF	11,852 V CC	Débordement haut
32512	7F00		
32511	7EFF	11,759 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valeurs négatives		Valeurs négatives impossibles	

A.8.4.5 Caractéristiques des sorties analogiques

Tableau A- 102 Sorties analogiques

Caractéristiques techniques	Description
Nombre de sorties	2
Type	Courant
Plage pleine échelle	0 à 20 mA
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	20,01 à 23,52 mA
Plage de dépassement (mot de données)	27649 à 32511
Plage de débordement haut	voir la note ¹
Plage de débordement haut (mot de données)	32512 à 32767
Résolution	10 bits
Impédance de sortie	≤500 Ω max.
Isolation (site à logique)	Aucune
Précision (25 °C / - 20 à 60 °C)	3,0% / 3,5% de la pleine échelle
Temps d'établissement	2 ms
Longueur de câble (mètres)	100 m, paire torsadée blindée

¹ En cas de débordement haut, les sorties analogiques se comportent conformément aux paramétrages des propriétés de configuration d'appareil. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : "Appliquer valeur de remplacement" ou "Conserver dernière valeur".

Tableau A- 103 Représentation de sorties analogiques pour courant (CPU 1215C et CPU 1217C)

Système		Plage de sortie du courant	
Décimal	Hexadécimal	0 mA à 20 mA	
32767	7FFF	Voir note 1	Débordement haut
32512	7F00	Voir note 1	
32511	7EFF	23,52 mA	Plage de dépassement haut
27649	6C01		
27648	6C00	20 mA	Plage nominale
20736	5100	15 mA	
34	22	0,0247 mA	
0	0	0 mA	
Valeurs négatives		Valeurs négatives impossibles	

¹ En cas de débordement haut, les sorties analogiques se comportent conformément aux paramétrages des propriétés de configuration d'appareil. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : "Appliquer valeur de remplacement" ou "Conserver dernière valeur".

A.8.5 Schémas de câblage de la CPU 1217C

Tableau A- 104 CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)

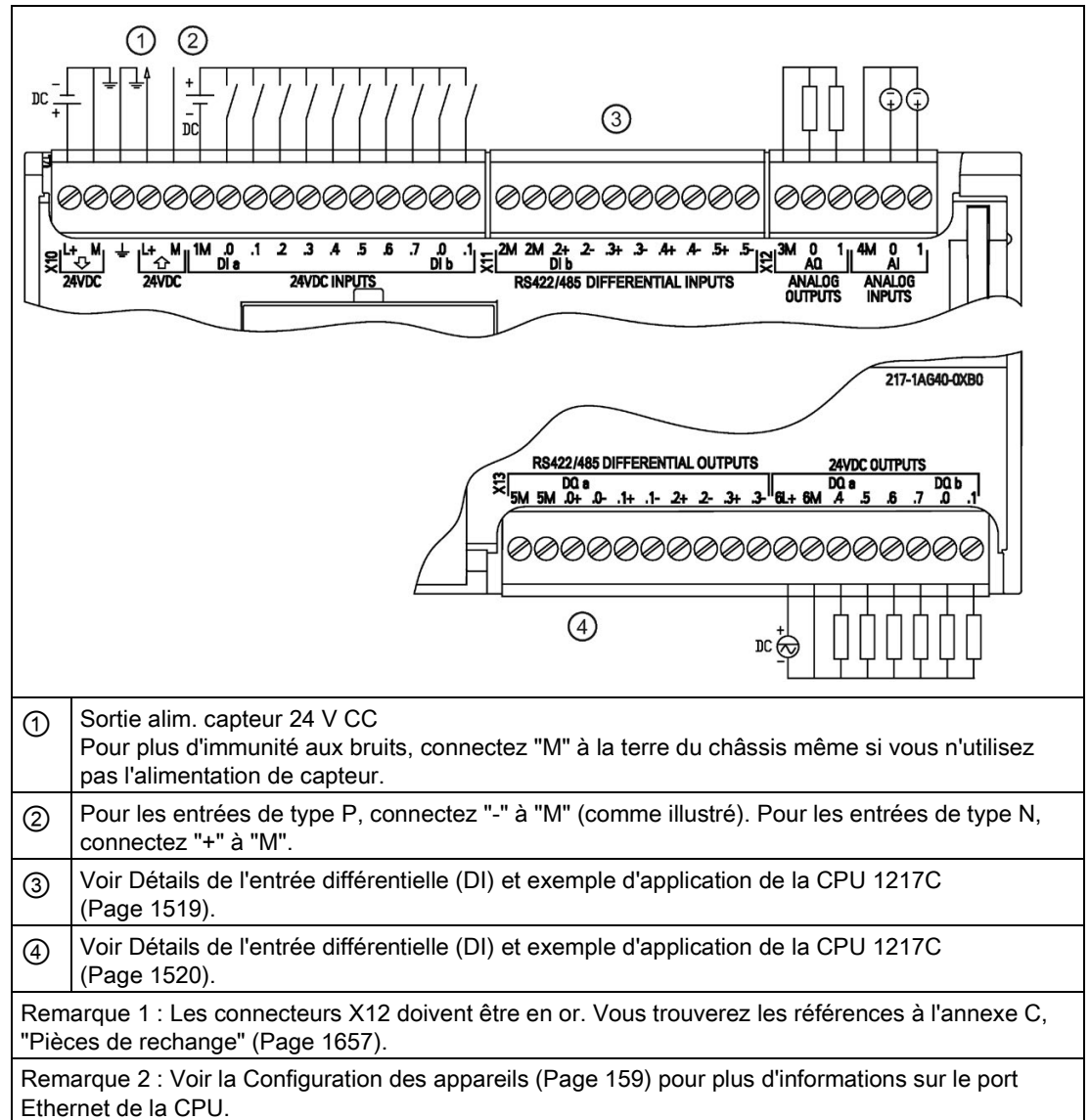


Tableau A- 105 Brochage pour la CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)

Broche	X10	X11	X12 (or)	X13
1	L+/24 V CC	2M	3M	5M
2	M/24 V CC	2M	AQ 0	5M
3	Terre fonctionnelle	DI b.2+	AQ 1	DQ a.0+
4	L+/Sortie capteur 24 V CC	DI b.2-	4M	DQ a.0-
5	M/Sortie capteur 24 V CC	DI b.3+	AI 0	DQ a.1+
6	1M	DI b.3-	AI 1	DQ a.1-
7	DI a.0	DI b.4+	--	DQ a.2+
8	DI a.1	DI b.4-	--	DQ a.2-
9	DI a.2	DI b.5+	--	DQ a.3+
10	DI a.3	DI b.5-	--	DQ a.3-
11	DI a.4	--	--	6L+
12	DI a.5	--	--	6M
13	DI a.6	--	--	DQ a.4
14	DI a.7	--	--	DQ a.5
15	DI b.0	--	--	DQ a.6
16	DI b.1	--	--	DQ a.7
17	--	--	--	DQ b.0
18	--	--	--	DQ b.1

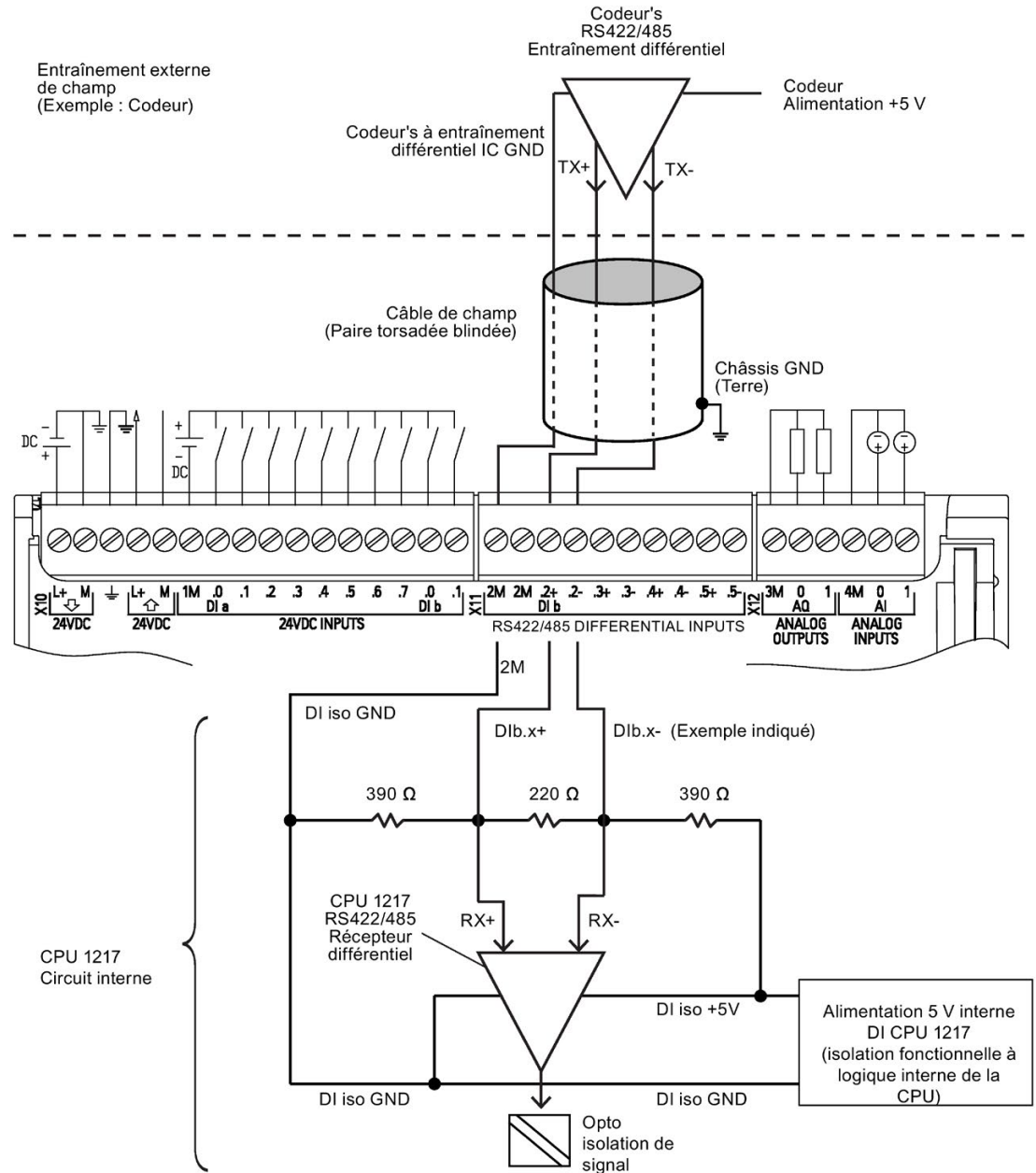
Remarque

Les entrées analogiques inutilisées doivent être court-circuitées.

Voir aussi

Entrées et sorties analogiques (Page 1495)

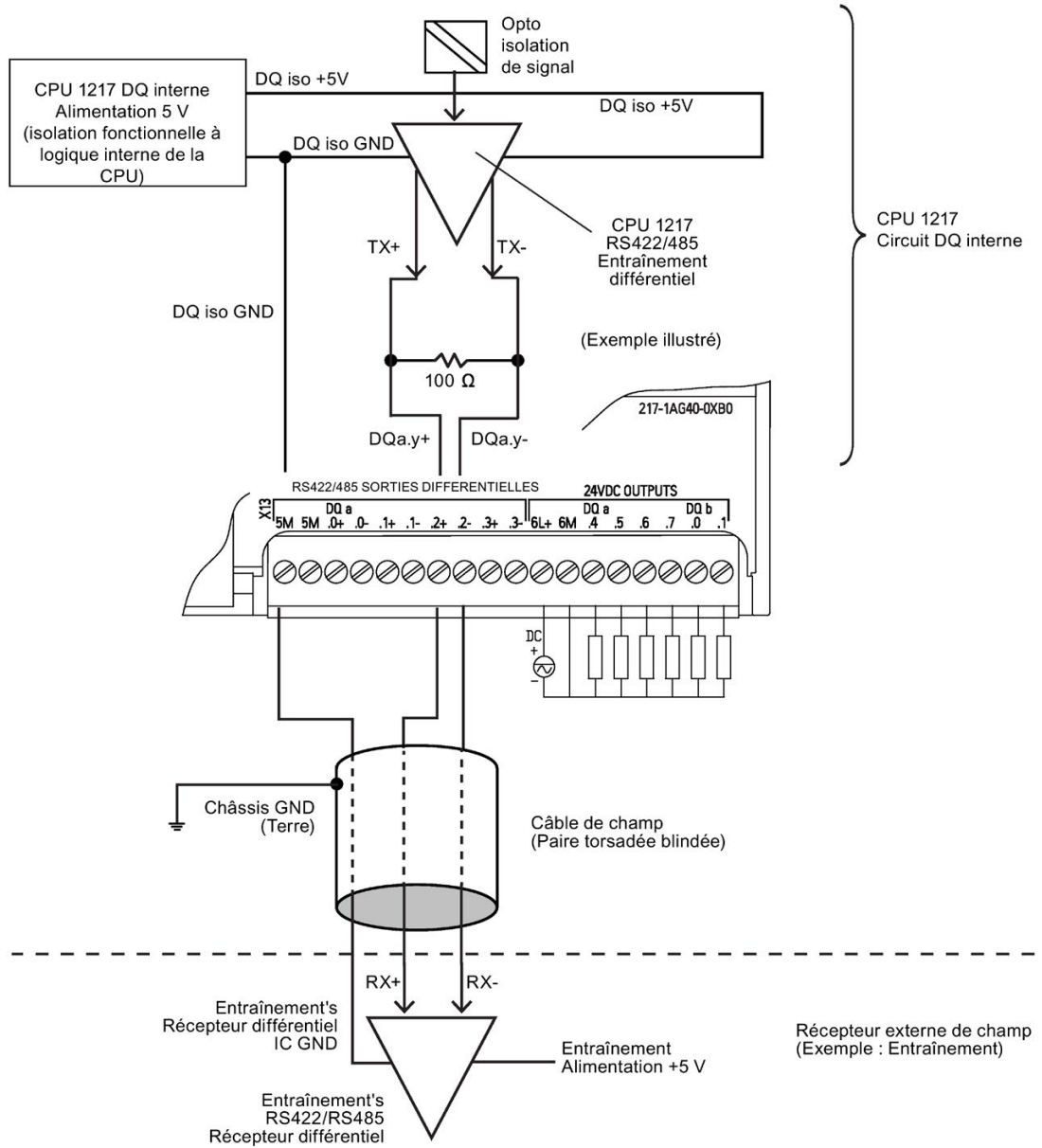
A.8.6 Détails de l'entrée différentielle (DI) et exemple d'application de la CPU 1217C



Re-
marqu
es

- Chaque DI différentielle n'est pas polarisée lorsque les bornes à vis sont en circuit ouvert.
- Terminaison et polarisation intégrées = impédance équivalente 100 Ω.
- Les résistances de terminaison et de polarisation DI intégrées limitent la plage de tension en mode commun continu. Pour plus de détails, reportez-vous aux spécifications électriques.

A.8.7 Détails de l'entrée différentielle (DI) et exemple d'application de la CPU 1217C



Re-
marq
ue

- Les résistances de terminaison DQ intégrées limitent la plage de tension en mode commun continu. Pour plus de détails, reportez-vous aux spécifications électriques.

A.9 Modules d'entrées-sorties TOR (SM)

A.9.1 Caractéristiques des modules d'entrées TOR SM 1221

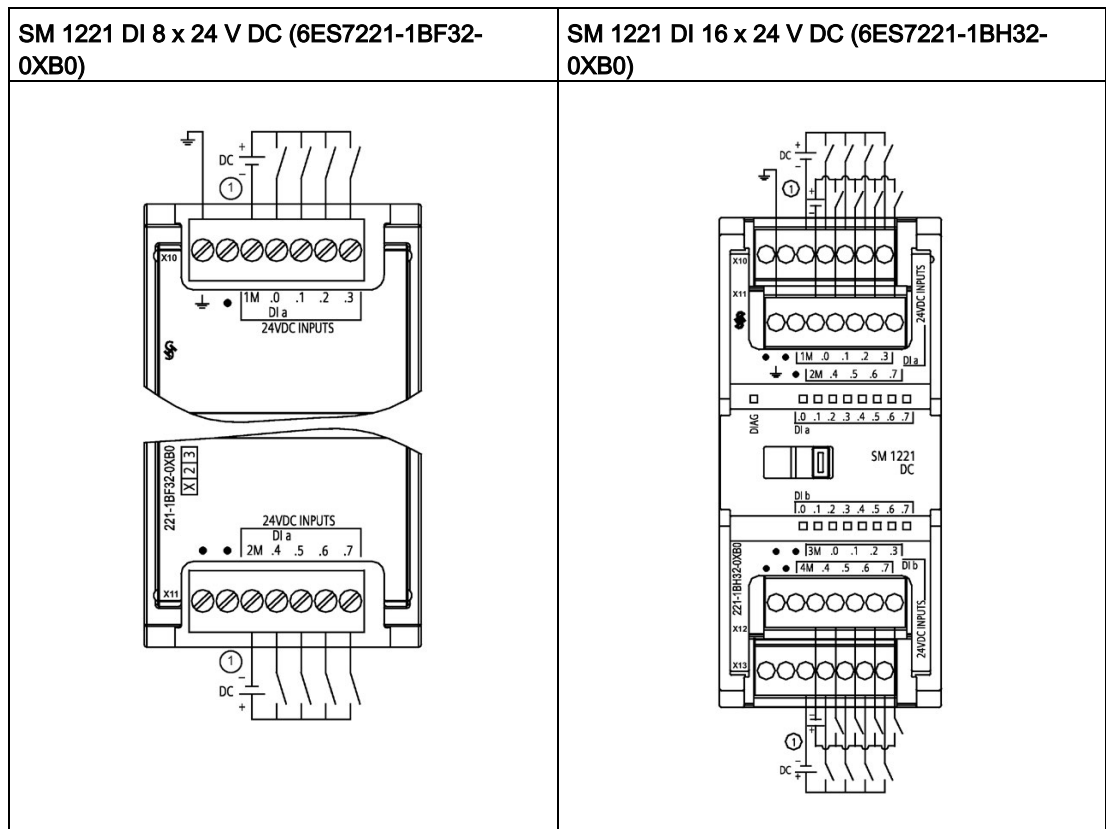
Tableau A- 106 Caractéristiques générales

Modèle	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Référence	6ES7221-1BF32-0XB0	6ES7221-1BH32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	
Poids	170 grammes	210 grammes
Dissipation de courant	1,5 W	2,5 W
Consommation de courant (bus SM)	105 mA	130 mA
Consommation de courant (24 V CC)	4 mA/entrée utilisée	

Tableau A- 107 Entrées TOR

Modèle	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Nombre d'entrées	8	16
Type	P/N (CEI type 1 en mode P)	
Tension nominale	24 V CC à 4 mA, nominal	
Tension continue admise	30 V CC max.	
Tension de choc	35 V CC pour 0,5 s	
Signal 1 logique (min.)	15 V CC à 2,5 mA	
Signal 0 logique (max.)	5 V CC à 1 mA	
Isolation (site à logique)	707 V CC (essai de type)	
Groupes d'isolation	2	4
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms (sélectionnables par groupes de 4)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms (sélectionnables par groupes de 4)
Nombre d'entrées simultanément à 1	8	16
Longueur de câble (mètres)	500 m blindé, 300 m non blindé	

Tableau A- 108 Schémas de câblage des modules SM d'entrées TOR



① Pour les entrées de type P, connectez "-" à "M" (comme illustré). Pour les entrées de type N, connectez "+" à "M".

Tableau A- 109 Brochage pour le SM 1221 DI 8 x 24 V DC (6ES7221-1BF32-0XB0)

Broche	X10	X11
1	Terre fonctionnelle	Pas de connexion
2	Pas de connexion	Pas de connexion
3	1M	2M
4	DI a.0	DI a.4
5	DI a.1	DI a.5
6	DI a.2	DI a.6
7	DI a.3	DI a.7

Tableau A- 110 Brochage pour le SM 1221 DI 16 x 24 V DC (6ES7221-1BH32-0XB0)

Broche	X10	X11	X12	X13
1	Pas de connexion	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
3	1M	2M	3 M	4 M
4	DI a.0	DI a.4	DI b.0	DI b.4
5	DI a.1	DI a.5	DI b.1	DI b.5
6	DI a.2	DI a.6	DI b.2	DI b.6
7	DI a.3	DI a.7	DI b.3	DI b.7

A.9.2 Caractéristiques des modules 8 sorties TOR SM 1222

Tableau A- 111 Caractéristiques générales

Modèle	SM 1222 DQ 8 x Relais	SM 1222 DQ 8 Relais Inverseur	SM 1222 DQ 8 x 24 V DC
Référence	6ES7222-1HF32-0XB0	6ES7222-1XF32-0XB0	6ES7222-1BF32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75
Poids	190 grammes	310 grammes	180 grammes
Dissipation de courant	4,5 W	5 W	1,5 W
Consommation de courant (bus SM)	120 mA	140 mA	120 mA
Consommation de courant (24 V CC)	11 mA / bobine de relais utilisée	16,7 mA / bobine de relais utilisée	50 mA

Tableau A- 112 Sorties TOR

Modèle	SM 1222 DQ 8 x Relais	SM 1222 DQ 8 Relais Inverseur	SM 1222 DQ 8 x 24 V DC
Nombre de sorties	8	8	8
Type	Relai, mécanique	Relais, contact inverseur	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V CC ou 5 à 250 V CA	5 à 30 V CC ou 5 à 250 V CA	20,4 à 28,8 V CC
Signal 1 logique à courant max.	--	--	20 V CC min.
Signal 0 logique avec charge 10 kΩ	--	--	0,1 V CC max.
Courant (max.)	2,0 A	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	30 W CC / 200 W CA	5 W
Résistance état activé (contact)	0,2 Ω max. lorsque neuf	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	--	10 µA max.
Courant de choc	7 A avec contacts fermés	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	Non	
Isolation (site à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)	1 500 V CA (bobine à contact)	707 V CC (essai de type)
Groupes d'isolation	2	8	1
Courant par commun (max.)	10 A	2 A	4 A
Tension de blocage inductive	--	--	L+ moins 48 V, dissipation 1 W
Retard de commutation	10 ms max.	10 ms max.	50 µs max. de 0 à 1 200 µs max. de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	1 Hz	--
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale (contact NO)	100 000 cycles ouvert/fermé	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	8	<ul style="list-style-type: none"> • 4 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical • 8 à 55 °C horizontal ou 45 °C vertical 	8
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé

A.9.3 Caractéristiques des modules 16 sorties TOR SM 1222

Tableau A- 113 Caractéristiques générales

Modèle	SM 1222 DQ 16 x Relais	SM 1222 DQ 16 x 24 V DC
Référence	6ES7222-1HH32-0XB0	6ES7222-1BH32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	45 x 100 x 75
Poids	260 grammes	220 grammes
Dissipation de courant	8,5 W	2,5 W
Consommation de courant (bus SM)	135 mA	140 mA
Consommation de courant (24 V CC)	11 mA / bobine de relais utilisée	100 mA

Tableau A- 114 Sorties TOR

Modèle	SM 1222 DQ 16 x Relais	SM 1222 DQ 16 x 24 V DC
Nombre de sorties	16	16
Type	Relai, mécanique	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V CC ou 5 à 250 V CA	20,4 à 28,8 V CC
Signal 1 logique à courant max.	-	20 V CC min.
Signal 0 logique avec charge 10 k Ω	-	0,1 V CC max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé (contact)	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 μ A max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	
Isolation (site à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)	707 V CC (essai de type)
Groupes d'isolation	4	1
Courant par commun (max.)	10 A	8 A
Tension de blocage inductive	-	L+ moins 48 V, dissipation 1 W
Retard de commutation	10 ms max.	50 μ s max. de 0 à 1 200 μ s max. de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	-
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	-
Durée d'utilisation des contacts à la charge nominale (contact NO)	100 000 cycles ouvert/fermé	-
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)

A.9 Modules d'entrées-sorties TOR (SM)

Modèle	SM 1222 DQ 16 x Relais	SM 1222 DQ 16 x 24 V DC
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> • 8 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical • 16 à 55 °C horizontal ou 45 °C vertical 	16
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	

Tableau A- 115 Schémas de câblage des modules SM à 8 sorties TOR

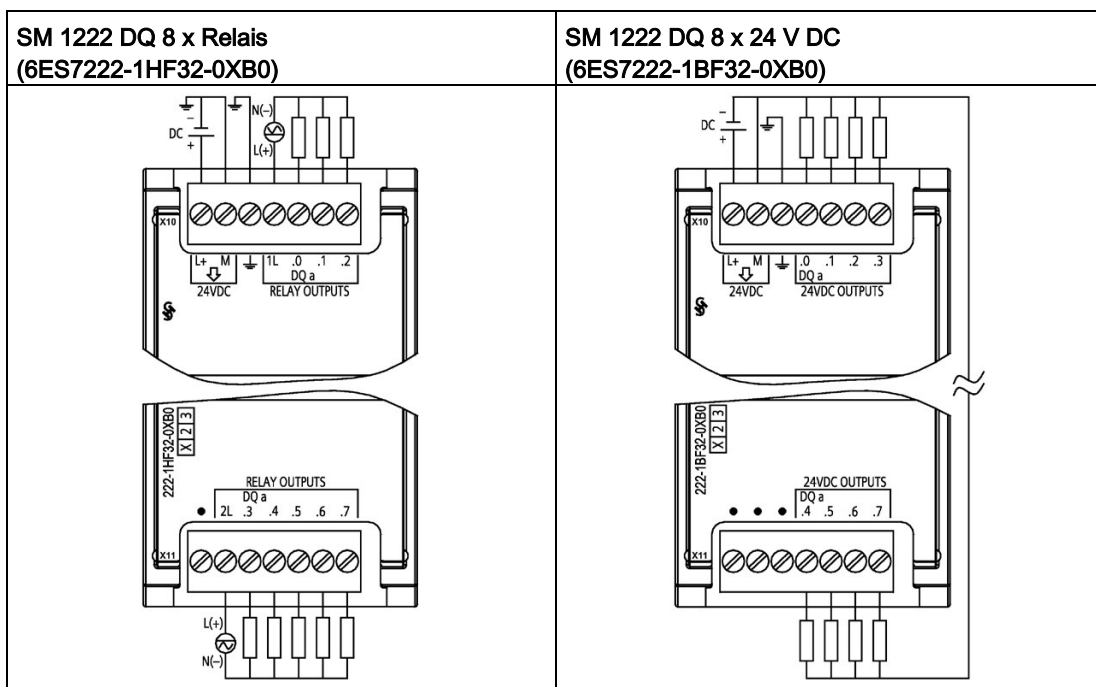


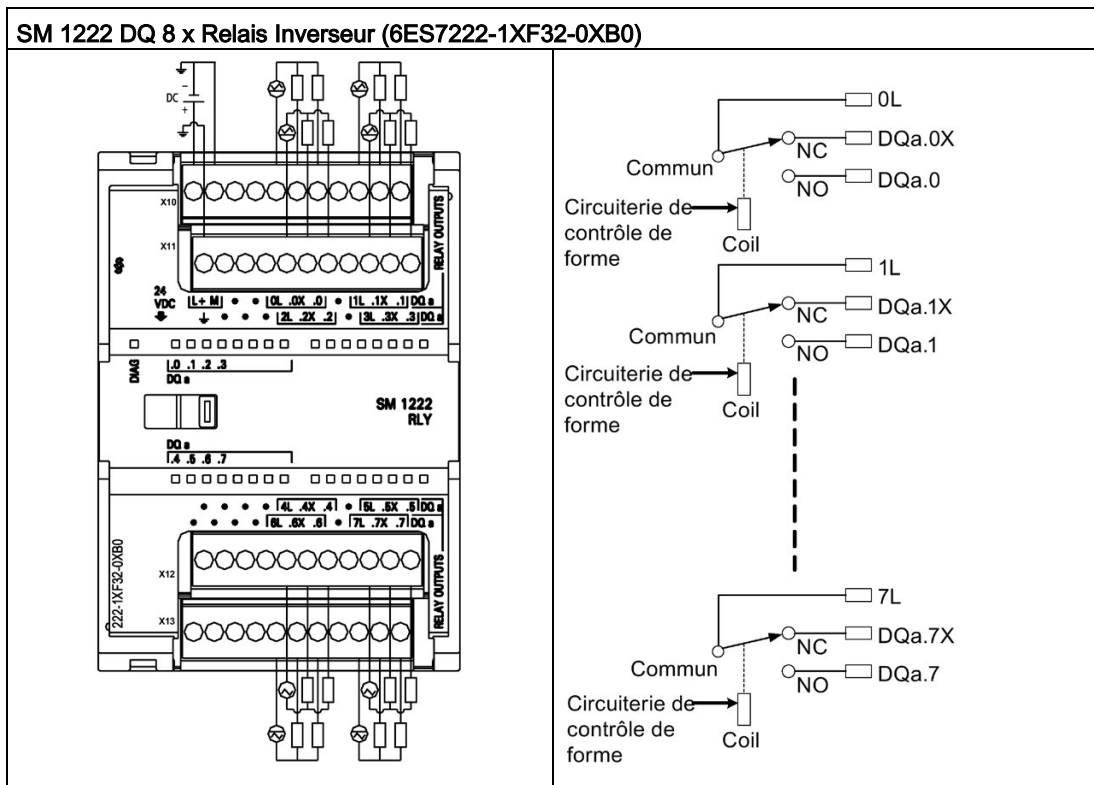
Tableau A- 116 Brochage pour le SM 1222 DQ 8 x Relais (6ES7222-1HF32-0XB0)

Broche	X10	X11
1	L+/24 V CC	Pas de connexion
2	M/24 V CC	2L
3	Terre fonctionnelle	DQ a.3
4	1L	DQ a.4
5	DQ a.0	DQ a.5
6	DQ a.1	DQ a.6
7	DQ a.2	DQ a.7

Tableau A- 117 Brochage pour le SM 1222 DQ 8 x 24 V DC (6ES7222-1BF32-0XB0)

Broche	X10	X11
1	L+/24 V CC	Pas de connexion
2	M/24 V CC	Pas de connexion
3	Terre fonctionnelle	Pas de connexion
4	DQ a.0	DQ a.4
5	DQ a.1	DQ a.5
6	DQ a.2	DQ a.6
7	DQ a.2	DQ a.7

Tableau A- 118 Schéma de câblage du module SM à 8 sorties TOR relais inverseur



Une sortie relais inverseur commande deux circuits à l'aide d'une borne commune : un contact à ouverture et un contact à fermeture. Si on prend la sortie "0" comme exemple, lorsque la sortie est désactivée, le commun (0L) est connecté au contact à ouverture (.0X) et déconnecté du contact à fermeture (.0). Lorsque la sortie est activée, le commun (0L) est déconnecté du contact à ouverture (.0X) et connecté au contact à fermeture (.0).

Tableau A- 119 Brochage pour le SM 1222 DQ 8 x Relais Inverseur (6ES7222-1XF32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
4	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
5	0L	2L	4L	6L
6	DQ a.0X	DQ a.2X	DQ a.4X	DQ a.6X
7	DQ a.0	DQ a.2	DQ a.4	DQ a.6
8	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
9	1L	3L	5L	7L
10	DQ a.1X	DQ a.3X	DQ a.5X	DQ a.7X
11	DQ a.1	DQ a.3	DQ a.5	DQ a.7

Tableau A- 120 Schémas de câblage des modules SM à 16 sorties TOR

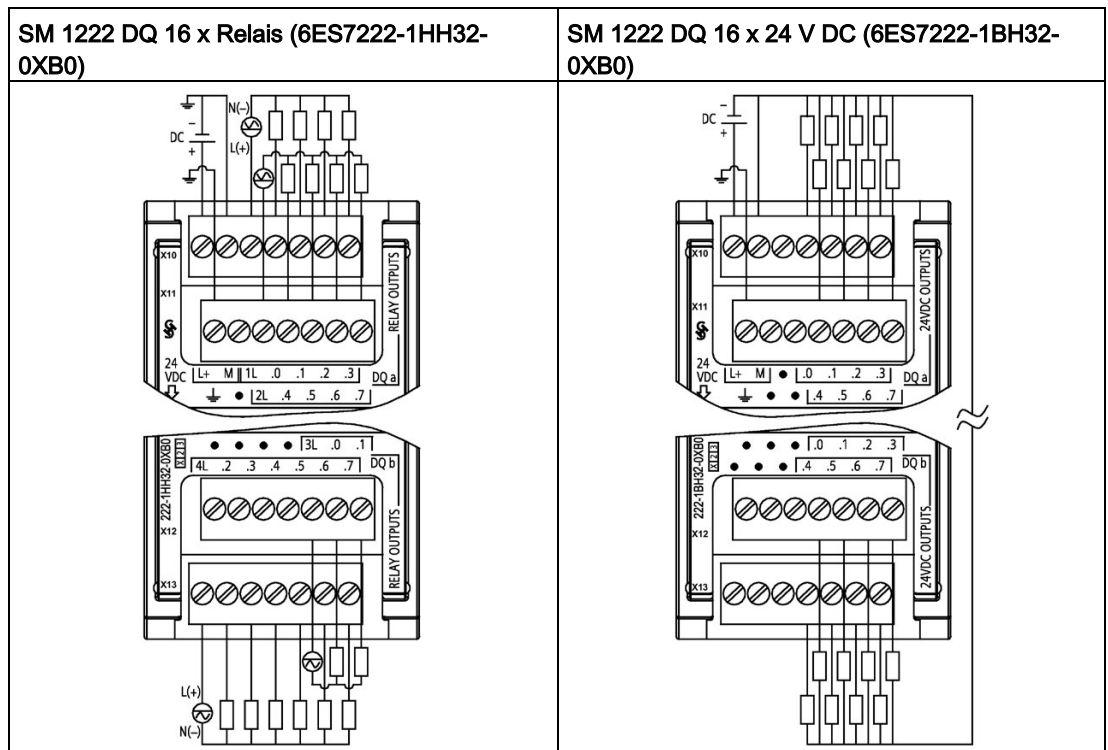


Tableau A- 121 Brochage pour le SM 1222 DQ 16 x Relais (6ES7222-1HH32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Terre fonctionnelle	Pas de connexion	4L
2	M/24 V CC	Pas de connexion	Pas de connexion	DQ b.2
3	1L	2L	Pas de connexion	DQ b.3
4	DQ a.0	DQ a.4	Pas de connexion	DQ b.4
5	DQ a.1	DQ a.5	3L	DQ b.5
6	DQ a.2	DQ a.6	DQ b.0	DQ b.6
7	DQ a.3	DQ a.7	DQ b.1	DQ b.7

Tableau A- 122 Brochage pour le SM 1222 DQ 16 x 24 V DC (6ES7222-1BH32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
4	DQ a.0	DQ a.4	DQ b.0	DQ b.4
5	DQ a.1	DQ a.5	DQ b.1	DQ b.5
6	DQ a.2	DQ a.6	DQ b.2	DQ b.6
7	DQ a.3	DQ a.7	DQ b.3	DQ b.7

A.9.4 Caractéristiques des modules SM 1223 DI/DQ V DC

Tableau A- 123 Caractéristiques générales

Modèle	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC
Référence	6ES7223-1PH32-0XB0	6ES7223-1PL32-0XB0	6ES7223-1BH32-0XB0	6ES7223-1BL32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75	70 x 100 x 75
Poids	230 grammes	350 grammes	210 grammes	310 grammes
Dissipation de courant	5,5 W	10 W	2,5 W	4,5 W
Consommation de courant (bus SM)	145 mA	180 mA	145 mA	185 mA
Consommation de courant (24 V CC)	4 mA / entrée utilisée 11 mA / bobine de relais utilisée		150 mA	200 mA

Tableau A- 124 Entrées TOR

Modèle	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC
Nombre d'entrées	8	16	8	16
Type	P/N (CEI type 1 en mode P)	P/N (CEI type 1 en mode P)	P/N (CEI type 1 en mode P)	P/N (CEI type 1 en mode P)
Tension nominale	24 V CC à 4 mA, nominal	24 V CC à 4 mA, nominal	24 V CC à 4 mA, nominal	24 V CC à 4 mA, nominal
Tension continue admise	30 V CC max.	30 V CC max.	30 V CC max.	30 V CC max.
Tension de choc	35 V CC pour 0,5 s	35 V CC pour 0,5 s	35 V CC pour 0,5 s	35 V CC pour 0,5 s
Signal 1 logique (min.)	15 V CC à 2,5 mA	15 V CC à 2,5 mA	15 V CC à 2,5 mA	15 V CC à 2,5 mA
Signal 0 logique (max.)	5 V CC à 1 mA	5 V CC à 1 mA	5 V CC à 1 mA	5 V CC à 1 mA
Isolation (site à logique)	707 V CC (essai de type)	707 V CC (essai de type)	707 V CC (es- sai de type)	707 V CC (essai de type)
Groupes d'isolation	2	2	2	2
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélectionnable en groupes de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélectionnable en groupes de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélec- tionnable en groupes de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélectionnable en groupes de 4
Nombre d'entrées simultanément à 1	8	<ul style="list-style-type: none"> 8 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical 16 à 55 °C horizontal ou 45 °C vertical 	8	16
Longueur de câble (mètres)	500 m blindé, 300 m non blindé	500 m blindé, 300 m non blindé	500 m blindé, 300 m non blindé	500 m blindé, 300 m non blindé

Tableau A- 125 Sorties TOR

Modèle	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC
Nombre de sorties	8	16	8	16
Type	Relai, mécanique	Relai, mécanique	Transistor à effet de champ MOS (type "source")	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V CC ou 5 à 250 V CA	5 à 30 V CC ou 5 à 250 V CA	20,4 à 28,8 V CC	20,4 à 28,8 V CC
Signal 1 logique à courant max.	--	--	20 V CC min.	20 V CC min.
Signal 0 logique avec charge 10 kΩ	--	--	0,1 V CC max.	0,1 V CC max.
Courant (max.)	2,0 A	2,0 A	0,5 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	30 W CC / 200 W CA	5 W	5 W
Résistance état activé (contact)	0,2 Ω max. lorsque neuf	0,2 Ω max. lorsque neuf	0,6 Ω max.	0,6 Ω max.
Courant de fuite par sortie	--	--	10 µA max.	10 µA max.
Courant de choc	7 A avec contacts fermés	7 A avec contacts fermés	8 A pour 100 ms max.	8 A pour 100 ms max
Protection contre la surcharge	Non	Non	Non	Non
Isolation (site à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)	707 V CC (essai de type)	707 V CC (essai de type)
Groupes d'isolation	2	4	1	1
Courant par commun	10 A	8 A	4 A	8 A
Tension de blocage inductive	--	--	L+ moins 48 V, dissipation 1 W	L+ moins 48 V, dissipation 1 W
Retard de commutation	10 ms max.	10 ms max.	50 µs max. de 0 à 1 200 µs max. de 1 à 0	50 µs max. de 0 à 200 µs max. de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	1 Hz	--	--
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	10 000 000 cycles ouvert/fermé	--	--
Durée d'utilisation des contacts à la charge nominale (contact NO)	100 000 cycles ouvert/fermé	100 000 cycles ouvert/fermé	--	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)

Modèle	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC
Nombre de sorties simultanément à 1	8	<ul style="list-style-type: none"> 8 (points non adjacents) à 60 °C horizontal ou 50 °C vertical 16 à 55 °C horizontal ou 45 °C vertical 	8	16
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé

Tableau A- 126 Schémas de câblage des modules SM DI V DC / sorties relais

SM 1223 DI 8 x 24 V DC, DQ 8 x Relais (6ES7223-1PH32-0XB0)	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais (6ES7223-1PL32-0XB0)	Remarques
		<p>① Pour les entrées "sink" (type P), connectez "-" à "M" (comme illustré).</p> <p>Pour les entrées "source" (type N), connectez "+" à "M".</p>

Tableau A- 127 Brochage pour le SM 1223 DI 8 x 24 V DC, DQ 8 x Relais (6ES7223-1PH32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	1M	2M	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

Tableau A- 128 Brochage pour le SM 1223 DI 16 x 24 V DC, DQ 16 x Relais (6ES7223-1PL32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Terre fonctionnelle	1L	3L
2	M/24 V CC	Pas de connexion	DQ a.0	DQ b.0
3	1M	2M	DQ a.1	DQ b.1
4	DI a.0	DI b.0	DQ a.2	DQ b.2
5	DI a.1	DI b.1	DQ a.3	DQ b.3
6	DI a.2	DI b.2	Pas de connexion	Pas de connexion
7	DI a.3	DI b.3	2L	4L
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

Tableau A- 129 Schémas de câblage des modules SM DI/DQ V DC

SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC (6ES7223-1BH32-0XB0)	SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC (6ES7223-1BL32-0XB0)	Remarques
		<p>① Pour les entrées "sink" (type P), connectez "-" à "M" (comme illustré).</p> <p>Pour les entrées "source" (type N), connectez "+" à "M".</p>

Tableau A- 130 Brochage pour le SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC (6ES7223-1BH32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	1M	2M	Pas de connexion	Pas de connexion
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

Tableau A- 131 Brochage pour le SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC (6ES7223-1BL32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	1M	2M	Pas de connexion	Pas de connexion
4	DI a.0	DI b.0	DQ a.0	DQ b.0
5	DI a.1	DI b.1	DQ a.1	DQ b.1
6	DI a.2	DI b.2	DQ a.2	DQ b.2
7	DI a.3	DI b.3	DQ a.3	DQ b.3
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

A.9.5 Caractéristiques des modules SM 1223 DI/DQ V AC

Tableau A- 132 Caractéristiques générales

Modèle	SM 1223 DI 8 x120/230 V AC / DQ 8 x Relais
Référence	6ES7223-1QH32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75 mm
Poids	190 grammes
Dissipation de courant	7,5 W
Consommation de courant (bus SM)	120 mA
Consommation de courant (24 V CC)	11 mA par sortie activée

Tableau A- 133 Entrées TOR

Modèle	SM 1223 DI 8 x 120/230 V AC / DQ 8 x Relais
Nombre d'entrées	8
Type	CEI type 1
Tension nominale	120 V CA à 6 mA, 230 V CA à 9 mA
Tension continue admise	264 V CA
Tension de choc	--
Signal 1 logique (min.)	79 V CA à 2,5 mA
Signal 0 logique (max.)	20 V CA à 1 mA
Courant de fuite (max.)	1 mA
Isolation (site à logique)	1 500 V CA
Groupes d'isolation ¹	4
Temps de retard d'entrée	Typique : 0,2 à 12,8 ms, personnalisable Maximum : -
Connexion de capteur de proximité à 2 fils (Bero) (max.)	1 mA
Longueur de câble	Non blindé : 300 mètres Blindé : 500 mètres
Nombre d'entrées simultanément à 1	8

¹ Les voies à l'intérieur d'un groupe doivent être de la même phase.

Tableau A- 134 Sorties TOR

Modèle	SM 1223 DI 8 x 120/230 V AC / DQ 8 x Relais
Nombre de sorties	8
Type	Relai, mécanique
Plage de tension	5 à 30 V CC ou 5 à 250 V CA
Signal 1 logique à courant max.	--
Signal 0 logique avec charge 10 kΩ	--
Courant (max.)	2,0 A
Charge de lampe	30 W CC / 200 W CA
Résistance état activé (contact)	0,2 Ω max. lorsque neuf
Courant de fuite par sortie	--
Courant de choc	7 A avec contacts fermés
Protection contre la surcharge	Non
Isolation (site à logique)	1 500 V CA (bobine à contact) Aucune (bobine à logique)
Groupes d'isolation	2
Courant par commun (max.)	10 A
Tension de blocage inductive	--
Retard de commutation (max.)	10 ms
Fréquence de commutation maximum des relais	1 Hz
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> • 4 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical • 8 à 55 °C horizontal ou 45 °C vertical
Longueur de câble (mètres)	500 m blindé, 150 m non blindé

Tableau A- 135 SM 1223 DI 8 x 120/230 V AC, DQ 8 x Relais (6ES7223-1QH32-0XB0)

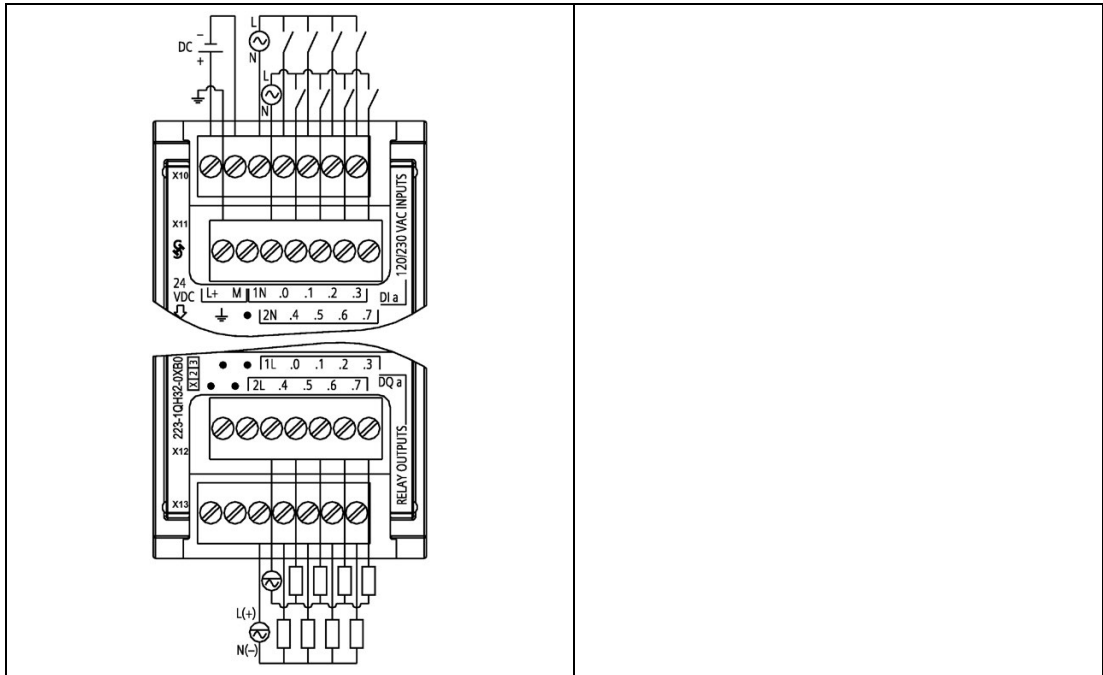


Tableau A- 136 Brochage pour le SM 1223 DI 8 x 120/240 V AC, DQ 8 x Relais (6ES7223-1QH32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	1N	2N	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

A.10 Modules d'entrées-sorties analogiques (SM)

A.10.1 Caractéristiques des modules d'entrées analogiques SM 1231

Tableau A- 137 Caractéristiques générales

Modèle	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Référence	6ES7231-4HD32-0XB0	6ES7231-4HF32-0XB0	6ES7231-5ND32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75		
Poids	180 grammes		
Dissipation de courant	2,2 W	2,3 W	2,0 W
Consommation de courant (bus SM)	80 mA	90 mA	80 mA
Consommation de courant (24 V CC)	45 mA		65 mA

Tableau A- 138 Entrées analogiques

Modèle	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Nombre d'entrées	4	8	4
Type	Tension ou courant (différentiel) : sélectionnable en groupes de 2		Tension ou courant (différentiel)
Plage	±10 V, ±5 V, ±2,5 V, 0 à 20 mA ou 4 mA à 20 mA		±10 V, ±5 V, ±2,5 V, ±1,25 V, 0 à 20 mA ou 4 mA à 20 mA
Plage pleine échelle (mot de données)	-27648 à 27648 tension / 0 à 27648 courant		
Plage de dépassement haut/bas (mot de données) Voir la partie sur les plages des entrées analogiques pour la tension et le courant (Page 1554).	Tension : 32511 à 27649 / - 27649 à - 32 512 Courant : 32511 à 27649 / 0 à - 4864		
Débordement haut/bas (mot de données) Voir la partie sur les plages des entrées pour la tension et le courant (Page 1554).	Tension : 32767 à 32512 / - 32513 à - 32768 Courant 0 à 20 mA : 32767 à 32512 / - 4865 à - 32768 Courant 4 à 20 mA : 32767 à 32512 (les valeurs inférieures à - 4864 indiquent un fil ouvert)		
Résolution ¹	12 bits + bit de signe		15 bits + bit de signe
Tension/courant de tenue maximum	±35 V / ±40 mA		
Lissage	Aucun, faible, moyen ou fort Voir la partie sur les temps de réponse indicielle (Page 1553).		
Réjection des bruits	400, 60, 50 ou 10 Hz Voir la partie sur les taux d'échantillonnage (Page 1553).		
Impédance d'entrée	≥ 9 MΩ (tension) / ≥ 270 Ω < 290 Ω (courant)		≥ 1 MΩ (tension) / < 315 Ω > 280 Ω (courant)
Isolation Site à logique Logique à 24 V CC Site à 24 V CC Voie à voie	Aucune		707 V CC (essai de type) 707 V CC (essai de type) 500 V CC (essai de type) Aucun
Précision (25 °C / - 20 à 60 °C)	±0,1% / ±0,2% de la pleine échelle		±0,1% / ±0,3% de la pleine échelle
Principe de mesure	Conversion de valeur effective		
Réjection en mode commun	40 dB, CC pour 60 Hz		
Plage de signal opérationnelle ¹	La tension signal plus mode commun doit être inférieure à + 12 V et supérieure à - 12 V.		
Longueur de câble (mètres)	100 m, torsadé et blindé		

¹ Des tensions en dehors de la plage opérationnelle appliquées à une voie peuvent provoquer des interférences sur les autres voies.

Tableau A- 139 Diagnostic

Modèle	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Débordement haut/bas	Oui		
24 V CC basse tension	Oui		
Fil ouvert	Plage 4 à 20 mA uniquement (si l'entrée est en dessous de - 4164 : 1,185 mA)		

Mesure de courant SM 1231

Vous pouvez réaliser une mesure de courant avec soit un transducteur 2 fils, soit un transducteur 4 fils comme illustré dans la figure suivante :

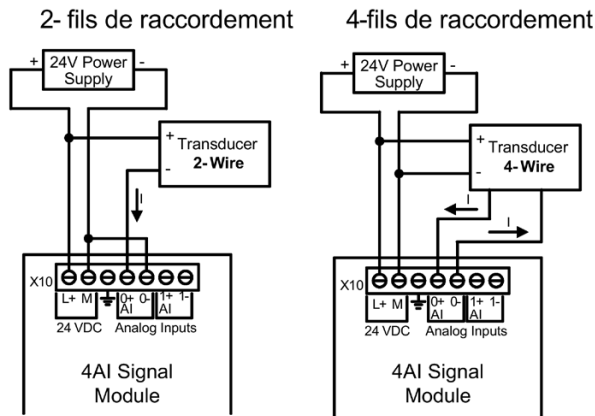
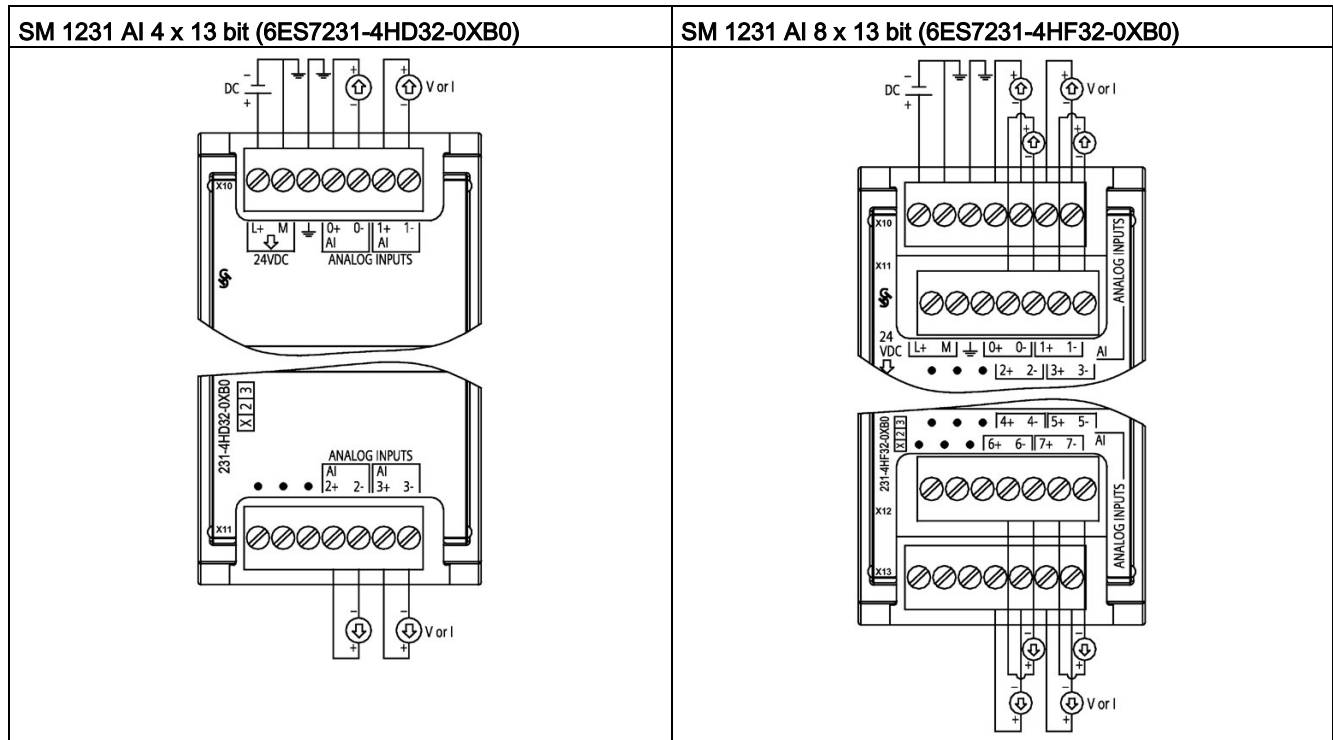


Tableau A- 140 Schémas de câblage des modules SM d'entrées analogiques



Remarque : les connecteurs doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Tableau A- 141 Brochage pour le SM 1231 AI 4 x 13 bit (6ES7231-4HD32-0XB0)

Broche	X10 (or)	X11 (or)
1	L+/24 V CC	Pas de connexion
2	M/24 V CC	Pas de connexion
3	Terre fonctionnelle	Pas de connexion
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Tableau A- 142 Brochage pour le SM 1231 AI 8 x 13 bit (6ES7231-4HF32-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	Terre fonctionnelle	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0+	AI 2+	AI 4+	AI 6+
5	AI 0-	AI 2-	AI 4-	AI 6-
6	AI 1+	AI 3+	AI 5+	AI 7+
7	AI 1-	AI 3-	AI 5-	AI 7-

Tableau A- 143 Schéma de câblage du SM d'entrées analogiques

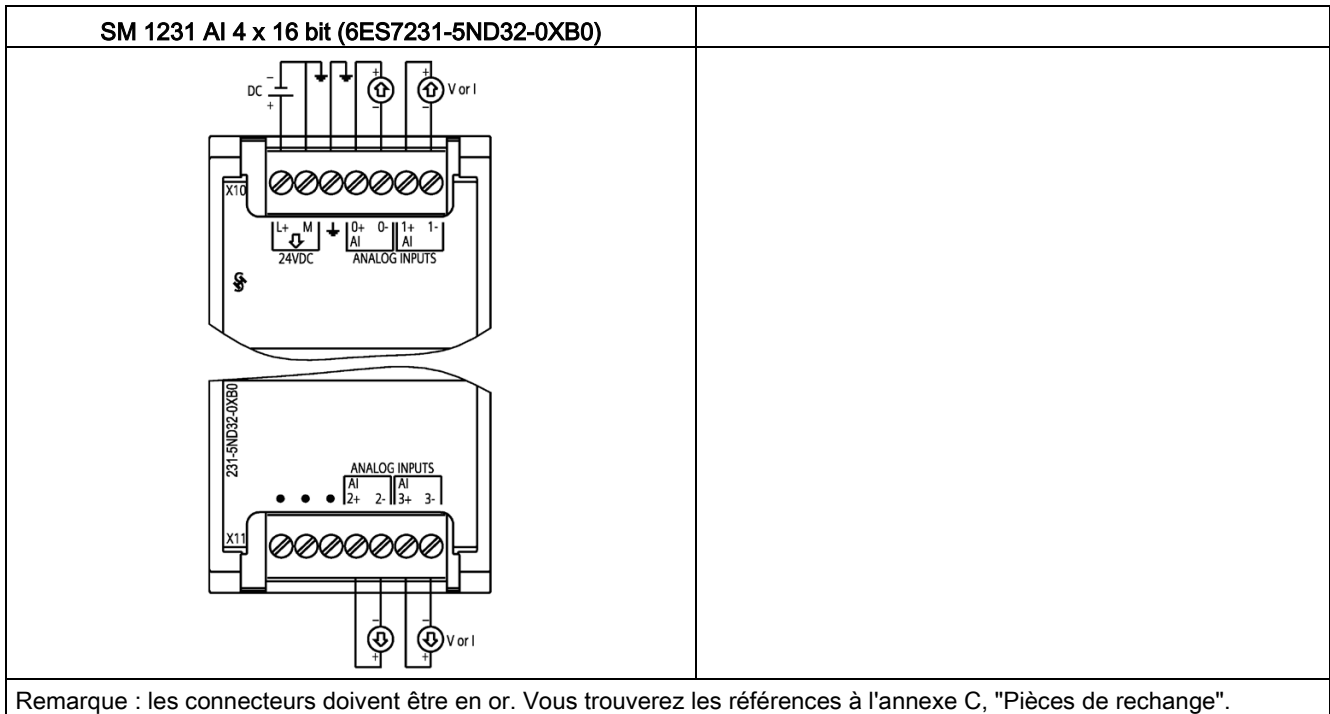


Tableau A- 144 Brochage pour le SM 1231 AI 4 x 16 bit (6ES7231-5ND32-0XB0)

Broche	X10 (or)	X11 (or)
1	L+/24 V CC	Pas de connexion
2	M/24 V CC	Pas de connexion
3	Terre fonctionnelle	Pas de connexion
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Remarque

Les entrées de tension inutilisées doivent être court-circuitées.

Les entrées de courant inutilisées doivent être mises sur la plage 0 à 20 mA et/ou désactiver compte rendu d'erreur fil brisé.

Les entrées configurées pour le mode courant ne conduisent pas de boucle de courant sauf si le module est sous tension et configuré.

Les voies d'entrée de courant ne sont pas opérationnelles tant que le transmetteur n'est pas alimenté par une source externe.

A.10.2 Caractéristiques des modules de sorties analogiques SM 1232

Tableau A- 145 Caractéristiques générales

Caractéristiques techniques	SM 1232 AQ 2 x 14 bits	SM 1232 AQ 4 x 14 bits
Référence	6ES7232-4HB32-0XB0	6ES7232-4HD32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	
Poids	180 grammes	
Dissipation de courant	1,8 W	2,0 W
Consommation de courant (bus SM)	80 mA	
Consommation de courant (24 V CC)	45 mA (sans charge)	

Tableau A- 146 Sorties analogiques

Caractéristiques techniques	SM 1232 AQ 2 x 14 bits	SM 1232 AQ 4 x 14 bits
Nombre de sorties	2	4
Type	Tension ou courant	
Plage	± 10 V, 0 à 20 mA ou 4 mA à 20 mA	
Résolution	Tension : 14 bits Courant : 13 bits	
Plage pleine échelle (mot de données)	Tension : -27 648 à 27 648 ; courant : 0 à 27 648 Voir les plages des sorties pour la tension et le courant (Page 1556).	
Précision (25 °C / - 20 à 60 °C)	$\pm 0,3\%$ / $\pm 0,6\%$ de la pleine échelle	
Temps d'établissement (95% de la nouvelle valeur)	Tension : 300 μ s (R), 750 μ s (1 μ F) Courant : 600 μ s (1 mH), 2 ms (10 mH)	
Impédance de charge	Tension : $\geq 1\ 000\ \Omega$ Courant : $\leq 600\ \Omega$	
Courant de court-circuit de sortie max.	Mode tension : ≤ 24 mA Mode courant : $\geq 38,5$ mA	
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Isolation (site à logique)	aucune	
Isolation (24 V à sortie)	aucune	
Longueur de câble (mètres)	100 m, torsadé et blindé	

Tableau A- 147 Diagnostic

Caractéristiques techniques	SM 1232 AQ 2 x 14 bits	SM 1232 AQ 4 x 14 bits
Débordement haut/bas	Oui	
Court-circuit à la terre (mode tension uniquement)	Oui	
Rupture de fil (mode courant uniquement) ¹	Oui	
24 V CC basse tension ²	Oui	

¹ La détection de court-circuit n'est possible que lorsque la tension de sortie est inférieure à -0,5 V ou supérieure à +0,5 V.

² La détection de rupture de fil n'est possible que lorsque le courant de sortie est supérieur à 1 mA.

Tableau A- 148 Schémas de câblage des modules SM de sorties analogiques

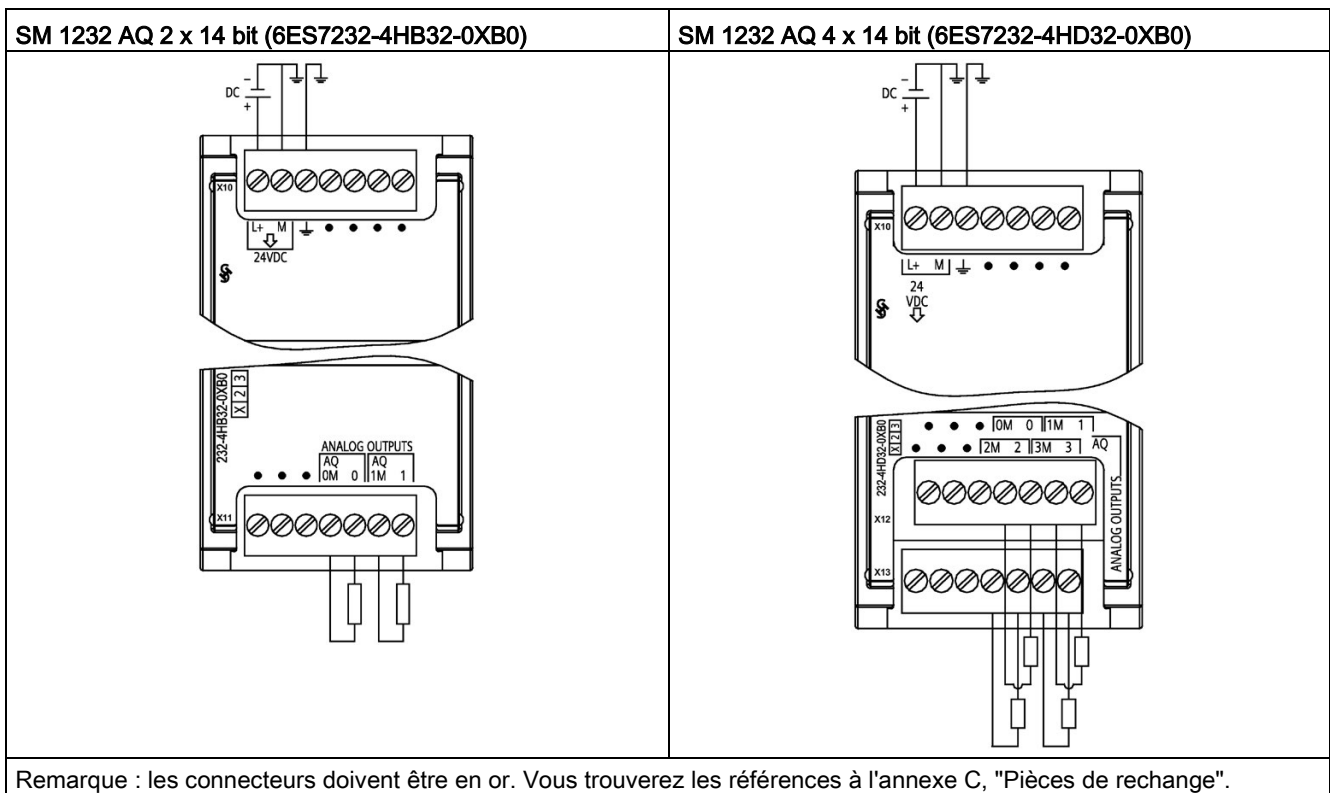


Tableau A- 149 Brochage pour le SM 1232 AQ 2 x 14 bit (6ES7232-4HB32-0XB0)

Broche	X10 (or)	X11 (or)
1	L+/24 V CC	Pas de connexion
2	M/24 V CC	Pas de connexion
3	Terre fonctionnelle	Pas de connexion
4	Pas de connexion	AQ 0M
5	Pas de connexion	AQ 0
6	Pas de connexion	AQ 1M
7	Pas de connexion	AQ 1

Tableau A- 150 Brochage pour le SM 1232 AQ 4 x 14 bit (6ES7232-4HD32-0XB0)

Broche	X10 (or)	X12 (or)	X13 (or)
1	L+/24 V CC	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion
3	Terre fonctionnelle	Pas de connexion	Pas de connexion
4	Pas de connexion	AQ 0M	AQ 2M
5	Pas de connexion	AQ 0	AQ 2
6	Pas de connexion	AQ 1M	AQ 3M
7	Pas de connexion	AQ 1	AIQ 3

A.10.3 Caractéristiques du module d'entrées/sorties analogiques SM 1234

Tableau A- 151 Caractéristiques générales

Caractéristiques techniques	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Référence	6ES7234-4HE32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75
Poids	220 grammes
Dissipation de courant	2,4 W
Consommation de courant (bus SM)	80 mA
Consommation de courant (24 V CC)	60 mA (sans charge)

Tableau A- 152 Entrées analogiques

Modèle	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Nombre d'entrées	4
Type	Tension ou courant (différentiel) : sélectionnable en groupes de 2
Plage	± 10 V, ± 5 V, $\pm 2,5$ V, 0 à 20 mA ou 4 mA à 20 mA
Plage pleine échelle (mot de données)	- 27648 à 27648
Plage de dépassement haut/bas (mot de données)	Tension : 32511 à 27649 / - 27649 à - 32512 Courant : 32511 à 27649 / 0 à - 4864 Voir la partie sur les plages des entrées pour la tension et le courant (Page 1554).
Débordement haut / bas (mot de données)	Tension : 32767 à 32512 / - 32513 à - 32768 Courant : 32767 à 32512 / - 4865 à - 32768 Voir la partie sur les plages des entrées pour la tension et le courant (Page 1554).
Résolution	12 bits + bit de signe
Tension/courant de tenue maximum	± 35 V / ± 40 mA
Lissage	Aucun, faible, moyen ou fort Voir la partie sur les temps de réponse indicielle (Page 1553).
Réjection des bruits	400, 60, 50 ou 10 Hz Voir la partie sur les taux d'échantillonnage (Page 1553).
Impédance d'entrée	≥ 9 M Ω (tension) / ≥ 270 Ω < 290 Ω (courant)
Isolation (site à logique)	Aucune
Précision (25 °C / - 20 à 60 °C)	$\pm 0,1\%$ / $\pm 0,2\%$ de la pleine échelle
Temps de conversion analogique-numérique	625 μ s (réjection 400 Hz)
Réjection en mode commun	40 dB, CC pour 60 Hz
Plage de signal opérationnelle ¹	La tension signal plus mode commun doit être inférieure à +12 V et supérieure à -12 V.
Longueur de câble (mètres)	100 m, torsadé et blindé

¹ Des tensions en dehors de la plage opérationnelle appliquées à une voie peuvent provoquer des interférences sur les autres voies.

Tableau A- 153 Sorties analogiques

Caractéristiques techniques	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Nombre de sorties	2
Type	Tension ou courant
Plage	± 10 V ou 0 à 20 mA ou 4 mA à 20 mA
Résolution	Tension : 14 bits ; courant : 13 bits
Plage pleine échelle (mot de données)	Tension : -27648 à 27648 ; courant : 0 à 27648 Voir la partie sur les plages des sorties pour la tension et le courant (Page 1556).
Précision (25 °C / - 20 à 60 °C)	$\pm 0,3\%$ / $\pm 0,6\%$ de la pleine échelle
Temps d'établissement (95% de la nouvelle valeur)	Tension : 300 μ s (R), 750 μ s (1 μ F) Courant : 600 μ s (1 mH), 2 ms (10 mH)
Impédance de charge	Tension : $\geq 1\,000\ \Omega$ Courant : $\leq 600\ \Omega$
Courant de court-circuit de sortie max.	Mode tension : ≤ 24 mA Mode courant : $\geq 38,5$ mA
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Isolation (site à logique)	aucune
Isolation (24 V à sortie)	aucune
Longueur de câble (mètres)	100 m, torsadé et blindé

Tableau A- 154 Diagnostic

Modèle	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Débordement haut/bas	Oui
Court-circuit à la terre (mode tension uniquement) ¹	Oui sur les sorties
Rupture de fil (mode courant uniquement) ²	Oui sur les sorties
24 V CC basse tension	Oui

¹ La détection de court-circuit n'est possible que lorsque la tension de sortie est inférieure à -0,5 V ou supérieure à +0,5 V.

² La détection de rupture de fil n'est possible que lorsque le courant de sortie est supérieur à 1 mA.

Mesure de courant SM 1234

Vous pouvez réaliser une mesure de courant avec soit un transducteur 2 fils, soit un transducteur 4 fils comme illustré dans la figure suivante :

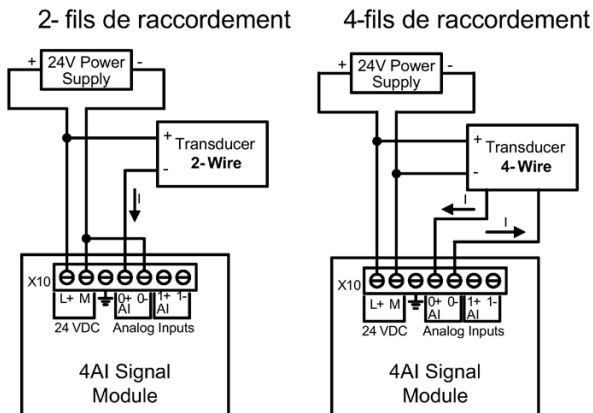


Tableau A- 155 Schémas de câblage du SM d'entrées/sorties analogiques

SM 1234 AI 4 x 13 bit / AQ 2 x 14 bit (6ES7234-4HE32-0XB0)	
<p>Remarque : les connecteurs doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".</p>	

Tableau A- 156 Brochage pour le SM 1234 AI 4 x 13 bit / AQ 2 x 14 bit (6ES7234-4HE32-0XB0)

Broche	X10 (or)	X11 (or)	X13 (or)
1	L+/24 V CC	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion
3	Terre fonctionnelle	Pas de connexion	Pas de connexion
4	AI 0+	AI 2+	AQ 0M
5	AI 0-	AI 2-	AQ 0
6	AI 1+	AI 3+	AQ 1M
7	AI 1-	AI 3-	AQ 1

Remarque

Les entrées de tension inutilisées doivent être court-circuitées.

Les entrées de courant inutilisées doivent être mises sur la plage 0 à 20 mA et/ou désactiver compte rendu d'erreur fil brisé.

Les entrées configurées pour le mode courant ne conduisent pas de boucle de courant sauf si le module est sous tension et configuré.

Les voies d'entrée de courant ne sont pas opérationnelles tant que le transmetteur n'est pas alimenté par une source externe.

A.10.4 Réponse indicielle des entrées analogiques

Tableau A- 157 Réponse indicielle (ms), 0 à pleine échelle mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Réduction des bruits/fréquence de réjection (sélection de temps d'intégration)			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Aucun (1 cycle) : Pas de moyennage	4 ms	18 ms	22 ms	100 ms
Faible (4 cycles) : 4 échantillons	9 ms	52 ms	63 ms	320 ms
Moyen (16 cycles) : 16 échantillons	32 ms	203 ms	241 ms	1200 ms
Fort (32 cycles) : 32 échantillons	61 ms	400 ms	483 ms	2410 ms

A.10.5 Temps d'échantillonnage et temps d'actualisation pour les entrées analogiques

Tableau A- 158 Temps d'échantillonnage et temps d'actualisation pour toutes les voies

Fréquence de réjection (temps d'intégration)	Temps d'échantillonnage et temps d'actualisation pour toutes les voies			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
SM 4 voies x 13 bits	0,625 ms	4,17 ms	5 ms	25 ms
SM 8 voies x 13 bits	1,25 ms	4,17 ms	5 ms	25 ms
SM 4 voies x 16 bits	0,417 ms	0,397 ms	0,400 ms	0,400 ms

A.10.6 Plages de mesures d'entrées analogiques pour tension et courant (SB et SM)

Tableau A- 159 Représentation des entrées analogiques pour la tension (SB et SM)

Technologie		Plage de mesure de tension				
Décimal	Hexadécimal	± 10 V	± 5 V	± 2,5 V	± 1,25 V	
32767	7FFF ¹	11,851 V	5,926 V	2,963 V	1,481 V	Débordement haut
32512	7F00					
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Plage de dépassement haut
27649	6C01					
27648	6C00	10 V	5 V	2,5 V	1,250 V	Plage nominale
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V	
1	1	361,7 µV	180,8 µV	90,4 µV	45,2 µV	
0	0	0 V	0 V	0 V	0 V	
-1	FFFF					
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V	
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V	Plage de dépassement bas
-27649	93FF					
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	Débordement bas
-32513	80FF					
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V	

¹ 7FFF peut être renvoyé pour l'une des raisons suivantes : débordement (comme noté dans ce tableau), avant que des valeurs valides soient disponibles (par exemple immédiatement lors d'un amorçage du système), ou si une rupture de fil est détectée.

Tableau A- 160 Représentation des entrées analogiques pour le courant (SB et SM)

Technologie		Plage de mesure de courant		
Décimal	Hexadécimal	0 mA à 20 mA	4 mA à 20 mA	
32767	7FFF	> 23,52 mA	> 22,81 mA	Débordement haut
32511	7EFF	23,52 mA	22,81 mA	Plage de dépassement haut
27649	6C01			
27648	6C00	20 mA	20 mA	Plage nominale
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			
-4864	ED00	-3,52 mA	1,185 mA	Plage de dépassement bas
32767 ¹	7FFF		< 1,185 mA	Rupture de fil (4 à 20 mA)
-32768	8000	< -3,52 mA		Débordement bas (0 à 20 mA)

¹ La valeur de rupture de fil 32767 (16#7FFF) est toujours renvoyée quel que soit l'état de l'alarme de rupture de fil.

Voir aussi

Détermination du type de condition de rupture de fil sur un module SM 1231 (Page 1420)

A.10.7 Plages de mesures de sorties analogiques pour tension et courant (SB et SM)

Tableau A- 161 Représentation de sorties analogiques pour tension (SB et SM)

Système		Plage de sortie de tension	
Décimal	Hexadécimal	± 10 V	
32767	7FFF	Voir note 1	Débordement haut
32512	7F00	Voir note 1	
32511	7EFF	11,76 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
1	1	361,7 µV	
0	0	0 V	
-1	FFFF	-361,7 µV	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		
-32512	8100	-11,76 V	Plage de dépassement bas
-32513	80FF	Voir note 1	
-32768	8000	Voir note 1	

¹ À l'état débordement haut ou bas, les sorties analogiques prendront la valeur de remplacement de l'état de fonctionnement Arrêt.

Tableau A- 162 Représentation de sorties analogiques pour courant (SB et SM)

Système		Plage de sortie du courant		
Décimal	Hexadécimal	0 mA à 20 mA	4 mA à 20 mA	
32767	7FFF	Voir note 1	Voir note 1	Débordement haut
32512	7F00	Voir note 1	Voir note 1	
32511	7EFF	23,52 mA	22,81 mA	Plage de dépassement haut
27649	6C01			
27648	6C00	20 mA	20 mA	Plage nominale
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4mA	
-1	FFFF		4 mA à 578,7 nA	Plage de dépassement bas
-6912	E500		0 mA	
-6913	E4FF			Impossible. Valeur de sortie limitée à 0 mA.
-32512	8100			
-32513	80FF	Voir note 1	Voir note 1	Débordement bas
-32768	8000	Voir note 1	Voir note 1	

¹ À l'état débordement haut ou bas, les sorties analogiques prendront la valeur de remplacement de l'état de fonctionnement Arrêt.

A.11 Modules d'entrées-sorties (SM) Thermocouple et RTD

A.11.1 SM 1231 Thermocouple

Tableau A- 163 Caractéristiques générales

Modèle	SM 1231 AI 4 x 16 bits TC	SM 1231 AI 8 x 16 bits TC
Référence	6ES7231-5QD32-0XB0	6ES7231-5QF32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	
Poids	180 grammes	190 grammes
Dissipation de courant	1,5 W	
Consommation de courant (bus SM)	80 mA	
Consommation de courant (24 V CC) ¹	40 mA	

¹ 20,4 à 28,8 V CC (classe 2, puissance limitée ou alimentation de capteur de l'API)

Tableau A- 164 Entrées analogiques

Modèle	SM 1231 AI 4 x 16 bits TC	SM 1231 AI 8 x 16 bits TC
Nombre d'entrées	4	8
Plage Plage nominale (mot de données) Dépassement haut/bas (mot de données) Débordement haut / bas (mot de données)	Voir le tableau de sélection pour thermocouples (Page 1563).	
Résolution	Température	0,1 °C / 0,1 °F
	Tension	15 bits plus signe
Tension de tenue maximum	± 35 V	
Réjection des bruits	85 dB pour le réglage de filtre sélectionné (10 Hz, 50 Hz, 60 Hz ou 400 Hz)	
Réjection en mode commun	> 120 dB à 120 V CA	
Impédance	≥ 10 MΩ	
Isolation	Site à logique	707 V CC (test de type)
	Site à 24 V CC	707 V CC (essai de type)
	24 V CC à logique	707 V CC (essai de type)
Voie à voie	120 V CA	
Précision	Voir le tableau de sélection pour thermocouples (Page 1563).	
Répétabilité	± 0,05% pleine échelle	
Principe de mesure	Intégration	
Temps d'actualisation du module	Voir le tableau de sélection pour la réduction des bruits (Page 1563).	
Erreur de soudure froide	± 1,5 °C	
Longueur de câble (mètres)	max. 100 mètres au capteur	
Résistance de fil	100 Ω max.	

Tableau A- 165 Diagnostic

Modèle	SM 1231 AI 4 x 16 bits TC	SM 1231 AI 8 x 16 bits TC
Débordement haut/bas ¹	Oui	
Rupture de fil ^{2,3}	Oui	
24 V CC basse tension ¹	Oui	

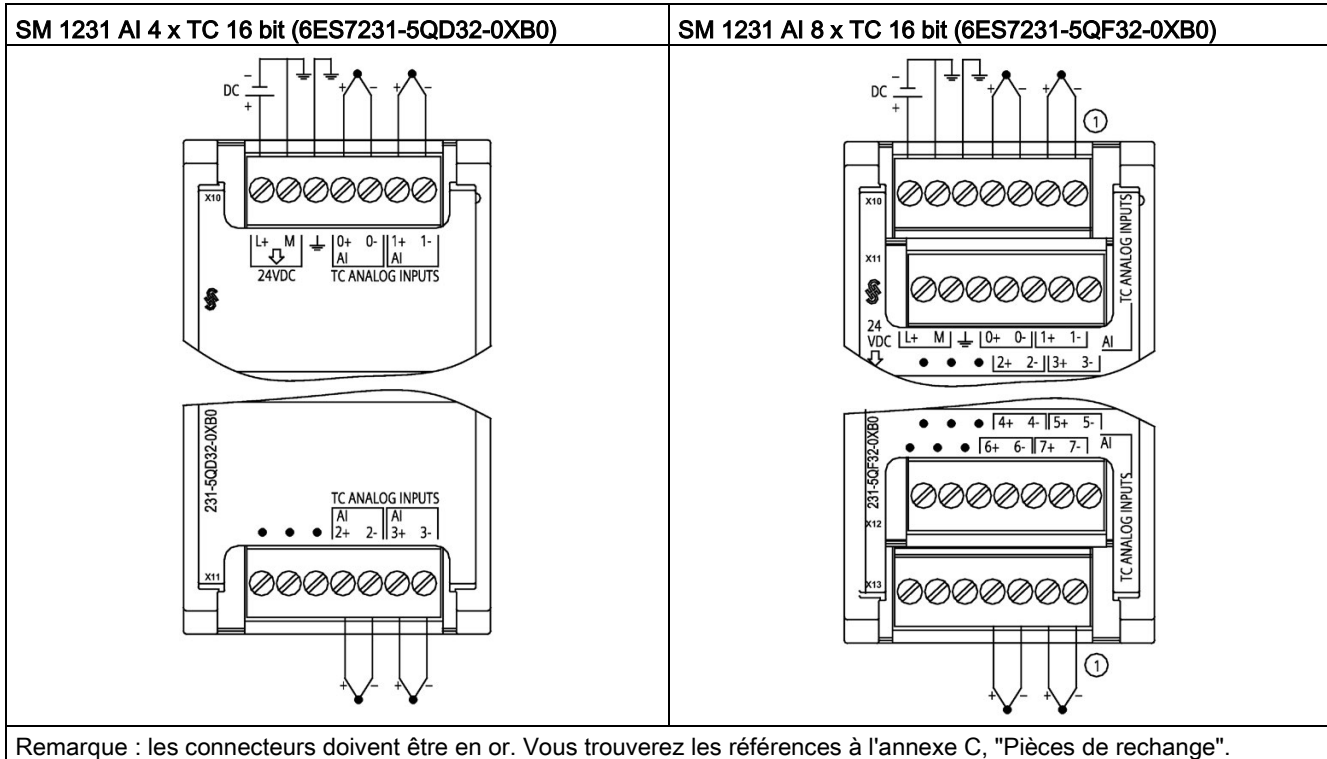
- ¹ Les informations d'alarmes de diagnostic débordement haut, débordement bas et basse tension sont notifiées dans les valeurs de données analogiques même si les alarmes sont désactivées dans la configuration du module.
- ² Lorsque l'alarme de rupture de fil est désactivée et qu'un fil est ouvert dans le câblage du capteur, le module peut émettre des valeurs aléatoires.
- ³ Le module réalise un test de rupture de fil toutes les 6 secondes, qui étend le délai de mise à jour de 9 ms pour chaque voie de validation une fois toutes les 6 secondes.

A.11 Modules d'entrées-sorties (SM) Thermocouple et RTD

Le module d'entrées analogiques SM 1231 Thermocouple (TC) mesure la valeur de tension connectée aux entrées du module. Le type de mesure de température peut être soit "Thermocouple", soit "Tension".

- "Thermocouple" : La valeur sera notifiée en degrés multipliés par dix (par exemple, 25,3 degrés seront indiqués sous la forme 253 en décimal).
- "Tension" : La valeur pleine échelle de la plage nominale sera 27648 en décimal.

Tableau A- 166 Schémas de câblage pour les SM Thermocouple



① Pour plus de clarté, les TC 2, 3, 4 et 5 ne sont pas connectés sur l'illustration.

Tableau A- 167 Brochage pour le SM 1231 AI 4 x TC 16 bit (6ES7231-5QD32-0XB0)

Broche	X10 (or)	X11 (or)
1	L+/24 V CC	Pas de connexion
2	M/24 V CC	Pas de connexion
3	Terre fonctionnelle	Pas de connexion
4	AI 0+ /TC	AI 2+ /TC
5	AI 0- /TC	AI 2- /TC
6	AI 1+ /TC	AI 3+ /TC
7	AI 1- /TC	AI 3- /TC

Tableau A- 168 Brochage pour le SM 1231 AI 8 x TC 16 bit (6ES7231-5QF32-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	Terre fonctionnelle	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0+ /TC	AI 2+ /TC	AI 4+ /TC	AI 6+ /TC
5	AI 0- /TC	AI 2- /TC	AI 4- /TC	AI 6- /TC
6	AI 1+ /TC	AI 3+ /TC	AI 5+ /TC	AI 7+ /TC
7	AI 1- /TC	AI 3- /TC	AI 5- /TC	AI 7- /TC

Remarque

Les entrées analogiques inutilisées doivent être court-circuitées.

Les voies thermocouple inutilisées peuvent être désactivées. Aucune erreur ne se produira si une voie inutilisée est désactivée.

A.11.1.1 Principe de fonctionnement des thermocouples

Un thermocouple est composé de deux conducteurs métalliques de nature différente, reliés électriquement par soudure l'un à l'autre. Une tension proportionnelle à la température de la soudure froide est générée. Cette tension est faible ; un microvolt peut représenter beaucoup de degrés. Mesurer la tension provenant d'un thermocouple, compenser les soudures supplémentaires, puis linéariser le résultat constituent le principe fondamental des mesures de température à l'aide de thermocouples.

Lorsque vous raccordez un thermocouple au module SM 1231 Thermocouple, les deux fils métalliques de nature différente sont reliés au module, au niveau du connecteur de signaux du module. L'endroit où les deux fils sont reliés l'un à l'autre constitue le point de mesure du thermocouple.

Deux autres thermocouples sont constitués à l'endroit où les deux fils de nature différente sont raccordés au connecteur de signaux. La température du connecteur provoque une tension qui s'ajoute à la tension du thermocouple capteur. Si l'on ne corrige pas cette tension, la température indiquée sera différente de la température du capteur.

On utilise donc la compensation de soudure froide pour compenser le thermocouple du connecteur. Les tables de données de thermocouples se basent sur une température de soudure de référence, 0 degré Celsius en général. La compensation de soudure froide compense le connecteur à 0 degré Celsius. Elle restaure la tension ajoutée par les thermocouples du connecteur. La température du module est mesurée en interne, puis convertie en une valeur à ajouter à la conversion de capteur. On linéarise ensuite la conversion de capteur corrigée à l'aide des tables de thermocouples.

Pour un fonctionnement optimal de la compensation de soudure froide, le module Thermocouple doit se trouver dans un environnement thermique stable. Une variation lente (inférieure à 0,1 °C/minute) à une température ambiante du module est correctement compensée dans les limites des spécifications du module. Un déplacement d'air à travers le module entraînera également des erreurs de compensation de la soudure froide.

Si une meilleure compensation d'erreur de soudure froide est souhaitée, un bornier isotherme externe peut être utilisé. Le module thermocouple met à disposition un bornier référencé à 0 °C ou à 50 °C.

A.11.1.2 Tableaux de sélection pour les SM 1231 Thermocouple

Les plages et la précision des différents types de thermocouple pris en charge par le module d'entrées SM 1231 Thermocouple sont présentées dans le tableau ci-dessous.

Tableau A- 169 Tableau de sélection de thermocouples

Type	Dépassement bas minimum ¹	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum ²	Précision plage normale ^{3,4} @ 25 °C	Précision plage normale ^{1,2} - 20 °C à 60 °C
J	- 210,0 °C	- 150,0 °C	1 200,0 °C	1 450,0 °C	± 0,3 °C	± 0,6 °C
	- 346,0 °F	- 238,0 °F	2192,0 °F	2642,0 °F	± 0,5 °F	± 1,1 °F
K	- 270,0 °C	- 200,0 °C	1 372,0 °C	1 622,0 °C	± 0,4 °C	± 1,0 °C
	- 454,0 °F	- 328,0 °F	2501,6 °F	2951,6 °F	± 0,7 °F	± 1,8 °F
T	- 270,0 °C	- 200,0 °C	400,0 °C	540,0 °C	± 0,5 °C	± 1,0 °C
	- 454,0 °F	- 328,0 °F	752,0 °F	1004,0 °F	± 0,9 °F	± 1,8 °F
E	- 270,0 °C	- 200,0 °C	1 000,0 °C	1 200,0 °C	± 0,3 °C	± 0,6 °C
	- 454,0 °F	- 328,0 °F	1832,0 °F	2192,0 °F	± 0,5 °F	± 1,1 °F
R & S	- 50,0 °C	100,0 °C	1 768,0 °C	2 019,0 °C	± 1,0 °C	± 2,5 °C
	- 58,0 °F	212,0 °F	3214,4 °F	3276,6 °F ⁵	± 1,8 °F	± 4,5 °F
B	0,0 °C	200,0 °C	800,0 °C	--	± 2,0 °C	± 2,5 °C
	32,0 °F	392,0 °F	1472,0 °F	--	± 3,6 °F	± 4,5 °F
	--	800,0 °C	1 820,0 °C	1 820,0 °C	± 1,0 °C	± 2,3 °C
	--	1472,0 °F	3276,6 °F ⁵	3276,6 °F ⁵	± 1,8 °F	± 4,1 °F
N	- 270,0 °C	- 200,0 °C	1 300,0 °C	1 550,0 °C	± 1,0 °C	± 1,6 °C
	- 454,0 °F	- 328,0 °F	2372,0 °F	2822,0 °F	± 1,8 °F	± 2,9 °F
C	0,0 °C	100,0 °C	2 315,0 °C	2 500,0 °C	± 0,7 °C	± 2,7 °C
	32,0 °F	212,0 °F	3276,6 °F ⁵	3276,6 °F ⁵	± 1,3 °F	± 4,9 °F
TXK/XK(L)	- 200,0 °C	- 150,0 °C	800,0 °C	1 050,0 °C	± 0,6 °C	± 1,2 °C
	- 328,0 °F	302,0 °F	1472,0 °F	1922,0 °F	± 1,1 °F	± 2,2 °F
Tension	- 32512	- 27648 - 80 mV	27648 80 mV	32511	± 0,05 %	± 0,1 %

- 1 Les valeurs de thermocouple inférieures à la valeur de dépassement bas minimum sont signalées en tant que -32768.
- 2 Les valeurs de thermocouple supérieures à la valeur de dépassement haut maximum sont signalées en tant que 32767.
- 3 L'erreur de soudure froide interne est de ± 1,5 °C pour toutes les plages. Cette valeur s'ajoute à l'erreur indiquée dans ce tableau. Le module requiert au minimum 30 minutes d'échauffement pour atteindre cette spécification.
- 4 La précision du SM 1231 AI 4 x 16 bits TC peut être réduite en présence de radiofréquence rayonnée entre 970 MHz et 990 MHz.
- 5 Limite inférieure de 3276.6 pour la signalisation en °F

Remarque

Voie du thermocouple

Chaque voie du module d'entrées-sorties Thermocouple peut être configurée avec un type de thermocouple différent (sélectionnable dans le logiciel pendant la configuration du module).

Tableau A- 170 Réduction des bruits et temps d'actualisation pour le SM 1231 Thermocouple

Sélection de la fréquence de réjection	Temps d'intégration	Temps d'actualisation module 4 voies (secondes)	Temps d'actualisation module 8 voies (secondes)
400 Hz (2,5 ms)	10 ms ¹	0,143	0,285
60 Hz (16,6 ms)	16,67 ms	0,223	0,445
50 Hz (20 ms)	20 ms	0,263	0,525
10 Hz (100 ms)	100 ms	1,225	2,450

¹ Pour conserver la précision et la résolution du module lorsque la réjection 400 Hz est sélectionnée, le temps d'intégration est de 10 ms. Avec cette sélection, on a également une réjection des bruits 100 Hz et 200 Hz.

Il est recommandé d'utiliser un temps d'intégration de 100 ms pour la mesure des thermocouples. Le choix de temps d'intégration inférieurs augmentera l'erreur de répétabilité des températures lues.

Remarque

Après sa mise sous tension, le module exécute un calibrage interne pour le convertisseur analogique-numérique. Pendant ce temps, il émet une valeur de 32767 sur chaque voie jusqu'à ce que des données valables soient disponibles sur cette voie. Il peut s'avérer nécessaire que votre programme utilisateur prévoie ce temps d'initialisation. Comme la configuration du module peut faire varier la longueur du temps d'initialisation, vous devez vérifier le comportement du module dans votre configuration. Si nécessaire, vous pouvez inclure de la logique dans votre programme utilisateur afin de prendre en compte le temps d'initialisation du module.

Représentation des valeurs analogiques des thermocouples de type J

Le tableau ci-dessous donne la représentation des valeurs analogiques des thermocouples de type J.

Tableau A- 171 Représentation des valeurs analogiques des thermocouples de type J

Type J en °C	Unités		Type J en °F	Unités		Plage
	Décimal	Hexadécimal		Décimal	Hexadécimal	
> 1450.0	32767	7FFF	> 2642.0	32767	7FFF	Déborde- ment haut
1450.0	14500	38A4	2642.0	26420	6734	Dépasse- ment haut
:	:	:	:	:	:	
1200.1	12001	2EE1	2192.2	21922	55A2	
1200.0	12000	2EE0	2192.0	21920	55A0	Plage nomi- nale
:	:	:	:	:	:	
-150.0	-1500	FA24	-238.0	-2380	F6B4	
< -150.0	-32768	8000	< -238.0	-32768	8000	Déborde- ment bas ¹

¹ Le module thermocouple pourrait signaler un débordement bas en cas de câblage défectueux (par exemple, inversion de polarité ou entrées ouvertes) ou d'erreur de capteur dans la plage négative (par exemple, mauvais type de thermocouple).

A.11.2 SM 1231 RTD

Caractéristiques des SM 1231 RTD

Tableau A- 172 Caractéristiques générales

Caractéristiques techniques	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Référence	6ES7231-5PD32-0XB0	6ES7231-5PF32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	70 x 100 x 75
Poids	220 grammes	270 grammes
Dissipation de courant	1,5 W	
Consommation de courant (bus SM)	80 mA	90 mA
Consommation de courant (24 V CC) 1	40 mA	

1 20,4 à 28,8 V CC (classe 2, puissance limitée ou alimentation de capteur de la CPU)

Tableau A- 173 Entrées analogiques

Caractéristiques techniques	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Nombre d'entrées	4	8
Type	RTD référencée au module et ohms	
Plage Plage nominale (mot de données) Plage de dépassement haut/bas (mot de données) Débordement haut / bas (mot de données)	Voir le tableau de sélection des capteurs RTD (Page 1570).	
Résolution	Température	0,1 °C/0,1 °F
	Résistance	15 bits plus signe
Tension de tenue maximum	± 35 V	
Réjection des bruits	85 dB pour la réduction des bruits sélectionnée (10 Hz, 50 Hz, 60 Hz ou 400 Hz)	
Réjection en mode commun	> 120 dB	
Impédance	≥ 10 MΩ	
Isolation	Site à logique	707 V CC (essai de type)
	Site à 24 V CC	707 V CC (essai de type)
	24 V CC à logique	707 V CC (essai de type)
Isolation voie à voie	aucune	
Précision	Voir le tableau de sélection des capteurs RTD (Page 1570).	
Répétabilité	± 0,05% pleine échelle	
Dissipation maximale du capteur	0,5 mW	
Principe de mesure	Intégration	
Temps d'actualisation du module	Voir le tableau de sélection pour la réduction des bruits (Page 1570).	

Caractéristiques techniques	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Longueur de câble (mètres)	max. 100 mètres au capteur	
Résistance de fil	20 Ω , 2,7 Ω pour RTD 10 Ω max.	

Tableau A- 174 Diagnostic

Caractéristiques techniques	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Débordement haut/bas ^{1,2}	Oui	
Rupture de fil ³	Oui	
24 V CC basse tension ¹	Oui	

¹ Les informations d'alarmes de diagnostic débordement haut, débordement bas et basse tension sont notifiées dans les valeurs de données analogiques même si les alarmes sont désactivées dans la configuration du module.

² La détection de débordement bas n'est jamais activée pour les plages de résistance.

³ Lorsque l'alarme de rupture de fil est désactivée et qu'un fil est ouvert dans le câblage du capteur, le module peut émettre des valeurs aléatoires.

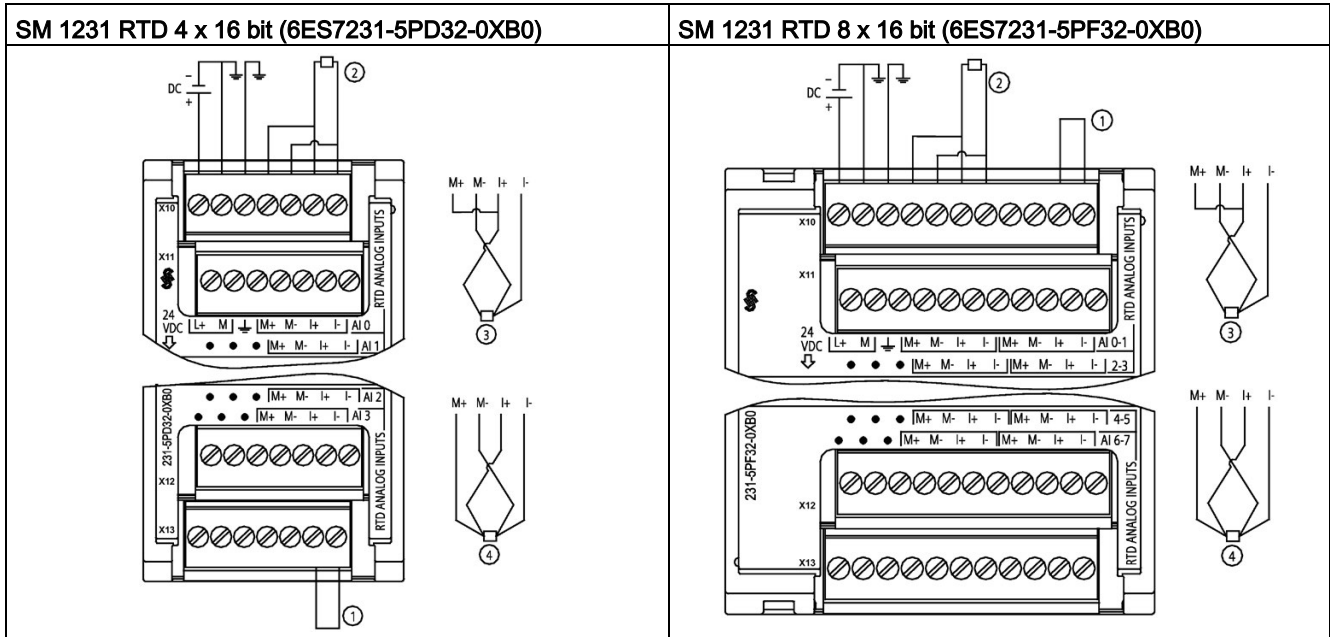
Le module d'entrées-sorties analogiques SM 1231 RTD mesure la valeur de résistance connectée aux entrées du module. Il est possible de sélectionner "Résistance" ou "Thermistance" comme type de mesure.

- "Résistance" : La valeur pleine échelle de la plage nominale sera 27648 en décimal.
- "Thermistance" : La valeur sera notifiée en degrés multipliés par dix (par exemple, 25,3 degrés seront indiqués sous la forme 253 en décimal). La plage des valeurs climatiques sera notifiée en degrés multipliés par cent (par exemple, 25,34 degrés seront indiqués sous la forme 2534 en décimal).

A.11 Modules d'entrées-sorties (SM) Thermocouple et RTD

Le module SM 1231 RTD prend en charge des montages 2 fils, 3 fils et 4 fils avec la résistance du capteur.

Tableau A- 175 Schémas de câblage pour les SM RTD



- ① Bouclage sur les entrées RTD inutilisées
- ② RTD 2 fils ③ RTD 3 fils ④ RTD 4 fils

REMARQUE : les connecteurs doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Tableau A- 176 Brochage pour le SM 1231 RTD 4 x 16 bit (6ES7231-5PD32-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	Terre fonctionnelle	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0 M+ /RTD	AI 1 M+ /RTD	AI 2 M+ /RTD	AI 3 M+ /RTD
5	AI 0 M- /RTD	AI 1 M- /RTD	AI 2 M- /RTD	AI 3 M- /RTD
6	AI 0 I+ /RTD	AI 1 I+ /RTD	AI 2 I+ /RTD	AI 3 I+ /RTD
7	AI 0 I- /RTD	AI 1 I- /RTD	AI 2 I- /RTD	AI 3 I- /RTD

Tableau A- 177 Brochage pour le SM 1231 RTD 8 x 16 bit (6ES7231-5PF32-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	Terre fonctionnelle	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0 M+ /RTD	AI 2 M+ /RTD	AI 4 M+ /RTD	AI 6 M+ /RTD
5	AI 0 M- /RTD	AI 2 M- /RTD	AI 4 M- /RTD	AI 6 M- /RTD
6	AI 0 I+ /RTD	AI 2 I+ /RTD	AI 4 I+ /RTD	AI 6 I+ /RTD
7	AI 0 I- /RTD	AI 2 I- /RTD	AI 4 I- /RTD	AI 6 I- /RTD
8	AI 1 M+ /RTD	AI 3 M+ /RTD	AI 5 M+ /RTD	AI 7 M+ /RTD
9	AI 1 M- /RTD	AI 3 M- /RTD	AI 5 M- /RTD	AI 7 M- /RTD
10	AI 1 I+ /RTD	AI 3 I+ /RTD	AI 5 I+ /RTD	AI 7 I+ /RTD
11	AI 1 I- /RTD	AI 3 I- /RTD	AI 5 I- /RTD	AI 7 I- /RTD

Remarque

Les voies RTD inutilisées peuvent être désactivées. Aucune erreur ne se produira si une voie inutilisée est désactivée.

La boucle de courant du module RTD doit être continue pour éliminer le temps de stabilisation supplémentaire qui est automatiquement ajouté à une voie inutilisée qui n'est pas désactivée. Pour des raisons de cohérence, le module RTD doit avoir une résistance connectée (comme le montage RTD 2 fils).

A.11.2.1 Tableaux de sélection pour les SM 1231 RTD

Tableau A- 178 Plages et précision pour les différents capteurs pris en charge par les modules RTD

Coefficient de température	Type de RTD	Dépassement bas minimum ¹	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum ²	Précision de plage normale à 25 °C	Précision de plage normale de - 20 °C à 60 °C
Pt 0.003850 ITS90 DIN EN 60751	Pt 100 plage climatique	- 145,00 °C	- 120,00 °C	145,00 °C	155,00 °C	± 0,20 °C	± 0,40 °C
	Pt 10	- 243,0 °C	- 200,0 °C	850,0 °C	1 000,0 °C	± 1,0 °C	± 2,0 °C
	Pt 50	- 243,0 °C	- 200,0 °C	850,0 °C	1 000,0 °C	± 0,5 °C	± 1,0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0.003902 Pt 0.003916 Pt 0.003920	Pt 100	- 243,0 °C	- 200,0 °C	850,0 °C	1 000,0 °C	± 0,5 °C	± 1,0 °C
	Pt 200	- 243,0 °C	- 200,0 °C	850,0 °C	1 000,0 °C	± 0,5 °C	± 1,0 °C
	Pt 500	- 243,0 °C	- 200,0 °C	850,0 °C	1 000,0 °C	± 0,5 °C	± 1,0 °C
	Pt 1000						
Pt 0,003910	Pt 10	- 273,2 °C	- 240,0 °C	1 100,0 °C	1 295 °C	± 1,0 °C	± 2,0 °C
	Pt 50	- 273,2 °C	- 240,0 °C	1 100,0 °C	1 295 °C	± 0,8 °C	± 1,6 °C
	Pt 100	- 273,2 °C	- 240,0 °C	1 100,0 °C	1 295 °C	± 0,8 °C	± 1,6 °C
	Pt 500						
Ni 0.006720 Ni 0.006180	Ni 100	- 105,0 °C	- 60,0 °C	250,0 °C	295,0 °C	± 0,5 °C	± 1,0 °C
	Ni 120	- 105,0 °C	- 60,0 °C	250,0 °C	295,0 °C	± 0,5 °C	± 1,0 °C
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0.005000	LG-Ni 1000	- 105,0 °C	- 60,0 °C	250,0 °C	295,0 °C	± 0,5 °C	± 1,0 °C
Ni 0.006170	Ni 100	- 105,0 °C	- 60,0 °C	180,0 °C	212,4 °C	± 0,5 °C	± 1,0 °C
Cu 0.004270	Cu 10	- 240,0 °C	- 200,0 °C	260,0 °C	312,0 °C	± 1,0 °C	± 2,0 °C
Cu 0.004260	Cu 10	- 60,0 °C	- 50,0 °C	200,0 °C	240,0 °C	± 1,0 °C	± 2,0 °C
	Cu 50	- 60,0 °C	- 50,0 °C	200,0 °C	240,0 °C	± 0,6 °C	± 1,2 °C
	Cu 100	- 60,0 °C	- 50,0 °C	200,0 °C	240,0 °C	± 0,6 °C	± 1,2 °C
Cu 100							
Cu 0.004280	Cu 10	- 240,0 °C	- 200,0 °C	200,0 °C	240,0 °C	± 1,0 °C	± 2,0 °C
	Cu 50	- 240,0 °C	- 200,0 °C	200,0 °C	240,0 °C	± 0,7 °C	± 1,4 °C
	Cu 100	- 240,0 °C	- 200,0 °C	200,0 °C	240,0 °C	± 0,7 °C	± 1,4 °C
Cu 100							

¹ Les valeurs RTD inférieures à la valeur de dépassement bas minimum sont signalées en tant que -32768.

² Les valeurs RTD supérieures à la valeur de dépassement haut maximum sont signalées en tant que +32767.

Tableau A- 179 Résistance

Plage	Dépassement bas minimum	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum ¹	Précision de plage normale à 25 °C	Précision de plage normale de - 20 °C à 60 °C
150 Ω	Sans objet	0 (0 Ω)	27648 (150 Ω)	176,383 Ω	± 0,05 %	± 0,1 %
300 Ω	Sans objet	0 (0 Ω)	27648 (300 Ω)	352,767 Ω	± 0,05 %	± 0,1 %
600 Ω	Sans objet	0 (0 Ω)	27648 (600 Ω)	705,534 Ω	± 0,05 %	± 0,1 %

¹ Les valeurs de résistance supérieures à la valeur de dépassement haut maximum sont signalées comme +32767.

Remarque

Le module signale 32767 sur toute voie activée à laquelle aucun capteur n'est connecté. Si la détection de fil ouvert est également activée, le module fait clignoter les DEL appropriées.

Si des plages RTD 500 Ω et 1000 Ω sont utilisées avec d'autres résistances à valeur inférieure, l'erreur peut atteindre deux fois l'erreur indiquée.

La meilleure précision est obtenue pour les plages RTD 10 Ω avec des montages 4 fils.

La résistance des fils de connexion en montage 2 fils provoquera une erreur dans la lecture du capteur. La précision ne peut donc pas être garantie.

Tableau A- 180 Réduction des bruits et temps d'actualisation pour les modules RTD

Sélection de la fréquence de réjection	Temps d'intégration	Temps d'actualisation (secondes)	
		module 4 voies	module 8 voies
400 Hz (2,5 ms)	10 ms ¹	4 / 2 fils : 0,142 3 fils : 0,285	4 / 2 fils : 0,285 3 fils : 0,525 0,525
60 Hz (16,6 ms)	16,67 ms	4 / 2 fils : 0,222 3 fils : 0,445 0,445	4 / 2 fils : 0,445 3 fils : 0,845 0,845
50 Hz (20 ms)	20 ms	4 / 2 fils : 0,262 3 fils : 0,505 0,505	4 / 2 fils : 0,524 3 fils : 1,015 1,015
10 Hz (100 ms)	100 ms	4 / 2 fils : 1,222 3 fils : 2,445 2,445	4 / 2 fils : 2,425 3 fils : 4,845

¹ Le temps d'intégration est de 10 ms pour conserver la résolution et la précision du module lorsque le filtre 400 Hz est sélectionné. Cette sélection correspond également à une réjection des bruits 100 Hz et 200 Hz.

Remarque

Après sa mise sous tension, le module exécute un calibrage interne pour le convertisseur analogique-numérique. Pendant ce temps, il émet une valeur de 32767 sur chaque voie jusqu'à ce que des données valables soient disponibles sur cette voie. Il peut s'avérer nécessaire que votre programme utilisateur prévoie ce temps d'initialisation. Comme la configuration du module peut faire varier la longueur du temps d'initialisation, vous devez vérifier le comportement du module dans votre configuration. Si nécessaire, vous pouvez inclure de la logique dans votre programme utilisateur afin de prendre en compte le temps d'initialisation du module.

Représentation des valeurs analogiques pour RTD

Les tableaux ci-dessous donnent la représentation de la valeur mesurée numérisée pour les sondes RTD dans la plage de température standard.

Tableau A- 181 Représentation des valeurs analogiques pour les thermomètres à résistance PT 100, 200, 500, 1000 et PT 10, 50, 100, 500 GOST (0.003850) standard

Pt x00 standard en °C (1 chiffre = 0,1 °C)	Unités		Pt x00 standard en °F (1 chiffre = 0,1 F)	Unités		Plage
	Décimal	Hexadécimal		Décimal	Hexadécimal	
> 1000,0	32767	7FFF	> 1832,0	32767	7FFF	Débordement haut
1000,0	10000	2710	1832,0	18320	4790	Dépassement haut
:	:	:	:	:	:	
850,1	8501	2135	1562,1	15621	3D05	
850,0	8500	2134	1562,0	15620	3D04	Plage nominale
:	:	:	:	:	:	
- 200,0	- 2000	F830	- 328,0	- 3280	F330	
- 200,1	- 2001	F82F	- 328,1	- 3281	F32F	Dépassement bas
:	:	:	:	:	:	
- 243,0	- 2430	F682	- 405,4	- 4054	F02A	
< -243,0	- 32768	8000	< -405,4	- 32768	8000	Débordement bas

A.12 Modules technologiques

A.12.1 SM 1278 4 x IO-Link maître

Tableau A- 182 Caractéristiques générales

Caractéristiques techniques		Module d'entrées-sorties du Maître SM 1278 4xIO-Link
Référence		6ES7278-4BD32-0XB0
Dimensions L x H x P (mm)		45 x 100 x 75
Poids		150 grammes
Informations générales		
	données I&M	Oui ; IM0 à IM3
Tension d'alimentation		
	Tension nominale (CC)	24 V CC
	Limite inférieure de plage valide (CC)	19,2 V ; 20,5 V si IO-Link est utilisé (la tension d'alimentation pour les appareils IO-Link sur le maître doit être au moins de 20 V)
	Limite supérieure de plage valide (CC)	28,8 V CC
	Protection contre l'inversion de polarité	Oui
Courant d'entrée		
	Consommation de courant	65 mA ; sans charge
Alimentation du codeur		
	Nombre de sorties	4
	Courant de sortie, valeur nominale	200 mA
Perte de puissance		
	Perte de puissance, typ.	1 W, à l'exclusion du chargement de port
Entrées/sorties TOR		
	Longueur de câble (mètres)	20 m, non blindé, max.
SDLC		
	Longueur de câble (mètres)	20 m, non blindé, max.
IO-Link		
	Nombre de ports	4
	Nombre de ports qui peuvent être commandés en même temps	4
	protocole 1.0 IO-Link	Oui
	protocole 1.1 IO-Link	Oui
Etat de fonctionnement		
	IO-Link	Oui
	DI	Oui
	DQ	Oui ; max. 100 mA

Caractéristiques techniques		Module d'entrées-sorties du Maître SM 1278 4xIO-Link
Raccordement d'appareils IO-Link		
Port type A		Oui
Vitesse de transmission		4,8 kBd (COM1)
		38,4 kBd (COM2)
		230,4 kBd (COM3)
Temps de cycle, min.		2 ms ; dynamique, dépend de la longueur des données de l'utilisateur
Taille des données de traitement, entrée par port		32 octets ; max.
Taille des données de traitement, entrée par module		32 octets
Taille des données de traitement, sortie par port		32 octets ; max.
Taille des données de traitement, sortie par module		32 octets
Taille mémoire pour les paramètres d'appareil		2 Ko
Longueur de câble non blindé, max. (mètres)		20 m
Informations sur les alarmes/les diagnostics/l'état		
Affichage de l'état		Oui
Alarmes		
Alarme de diagnostic		Oui ; le diagnostic de port n'est disponible que dans le mode IO-Link
Alarmes de diagnostic		
Diagnostic		
Surveillance de la tension d'alimentation		Oui
	Court-circuit	Oui
LED de signalisation de diagnostic		
Surveillance de la tension d'alimentation		Oui ; DEL DIAG rouge clignotante
Affichage d'état de canal		Oui ; une DEL verte par canal pour l'état de canal Qn (état SIO) et l'état de PORT Cn (état IO-Link)
Pour le diagnostic de canal		Oui ; DEL Fn rouge
Pour le diagnostic de modules		Oui ; DEL DIAG verte/rouge
Séparation galvanique		
Canaux de séparation galvanique		
Entre les canaux		Non
	Entre les canaux et le bus du fond de panier	Oui
Isolation		
Isolation testée avec		707 V CC (essai de type)
Conditions ambiantes		
Température de fonctionnement		
Min.		- 20 °C
Débit		60 °C
Montage horizontal, min.		-20 °C
Montage horizontal, max.		60 °C
Montage vertical, min.		-20 °C
Montage vertical, max.		50 °C

Présentation générale du temps de réponse

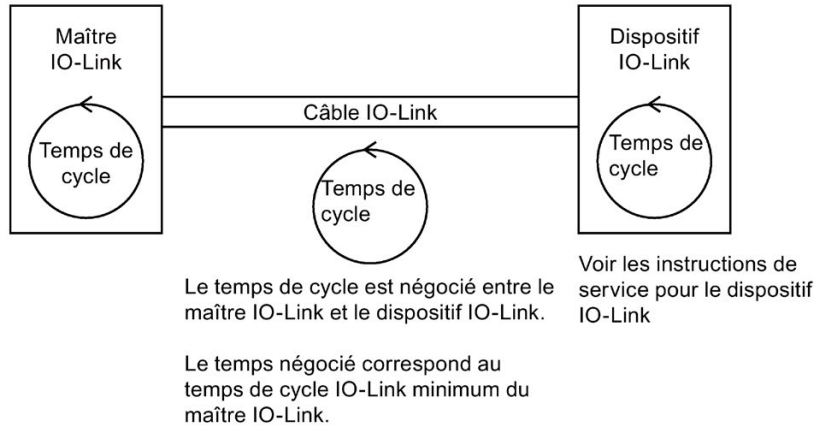


Tableau A- 183 Schéma de câblage du module SM 1278 IO-Link maître

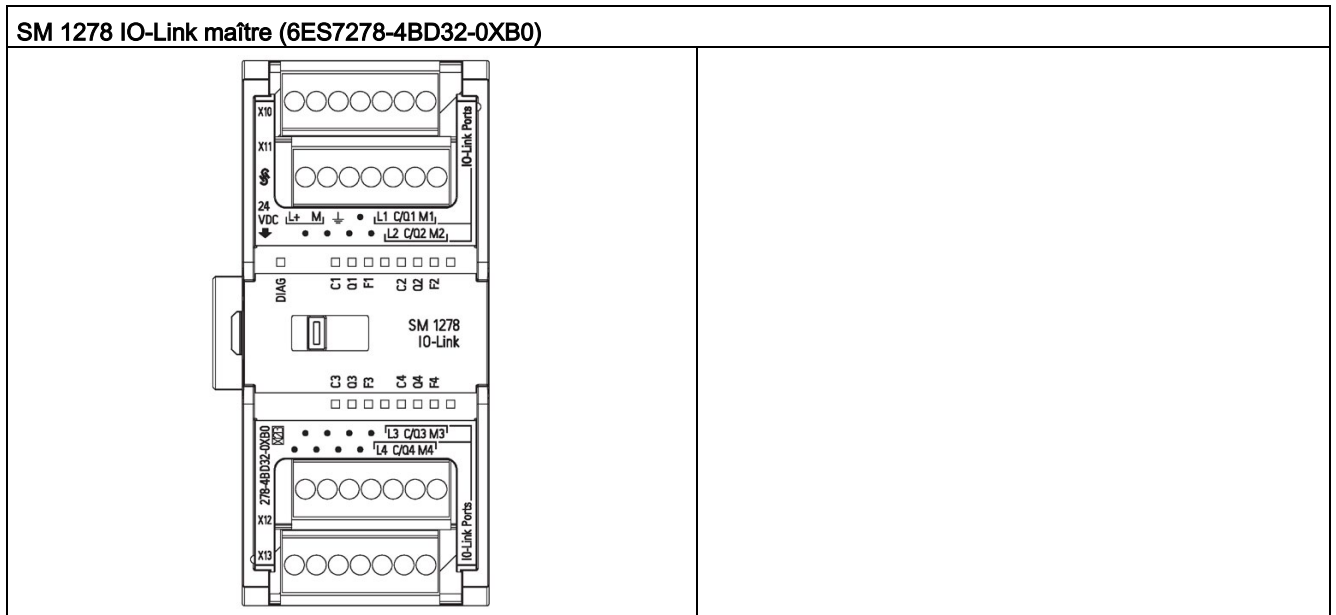


Tableau A- 184 Brochage pour le SM 1278 IO-Link maître (6ES7278-4BD32-0XB0)

Broche	X10	X11	X12	X13
1	L+/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
2	M/24 V CC	Pas de connexion	Pas de connexion	Pas de connexion
3	Terre fonctionnelle	Pas de connexion	Pas de connexion	Pas de connexion
4	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
5	L ₁	L ₂	L ₃	L ₄
6	C/Q ₁	C/Q ₂	C/Q ₃	C/Q ₄
7	M ₁	M ₂	M ₃	M ₄

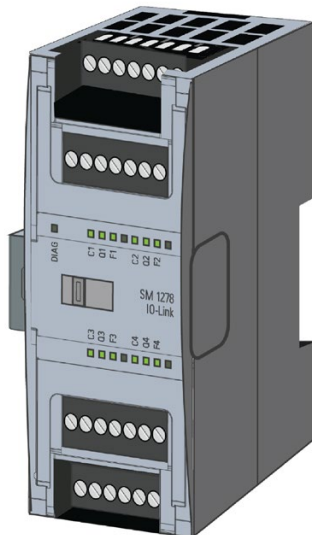
A.12.1.1 Présentation du SM 1278 4 x IO-Link maître

Le maître SM 1278 4xIO-Link est un module à 4 ports qui fait office à la fois de module d'entrées-sorties et de communication. Chaque port peut fonctionner en mode IO-Link, entrée TOR 24 V DC ou sortie TOR 24 V DC uniques.

Le maître IO-Link programme une communication acyclique avec un appareil IO-Link à l'aide du bloc fonctionnel (FB) IO_LINK_DEVICE dans le programme STEP 7 de votre automate S7-1200. Le FB IO_LINK_DEVICE indique le maître IO-Link utilisé par votre programme, ainsi que les ports utilisés par le maître pour l'échange de données.

Pour plus de détails sur l'utilisation de la bibliothèque IO-Link , voir le site Industry Online Support de Siemens (<https://support.industry.siemens.com/cs/ww/fr>). Tapez "IO-Link" dans la zone de recherche pour consulter les informations relatives aux produits IO-Link et à leur emploi.

Vue du module



Propriétés

Caractéristiques techniques

- IO-Link maître selon la spécification IO-Link V1.1 (consultez le site Web du consortium IO-Link (<http://io-link.com/en/index.php>) pour plus de détails)
- Module de communication série à quatre ports (voies)
- Vitesse de transmission des données COM1 (4.8 kbaud), COM2 (38.4 kbaud), COM3 (230.4 kbaud)
- Mode SIO (IO standard)
- Connexion de quatre périphériques IO-Link au maximum (montage 3 fils) ou de quatre actionneurs ou codeurs standard
- Diagnostic programmable par port

Fonctions prises en charge

- Données d'identification I&M (installation and maintenance)
- Mise à jour du firmware
- Paramétrage IO-Link au moyen de l'outil de configuration de port S7-PCT, STEP 7 Professional et une CPU. S7-1200 de version V4.0 ou plus. Dans STEP 7 Professional V15 (avec HSP V2.1), le paramétrage IO-Link peut être effectué au moyen de TIA Portal avec fonctionnalité restreinte.
- Bits PQI (Port Qualifier Information)
- Sauvegarde et restauration à l'aide des FB de la bibliothèque IO-Link

IO-Link est une liaison point à point entre un maître et un périphérique. Il est possible d'utiliser des capteurs/actionneurs aussi bien classiques qu'intelligents en tant que périphériques sur l'IO-Link par le biais de câbles standard non blindés au moyen de la technologie éprouvée de montage 3 fils. IO-Link est compatible avec les capteurs et actionneurs TOR classiques. Les voies de données et d'état du circuit sont réalisées en technologie 24 V DC reconnue.

Pour plus d'informations sur la technologie SIMATIC-IO-Link, voir la "description fonctionnelle du système IO-Link" sur le site Industry Online Support de Siemens (<http://support.automation.siemens.com>).

Remarque

Données de paramétrage IO-Link

Lorsque vous remplacez le module SM 4xIO-Link maître, les données de paramétrage ne lui sont pas automatiquement affectées.



Débrochage et insertion

L'insertion du module SM 4xIO-Link maître avec la charge connectée peut entraîner des situations dangereuses dans votre usine.

Il peut en résulter des dégâts physiques sur le système d'automatisation S7-1200.

Retirez ou insérez le module SM 4xIO-Link maître uniquement lorsque la charge est déconnectée.

Effets de la restauration des paramètres d'usine

Utilisez la fonction "Restaurer les paramètres d'usine" pour remettre le paramétrage que vous avez effectué avec S7-PCT sur l'état de livraison.

Après une "Restauration des réglages d'usine", les paramètres du module SM 1278 4xIO-Link sont affectés de la manière suivante :

- Les ports sont en mode DI
- Les ports sont mappés sur les adresses relatives 0.0 à 0.3
- Le PortQualifier est désactivé
- Les données de maintenance 1 à 3 sont supprimées

Remarque

Lorsque vous restaurez les paramètres d'usine, les paramètres de l'appareil sont effacés et l'état de livraison est rétabli.

Si vous retirez un module d'entrées-sorties SM 1278 4xIO-Link, restaurez ses paramètres d'usine avant de le ranger.


Marche à suivre

Pour "Restaurer les paramètres d'usine", procédez comme décrit dans l'aide en ligne de S7-PCT sous "Configuration des maîtres > dans l'onglet 'Commandes'".

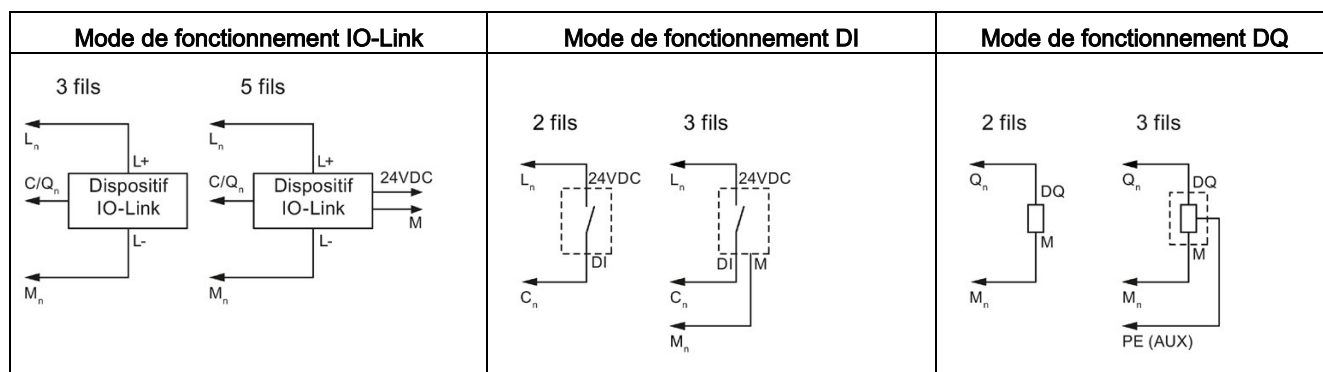
A.12.1.2 Raccordement

Pour plus de détails sur le brochage, voir le tableau Brochage pour le Maître SM 1278 I/O-Link (6ES 278-4BD32-0XB0). (Page 1573)

Le tableau suivant montre l'affectation des bornes pour le SM 1278 4 x IO-Link maître :

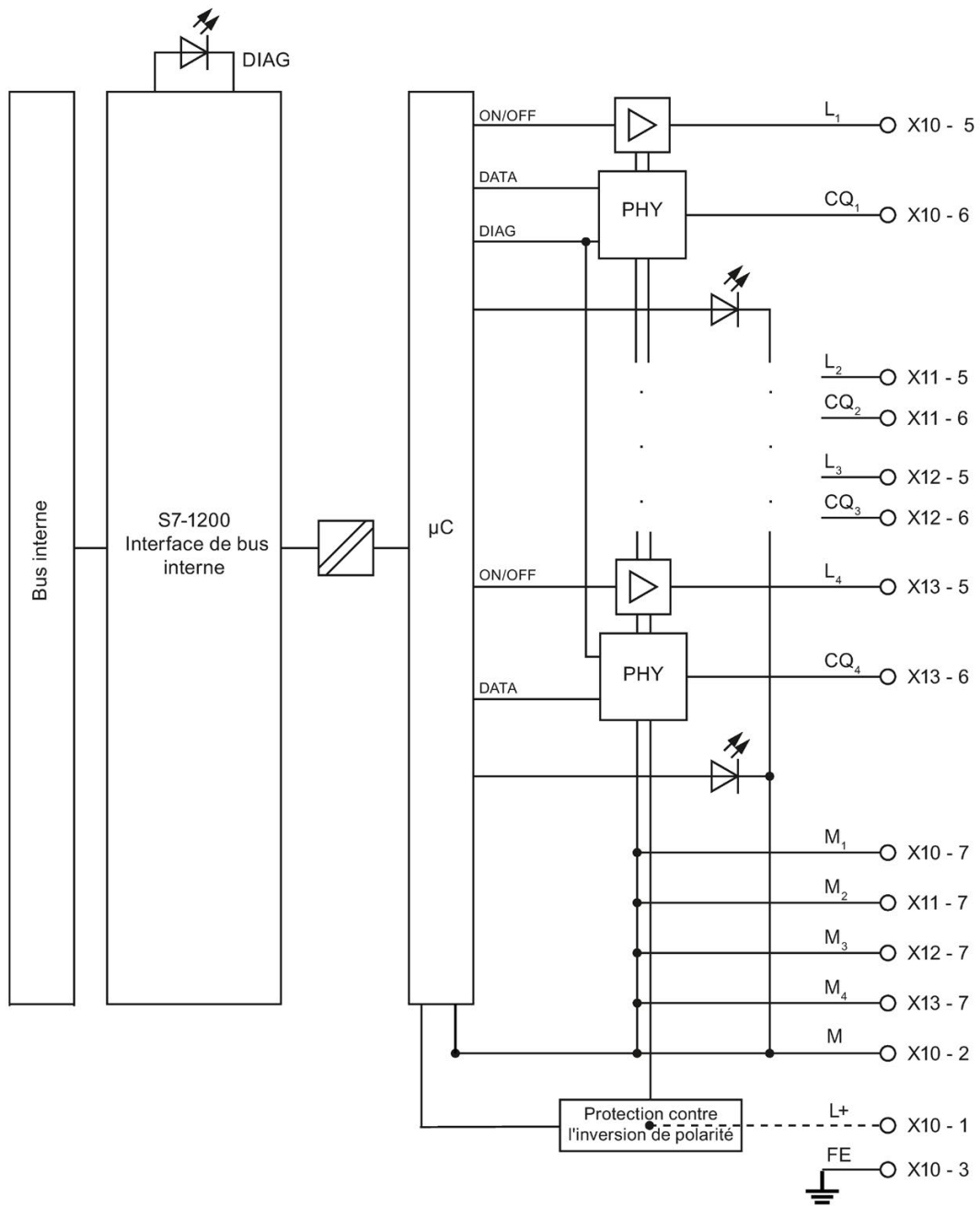
Broche	X10	X11	X12	X13	Remarques	BaseUnits
7	M ₁	M ₂	M ₃	M ₄	<ul style="list-style-type: none"> M_n : terre à esclave C/Q₁ : SDLC, DI ou DQ L_n : 24 V CC à esclave M : terre L+ : 24 V CC à maître RES : réservé ; ne peut pas être affecté 	A1
6	C/Q ₁	C/Q ₂	C/Q ₃	C/Q ₄		
5	L ₁	L ₂	L ₃	L ₄		
4	RES	RES	RES	RES		
3	 (terre fonctionnelle)	RES	RES	RES		
2	M	RES	RES	RES		
1	L+	RES	RES	RES		

Le tableau suivant illustre des exemples de raccordement (n représentant le numéro du port) :



Remarque

Les capteurs raccordés doivent utiliser l'alimentation d'appareil fournie par la connexion L_n du module maître.



A.12.1.3 Paramètres/espace d'adressage

Configuration du module SM 1278 4xIO-Link maître

Vous avez besoin de STEP 7 V13 ou de version ultérieure pour l'intégration, le paramétrage et la mise en service du module. L'outil de configuration de port S7-PCT est également nécessaire pour certaines fonctions.

Le tableau suivant indique les situations dans lesquelles vous avez besoin de S7-PCT :

	Maître SM 1278 V2.0 4xIO-Link	Maître SM 1278 V2.1 4xIO-Link
STEP7 V13.x et STEP V14.x	S7-PCT nécessaire	S7-PCT nécessaire
STEP 7 15.x	S7-PCT nécessaire	S7-PCT pas nécessaire pour les fonctions de base S7-PCT nécessaire pour les fonctions avancées

Pour plus d'informations, voir le manuel Système IO-Link SIMATIC (<https://support.industry.siemens.com/cs/ww/fr/view/65949252>).

Le tableau suivant montre les paramètres du module SM 1278 4 x IO-Link maître :

Paramètres	Plage de valeurs	Valeur par défaut	Configuration à l'état MARCHE	Portée
Diagnostic port 1	<ul style="list-style-type: none"> • Désactiver • Activer 	Désactiver	Oui	Port (voie)
Diagnostic port 2	<ul style="list-style-type: none"> • Désactiver • Activer 	Désactiver	Oui	Port (voie)
Diagnostic port 3	<ul style="list-style-type: none"> • Désactiver • Activer 	Désactiver	Oui	Port (voie)
Diagnostic port 4	<ul style="list-style-type: none"> • Désactiver • Activer 	Désactiver	Oui	Port (voie)

Paramètre d'activation du diagnostic pour les ports 1 à 4

Ce paramètre permet d'activer le diagnostic pour des ports spécifiques parmi les quatre ports IO-Link.

Les ports sont affectés comme suit :

Port 1 → voie 1

Port 2 → voie 2

Port 3 → voie 3

Port 4 → voie 4

La taille maximale des adresses d'entrée et de sortie du module SM 4xIO-Link Master est de 32 octets dans chaque cas. Vous affectez les espaces d'adresses à l'aide de l'outil de configuration de port S7-PCT ou, à partir de V15 HSP V2.1, de la configuration matérielle de TIA Portal

Enregistrement de paramètres

Affectation des paramètres dans le programme utilisateur

Vous pouvez configurer l'appareil lors de l'exécution.

Modification des paramètres lors de l'exécution

Les paramètres du module se trouvent dans l'enregistrement 128. Vous pouvez transmettre les paramètres modifiables au module à l'aide de l'instruction WRREC.

Lorsque vous réinitialisez la CPU (par mise hors tension puis sous tension), la CPU écrase les paramètres qui avaient été envoyés au module par l'instruction WRREC lors du processus de paramétrage.

Instruction pour le paramétrage

L'instruction suivante permet d'affecter des paramètres au module d'E/S dans le programme utilisateur :

Instruction	Application
SFB 53 WRREC	Transfert des paramètres modifiables au module

Message d'erreur

Les codes d'erreur suivants sont renvoyés en cas d'erreur :

Code d'erreur	Signification
80B1 _H	Erreur dans la longueur des données
80E0 _H	Erreur dans les informations d'en-tête
80E1 _H	Erreur de paramètre

Structure de l'enregistrement

Le tableau suivant montre les paramètres IO-Link :

Déca-lage	Label	Type	Valeur par défaut	Description
0	Version	1 octet	0x02	Montre la structure de l'enregistrement 0x02 pour le maître IO-Link selon IO-Link V1.1
1	Parameter length	1 octet	0x02	Longueur des paramètres (2 octets + 2 en-têtes)
Paramètres initiaux IO-Link				
2	Port diagnostics (Port 1 à n)	1 octet	0x00	Activation du diagnostic pour le port 1 à n
3	IOL properties	1 octet	0x00	Propriétés du module

Le tableau suivant montre l'enregistrement "Version" :

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Réservés		Version majeure (00)		Version mineure (0010)			

Le tableau suivant montre l'enregistrement "Port diagnostics" :

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Réservés				EN_Port4	EN_Port3	EN_Port2	EN_Port1

EN_Portx :

0 = Diagnostic désactivé

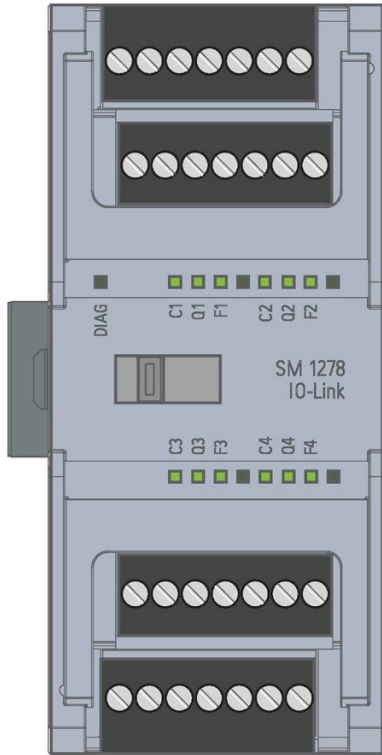
1 = Diagnostic activé

Le tableau suivant montre l'enregistrement "IOL properties" :

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Réservés							

A.12.1.4 Alarmes, erreurs et alarmes système

Signalisation par DEL



Signification des signalisations par DEL




Le tableau suivant donne la signification des signalisations d'état et d'erreur. Des mesures permettant de remédier aux alarmes de diagnostic sont présentées au paragraphe "Alarmes de diagnostic".

DEL DIAG

DIAG	Signification
<p>□ Eteinte</p>	L'alimentation du bus interne du S7-1200 est défectueuse.
<p>⚡ Clignote</p>	Le module n'est pas configuré.
<p>■ Allumée</p>	Module paramétré et pas de diagnostic du module
<p>⚡ Clignote</p>	Module paramétré et diagnostic du module OR Alimentation L+ non raccordée



DEL d'état des ports

S'applique au port IO-Link qui est en mode IO-Link.



COM/1 ... COM/4	Signification
 Eteinte	Port désactivé
 Clignote	Port activé, appareil non connecté OU Port non connecté à l'appareil configuré
 Allumée	Port activé, appareil connecté

DEL d'état des voies

S'applique au port IO-Link qui est en mode DI/Q.

DI/Q1 ... DI/Q4	Signification
 Eteinte	Signal du processus = 0
 Allumée	Signal du processus = 1

DEL d'erreur des ports

F1 ... F4	Signification
 Eteinte	Pas d'erreur
 Allumée	Erreur

Les erreurs de module ne sont signalées en tant que diagnostics (état du module) qu'en mode IO-Link.

Alarme diagnostic	Code d'erreur (décimal)	STATUS (W#16#...)	Signification (code d'erreur IO-Link)	Maître IO-Link	Appareil IO-Link
Court-circuit	1	1804	Court-circuit des câbles du processus sur l'appareil IO-Link	X	
		7710	Court-circuit sur l'appareil IO		X
Sous-tension	2	5111 5112	Tension d'alimentation trop faible		X
Surtension	3	5110	Tension d'alimentation trop élevée		X
Surchauffe	5	1805	Dépassement de température sur le maître	X	
		4000 4210	Dépassement de température sur l'appareil		X
Rupture de fil	6	1800	<ul style="list-style-type: none"> Aucun appareil IO-Link connecté Il y a une rupture sur la ligne de signaux vers l'appareil IO-Link IO-Link ne peut pas communiquer en raison d'une autre erreur 	X	
Débordement haut	7	8C10 8C20	Plage de la variable de processus dépassée		X
		8C20	Etendue de mesure dépassée		
Débordement bas	8	8C30	Plage de la variable de processus trop basse		X
Erreur	9	---	Tous les codes d'erreur IO-Link qui ne figurent pas dans cette liste sont mappés avec cette erreur PROFIBUS DP		X
Erreur de paramétrage	16	1882 1883	Impossible de configurer le maître IO-Link	X	
		1802	Appareil incorrect		
		1886	Erreur de stockage		
		6320 6321 6350	L'appareil n'a pas été configuré correctement		X
Tension d'alimentation manquante	17	1806	Tension d'alimentation L+ de l'appareil manquante	X	
		1807	Tension d'alimentation L+ de l'appareil trop faible (< 20 V)		
Fusible défectueux	18	5101	Le fusible de l'appareil est défectueux		X

Alarme diagnostic	Code d'erreur (décimal)	STATUS (W#16#...)	Signification (code d'erreur IO-Link)	Maître IO-Link	Appareil IO-Link
Coupure de sécurité	25	1880	Erreur fatale (remplacer le maître)	X	
Défaut externe	26	1809 180A 180B 180C 180D	Erreur dans le stockage de données	X	
		1808			

A.13 Signal Boards (SB) TOR

A.13.1 Caractéristiques des SB 1221 entrées TOR, 200 kHz

Tableau A- 185 Caractéristiques générales

Caractéristiques techniques	SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Référence	6ES7221-3BD30-0XB0	6ES7221-3AD30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21	
Poids	35 grammes	
Dissipation de courant	1,5 W	1,0 W
Consommation de courant (bus SM)	40 mA	
Consommation de courant (24 V CC)	7 mA / entrée + 20 mA	15 mA / entrée + 15 mA

Tableau A- 186 Entrées TOR

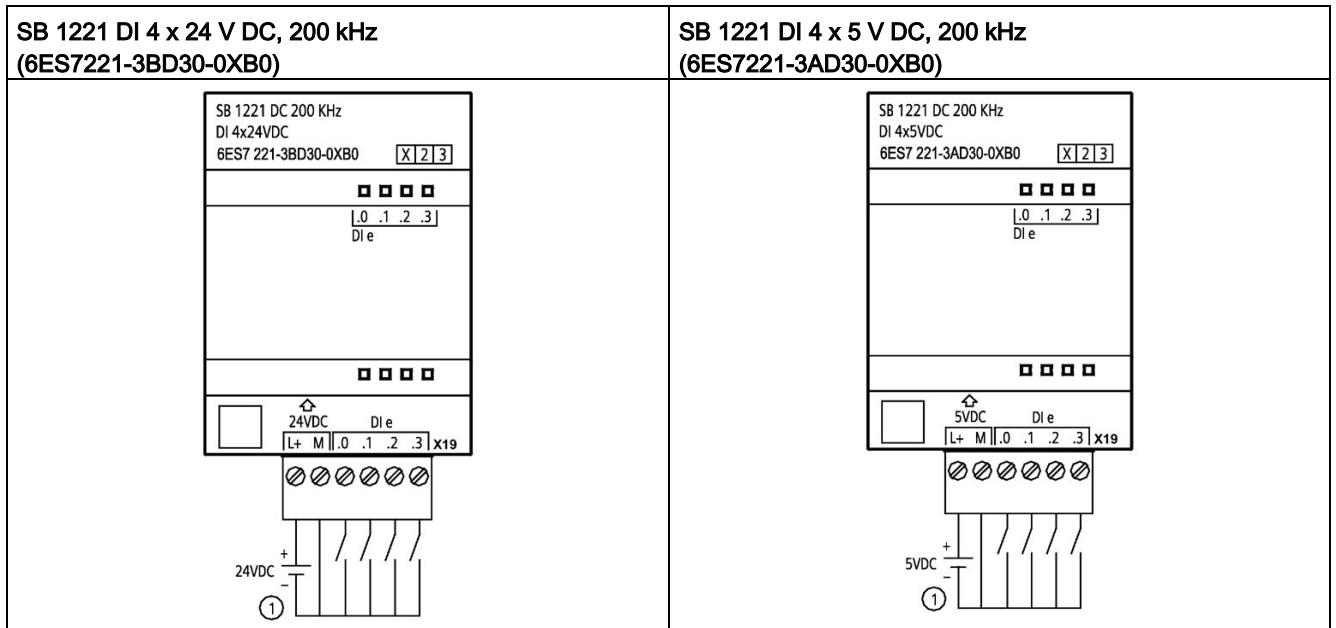
Caractéristiques techniques		SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Nombre d'entrées		4	
Type		N	
Tension nominale		24 V CC à 7 mA, nominal	5 V CC à 15 mA, nominal
Tension continue admise		28,8 V CC	6 V CC
Tension de choc		35 V CC pour 0,5 s	6 V
Signal 1 logique		0 V (10 mA) à L+ moins 10 V (2,9 mA)	0 V (20 mA) à L+ moins 2,0 V (5,1 mA)
Signal 0 logique		L+ moins 5 V (1,4 mA) à L+ (0 mA)	L+ moins 1,0 V (2,2 mA) à L+ (0 mA)
Fréquences d'entrée d'horloge HSC (max.)		Monophasé : 200 kHz Quadrature de phase : 160 kHz	
Isolation (site à logique)		707 V CC (essai de type)	
Groupes d'isolation		1	
Temps de filtre	microsecondes	0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0	
	millisecondes	0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0	
Nombre d'entrées simultanément à 1		<ul style="list-style-type: none"> • 2 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical • 4 à 55 °C horizontal ou 45 °C vertical 	4
Longueur de câble (mètres)		50, paire torsadée blindée	

Remarque

En cas de commutation de fréquences au-delà de 20 kHz, il est important que les entrées TOR reçoivent une onde carrée. Tenez compte des possibilités suivantes pour améliorer la qualité du signal arrivant aux entrées :

- Diminuez la longueur du câble.
- Remplacez un étage P seul par un étage P et N.
- Choisissez un câble de qualité supérieure.
- Réduisez le circuit/les composants de 24 V à 5 V.
- Ajoutez une charge externe à l'entrée.

Tableau A- 187 Schémas de câblage des SB entrées TOR, 200 kHz



① Prise en charge d'entrées de type N uniquement

Tableau A- 188 Brochage pour le SB 1221 DI 4 x 24 V DC, 200 kHz (6ES7221-3BD30-0XB0)

Broche	X19
1	L+/24 V CC
2	M/24 V CC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

Tableau A- 189 Brochage pour le SB 1221 DI 4 x 5 V DC, 200 kHz (6ES7221-3AD30-0XB0)

Broche	X19
1	L+/5 V CC
2	M/5 V CC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

A.13.2 Caractéristiques des SB 1222 sorties TOR, 200 kHz

Tableau A- 190 Caractéristiques générales

Caractéristiques techniques	SB 1222 DQ 4 x 24 V DC, 200 kHz	SB 1222 DQ 4 x 5 V DC, 200 kHz
Référence	6ES7222-1BD30-0XB0	6ES7222-1AD30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21	
Poids	35 grammes	
Dissipation de courant	0,5 W	
Consommation de courant (bus SM)	35 mA	
Consommation de courant (24 V CC)	15 mA	

Tableau A- 191 Sorties TOR

Caractéristiques techniques	SB 1222 DQ 4 x 24 V DC, 200 kHz	SB 1222 DQ 4 x 5 V DC, 200 kHz
Nombre de sorties	4	
Type des sorties	Transistor à effet de champ MOS (P et N) ¹	
Plage de tension	20,4 à 28,8 V CC	4,25 à 6,0 V CC
Signal 1 logique à courant max.	L+ moins 1,5 V	L+ moins 0,7 V
Signal 0 logique à courant max.	1,0 V CC max.	0,2 V CC max.
Courant (max.)	0,1 A	
Charge de lampe	--	
Résistance état activé (contact)	11 Ω max.	7 Ω max.
Résistance état désactivé	6 Ω max.	0,2 Ω max.
Courant de fuite par sortie	--	
Fréquence de sortie de trains d'impulsions	200 kHz max., 2 Hz min.	
Courant de choc	0,11 A	
Protection contre la surcharge	Non	
Isolation (site à logique)	707 V CC (essai de type)	
Groupes d'isolation	1	
Courants par commun	0,4 A	
Tension de blocage inductive	Non	
Retard de commutation	1,5 μs + 300 ns à la montée 1,5 μs + 300 ns à la retombée	200 ns + 300 ns à la montée 200 ns + 300 ns à la retombée
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> • 2 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical • 4 à 55 °C horizontal ou 45 °C vertical 	4
Longueur de câble (mètres)	50, paire torsadée blindée	

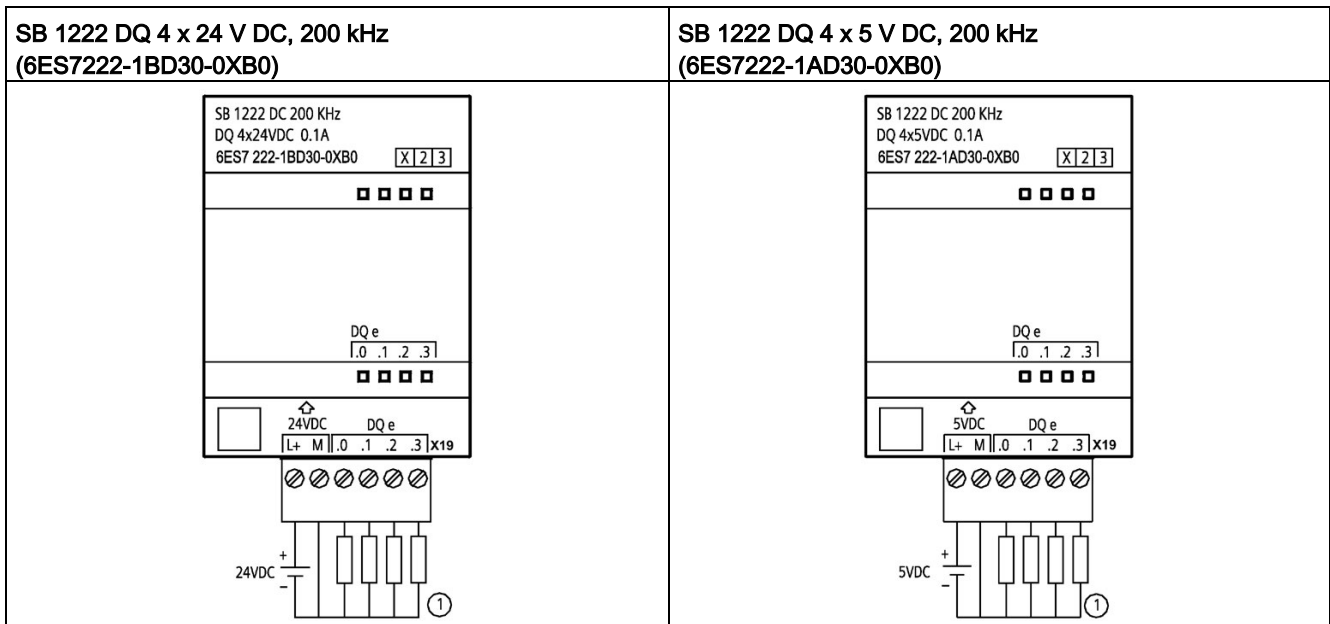
¹ Comme les deux configurations "sink" et "source" (P et N) sont prises en charge par la même circuiterie, l'état actif d'une charge "source" est l'opposé de celui d'une charge "sink". Une sortie "source" fournit une logique positive (bit Q et DEL activés lorsque la charge présente un flux de courant) alors qu'une sortie "sink" fournit une logique négative (bit Q et DEL désactivés lorsque la charge présente un flux de courant). Si le module est enfiché sans programme utilisateur, le paramétrage par défaut pour ce module est 0 V, ce qui signifie qu'une charge "sink" sera activée.

Remarque

En cas de commutation de fréquences au-delà de 20 kHz, il est important que les entrées TOR reçoivent une onde carrée. Tenez compte des possibilités suivantes pour améliorer la qualité du signal arrivant aux entrées :

- Diminuez la longueur du câble.
- Remplacez un étage P seul par un étage P et N
- Choisissez un câble de qualité supérieure
- Réduisez le circuit/les composants de 24 V à 5 V.
- Ajoutez une charge externe à l'entrée

Tableau A- 192 Schémas de câblage des SB sorties TOR, 200 kHz



① Pour les sorties de type N, connectez "Charge" à "-" (comme illustré). Pour les sorties de type P, connectez "Charge" à "+". Comme les deux configurations "sink" et "source" (P et N) sont prises en charge par la même circuiterie, l'état actif d'une charge "source" est l'opposé de celui d'une charge "sink". Une sortie "source" fournit une logique positive (bit Q et DEL activés lorsque la charge présente un flux de courant) alors qu'une sortie "sink" fournit une logique négative (bit Q et DEL désactivés lorsque la charge présente un flux de courant). Si le module est enfiché sans programme utilisateur, le paramétrage par défaut pour ce module est 0 V, ce qui signifie qu'une charge "sink" sera activée.

Remarque

Veillez à ce que le fil de liaison M soit mis à la terre de manière sécurisée. La coupure de la connexion du fil de terre aux SB DQ rapides peut permettre un courant de fuite suffisant pour activer une charge CC. Si les sorties sont utilisées pour des applications de charge CC critiques, il faut prendre des précautions supplémentaires en utilisant un fil de terre redondant connecté au SB.

Tableau A- 193 Brochage pour le SB 1222 DQ 4 x 24 V DC, 200 kHz (6ES7222-1BD30-0XB0)

Broche	X19
1	L+/24 V CC
2	M/24 V CC
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

Tableau A- 194 Brochage pour le SB 1222 DQ 4 x 5 V DC, 200 kHz (6ES7222-1AD30-0XB0)

Broche	X19
1	L+/5 V CC
2	M/5 V CC
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

A.13.3 Caractéristiques des SB 1223 entrées/sorties TOR, 200 kHz

Tableau A- 195 Caractéristiques générales

Caractéristiques techniques	SB 1223 DI 2 x 24 V DC/ DQ 2 x 24 V DC, 200 kHz	SB 1223 DI 2 x 5 V DC/ DQ 2 x 5 V DC, 200 kHz
Référence	6ES7223-3BD30-0XB0	6ES7223-3AD30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21	
Poids	35 grammes	
Dissipation de courant	1,0 W	0,5 W
Consommation de courant (bus SM)	35 mA	
Consommation de courant (24 V CC)	7 mA / entrée + 30 mA	15 mA / entrée + 15 mA

Tableau A- 196 Entrées TOR

Caractéristiques techniques		SB 1223 DI 2 x 24 V DC/ DQ 2 x 24 V DC, 200 kHz	SB 1223 DI 2 x 5 V DC/ DQ 2 x 5 V DC, 200 kHz
Nombre d'entrées		2	
Type		N	
Tension nominale		24 V CC à 7 mA, nominal	5 V CC à 15 mA, nominal
Tension continue admise		28,8 V CC	6 V CC
Tension de choc		35 V CC pour 0,5 s	6 V
Signal 1 logique		0 V (10 mA) à L+ moins 10 V (2,9 mA)	0 V (20 mA) à L+ moins 2,0 V (5,1 mA)
Signal 0 logique		L+ moins 5 V (1,4 mA) à L+ (0 mA)	L+ moins 1,0 V (2,2 mA) à L+ (0 mA)
Fréquences d'entrée d'horloge HSC (max.)		Monophasé : 200 kHz Quadrature de phase : 160 kHz	
Isolation (site à logique)		707 V CC (essai de type)	
Groupes d'isolation		1 (sans isolation vers les sorties)	
Temps de filtre	microsecondes	0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0	
	millisecondes	0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0	
Nombre d'entrées simultanément à 1		2	
Longueur de câble (mètres)		50, paire torsadée blindée	

Tableau A- 197 Sorties TOR

Caractéristiques techniques	SB 1223 DI 2 x 24 V DC/ DQ 2 x 24 V DC, 200 kHz	SB 1223 DI 2 x 5 V DC/ DQ 2 x 5 V DC, 200 kHz
Nombre de sorties	2	
Type des sorties	Transistor à effet de champ MOS (type "sink" et "source") ¹	
Plage de tension	20,4 à 28,8 V CC	4,25 à 6,0 V CC
Valeur nominale	24 V CC	5 V CC
Signal 1 logique à courant max.	L+ moins 1,5 V	L+ moins 0,7 V
Signal 0 logique à courant max.	1,0 V CC max.	0,2 V CC max.
Courant (max.)	0,1 A	
Charge de lampe	--	
Résistance état activé (contact)	11 Ω max.	7 Ω max.
Résistance état désactivé	6 Ω max.	0,2 Ω max.
Courant de fuite par sortie	--	
Fréquence de sortie de trains d'impulsions	200 kHz max., 2 Hz min.	
Courant de choc	0,11 A	
Protection contre la surcharge	Non	
Isolation (site à logique)	707 V CC (essai de type)	
Groupes d'isolation	1 (sans isolation vers les entrées)	
Courants par commun	0,2 A	
Tension de blocage inductive	Non	
Retard de commutation	1,5 μs + 300 ns à la montée 1,5 μs + 300 ns à la retombée	200 μs + 300 ns à la montée 200 μs + 300 ns à la retombée
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Nombre de sorties simultanément à 1	2	
Longueur de câble (mètres)	50, paire torsadée blindée	

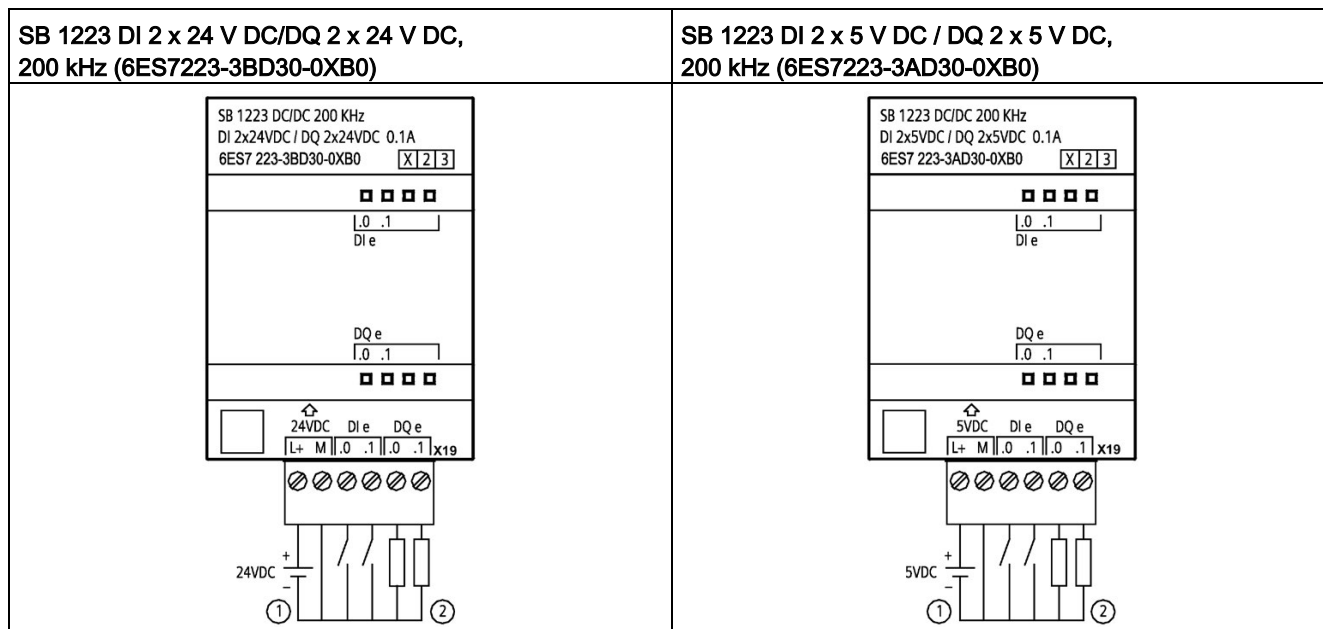
¹ Comme les deux configurations "sink" et "source" (P et N) sont prises en charge par la même circuiterie, l'état actif d'une charge "source" est l'opposé de celui d'une charge "sink". Une sortie "source" fournit une logique positive (bit Q et DEL activés lorsque la charge présente un flux de courant) alors qu'une sortie "sink" fournit une logique négative (bit Q et DEL désactivés lorsque la charge présente un flux de courant). Si le module est enfiché sans programme utilisateur, le paramétrage par défaut pour ce module est 0 V, ce qui signifie qu'une charge "sink" sera activée.

Remarque

En cas de commutation de fréquences au-delà de 20 kHz, il est important que les entrées TOR reçoivent une onde carrée. Tenez compte des possibilités suivantes pour améliorer la qualité du signal arrivant aux entrées :

- Diminuez la longueur du câble
- Remplacez un étage P seul par un étage P et N
- Choisissez un câble de qualité supérieure
- Réduisez le circuit/les composants de 24 V à 5 V
- Ajoutez une charge externe à l'entrée

Tableau A- 198 Schémas de câblage des SB entrées / sorties TOR, 200 kHz



① Prise en charge d'entrées de type N uniquement

② Pour les sorties de type N, connectez "Charge" à "-" (comme illustré). Pour les sorties de type P, connectez "Charge" à "+". 1 Comme les deux configurations "sink" et "source" (P et N) sont prises en charge par la même circuiterie, l'état actif d'une charge "source" est l'opposé de celui d'une charge "sink". Une sortie "source" fournit une logique positive (bit Q et DEL activés lorsque la charge présente un flux de courant) alors qu'une sortie "sink" fournit une logique négative (bit Q et DEL désactivés lorsque la charge présente un flux de courant). Si le module est enfiché sans programme utilisateur, le paramétrage par défaut pour ce module est 0 V, ce qui signifie qu'une charge "sink" sera activée.

Remarque

Veillez à ce que le fil de liaison M soit mis à la terre de manière sécurisée. La coupure de la connexion du fil de terre aux SB DQ rapides peut permettre un courant de fuite suffisant pour activer une charge CC. Si les sorties sont utilisées pour des applications de charge CC critiques, il faut prendre des précautions supplémentaires en utilisant un fil de terre redondant connecté au SB.

Tableau A- 199 Brochage pour le SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz (6ES7223-3BD30-0XB0)

Broche	X19
1	L+/24 V CC
2	M/24 V CC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

Tableau A- 200 Brochage pour le SB 1223 DI 2 x 5 V DC/DQ 2 x 5 V DC, 200 kHz (6ES7223-3AD30-0XB0)

Broche	X19
1	L+/5 V CC
2	M/5 V CC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

A.13.4 Caractéristiques du SB 1223 entrées 2 X 24 V DC/sorties 2 X 24 V DC

Tableau A- 201 Caractéristiques générales

Caractéristiques techniques	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC
Référence	6ES7223-0BD30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	40 grammes
Dissipation de courant	1,0 W
Consommation de courant (bus SM)	50 mA
Consommation de courant (24 V CC)	4 mA/entrée utilisée

Tableau A- 202 Entrées TOR

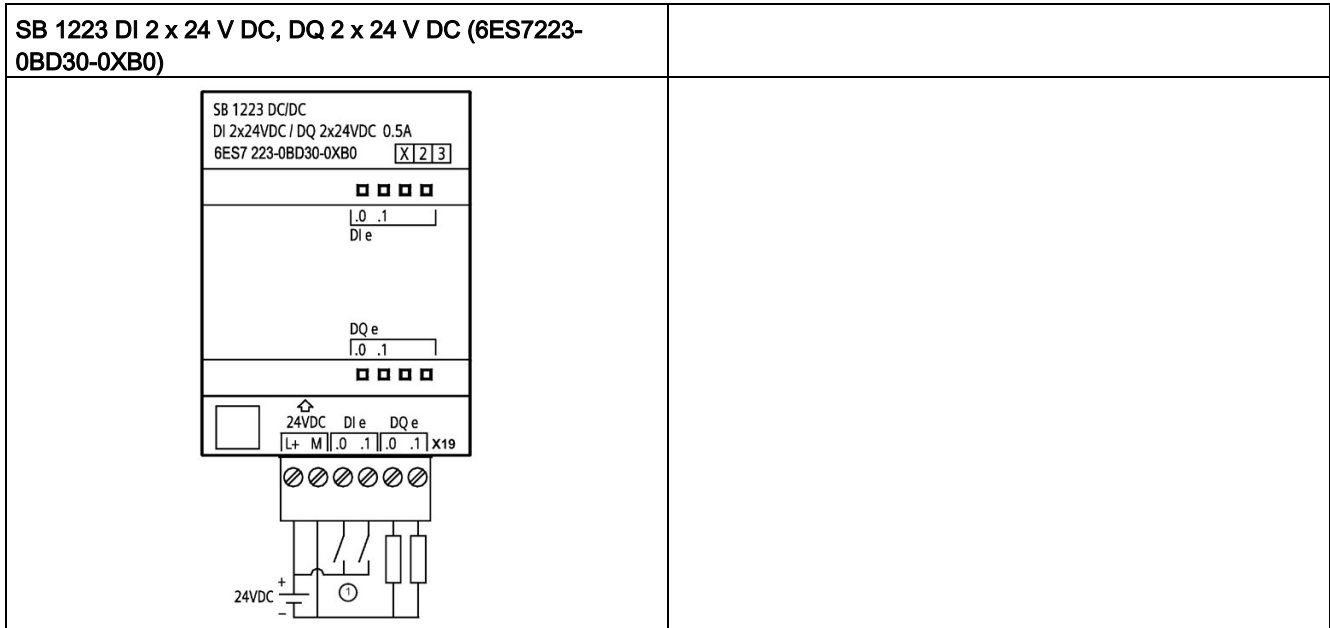
Caractéristiques techniques	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC	
Nombre d'entrées	2	
Type	CEI type 1 en mode P	
Tension nominale	24 V CC à 4 mA, nominal	
Tension continue admise	30 V CC max.	
Tension de choc	35 V CC pour 0,5 s	
Signal 1 logique (min.)	15 V CC à 2,5 mA	
Signal 0 logique (max.)	5 V CC à 1 mA	
Fréquences d'entrée d'horloge HSC (max.)	Monophasé : 30 kHz (15 à 26 V CC) Quadrature de phase : 20 kHz (15 à 26 V CC)	
Isolation (site à logique)	707 V CC (essai de type)	
Groupes d'isolation	1	
Temps de filtre	microsecondes	0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
	millisecondes	0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Nombre d'entrées simultanément à 1	2	
Longueur de câble (mètres)	500 blindé, 300 non blindé	

Tableau A- 203 Sorties TOR

Caractéristiques techniques	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC
Nombre de sorties	2
Type des sorties	Transistor à effet de champ MOS (type "source")
Plage de tension	20,4 à 28,8 V CC
Signal 1 logique à courant max.	20 V CC min.
Signal 0 logique avec charge 10 kΩ	0,1 V CC max.
Courant (max.)	0,5 A
Charge de lampe	5 W
Résistance état activé (contact)	0,6 Ω max.
Courant de fuite par sortie	10 μA max.
Fréquence de sortie de trains d'impulsions (PTO)	20 kHz max., 2 Hz min. ¹
Courant de choc	5 A pour 100 ms max.
Protection contre la surcharge	Non
Isolation (site à logique)	707 V CC (essai de type)
Groupes d'isolation	1
Courants par commun	1 A
Tension de blocage inductive	L+ moins 48 V, dissipation 1 W
Retard de commutation	2 μs max. de 0 à 1 10 μs max. de 1 à 0
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	2
Longueur de câble (mètres)	500 m blindé, 150 m non blindé

¹ Selon vos câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10% du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

Tableau A- 204 Schéma de câblage du SB entrées / sorties TOR



① Prise en charge d'entrées de type P uniquement

Tableau A- 205 Brochage pour le SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC (6ES7223-0BD30-0XB0)

Broche	X19
1	L+/24 V CC
2	M/24 V CC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

A.14 Signal Boards (SB) analogiques

A.14.1 Caractéristiques du SB 1231 1 sortie analogique

Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ce SB.

Tableau A- 206 Caractéristiques générales

Caractéristiques techniques	SB 1231 AI 1 x 12 bits
Référence	6ES7231-4HA30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	35 grammes
Dissipation de courant	0,4 W
Consommation de courant (bus SM)	55 mA
Consommation de courant (24 V CC)	aucune

Tableau A- 207 Entrées analogiques

Caractéristiques techniques	SB 1231 AI 1x12 bits
Nombre d'entrées	1
Type	Tension ou courant (différentiel)
Plage	± 10 V, ± 5 V, $\pm 2,5$ V ou 0 à 20 mA
Résolution	11 bits + bit de signe
Plage pleine échelle (mot de données)	- 27648 à 27648
Dépassement haut/bas (mot de données)	Tension : 32511 à 27649 / - 27649 à - 32512 Courant : 32511 à 27649 / 0 à - 4864 (voir Représentation des entrées analogiques pour la tension et Représentation des entrées analogiques pour le courant (Page 1607))
Débordement haut / bas (mot de données)	Tension : 32767 à 32512 / - 32513 à - 32768 Courant : 32767 à 32512 / - 4865 à - 32768 (voir Représentation des entrées analogiques pour la tension et Représentation des entrées analogiques pour le courant (Page 1607))
Tension/courant de tenue maximum	± 35 V / ± 40 mA
Lissage	Aucun, faible, moyen ou fort (voir Temps de réponse des entrées analogiques pour le temps de réponse indicelle (Page 1606))
Réjection des bruits	400, 60, 50 ou 10 Hz (voir Temps de réponse des entrées analogiques pour les taux d'échantillonnage (Page 1606))
Précision (25 °C / - 20 à 60 °C)	$\pm 0,3\%$ / $\pm 0,6\%$ de la pleine échelle

Caractéristiques techniques	SB 1231 AI 1x12 bits
Impédance d'entrée	Tension : 150 kΩ ; courant : 250 Ω
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Principe de mesure	Conversion de valeur effective
Réjection en mode commun	40 dB, CC pour 60 Hz
Plage de signal opérationnelle	La tension signal plus mode commun doit être inférieure à + 35 V et supérieure à - 35 V.
Isolation (site à logique)	Aucune
Longueur de câble (mètres)	100 m, torsadé et blindé

Tableau A- 208 Diagnostic

Caractéristiques techniques	SB 1231 AI 1 x 12 bits
Débordement haut/bas	Oui
24 V CC basse tension	Non

Transmetteurs SB 1231

Ils sont disponibles en transmetteurs 2 fils ou 4 fils comme décrit ci-dessous.

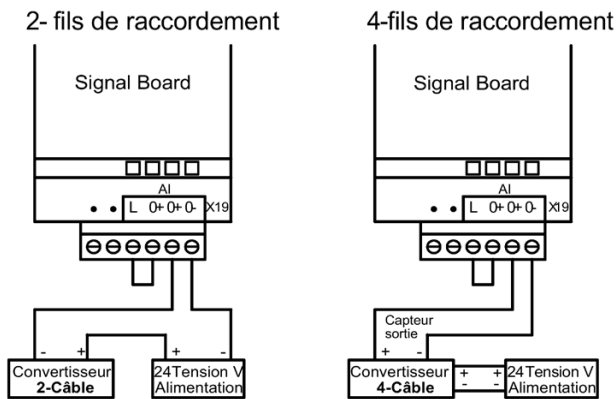


Tableau A- 209 Schéma de câblage du SB entrée analogique

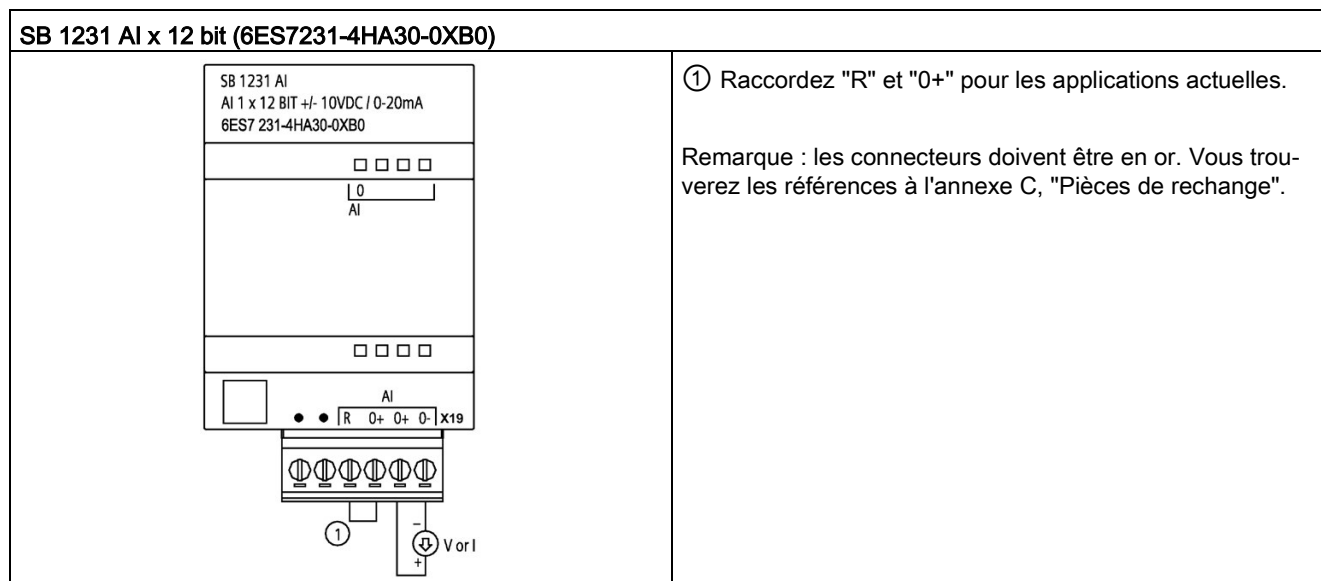


Tableau A- 210 Brochage pour le SB 1231 AI x 12 bit (6ES7231-4HA30-0XB0)

Broche	X19 (or)
1	Pas de connexion
2	Pas de connexion
3	AI R
4	AI 0+
5	AI 0+
6	AI 0-

A.14.2 Caractéristiques du SB 1232 1 sortie analogique

Tableau A- 211 Caractéristiques générales

Caractéristiques techniques	SB 1232 AQ 1 x 12 bits
Référence	6ES7232-4HA30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	40 grammes
Dissipation de courant	1,5 W
Consommation de courant (bus SM)	15 mA
Consommation de courant (24 V CC)	40 mA (sans charge)

Tableau A- 212 Sorties analogiques

Caractéristiques techniques	SB 1232 AQ 1 x 12 bits
Nombre de sorties	1
Type	Tension ou courant
Plage	±10 V ou 0 à 20 mA
Résolution	Tension : 12 bits Courant : 11 bits
Plage pleine échelle (mot de données) Voir les plages des sorties pour la tension et le courant (Page 1608).	Tension : -27 648 à 27 648 Courant : 0 à 27648
Précision (25 °C / - 20 à 60 °C)	±0,5% / ±1% de la pleine échelle
Temps d'établissement (95% de la nouvelle valeur)	Tension : 300 µs (R), 750 µs (1 µF) Courant : 600 µs (1 mH), 2 ms (10 mH)
Impédance de charge	Tension : ≥ 1 000 Ω Courant : ≤ 600 Ω
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Isolation (site à logique)	Aucune
Longueur de câble (mètres)	100 m, torsadé et blindé

Tableau A- 213 Diagnostic

Caractéristiques techniques	SB 1232 AQ 1 x 12 bits
Débordement haut/bas	Oui
Court-circuit à la terre (mode tension uniquement)	Oui
Rupture de fil (mode courant uniquement)	Oui

Tableau A- 214 Schéma de câblage du SB 1232 AQ 1 x 12 bits

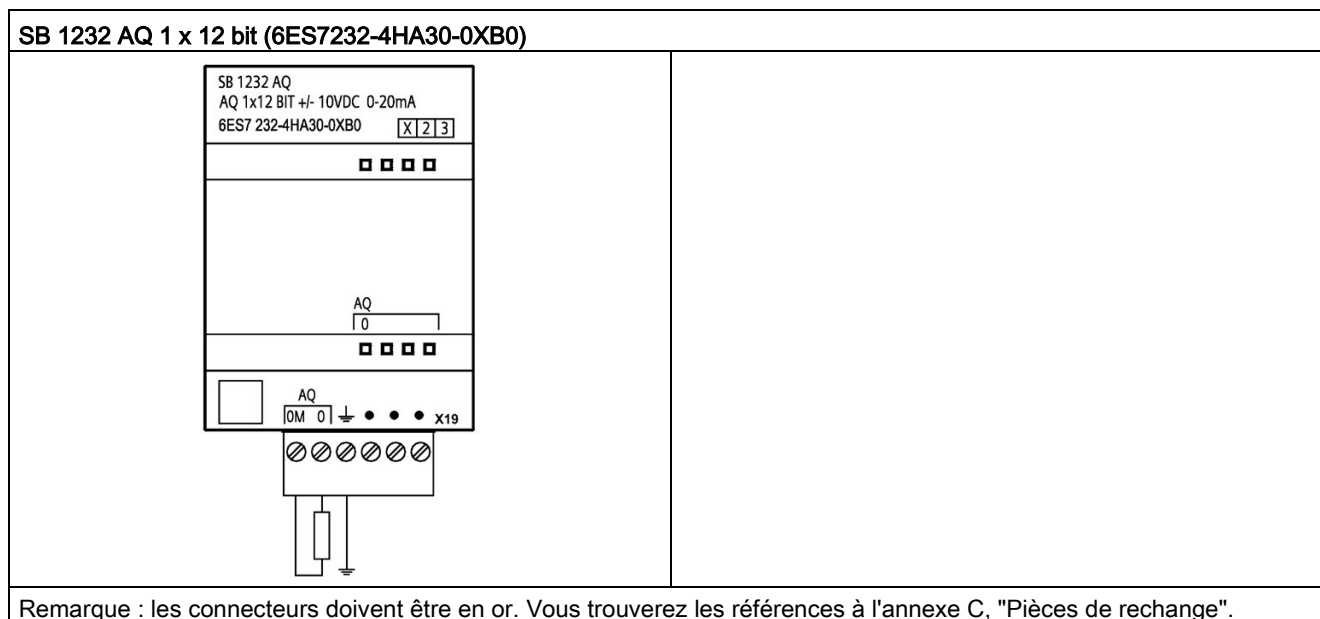


Tableau A- 215 Brochage pour le SB 1232 AQ 1 x 12 bit (6ES7232-4HA30-0XB0)

Broche	X19 (or)
1	AQ 0M
2	AQ 0
3	Terre fonctionnelle
4	Pas de connexion
5	Pas de connexion
6	Pas de connexion

A.14.3 Plages de mesure pour les entrées et sorties analogiques

A.14.3.1 Réponse indicielle des entrées analogiques

Tableau A- 216 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Sélection de temps d'intégration			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Aucun (1 cycle) : Pas de moyennage	4,5 ms	18,7 ms	22,0 ms	102 ms
Faible (4 cycles) : 4 échantillons	10,6 ms	59,3 ms	70,8 ms	346 ms
Moyen (16 cycles) : 16 échantillons	33,0 ms	208 ms	250 ms	1240 ms
Fort (32 cycles) : 32 échantillons	63,0 ms	408 ms	490 ms	2440 ms
Temps d'échantillonnage	0,156 ms	1,042 ms	1,250 ms	6,250 ms

A.14.3.2 Temps d'échantillonnage et temps d'actualisation pour les entrées analogiques

Tableau A- 217 Temps d'échantillonnage et temps d'actualisation

Sélection	Temps d'échantillonnage	Temps d'actualisation du SB
400 Hz (2,5 ms)	0,156 ms	0,156 ms
60 Hz (16,6 ms)	1,042 ms	1,042 ms
50 Hz (20 ms)	1,250 ms	1,25 ms
10 Hz (100 ms)	6,250 ms	6,25 ms

A.14.3.3 Plages de mesures d'entrées analogiques pour tension et courant (SB et SM)

Tableau A- 218 Représentation des entrées analogiques pour la tension (SB et SM)

Technologie		Plage de mesure de tension				
Décimal	Hexadécimal	± 10 V	± 5 V	± 2,5 V	± 1,25 V	
32767	7FFF ¹	11,851 V	5,926 V	2,963 V	1,481 V	Débordement haut
32512	7F00					
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Plage de dépassement haut
27649	6C01					
27648	6C00	10 V	5 V	2,5 V	1,250 V	Plage nominale
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V	
1	1	361,7 µV	180,8 µV	90,4 µV	45,2 µV	
0	0	0 V	0 V	0 V	0 V	
-1	FFFF					
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V	
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V	
-27649	93FF					Plage de dépassement bas
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	
-32513	80FF					Débordement bas
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V	

¹ 7FFF peut être renvoyé pour l'une des raisons suivantes : débordement (comme noté dans ce tableau), avant que des valeurs valides soient disponibles (par exemple immédiatement lors d'un amorçage du système), ou si une rupture de fil est détectée.

Tableau A- 219 Représentation des entrées analogiques pour le courant (SB et SM)

Technologie		Plage de mesure de courant		
Décimal	Hexadécimal	0 mA à 20 mA	4 mA à 20 mA	
32767	7FFF	> 23,52 mA	> 22,81 mA	Débordement haut
32511	7EFF	23,52 mA	22,81 mA	Plage de dépassement haut
27649	6C01			
27648	6C00	20 mA	20 mA	Plage nominale
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			
-4864	ED00	-3,52 mA	1,185 mA	Plage de dépassement bas
32767 ¹	7FFF		< 1,185 mA	
-32768	8000	< -3,52 mA		Débordement bas (0 à 20 mA)

¹ La valeur de rupture de fil 32767 (16#7FFF) est toujours renvoyée quel que soit l'état de l'alarme de rupture de fil.

A.14.3.4 Plages de mesures de sorties analogiques pour tension et courant (SB et SM)

Tableau A- 220 Représentation de sorties analogiques pour tension (SB et SM)

Système		Plage de sortie de tension	
Décimal	Hexadécimal	± 10 V	
32767	7FFF	Voir note 1	Débordement haut
32512	7F00	Voir note 1	
32511	7EFF	11,76 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
1	1	361,7 µV	
0	0	0 V	
-1	FFFF	-361,7 µV	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		
-32512	8100	-11,76 V	Plage de dépassement bas
-32513	80FF	Voir note 1	
-32768	8000	Voir note 1	
			Débordement bas

¹ À l'état débordement haut ou bas, les sorties analogiques prendront la valeur de remplacement de l'état de fonctionnement Arrêt.

Tableau A- 221 Représentation de sorties analogiques pour courant (SB et SM)

Système		Plage de sortie du courant		
Décimal	Hexadécimal	0 mA à 20 mA	4 mA à 20 mA	
32767	7FFF	Voir note 1	Voir note 1	Débordement haut
32512	7F00	Voir note 1	Voir note 1	
32511	7EFF	23,52 mA	22,81 mA	Plage de dépassement haut
27649	6C01			
27648	6C00	20 mA	20 mA	Plage nominale
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4mA	
-1	FFFF		4 mA à 578,7 nA	Plage de dépassement bas
-6912	E500		0 mA	
-6913	E4FF			Impossible. Valeur de sortie limitée à 0 mA.
-32512	8100			
-32513	80FF	Voir note 1	Voir note 1	Débordement bas
-32768	8000	Voir note 1	Voir note 1	

¹ À l'état débordement haut ou bas, les sorties analogiques prendront la valeur de remplacement de l'état de fonctionnement Arrêt.

A.14.4 Signal Boards (SB) Thermocouple

A.14.4.1 Caractéristiques du SB 1231 1 entrée analogique Thermocouple

Remarque

Le firmware de votre CPU doit être de version V2.0 ou supérieure pour que vous puissiez utiliser ce SB.

Tableau A- 222 Caractéristiques générales

Caractéristiques techniques	SB 1231 AI 1 x 16 bits Thermocouple
Référence	6ES7231-5QA30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	35 grammes
Dissipation de courant	0,5 W
Consommation de courant (bus SM)	5 mA
Consommation de courant (24 V CC)	20 mA

Tableau A- 223 Entrées analogiques

Caractéristiques techniques	SB 1231 AI 1x16 bits Thermocouple	
Nombre d'entrées	1	
Type	TC flottant et mV	
Plage <ul style="list-style-type: none"> • Plage nominale (mot de données) • Dépassement haut/bas (mot de données) • Débordement haut / bas (mot de données) 	Voir le tableau de sélection de filtre pour thermocouples (Page 1611).	
Résolution	Température	0,1° C / 0,1° F
	Tension	15 bits plus signe
Tension de tenue maximum	± 35 V	
Réjection des bruits	85 dB pour le réglage de filtre sélectionné (10 Hz, 50 Hz, 60 Hz, 400 Hz)	
Réjection en mode commun	> 120 dB à 120 V CA	
Impédance	≥ 10 M Ω	
Précision	Voir le tableau de sélection pour thermocouples (Page 1611).	
Répétabilité	± 0,05% pleine échelle	
Principe de mesure	Intégration	
Temps d'actualisation du module	Voir le tableau de sélection de filtre pour thermocouples (Page 1611).	
Erreur de soudure froide	±1,5° C	
Isolation (site à logique)	707 V CC (essai de type)	

Caractéristiques techniques	SB 1231 AI 1x16 bits Thermocouple
Longueur de câble (mètres)	max. 100 m au capteur
Résistance de fil	100 Ω max.

Tableau A- 224 Diagnostic

Caractéristiques techniques	SB 1231 AI 1 x 16 bits Thermocouple
Débordement haut/bas ¹	Oui
Rupture de fil ^{2, 3}	Oui

- ¹ Les informations d'alarmes de diagnostic de débordement haut et bas sont notifiées dans les valeurs de données analogiques même si les alarmes sont désactivées dans la configuration du module.
- ² Lorsque l'alarme de rupture de fil est désactivée et qu'un fil est ouvert dans le câblage du capteur, le module peut émettre des valeurs aléatoires.
- ³ Le module réalise un test de rupture de fil toutes les 6 secondes, qui étend le délai de mise à jour de 9 ms pour chaque voie de validation une fois toutes les 6 secondes.

Le module d'entrées analogiques SM 1231 Thermocouple (TC) mesure la valeur de tension connectée aux entrées du module.

Le Signal Board analogique SB 1231 Thermocouple mesure la valeur de tension connectée aux entrées du Signal Board. Le type de mesure de température peut être soit "Thermocouple", soit "Tension".

- "Thermocouple" : La valeur sera notifiée en degrés multipliés par dix (par exemple, 25,3 degrés seront indiqués sous la forme 253 en décimal).
- "Tension" : La valeur pleine échelle de la plage nominale sera 27648 en décimal.

A.14.4.2 Principe de fonctionnement des thermocouples

Un thermocouple est composé de deux conducteurs métalliques de nature différente, reliés électriquement par soudure l'un à l'autre. Une tension proportionnelle à la température de la soudure froide est générée. Cette tension est faible ; un microvolt peut représenter beaucoup de degrés. Mesurer la tension provenant d'un thermocouple, compenser les soudures supplémentaires, puis linéariser le résultat constituent le principe fondamental des mesures de température à l'aide de thermocouples.

Lorsque vous raccordez un thermocouple au module SM 1231 Thermocouple, les deux fils métalliques de nature différente sont reliés au module, au niveau du connecteur de signaux du module. L'endroit où les deux fils sont reliés l'un à l'autre constitue le point de mesure du thermocouple.

Deux autres thermocouples sont constitués à l'endroit où les deux fils de nature différente sont raccordés au connecteur de signaux. La température du connecteur provoque une tension qui s'ajoute à la tension du thermocouple capteur. Si l'on ne corrige pas cette tension, la température indiquée sera différente de la température du capteur.

On utilise donc la compensation de soudure froide pour compenser le thermocouple du connecteur. Les tables de données de thermocouples se basent sur une température de soudure de référence, 0 degré Celsius en général. La compensation de soudure froide compense le connecteur à 0 degré Celsius. Elle restaure la tension ajoutée par les thermocouples du connecteur. La température du module est mesurée en interne, puis convertie en une valeur à ajouter à la conversion de capteur. On linéarise ensuite la conversion de capteur corrigée à l'aide des tables de thermocouples.

Pour un fonctionnement optimal de la compensation de soudure froide, le module Thermocouple doit se trouver dans un environnement thermique stable. Une variation lente (inférieure à 0,1 °C/minute) à une température ambiante du module est correctement compensée dans les limites des spécifications du module. Un déplacement d'air à travers le module entraînera également des erreurs de compensation de la soudure froide.

Si une meilleure compensation d'erreur de soudure froide est souhaitée, un bornier isotherme externe peut être utilisé. Le module thermocouple met à disposition un bornier référencé à 0 °C ou à 50 °C.

Tableau de sélection pour le SB 1231 Thermocouple

Les plages et la précision des différents types de thermocouple pris en charge par le Signal Board SB 1231 Thermocouple sont présentées dans le tableau ci-dessous.

Tableau A- 225 Tableau de sélection de thermocouples

Type	Dépasse- ment bas minimum ¹	Limite basse plage nomi- nale	Limite haute plage nomi- nale	Dépas- sement haut maxi- mum ²	Précision plage normale ^{3,4} 25 °C	Précision plage normale ^{1,2 -} 20 °C à 60 °C
J	- 210,0 °C	- 150,0 °C	1 200,0 °C	1 450,0 ° C	± 0,3 °C	± 0,6 °C
	- 346.0 °F	- 238.0 °F	2192.0 °F	2642.0 °F	± 0.5 °F	± 1.1 °F
K	- 270,0 °C	- 200,0 °C	1 372,0 °C	1 622,0 ° C	± 0,4 °C	± 1,0 °C
	- 454.0 °F	- 328.0 °F	2501.6 °F	2951.6 °F	± 0.7 °F	± 1.8 °F
T	- 270,0 °C	- 200,0 °C	400,0 °C	540,0 °C	± 0,5 °C	± 1,0 °C
	- 454.0 °F	- 328.0 °F	752.0 °F	1004.0 °F	± 0.9 °F	± 1.8 °F
E	- 270,0 °C	- 200,0 °C	1 000,0 °C	1 200,0 ° C	± 0,3 °C	± 0,6 °C
	- 454.0 °F	- 328.0 °F	1832.0 °F	2192.0 °F	± 0.5 °F	± 1.1 °F
R & S	- 50,0 °C	100,0 °C	1 768,0 °C	2 019,0 ° C	± 1,0 °C	± 2,5 °C
	- 58.0 °F	212.0 °F	3214.4 °F	3276.6 °F ⁵	± 1.8 °F	± 4.5 °F

Type	Dépassement bas minimum ¹	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum ²	Précision plage normale ^{3,4} 25 °C	Précision plage normale ^{1,2} 20 °C à 60 °C
B	0,0 °C	200,0 °C	800,0 °C	--	± 2,0 °C	± 2,5 °C
	32.0 °F	392.0 °F	1472.0 °F	--	± 3.6 °F	± 4.5 °F
	--	800,0 °C	1 820,0 °C	1 820,0 °C	± 1,0 °C	± 2,3 °C
	--	1472.0 °F	3276.6 °F ⁵	3276.6 °F ⁵	± 1.8 °F	± 4.1 °F
N	- 270,0 °C	- 200,0 °C	1 300,0 °C	1 550,0 °C	± 1,0 °C	± 1,6 °C
	- 454.0 °F	- 328.0 °F	2372.0 °F	2822.0 °F	± 1.8 °F	± 2.9 °F
C	0,0 °C	100,0 °C	2 315,0 °C	2 500,0 °C	± 0,7 °C	± 2,7 °C
	32.0 °F	212.0 °F	3276.6 °F ⁵	3276.6 °F ⁵	± 1.3 °F	± 4.9 °F
TXK/XK(L)	- 200,0 °C	- 150,0 °C	800,0 °C	1 050,0 °C	± 0,6 °C	± 1,2 °C
	- 328.0 °F	302.0 °F	1472.0 °F	1922.0 °F	± 1.1 °F	± 2.2 °F
Tension	- 32512	- 27648 - 80 mV	27648 80 mV	32511	± 0,05 %	± 0,1 %

¹ Les valeurs de thermocouple inférieures à la valeur de dépassement bas minimum sont signalées en tant que -32768.

² Les valeurs de thermocouple supérieures à la valeur de dépassement haut maximum sont signalées en tant que 32767.

³ L'erreur de soudure froide interne est de ± 1,5 °C pour toutes les plages. Cette valeur s'ajoute à l'erreur indiquée dans ce tableau. Le module requiert au moins 30 minutes de temps de chauffe pour atteindre cette spécification.

⁴ La précision du SM 1231 AI 4 x 16 bits TC peut être réduite en présence de radiofréquence rayonnée entre 970 MHz et 990 MHz.

⁵ Limite inférieure de 3276.6 °F pour la signalisation en °F

Tableau A- 226 Tableau de sélection de filtre pour le SB 1231 Thermocouple

Fréquence de réjection (Hz)	Temps d'intégration (ms)	Temps d'actualisation du Signal Board (secondes)
10	100	0.306
50	20	0.066
60	16.67	0.056
400 ¹	10	0.036

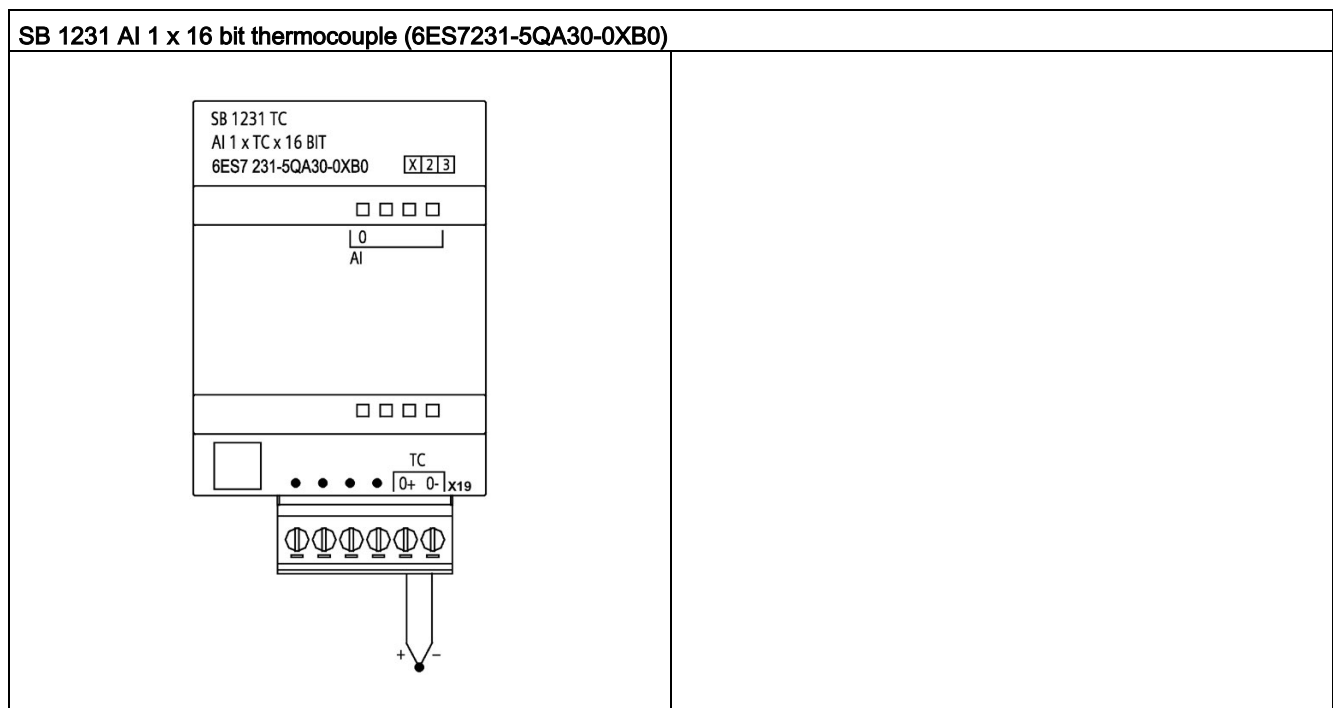
¹ Le temps d'intégration est de 10 ms pour conserver la résolution et la précision du module lorsque la réjection 400 Hz est sélectionnée. Cette sélection correspond également à une réjection des bruits 100 Hz et 200 Hz.

Il est recommandé d'utiliser un temps d'intégration de 100 ms pour la mesure des thermocouples. Le choix de temps d'intégration inférieurs augmentera l'erreur de répétabilité des températures lues.

Remarque

Après sa mise sous tension, le module exécute un calibrage interne pour le convertisseur analogique-numérique. Pendant ce temps, il émet une valeur de 32767 sur chaque voie jusqu'à ce que des données valables soient disponibles sur cette voie. Il peut s'avérer nécessaire que votre programme utilisateur prévoie ce temps d'initialisation.

Tableau A- 227 Schéma de câblage du SB 1231 AI 1 x 16 thermocouple



Remarque : les connecteurs doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Tableau A- 228 Brochage pour le SB 1231 AI 1 x 16 bit thermocouple (6ES7231-5QA30-0XB0)

Broche	X19 (or)
1	Pas de connexion
2	Pas de connexion
3	Pas de connexion
4	Pas de connexion
5	AI 0- /TC
6	AI 0+ /TC

A.14.5 Signal Boards (SB) RTD

A.14.5.1 Caractéristiques du SB 1231 1 entrée analogique RTD

Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ce SB.

Tableau A- 229 Caractéristiques générales

Caractéristiques techniques	SB 1231 AI 1 x 16 bits RTD
Référence	6ES7231-5PA30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 2
Poids	35 grammes
Dissipation de courant	0,7 W
Consommation de courant (bus SM)	5 mA
Consommation de courant (24 V CC)	25 mA

Tableau A- 230 Entrées analogiques

Caractéristiques techniques	SB 1231 AI 1 x 16 bits RTD	
Nombre d'entrées	1	
Type	RTD référencée au module et ohms	
Plage <ul style="list-style-type: none"> • Plage nominale (mot de données) • Dépassement haut/bas (mot de données) • Débordement haut / bas (mot de données) 	Voir les tableaux de sélection (Page 1618).	
Résolution	Température	0,1 °C/0,1 °F
	Tension	15 bits plus signe
Tension de tenue maximum	± 35 V	
Réjection des bruits	85 dB (10 Hz, 50 Hz, 60 Hz, 400 Hz)	
Réjection en mode commun	> 120 dB	
Impédance	≥ 10 MΩ	
Précision	Voir les tableaux de sélection (Page 1618).	
Répétabilité	± 0,05% pleine échelle	
Dissipation maximale du capteur	0,5 mW	
Principe de mesure	Intégration	
Temps d'actualisation du module	Voir le tableau de sélection (Page 1618).	
Isolation (site à logique)	707 V CC (essai de type)	

Caractéristiques techniques	SB 1231 AI 1 x 16 bits RTD
Longueur de câble (mètres)	max. 100 m au capteur
Résistance de fil	20 Ω, 2,7 pour RTD 10 Ω max.

Tableau A- 231 Diagnostic

Caractéristiques techniques	SB 1231 AI 1 x 16 bits RTD
Dépassement haut / bas ^{1,2}	Oui
Rupture de fil ³	Oui

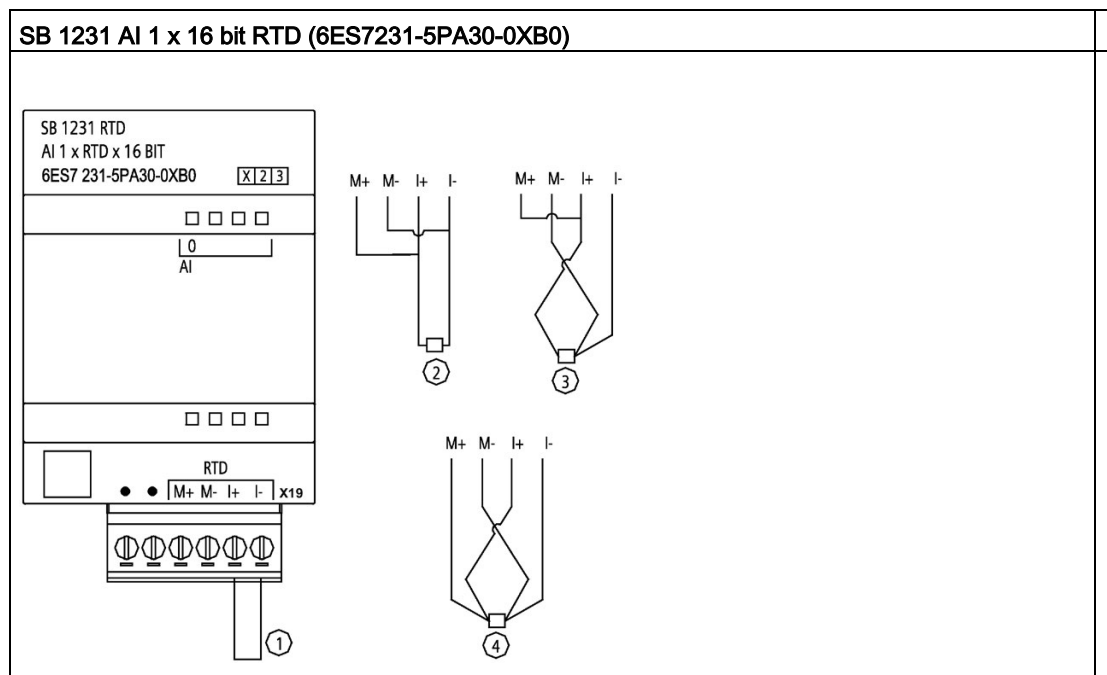
- ¹ Les informations d'alarmes de diagnostic de débordement haut et bas sont notifiées dans les valeurs de données analogiques même si les alarmes sont désactivées dans la configuration du module.
- ² La détection de débordement bas n'est jamais activée pour les plages de résistance.
- ³ Lorsque l'alarme de rupture de fil est désactivée et qu'un fil est ouvert dans le câblage du capteur, le module peut émettre des valeurs aléatoires.

Le Signal Board analogique SB 1231 RTD mesure la valeur de résistance connectée aux entrées du Signal Board. Il est possible de sélectionner "Résistance" ou "Thermistance" comme type de mesure.

- "Résistance" : La valeur pleine échelle de la plage nominale sera 27648 en décimal.
- "Thermistance" : La valeur sera notifiée en degrés multipliés par dix (par exemple, 25,3 degrés seront indiqués sous la forme 253 en décimal). La plage des valeurs climatiques sera notifiée en degrés multipliés par cent (par exemple, 25,34 degrés seront indiqués sous la forme 2534 en décimal).

Le Signal Board SB 1231 RTD prend en charge des montages 2 fils, 3 fils et 4 fils avec la résistance du capteur.

Tableau A- 232 Schéma de câblage du SB 1231 AI 1 x 16 bits RTD



- ① Bouclage sur l'entrée RTD inutilisée
- ② RTD 2 fils
- ③ RTD 3 fils
- ④ RTD 4 fils

Remarque : les connecteurs doivent être en or. Vous trouverez les références à l'annexe C, "Pièces de rechange".

Tableau A- 233 Brochage pour le SB 1231 AI 1 x 16 bit RTD (6ES7231-5PA30-0XB0)

Broche	X19 (or)
1	Pas de connexion
2	Pas de connexion
3	AI 0 M+ /RTD
4	AI 0 M- /RTD
5	AI 0 I+ /RTD
6	AI 0 I- /RTD

A.14.5.2 Tableaux de sélection pour le SB 1231 RTD

Tableau A- 234 Plages et précision pour les différents capteurs pris en charge par les modules RTD

Coefficient de température	Type de RTD	Dépassement bas minimum ¹	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum ²	Précision de plage normale à 25 °C	Précision de plage normale de - 20 °C à 60 °C
Pt 0.003850 ITS90 DIN EN 60751	Pt 100 plage climatique	- 145,00 °C	- 120,00 °C	- 145,00 °C	- 155,00 °C	± 0,20 °C	± 0,40 °C
	Pt 10	- 243,0 °C	- 200,0 °C	850,0 °C	1 000,0 °C	± 1,0 °C	± 2,0 °C
	Pt 50	- 243,0 °C	- 200,0 °C	850,0 °C	1 000,0 °C	± 0,5 °C	± 1,0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0.003902 Pt 0.003916 Pt 0.003920	Pt 100	- 243,0 °C	- 200,0 °C	850,0 °C	1 000,0 °C	± 0,5 °C	± 1,0 °C
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	- 273,2 °C	- 240,0 °C	1 100,0 °C	1 295 °C	± 1,0 °C	± 2,0 °C
	Pt 50	- 273,2 °C	- 240,0 °C	1 100,0 °C	1 295 °C	± 0,8 °C	± 1,6 °C
	Pt 100						
	Pt 500						
Ni 0.006720 Ni 0.006180	Ni 100	- 105,0 °C	- 60,0 °C	250,0 °C	295,0 °C	± 0,5 °C	± 1,0 °C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0.005000	LG-Ni 1000	- 105,0 °C	- 60,0 °C	250,0 °C	295,0 °C	± 0,5 °C	± 1,0 °C
Ni 0.006170	Ni 100	- 105,0 °C	- 60,0 °C	180,0 °C	212,4 °C	± 0,5 °C	± 1,0 °C
Cu 0.004270	Cu 10	- 240,0 °C	- 200,0 °C	260,0 °C	312,0 °C	± 1,0 °	± 2,0 °C
Cu 0.004260	Cu 10	- 60,0 °C	- 50,0 °C	200,0 °C	240,0 °C	± 1,0 °C	± 2,0 °C
	Cu 50	- 60,0 °C	- 50,0 °C	200,0 °C	240,0 °C	± 0,6 °C	± 1,2 °C
	Cu 100						
Cu 0.004280	Cu 10	- 240,0 °C	- 200,0 °C	200,0 °C	240,0 °C	± 1,0 °C	± 2,0 °C
	Cu 50	- 240,0 °C	- 200,0 °C	200,0 °C	240,0 °C	± 0,7 °C	± 1,4 °C
	Cu 100						

¹ Les valeurs RTD inférieures à la valeur de dépassement bas minimum sont signalées en tant que -32768.

² Les valeurs RTD supérieures à la valeur de dépassement haut maximum sont signalées en tant que +32768.

Tableau A- 235 Résistance

Plage	Dépassement bas minimum	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum ¹	Précision de plage normale à 25 °C	Précision de plage normale de - 20 °C à 60 °C
150 Ω	Sans objet	0 (0 Ω)	27648 (150 Ω)	176,383 Ω	± 0,05 %	± 0,1 %
300 Ω	Sans objet	0 (0 Ω)	27648 (300 Ω)	352,767 Ω	± 0,05 %	± 0,1 %
600 Ω	Sans objet	0 (0 Ω)	27648 (600 Ω)	705,534 Ω	± 0,05 %	± 0,1 %

¹ Les valeurs de résistance supérieures à la valeur de dépassement haut maximum sont signalées comme 32767.

Remarque

Le module signale 32767 sur toute voie activée à laquelle aucun capteur n'est connecté. Si la détection de fil ouvert est également activée, le module fait clignoter les DEL appropriées.

La meilleure précision est obtenue pour les plages RTD 10 Ω avec des montages 4 fils.

La résistance des fils de connexion en montage 2 fils provoquera une erreur dans la lecture du capteur. La précision ne peut donc pas être garantie.

Tableau A- 236 Réduction des bruits et temps d'actualisation pour les modules RTD

Sélection de la fréquence de réjection	Temps d'intégration	Module 1 voie, 4 / 2 fils	Module 1 voie, 3 fils
		Temps d'actualisation (secondes)	Temps d'actualisation (secondes)
400 Hz (2,5 ms)	10 ms ¹	0,036	0,071
60 Hz (16,6 ms)	16,67 ms	0,056	0,111
50 Hz (20 ms)	20 ms	0,066	1,086
10 Hz (100 ms)	100 ms	0,306	0,611

¹ Le temps d'intégration est de 10 ms pour conserver la résolution et la précision du module lorsque le filtre 400 Hz est sélectionné. Cette sélection correspond également à une réjection des bruits 100 Hz et 200 Hz.

Remarque

Après sa mise sous tension, le module exécute un calibrage interne pour le convertisseur analogique-numérique. Pendant ce temps, il émet une valeur de 32767 sur chaque voie jusqu'à ce que des données valables soient disponibles sur cette voie. Il peut s'avérer nécessaire que votre programme utilisateur prévoie ce temps d'initialisation. Comme la configuration du module peut faire varier la longueur du temps d'initialisation, vous devez vérifier le comportement du module dans votre configuration. Si nécessaire, vous pouvez inclure de la logique dans votre programme utilisateur afin de prendre en compte le temps d'initialisation du module.

A.15 Battery Board BB 1297

Battery Board BB 1297

Le Battery Board BB 1297 S7-1200 est conçu pour une sauvegarde à long terme de l'horloge temps réel. Il s'enfiche dans le logement pour Signal Board de la CPU S7-1200 (firmware 3.0 et versions ultérieures). Pour que le BB 1297 soit opérationnel, vous devez l'ajouter à la configuration des appareils et charger dans la CPU la configuration matérielle pour le BB.

La pile (de type CR1025) n'est pas fournie avec le BB 1297 et doit être achetée par l'utilisateur.

Remarque

La conception mécanique du BB 1297 est adaptée aux CPU de firmware 3.0 et plus.

N'utilisez pas le BB 1297 avec des CPU de version antérieure, car son connecteur ne rentrera pas dans la CPU.

 **ATTENTION**

Installer une pile non spécifiée dans le BB 1297, ou connecter de toute autre manière une pile non spécifiée au circuit peut provoquer un incendie ou endommager les composants et peut entraîner un fonctionnement imprévisible des machines.

Un incendie ou un fonctionnement imprévisible des machines peut entraîner la mort, des blessures graves ou des dommages matériels.

Utilisez uniquement une pile CR1025 spécifiée pour effectuer une sauvegarde de l'horloge temps réel.

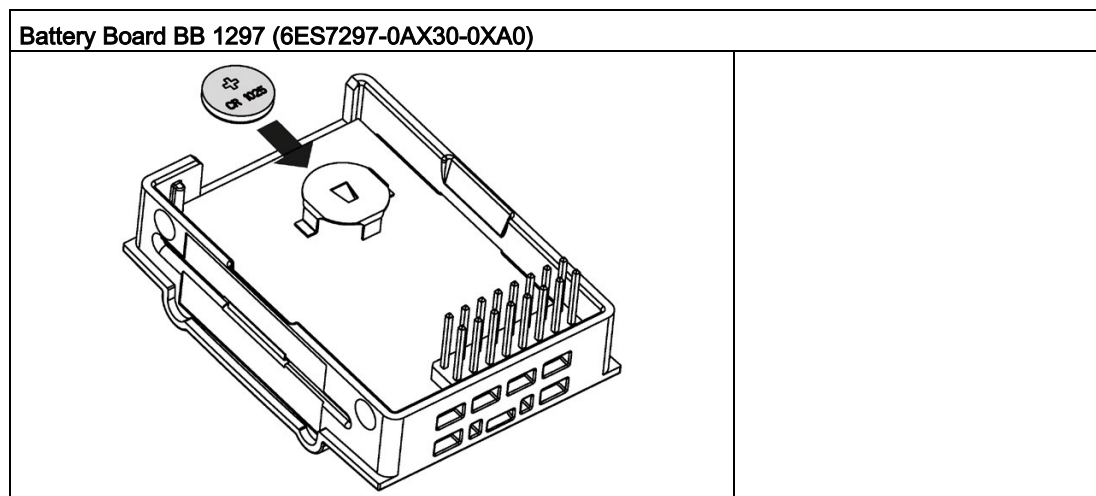
Tableau A- 237 Caractéristiques générales

Caractéristiques techniques	Battery Board BB 1297
Référence	6ES7297-0AX30-0XA0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	28 grammes
Dissipation de courant	0,5 W
Consommation de courant (bus SM)	11 mA
Consommation de courant (24 V CC)	aucune

Pile (non incluse)	Battery Board BB 1297
Temps de conservation	Environ 1 an
Type de pile	CR1025 ; voir Installation ou remplacement de la pile dans le Battery Board BB 1297 (Page 64)
Tension nominale	3 V
Capacité nominale	Au moins 30 mAh

Diagnostic	Battery Board BB 1297
Niveau de pile critique	< 2,5 V
Diagnostic de pile	Indicateur de tension basse : <ul style="list-style-type: none"> Lorsque la tension de la pile est faible, la DEL de maintenance de la CPU s'allume en continu (orange). Evénement de tampon de diagnostic : 16#06:2700 "Requête de maintenance du sous-module : Au moins une batterie épuisée (BATTF)"
Etat de la pile	Bit d'état de batterie fourni 0 = batterie OK 1 = batterie faible
Actualisation de l'état de la pile	L'état de la pile est actualisé à la mise sous tension, puis une fois par jour lorsque la CPU est à l'état MARCHÉ.

Tableau A- 238 Schéma d'insertion pour le Battery Board BB 1297



A.16 Interfaces de communication

A.16.1 PROFIBUS

A.16.1.1 ESCLAVE PROFIBUS DP CM 1242-5

Tableau A- 239 Caractéristiques techniques du CM 1242-5

Caractéristiques techniques	
Référence	6GK7242-5DX30-0XE0
Interfaces	
Connexion à PROFIBUS	Connecteur femelle subD 9 broches
Consommation maximale de courant sur l'interface PROFIBUS lors de la connexion de composantes réseau (par exemple composantes de réseau optique)	15 mA à 5 V (uniquement pour terminaison de bus) *)
Conditions ambiantes autorisées	
Température ambiante <ul style="list-style-type: none"> • pendant le stockage • pendant le transport • pendant le fonctionnement avec une installation verticale (profilé support horizontal) • pendant le fonctionnement avec une installation horizontale (profilé support vertical) 	<ul style="list-style-type: none"> • -40 °C à 70 °C • -40 °C à 70 °C • 0 °C à 55 °C • 0 °C à 45 °C
Humidité relative à 25 °C durant le fonctionnement, sans condensation, maximum	95 %
Degré de protection	IP20
Alimentation, consommation de courant et dissipation de courant	
Type d'alimentation	Courant continu
Alimentation électrique depuis le bus interne	5 V
Consommation de courant (typique)	150 mA
Perte de puissance effective (typique)	0,75 W
Séparation galvanique <ul style="list-style-type: none"> • Interface PROFIBUS à la terre • Interface PROFIBUS au circuit interne 	710 V CC pour 1 minute
Dimensions et poids	
<ul style="list-style-type: none"> • Largeur • Hauteur • Profondeur 	<ul style="list-style-type: none"> • 30 mm • 100 mm • 75 mm

Caractéristiques techniques	
Poids	
<ul style="list-style-type: none"> • Poids net • Poids avec l'emballage 	<ul style="list-style-type: none"> • 115 g • 152 g

*)La charge actuelle d'un client externe connecté entre VP (broche 6) et DGND (broche 5) ne doit pas dépasser un maximum de 15 mA (résistance au court-circuit) pour la terminaison de bus.

A.16.1.2 Brochage de la prise sub D du CM 1242-5

Interface PROFIBUS

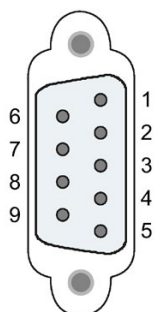


Tableau A- 240 Brochage de la prise subD

Broche	Description	Broche	Description
1	- non utilisée -	6	P5V2 : alimentation électrique +5V
2	- non utilisée -	7	- non utilisée -
3	RxD/TxD-P : Ligne de données B	8	RxD/TxD-N : Ligne de données A
4	RTS	9	- non utilisée -
5	M5V2 : Potentiel de référence de données (DGND terre)	Boîtier	Raccord de mise à la terre

A.16.1.3 Maître PROFIBUS DP CM 1243-5

Tableau A- 241 Caractéristiques techniques du CM 1243-5

Caractéristiques techniques	
Référence	6GK7243-5DX30-0XE0
Interfaces	
Connexion à PROFIBUS	Connecteur femelle subD 9 broches
Consommation maximale de courant sur l'interface PROFIBUS lors de la connexion de composants réseau (par exemple composantes de réseau optique)	15 mA à 5 V (uniquement pour terminaison de bus) *)
Conditions ambiantes autorisées	
Température ambiante	<ul style="list-style-type: none"> • pendant le stockage • pendant le transport • pendant le fonctionnement avec une installation verticale (profilé support horizontal) • pendant le fonctionnement avec une installation horizontale (profilé support vertical)
	<ul style="list-style-type: none"> • -40 °C à 70 °C • -40 °C à 70 °C • 0 °C à 55 °C • 0 °C à 45 °C
Humidité relative à 25 °C durant le fonctionnement, sans condensation, maximum	95 %
Degré de protection	IP20
Alimentation, consommation de courant et dissipation de courant	
Type d'alimentation	Courant continu
Alimentation électrique / externe	24 V
<ul style="list-style-type: none"> • minimum • maximum 	<ul style="list-style-type: none"> • 19,2 V • 28,8 V
Consommation de courant (typique)	
<ul style="list-style-type: none"> • à partir de 24 V CC • depuis le bus interne du S7-1200 	<ul style="list-style-type: none"> • 100 mA • 0 mA
Perte de puissance effective (typique)	
<ul style="list-style-type: none"> • à partir de 24 V CC • depuis le bus interne du S7-1200 	<ul style="list-style-type: none"> • 2,4 W • 0 W
Alimentation 24 V CC/externe	
<ul style="list-style-type: none"> • Section du câble min. • Section du câble max. • Couple de serrage des bornes à vis 	<ul style="list-style-type: none"> • min. : 0,14 mm² (AWG 25) • max. : 1,5 mm² (AWG 15) • 0,45 Nm (4 lb-in)
Séparation galvanique	710 V CC pour 1 minute
<ul style="list-style-type: none"> • Interface PROFIBUS à la terre • Interface PROFIBUS au circuit interne 	

Caractéristiques techniques	
Dimensions et poids	
<ul style="list-style-type: none"> • Largeur • Hauteur • Profondeur 	<ul style="list-style-type: none"> • 30 mm • 100 mm • 75 mm
Poids	
<ul style="list-style-type: none"> • Poids net • Poids avec l'emballage 	<ul style="list-style-type: none"> • 134 g • 171 g

*)La charge actuelle d'un client externe connecté entre VP (broche 6) et DGND (broche 5) ne doit pas dépasser un maximum de 15 mA (résistance au court-circuit) pour la terminaison de bus.

Remarque

Le CM 1243-5 (module maître PROFIBUS) doit être alimenté en courant par l'alimentation de capteur 24 V CC de la CPU.

A.16.1.4 Brochage de la prise subD du CM 1243-5

Interface PROFIBUS

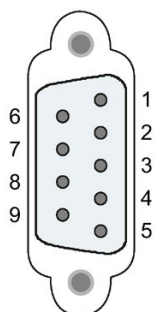


Tableau A- 242 Brochage de la prise subD

Broche	Description	Broche	Description
1	- non utilisée -	6	VP : Alimentation +5 V uniquement pour les résistances de terminaison de bus ; pas pour alimenter des périphériques externes
2	- non utilisée -	7	- non utilisée -
3	RxD/TxD-P : Ligne de données B	8	RxD/TxD-N : Ligne de données A
4	CNTR-P : RTS	9	- non utilisée -
5	DGND : Terre pour les signaux de données et VP	Boîtier	Raccord de mise à la terre

Câble PROFIBUS

Remarque

Mise en contact du blindage du câble PROFIBUS

Le blindage du câble PROFIBUS doit être mis en contact.

Pour ce faire, dénudez l'isolation de l'extrémité du câble PROFIBUS et reliez le blindage à la terre fonctionnelle.

A.16.2 CP 1242-7

Remarque

Le CP 1242-7 n'est pas homologué pour le domaine maritime

Le CP 1242-7 ne dispose pas de l'homologation pour le domaine maritime :

Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ces modules.

A.16.2.1 CP 1242-7 GPRS

Tableau A- 243 Caractéristiques techniques du CP 1242-7 GPRS V2

Caractéristiques techniques	
Référence	6GK7242-7KX3-0XE0
Interface sans fil	
Connecteur d'antenne	Prise SMA
Impédance nominale	50 ohms
Liaison sans fil	
Puissance d'émission maximum	<ul style="list-style-type: none">• GSM 850, classe 4 : +33 dBm ±2dBm• GSM 900, classe 4 : +33 dBm ±2dBm• GSM 1800, classe 1 : +30 dBm ±2dBm• GSM 1900, classe 1 : +30 dBm ±2dBm
GPRS	Multi-emplacement classe 10 classe d'appareil B schéma de codage 1...4 (GMSK)
SMS	Mode disparaissant : MO service : point à point

Caractéristiques techniques	
Conditions ambiantes autorisées	
Température ambiante <ul style="list-style-type: none"> • pendant le stockage • pendant le transport • pendant le fonctionnement avec une installation verticale (profilé support horizontal) • pendant le fonctionnement avec une installation horizontale (profilé support vertical) 	<ul style="list-style-type: none"> • -40 °C à 70 °C • -40 °C à 70 °C • 0 °C à 55 °C • 0 °C à 45 °C
Humidité relative à 25 °C durant le fonctionnement, sans condensation, maximum	95 %
Degré de protection	IP20
Alimentation, consommation de courant et dissipation de courant	
Type d'alimentation	Courant continu
Alimentation électrique / externe <ul style="list-style-type: none"> • minimum • maximum 	24 V <ul style="list-style-type: none"> • 19,2 V • 28,8 V
Consommation de courant (typique) <ul style="list-style-type: none"> • à partir de 24 V CC • depuis le bus interne du S7-1200 	<ul style="list-style-type: none"> • 100 mA • 0 mA
Perte de puissance effective (typique) <ul style="list-style-type: none"> • à partir de 24 V CC • depuis le bus interne du S7-1200 	<ul style="list-style-type: none"> • 2,4 W • 0 W
Alimentation 24 V CC <ul style="list-style-type: none"> • Section du câble min. • Section du câble max. • Couple de serrage des bornes à vis 	<ul style="list-style-type: none"> • min. : 0,14 mm² (AWG 25) • max. : 1,5 mm² (AWG 15) • 0,45 Nm (4 lb-in)
Séparation galvanique Unité d'alimentation électrique au circuit interne	710 V CC pour 1 minute
Dimensions et poids	
<ul style="list-style-type: none"> • Largeur • Hauteur • Profondeur 	<ul style="list-style-type: none"> • 30 mm • 100 mm • 75 mm
Poids <ul style="list-style-type: none"> • Poids net • Poids avec l'emballage 	<ul style="list-style-type: none"> • 133 g • 170 g

Remarque

Empêcher les interférences entre CPU et antennes

Des interférences avec la CPU peuvent se produire si des antennes sont trop proches ou si vous n'utilisez pas les antennes recommandées. Pour les antennes recommandées, voir la notice de service Antenna ANT794-4MR for LTE/UMTS/GSM (<https://support.industry.siemens.com/cs/ww/fr/view/23119005/en>) (disponible en anglais et en allemand uniquement).

A.16.2.2 Antenne GSM/GPRS ANT794-4MR

Caractéristiques techniques de l'antenne GSM/GPRS ANT794-4MR

ANT794-4MR	
Référence	6NH9860-1AA00
Réseaux mobiles sans fil	GSM/GPRS
Plages de fréquence	<ul style="list-style-type: none"> • 824 à 960 MHz (GSM 850, 900) • 1 710 à 1 880 MHz (GSM 1 800) • 1 900 à 2 200 MHz (GSM / UMTS)
Particularités	omnidirectionnel
Gain d'antenne	0 dB
Impédance	50 ohms
Rapport d'ondes stationnaires (SWR)	< 2,0
Puissance max.	20 W
Polarité	linéaire verticale
Connecteur	SMA
Longueur du câble d'antenne	5 m
Matériau extérieur	PVC dur, résistant aux UV
Degré de protection	IP20
Conditions ambiantes autorisées	<ul style="list-style-type: none"> • Température de fonctionnement • Température de transport/stockage • Humidité relative
	<ul style="list-style-type: none"> • -40 °C à +70 °C • -40 °C à +70 °C • 100 %
Matériau extérieur	PVC dur, résistant aux UV
Construction	Antenne avec câble fixe de 5 m et connecteur mâle SMA
Dimensions (P x H) en mm	25 x 193
Poids	<ul style="list-style-type: none"> • Antenne comprenant un câble • Raccords
	<ul style="list-style-type: none"> • 310 g • 54 g
Installation	Avec support fourni

A.16.2.3 Antenne en nappe ANT794-3M

Caractéristiques techniques de l'antenne en nappe ANT794-3M

ANT794-3M		
Référence	6NH9870-1AA00	
Réseaux mobiles sans fil	GSM 900	GSM 1800/1900
Plages de fréquence	890 - 960 MHz	1710 - 1990 MHz
Rapport d'ondes stationnaires (VSWR)	≤ 2:1	≤ 1,5:1
Affaiblissement d'adaptation (Tx)	≈ 10 dB	≈ 14 dB
Gain d'antenne	0 dB	
Impédance	50 ohms	
Puissance max.	10 W	
Câble d'antenne	Câble HF RG 174 (fixe) avec connecteur mâle SMA	
Longueur de câble	1,2 m	
Degré de protection	IP64	
Plage de températures autorisée	-40 °C à +75 °C	
Inflammabilité	UL 94 V2	
Matériau extérieur	ABS Polylac PA-765, gris clair (RAL 7035)	
Dimensions (l x L x H) en mm	70,5 x 146,5 x 20,5	
Poids	130 g	

A.16.3 CM 1243-2 maître AS-i

A.16.3.1 Données techniques pour l'interface maître AS-i CM 1243-2

Tableau A- 244 Caractéristiques techniques de l'interface maître AS-i CM 12432

Caractéristiques techniques	
Référence	3RK7243-2AA30-0XB0
Version de firmware	V1.0
Date	01.12.2011
Interfaces	
Consommation maximale de courant Depuis le bus interne du S7-1200 Depuis le câble AS-i	Max. 250 mA, tension d'alimentation du bus de communication S7-1200 5 V CC max. 100 mA
Capacité de transport de courant maximale entre les bornes ASI+/ASI-	8 A
Brochage	Reportez-vous à la rubrique Raccordements électriques au maître AS-i (Page 1631)
Section du conducteur	0,2 mm ² (AWG 24) ... 3,3 mm ² (AWG 12)
Couple de serrage du connecteur ASI	0,56 Nm
Conditions ambiantes admissibles	
Température ambiante Pendant le stockage Pendant le transport Durant le fonctionnement, avec une installation verticale (rail de montage standard horizontal) Durant le fonctionnement, avec une installation horizontale (rail de montage standard vertical)	-40 °C ... 70 °C -40 °C ... 70 °C 0 °C ... 55 °C 0 °C... 45 °C
Humidité relative à 25 °C durant le fonctionnement, aucune condensation, maximum	95 %
Degré de protection	IP20
Alimentation, consommation de courant, dissipation de courant	
Type d'alimentation	Courant continu
Consommation de courant (typ.) Depuis le bus interne du S7-1200	200 mA
Perte de puissance totale (typique) : • Depuis le bus interne du S7-1200 • Depuis le câble AS-i	1 W 2,4 W

Caractéristiques techniques	
Dimensions et poids	
Largeur	30 mm
Hauteur	100 mm
Profondeur	75 mm
Poids	
Poids net	122 g
Poids avec l'emballage	159 g

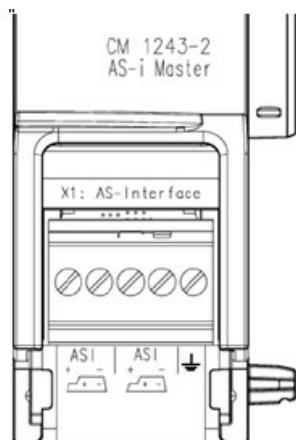
A.16.3.2 Raccordements électriques au maître AS-i

Alimentation de l'interface maître AS-i CM 1243-2

L'interface maître AS-i CM 1243-2 est alimentée via le bus de communication du S7-1200. Cela signifie qu'un message de diagnostic peut être envoyé vers le S7-1200 même après une défaillance de la tension d'alimentation de l'interface AS-i. La connexion au bus de communication se trouve à droite de l'interface maître AS-i CM 1243-2.

Terminaux de l'interface AS

Le terminal amovible permettant de connecter le câble AS-i se trouve derrière le volet frontal inférieur de l'interface AS-i master CM 1243-2.



Si le câble AS-i est utilisé, vous pouvez vérifier la polarité du câble à l'aide du symbole



Vous trouverez les informations relatives au retrait et à la réinstallation du bornier dans le Chapitre d'installation (Page 69).

Remarque


Capacité de transport de courant maximale des contacts de bornes

La capacité de transport de courant des contacts de connexion est de max. 8 A. Si cette valeur est dépassée sur le câble AS-i, le maître AS-i CM 1243-2 ne doit pas être "mis en boucle" avec le câble AS-i, mais doit être au contraire être connecté via un embranchement (une seule paire de connexions affectée sur le maître AS-i CM 1243-2).

Veillez également vous assurer que les câbles utilisés sont adaptés à des températures de fonctionnement d'au moins 75 °C si le courant est conduit via le maître AS-i et si des courants supérieurs à 4 ampères sont présents.

Vous trouverez des informations supplémentaires sur le raccordement du câble AS-i dans la rubrique "Installation, connexion et mise en service des modules" dans le manuel "Unité de découplage maître AS-i CM 1243-2 et données AS-i DCM 1271 pour SIMATIC S7-1200".

Affectation des bornes

Repère	Signification
ASI+	Connexion AS-i – polarité positive
ASI-	Connexion AS-i – polarité négative
	Terre fonctionnelle

A.16.4 RS232, RS422 et RS485

A.16.4.1 Caractéristiques CB 1241 RS485

Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ce CB.

Tableau A- 245 Caractéristiques générales

Caractéristiques techniques	CB 1241 RS485
Référence	6ES7241-1CH30-1XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	40 grammes

Tableau A- 246 Emetteur et récepteur

Caractéristiques techniques	CB 1241 RS485
Type	RS485 (semi-duplex 2 fils)
Plage de tension en mode commun	-7 V à +12 V, 1 seconde, 3 V eff. continu
Tension de sortie différentielle émetteur	2 V min. pour $R_L = 100 \Omega$ 1,5 V min. pour $R_L = 54 \Omega$
Terminaison et polarisation	10K vers +5 V sur B, RS485 broche 3 10K vers GND sur A, RS485 broche 4
Terminaison optionnelle	Court-circuiter broche TB sur broche T/RB, impédance de terminaison effective 127 Ω , connexion à RS485 broche 3 Court-circuiter broche TA sur broche T/RA, impédance de terminaison effective 127 Ω , connexion à RS485 broche 4
Impédance d'entrée de récepteur	5,4 k Ω min., terminaison incluse
Seuil/sensibilité récepteur	+/- 0,2 V min., hystérésis typique 60 mV
Isolation signal RS485 à la terre du châssis signal RS485 au commun logique CPU	707 V CC (essai de type)
Longueur de câble, blindé	1000 m max.
Débit en bauds	300 bauds, 600 bauds, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (par défaut), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Parité	Sans parité (par défaut), parité paire, parité impaire, parité marque (bit de parité toujours à 1), parité espace (bit de parité toujours à 0)
Nombre de bits d'arrêt	1 (par défaut), 2
Contrôle du flux	Non pris en charge
Temps d'attente	0 à 65 535 ms

Tableau A- 247 Alimentation

Caractéristiques techniques	CB 1241 RS485
Perte de puissance (dissipation)	1,5 W
Consommation de courant (bus SM), max.	50 mA
Consommation de courant (24 V CC) max.	80 mA

CB 1241 RS485 (6ES7241-1CH30-1XB0)	
<p>The diagram shows the terminal block for the CB 1241 RS485 module. It features a 5-pin terminal block with pins labeled M, TA, T1RA, T1RB, and TB. Below the terminal block, a shielded twisted pair cable is shown with its braid connected to ground. The cable is labeled with circled numbers 1 and 2. The terminal block also has labels for TXD, RXD, and X20: RS485.</p>	
<p>① Connectez "TA" et TB" comme indiqué pour terminer le réseau. La terminaison ne concerne que les appareils aux extrémités du réseau RS485.</p>	
<p>② Utilisez un câble à paire torsadée blindée et connectez le blindage du câble à la terre.</p>	

La terminaison ne concerne que les deux extrémités du réseau RS485. Il n'y a pas de terminaison ni de polarisation des appareils situés entre les deux appareils d'extrémité. Voir le chapitre "Polarisation et terminaison d'un connecteur de réseau RS485" (Page 1130).

Tableau A- 248 Brochage pour le CB 1241 RS485 (6ES7241-1CH30-1XB0)

Broche	Connecteur à 9 broches	X20
1	RS485 / Terre logique	--
2	RS485 / Non utilisé	--
3	RS485 / TxD+	4 - T/RB
4	RS485 / RTS	6 - RTS
5	RS485 / Terre logique	--
6	RS485 / Alim. 5 V	--
7	RS485 / Non utilisé	--
8	RS485 / TxD-	3 - T/RA
9	RS485 / Non utilisé	--
Boîtier		1 - M

A.16.4.2 Caractéristiques du CM 1241 RS232

Tableau A- 249 Caractéristiques générales

Caractéristiques techniques	CM 1241 RS232
Référence	6ES7241-1AH32-0XB0
Dimensions (mm)	30 x 100 x 75
Poids	150 grammes

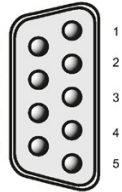
Tableau A- 250 Emetteur et récepteur

Caractéristiques techniques	CM 1241 RS232
Type	RS232 (duplex intégral)
Tension de sortie émetteur	+/- 5 V min. pour $R_L = 3 \text{ k}\Omega$
Tension de sortie émetteur	+/- 15 V CC max.
Impédance d'entrée de récepteur	3 k Ω min.
Seuil/sensibilité récepteur	0,8 V min. bas, 2,4 V max. haut, 0,5 V hystérésis typique
Tension d'entrée récepteur	+/- 30 V CC max.
Isolation signal RS232 à la terre du châssis signal RS232 au commun logique CPU	707 V CC (essai de type)
Longueur de câble, blindé	10 m max.
Débit	300 bauds, 600 bauds, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (par défaut), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Parité	Sans parité (par défaut), parité paire, parité impaire, parité marque (bit de parité toujours à 1), parité espace (bit de parité toujours à 0)
Nombre de bits d'arrêt	1 (par défaut), 2
Contrôle du flux	Matériel, logiciel
Temps d'attente	0 à 65 535 ms

Tableau A- 251 Alimentation

Caractéristiques techniques	CM 1241 RS232
Perte de puissance (dissipation)	1 W
De + 5 V CC	200 mA

Tableau A- 252 Connecteur RS232 (mâle)

Broche	Description	Connecteur (mâle)	Broche	Description
1 DCD	Détection de porteuse : Entrée		6 DSR	Modem prêt : Entrée
2 RxD	Données reçues de DCE : Entrée		7 RTS	Demande pour émettre : Sortie
3 TxD	Données émises vers DCE : Sortie		8 CTS	Prêt à émettre : Entrée
4 DTR	Terminal de données prêt : Sortie		9 RI	Indicateur d'appel (inutilisé)
5 GND	Terre logique		Boîtier	Terre du châssis

A.16.4.3 Caractéristiques du module CM 1241 RS422/485

Caractéristiques du module CM 1241 RS422/485

Tableau A- 253 Caractéristiques générales

Caractéristiques techniques	CM 1241 RS422/485
Référence	6ES7241-1CH32-0XB0
Dimensions L x H x P (mm)	30 x 100 x 75
Poids	155 grammes

Tableau A- 254 Emetteur et récepteur

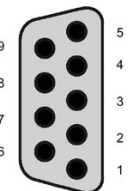
Caractéristiques techniques	CM 1241 RS422/485
Type	RS422 ou RS485, connecteur femelle subD 9 broches
Plage de tension en mode commun	-7 V à + 12 V, 1 seconde, 3 V eff. continu
Tension de sortie différentielle émetteur	2 V min. pour $R_L = 100 \Omega$ 1,5 V min. pour $R_L = 54 \Omega$
Terminaison et polarisation	10 k Ω vers + 5 V sur B, PROFIBUS broche 3 10 k Ω vers GND sur A, PROFIBUS broche 8 Options de polarisation interne fournies ou pas de polarisation interne. Une terminaison externe est nécessaire dans tous les cas, voir Polarisation et terminaison d'un connecteur réseau RS485 (Page 1130) et Configuration de RS422 et RS485 dans le Manuel système de l'automate programmable S7-1200 (Page 1192).
Impédance d'entrée de récepteur	5,4 k Ω min., terminaison incluse
Seuil/sensibilité récepteur	+/- 0,2 V min., hystérésis typique 60 mV
Isolation signal RS485 à la terre du châssis signal RS485 au commun logique CPU	707 V CC (essai de type)
Longueur de câble, blindé	1000 m max. (en fonction du débit)
Débit en bauds	300 bauds, 600 bauds, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (par défaut), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Parité	Sans parité (par défaut), parité paire, parité impaire, parité marque (bit de parité toujours à 1), parité espace (bit de parité toujours à 0)
Nombre de bits d'arrêt	1 (par défaut), 2
Contrôle du flux	XON/XOFF pris en charge pour le mode RS422
Temps d'attente	0 à 65 535 ms

Tableau A- 255 Alimentation

Caractéristiques techniques	CM 1241 RS422/485
Perte de puissance (dissipation)	1,1 W
De + 5 V CC	220 mA

A.17 TeleService (TS Adapter et TS Adapter modulaire)

Tableau A- 256 Connecteur RS485 ou RS422 (femelle)

Broche	Description	Connecteur (femelle)	Broche	Description
1	Terre logique ou communication		6 PWR	+ 5 V avec résistance série 100 ohms : sortie
2 TxD+ ¹	Connecté pour RS422 Non utilisé pour RS485 : sortie		7	Non connecté
3 TxD+ ²	Signal B (RxD/TxD+) : entrée/sortie		8 TXD- ²	Signal A (RxD/TxD-) : entrée/sortie
4 RTS ³	Demande pour émettre (niveau TTL) : sortie		9 TXD- ¹	Connecté pour RS422 Non utilisé pour RS485 : sortie
5 GND	Terre logique ou communication		Boîtier	Terre du châssis

- ¹ Les broches 2 (TxD+) et 9 (TxD-) sont utilisées uniquement comme signaux d'émission RS422.
- ² Les broches 3 (RxD/Tx+) et 8 (RxD/TxD-) sont utilisées comme signaux d'émission et de réception RS485. Pour RS422, la broche 3 est RxD+ et la broche 8 est RxD-.
- ³ RTS est un signal de niveau TTL et peut être utilisé pour piloter un autre appareil semi-duplex en fonction de ce signal. Il est actif lorsque vous émettez et inactif le reste du temps.

A.17 TeleService (TS Adapter et TS Adapter modulaire)

Les manuels suivants contiennent les caractéristiques techniques des adaptateurs TS Adapter IE Basic et TS Adapter modulaire :

- Industrial Software Engineering Tools
TS Adapter modulaire
- Industrial Software Engineering Tools
TS Adapter IE Basic

Pour plus d'informations sur ce produit et pour obtenir la documentation du produit, reportez-vous au site Web du catalogue produit pour l'adaptateur TS Adapter (<https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=TS%20Adapteur%20IE%20basic&tab=>).

A.18 Cartes mémoire SIMATIC

Capacité	Référence
32 Go	6ES7954-8LT02-0AA0
2 Go	6ES7954-8LP01-0AA0
256 Mo	6ES7954-8LL02-0AA0
24 Mo	6ES7954-8LF02-0AA0
12 Mo	6ES7954-8LE02-0AA0
4 Mo	6ES7954-8LC02-0AA0

A.19 Simulateurs d'entrées

Tableau A- 257 Caractéristiques générales

Caractéristiques techniques	Simulateur 8 positions	Simulateur 14 positions	Simulateur de la CPU 1217C
Référence	6ES7274-1XF30-0XA0	6ES7274-1XH30-0XA0	6ES7274-1XK30-0XA0
Dimensions L x H x P (mm)	43 x 35 x 23	67 x 35 x 23	93 x 40 x 23
Poids	20 grammes	30 grammes	43 grammes
Entrées	8	14	14
Utilisé avec la CPU	CPU 1211C, CPU 1212C	CPU 1214C, CPU 1215C	CPU 1217C

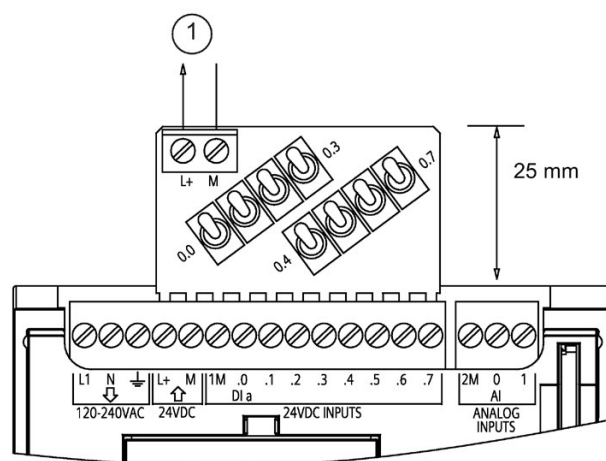
! ATTENTION

Utilisation sûre des simulateurs d'entrées

Ces simulateurs d'entrées ne sont pas homologués pour une utilisation dans des lieux dangereux de classe I DIV 2 ou de classe I zone 2. En effet, les commutateurs présentent un risque potentiel d'étincelle/d'explosion en cas d'utilisation dans un site de classe I DIV 2 ou de classe I zone 2. Une utilisation non approuvée pouvant entraîner la mort, des blessures graves et des dommages matériels importants,

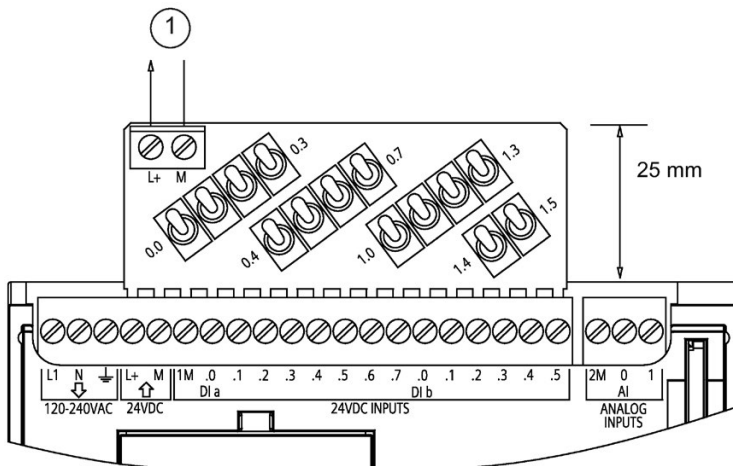
utilisez ces simulateurs d'entrées uniquement à des emplacements non dangereux. Vous ne devez donc pas les utiliser dans des lieux dangereux de classe I DIV 2 ou de classe I zone 2.

Simulateur 8 positions (6ES7274-1XF30-0XA0)



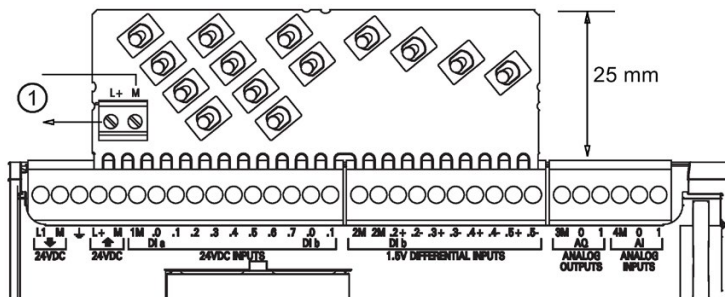
① Sortie alim. cap-
teur 24 V CC

Simulateur 14 positions (6ES7274-1XF30-0XA0)



① Sortie alim. capteur
24 V CC

Simulateur CPU 1217C (6ES7274-1XK30-0XA0)



① Sortie alim. capteur
24 V CC

A.20 Module de potentiomètre S7-1200

Le module de potentiomètre S7-1200 est un accessoire pour la CPU S7-1200. Chaque potentiomètre crée une tension de sortie proportionnelle à la position du potentiomètre pour piloter chacune des deux entrées analogiques 0 V CC à 10 V CC de la CPU. Pour installer le potentiomètre :

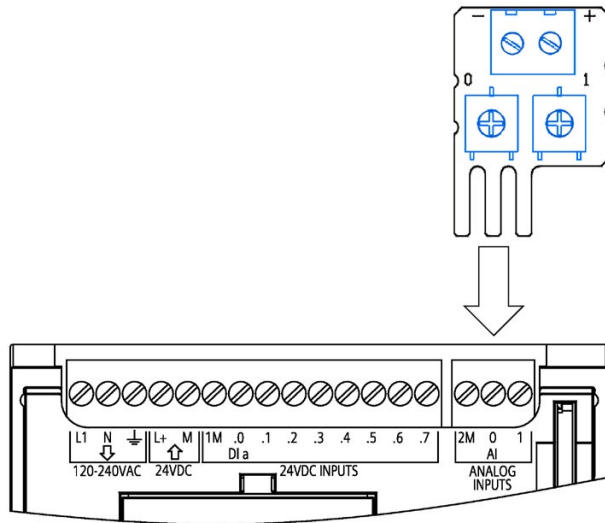
1. Insérez les 'doigts' de la carte du circuit imprimé dans un bornier d'entrée analogique de la CPU S7-1200, puis connecter une alimentation en courant continu externe au connecteur 2 positions du module de potentiomètre.
2. Réglez les potentiomètres à l'aide d'un petit tournevis. Tournez le potentiomètre dans le sens des aiguilles d'une montre (vers la droite) pour augmenter la tension de sortie et dans le sens inverse des aiguilles d'une montre (vers la gauche) pour diminuer la tension de sortie.

Remarque

Suivez les directives pour composants CSDE lors de la manipulation du module de potentiomètre S7-1200.

Caractéristiques techniques	Module potentiomètre S7-1200
Référence	6ES7274-1XA30-0XA0
Utilisé avec la CPU	Toutes les CPU S7-1200
Nombre de potentiomètres	2
Dimensions L x H x P (mm)	20 x 33 x 14
Poids	26 grammes
Entrée de la tension externe sur le connecteur 2 positions ¹ (classe 2, puissance limitée, ou alimentation de capteur de l'API)	16,4 V CC à 28,8 V CC
Longueur de câble (mètres)/type	< 30 m, paire torsadée blindée
Consommation de courant d'entrée	10 mA max.
Tension de sortie potentiomètre vers entrées analogiques de CPU S7-1200 ¹	0 V CC à 10,5 V CC min.
Isolation	Non isolée
Plage de température ambiante	- 20 °C à 60 °C

¹ La stabilité de la tension de sortie du potentiomètre dépend de la qualité de la tension externe fournie sur le connecteur 2 positions - à considérer comme une entrée analogique de tension.

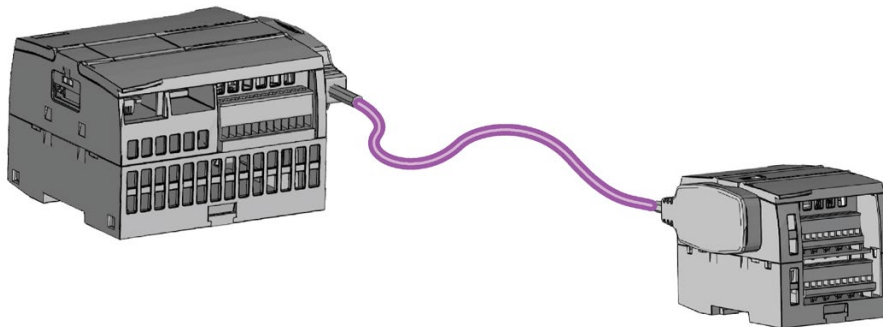


A.21 Câble d'extension d'E/S

Tableau A- 258 Câbles d'extension

Caractéristiques techniques	
Référence	6ES7290-6AA30-0XA0
Longueur de câble	2 m
Poids	200 g

Vous trouverez dans la partie sur l'installation (Page 70) des informations sur l'installation et la désinstallation du câble d'extension S7-1200.



A.22 Produits connexes

A.22.1 Module d'alimentation PM 1207

Le PM 1207 est un module d'alimentation pour le SIMATIC S7-1200. Il présente les fonctions suivantes :

- Entrée 120/230 V CA, sortie 24 V CC/2,5 A

Pour plus d'informations sur ce produit et pour obtenir la documentation du produit, recherchez PM 1207 dans le catalogue de produits en ligne

(<https://mall.industry.siemens.com/mall/en/de/Catalog/Product/6EP1332-1SH71>).

A.22.2 Module commutateur compact CSM 1277

Le CSM1277 est un module commutateur compact Industrial Ethernet. Il peut être utilisé pour multiplier l'interface Ethernet du S7-1200 pour permettre des communications simultanées avec les pupitres opérateurs, les consoles de programmation et autres contrôleurs. Il présente les caractéristiques suivantes :

- 4 prises RJ45 pour connexion à l'Industrial Ethernet
- Bornier enfichable 3 points pour le raccordement de l'alimentation externe 24 V CC à la partie supérieure
- Des DEL pour l'affichage de diagnostic et d'état des ports Industrial Ethernet
- Numéro d'article 6GK7277-1AA00-0AA0

Pour plus d'informations sur ce produit et pour obtenir la documentation du produit, recherchez CSM 1277 dans le catalogue de produits en ligne

(<https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=csm%201277&tab=>).

A.22.3 Module CM CANopen

Le module CM CANopen est un module enfichable entre l'API SIMATIC S7-1200 et tout appareil compatible CANopen. Le CM CANopen peut être configuré pour être aussi bien maître qu'esclave. Il y a deux CM CANopen modules : le module CANopen (référence 021620-B) et le module CANopen (Ruggedized) (référence 021730-B).

Le module CANopen présente les caractéristiques suivantes :

- Capable de connecter 3 modules par CPU
- Connecte jusqu'à 16 abonnés esclaves CANopen
- 256 octets d'entrée et 256 octets de sortie par module
- 3 LED fournissent des informations de diagnostic sur le module, le réseau et l'état d'E/S
- Prend en charge le stockage de la configuration de réseau CANopen dans l'API
- Le module peut être intégré dans le catalogue du matériel de la suite de configuration TIA Portal
- Configuration CANopen avec CANopen Configuration Studio (inclus) ou avec tout autre outil de configuration CANopen externe
- Est conforme aux profils de communication CANopen CiA 301 rev. 4.2 et CiA 302 rev. 4.1
- Prend en charge CAN 2.0A pour la gestion de protocoles personnalisés
- Blocs fonctionnels préprogrammés disponibles pour la programmation chaque API dans TIA Portal
- Les modules CM CANopen comprennent des connecteurs Sub-D avec bornes à vis pour sous-réseau, un CD CM CANopen Configuration Studio et un câble de configuration USB

Pour plus d'informations sur ce produit et pour obtenir la documentation du produit, recherchez CM CANopen dans le catalogue de produits en ligne.

A.22.4 Module de communication RF120C

Le RF10C permet aux systèmes RFID et de lecture de code de Siemens d'être directement et facilement connectés à un S7-1200. Le lecteur est connecté au RF120C via une liaison point-à-point. Un maximum de trois modules de communication peuvent être connectés à un S7-1200 sur la gauche de la CPU. Le module de communication RF120C est configuré via le TIA Portal. La référence du module de communication RF120C est 6GT2002-0LA00.

Pour plus d'informations sur ce produit et pour obtenir la documentation du produit, recherchez RF120C dans le catalogue de produits en ligne.

A.22.5 Module SM 1238 Energy Meter

Le compteur d'énergie SM 1238 Energy Meter 480 V AC est conçu pour une mise en œuvre au niveau machine dans un système S7-1200. Il enregistre plus de 200 valeurs d'énergie et de mesure électriques différentes, permettant ainsi une gestion transparente des besoins énergétiques des composants individuels d'une installation de production, et ce jusqu'au niveau des machines. Grâce aux mesures fournies par le module SM 1238 Energy Meter, vous pouvez mesurer la consommation et les besoins en énergie.

Pour plus d'informations sur ce produit et pour en obtenir la documentation et les caractéristiques techniques, voir Module SM 1238 Energy Meter (<https://support.industry.siemens.com/cs/ww/en/view/109483435>) sur le site Web du catalogue de produits.

A.22.6 Électronique de pesage SIWAREX

SIWAREX WP231, WP241 et WP251

Les électroniques de pesage SIWAREX WP231, WP241 et WP251 peuvent être utilisées dans le S7-1200. Ces modules utilisent toutes les fonctions d'un système moderne d'automatisation, telles que la communication intégrée, le contrôle-commande, le diagnostic et les outils de configuration dans TIA Portal.

- SIWAREX WP231 (<https://support.industry.siemens.com/cs/ww/fr/view/90229056>), électronique de pesage calibrateur (1 voie) pour pesons à jauges extensiométriques/ponts intégraux (1-4 mV/V) pour SIMATIC S7-1200, interface RS485 et Ethernet, E/S intégrées : 4 DI/4 DQ, 1 AQ (0/4...20 mA)
- SIWAREX WP241 (<https://support.industry.siemens.com/cs/ww/fr/view/90229063>), électronique de pesage à bande (1 voie) pour pesons à jauges extensiométriques/ponts intégraux (1-4 mV/V) pour SIMATIC S7-1200, interface RS485 et Ethernet, E/S intégrées : 4 DI/4 DQ, 1 AQ (0/4...20 mA)
- SIWAREX WP251, électronique de pesage pour processus de dosage et de conditionnement (1 voie) pour pesons à jauges extensiométriques / ponts intégraux (1-4 mV/V) pour SIMATIC S7-1200, interface RS485 et Ethernet, E/S intégrées : 4 DI/4 DQ, 1 AQ (0/4...20 mA)

Voir aussi

SIWAREX WP251 (<https://support.industry.siemens.com/cs/ww/fr/view/109481751/en>)

Calcul d'un bilan de consommation

La CPU possède une alimentation interne fournissant du courant à la CPU elle-même, aux modules d'extension, ainsi qu'à d'autres équipements consommant du courant 24 V CC.

Il existe quatre types de modules d'extension :

- Les modules d'entrées-sorties (SM) sont installés sur le côté droit de la CPU. Chaque CPU autorise un nombre maximum de modules d'entrées-sorties sans prise en compte du bilan de consommation.
 - Les CPU 1214C, 1215C et 1217C autorisent 8 modules d'entrées-sorties.
 - La CPU 1212C autorise 2 modules d'entrées-sorties.
 - La CPU 1211C n'autorise pas de module d'entrées-sorties.
- Les modules de communication (CM) sont installés sur le côté gauche de la CPU. Trois modules de communication au maximum sont autorisés quelle que soit la CPU sans prise en compte du bilan de consommation.
- Les Signal Boards (SB), Communication Boards (CB) et Battery Boards (BB) sont installés sur le dessus de la CPU. Un Signal Board, un Communication Board ou un Battery Board au maximum est autorisé quelle que soit la CPU.

Les informations ci-après doivent vous aider à déterminer combien d'énergie ou de courant la CPU peut mettre à la disposition de votre configuration.

Chaque CPU fournit du courant 5 V CC et 24 V CC :

- La CPU fournit du courant 5 V CC pour les modules d'extension lorsqu'un tel module est connecté. Si les besoins en courant 5 V CC des modules d'extension dépassent le courant fourni par la CPU, vous devez supprimer des modules d'extension jusqu'à ce que leurs besoins soient couverts.
- Chaque CPU a une alimentation de capteur 24 V CC pouvant fournir du courant 24 V CC aux entrées locales ou aux bobines de relais sur les modules d'extension. Si les besoins en courant 24 V CC dépassent le courant fourni par la CPU, vous pouvez ajouter une alimentation 24 V CC externe afin de fournir ce courant aux modules d'extension. Vous devez connecter l'alimentation 24 V CC aux entrées ou aux bobines de relais manuellement.

 **ATTENTION**

Connecter une alimentation 24 V CC externe en parallèle avec l'alimentation de capteur 24 V CC peut entraîner un conflit entre les deux alimentations, chacune cherchant à établir son propre niveau de tension de sortie préféré.

Ce conflit peut réduire la durée de vie ou provoquer une défaillance immédiate de l'une ou des deux alimentations, ayant pour effet un fonctionnement imprévisible du système d'automatisation pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

L'alimentation de capteur CC sur la CPU et toute alimentation externe doivent fournir du courant à des points différents. Une seule connexion des conducteurs communs est autorisée.

Certains ports d'entrée d'alimentation 24 V CC dans le système d'automatisation sont interconnectés, avec un circuit commun logique connectant plusieurs bornes M. L'entrée d'alimentation 24 V CC de la CPU, l'entrée d'alimentation de bobine de relais des SM et une entrée d'alimentation analogique non isolée signalées comme non isolées dans les fiches techniques sont des exemples de circuits interconnectés. Toutes les bornes M non isolées doivent être connectées au même potentiel de référence externe.

 **ATTENTION**

Connecter des bornes M non isolées à des potentiels de référence différents provoque des flux de courant indésirables qui peuvent être à l'origine de dégâts ou d'un fonctionnement imprévisible dans l'automate et les équipements connectés

pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Vérifiez toujours que toutes les bornes M non isolées dans un système d'automatisation sont connectées au même potentiel de référence.

Les caractéristiques techniques (Page 1429) fournissent des informations sur les courants fournis par les CPU et sur les besoins en courant des modules d'entrées-sorties.

Remarque

Si le bilan de consommation CPU est déficitaire, vous ne pourrez peut-être pas connecter le nombre maximal de modules autorisés pour votre CPU.

Exemple de bilan de consommation

L'exemple suivant montre un exemple de calcul du bilan de consommation pour une configuration comprenant une CPU 1214C AC/DC/Relais, un SB 1223 2 entrées 24 V DC/2 sorties 24 V DC, un CM 1241, trois SM 1223 8 entrées DC/8 sorties relais et un SM 1221 8 entrées DC. Cette configuration comporte 48 entrées et 36 sorties au total.

Remarque

La CPU a déjà alloué le courant nécessaire pour piloter les bobines de relais internes. Vous n'avez donc pas besoin d'inclure les besoins en courant des bobines de relais internes dans votre bilan de consommation.

Dans cet exemple, la CPU fournit suffisamment de courant continu 5 V CC pour les SM, mais pas suffisamment de courant continu 24 V CC à partir de l'alimentation de capteur pour toutes les entrées et bobines de relais d'extension. Les E/S requièrent 456 mA alors que la CPU ne fournit que 400 mA. Il faut donc une source supplémentaire d'au moins 56 mA à 24 V CC pour opérer toutes les entrées et sorties 24 V CC présentes.

Tableau B- 1 Exemple de bilan de consommation

Courant fourni par la CPU	5 V CC	24 V CC
CPU 1214C AC/DC/Relais	1600 mA	400 mA
<i>moins</i>		
Besoins du système	5 V CC	24 V CC
CPU 1214C, 14 entrées	-	14 * 4 mA = 56 mA
1 SB 1223 2 entrées 24 V CC/2 sorties 24 V CC	50 mA	2 * 4 mA = 8 mA
1 CM 1241 RS422/485, courant 5 V	220 mA	
3 SM 1223, courant 5 V	3 * 145 mA = 435 mA	-
1 SM 1221, courant 5 V	1 * 105 mA = 105 mA	-
3 SM 1223, 8 entrées chacun	-	3 * 8 * 4 mA = 96 mA
3 SM 1223, 8 bobines de relais chacun	-	3 * 8 * 11 mA = 264 mA
1 SM 1221, 8 entrées	-	8 * 4 mA = 32 mA
Total des besoins	810 mA	456 mA
<i>égale</i>		
Bilan de consommation	5 V CC	24 V CC
Excédent/déficit de courant	790 mA	(56 mA)

Formulaire pour calculer votre bilan de consommation

Servez-vous du tableau suivant pour déterminer combien de courant la CPU S7-1200 peut mettre à la disposition de votre configuration. Vous trouverez dans les caractéristiques techniques (Page 1429) des informations sur le courant fourni par votre modèle de CPU et sur les besoins en courant de vos modules d'entrées-sorties.

Tableau B- 2 Calcul d'un bilan de consommation

Courant fourni par la CPU	5 V CC	24 V CC
<i>moins</i>		
Besoins du système	5 V CC	24 V CC
Total des besoins		
<i>égale</i>		
Bilan de consommation	5 V CC	24 V CC
Excédent/déficit de courant		

Numéro de référence

C.1 Modules CPU

Tableau C- 1 CPU S7-1200

Modèles de CPU		Référence
CPU 1211C	CPU 1211C DC/DC/DC	6ES7211-1AE40-0XB0
	CPU 1211C AC/DC/Relais	6ES7211-1BE40-0XB0
	CPU 1211C DC/DC/Relais	6ES7211-1HE40-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7212-1AE40-0XB0
	CPU 1212C AC/DC/Relais	6ES7212-1BE40-0XB0
	CPU 1212C DC/DC/Relais	6ES7212-1HE40-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7214-1AG40-0XB0
	CPU 1214C AC/DC/Relais	6ES7214-1BG40-0XB0
	CPU 1214C DC/DC/Relais	6ES7214-1HG40-0XB0
CPU 1215C	CPU 1215C DC/DC/DC	6ES7215-1AG40-0XB0
	CPU 1215C AC/DC/Relais	6ES7215-1BG40-0XB0
	CPU 1215C DC/DC/Relais	6ES7215-1HG40-0XB0
CPU 1217C	CPU 1217C DC/DC/DC	6ES7217-1AG40-0XB0

C.2 Module d'entrées-sorties (SM), Signal Boards (SB) et Battery boards (BB)

Tableau C- 2 Modules d'entrées-sorties (SM)

Modules d'entrées-sorties		Référence
Entrée TOR	SM 1221 8 entrées 24 V CC (sink/source)	6ES7221-1BF32-0XB0
	SM 1221 16 entrées 24 V CC (sink/source)	6ES7221-1BH32-0XB0
Sorties TOR	SM 1222 8 sorties 24 V CC (source)	6ES7222-1BF32-0XB0
	SM 1222 16 sorties 24 V CC (source)	6ES7222-1BH32-0XB0
	SM 1222 8 sorties relais	6ES7222-1HF32-0XB0
	SM 1222 8 sorties relais (inverseur)	6ES7222-1XF32-0XB0
	SM 1222 16 sorties relais	6ES7222-1HH32-0XB0
Entrées/sorties TOR	SM 1223 8 entrées 24 V CC (sink/source)/8 sorties 24 V CC (source)	6ES7223-1BH32-0XB0
	SM 1223 16 entrées 24 V CC (sink/source)/16 sorties 24 V CC (source)	6ES7223-1BL32-0XB0
	SM 1223 8 entrées 24 V CC (sink/source)/8 sorties relais	6ES7223-1PH32-0XB0
	SM 1223 16 entrées 24 V CC (sink/source)/16 sorties relais	6ES7223-1PL32-0XB0
	SM 1223 8 entrées 120/230 V CC (sink/source)/8 sorties relais	6ES7223-1QH32-0XB0
Entrées analogiques	SM 1231 4 entrées analogiques	6ES7231-4HD32-0XB0
	SM 1231 8 entrées analogiques	6ES7231-4HF32-0XB0
	SM 1231 4 entrées analogiques x 16 bits (High Feature)	6ES7231-5ND32-0XB0
	SM 1238 Energy Meter 480 V CA	6ES7238-5XA32-0XB0
Sorties analogiques	SM 1232 2 sorties analogiques	6ES7232-4HB32-0XB0
	SM 1232 4 sorties analogiques	6ES7232-4HD32-0XB0
Entrées/sorties analogiques	SM 1234 4 entrées analogiques / 2 sorties analogiques	6ES7234-4HE32-0XB0
RTD et Thermo-couple	SM 1231 TC 4 x 16 bits	6ES7231-5QD32-0XB0
	SM 1231 TC 8 x 16 bits	6ES7231-5QF32-0XB0
	SM 1231 RTD 4 x 16 bits	6ES7231-5PD32-0XB0
	SM 1231 RTD 8 x 16 bits	6ES7231-5PF32-0XB0
Modules technologiques	SM 1278 4 x IO-Link maître	6ES7278-4BD32-0XB0
	SIWAREX WP231, électronique de pesage calibreteur (1 voie) pour pesons à jauges extensiométriques/ponts intégraux (1-4 mV/V) pour SIMATIC S7-1200, interface RS485 et Ethernet, E/S intégrées : 4 DI/4 DQ, 1 AQ (0/4...20 mA)	7MH4960-2AA01
	SIWAREX WP241, électronique de pesage à bande (1 voie) pour pesons à jauges extensiométriques/ponts intégraux (1-4 mV/V) pour SIMATIC S7-1200, interface RS485 et Ethernet, E/S intégrées : 4 DI/4 DQ, 1 AQ (0/4...20mA)	7MH4960-4AA01
	SIWAREX WP251, électronique de pesage pour processus de dosage et de conditionnement (1 voie) pour pesons à jauges extensiométriques/ponts intégraux (1-4 mV/V) pour SIMATIC S7-1200, interface RS485 et Ethernet, E/S intégrées : 4 DI/4 DQ, 1 AQ (0/4...20 mA)	7MH4960-6AA01

Tableau C- 3 Signal Boards (SB) et Battery Boards (BB)

Signal Boards et Battery Boards		Référence
Entrée TOR	SB 1221 200 kHz 4 entrées 24 V CC (source)	6ES7221-3BD30-0XB0
	SB 1221 200 kHz 4 entrées 5 V CC (source)	6ES7221-3AD30-0XB0
Sorties TOR	SB 1222 200 kHz 4 sorties 24 V CC (sink/source)	6ES7222-1BD30-0XB0
	SB 1222 200 kHz 4 sorties 5 V CC (sink/source)	6ES7222-1AD30-0XB0
Entrées/sorties TOR	SB 1223 2 entrées 24 V CC (sink)/2 sorties 24 V CC (source)	6ES7223-0BD30-0XB0
	SB 1223 200 kHz 2 entrées 24 V CC (source)/2 sorties 24 V CC (sink/source)	6ES7223-3BD30-0XB0
	SB 1223 200 kHz 2 entrées 5 V CC (source)/2 sorties 5 V CC (sink/source)	6ES7223-3AD30-0XB0
Analogiques	SB 1232 1 sortie analogique	6ES7232-4HA30-0XB0
	SB 1231 1 entrée analogique	6ES7231-4HA30-0XB0
	SB 1231 1 entrée analogique Thermocouple	6ES7231-5QA30-0XB0
	SB 1231 1 entrée analogique RTD	6ES7231-5PA30-0XB0
Pile	Battery Board BB 1297 (pile de type CR1025 non comprise)	6ES7297-0AX30-0XA0

C.3 Communication

Tableau C- 4 Module de communication (CM)

Module de communication (CM)			Référence
RS232, RS422 et RS485	CM 1241 RS232	RS232	6ES7241-1AH32-0XB0
	CM 1241 RS422/485	RS422/485	6ES7241-1CH32-0XB0
PROFIBUS	CM 1243-5	Maître PROFIBUS	6GK7243-5DX30-0XE0
	CM 1242-5	Esclave PROFIBUS	6GK7242-5DX30-0XE0
Maître interface AC	CM 1243-2	Maître interface AC	3RK7243-2AA30-0XB0

Tableau C- 5 Communication Board (CB)

Communication Board (CB)			Référence
RS485	CB 1241 RS485	RS485	6ES7241-1CH30-1XB0

Tableau C- 6 Processeur de communication (CP)

CP	Interface	Référence
CP 1242-7 GPRS V2	GPRS	6GK7242-7KX31-0XE0
CP 1243-7 LTE-US	LTE	6GK7243-7KX30-0XE0
CP 1243-7 LTE-EU	LTE	6GK7243-7KX30-0XE0
CP 1243-1	Interface IE	6GK7243-1BX30-0XE0
CP 1243-8 IRC	Interface IE et série	6GK7243-8RX30-0XE0

Tableau C- 7 TeleService

Adaptateur TS Adapter	Référence
Adaptateur TS Adapter IE Basic	6ES7972-0EB00-0XA0
Adaptateur TS Adapter IE Advanced	6ES7972-0EA00-0XA0
Module TS GSM	6GK7972-0MG00-0XA0
Module TS RS232	6ES7792-0MS00-0XA0
Module TS Modem	6ES7972-0MM00-0XA0
Module TS RNIS	6ES7972-0MD00-0XA0

Tableau C- 8 Accessoires

Accessoire			Référence
Antenne	ANT794-4MR	Antenne GSM/GPRS	6NH9860-1AA00
	ANT794-3M	Antenne en nappe	6NH9870-1AA00

Tableau C- 9 Connecteurs

Type de connecteur		Référence
RS485	Sortie de câble 35 degrés, connexion par bornes à vis	6ES7972-0BA42-0XA0
	Sortie de câble 35 degrés, connexion autodénudante	6ES7972-0BA60-0XA0

C.4 Modules d'entrées-sorties et CPU de sécurité

Tableau C- 10 CPU de sécurité

Modèles de CPU de sécurité		Référence
CPU 1212FC	CPU 1212FC DC/DC/DC	6ES7212-1AF40-0XB0
	CPU 1212FC DC/DC/Relais	6ES7212-1HF40-0XB0
CPU 1214FC	CPU 1214FC DC/DC/DC	6ES7214-1AF40-0XB0
	CPU 1214FC DC/DC/Relais	6ES7214-1HF40-0XB0
CPU 1215FC	CPU 1215FC DC/DC/DC	6ES7215-1AF40-0XB0
	CPU 1215FC DC/DC/Relais	6ES7215-1HF40-0XB0

Tableau C- 11 Modules d'entrées-sorties de sécurité

Modules d'entrées-sorties de sécurité fonctionnelle		Référence
Entrée TOR	SM 1226 F-DI 16 x 24 V DC	6ES7226-6BA32-0XB0
Sortie TOR	SM 1226 F-DQ 4 x 24 V DC	6ES7226-6DA32-0XB0
	SM 1226 F-DQ 2 x Relay	6ES7226-6RA32-0XB0

C.5 Autres modules

Tableau C- 12 Produits connexes

Elément		Référence
Alimentation	Alimentation PM 1207	6EP1332-1SH71
Commutateur Ethernet	Commutateur Ethernet CSM 1277, 4 ports	6GK7277-1AA10-0AA0
CM CANopen	CANopen pour SIMATIC S7-1200	021620-B
	CANopen (Ruggedized) pour SIMATIC S7-1200	021730-B
RF120C	Module de communication RF120C	6GT2002-0LA00

C.6 Cartes mémoire

Tableau C- 13 Cartes mémoire

Cartes mémoire SIMATIC	Référence
SIMATIC MC 32 Go	6ES7954-8LT02-0AA0
SIMATIC MC 2 GB	6ES7954-8LP01-0AA0
SIMATIC MC 256 MB	6ES7954-8LL02-0AA0
SIMATIC MC 24 Mo	6ES7954-8LF02-0AA0
SIMATIC MC 12 Mo	6ES7954-8LE02-0AA0
SIMATIC MC 4 Mo	6ES7954-8LC02-0AA0

C.7 Appareils IHM Basic

Tableau C- 14 Appareils IHM

Pupitres HMI Basic Panels	Référence
KTP400 Basic (monochrome, PN)	6AV2123-2DB03-0AX0
KTP700 Basic	6AV2123-2GB03-0AX0
KTP700 Basic DP	6AV2123-2GA03-0AX0
KTP900 Basic	6AV2123-2JB03-0AX0
KTP1200 Basic	6AV2123-2MB03-0AX0
KTP1200 Basic DP	6AV2123-2MA03-0AX0

C.8 Pièces détachées et autres matériels

Tableau C- 15 Câbles d'extension, simulateurs, et butées

Elément		Référence
Câble d'extension d'E/S	Câble d'extension d'E/S, 2 m	6ES7290-6AA30-0XA0
Simulateur d'E/S	Simulateur (1211C/1212C - 8 positions)	6ES7274-1XF30-0XA0
	Simulateur (1214C/1215C - 14 positions)	6ES7274-1XH30-0XA0
	Simulateur CPU 1217C	6ES7274-1XK30-0XA0
Module potentiomètre	Module potentiomètre S7-1200	6ES7274-1XA30-0XA0
Serre-câble Ethernet	Serre-câble RJ45 à port unique, 10/100 Mbit/s	6ES7290-3AA30-0XA0
	Serre-câble RJ45 à port double, 10/100 Mbit/s	6ES7290-3AB30-0XA0
Kit volets d'accès	CPU 1211C/1212C	6ES7291-1AA30-0XA0
	CPU 1214C	6ES7291-1AB30-0XA0
	CPU 1215C	6ES7291-1AC30-0XA0
	CPU 1217C	6ES7291-1AD30-0XA0
	Module d'entrées-sorties, 45 mm	6ES7291-1BA30-0XA0
	Module d'entrées-sorties, 70 mm	6ES7291-1BB30-0XA0
	Module de communication (à utiliser avec les modules 6ES72xx-xxx32-0XB0 et 6ES72xx-xxx30-0XB0)	6ES7291-1CC30-0XA0
Butée	Butée en thermoplastique, 10 MM	8WA1808
	Butée, acier, 10,3 MM	8WA1805

Remplacement du bornier de connexion

Il est important d'utiliser le bornier correct pour votre module. Consultez les tableaux ci-dessous ainsi que les caractéristiques techniques de vos modules pour déterminer quel bornier peut être utilisé en remplacement.

Remarque

Borniers amovibles (à détrompeur)

Les API nécessitent toujours un câblage correct pour que la sécurité et un fonctionnement correct soient garantis.

Lorsque vous remplacez le bornier dans votre CPU ou votre SM, il est important d'utiliser le bornier et la source de câblage corrects pour votre module.

L'élément de détrompage vous empêche de placer accidentellement un bornier câblé en haute tension dans un module basse tension ou de placer un bornier câblé en tension spéciale dans un module à tension normale. Certains borniers présentent un détrompeur spécifiquement à gauche, à droite ou au milieu.

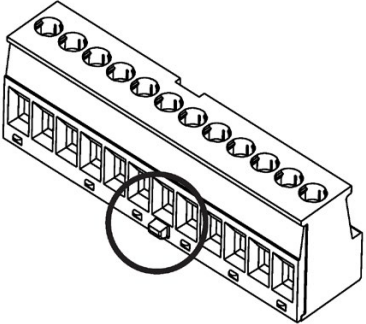
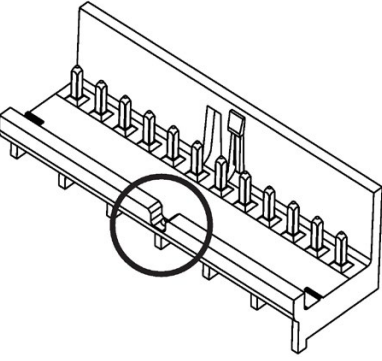
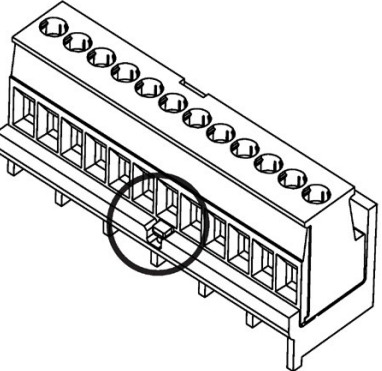
		
<p>Bornier amovible (à détrompeur)</p>	<p>Détrompeur sur l'appareil</p>	<p>Le détrompeur sur l'appareil est uniquement adapté au bornier amovible approprié.</p>

Tableau C- 16 CPU S7-1200 V4.0 et version ultérieure - Kits de pièces de rechange du bornier

Si vous possédez CPU S7-1200 V4.0 et version ultérieure (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AH30-0XA0	8 broches, plaqué à l'étain
	6ES7292-1AP30-0XA0	14 broches, plaqué à l'étain
CPU 1211C DC/DC/Relais (6ES7211-1HE40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AH40-0XA0	8 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AP30-0XA0	14 broches, plaqué à l'étain
CPU 1211C AC/DC/Relais (6ES7211-1BE40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AH40-0XA0	8 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AP40-0XA0	14 broches, plaqué à l'étain, doté d'un détrompeur
CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AH30-0XA0	8 broches, plaqué à l'étain
	6ES7292-1AP30-0XA0	14 broches, plaqué à l'étain
CPU 1212C DC/DC/Relais (6ES7212-1HE40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or

Si vous possédez CPU S7-1200 V4.0 et version ultérieure (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
	6ES7292-1AH40-0XA0	8 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AP30-0XA0	14 broches, plaqué à l'étain
CPU 1212C AC/DC/Relais (6ES7212-1BE40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AH40-0XA0	8 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AP40-0XA0	14 broches, plaqué à l'étain, doté d'un détrompeur
CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AM30-0XA0	12 broches, plaqué à l'étain
	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain
CPU 1214C DC/DC/Relais (6ES7214-1HG40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AM40-0XA0	12 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain
CPU 1214C AC/DC/Relais (6ES7214-1BG40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AM40-0XA0	12 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AV40-0XA0	20 broches, plaqué à l'étain, doté d'un détrompeur
CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)	6ES7292-1BF30-0XB0	6 broches, plaqué-or
	6ES7292-1AM30-0XA0	12 broches, plaqué à l'étain
	6ES7292-1AV30-0XB0	20 broches, plaqué à l'étain
CPU 1215C DC/DC/Relais (6ES7215-1HG40-0XB0)	6ES7292-1BF30-0XB0	6 broches, plaqué-or
	6ES7292-1AM40-0XA0	12 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain
CPU 1215C AC/DC/Relais (6ES7215-1BG40-0XB0)	6ES7292-1BF30-0XB0	6 broches, plaqué-or
	6ES7292-1AM40-0XA0	12 broches, plaqué à l'étain, doté d'un détrompeur

Si vous possédez CPU S7-1200 V4.0 et version ultérieure (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
	6ES7292-1AV40-0XA0	20 broches, plaqué à l'étain, doté d'un détrompeur
CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)	6ES7292-1BF30-0XB0	6 broches, plaqué-or
	6ES7292-1AK30-0XA0	10 broches, plaqué broche
	6ES7292-1AR30-0XA0	16 broches, plaqué broche
	6ES7292-1AT30-0XA0	18 broches, plaqué à l'étain

Tableau C- 17 SM S7-1200 V3.2 et version ultérieure - Kits de pièces de rechange du bornier

Si vous possédez SM S7-1200 V3.2 et version ultérieure (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
SM 1221 DI 8 x DC (6ES7221-1BF32-0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1222 DQ 8 x DC (6ES7222-1BF32-0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1222 DQ 8 x Relais (6ES7222-1HF32-0XB0)	6ES7292-1AG40-0XA1	7 broches, plaqué à l'étain, doté d'un détrompeur à gauche
SM 1238 Energy Meter 480 V AC (6ES7238-5XA32-0XB0) pour l'entrée de courant (sommet)	6ES7292-1AG40-0XA2	7 broches, plaqué à l'étain, doté d'un détrompeur au centre
SM 1238 Energy Meter 480 V AC (6ES7238-5XA32-0XB0) pour l'entrée de courant (fond)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1231 AI 4 x 13 bit (6ES7231-4HD32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1232 AQ 2 x 14 bit (6ES7232-4HB32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1231 AI 4 x TC (6ES7231-5QD32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1231 AI 4 x 16 bit (6ES7231-5ND32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1221 DI 16 x DC (6ES7221-1BH32-0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1222 DQ 16 x DC (6ES7222-1BH32-0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1222 DQ 16 x Relais (6ES7222-1HH32-0XB0)	6ES7292-1AG40-0XA0	7 broches, plaqué à l'étain, doté d'un détrompeur à droite
SM 1223 DI 8 x DC/DQ 8 x DC (6ES7223-1BH32-0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1223 8 x DC/8 x Relais (6ES7223-1PH32-0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain

Si vous possédez SM S7-1200 V3.2 et version ultérieure (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
	6ES7292-1AG40-0XA0	7 broches, plaqué à l'étain, doté d'un détrompeur à droite
SM 1223 8 x AC/8 x Relais (6ES7223-1QH32-0XB0)	6ES7292-1AG40-0XA0	7 broches, plaqué à l'étain, doté d'un détrompeur à droite
SM 1234 AI 4/AQ 2 (6ES7234-4HE32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1231 AI 8 x 13 bit (6ES7231-4HF32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1232 AQ 4 x 14 bit (6ES7232-4HD32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1231 AI 4 x RTD (6ES7231-5PD32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1231 AI 8 x TC (6ES7231-5QF32-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1278 IO LINK (6ES7278-4BD32 0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1222 DQ 8 x Relais (inverseur) (6ES7222-1XF32-0XB0)	6ES7292-1AL30-0XA0	11 broches, plaqué à l'étain
SM 1223 DI 16 x DC/DQ 16 x DC (6ES7223-1BL32-0XB0)	6ES7292-1AL30-0XA0	11 broches, plaqué à l'étain
SM 1223 DI 16 x DC/DQ 16 x Relais (6ES7223-1PL32-0XB0)	6ES7292-1AL30-0XA0	11 broches, plaqué à l'étain
	6ES7292-1AL40-0XA0	11 broches, plaqué à l'étain, doté d'un détrompeur
SM 1231 AI 8 x RTD (6ES7231-5PF32-0XB0)	6ES7292-1BL30-0XA0	11 broches, plaqué-or

Tableau C- 18 SB, CB et BB pour S7-1200 - Kits de pièces de rechange du bornier

Si vous possédez SB, CB ou BB (numéro d'article) pour S7-1200	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
SB 1221 DI 4 x 5 V DC (6ES7221-3AD30-0XB0)	6ES7292-1BF30-0XA0	6 broches
SB 1221 DI 4 x 5 V DC (6ES7221-3AD30-0XB0)		
SB 1221 DI 4 x 24 V DC (6ES7221-3BD30-0XB0)		
SB 1222 DQ 4 x 5 V DC (6ES7222-1AD30-0XB0)		
SB 1222 DQ 4 x 24 V DC (6ES7222-1BD30-0XB0)		
SB 1223 DI 2x24 V DC/DQ 2x24 V DC (6ES7223-0BD30-0XB0)		
SB 1223 DI 2x5 V DC/DQ 2x5 V DC (6ES7223-3AD30-0XB0)		
SB 1223 DI 2x24 V DC/DQ 2x24 V DC (6ES7223-3BD30-0XB0)		
SB 1231 AI 1 x 12 BIT (6ES7231-4HA30-0XB0)		
SB 1231 AI 1 x RTD (6ES7231-5PA30-0XB0)		
SB 1231 AI 1 x TC (6ES7231-5QA30-0XB0)		
SB 1232 AQ 1x12 BIT (6ES7232-4HA30-0XB0)		
CB 1231 RS485 (6ES7241-1CH30-1XB0)		
BB 1297 Battery (6ES7297-0AX30-0XA0)		

Tableau C- 19 CPU de sécurité - Kit de pièces de rechange du bornier

Si vous possédez une CPU de sécurité (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
CPU 1214FC DC/DC/DC (6ES7214-1AF40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AM30-0XA0	12 broches, plaqué à l'étain
	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain
CPU 1214FC DC/DC/Relais (6ES7214-1HF40-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
	6ES7292-1AM40-0XA0	12 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain
CPU 1215FC DC/DC/DC (6ES7215-1AF40 0XB0)	6ES7292-1BF30-0XB0	6 broches, plaqué-or
	6ES7292-1AM30-0XA0	12 broches, plaqué à l'étain
	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain
CPU 1215FC DC/DC/Relais (6ES7215-1HF40 0XB0)	6ES7292-1BF30-0XB0	6 broches, plaqué-or
	6ES7292-1AM40-0XA0	2 broches, plaqué à l'étain, doté d'un détrompeur
	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain

Tableau C- 20 Modules d'entrées-sorties de sécurité - Kit de pièces de rechange du bornier

Si vous possédez un module d'entrées-sorties de sécurité (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
SM 1226 F-DI (6ES7226-6BA32-0XB0)	6ES7292-1AL30-0XA0	11 broches, plaqué à l'étain
SM 1226 F-DQ (6ES7226-6DA32-0XB0)	6ES7292-1AL30-0XA0	11 broches, plaqué à l'étain
SM 1226 F-Relais (6ES7226-6RA32-0XB0)	6ES7292-1AL40-0XA0	11 broches, plaqué à l'étain, doté d'un détrompeur

C.9 Logiciel de programmation

Tableau C- 21 Logiciel de programmation

Logiciel SIMATIC		Référence
Logiciel de programmation	STEP 7 Basic V15	6ES7822-0AA05-0YA5
	STEP 7 Professional V15	6ES7822-1AA05-0YA5
Logiciel de visualisation	WinCC Basic V15	6AV2100-0AA05-0AA5
	WinCC Comfort V15	6AV2101-0AA05-0AA5
	WinCC Advanced V15	6AV2102-0AA05-0AA5
	WinCC Professional 512 PowerTags V15	6AV2103-0DA05-0AA5
	WinCC Professional 4096 PowerTags V15	6AV2103-0HA05-0AA5
	WinCC Professional max. PowerTags V15	6AV2103-0XA05-0AA5

Remplacement de l'appareil et compatibilité des pièces de rechange



D.1 Remplacement d'une CPU V3.0 par une CPU V4.2.x

Pour mettre à niveau une CPU V3.0 à CPU V4.2.x, vous devez remplacer la CPU matérielle. Vous ne pouvez pas mettre à niveau une CPU V3.0 à la version V4.2.x par une mise à jour du firmware.

Vous pourrez alors, dans votre projet STEP 7, remplacer votre CPU V3.0 par une CPU V4.2.x (Page 177) et utiliser le projet STEP 7 existant que vous aviez conçu pour la CPU V3.0.

Lorsque vous remplacez une CPU V3.0 par une CPU V4.2.x, vous voudrez peut-être également vérifier s'il y a des mises à jour du firmware (Page 153) et les appliquer à vos modules d'entrées-sorties et de communication connectés.

Remarque

Remplacement d'un appareil V4.2.x par un appareil V3.0 impossible dans STEP 7

Vous pouvez remplacer une CPU V3.0 par une CPU V4.2.x, mais il n'est pas possible de remplacer une CPU V4.2.x par une CPU V3.0 après y avoir chargé la configuration. Si vous souhaitez visualiser ou utiliser votre projet STEP 7 V3.0 existant, créez une archive de votre projet STEP 7 V3.0 avant de changer l'appareil.

Notez que si vous n'avez pas téléchargé la configuration de l'appareil remplacé, vous pouvez annuler. Cependant, une fois la configuration chargée, vous ne pourrez pas annuler le remplacement de V3.0 par V4.2.x.

Vous devez tenir compte de certains changements dans la configuration et le fonctionnement entre les deux versions de CPU :

Mise à niveau de projets STEP 7

La mise à niveau directe de projets STEP 7 V11 ou V12 à STEP 7 V15 n'est pas possible. Effectuez d'abord la mise à niveau de ces projets à STEP 7 V13 SP1 ou STEP 7 V13 SP2, puis utiliser ces nouveaux projets comme base pour la mise à niveau à STEP 7 V15.

ATTENTION

Risques liés à la copie et à l'insertion de logique de programme provenant de versions antérieures de STEP 7

La copie de logique de programme d'une ancienne version de STEP 7, STEP 7 V12 par exemple, dans STEP 7 V15 peut provoquer un comportement imprévisible lors de l'exécution du programme ou un échec de la compilation. En effet, différentes versions de STEP 7 réalisent les éléments de programme de différentes manières. Le compilateur ne détecte pas toujours les différences si vous avez apporté les modifications par copier-coller d'une version antérieure dans STEP 7 V15. L'exécution d'une logique de programme imprévisible peut entraîner la mort ou des blessures graves si vous ne corrigez pas le programme.

Si vous utilisez une logique de programme d'une version de STEP 7 antérieure à STEP 7 V15, une mise à niveau du projet entier à STEP 7 V15 est requise. Vous pourrez alors copier, couper, coller et éditer la logique de programme selon vos besoins. Dans STEP 7 V15, vous pouvez ouvrir un projet de STEP 7 V13 SP1 ou plus. STEP 7 effectue les conversions de compatibilité nécessaires et met le programme à niveau de manière appropriée. De telles corrections et conversions de mise à niveau sont nécessaires pour assurer une compilation et une exécution correctes du programme. Si votre projet est antérieur à STEP 7 V13 SP1, une mise à niveau progressive à la version STEP 7 V15 est requise.

Blocs d'organisation

Avec la version V4.2.x, vous pouvez configurer l'exécution de l'OB de manière à ce qu'elle soit interruptible ou non interruptible (Page 106). Pour les projet d'anciennes CPU V3.0, STEP 7 définit tous les OB pour qu'ils soient non interruptibles par défaut.

STEP 7 règle toutes les priorités de l'OB (Page 106) sur les mêmes valeurs qu'elles avaient dans le projet STEP 7 CPU V3.0.

Si vous le souhaitez, vous pouvez par la suite modifier les paramètres de l'interruptibilité ou de priorité.

Les informations de déclenchement de l' OB d'alarme de diagnostic (Page 99) se réfèrent à l'ensemble du sous-module si aucun événement de diagnostic n'est en attente.

Protection par mot de passe de la CPU

STEP 7 définit le niveau de protection par mot de passe (Page 218) pour la CPU V4.2.x de sorte qu'il soit équivalent à celui défini pour la CPU V3.0 et affecte le mot de passe V3.0 au mot de passe "Accès complet (pas de protection)" pour la CPU V4.2.x :

Niveau de protection V3.0	Niveau d'accès V4.2.x
Pas de protection	Accès illimité (aucune protection)
Protection en écriture	Accès en lecture
Protection en écriture / lecture	Accès IHM

Notez que le niveau d'accès "Aucun accès (protection complète)" de V4.2.x n'existait pas dans V3.0.

Serveur Web

Si vous utilisez les pages Web personnalisées dans votre projet V3.0, sauvegardez-les dans votre dossier d'installation du projet dans le sous-dossier "UserFiles\Webserver" avant de mettre à niveau votre projet. Si vous sauvegardez vos pages personnalisées à cet emplacement, enregistrer le projet STEP 7 enregistrera également les pages Web personnalisées.

Si vous remplacez une CPU V3.0 par une CPU V4.2.x, votre paramétrage de projet pour l'activation du serveur Web (Page 1040) et votre paramétrage HTTPS seront les mêmes que dans V3.0. Vous pouvez alors configurer les utilisateurs, droits, mots de passe (Page 1042) et les langues (Page 1040) comme vous en avez besoin pour utiliser le serveur Web. Si vous ne conférez pas de droits supplémentaires aux utilisateurs, il y aura des restrictions sur les pages visibles hors pages Web standard (Page 1048). La CPU S7-1200 V4.2.x ne prend pas en charge l'utilisateur et le mot de passe "admin" anciennement préconfigurés.

Auparavant, la page Data log Web server du S7-1200 V3.0 permettait une opération "Télécharger et Effacer". La page Navigateur de fichiers (Page 1075) du serveur Web V4.2.x, à partir de laquelle vous pouvez accéder aux journaux de données, n'offre plus cette fonction. A la place, le serveur Web offre la possibilité de télécharger, renommer et d'effacer les fichiers journaux.

Incompatibilité de la carte de transfert

Vous ne pouvez pas utiliser de carte transfert (Page 143) V3.0 pour transférer un programme V3.0 dans une CPU V4.2.x. Vous devez ouvrir le projet V3.0 dans STEP 7, remplacer l'appareil par une CPU V4.2.x (Page 177) et charger le projet STEP 7 dans votre CPU V4.2.x. Une fois que vous avez changé votre projet en un projet V4.2.x, vous pourrez créer une carte transfert V4.2.x pour les transferts de programme à venir.

Communication GET/PUT

La communication GET/PUT est activée par défaut dans les CPU S7-1200 V3.0. Lorsque vous remplacez votre CPU V3.0 par une CPU V4.2.x (Page 177), un message signalant que GET/PUT est activé apparaît dans la section des informations de compatibilité.

Prise en charge de Motion Control

Les CPU S7-1200 V4.2.x ne prennent pas en charge les bibliothèques Motion Control V1.0 et V2.0. Si vous changez d'appareil pour un projet STEP 7 avec des bibliothèques Motion Control V1.0 ou V2.0, le changement d'appareil entraîne le remplacement des instructions de la bibliothèque Motion Control V1.0 ou V2.0 par les instructions Motion Control (Page 754) V3.0 compatibles à la compilation.

Si vous remplacez une CPU V3.0 par une CPU V4.2.x pour un projet STEP 7 qui contient deux versions différentes d'instructions Motion Control (V3.0 et V5.0), le changement d'appareil entraîne la substitution d'instructions Motion Control (Page 754) compatibles avec V5.0 lors de la compilation.

Lors du remplacement d'une CPU V3.0 par une CPU V4.2.x, la version de l'objet technologique (TO) Motion control ne passe pas automatiquement de la version V3.0 à la version V5.0. Si vous voulez effectuer une mise à niveau aux dernières versions, vous devez aller dans l'arborescence des instructions et sélectionner la version Motion Control S7-1200 requise pour votre projet, comme indiqué dans le tableau ci-dessous :

Version de la CPU	Versions autorisées de Motion Control
V4.2.x (Motion Control V5.0)	V5.0 ou V4.0 ou V3.0
V4.1 (Motion Control V5.0)	V5.0 ou V4.0 ou V3.0
V4.0 (Motion Control V4.0)	V4.0 ou V3.0
V3.0 (Motion Control V3.0)	V3.0

La structure de l'objet technologique est différente entre la version V3.0 et la version V5.0 de Motion Control. Tous les blocs associés sont également associés. Les interfaces de bloc, les tables de visualisation et les traces sont actualisées pour être conformes à la nouvelle structure de Motion Control V5.0. Les différences entre les paramètres d'axe de Motion Control des CPU V3.0 et V4.2.x sont présentées dans les deux tableaux suivants :

CPU V3.0 (Motion Control V3.0)	CPU V4.2.x (Motion Control V5.0)
Config.General.LengthUnit	Units.LengthUnit
Config.Mechanics.PulsesPerDriveRevolution	Actor.DriveParameter.PulsesPerDriveRevolution
Config.Mechanics.LeadScrew	Mechanics.LeadScrew
Config.Mechanics.InverseDirection	Actor.InverseDirection
Config.DynamicLimits.MinVelocity	DynamicLimits.MinVelocity
Config.DynamicLimits.MaxVelocity	DynamicLimits.MaxVelocity
Config.DynamicDefaults.Acceleration	DynamicDefaults.Acceleration
Config.DynamicDefaults.Deceleration	DynamicDefaults.Deceleration
Config.DynamicDefaults.EmergencyDeceleration	DynamicDefaults.EmergencyDeceleration
Config.DynamicDefaults.Jerk	DynamicDefaults.Jerk
Config.PositionLimits_SW.Active	PositionLimitsSW.Active
Config.PositionLimits_SW.MinPosition	PositionLimitsSW.MinPosition
Config.PositionLimits_SW.MaxPosition	PositionLimitsSW.MaxPosition
Config.PositionLimits_HW.Active	PositionLimitsHW.Active
Config.PositionLimits_HW.MinSwitchedLevel	PositionLimitsHW.MinSwitchLevel
Config.PositionLimits_HW.MaxSwitchedLevel	PositionLimitsHW.MaxSwitchLevel

CPU V3.0 (Motion Control V3.0)	CPU V4.2.x (Motion Control V5.0)
Config.Homing.AutoReversal	Homing.AutoReversal
Config.Homing.Direction	Homing.ApproachDirection
Config.Homing.SideActiveHoming	Sensor[1].ActiveHoming.SideInput
Config.Homing.SidePassiveHoming	Sensor[1].PassiveHoming.SideInput
Config.Homing.Offset	Sensor[1].ActiveHoming.HomePositionOffset
Config.Homing.FastVelocity	Homing.ApproachVelocity
Config.Homing.SlowVelocity	Homing.ReferencingVelocity
MotionStatus.Position	Position
MotionStatus.Velocity	Velocity
MotionStatus.Distance	StatusPositioning.Distance
MotionStatus.TargetPosition	StatusPositioning.TargetPosition
StatusBits.SpeedCommand	StatusBits.VelocityCommand
StatusBits.Homing	StatusBits.HomingCommand

Le seul paramètre "commandtable" renommé est le tableau avec les commandes :

V3.0	V4.2.x
Config.Command[]	Command[]

Remarque : Le tableau "Command[]" est un type de données utilisateur de type "TO_CmdTab_Config_Command" dans V3.0 et de type "TO_Struct_Command" dans V4.2.x.

Modifications des instructions

Les instructions suivantes ont subi des modifications de paramètres ou de comportement :

- RDREC et WRREC (Page 395)
- CONV (Page 302)

Communication de pupitres IHM

Si vous aviez un ou plusieurs pupitres IHM (Page 33) connectés à votre CPU S7-1200 V3.0, la communication avec la CPU S7-1200 V4.2.x dépendra du type de communication utilisé et de la version de firmware du pupitre IHM. Recompilez puis téléchargez le projet sur la CPU et l'IHM et/ou mettez à jour le firmware IHM.

Conditions pour recompiler des blocs de programme

Après avoir remplacé une CPU V3.0 par une CPU V4.2.x, vous devez recompiler tous les blocs de programme avant de pouvoir les charger dans la CPU V4.2.x. De plus, si des blocs ont une protection know-how (Page 222) ou une protection contre la copie liée à un numéro de série API (Page 223), vous devez retirer la protection avant de compiler et charger les blocs. (En revanche, il n'est pas nécessaire de désactiver la protection contre la copie liée à une carte mémoire.) Après une compilation réussie, vous pouvez reconfigurer la protection know-how et/ou la protection contre la copie du numéro de série API. Notez que si votre projet inclut des blocs avec une protection know-how fournis par un OEM, vous devez contacter l'OEM pour qu'il fournisse les versions V4.2.x de ces blocs.

En général, Siemens vous recommande de recompiler la configuration matérielle et le logiciel dans STEP 7 et de les charger sur tous les appareils dans votre projet après le changement d'appareil. Corrigez toutes les erreurs rencontrées lors de la compilation du projet, puis recompilez jusqu'à ce qu'il n'y ait plus d'erreur. Vous pourrez ensuite charger le projet dans la CPU V4.2.x.

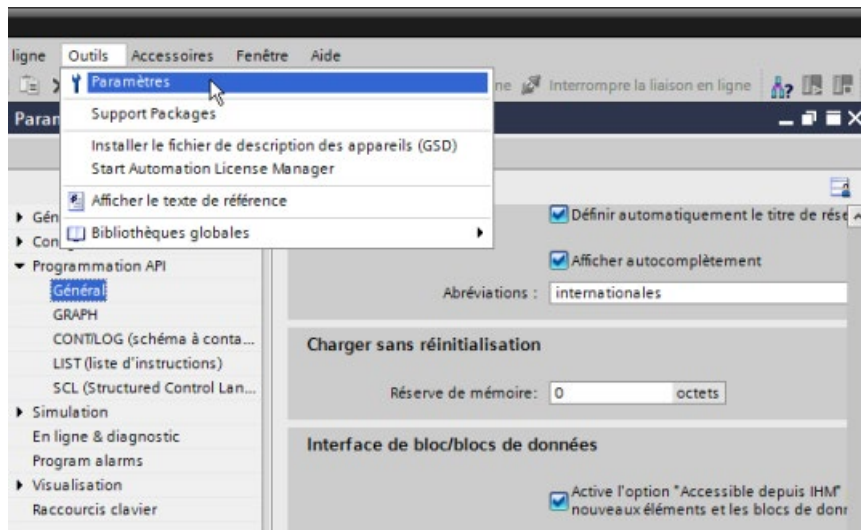
Il se peut que les projets S7-1200 V3.0 ne disposent pas de mémoire suffisante dans les CPU S7-1200 V4.2.x.

Les versions S7-1200 V4.0 et plus ont ajouté un espace de réserve de 100 octets à chaque DB pour prendre en charge le chargement sans réinitialisation.

Vous pouvez supprimer l'espace de réserve de 100 octets des DB avant d'essayer de charger un projet V3.0 dans une CPU V4.2.x.

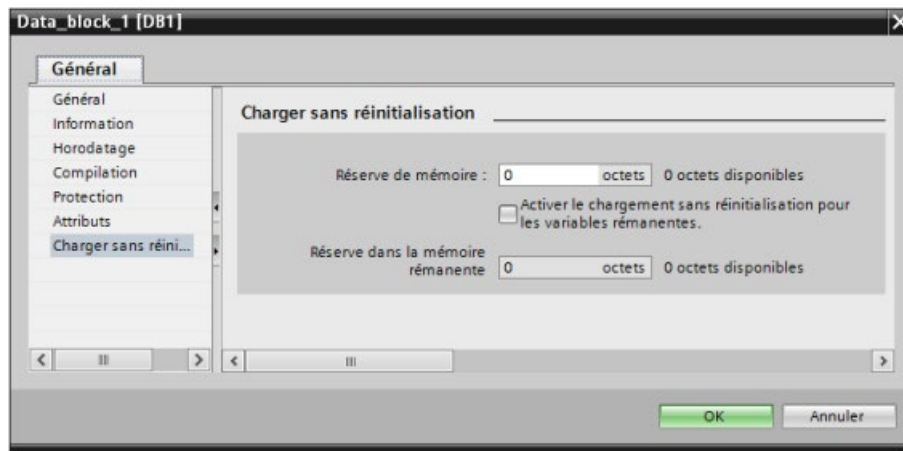
Pour supprimer l'espace de réserve de 100 octets, suivez les étapes suivantes avant de procéder au remplacement d'un appareil :

1. à partir du menu principal du Portail TIA, sélectionnez la commande de menu Outils > Paramètres.
2. Dans l'arbre de navigation, ouvrez le nœud Programmation API > Général.
3. Dans la rubrique "Charger sans réinitialisation", réglez la réserve de mémoire sur 0 octets.



Si vous avez déjà effectué le remplacement de l'appareil, vous devez supprimer la réserve de 100 octets dans chaque bloc individuellement :

1. à partir de l'arbre du projet, effectuez un clic droit avec la souris sur un bloc de données à partir du dossier Blocs de programme et sélectionnez Propriétés à partir du menu contextuel.
2. Dans la boîte de dialogue des propriétés du bloc de données, sélectionnez le nœud "Charger sans réinitialisation".
3. Réglez la réserve de mémoire sur 0 octets.
4. Répétez ces étapes pour chaque bloc de données de votre projet.



Remarque

Les projets pour les CPU V4.0 et V4.1 peuvent s'exécuter sans modification dans les CPU V4.2.x.

D.2 Kits de pièces de rechange du bornier pour S7-1200 V3.0 et antérieure

Tableau D- 1 CPU S7-1200 V3.0 et version antérieure - Kits de pièces de rechange du bornier

Si vous possédez CPU S7-1200 V3.0 et version antérieure (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
CPU 1211C DC/DC/DC (6ES7211-1AE31-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
CPU 1211C AC/DC/Relais (6ES7211-1BE31-0XB0)	6ES7292-1AH30-0XA0	8 broches, plaqué-or
CPU 1211C DC/DC/Relais (6ES7211-1HE31-0XB0)	6ES7292-1AP30-0XA0	14 broches, plaqué à l'étain
CPU 1212C DC/DC/DC (6ES7212-1AE31-0XB0)		
CPU 1212C AC/DC/Relais (6ES7212-1BE31-0XB0)		
CPU 1212C DC/DC/Relais (6ES7212-1HE31-0XB0)		
CPU 1214C DC/DC/DC (6ES7214-1AG31-0XB0)	6ES7292-1BC30-0XA0	3 broches, plaqué-or
CPU 1214C AC/DC/Relais (6ES7214-1BG31-0XB0)	6ES7292-1AM30-0XA0	12 broches, plaqué à l'étain
CPU 1214C DC/DC/Relais (6ES7214-1HG31-0XB0)	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain
CPU 1215C DC/DC/DC (6ES7215-1AG31-0XB0)	6ES7292-1BF30-0XB0	6 broches, plaqué-or
CPU 1215C AC/DC/Relais (6ES7215-1BG31-0XB0)	6ES7292-1AM30-0XA0	12 broches, plaqué à l'étain
CPU 1215C DC/DC/Relais (6ES7215-1HG31-0XB0)	6ES7292-1AV30-0XA0	20 broches, plaqué à l'étain

Tableau D- 2 SM S7-1200 V3.0 et antérieure - Kits de pièces de rechange du bornier

Si vous possédez SM S7-1200 V3.0 et version antérieure (numéro d'article)	Utilisez ce kit de pièces de rechange du bornier (4/pk)	
	Numéro d'article du bornier	Description du bornier
SM 1221 DI 8 x DC (6ES7221-1BF30-0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1222 DQ 8 x DC (6ES7222-1BF30-0XB0)		
SM 1222 DQ 8 x Relais (6ES7222-1HF30-0XB0)		
SM 1231 AI 4 x 13 bit (6ES7231-4HD30-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1232 AQ 2 x 14 bit (6ES7232-4HB30-0XB0)		
SM 1231 AI 4 x TC (6ES7231-5QD30-0XB0)		
SM 1231 AI 4 x 16 bit (6ES7231-5ND30-0XB0)		
SM 1221 DI 16 x DC (6ES7221-1BH30-0XB0)	6ES7292-1AG30-0XA0	7 broches, plaqué à l'étain
SM 1222 DQ 16 x DC (6ES7222-1BH30-0XB0)		
SM 1222 DQ 16 x Relais (6ES7222-1HH30-0XB0)		
SM 1223 DI 8 x DC/DQ 8x DC (6ES7223-1BH30-0XB0)		
SM 1223 8 x DC/8 x Relais (6ES7223-1PH30-0XB0)		
SM 1223 8 x AC/8 x Relais (6ES7223-1QH30-0XB0)		
SM 1234 AI 4/AQ 2 (6ES7234-4HE30-0XB0)	6ES7292-1BG30-0XA0	7 broches, plaqué-or
SM 1231 AI 8 x 13 bit (6ES7231-4HF30-0XB0)		
SM 1232 AQ 4 x 14 bit (6ES7232-4HD30-0XB0)		
SM 1231 AI 4 x RTD (6ES7231-5PD30-0XB0)		
SM 1231 AI 8 x TC (6ES7231-5QF30-0XB0)		
SM 1222 DQ 8 x Relais (inverseur) (6ES7222-1XF30-0XB0)	6ES7292-1AL30-0XA0	11 broches, plaqué à l'étain
SM 1223 DI 16 x DC/DQ 16 x DC (6ES7223-1BL30-0XB0)		
SM 1223 16 x DC/16 x Relais (6ES7223-1PL30-0XB0)		
SM 1231 AI 8 x RTD (6ES7231-5PF30-0XB0)	6ES7292-1BL30-0XA0	11 broches, plaqué-or

Index

A

- Abonnés accessibles
 - formatage d'une carte mémoire, 1391
- Abonnés accessibles, mise à jour du firmware, 1390
- ABS (Créer valeur absolue), 272
- AC
 - conseils de câblage, 79
- Accès
 - journaux à partir du PC, 1076
 - pages Web personnalisées, 1101
- ACOS (Créer valeur arc cosinus), 275
- Acquisition des valeurs dans un DB en ligne, 1398
- ACT_TINT (activer alarme horaire), 440
- Actualisation de pages Web personnalisées, 1080
- Actualisation des pages Web personnalisées, 1080
- Adaptateur TeleService et module, 72
- Adaptateur TS Adapter, 31
 - carte SIM, 74
 - installation d'un module TS, 72
 - installation sur un profilé support, 75
 - montage mural, 76
- ADD (Addition), 269
- Adressage
 - entrées (I) ou sorties (Q) individuelles, 123
 - mémoire image, 122
 - Valeurs booléennes ou de bit, 123
 - zones de mémoire, 122
- Adresse IP, 840, 841
 - adresse MAC, 840
 - Affectation, 836, 845
 - affectation en ligne, 838
 - configuration, 840
 - configuration de la CPU en ligne, 1386
 - configuration des appareils, 177
- Adresse IP de routeur, 841
- Adresse IP d'urgence (temporaire), 1036
- adresse MAC, 840
- Adresse MAC, 846
- Adresses
 - Lire l'adresse de station avec GetStationInfo, 471
 - Lire l'adresse MAC avec GetStationInfo, 471
- Adresses de mémoire, 121, 123
- Affectation de mémoire locale de la structure d'appel, 125
- Affectation de types Enum, pages Web personnalisées, 1090
- Affichage de l'usage des références croisées, 233
- Affichage des adresses MAC et IP, 846
- Ajout d'un type de données à une variable via AT, 141
- Ajouter des entrées ou sorties aux opérations CONT ou LOG, 44
- Ajouter un appareil
 - CPU, 160
 - CPU non spécifiée, 163
 - détection du matériel existant, 163
- Alarmes
 - ATTACH (affecter un OB à un événement déclencheur d'alarme), 430
 - CAN_DINT (annuler une alarme temporisée), 442
 - CAN_DINT (interroger l'état de l'alarme temporisée), 442
 - DETACH (dissocier un OB d'un événement déclencheur d'alarme), 430
 - présentation, 94
 - SRT_DINT (démarrer une alarme temporisée), 442
 - temps d'attente, 106
- Alarmes temporisées, 442
- Alias dans les pages Web personnalisées, 1088
- Analyseur logique, 1418
- API
 - affectation d'une adresse IP à une CPU en ligne, 838
 - ajouter des modules, 164
 - bilan de consommation, 56
 - bornier de connexion, 69
 - câble d'extension, 70
 - charge due à la communication, 112
 - chargement dans la CPU, 225
 - comparaison et synchronisation, 1394
 - Conception de système, 191
 - configuration des appareils, 159
 - Configuration HSC, 590
 - copie des blocs depuis une CPU en ligne, 231
 - états de fonctionnement, 89
 - forçage permanent, 1403, 1404
 - installation, 60, 62
 - présentation de la CPU, 27
 - propriété Synchronisation de l'heure, 849
 - protection du savoir-faire, 222
 - RTM (Compteurs d'heures de fonctionnement), 355
 - table de visualisation, 1400
 - temps de cycle, 111, 112
 - temps de cycle, 111, 112
 - traitement de mise en route, 92

- Utilisation de blocs, 192
 - variables, 121
 - visualisation, 1397
 - Appareil
 - noms du périphérique PROFINET IO, 955
 - partagé, 971
 - PROFINET IO, 954
 - Appareil partagé
 - concept, 968
 - configuration, 971
 - Appareils IHM
 - configuration de la communication PROFINET, 947
 - liaison réseau, 832
 - présentation, 33
 - Appel de blocs de code dans le programme utilisateur, 194
 - Architecture d'interrogation, 1187
 - Architecture d'interrogation esclave, 1187
 - Architecture d'interrogation maître, 1187
 - AS-i
 - adresse, 1012
 - affectation système, 1015
 - affectation système d'adresses d'esclaves, 1015
 - ajout d'un esclave AS-i, 1010
 - ajout d'un module maître AS-i CM 1243-2, 1010
 - configuration d'esclaves avec STEP 7, 1016
 - configuration d'esclaves sans STEP 7, 1015
 - instructions de périphérie décentralisée, 394
 - liaison réseau, 1011
 - maître AS-i CM 1243-2, 1009
 - RDREC (lire l'enregistrement), 395
 - transfert de valeurs analogiques, 1016
 - transfert de valeurs TOR, 1016
 - WRREC (écrire l'enregistrement), 395
 - ASIN (Créer valeur arc sinus), 275
 - Assistance, 3
 - Assistance client, 3
 - Assistance technique, 3
 - Assistance technique Siemens, 3
 - Assistant d'importation du certificat, 1126
 - ATH (Convertir la chaîne de caractères ASCII en nombre hexadécimal), 369
 - ATTACH (affecter un OB à un événement déclencheur d'alarme), 430
 - ATTR_DB (Lire les attributs d'un bloc de données), 559
 - Australie et Nouvelle-Zélande - autorisation de Marque RCM, 1432
 - Autonégociation, 843
 - AWP_Enum_Def, 1089
 - AWP_Import_Fragment, 1092
 - AWP_In_Variable, 1083, 1087
 - AWP_Out_Variable, 1085
 - AWP_Start_Fragment, 1091
- ## B
- Barre d'outils Favoris, 41
 - Basic panels (HMI), 33
 - Battery Board (BB)
 - BB 1297, 1620
 - insérer batterie, 1621
 - BB 1297, 1620
 - Besoins du système, 38
 - Bibliothèque du protocole USS
 - codes d'état, 1215
 - conditions requises pour l'utilisation, 1203
 - présentation, 1199
 - USS_Drive_Control (Données permutées avec l'entraînement), 1208
 - USS_Port_Scan (Editer communication via réseau USS), 1206
 - USS_Read_Param (lire des paramètres de l'entraînement), 1211
 - USS_Write_Param (modifier les paramètres dans l'entraînement), 1213
 - bibliothèque du protocole USS d'héritage
 - codes d'état, 1313
 - conditions requises pour l'utilisation, 1302
 - présentation, 1300
 - USS_DRV (Données permutées avec l'entraînement), 1306
 - USS_PORT (Editer communication via réseau USS), 1305
 - USS_RPM (lire des paramètres de l'entraînement), 1309
 - USS_WPM (modifier les paramètres dans l'entraînement), 1311
 - Bibliothèque globale
 - présentation du protocole USS, 1199
 - présentation du protocole USS d'héritage, 1300
 - Bilan de consommation, 56
 - exemple, 1649
 - formulaire pour le calcul, 1650
 - présentation, 1647
 - Bits d'arrêt, 1138
 - Bits de capture d'impulsions, configuration des entrées TOR, 181
 - Bloc de code
 - alarmes, 30, 1445, 1459, 1473, 1489, 1506
 - appel de blocs de code dans le programme utilisateur, 194
 - appels de blocs, 85
 - bloc de données d'instance (DB), 197

- blocs d'organisation
 - (OB), 30, 1445, 1459, 1473, 1489, 1506
 - Blocs d'organisation (OB), 195
 - compteurs (exigences en termes de quantité et de mémoire), 30, 1445, 1459, 1474, 1489, 1506
 - DB (bloc de données), 85, 199
 - FB (bloc fonctionnel), 85, 197
 - FC (fonction), 85, 197
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 - nombre de blocs de code, 30, 1445, 1459, 1473, 1489, 1506
 - nombre d'OB, 30, 1445, 1459, 1473, 1489, 1506
 - numéros de FC, FB et DB valides, 85
 - profondeur d'imbrication, 30, 1445, 1459, 1473, 1489, 1506
 - programmes linéaires et structurés, 192
 - protection contre la copie, 223
 - protection du savoir-faire, 222
 - taille du programme utilisateur, 30, 1445, 1459, 1473, 1489, 1506
 - temporisations (exigences en termes de quantité et de mémoire), 30, 1445, 1459, 1474, 1489, 1506
 - types de blocs de code, 85
 - valeur initiale d'un FB, 197
 - visualisation, 30, 1445, 1459, 1473, 1489, 1506
- Bloc de données
 - accès optimisé, 199
 - accès standard, 199
 - acquisition et réinitialisation des valeurs, 1398
 - bloc de données d'instance, 121
 - bloc de données global, 121, 199
 - Blocs d'organisation (OB), 195
 - CONF_DATA, 935
 - création avec CREATE_DB, 551
 - FB unique avec plusieurs DB d'instance, 198
 - importation de fragments dans les pages Web personnalisées, 1092
 - Lire les attributs avec ATTR_DB, 559
 - présentation, 85, 199
 - READ_DBL (lire dans un bloc de données dans la mémoire de chargement), 556
 - Structure, 85
 - Supprimer avec DELETE_DB, 561
 - WRIT_DBL (écrire dans un bloc de données dans la mémoire de chargement), 556
- Bloc de données
 - synchronisation des valeurs initiales en ligne et hors ligne, 229
- Bloc de données d'instance, 121
- Bloc de données global, 121, 199
- Bloc de gestion de données (DHB), 199
- Bloc de transmission (bloc T), 951
- Bloc d'organisation
 - affectation de mémoire temporaire, 125
 - alarme cyclique, 96
 - appel, 94
 - appel de blocs de code dans le programme utilisateur, 194
 - classes de priorité, 94
 - Configuration du fonctionnement, 196
 - création, 196
 - fonction, 94
 - plusieurs OB de cycle de programme, 196
 - présentation, 85
 - programmation linéaire et structurée, 192
 - protection du savoir-faire, 222
 - traitement, 195, 195
 - traitement de mise en route, 92
- Bloc d'organisation (OB)
 - Lecture des informations de déclenchement avec RD_SINFO, 450
- Bloc fonctionnel (FB)
 - appel de blocs de code dans le programme utilisateur, 194
 - bloc de données d'instance, 197
 - FB unique avec plusieurs DB d'instance, 198
 - numéros de FB valides, 85
 - paramètres de sortie, 197
 - présentation, 85, 197
 - programmes linéaires et structurés, 192
 - protection du savoir-faire, 222
 - valeur initiale, 197
- Blocs
 - alarmes, 30, 106, 1445, 1459, 1473, 1489, 1506
 - appel d'un FB ou d'une FC avec SCL, 210
 - appels de blocs, 85
 - bloc de données (DB), 85
 - bloc de données d'instance (DB), 197
 - bloc fonctionnel (FB), 85, 197
 - blocs d'organisation (OB), 30, 85, 94, 1445, 1459, 1473, 1489, 1506
 - Blocs d'organisation (OB), 106
 - chargement dans la CPU, 225
 - compteurs (exigences en termes de quantité et de mémoire), 30, 1445, 1459, 1474, 1489, 1506
 - contrôle de cohérence, 234
 - copie des blocs depuis une CPU en ligne, 231
 - DB mono-instance ou multi-instance, 197
 - événements, 106
 - fonction (FC), 85, 197
 - nombre de blocs de code, 30, 1445, 1459, 1473, 1489, 1506

- nombre
 - d'OB, 30, 106, 1445, 1459, 1473, 1489, 1506
 - numéros de FC, FB et DB valides, 85
 - OB de démarrage, 106
 - profondeur
 - d'imbrication, 30, 85, 1445, 1459, 1473, 1489, 1506
 - programmes linéaires et structurés, 192
 - protection par mot de passe, 222
 - taille du programme
 - utilisateur, 30, 85, 1445, 1459, 1473, 1489, 1506
 - temporisations (exigences en termes de quantité et de mémoire), 30, 1445, 1459, 1474, 1489, 1506
 - types de, 85
 - types de blocs de code, 85
 - valeur initiale d'un FB, 197
 - visualisation, 30, 1445, 1459, 1473, 1489, 1506
- Blocs de données optimisés, 199
- Blocs de données standard, 199
- Bobine à fermeture/ouverture, 239
- Bobines, (Voir les instructions logiques sur bits)
- boîte & (ET logique en LOG), 238
- boîte /= (inverser affectation en LOG), 239
- boîte = (affectation en LOG), 239
- boîte >=1 (OU logique en LOG), 238
- boîte x (OU exclusif en LOG), 238
- Bornier de connexion, 69
- Bornier, montage et démontage, 69
- Boucle d'asservissement de position
 - configuration de l'axe, 694
 - entraînement analogique, 694
 - PROFIdrive, 694
- Boutons MARCHE/ARRET, 47
- BUFFER, paramètre de SEND_P2P, 1176

- C**
- Câble
 - Communication réseau, 1130
 - extension, 1642
- Câble d'extension, 1642
 - démontage, 70
 - installation, 70
- CALCULER (calculer), 268
 - mise à l'échelle de valeurs analogiques, 43
 - utilisé pour les équations complexes, 42
- Calendrier, 347
- CAN_DINT (annuler une alarme temporisée), 442
- CAN_DINT (interroger l'état de l'alarme temporisée), 442
- CAN_TINT (annuler l'alarme horaire), 439
- Capture d'impulsions, 181, 185
- Caractère de début de message, 1143
- Caractère de fin de message, 1147
- Caractères spéciaux
 - pages Web personnalisées, 1094
- Caractéristiques
 - BB 1297, 1620
 - caractéristiques techniques d'ordre général, 1429
 - cartes mémoire, 1638
 - CB 1241 RS485, 1634
 - CM 1241 RS232, 1635
 - CM 1241 RS422/485, 1636
 - Compatibilité électromagnétique (CEM), 1434
 - Conditions ambiantes, 1435
 - CPU 1211C CA/CC/Relais, 1442
 - CPU 1211C CC/CC/CC, 1442
 - CPU 1211C CC/CC/Relais, 1442
 - CPU 1212C CA/CC/Relais, 1456
 - CPU 1212C CC/CC/CC, 1456
 - CPU 1212C CC/CC/Relais, 1456
 - CPU 1214C CA/CC/Relais, 1471
 - CPU 1214C CC/CC/CC, 1471
 - CPU 1214C CC/CC/Relais, 1471
 - CPU 1215C CA/CC/Relais, 1487
 - CPU 1215C CC/CC/CC, 1487
 - CPU 1215C CC/CC/Relais, 1487
 - CPU 1217C CC/CC/CC, 1504
 - environnements industriels, 1433
 - homologations, 1430
 - module de potentiomètre, 1641
 - module SM 1231 AI 4 x RTD x 16 bits, 1566
 - module SM 1231 AI 8 x RTD x 16 bits, 1566
 - représentation des entrées analogiques (courant), 1555, 1607
 - représentation des entrées analogiques (tension), 1554, 1607
 - SB 1221 DI 4 x 24 V DC, 200 kHz, 1587
 - SB 1221 DI 4 x 5 V DC, 200 kHz, 1587
 - SB 1222 DQ 4 x 24 V DC, 200 kHz, 1590
 - SB 1222 DQ 4 x 5 V DC, 200 kHz, 1590
 - SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC, 1598
 - SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz, 1593
 - SB 1223 DI 2 x 5 V DC/DQ 2 x 5 V DC, 200 kHz, 1593
 - SB 1231 AI 1 x 12 bits, 1601
 - SB 1231 AI 1 x 16 bits RTD, 1615
 - SB 1231 AI 1 x 16 bits Thermocouple, 1610
 - SB 1232 AQ 1 x 12 bits, 1604
 - simulateurs d'entrées, 1639
 - SM 1221 DI 16 x 24 V DC, 1521
 - SM 1221 DI 8 x 24 V DC, 1521
 - SM 1222 DQ 16 x 24 V DC, 1525
 - SM 1222 DQ 16 x Relais, 1525

- SM 1222 DQ 8 Relais Inverseur, 1523
- SM 1222 DQ 8 x 24 V DC, 1523
- SM 1222 DQ 8 x Relais, 1523
- SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC, 1531
- SM 1223 DI 16 x 24 V DC, DQ 16 x Relais, 1531
- SM 1223 DI 8 24 V DC, DQ 8 x Relais, 1531
- SM 1223 DI 8 x 120/230 V AC/DQ 8 x Relais, 1538
- SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC, 1531
- SM 1231 AI 4 x 13 bits, 1541
- SM 1231 AI 4 x 16 bits, 1541
- SM 1231 AI 4 x 16 bits TC, 1558
- SM 1231 AI 8 x 13 bits, 1541
- SM 1231 AI 8 x 16 bits TC, 1558
- SM 1232 AQ 2 x 14 bits, 1547
- SM 1232 AQ 4 x 14 bits, 1547
- SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits, 1550
- temps de réponse indicielle (CPU), 1451, 1465, 1479, 1495, 1514
- temps de réponse indicielle (SB), 1606
- temps de réponse indicielle (SM), 1553
- tensions nominales, 1437
- Caractéristiques techniques, 1429
 - Maître SM 1278 4xIO-Link, 1573
 - représentation des sorties analogiques (courant), 1557, 1609
 - représentation des sorties analogiques (tension), 1556, 1608
- Carte mémoire, 1638
 - carte programme, 151
 - Carte transfert, 148
 - carte transfert vide en cas d'oubli du mot de passe, 157
 - configuration des paramètres de mise en route, 147
 - erreur d'incompatibilité, 1380
 - insertion dans la CPU, 143
 - mise à jour du firmware, 153
 - oubli du mot de passe, 157
 - présentation, 143
- carte mémoire CPU
 - carte programme, 151
 - carte transfert, 148
 - Insertion, 143
- Carte programme
 - configuration des paramètres de mise en route, 147
 - création, 151
 - insertion dans la CPU, 143
 - présentation, 143
- Carte transfert, 148
 - carte transfert vide en cas d'oubli du mot de passe, 157
 - configuration des paramètres de mise en route, 147
 - insertion dans la CPU, 143
 - oubli du mot de passe, 157
 - présentation, 143
- Cartes transfert (programme), 1638
- Catégorie de surtension, 1436
- CB 1241
 - terminaison et polarisation, 1131
- CB 1241 RS485, 1634
- CC
 - conseils de câblage, 79
- CEIL (Arrondir à l'entier supérieur), 307
- Certificat de sécurité Siemens, pages Web, 1054, 1126
- Chaîne
 - présentation des instructions sur chaîne, 371
 - S_MOVE (Déplacer la chaîne de caractères), 357
- Chaîne de caractères
 - STRING, type de données, 135
- Changement de langue pour les pages Web personnalisées, 1115
- Changer l'appareil, 177
- Char (type de données caractère), 135
- Chargement à l'état MARCHE
 - chargement sans réinitialisation, 1412
 - interface de bloc agrandie, 1412
 - paramètres globaux de réserve de mémoire, 1414
 - réserve de mémoire et réserve dans la mémoire rémanente, 1412
 - Restrictions, 1414
- Chargement à partir de l'appareil
 - copie des blocs depuis une CPU en ligne, 231
 - programme utilisateur, 231
- Chargement dans la CPU
 - affichage des adresses MAC et IP, 846
 - DB de pages Web personnalisées, 1101
 - programme utilisateur, 225
 - projet, 225
- Chargement dans la CPU à l'état MARCHE
 - chargement de blocs sélectionnés, 1409
 - conditions requises, 1407
 - considérations, 1415
 - déclenchement depuis STEP 7, 1408
 - échec du chargement, 1415
 - erreurs de compilation, 1411
 - présentation, 1406
- Charges de lampe, 80
- Charges inductives, 81
- Chars_TO_Strg (Convertir array of CHAR en chaîne de caractères), 367
- Circuits de protection par écrêtage pour les charges inductives, 81
- Circuits de suppression pour les charges inductives, 81
- Classe de protection, 1436
- Classes d'erreurs point à point, 1157, 1281

- Colonnes et en-têtes dans les Task Cards, 45
- Commande de configuration (gestion des options), 165
 - enregistrement de commande, 170
 - exemple, 173
- Commande de DB de fragment manuels, 1119
- Commande de mouvement
 - configuration de l'axe, 684
 - fins de course matériels et logiciels, 736
 - MC_ReadParam (lire les paramètres de l'objet technologique), 784
 - MC_WriteParam (écrire dans les paramètres d'un objet technologique), 782
 - paramètres de configuration du référencement, 747
 - phasage, 680
 - présentation, 674
 - PTO, 684
 - référencement (séquence pour le référencement actif), 751
 - référencement de l'axe, 746
- Commandes AWP, 1081
 - combinaison de définitions, 1093
 - définition d'un type Enum, 1089
 - écriture de variables, 1083
 - écriture de variables spéciales, 1087
 - générations de fragments, 1091
 - importation de fragments, 1092
 - lecture de variables spéciales, 1085
 - référencement d'un type Enum, 1090
 - utilisation d'un alias, 1088
- Communication
 - active/passive, 833, 857, 1028
 - adresse AS-i, 1012
 - adresse IP, 840
 - adresse MAC, 840
 - adresse PROFIBUS, 1007
 - Architecture d'interrogation, 1187
 - charge due à la communication, 112
 - configuration, 833, 857, 1028
 - Contrôle du flux, 1139
 - ID de liaison, 854
 - Liaison matérielle, 945
 - liaison réseau, 832
 - nombre de liaisons (PROFINET/PROFIBUS), 828
 - paramètres d'émission et de réception, 1140
 - perte, débrogage/enfichage de modules, 101
 - PROFINET et PROFIBUS, 825
 - propriété Synchronisation de l'heure (PROFINET), 849
 - protocoles, 853
 - Réseau, 945
 - TCON_Param, 857
 - temps de cycle, 112
- Communication active/passive
 - configuration des partenaires, 833, 1028
 - ID de liaison, 854
 - paramètres, 857
- Communication Board (CB)
 - ajouter des modules, 164
 - CB 1241 RS485, 1634
 - configuration des appareils, 159
 - configuration des paramètres, 185
 - DEL de signalisation, 1129, 1379
 - démontage, 64
 - installation, 64
 - présentation, 32
 - programmation, 1186
 - RS485, 1129
 - tableau comparatif, 31
- Communication ouverte (Open User Communication)
 - établissant une liaison et un envoi des données avec les instructions d'héritage TSEND_C, 877
 - établissant une liaison et une lecture des données avec les instructions d'héritage TRCV_C, 877
 - établissement d'une connexion et envoi de données avec TSEND_C, 863
 - établissement d'une connexion et lecture de données avec TRCV_C, 863
- Communication passive/active
 - configuration des partenaires, 833, 1028
 - ID de liaison, 854
 - paramètres, 857
- Communication point à point, 1132, 1132
 - configuration de l'exemple de programme, 1189
 - configuration des paramètres, 1140
 - configuration des ports, 1137
 - émulateur de terminal pour l'exemple de programme, 1197
 - exemple de programme, 1188
 - exemple de programme, exécution, 1198
 - exemple de programme, programmation STEP 7, 1196
 - programmation, 1186
- Communication programmable, protocole, 1132
- Communication PtP, 3964(R)
 - configuration de la priorité et des paramètres du protocole, 1153
 - configuration des ports, 1151
- Communication réseau, 945
 - polarisation et terminaison du câble, 1130
- Communication S7
 - configuration de la liaison, 834
- Communication série, 1132
- Communication TCP/IP, 851

- Communication TeleService
 - TM_MAIL (envoyer e-mail), 1371
- Communication, page Web standard, 1064
- Comparaison et synchronisation des CPU hors ligne/en ligne, 1394
- Comparer valeurs, 262
- Compatibilité, 51
- Compatibilité électromagnétique, 1434
- Compatibilité électromagnétique (CEM), 1434
- Compensation de soudure froide, thermocouples, 1562, 1612
- Compteur rapide, 574, 603
 - Configuration, 590
 - forçage permanent impossible, 1405
 - modes de comptage, 592
 - phase de fonctionnement, 593
- Compteurs
 - Configuration HSC, 590
 - CTD (Décomptage), 255
 - CTRL_HSC (Commande de compteurs rapides), 603
 - CTRL_HSC_EXT (Commander le compteur rapide (avancé)), 574
 - CTU (Comptage), 255
 - CTUD (Comptage et décomptage), 255
 - opération (compteurs standard), 257
 - quantité, 30, 1445, 1459, 1474, 1489, 1506
 - taille, 30, 1445, 1459, 1474, 1489, 1506
- Compteurs d'heures de fonctionnement (RTM), 355
- CONCAT (combinaison des chaînes de caractères), 372
- Conception d'un système d'automatisation, 191, 192
- Conditions
 - d'installation, 38
- Conditions ambiantes
 - conditions de service, 1435
 - conditions de transport et stockage, 1435
- Conditions de début, 1143
- Conditions de fin, 1146
- Configuration
 - adresse IP, 840
 - adresse MAC, 840
 - adresse PROFIBUS, 1007
 - ajouter des modules, 164
 - AS-i, 1012
 - charge due à la communication, 112
 - chargement dans la CPU, 225
 - communication API-API, 949
 - découvrir, 163
 - HSC (compteur rapide), 590
 - Instructions PID_Compact et PID_3Step, 643
 - Instructions PID_Temp, 646
 - Interfaces de communication, 1137
 - liaison réseau, 832
 - modules, 185
 - paramètres de la CPU, 177
 - paramètres de mise en route, 147
 - port AS-i, 1011
 - Port Ethernet, 840
 - port PROFIBUS, 840
 - Ports, 1137
 - PROFIBUS, 1006
 - propriété Synchronisation de l'heure (PROFINET), 849
 - réception des messages, 1142
 - RS422, modes de fonctionnement, 1192
 - RS485, modes de fonctionnement, 1194
 - temps de cycle, 111
- Configuration de la CPU
 - communication avec des IHM, 947
 - Paramètres de fonctionnement, 177
 - plusieurs CPU, 949
 - propriétés de module, 185
 - surveillance du temps de cycle, 111
 - voies d'impulsion, 507
- Configuration de la transmission de messages, 1141
- Configuration de l'appareil
 - Changer un type d'appareil, 177
- Configuration de l'émission de messages
 - configuration de l'appareil PtP, 1141
 - exemple de programme point à point, 1190
- Configuration des appareils, 159, 946
 - ajouter des modules, 164
 - ajouter un appareil, 160
 - AS-i, 1012
 - chargement dans la CPU, 225
 - configuration de la CPU, 177
 - configuration des modules, 185
 - découvrir, 163
 - liaison réseau, 832
 - modules retirés, 50
 - port AS-i, 1012
 - Port Ethernet, 840
 - port PROFIBUS, 840
 - PROFIBUS, 1006
 - propriété Synchronisation de l'heure (PROFINET), 849
- Configuration des messages
 - émission, 1141
 - Instructions, 1186
 - réception, 1142
- Configuration des paramètres
 - Emission, 952
 - LENGTH et BUFFER pour SEND_P2P, 1176
 - Réception, 954

- Configuration du port, 1137
 - erreurs, 1161, 1284
 - exemple de programme point à point, 1189
 - Instructions, 1186
- Configuration matérielle, 159
 - ajouter des modules, 164
 - ajouter un appareil, 160
 - AS-i, 1012
 - chargement dans la CPU, 225
 - configuration de la CPU, 177
 - configuration des modules, 185
 - découvrir, 163
 - liaison réseau, 832
 - port AS-i, 1012
 - Port Ethernet, 840
 - port PROFINET, 840
 - PROFIBUS, 1006
- Configuration utilisateur, serveur Web, 1042
- Configuration, 3964(R)
 - interfaces de communication, 1151
 - ports, 1151
 - priorité et paramètres du protocole, 1153
- Configuration, pages Web personnalisées
 - Configuration dans STEP 7, 1097
 - configuration de plusieurs langues, 1119
- Connecteur de bus, 32
- Connecteur RS485
 - terminaison et polarisation, 1130
- Connexion sans fil au serveur Web, 1046
- Connexions
 - nombre de liaisons (PROFINET/PROFIBUS), 828
- Connexions maximum du serveur Web, 1125
- Conseils
 - charges de lampe, 80
 - charges inductives, 81
 - conseils de câblage, 77, 79
 - installation, 53
 - installation de la CPU, 62
 - isolation, 78
 - mise à la terre, 78
 - procédures d'installation, 60
- Conseils de câblage, 79
 - conditions requises, 77
 - dégagement pour l'écoulement d'air et le refroidissement, 55
 - mise à la terre, 78
- consommation de courant, 56, 1647
- CONT (schéma à contacts)
 - éditeur de programme, 1398
 - état, 1398, 1403
 - présentation, 206
 - visualisation, 1398
 - visualisation de l'état ou de la valeur, 1397
- Contact à fermeture/ouverture, 237
- Contacts, (Voir les instructions logiques sur bits)
- Contacts de connexion
 - Capacité de transport de courant maximale, 1632
- CONTINUE, SCL, 336
- Contraintes
 - pages Web personnalisées, 1102
 - serveur Web, 1124
- Contrôle de cohérence, 234
- Contrôle de flux logiciel, 1140
- Contrôle de flux matériel, 1139
- Contrôle du flux, 1138
 - Configuration, 1138
 - gestion, 1139
- CONV (Convertir valeur), 302
- Conversion (instructions SCL), 303
- Cookie siemens_automation_language, 1115
- Copie des blocs depuis une CPU en ligne, 231
- Copier, couper et coller dans STEP 7, 51
- COS (Créer valeur cosinus), 275
- CountOfElements (Obtenir le nombre d'éléments du TABLEAU), 298
- Courant alternatif
 - conseils de câblage, 77
 - conseils d'isolation, 78
 - mise à la terre, 78
- Courant continu
 - charges inductives, 81
 - conseils de câblage, 77
 - conseils d'isolation, 78
 - mise à la terre, 78
 - sorties, 1437
- Courriel, envoi avec TMAIL_C, 913
- CPU
 - acquisition et réinitialisation de valeurs de DB, 1398
 - adresse AS-i, 1012
 - adresse IP, 840
 - adresse MAC, 840, 840, 846
 - adresse PROFIBUS, 1007
 - affectation d'une adresse IP à une CPU en ligne, 838
 - affichage des adresses MAC et IP, 846
 - ajouter des modules, 164
 - ajouter un appareil, 160, 160
 - AS-i, 1011
 - besoins en courant, 1647
 - bilan de consommation, 56
 - bornier de connexion, 69
 - boutons MARCHE/ARRET, 47
 - câble d'extension, 70

- Carte transfert vide, 157
 - charge due à la communication, 112
 - chargement dans la CPU, 225
 - Charger dans l'appareil, 846
 - charges de lampe, 80
 - charges inductives, 81
 - Communication, 830
 - Communication Boards (CB), 32
 - comparaison et synchronisation des blocs, 1394
 - compatibilité des versions, 51
 - comportement en surcharge, 108
 - configuration des appareils, 159, 159
 - configuration du temps de cycle, 112
 - Configuration HSC, 590
 - conseils de câblage, 77, 79
 - conseils d'isolation, 78
 - copie des blocs depuis une CPU en ligne, 231
 - CPU 1211C CA/CC/Relais, 1442
 - CPU 1211C CC/CC/CC, 1442
 - CPU 1211C CC/CC/Relais, 1442
 - CPU 1212C CA/CC/Relais, 1456
 - CPU 1212C CC/CC/CC, 1456
 - CPU 1212C CC/CC/Relais, 1456
 - CPU 1214C CA/CC/Relais, 1471
 - CPU 1214C CC/CC/CC, 1471
 - CPU 1214C CC/CC/Relais, 1471
 - CPU 1215C CA/CC/Relais, 1487
 - CPU 1215C CC/CC/CC, 1487
 - CPU 1215C CC/CC/Relais, 1487
 - CPU 1217C CC/CC/CC, 1504
 - CPU non spécifiée, 163
 - déblocage des sorties à l'état ARRET, 1402
 - DEL de signalisation, 1379
 - En ligne, 1386
 - états de fonctionnement, 89
 - Etats RUN/STOP, 1392
 - exécution du programme, 85
 - forçage permanent, 1403, 1404
 - installation, 60, 62
 - liaison réseau, 832
 - mise à la terre, 78
 - niveaux de sécurité, 218
 - nombre de liaisons de communication, 828
 - oubli du mot de passe, 157
 - panneau de commande (CPU en ligne), 1392
 - paramètres de mise en route, 147
 - passage en ligne, 1383
 - port AS-i, 1011
 - Port Ethernet, 840
 - port PROFINET, 840
 - présentation, 27
 - PROFINET IO, 954
 - propriété Synchronisation de l'heure, 849
 - protection d'accès par mots de passe, 218
 - protection du savoir-faire, 222
 - pupitre opérateur, 47
 - récupération en cas d'oubli du mot de passe, 157
 - restauration des réglages d'usine, 1387
 - restauration d'une sauvegarde, 1428
 - RTM (Compteurs d'heures de fonctionnement), 355
 - sauvegarde, 1425
 - Signal Boards (SB), 32
 - sorties d'impulsions, 504
 - table de visualisation, 1400
 - tableau comparatif, 28
 - temps de réponse
 - indicielle, 1451, 1465, 1479, 1495, 1514
 - traitement de mise en route, 92
 - traitement des OB, 195
 - types de communication, 825
 - visualisation en ligne, 1397
 - zone thermique, 55, 59
 - CPU non spécifiée, 163
 - CREATE_DB (Créer un bloc de données), 551
 - Création de DB de pages Web personnalisées, 1098
 - Création de pages Web personnalisées, 1080
 - Création d'une liaison réseau
 - entre API, 832
 - CTD (Décomptage), 255
 - CTRL_HSC (commande de compteurs rapides), 603
 - CTRL_HSC_EXT (Commander le compteur rapide (avancé)), 574
 - CTS (contrôle de flux matériel, PtP), 1139
 - CTU (Comptage), 255
 - CTUD (Comptage et décomptage), 255
 - Cycle
 - forçage permanent, 1404, 1404
 - présentation, 111
- ## D
- D_ACT_DP, 408
 - DataLogClear, 534
 - DataLogDelete, 537
 - Date
 - DTL (type de données date et heure long), 134
 - SET_TIMEZONE (Définir fuseau horaire), 354
 - T_ADD (Additionner les temps), 348
 - T_COMBINE (Combiner temps), 349
 - T_CONV (Convertir et extraire les temps), 347
 - T_DIFF (Différence de temps), 349
 - T_SUB (Soustraire les temps), 348
 - type de données Date, 133
 - DB (bloc de données), (Bloc de données)

- DB de commande pour les pages Web personnalisées
 - commandes et états de requête, 1119
 - commandes globales, 1119
 - paramètre pour l'instruction WWW, 1099
 - DB de fragment (pages Web personnalisées)
 - création par une commande AWP, 1091
 - générations, 1098
 - importation avec une commande AWP, 1092
 - DB_ANY_TO_VARIANT (Convertir DB_ANY en VARIANT), 312
 - Débogage à l'état MARCHE, 1406, 1415
 - DEC (Décrémenter), 272
 - Déclenchement
 - traçage, 1418
 - valeurs dans la table de visualisation, 1401
 - DECO (Décoder), 340
 - Découvrir pour télécharger une CPU en ligne, 163
 - Définition de plusieurs variables AWP, 1093
 - Définition de types Enum, pages Web personnalisées, 1089
 - Dégagement, écoulement d'air et refroidissement, 55
 - Degré de pollution, 1436
 - Degré de protection, 1436
 - DEL de signalisation
 - état de la CPU, 1379
 - interface de communication, 1129, 1379
 - Délai inter-caractères, 1147
 - DELETE (supprimer caractères dans une chaîne de caractères), 375
 - DELETE_DB (Supprimer le bloc de données), 561
 - Démarrage à chaud, 89
 - Démarrage après mise sous tension, 89
 - traitement de mise en route, 92
 - Démarrage, page Web standard, 1054
 - DEMUX (Démultiplexeur), 343
 - Dépannage
 - DEL de signalisation, 1379
 - mémoire tampon de diagnostic, 1393
 - Déphasage, OB d'alarme cyclique, 96
 - Deserialize, 281
 - DETACH (dissocier un OB d'un événement déclencheur d'alarme), 430
 - DeviceStates (lire l'état du module d'un système I/O), 479
 - DeviceStates, exemple, 481
 - Diagnostic
 - DEL de signalisation, 1379
 - DeviceStates (lire l'état du module d'un système I/O), 479
 - GET_DIAG (lire l'information de diagnostic), 491
 - Get_IM_Data (Lire les données d'identification et de maintenance), 462
 - indicateur d'état, 116
 - LED (lire l'état de DEL), 461
 - mémoire tampon de diagnostic, 1393
 - ModuleStates (lire les informations d'état d'un module), 485
 - table de visualisation, 1400
 - tampon, 118
 - temps de cycle, 1392
 - utilisation de la mémoire, 1392
 - Diagnostic, page Web standard, 1056, 1059
 - Diagnostic, réduction des événements de sécurité, 118
 - Différences
 - concernant les instructions Modbus RTU, 1251
 - concernant les instructions Modbus TCP, 1224
 - concernant les instructions point à point, 1133
 - concernant les instructions USS, 1200
 - dans les instructions TCON, TDISCON, TSEND, et TRCV, 884
 - dans les instructions TSEND_C et TRCV_C, 862
 - DIS_AIRT (désactiver les alarmes de priorité supérieure et les erreurs asynchrones), 444
 - Dispositif mobile, accès au serveur Web, 1046
 - Dispositifs mobiles
 - disposition des pages Web, 1049
 - DIV (Division), 269
 - Diverses erreurs de paramètres de la communication point à point, 1157
 - Documentation, 4
 - Dossiers, langues pour les pages Web personnalisées, 1115
 - DPNRM_DG, 426
 - DPRD_DAT (lire données cohérentes d'un esclave DP standard), 418
 - DPWR_DAT (écrire données cohérentes d'un esclave DP standard), 418
 - Durée d'utilisation électrique des relais, 1438
- E**
- E/S
 - adressage, 127
 - charges inductives, 81
 - forçage permanent, 1404
 - indicateurs d'état analogiques, 1382
 - indicateurs d'état TOR, 1380
 - représentation des entrées analogiques (courant), 1555, 1607
 - représentation des entrées analogiques (tension), 1554, 1607
 - représentation des sorties analogiques (courant), 1557, 1609

- représentation des sorties analogiques (tension), 1556, 1608
- temps de réponse indicielle (CPU), 1451, 1465, 1479, 1495, 1514
- temps de réponse indicielle (SB), 1606
- temps de réponse indicielle (SM), 1553
- visualisation avec une table de visualisation, 1400
- visualisation de l'état dans CONT, 1398
- E/S analogiques
 - configuration, 185
 - conversion en unités physiques, 43, 128, 310
 - indicateurs d'état, 1382
 - représentation des entrées (courant), 1555, 1607
 - représentation des entrées (tension), 1554, 1607
 - représentation des sorties (courant), 1557, 1609
 - représentation des sorties (tension), 1556, 1608
 - temps de réponse indicielle (CPU), 1451, 1465, 1479, 1495, 1514
 - temps de réponse indicielle (SB), 1606
 - temps de réponse indicielle (SM), 1553
- E/S TOR
 - Capture d'impulsions, 185
 - configuration, 185
 - indicateurs d'état, 1380
- Echange de données entre systèmes IO, 962
- Ecoulement d'air, 55
- Ecriture dans les DB, les E/S ou la mémoire, 214, 293
- Éditeur de programmes
 - état, 1398
 - visualisation, 1398
- Edition à l'état MARCHE, (Chargement dans la CPU à l'état MARCHE)
- Emission de messages, configuration, 1141
- Emission, configuration des paramètres, 833, 952, 1028
- Émulateur de terminal pour l'exemple de programme point à point, 1197
- EN et ENO (flux de courant), 216
- En ligne
 - acquisition et réinitialisation de valeurs de DB, 1398
 - Adresse IP, 1386
 - affectation d'une adresse IP, 838
 - boutons MARCHE/ARRET, 47
 - comparaison et synchronisation, 1394
 - état, 1398
 - forçage permanent, 1403, 1404
 - heure, 1386
 - mémoire tampon de diagnostic, 1393
 - outils, 1396
 - panneau de commande, 1392
 - passage en ligne, 1383
 - pupitre opérateur, 47
 - surveillance de l'état ou de la valeur, 1397
 - table de visualisation, 1397, 1398, 1400
 - temps de cycle, 1392
 - utilisation de la mémoire, 1392
- EN_AIRT (activer les alarmes de priorité supérieure et les erreurs asynchrones), 444
- ENCO (Encoder), 340
- ENDIS_PW (activer/désactiver les mots de passe), 319
- Enregistrement des fichiers de sauvegarde, 1427
- Entraînement analogique, 694
- entraînement MicroMaster, connexion, 1218
- Entraînements, configuration de l'entraînement MicroMaster série 4, 1220
- Entrées
 - Bits de capture d'impulsions, 181
- Entrées et sorties
 - visualisation, 1397
- Environnement matériel requis, 38
- Environnements industriels
 - homologations, 1433
- EQ_ElemType (Comparer le type de données d'un élément TABLEAU pour INEGAL avec le type de données d'une variable), 265
- EQ_Type (Comparer le type de données pour EGAL avec le type de données d'une variable), 265
- Erreur version de CPU incompatible, 1380
- Erreur version de CPU inconnue, 1380
- Erreurs
 - erreurs communes pour les instructions avancées, 571
 - erreurs de diagnostic, 100
 - erreurs de temps, 98
- Erreurs à l'exécution de la réception, 1177, 1293
- Erreurs à l'exécution de l'émission, 1175, 1293
- Erreurs de configuration de réception, 1169, 1290
- Erreurs de configuration d'émission, 1164, 1285
- Erreurs de gestion des signaux, 1181, 1183, 1298, 1299
- Esclave PN
 - Activation et désactivation avec D_ACT_DP, 408
- Esclaves DP norme
 - Écrire toutes les sorties avec SETIO, 399
 - Écrire une partie des sorties avec SETIO_PART, 403
 - Lire toutes les entrées avec GETIO, 398
 - Lire une partie des entrées avec GETIO_PART, 401
- ET (opération logique), 339
- Etat
 - DEL de signalisation, 1379
 - DEL de signalisation (interface de communication), 1129

- Etat ARRET, 89, 1392
 - boutons de la barre d'outils, 47
 - déblocage des sorties à l'état ARRET, 1402
 - forçage permanent, 1404
 - Pupitre opérateur, 47
- Etat de fonctionnement, 47
- État de fonctionnement, 47
- État des variables, page Web standard, 1068
- Etat MARCHE, 89, 93, 1392
 - boutons de la barre d'outils, 47
 - forçage permanent, 1404
 - Pupitre opérateur, 47
- Etat MISE EN ROUTE
 - forçage permanent, 1404
- Ethernet
 - adresse IP, 840
 - adresse MAC, 840
 - DPNRM_DG (lire les données de diagnostic d'un esclave DP), 426
 - DPRD_DAT (lire données cohérentes d'un esclave DP standard), 418
 - DPWR_DAT (écrire données cohérentes d'un esclave DP standard), 418
 - GET (lire les données d'une CPU distante), 1021
 - ID de liaison, 854
 - Instructions d'héritage TCON, TDISCON, TSEND, et TRCV, 896
 - Instructions d'héritage TRCV_V (recevoir des données via Ethernet (TCP)), 877
 - Instructions d'héritage TSEND_C (envoyer des données via Ethernet (TCP)), 877
 - liaison réseau, 832
 - mode ad hoc, 854
 - Module commutateur compact CSM 1277, 1643
 - nombre de liaisons de communication, 828
 - présentation, 851
 - PRVREC (Mettre l'enregistrement à disposition), 424
 - PUT (écrire les données sur une CPU distante), 1021
 - RALRM (alarme de réception), 404
 - RCVREC (Recevoir l'enregistrement), 421
 - RDREC (lire l'enregistrement), 395
 - T_CONFIG (configurer interface), 931
 - TCON, 885
 - TDISCON, 885
 - TRCV, 885
 - TRCV_C, 863
 - TSEND, 885
 - TSEND_C, 863
 - TURCV (recevoir des données via Ethernet (UDP)), 924
 - TUSEND (envoyer des données via Ethernet (UDP)), 924
 - types de communication, 825
 - WRREC (écrire l'enregistrement), 395
- Événements de sécurité dans le tampon de diagnostic, 118
- Exécute des commandes d'axe en tant que séquence de mouvement (MC_CommandTable), 776
- Exécution d'événement et mise en fil d'attente, 106
- Exécution du programme, 85
- Exemple
 - ajout d'un entraînement SINAMICS S120, 707
 - configuration de détection de front pour limite de position ou came de référence d'entrée, 740
 - sélection d'un niveau pour la came de référence active, 751
 - sélection d'un niveau pour la came de référence passive, 750
- Exemple de programme de recette, 521
- Exemple ModuleStates, 487
- Exemples Modbus RTU hérité
 - programme esclave, 1356
 - programme maître, 1354
- Exemples Modbus TCP hérité
 - adressage du registre de maintien, 1329
 - demande d'écriture dans la mémoire image des sorties MB_CLIENT, 1334
 - Liaisons Modbus TCP multiples
 - MB_SERVER, 1331
 - MB_CLIENT : plusieurs demandes avec une liaison Modbus TCP commune,
 - MB_CLIENT coordonnant plusieurs demandes Modbus TCP, 1334
 - Paramètre MB_HOLD_REG, 1327
 - Plusieurs demandes MB_CLIENT avec des liaisons Modbus TCP différentes, 1333
- Exemples variés
 - accès à des éléments du tableau, 301
 - calcul de bilan de consommation, 1649
 - commande de configuration (gestion des options), 173
 - Connexion maître S7-1200 IO-Link, 1579
 - Entrée différentielle et application de la CPU 1217C, 1519
 - Evaluation ENO dans SCL, 217
 - instructions CASE imbriquées, SCL, 333
 - programme de fichiers journaux, 546
 - recette, 513, 521
 - Sortie différentielle et application de la CPU 1217C, 1520

- téléchargement de blocs sélectionnés à l'état MARCHE, 1409
- traitement des valeurs analogiques, 128, 310
- Exemples, commande de mouvement
 - caractéristiques de vitesse du référencement MC, 751
 - Commande d'axe avec le module TM Pulse, 724
 - comportement de l'axe, 786
 - Configuration de fréquence de sorties d'impulsions pour la CPU 1217C, 677
 - configuration d'une table de commande de mouvement d'un objet technologique, 730
 - Configurations de fréquence de sorties d'impulsions pour les CPU 1211C, CPU 1212C, CPU 1214C et CPU 1215C, 678
 - limitation d'à-coup, 753
- Exemples, communication
 - Adressage d'esclave AS-i, 1013
 - appareil partagé, 971
 - Communication CPU avec connexions d'émission et de réception distinctes, 855
 - Communication CPU avec une connexion d'émission et de réception commune, 856
 - Communication CPU par le biais de connexions TSEND_C ou TRCV_C, 857
 - configuration d'une liaison S7 PROFIBUS, 1033
 - configuration d'une liaison S7 PROFINET, 1031
 - I-device en tant que périphérique IO et contrôleur IO, 962
 - I-device partagé, 977
 - Protocoles de communication PROFINET, 851
 - T_CONFIG, modification des paramètres IP, 939
 - T_CONFIG, modification des paramètres IP des serveurs NTP, 942
 - T_CONFIG, modifications des paramètres IP et des noms des périphériques PROFINET IO, 941
 - telecontrol, 1365
- Exemples, communication PtP
 - Communication PtP d'héritage, RCV_CFG, 1288
 - condition de début de message, 1144
 - condition de fin de message, 1148
 - configuration, 1189
 - émulateur de terminal, 1188, 1197
 - exécuter l'exemple d'émulateur de terminal, 1198
 - longueur de message à l'intérieur du message, 1149
 - programmation STEP 7, 1196
 - Receive_Config, 1167
- Exemples, communication USS
 - compte rendu d'erreurs de communication d'héritage USS, 1314
 - Compte rendu d'erreurs de communication USS, 1216
- Exemples, divers
 - Ajout d'un type de données à une variable via AT, 141
 - fonctions de traçage et d'analyse logique, 1419
 - glisser-déplacer entre les éditeurs, 46
 - tranche de type de données de variable, 140
- Exemples, instructions
 - ATH (ASCII en hexadécimal), 370
 - bobines de temporisation, 248
 - CALCULATE, 42
 - CONTINUE, SCL, 336
 - CTRL_PWM, 509
 - DECO (Décoder), 341
 - Deserialize, 283
 - DeviceStates, PROFIBUS et PROFINET, 481
 - Exécution d'alarme cyclique SET_CINT et paramètre de temps, 434
 - EXIT, SCL, 337
 - GET_DIAG et modes, 497
 - GOTO (SCL), 338
 - HTA (Hexadécimal en ASCII), 370
 - LIMIT (Définir une limite), 275
 - ModuleStates, PROFIBUS et PROFINET, 487
 - RETURN, SCL, 338
 - ROR (Rotation à droite), SCL, 346
 - RUNTIME (Mesurer le temps d'exécution du programme), 329
 - S_CONV (Convertir la chaîne de caractères), 365
 - Serialize, 286
 - SHL (Décaler à gauche), SCL, 345
 - STRG_VAL (Convertir la chaîne en valeur numérique), 366
 - SWAP (Permuter octets), 289
 - TM_MAIL, 1376
 - VAL_STRG (Convertir la valeur numérique en chaîne de caractères), 367
 - Variations PEEK et POKE, 214, 293
- Exemples, instructions de chaîne à l'exécution
 - GetBlockName, 393
 - GetInstanceName, 388
 - GetInstancePath, 389
 - GetSymbolName, 382
 - GetSymbolPath, 385
- Exemples, Modbus
 - Demande d'écriture dans la mémoire image des sorties MB_CLIENT Modbus TCP, 1249

- Liaisons Modbus TCP multiples
 - MB_SERVER, 1246
 - MB_CLIENT Modbus TCP coordonnant plusieurs demandes, 1250
 - Modbus TCP, adressage de registre de maintien, 1242
 - Modbus TCP, exemples de paramètre
 - MB_HOLD_REG, 1239
 - Modbus TCP, paramètres de liaison
 - MB_CLIENT, 1231
 - Modbus TCP, paramètres de liaison
 - MB_SERVER, 1236
 - Plusieurs demandes MB_CLIENT avec des liaisons Modbus TCP différentes, 1248
 - Plusieurs demandes MB_CLIENT avec une liaison Modbus TCP commune, 1247
 - Programme esclave Modbus RTU, 1280
 - Programme maître Modbus RTU, 1277
 - Exemples, Modbus d'héritage
 - Modbus RTU d'héritage, adressage de registre de maintien, 1351
 - Modbus RTU d'héritage, exemples pour le paramètre MB_HOLD_REG, 1349
 - Exemples, Motion Control
 - comportement manuel à vue, 794
 - vitesse, 790
 - Exemples, PID
 - PID_3Step, paramètres de configuration, 644
 - PID_Compact, paramètres de configuration, 643
 - PID_Temp, paramètres de configuration, 646
 - Exemples, serveur web
 - alias, 1089
 - écriture de variables spéciales, 1088
 - Exemples, serveur Web
 - accès à partir d'un appareil mobile, 1046
 - alias, 1083
 - caractères spéciaux dans les commandes AWP, 1095
 - combinaison des déclarations AWP, 1093
 - DB de fragment, 1093
 - écriture de variables, 1084, 1108
 - écriture de variables spéciales, 1109
 - lecture de variables, 1083, 1106
 - lecture de variables spéciales, 1086
 - page Web personnalisée, 1103, 1109
 - page Web personnalisée pour changer de langues, 1116
 - programme STEP 7 pour vérifier les fragments, 1123
 - types enum, 1089, 1090, 1107
 - EXIT, SCL, 337
 - EXP (Créer valeur exponentielle), 275
 - EXPT (Elever à la puissance), 275
 - Extension des fonctionnalités du S7-1200, 31
- ## F
- F_TRIG (définir une variable sur le front descendant), 244
 - FAQ, 4
 - FB (bloc fonctionnel)
 - présentation, 85
 - FC (fonction), 85, 197
 - FieldRead (Lire champ), 300
 - FieldWrite (Ecrire champ), 300
 - File d'attente, 106
 - FILL_BLK (Compléter zone), 287
 - FIND (trouver caractères dans une chaîne de caractères), 378
 - FLOOR (Arrondir à l'entier inférieur), 307
 - Fonction (FC)
 - appel de blocs de code dans le programme utilisateur, 194
 - numéros de FC valides, 85
 - présentation, 85, 197
 - programmes linéaires et structurés, 192
 - protection du savoir-faire, 222
 - Fonctionnalité, I-device, 958
 - Fonctions mathématiques, 42, 268, 269
 - Fonctions, nouvelles, 35
 - FOR, SCL, 333
 - Forçage
 - des variables à partir du serveur Internet, 1068
 - table de visualisation, 1400
 - Forçage permanent, 1403
 - cycle, 1404
 - entrées et sorties, 1404
 - mémoire I, 1404
 - Mémoire I, 1403
 - périphérie d'entrée, 1403, 1404
 - table de visualisation, 1400
 - Formatage d'une carte mémoire, 1391
 - FRAC (Retourner le nombre de décimales), 275
 - Fréquence, bits de memento de cadence, 117
- ## G
- Gen_UsrMsg (Créer des alarmes de diagnostic utilisateur), 446
 - Génération de DB de pages Web personnalisées, 1098
 - GEO2LOG (Déterminer l'identificateur matériel sur la base des informations relatives à l'emplacement), 563
 - GEOADDR, 569

- Gestion des options (commande de configuration), 165
- Gestion du programme (SCL), 330
- CASE, 332
 - CONTINUE, 336
 - EXIT, 337
 - FOR, 333
 - GO TO, 338
 - IF-THEN, 331
 - REPEAT, 335
 - RETURN, 338
 - WHILE, 334
- GET (lire les données d'une CPU distante), 1021
- configuration de la liaison, 834
- GET_DIAG (lire l'information de diagnostic), 491
- GET_ERROR (Interrogation locale des erreurs), 323
- GET_ERROR_ID (Interrogation locale des ID d'erreur), 325
- Get_Features (Lire des fonctionnalités avancées), 1183
- Get_IM_Data (Lire les données d'identification et de maintenance), 462
- GetBlockName (Lire le nom du bloc), 390
- GetInstanceName (Lire le nom de l'instance de bloc), 386
- GetInstancePath (Interroger le nom global composite de l'instance de bloc), 388
- GETIO, 398
- GETIO_PART, 401
- GetStationInfo, 471
- GetSymbolName (Lecture d'un nom de variable au paramètre d'entrée), 379
- GetSymbolPath (Interroger un nom global composite de l'affectation de paramètre d'entrée), 383
- Glisser-déplacer entre les éditeurs, 46
- GOTO, SCL, 338
- GSD, fichier, 967
- Guillemets et apostrophes, conventions pour le serveur Web, 1094
- ## H
- Heure
- configuration de la CPU en ligne, 1386
 - DTL (type de données date et heure long), 134
 - RD_LOC_T (Lire l'heure locale), 351
 - SET_TIMEZONE (Définir fuseau horaire), 354
 - T_COMBINE (Combiner temps), 349
 - T_DIFF (Différence de temps), 349
 - TOD (type de données heure), 133
 - type de données Time, 133
 - WR_SYS_T (Régler l'heure), 351
- Heure d'été TimeTransformationRule, 353
- Heure locale
- RD_LOC_T (Lire l'heure locale), 351
 - WR_LOC_T (Régler l'heure locale), 351
- homologation ATEX, 1432
- Homologation CE, 1430
- homologation cULus, 1430
- Homologation de certification coréenne, 1432
- homologation FM, 1431
- Homologation pour le domaine maritime, 1433
- Homologations
- ATEX, 1432
 - Australie et Nouvelle-Zélande - Marque RCM, 1432
 - CE, 1430
 - Certification coréenne, 1432
 - cULus, 1430
 - FM, 1431
 - Maritime, 1433
- Horloge
- horloge temps réel, 119
 - RD_LOC_T (Lire l'heure locale), 351
 - RD_SYS_T (Lire l'heure), 351
 - WR_LOC_T (Régler l'heure locale), 351
 - WR_SYS_T (Régler l'heure), 351
- Horloge système
- RD_SYS_T (Lire l'heure), 351
 - WR_LOC_T (Régler l'heure locale), 351
 - WR_SYS_T (Régler l'heure), 351
- HSC (compteur rapide)
- configuration, 590
 - modes de comptage, 592
 - phase de fonctionnement, 593
- HTA (Convertir le nombre hexadécimal en chaîne de caractères ASCII), 369
- HTTP, connexions HTTP du serveur Web, 1125
- ## I
- Identification de la CPU, visualisation avec le serveur Web, 1056
- I-Device (intelligent IO device)
- configuration, 965
- I-device (périphérique IO intelligent)
- configurer à l'aide d'un fichier GSD, 967
 - fonctionnalité, 958
 - partagé, 977
 - propriétés, 959
 - système PN IO de niveau inférieur, 960
- I-device partagé, configuration, 977
- IF-THEN, SCL, 331
- Immunité aux pointes de tension, 1435
- Importation du certificat de sécurité Siemens, 1126
- IN_Range (valeur dans la page), 263

- INC (Incrémenter), 272
- Indexation de tableaux par des variables, 301
- Indicateur Premier cycle, 116
- Indice variable pour un tableau, 301
- Information sur le programme
 - Dans la structure d'appel, 234
- Informations pour nous contacter, 3, 177
- Informations sur les modules, page Web standard, 1060
- INSERT (insérer caractères dans une chaîne de caractères), 376
- Insertion de la carte mémoire dans la CPU, 143
- Insertion d'instructions
 - Favoris, 41
 - Glisser-déplacer, 41
- Insertion d'opérations
 - Glisser-déplacer entre les éditeurs, 46
- Insertion d'un appareil
 - CPU non spécifiée, 163
- Installation
 - bilan de consommation, 56
 - bornier de connexion, 69
 - câble d'extension, 70
 - carte SIM du TS Adapter, 74
 - charges de lampe, 80
 - charges inductives, 81
 - Communication Board (CB), 64
 - conseils, 53
 - conseils de câblage, 77, 79
 - conseils d'isolation, 78
 - CPU, 62
 - dégagement, 55
 - dimensions de montage, 59
 - écoulement d'air, 55
 - mise à la terre, 78
 - module de communication (CM), 68
 - module d'entrées-sorties (SM), 66
 - modules d'entrées-sorties (SM), 32
 - montage mural de l'adaptateur TS, 76
 - présentation, 53, 60
 - refroidissement, 55
 - Signal Board (SB), 64
 - TS Adapter et module TS, 72
 - TS Adapter sur un profilé support, 75
 - zone thermique, 55, 59
- Installation, conditions requises, 38
- Instantané des valeurs d'un DB, 1398
- Instruction Modbus_Comm_Load (Configurer SIPLUS I/O ou un port sur le module PtP pour Modbus RTU), 1254
- Instructions
 - (/)- (bobine à ouverture), 239
 - (RESET_BF) (mettre à 0 champ de bits), 241
 - (SET_BF) (mettre à 1 champ de bits), 241
 - |/|- contact normalement fermé (NF), 237
 - ||- contact normalement ouvert (NO), 237
 - |N|- (définir opérande sur front descendant), 243
 - |N|- (surveiller opérande de front descendant), 243
 - |P|- (définir opérande front montant), 243
 - |P|- (surveiller opérande de front montant), 243
 - ABS (Créer valeur absolue), 272
 - ACOS (Créer valeur arc cosinus), 275
 - ACT_TINT (activer alarme horaire), 440
 - activer sortie, 240
 - ADD (Addition), 269
 - ajouter des entrées ou sorties aux opérations CONT ou LOG, 44
 - ASIN (Créer valeur arc sinus), 275
 - ATAN (Créer valeur arc tangente), 275
 - ATH (Convertir la chaîne de caractères ASCII en nombre hexadécimal), 369
 - ATTACH (affecter un OB à un événement déclencheur d'alarme), 430
 - ATTR_DB (Lire les attributs d'un bloc de données), 559
 - boîte & (ET logique en LOG), 238
 - boîte /= (inverser affectation en LOG), 239
 - boîte = (affectation en LOG), 239
 - boîte >=1 (OU logique en LOG), 238
 - boîte x (OU exclusif en LOG), 238
 - CALCULATE, 42
 - CALCULER (calculer), 268
 - calendrier, 347
 - CAN_DINT (annuler une alarme temporisée), 442
 - CAN_DINT (interroger l'état de l'alarme temporisée), 442
 - CAN_TINT (annuler l'alarme horaire), 439
 - CASE (SCL), 332
 - CEIL (Arrondir à l'entier supérieur), 307
 - Chars_TO_Strg (Convertir array of CHAR en chaîne de caractères), 367
 - Codes d'état USS, 1215
 - codes d'état USS d'héritage, 1313
 - colonnes et entêtes, 45, 862, 876, 884, 895, 1202, 1224, 1253, 1301, 1316, 1336
 - comparer valeurs, 262
 - CONCAT (combinaison des chaînes de caractères), 372
 - CONTINUE (SCL), 336
 - CONV (Convertir valeur), 302
 - COS (Créer valeur cosinus), 275

- CountOfElements (Obtenir le nombre d'éléments du TABLEAU), 298
- CREATE_DB (Créer un bloc de données), 551
- CTD (Décomptage), 255
- CTRL_HSC (commande de compteurs rapides), 603
- CTRL_HSC_EXT (Commander le compteur rapide (avancé)), 574
- CTRL_PTO (Émettre un train d'impulsions à fréquence prédéfinie), 501
- CTRL_PWM (modulation de largeur d'impulsion), 499
- CTU (Comptage), 255
- CTUD (Comptage et décomptage), 255
- DataLogClose (fermer un journal), 536
- DataLogCreate (créer un journal), 525
- DataLogNewFile (journal dans un nouveau fichier), 538
- DataLogOpen (ouvrir un journal), 530
- DataLogWrite (écrire dans un journal), 532
date, 347
- DB_ANY_TO_VARIANT (Convertir DB_ANY en VARIANT), 312
- DEC (Décrémenter), 272
- DECO (Décoder), 340
- DELETE (supprimer caractères dans une chaîne de caractères), 375
- DELETE_DB (Supprimer le bloc de données), 561
- DEMUX (Démultiplexeur), 343
désactiver sortie, 240
- Deserialize, 281
- DETACH (dissocier un OB d'un événement déclencheur d'alarme), 430
- DeviceStates (lire l'état du module d'un système I/O), 479
- DIS_AIRT (désactiver les alarmes de priorité supérieure et les erreurs asynchrones), 444
- DIV (Division), 269
- DPNRM_DG (lire les données de diagnostic d'un esclave DP), 426
- DPRD_DAT (lire données cohérentes d'un esclave DP standard), 418
- DPWR_DAT (écrire données cohérentes d'un esclave DP standard), 418
- EN_AIRT (activer les alarmes de priorité supérieure et les erreurs asynchrones), 444
- ENCO (Encoder), 340
- ENDIS_PW (activer/désactiver les mots de passe), 319
- EQ_ElemType (Comparer le type de données d'un élément TABLEAU pour EGAL avec le type de données d'une variable), 265
- EQ_Type (Comparer le type de données pour EGAL avec le type de données d'une variable), 265
- ET (opération logique), 339
état, 1398
- EXIT (SCL), 337
- EXP (Créer valeur exponentielle), 275
- EXPT (Elever à la puissance), 275
- F_TRIG (définir une variable sur le front descendant), 244
- Favoris, 41
- FieldRead (Lire champ), 300
- FieldWrite (Ecrire champ), 300
- FILL_BLK (Compléter zone), 287
- FIND (trouver caractères dans une chaîne de caractères), 378
- FLOOR (Arrondir à l'entier inférieur), 307
- FOR (SCL), 333
forçage permanent, 1404
- FRAC (Retourner le nombre de décimales), 275
- Gen_UsrMsg (Créer des alarmes de diagnostic utilisateur), 446
- GEO2LOG (Déterminer l'identificateur matériel sur la base des informations relatives à l'emplacement), 563
- Gestion du programme (SCL), 330
- GET (lire les données d'une CPU distante), 1021
- GET_DIAG (lire l'information de diagnostic), 491
- GET_ERROR (Interrogation locale des erreurs), 323
- GET_ERROR_ID (Interrogation locale des ID d'erreur), 325
- Get_Features (Lire des fonctionnalités avancées), 1183
- Get_IM_Data (Lire les données d'identification et de maintenance), 462
- GetBlockName (Lire le nom du bloc), 390
- GetInstanceName (Lire le nom de l'instance de bloc), 386
- GetInstancePath (Interroger le nom global composite de l'instance de bloc), 388
- GetSymbolName (Lecture d'un nom de variable au paramètre d'entrée), 379
- GetSymbolPath (Interroger un nom global composite de l'affectation de paramètre d'entrée), 383
- Glisser-déplacer, 41
- Glisser-déplacer entre les éditeurs, 46
- GOTO (SCL), 338
- graduation valeurs analogiques, 43
- heure, 347
- horloge, 351

- HTA (Convertir le nombre hexadécimal en chaîne de caractères ASCII), 369
- IF-THEN (SCL), 331
- IN_Range (valeur dans la plage), 263
- INC (Incrémenter), 272
- INSERT (insérer caractères dans une chaîne de caractères), 376
- Insertion, 41
- Instruction RecipeExport (exportation de recette), 517
- instructions de conversion SCL, 303
- Instructions d'héritage TCON, TDISCON, TSEND, et TRCV, 896
- Instructions d'héritage TRCV_V (recevoir des données via Ethernet (TCP)), 877
- Instructions d'héritage TSEND_C (envoyer des données via Ethernet (TCP)), 877
- instructions extensibles, 44
- INV (Former le complément à 1), 339
- IO2MOD (Déterminer l'identification matérielle à partir d'une adresse E/S), 566
- IS_ARRAY (vérifier le TABLEAU), 267
- IS_NULL (Interrogation de pointeur EGAL A zéro), 266
- JMP (Saut si RLO = 1), 314
- JMP_LIST (Définir liste de sauts), 315
- JMPN (Saut si RLO = 0), 314
- Label (Repère de saut), 314
- LED (lire l'état de DEL), 461
- LEFT (lire les caractères de gauche d'une chaîne de caractères), 374
- LEN (Déterminer longueur de chaîne), 372
- LIMIT (Définir une limite), 274
- LN (Créer logarithme népérien), 275
- LOG2GEO (Déterminer l'emplacement à partir de l'identificateur matériel), 565
- LOWER_BOUND (Lire la limite inférieure d'un ARRAY), 290
- MAX (Calculer le maximum), 273
- MAX_LEN (longueur de chaîne maximale), 371
- MB_CLIENT, 1225
- MC_ChangeDynamic (modifier les réglages dynamiques pour l'axe), 779
- MC_CommandTable, 776
- MC_Halt (pause de l'axe), 763
- MC_Home (référencer l'axe), 760
- MC_MoveAbsolute (positionnement absolu de l'axe), 765
- MC_MoveJog (déplacer l'axe en mode Marche par à-coups), 773
- MC_MoveRelative (positionnement relatif de l'axe), 768
- MC_MoveVelocity (déplacer l'axe à une vitesse prédéfinie), 770
- MC_Power (libérer/bloquer l'axe), 756
- MC_ReadParam (lire les paramètres de l'objet technologique), 784
- MC_Reset (confirmer l'erreur), 759
- MC_WriteParam (écrire dans les paramètres d'un objet technologique), 782
- MID (lire les caractères du milieu d'une chaîne de caractères), 374
- MIN (Calculer le minimum), 273
- MOD (Calculer le reste de la division), 270
- Modbus_Comm_Load (Configurer SIPLUS I/O ou un port sur le module PtP pour Modbus RTU), 1254
- Modbus_Master (Communiquer à l'aide de SIPLUS I/O ou du port PtP en tant que maître Modbus RTU), 1259
- Modbus_Slave (Communiquer à l'aide de SIPLUS I/O ou du port PtP en tant qu'esclave Modbus RTU), 1266
- ModuleStates (lire les informations d'état d'un module), 485
- Motion Control, 754
- MOVE (Copier valeur), 277
- MOVE_BLK (Copier zone), 277
- MUL (Multiplication), 269
- MUX (Multiplexeur), 342
- N (surveiller opérande de front descendant), 243
- N = boîte et bobine N (définir opérande sur front descendant), 243
- N_TRIG (surveiller RLO pour front descendant), 244
- NE_ElemType (Comparer le type de données pour INEGAL avec le type de données d'une variable), 265
- NE_Type (Comparer le type de données pour INEGAL avec le type de données d'une variable), 265
- NEG (Créer le complément à deux), 271
- NORM_X (Normaliser), 308
- NOT (inverser RLO), 239
- NOT_NULL (Interrogation de pointeur DIFFERENT DE zéro), 266
- NOT_OK (contrôler invalidité), 264
- OK (Contrôler validité), 264
- OU (opération logique), 339
- OU EXCLUSIF (opération logique), 339
- OUT_Range (valeur en dehors de la plage), 263
- P (surveiller opérande de front montant), 243
- P = boîte et bobine P (définir opérande sur front montant), 243
- P_TRIG (surveiller RLO pour front montant), 244

- P3964_Config (Configurer le protocole 3964(R)), 1170
paramètres communs, 943
périphérie décentralisée AS-i, 394
périphérie décentralisée PROFIBUS, 394
périphérie décentralisée PROFINET, 394
PID_Compact (régulateur PID universel avec fonction d'optimisation intégrée), 611
PID_Temp (régulateur PID universel qui permet de gérer la régulation de la température), 629
Port_Config (configuration de port), 1158
PRVREC (Mettre l'enregistrement à disposition), 424
PUT (écrire les données sur une CPU distante), 1021
QRY_CINT (interroger les paramètres de l'alarme cyclique), 436
QRY_TINT (interroger l'état de l'alarme horaire), 441
R (mettre à 0 sortie), 240
R_TRIG (définir une variable sur le front montant), 244
RALRM (alarme de réception), 404
RCVREC (Recevoir l'enregistrement), 421
RD_ADDR (Déterminer les adresses IO à partir de l'identificateur matériel), 568
RD_LOC_T (Lire l'heure locale), 351
RD_SYS_T (Lire l'heure), 351
RDREC (lire l'enregistrement), 395
RE_TRIGR, 111
RE_TRIGR (Redéclencher le temps de surveillance du cycle), 321
READ_BIG (Lire des données au format big endian), 295
READ_DBL (lire dans un bloc de données dans la mémoire de chargement), 556
READ_LITTLE (Lire des données au format little endian), 295
Receive_Config (configuration de réception), 1164
Receive_P2P (Réception point à point), 1177
Receive_Reset (Réinitialiser récepteur), 1179
RecipelImport (importation de recette), 519
REPEAT (SCL), 335
REPLACE (remplacer caractères dans une chaîne de caractères), 377
RESET_BF (mettre à 0 champ de bits), 241
RET (Retour de saut), 318
RETURN (SCL), 338
RIGHT (lire les caractères de droite d'une chaîne de caractères), 374
ROL (Rotation à gauche) et ROR (Rotation à droite), 346
ROUND (Arrondir nombre), 306
RS (Bascules avec mise à 0/mise à 1), 241
RT (Réinitialiser temporisation), 246
RTM (Compteurs d'heures de fonctionnement), 355
RUNTIME (Mesurer le temps d'exécution du programme), 327
S (mettre à 1 sortie), 240
S_CONV (Convertir la chaîne de caractères), 358
S_MOVE (Déplacer la chaîne de caractères), 357
SCALE_X (Mettre à l'échelle), 308
SEL (Sélectionner), 341
Send_Config (Configuration d'émission), 1162
Send_P2P (Emission de données point à point), 1173
Serialize, 284
SET_BF (mettre à 1 champ de bits), 241
SET_CINT (définir les paramètres de l'alarme cyclique), 434
Set_Features (Définir des fonctionnalités avancées), 1184
SET_TIMEZONE (Sélectionner le fuseau horaire), 354
SET_TINTL (définir une date et une alarme horaire), 438
SGN_GET (Lire signaux RS232), 1180
SHL (Décaler à gauche) et SHR (Décaler à droite), 345
Signal_Set (Activer signaux RS232), 1181
SIN (Créer valeur sinus), 275
SQR (Créer carré), 275
SQRT (Créer racine carrée), 275
SR (Bascules avec mise à 1/mise à 0), 241
SRT_DINT (démarrer une alarme temporisée), 442
STP (Arrêter le programme), 323
Strg_TO_Chars (Convertir la chaîne de caractères en array of CHAR), 367
STRG_VAL (Convertir la chaîne de caractères en valeur numérique), 358
SUB (Soustraction), 269
SWAP (Permuter octets), 289
SWITCH (Branchement conditionnel), 316
T_ADD (Additionner les temps), 348
T_COMBINE (Combiner temps), 349
T_CONFIG (configurer interface), 931
T_CONV (Convertir et extraire les temps), 347
T_DIAG, 908
T_DIFF (Différence de temps), 349
T_RESET, 906
T_SUB (Soustraire les temps), 348
TAN (Créer valeur tangente), 275
TCON, 885
TDISCON, 885

- Temporisation, 246
- TM_MAIL (envoyer e-mail), 1371
- TOF (Temporisation "Retard à la retombée"), 246
- TON (Temporisation "Retard à la montée"), 246
- TONR (Temporisation "Retard à la montée mémorisé"), 246
- TP (Temporisation "Impulsion"), 246
- TRCV, 885
- TRCV_C, 863, 953
- TRUNC (Former un nombre entier), 306
- TSEND, 885
- TSEND_C, 863, 951
- TURCV (recevoir des données via Ethernet (UDP)), 924
- TUSEND (envoyer des données via Ethernet (UDP)), 924
- UFILL_BLK (Compléter zone contiguë), 287
- UMOVE_BLK (Copier zone contiguë), 277
- UPPER_BOUND (Lire la limite supérieure d'un ARRAY), 291
- USS_Drive_Control (Données permutées avec l'entraînement), 1208
- USS_Port_Scan (Editer communication via réseau USS), 1206
- USS_Read_Param (lire des paramètres de l'entraînement), 1211
- USS_Write_Param (modifier les paramètres dans l'entraînement), 1213
- VAL_STRG (Convertir la valeur numérique en chaîne de caractères), 358
- VARIANT_TO_DB_ANY (Convertir VARIANT en DB_ANY), 311
- VariantGet (Lire la valeur de variable VARIANT), 296
- VariantPut (Ecrire la valeur de variable VARIANT), 297
- Variations PEEK et POKE, 214, 293
- versions des
 - instructions, 45, 862, 876, 884, 895, 1202, 1224, 1253, 1301, 1316, 1336
 - visualisation, 1398
 - visualisation de l'état ou de la valeur, 1397
- WHILE (SCL), 334
- WR_LOC_T (Régler l'heure locale), 351
- WR_SYS_T (Régler l'heure), 351
- WRIT_DBL (écrire dans un bloc de données dans la mémoire de chargement), 556
- WRITE_BIG (Ecrire des données au format big endian), 295
- WRITE_LITTLE (Ecrire des données au format little endian), 295
- WRREC (écrire l'enregistrement), 395
- WWW (synchronisation des pages Web personnalisées), 1099
- Instructions d'héritage TCON, TDISCON, TSEND, et TRCV, 896
- Instructions d'héritage TRCV_V (recevoir des données via Ethernet (TCP)), 877
- Instructions d'héritage TSEND_C (envoyer des données via Ethernet (TCP)), 877
- Instructions extensibles, 44
- Instructions héritées
 - MB_CLIENT (communiquer via PROFINET en tant que client Modbus TCP), 1317
 - MB_COMM_LOAD (configurer un port sur le module PtP pour Modbus RTU), 1337
 - MB_MASTER (communiquer via le port PtP en tant que maître Modbus), 1341
 - MB_SERVER (communiquer via PROFINET en tant que serveur Modbus TCP), 1325
 - MB_SLAVE (communiquer via le port PtP en tant qu'esclave Modbus), 1347
 - PORT_CFG (configurer dynamiquement les paramètres de communication), 1282
 - RCV_CFG (configurer dynamiquement les paramètres de réception en série), 1286
 - RCV_PTP (activer les messages de réception), 1293
 - RCV_RST (supprimer le tampon de réception), 1296
 - SEND_CFG (configurer dynamiquement les paramètres de transmission en série), 1284
 - SEND_PTP (transmettre les données de tampon de réception), 1291
 - SGN_GET (activer signaux RS-232), 1298
 - SGN_GET (Interroger les signaux RS-232), 1297
 - USS_DRV (Échanger des données avec l'entraînement), 1306
 - USS_PORT (Éditer la communication via le réseau USS), 1305
 - USS_RPM (Lire des paramètres de l'entraînement), 1309
 - USS_WPM (Modifier des paramètres dans l'entraînement) hérité, 1311
- Instructions logiques sur bits
 - bobines à fermeture et bobines à ouverture, 239
 - contacts à fermeture et contacts à ouverture, 237
 - Instruction NOT (inverser RLO), 239
 - Instructions AND, OR et XOR, 238
 - instructions Front montant et Front descendant, 243
 - instructions Mise à 1 et Mise à 0, 240
- Instructions Motion Control, 754
- Instructions point à point, valeurs en retour, 1156

- Instructions PROFINET
 - Instructions d'héritage TCON, TDISCON, TSEND, et TRCV, 896
 - Instructions d'héritage TSEND_C (envoyer des données via Ethernet (TCP)), 877
 - T_CONFIG (configurer interface), 931
 - T_DIAG, 908
 - T_RESET, 906
 - TCON, 885
 - TDISCON, 885
 - TRCV, 885
 - TRCV_C, 953
 - TSEND, 885
 - TSEND_C, 863
 - TURCV (recevoir des données via Ethernet (UDP)), 924
 - TUSEND (envoyer des données via Ethernet (UDP)), 924
 - Instructions sur front, montant et descendant, 243
 - Instructions technologiques, 574, 603
 - Instructions trigonométriques, 275
 - Interface de bloc agrandie
 - chargement à l'état MARCHE, 1412
 - Interface utilisateur
 - Projet et vues de portail STEP 7, 39
 - Interfaces de communication
 - ajouter des modules, 164
 - CB 1241 RS485, 1634
 - CM 1241 RS232, 1635
 - Configuration, 1137
 - configuration des appareils, 159
 - DEL de signalisation, 1379
 - programmation, 1186
 - RS232 et RS485, 1129
 - tableau comparatif des modules, 31
 - Interfaces de communication, 3964(R), 1151
 - Intro, page Web standard, 1054
 - INV (Former le complément à 1), 339
 - IO2MOD (Déterminer l'identification matérielle à partir d'une adresse E/S), 566
 - IO-Link
 - brochage, 1579
 - configuration, 1581
 - diagnostic, 1586
 - diagramme, 1580
 - enregistrement, 1583
 - espace d'adressage, 1582
 - fonctions, 1577
 - messages d'erreur, 1582, 1584
 - Messages d'erreur, 1586
 - modification des paramètres lors de l'exécution, 1582
 - paramètres, 1581
 - profil d'appareil, 1576
 - rangement de l'appareil, 1578
 - remplacement, 1577
 - restauration des réglages d'usine, 1578
 - Signalisation par DEL, 1584
 - IS_ARRAY (vérifier le TABLEAU), 267
 - IS_NULL (Interrogation de pointeur EGAL A ZERO), 266
 - ISO sur TCP
 - configuration des liaisons, 833
 - ID de liaison, 854
 - mode ad hoc, 854
 - paramètres, 857
 - ISO sur TCP, protocole, 851
 - Isolation, conseils, 78
- ## J
- JavaScript, pages Web standard, 1125
 - JMP (Saut si RLO = 1), 314
 - JMP_LIST (Définir liste de sauts), 315
 - JMPN (Saut si RLO = 0), 314
 - Journal de données
 - DataLogClose (fermer un journal), 536
 - DataLogCreate (créer un journal), 525
 - DataLogNewFile (journal dans un nouveau fichier), 538
 - DataLogOpen (ouvrir un journal), 530
 - DataLogWrite (Écrire Data Log), 532
 - exemple de programme, 546
 - limite de taille et calcul de la taille, 542
 - présentation générale du journal de données, 524
 - structure des enregistrements de données, 524
 - Supprimer avec DataLogDelete, 537
 - Vider avec DataLogClear, 534
 - visualisation de journaux de données, 541
 - Journaux, page Web standard, 1076
- ## L
- Label (Repère de saut), 314
 - Langues pour les pages Web personnalisées, 1115
 - Lecture de variables HTTP, 1085
 - Lecture depuis les DB, les E/S ou la mémoire, 214, 293
 - LED (lire l'état de DEL), 461
 - LEFT (lire les caractères de gauche d'une chaîne de caractères), 374
 - LEN (Déterminer longueur de chaîne), 372
 - LENGTH, paramètre de SEND_P2P, 1176

Liaison active/passive, 833
 Liaison appareil local/partenaire, 833
 Liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 Liaison dynamique, 223
 Liaison réseau

- connexion d'appareils, 832
- plusieurs CPU, 948, 950, 954, 1006, 1011

 Liaisons

- configuration, 857
- ID de liaison, 854
- liaison S7, 1026
- partenaires, 833, 1028
- protocoles Ethernet, 1026
- Serveur Web, 1125
- types de communication, 825
- types de liaisons multinoeuds, 1026

 Liaisons multinoeuds

- protocoles Ethernet, 1026
- types de liaisons, 1026

 Ligne d'assistance, 3
 Ligne inactive, 1141, 1143
 LIMIT (Définir une limite), 274
 Limitation d'à-coup, 753
 LN (Créer logarithme népérien), 275
 LOG (logigramme), 207
 LOG2GEO (Déterminer l'emplacement à partir de l'identificateur matériel), 565
 Longueur de message maximale, 1147
 Longueur définie, 1147
 Longueur, message PtP, 1149
 LOWER_BOUND (Lire la limite inférieure d'un ARRAY), 290

M

Machines flexibles (commande de configuration), 165
 Manuels, 4
 Masque de sous-réseau, 840
 MAX (Calculer le maximum), 273
 MAX_LEN (longueur de chaîne maximale), 371
 MB_CLIENT, 1225
 MB_CLIENT (Communiquer comme client Modbus TCP via PROFINET) hérité, 1317
 MB_COMM_LOAD (Configurer le port sur le module PtP pour Modbus RTU) hérité, 1337
 MB_MASTER (Communiquer via le port PtP en tant que maître Modbus) hérité, 1341
 MB_SERVER, 1235
 MB_SERVER (Communiquer comme serveur Modbus TCP via PROFINET) hérité, 1325

MB_SLAVE (Communiquer via le port PtP en tant qu'esclave Modbus) hérité, 1347
 MC_ChangeDynamic (modifier les réglages dynamiques pour l'axe), 779
 MC_CommandTable, 776
 MC_Halt (pause de l'axe), 763
 MC_Home (référencer l'axe), 760
 MC_MoveAbsolute (positionnement absolu de l'axe), 765
 MC_MoveJog (déplacer l'axe en mode Marche par à-coups), 773
 MC_MoveRelative (positionnement relatif de l'axe), 768
 MC_MoveVelocity (déplacer l'axe à une vitesse prédéfinie), 770
 MC_Power (libérer/bloquer l'axe), 756
 MC_ReadParam (lire les paramètres de l'objet technologique), 784
 MC_Reset (confirmer l'erreur), 759
 MC_WriteParam (écrire dans les paramètres d'un objet technologique), 782

Mémoire

adresses de périphérie d'entrée (table de forçage permanent), 1403
 I (mémoire image des entrées), 123
 L (mémoire locale), 121
 M (mémentos), 125
 Mémento de cadence, 115
 Mémento système, 115
 Mémoire de chargement, 113
 Mémoire de travail, 113
 Mémoire rémanente, 113
 Mémoire temporaire, 125
 Q (mémoire image des sorties), 124
 visualisation de l'utilisation de la mémoire, 1392

Mémoire de chargement, 28

CPU 1211C, 1442
 CPU 1212C, 1456
 CPU 1214C, 1471
 CPU 1215C, 1487
 CPU 1217C, 1504
 pages Web personnalisées, 1102

Mémoire de travail, 28

CPU 1211C, 1442
 CPU 1212C, 1456
 CPU 1214C, 1471
 CPU 1215C, 1487
 CPU 1217C, 1504

Mémoire I

adresses de périphérie d'entrée (table de forçage permanent), 1403
 forçage permanent, 1403, 1404, 1404
 table de forçage permanent, 1403

- table de visualisation, 1397
- visualisation, 1397
- visualiser CONT, 1398
- Mémoire image
 - Écrire les sorties avec SETIO, 399
 - état, 1403
 - forçage permanent, 1403, 1404
 - Lire la zone de mémoire image avec GETIO_PART, 401
 - Lire les entrées avec GETIO, 398
 - Transférer la zone de mémoire image avec SETIO_PART, 403
 - visualisation de l'état ou de la valeur, 1397
- Mémoire locale
 - consommation par blocs, 125
 - max. par niveau de priorité de l'OB, 125
- Mémoire Q
 - configuration des voies d'impulsion, 507
 - sorties d'impulsions, 504
- Mémoire rémanente, 28, 113
 - CPU 1211C, 1442
 - CPU 1212C, 1456
 - CPU 1214C, 1471
 - CPU 1215C, 1487
 - CPU 1217C, 1504
- Mémoire temporaire
 - consommation par blocs, 125
 - max. par niveau de priorité de l'OB, 125
- Message
 - démarrer, 1143
 - fin, 1146
 - longueur, 1147
- Mesures, tâches de trace, 1419
- MID (lire les caractères du milieu d'une chaîne de caractères), 374
- MIN (Calculer le minimum), 273
- Mise à jour du firmware
 - à partir du serveur Internet, 1063, 1063
 - avec une carte mémoire, 153, 153
 - depuis STEP 7, 1389, 1389
- Mise à l'échelle de valeurs analogiques, 43
- Mise à l'échelle des valeurs analogiques, 310
- Mise à niveau d'une CPU V3.0 à V4.2.x, 1665
- Mise en service
 - Instruction PID_Temp, 663
 - Instructions PID_Compact et PID_3Step, 661
- MOD (Calculer le reste de la division), 270
- Modbus
 - Adresses de mémoire, 1222
 - Adresses de station de réseau, 1222
 - Codes de fonction, 1221
 - Communication RTU, 1222
 - MB_CLIENT (Communiquer comme client Modbus TCP via PROFINET) hérité, 1317
 - MB_COMM_LOAD (Configurer le port sur le module PtP pour Modbus RTU) hérité, 1337
 - MB_MASTER (Communiquer via le port PtP en tant que maître Modbus) hérité, 1341
 - MB_SERVER (Communiquer comme serveur Modbus TCP via PROFINET) hérité, 1325
 - MB_SLAVE (Communiquer via le port PtP en tant qu'esclave Modbus) hérité, 1347
 - Modbus_Comm_Load (Configurer SIPLUS I/O ou un port sur le module PtP pour Modbus RTU), 1254
 - Modbus_Master (Communiquer à l'aide de SIPLUS I/O ou du port PtP en tant que maître Modbus RTU), 1259
 - Modbus_Slave (Communiquer à l'aide de SIPLUS I/O ou du port PtP en tant qu'esclave Modbus RTU), 1266
 - versions, 45, 1202, 1253, 1301, 1336
- MODBUS
 - MB_CLIENT, 1225
 - MB_SERVER, 1235
- Modbus RTU
 - exemple d'esclave, 1280
 - programme maître, 1277
- Modbus TCP
 - versions, 1224, 1316
- Modbus_Master (Communiquer à l'aide de SIPLUS I/O ou du port PtP en tant que maître Modbus RTU), 1259
- Modbus_Slave (Communiquer à l'aide de SIPLUS I/O ou du port PtP en tant qu'esclave Modbus RTU), 1266
- Mode ad hoc, TCP et ISO sur TCP, 854
- Mode de fonctionnement
 - changement ARRET/MARCHE, 1392
 - états de fonctionnement de la CPU, 89
- Modes de comptage
 - compteur rapide, 592
- Modification
 - état de l'éditeur de programme, 1398
- Modification des paramètres de STEP 7, 45
- Module commutateur compact CSM 1277, 1643
- Module commutateur compact, CSM 1277, 1643
- Module CP
 - accès au serveur Web, 1047
 - Page de démarrage (Start Page) du serveur Web, 1054
- Module d'alimentation
 - PM1207, 1643
- Module d'alimentation PM 1207, 1643
- Module de communication (CM)
 - ajout d'un module CM 1243-5 (maître DP), 1005
 - ajout d'un module maître AS-i CM 1243-2, 1010

- ajouter des modules, 164
- besoins en courant, 1647
- CM 1241 RS232, 1635
- CM 1241 RS422/RS485, 1636
- configuration des appareils, 159
- configuration des paramètres, 185
- configuration pour l'exemple de programme point à point, 1189
- DEL de signalisation, 1129, 1379
- démontage, 68
- installation, 68
- présentation, 32
- programmation, 1186
- réception de données, 1177, 1293
- RS232 et RS485, 1129
- tableau comparatif, 31
- Module de potentiomètre
 - caractéristiques, 1641
- Module d'entrées-sorties Maître IO-Link, 1573
- Module technologique, SM 1278 4xIO-Link
- Master, 1573
- Modules
 - Communication Boards (CB), 32
 - configuration des paramètres, 185
 - module de communication (CM), 32
 - modules d'entrées-sorties (SM), 32
 - processeur de communication (CP), 32
 - Signal Board (SB), 32
 - tableau comparatif, 31
 - zone thermique, 55, 59
- Modules CANopen
 - 021620-B, 021630-B, 1644
- Modules de remplacement, 50
- Modules d'entrées/sorties TOR
 - SM 1221, 1521
 - SM 1222, 1523, 1525
 - SM 1223, 1531, 1538
- Modules d'entrées-sorties (SM)
 - ajouter des modules, 164
 - besoins en courant, 1647
 - câble d'extension, 70
 - configuration des paramètres, 185
 - démontage, 67
 - installation, 66
 - Maître SM 1278 4xIO-Link, 1573
 - présentation, 32
 - représentation des entrées analogiques (courant), 1555, 1607
 - représentation des entrées analogiques (tension), 1554, 1607
 - représentation des sorties analogiques (courant), 1557, 1609
 - représentation des sorties analogiques (tension), 1556, 1608
 - SM 1221 DI 16 x 24 V DC, 1521
 - SM 1221 DI 8 x 24 V DC, 1521
 - SM 1222 DQ 16 x 24 V DC, 1525
 - SM 1222 DQ 16 x Relais, 1525
 - SM 1222 DQ 8 Relais Inverseur, 1523
 - SM 1222 DQ 8 x 24 V DC, 1523
 - SM 1222 DQ 8 x Relais, 1523
 - SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC, 1531
 - SM 1223 DI 16 x 24 V DC, DQ 16 x Relais, 1531
 - SM 1223 DI 8 x 120/230 V AC/DQ 8 x Relais, 1538
 - SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC, 1531
 - SM 1223 DI 8 x 24 V DC, DQ 8 x Relais, 1531
 - SM 1231 AI 4 x 13 bits, 1541
 - SM 1231 AI 4 x 16 bits, 1541
 - SM 1231 AI 4 x 16 bits TC, 1558
 - SM 1231 AI 4 x RTD x 16 bits, 1566
 - SM 1231 AI 8 x 13 bits, 1541
 - SM 1231 AI 8 x 16 bits TC, 1558
 - SM 1231 AI 8 x RTD x 16 bits, 1566
 - SM 1232 AQ 2 x 14 bits, 1547
 - SM 1232 AQ 4 x 14 bits, 1547
 - SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits, 1550
 - temps de réponse indicielle, 1553
- Modules d'entrées-sorties analogiques
 - SM 1231, 1541
 - SM 1231 RTD, 1566
 - SM 1231 Thermocouple, 1558
 - SM 1232, 1547
 - SM 1234, 1550
- Modules retirés, 50
- ModuleStates, 485
- Montage
 - bornier de connexion, 69
 - câble d'extension, 70
 - charges de lampe, 80
 - charges inductives, 81
 - Communication Board (CB), 64
 - conseils, 53
 - conseils de câblage, 77, 79
 - CPU, 62
 - dégagement, 55
 - dimensions, 59
 - écoulement d'air, 55
 - isolation, 78
 - mise à la terre, 78
 - module de communication (CM), 68
 - module d'entrées-sorties (SM), 66
 - présentation, 60
 - refroidissement, 55

- Signal Board (SB), 64
 - zone thermique, 55, 59
 - Motion control
 - MC_Home (référencer l'axe), 760
 - MC_Power (libérer/bloquer l'axe), 756
 - Motion Control
 - Liste des valeurs ErrorID et ErrorInfo, 798
 - MC_ChangeDynamic (modifier les réglages dynamiques pour l'axe), 779
 - MC_CommandTable, 776
 - MC_Halt (pause de l'axe), 763
 - MC_MoveAbsolute (positionnement absolu de l'axe), 765
 - MC_MoveJog (déplacer l'axe en mode Marche par à-coups), 773
 - MC_MoveRelative (positionnement relatif de l'axe), 768
 - MC_MoveVelocity (déplacer l'axe à une vitesse prédéfinie), 770
 - MC_Reset (confirmer l'erreur), 759
 - Module TM Pulse, 724
 - Mouvement
 - Télégramme 4, 706
 - MOVE (Copier valeur), 277
 - MOVE_BLK (Copier zone), 277
 - MRES, panneau de commande, 47
 - MUL (Multiplication), 269
 - MUX (Multiplexeur), 342
- N**
- N (surveiller opérande de front descendant), 243
 - N = boîte et bobine N (définir opérande sur front descendant), 243
 - N_TRIG (surveiller RLO pour front descendant), 244
 - Navigateurs pris en charge par le serveur Web, 1038
 - NE_ElemType (Comparer le type de données pour INEGAL avec le type de données d'une variable), 265
 - NE_Type (Comparer le type de données pour INEGAL avec le type de données d'une variable), 265
 - NEG (Créer le complément à deux), 271
 - Niveau de pollution/catégorie de surtension, 1436
 - Niveau de protection
 - bloc de code, 222
 - CPU, 218
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 - Oubli du mot de passe, 157
 - Nombres
 - binaires, 131
 - entiers, 132
 - réels, 132
 - Noms d'appareils en ligne
 - PROFINET IO, 1384
 - NORM_X (Normaliser), 308
 - Normalisation des valeurs analogiques, 310
 - NOT (inverser RLO), 239
 - NOT_NULL (Interrogation de pointeur DIFFERENT DE ZERO), 266
 - NOT_OK (contrôler invalidité), 264
 - Nouvelles fonctions, 35
 - NTP (Network time protocol), 848
 - Numéros d'articles
 - borniers de connexion, 1657
 - butée, 1657
 - simulateurs, 1657
 - Numéros de port
 - affectation aux partenaires de communication, 851
 - limités, 944
- O**
- OB, (Bloc d'organisation)
 - OB d'alarme cyclique, 96
 - OB d'alarme de diagnostic, 99
 - OB d'alarme de processus, 97
 - OB d'alarme horaire, 103
 - OB d'alarme temporisée, 95
 - OB de cycle de programme, 94
 - OB de débrogage/enfichage de modules, 101
 - OB de défaillance du châssis ou de la station, 102
 - OB de démarrage, 95
 - OB de mise à jour, 104
 - OB de profil, 104
 - OB d'erreur de temps, 98
 - OB d'état, 103
 - OB MC-PostServo, 106
 - OB MC-PreServo, 105
 - Objets technologiques
 - commande de mouvement, 683
 - PID, 609
 - Octet de memento de cadence, 117
 - Octet de memento système, 116
 - OK (Contrôler validité), 264
 - OPC, configuration, 1366
 - Open User Communication, valeurs en retour des instructions, 944
 - Opérations PROFINET
 - Instructions d'héritage TRCV_V (recevoir des données via Ethernet (TCP)), 877
 - TRCV_C, 863
 - Optimisation de la vanne PID, 619
 - OU (opération logique), 339
 - OU EXCLUSIF (opération logique), 339

- Oubli du mot de passe, 157
- OUT_Range (valeur en dehors de la plage), 263
- Outils en ligne et de diagnostic
 - chargement dans la CPU à l'état MARCHE, 1406
- Ouverture/fermeture de session, pages Web standard, 1051

- P**
- P (surveiller opérande de front montant), 243
- P = boîte et bobine P (définir opérande sur front montant), 243
- P_TRIG (surveiller RLO pour front montant), 244
- P3964_Config (Configurer le protocole 3964(R)), 1170
 - erreurs, 1172
- pages HTML
 - Listage, exemple de pages Web personnalisées, 1109
 - personnalisées, 1079
- Pages HTML personnalisées
 - accès aux données du S7-1200, 1081
 - actualisation, 1080
 - conception, 1080
 - emplacements, 1097
 - emplacements des langues, 1119
- Pages Web
 - Service, assistance et documentation STEP 7, 4
- Pages Web personnalisées, 1038, 1079
 - accès depuis le PC, 1101
 - activation de l'instruction WWW, 1099
 - activation et désactivation à partir du DB de commande, 1119
 - actualisation, 1080
 - chargement des DB correspondants dans la CPU, 1101
 - commande de DB de fragment manuels, 1119
 - commandes AWP pour accéder aux données du S7-1200, 1081
 - configuration, 1097
 - configuration multilingue, 1119
 - contraintes de mémoire de chargement, 1102
 - création avec un éditeur HTML, 1080
 - création de fragments, 1091
 - écriture de variables, 1083
 - écriture de variables spéciales, 1087
 - exemple, 1103
 - génération de blocs de programme, 1098
 - gestion des caractères spéciaux, 1094
 - importation de fragments, 1092
 - lecture de variables, 1082
 - lecture de variables spéciales, 1085
 - listage HTML, 1109
 - plusieurs langues, 1115
 - programmation dans STEP 7, 1099
 - suppression de blocs de programme, 1098
- Pages Web standard, 1037
 - accès depuis le PC, 1044
 - accès sécurisé, 1045
 - changement de l'état de fonctionnement, 1054
 - communication, 1064
 - Démarrage, 1054
 - Diagnostic, 1056, 1059
 - disposition, 1048
 - État des variables, 1068
 - Informations sur les modules, 1060
 - Intro, 1054
 - JavaScript, 1125
 - Journaux, 1076
 - ouverture et fermeture de session, 1051
 - restrictions de cookies, 1125
- Pages Web STEP 7, 4
- Paramétrage, 197
- Paramètres, 45
- Paramètres de mise en route, 147
- Paramètres de sortie, 197
 - configuration des voies d'impulsion, 507
 - sorties d'impulsions, 504
- Parité, 1138
- Pas de redémarrage, 89
- Passage de MARCHE à ARRET, 120
- Pause, 1141, 1143
- PEEK, PEEK_WORD, PEEK_BOOL, PEEK_DWORD, PEEK_BLK, 214, 293
- Périphériques PROFINET IO
 - Écrire toutes les sorties avec SETIO, 399
 - Écrire une partie des sorties avec SETIO_PART, 403
 - Lire une partie des entrées avec GETIO_PART, 401
- Perte de communication de la CPU avec les modules, 101
- Phasage, 680
- Phase de fonctionnement
 - HSC (compteur rapide), 593
- PID
 - algorithme PID_3Step, 608
 - algorithme PID_Compact, 608
 - Limites de la mesure PID_Compact, 615
 - Paramètres ErrorBit de l'instruction PID_3Step, 626
 - Paramètres ErrorBit de l'instruction PID_Compact, 616
 - Paramètres ErrorBit de l'instruction PID_Temp, 640
 - PID_3STEP (Régulateur PID avec fonction d'optimisation pour vannes), 619

- PID_Compact (régulateur PID universel avec fonction d'optimisation intégrée), 611
- PID_Temp (régulateur PID universel qui permet de gérer la régulation de la température), 629
- présentation, 608
- régulateurs en cascade, 636
- Pointeurs
 - type de données Variant, 139
- POKE, POKE_BOOL, POKE_BLK, 214, 293
- Port anneau, 995
- Port PROFINET
 - autonégociation, 843
- Port, configuration 3964(R), 1151
- PORT_CFG (Configurer des paramètres de communication dynamiquement) hérité, 1282
- Port_Config (configuration de port), 1158
- Position de caractère, longueur du message, 1149
- Priorité
 - classe de priorité, 94
 - priorité dans le traitement, 106
- Processeur de communication (CP)
 - ajouter des modules, 164
 - configuration des appareils, 159
 - configuration des paramètres, 185
 - présentation, 32
 - tableau comparatif, 31
- PROFIBUS
 - adresse, 1007
 - adresse, configuration, 1007
 - ajout d'un esclave DP, 1005
 - ajout d'un module CM 1243-5 (maître DP), 1005
 - DPNRM_DG (lire les données de diagnostic d'un esclave DP), 426
 - DPRD_DAT (lire données cohérentes d'un esclave DP standard), 418
 - DPWR_DAT (écrire données cohérentes d'un esclave DP standard), 418
 - esclave, 1002
 - GET (lire les données d'une CPU distante), 1021
 - instructions de périphérie décentralisée, 394
 - liaison réseau, 832, 1006
 - liaison S7, 1026
 - maître, 1002
 - module CM 1242-5 (esclave DP), 1002
 - module CM 1243-5 (maître DP), 1002
 - nombre de liaisons de communication, 828
 - PUT (écrire les données sur une CPU distante), 1021
 - RALRM (alarme de réception), 404
 - RDREC (lire l'enregistrement), 395
 - WRREC (écrire l'enregistrement), 395
- PROFIBUS et PROFINET
 - Exemple DeviceStates, 481
 - Exemple ModuleStates, 487
- PROFIdrive, 694
- PROFIenergy, 429
- Profilé support, 60
- PROFINET
 - adresse IP, 840
 - adresse MAC, 840
 - affectation d'adresse IP, 850
 - communication API-API, 949
 - communication CPU à CPU, 949
 - configuration de la communication entre CPU et IHM, 947
 - configuration de l'adresse IP, 177
 - DPRD_DAT (lire données cohérentes d'un esclave DP standard), 418
 - DPWR_DAT (écrire données cohérentes d'un esclave DP standard), 418
 - GET (lire les données d'une CPU distante), 1021
 - ID de liaison, 854
 - instructions de périphérie décentralisée, 394
 - liaison réseau, 832, 948, 950, 954
 - liaison S7, 1026
 - mode ad hoc, 854
 - nom et adresse des appareils, 850
 - nombre de liaisons de communication, 828
 - présentation, 851
 - propriété Synchronisation de l'heure, 849
 - propriétés de l'adresse Ethernet, 841
 - PRVREC (Mettre l'enregistrement à disposition), 424
 - PUT (écrire les données sur une CPU distante), 1021
 - RALRM (alarme de réception), 404
 - RCVREC (Recevoir l'enregistrement), 421
 - RDREC (lire l'enregistrement), 395
 - réinitialisation d'une liaison, 906
 - synchronisation d'horloge, 177
 - temps de démarrage système, 850
 - Test d'un réseau, 845
 - types de communication, 825
 - WRREC (écrire l'enregistrement), 395
- PROFINET IO
 - affectation de noms d'appareils, 955
 - affectation de noms d'appareils en ligne, 1384
 - affectation d'une CPU, 955
 - ajout d'un appareil, 954
 - appareils, 954
 - noms d'appareils en ligne, 1384
 - noms de l'appareil, 955
- PROFINET RT, 851

- Profondeur d'imbrication, 85
- Programmation
 - ajouter des entrées ou sorties aux opérations CONT ou LOG, 44
 - algorithme PID_3Step, 608
 - algorithme PID_Compact, 608
 - appel de blocs de code dans le programme utilisateur, 194
 - appels de blocs, 85
 - bloc de données (DB), 85
 - bloc de données d'instance (DB), 197
 - bloc fonctionnel (FB), 85, 197
 - classe de priorité, 94
 - comparaison et synchronisation des blocs de code, 1394
 - CONT (schéma à contacts), 206
 - CPU non spécifiée, 163
 - états de fonctionnement de la CPU, 89
 - Favoris, 41
 - flux de courant (EN et ENO), 216
 - fonction (FC), 197
 - Glisser-déplacer entre les éditeurs, 46
 - heure système, 351
 - Insertion d'opérations, 41
 - instructions extensibles, 44
 - Instructions point à point, 1186
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 - LOG (logigramme), 207
 - modules retirés, 50
 - numéros de FC, FB et DB valides, 85
 - PID_3STEP (Régulateur PID avec fonction d'optimisation pour vannes), 619
 - PID_Compact (régulateur PID universel avec fonction d'optimisation intégrée), 611
 - PID_Temp (régulateur PID universel qui permet de gérer la régulation de la température), 629
 - présentation de PID, 608
 - programme linéaire, 192
 - programme structuré, 192
 - RTM (Compteurs d'heures de fonctionnement), 355
 - SCL (Structured Control Language), 208, 208, 210
 - types de blocs de code, 85
 - valeur initiale d'un FB, 197
- Programmation du changement de langue d'une page Web personnalisée, 1115
- Programmation linéaire, 192
- Programmation point à point, 1186
- Programmation STEP 7
 - exemple de programme point à point, 1196
 - pages Web personnalisées, 1099
- Programmation structurée, structure de bloc, 192
- Programme
 - appel de blocs de code dans le programme utilisateur, 194
 - Blocs d'organisation (OB), 195
 - carte mémoire, 143
 - chargement dans la CPU, 225
 - classe de priorité, 94
 - copie des blocs depuis une CPU en ligne, 231
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 - programmes linéaires et structurés, 192
 - protection par mot de passe, 222
- Programme utilisateur
 - ajouter des entrées ou sorties aux opérations CONT ou LOG, 44
 - appel de blocs de code dans le programme utilisateur, 194
 - Blocs d'organisation (OB), 195
 - carte mémoire, 143
 - carte programme, 143
 - carte transfert, 143
 - chargement dans la CPU, 225
 - copie des blocs depuis une CPU en ligne, 231
 - Favoris, 41
 - Glisser-déplacer entre les éditeurs, 46
 - Insertion d'instructions, 41
 - instructions extensibles, 44
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 - programmes linéaires et structurés, 192
 - protection par mot de passe, 222
- Projet
 - carte programme, 151
 - carte transfert, 148
 - carte transfert vide, 157
 - chargement dans la CPU, 225
 - comparaison et synchronisation, 1394
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 - oubli du mot de passe, 157
 - protection d'accès par mots de passe, 218
 - protéger un bloc de code, 222
- Projets de machine standard (commande de configuration), 165
- Propriété Synchronisation de l'heure, 849
- Propriétés de la CPU, pages Web personnalisées
 - Configuration dans STEP 7, 1097
 - configuration de plusieurs langues, 1119
- Protection contre la copie
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
- Protection contre la tension inverse, 1437

- Protection d'accès, CPU, 218
 - Protection du savoir-faire
 - protection par mot de passe, 222
 - Protection du savoir-faire, visualisation avec le serveur Web, 1056
 - Protection par mot de passe
 - accès à la CPU, 218
 - bloc de code, 222
 - carte transfert vide, 157
 - ENDIS_PW (activer/désactiver les mots de passe), 319
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 - Oubli du mot de passe, 157
 - protection contre la copie, 223
 - Protocole
 - Communication, 1132
 - Communication programmable, 1132
 - ISO sur TCP, 851
 - Modbus, 1132
 - PROFINET RT, 851
 - TCP, 851
 - UDP, 851
 - USS, 1132
 - Protocoles Ethernet, 851
 - liaisons multinoeuds, 1026
 - Protocoles, communication, 853
 - PRVREC (Mettre l'enregistrement à disposition), 424
 - PTO, 684
 - PTO (Sortie de trains d'impulsions)
 - configuration des voies d'impulsion, 507
 - CTRL_PTO (Émettre un train d'impulsions à fréquence prédéfinie), 501
 - CTRL_PWM (modulation de largeur d'impulsion), 499
 - fonctionnement, 504
 - forçage permanent impossible, 1405
 - Pupitre opérateur, 47
 - Pupitres (HMI), 33
 - Pupitres opérateur, 33
 - PUT (écrire les données sur une CPU distante), 1021
 - configuration de la liaison, 834
 - PWM (modulation de largeur d'impulsion)
 - Durée d'impulsion, 508
 - Temps de cycle, 508
 - PWM (Modulation de largeur d'impulsion)
 - Adresses E/S, 511
 - configuration des voies d'impulsion, 507
 - CTRL_PTO (Émettre un train d'impulsions à fréquence prédéfinie), 501
 - CTRL_PWM (modulation de largeur d'impulsion), 499
 - fonctionnement, 504
 - forçage permanent impossible, 1405
 - modification de la durée d'impulsion, 511
 - modification du temps de cycle, 511
- Q**
- QRY_CINT (interroger les paramètres de l'alarme cyclique), 436
 - QRY_TINT (interroger l'état de l'alarme horaire), 441
- R**
- R (mettre à 0 sortie), 240
 - R_TRIG (définir une variable sur le front montant), 244
 - RALRM (alarme de réception), 404, 413
 - RCV_CFG (Configurer dynamiquement les paramètres de réception série) hérité, 1286
 - RCV_PTP (Valider la réception des messages) hérité, 1293
 - RCV_RST (Effacer le tampon de réception) hérité, 1296
 - RCVREC (Recevoir l'enregistrement), 421
 - RD_ADDR (Déterminer les adresses IO à partir de l'identificateur matériel), 568
 - RD_LOC_T (Lire l'heure locale), 351
 - RD_SINFO (Lire l'information de déclenchement de l'OB actuel), 450
 - RD_SYS_T (Lire l'heure), 351
 - RDREC (lire l'enregistrement), 395, 413
 - RE_TRIGR (Redéclencher le temps de surveillance du cycle), 321
 - READ_BIG (Lire des données au format big endian), 295
 - READ_DBL (lire dans un bloc de données dans la mémoire de chargement), 556
 - READ_LITTLE (Lire des données au format little endian), 295
 - Receive_Config (configuration de réception), 1164
 - Receive_P2P (Réception point à point), 1177
 - Receive_Reset (Réinitialiser récepteur), 1179
 - Réception de messages, configuration
 - configuration de l'appareil PtP, 1142
 - exemple de programme point à point, 1190
 - Réception, configuration des paramètres, 954
 - Recette
 - exemple de programme, 521
 - Instruction RecipeExport (exportation de recette), 517
 - présentation, 512

- RecipelImport (importation de recette), 519
 - Structure de DB, 513
 - Redondance
 - Clients de redondance, 987
 - Domaines de redondance, 990
 - Redondance des supports
 - Configuration, 993
 - Fonctions dans une topologie en anneau, 988
 - Référencement de types Enum, pages Web personnalisées, 1090
 - Références
 - alimentation PM 1207, 1655
 - câbles d'extension, 1657
 - cartes mémoire, 1656
 - commutateur Ethernet CSM 1277, 1655
 - Connecteurs et connexions par borne, 1654
 - CPU, 1651
 - CPU 1214FC, CPU 1215FC, 1655
 - Interfaces de communication (CM, CB et CP), 1653, 1653, 1654, 1654, 1654
 - logiciel de programmation, 1664
 - logiciel de visualisation, 1664
 - Modules d'entrées-sorties, 1652
 - Modules d'entrées-sorties FS, 1655
 - pupitres HMI Basic Panels, 1656
 - Signal Boards, Battery Boards, 1653
 - STEP 7, 1664
 - WinCC, 1664
 - Refroidissement, 55
 - Régulateurs PID en cascade, 636
 - Réinitialisation des valeurs d'un DB, 1398
 - Réinitialiser temporisation (RT), 246
 - Remplacement de l'appareil
 - CPU V3.0 par une CPU V4.2.x, 1665
 - Marche à suivre, 177
 - Remplacement d'une CPU V3.0 par une CPU V4.2.x, 1665
 - REPEAT, SCL, 335
 - Répertoires, langues pour les pages Web personnalisées, 1115
 - REPLACE (remplacer caractères dans une chaîne de caractères), 377
 - RESET_BF (mettre à 0 champ de bits), 241
 - Restauration des réglages d'usine, 1387, 1387
 - Restauration d'une sauvegarde, 1428
 - Restrictions cookies, pages Web standard, 1125
 - RET (Retour de saut), 318
 - Retard RTS activé, désactivé, 1141
 - RETURN, SCL, 338
 - RIGHT (lire les caractères de droite d'une chaîne de caractères), 374
 - ROL (Rotation à gauche) et ROR (Rotation à droite), 346
 - ROUND (Arrondir nombre), 306
 - Routage S7, 996
 - Routeur IP, 840
 - RS (Bascules avec mise à 0/mise à 1), 241
 - RS232 et RS485, modules de communication, 1129
 - RT (Réinitialiser temporisation), 246
 - RTS (contrôle de flux matériel, PtP), 1139
 - RUNTIME (Mesurer le temps d'exécution du programme), 327
- ## S
- S (mettre à 1 sortie), 240
 - S_CONV (Convertir la chaîne de caractères), 358
 - S_MOVE (Déplacer la chaîne de caractères), 357
 - Sauvegarde d'une CPU, 1425
 - SCALE_X (Mettre à l'échelle), 308
 - Schémas de câblage
 - CB 1241 RS 485, 1634
 - CPU 1211C, 1452
 - CPU 1212C, 1467
 - CPU 1214C, 1481
 - CPU 1215C, 1498
 - CPU 1217C, 1517
 - SB 1221, 1589
 - SB 1222, 1592
 - SB 1223, 1596, 1600
 - SB 1231, 1603
 - SB 1231 RTD, 1617
 - SB 1231 thermocouple, 1614
 - SB 1232, 1605
 - SM 1221, 1522
 - SM 1222, 1526
 - SM 1223, 1534, 1541
 - SM 1231, 1544
 - SM 1231 RTD, 1568
 - SM 1231 thermocouple, 1560
 - SM 1232, 1548
 - SM 1234, 1552
 - SM 1278 IO-Link maître, 1575
 - SCL (Structured Control Language)
 - adressage, 210
 - appel de blocs, 194
 - appel d'un FB ou d'une FC, 210
 - comparer valeurs, 262
 - conditions, 210
 - éditeur de programmes, 208
 - EN et ENO (flux de courant), 216
 - expressions, 210
 - gestion du programme, 330

- instructions de contrôle, 210, 330
- instructions de conversion, 303
- instructions logiques sur bits, 237
- opérateurs, 210
- présentation, 208
- priorité des opérateurs, 210
- section Var, 208
- temporisations, 246
- Sécurité
 - liaison avec une CPU, une carte mémoire ou un mot de passe, 223
 - oubli du mot de passe, 157
 - protection contre la copie, 223
 - protection d'accès de la CPU, 218
 - protection du savoir-faire pour un bloc de code, 222
- SEL (Sélectionner), 341
- SEND_CFG (Configurer dynamiquement les paramètres de transmission série) hérité, 1284
- Send_Config (Configuration d'émission), 1162
- Send_P2P (Emission de données point à point), 1173
- SEND_P2P (émission de données point à point)
 - paramètres LENGTH et BUFFER, 1176
- SEND_PTP (Envoyer les données du tampon d'émission) hérité, 1291
- Séquence de caractères
 - début de message, 1143
 - Fin de message, 1147
- Séquence de mouvement (MC_CommandTable), 776
- Serialize, 284
- Serveur Web
 - accès à travers un module CP, 1047
 - accès avec un dispositif mobile, 1046
 - activation, 1040
 - apparence sur un dispositif mobile, 1049
 - configuration utilisateur, 1042
 - connexions HTTP maximum, 1125
 - contraintes, 1124
 - conventions pour les guillemets et apostrophes, 1094
 - pages Web personnalisées, 1079
 - pages Web standard, 1044
 - prise en charge des navigateurs, 1038
 - vitesse d'actualisation, 1040
- Service et assistance, 3
- SET_BF (mettre à 1 champ de bits), 241
- SET_CINT (définir les paramètres de l'alarme cyclique), 434
- Set_Features (Définir des fonctionnalités avancées), 1184
- SET_TIMEZONE (Définir fuseau horaire), 354
- SET_TINTL (définir une date et une alarme horaire), 438
- SETIO, 399
- SETIO_PART, 403
- SGN_GET (Interroger les signaux RS-232) hérité, 1297
- SGN_GET (Lire signaux RS232), 1180
- SGN_SET (Régler les signaux RS-232) hérité, 1298
- SHL (Décaler à gauche) et SHR (Décaler à droite), 345
- siemens_automation_language, cookie, 1115
- Signal Boards (SB)
 - ajouter des modules, 164
 - besoins en courant, 1647
 - configuration des paramètres, 185
 - démontage, 64
 - installation, 64
 - présentation, 32
 - représentation des entrées (courant), 1555, 1607
 - représentation des entrées (tension), 1554, 1607
 - représentation des sorties analogiques (courant), 1557, 1609
 - représentation des sorties analogiques (tension), 1556, 1608
 - SB 1221 DI 4 x 24 V DC, 200 kHz, 1587
 - SB 1221 DI 4 x 5 V DC, 200 kHz, 1587
 - SB 1222 DQ 4 x 24 V DC, 200 kHz, 1590
 - SB 1222 DQ 4 x 5 V DC, 200 kHz, 1590
 - SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC, 1598
 - SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz, 1593
 - SB 1223 DI 2 x 5 V DC/DQ 2 x 5 V DC, 200 kHz, 1593
 - SB 1231 AI 1 x 12 bits, 1601
 - SB 1231 AI 1 x 16 bits RTD, 1615
 - SB 1231 AI 1 x 16 bits Thermocouple, 1610
 - SB 1232 AQ 1 x 12 bits, 1604
- Signal Boards analogiques
 - SB 1231, 1601
 - SB 1231 RTD, 1615
 - SB 1231 Thermocouple, 1610
 - SB 1232, 1604
- Signal Boards TOR
 - SB 1221, 1587
 - SB 1222, 1590
 - SB 1223, 1593, 1598
- Signal_Set (Activer signaux RS232), 1181
- Simulateurs, 1639
- Simulateurs d'entrées, 1639
- SIN (Créer valeur sinus), 275
- SM 1231 RTD
 - tableaux de sélection, 1570, 1618
- SM et SB
 - configuration des appareils, 159, 159
 - tableau comparatif, 31
- Smart phone, accès au serveur Web, 1046

- SMS, 1365
 - Sorties d'impulsions, 504
 - Sources d'informations, 4
 - SQR (Créer carré), 275
 - SQRT (Créer racine carrée), 275
 - SR (Bascules avec mise à 1/mise à 0), 241
 - SRT_DINT (démarrer une alarme temporisée), 442
 - Station
 - Lire des informations avec GetStationInfo, 471
 - STEP 7
 - affectation d'une adresse IP à une CPU en ligne, 838
 - ajout d'un périphérique PROFINET IO, 954
 - ajouter des entrées ou sorties aux opérations CONT ou LOG, 44
 - ajouter des modules, 164
 - ajouter un appareil, 160
 - appel de blocs de code dans le programme utilisateur, 194
 - appels de blocs, 85
 - AS-i, 1012
 - bloc de données (DB), 85
 - bloc de données d'instance (DB), 197
 - bloc fonctionnel (FB), 85, 197
 - boutons MARCHE/ARRET, 47
 - carte programme, 143
 - charge due à la communication, 112
 - chargement dans la CPU, 225
 - classe de priorité (OB), 94
 - comparaison et synchronisation, 1394
 - compatibilité des versions, 51
 - configuration de la CPU, 177
 - configuration des appareils, 159
 - configuration des modules, 185
 - Configuration HSC, 590
 - copie des blocs depuis une CPU en ligne, 231
 - entrées et sorties affichées ou masquées, 44
 - états de fonctionnement, 89
 - Favoris, 41
 - fonction (FC), 197
 - fonctionnement, 1400
 - forçage permanent, 1403, 1404
 - Glisser-déplacer entre les éditeurs, 46
 - Insertion d'opérations, 41
 - liaison réseau, 832
 - Modification des paramètres, 45
 - modules retirés, 50
 - numéros de FC, FB et DB valides, 85
 - port AS-i, 1011
 - Port Ethernet, 840
 - port PROFINET, 840
 - PROFIBUS, 1006
 - programmes linéaires et structurés, 192
 - propriété Synchronisation de l'heure (PROFINET), 849
 - protection par mot de passe, 222
 - pupitre opérateur, 47
 - RTM (Compteurs d'heures de fonctionnement), 355
 - temps de cycle, 111, 112
 - temps de cycle, 111, 112
 - traitement de mise en route, 92
 - types de blocs de code, 85
 - valeur initiale d'un FB, 197
 - visualisation, 1397, 1398
 - Vue du portail et vue du projet, 39
 - STP (Arrêter le programme), 323
 - Strg_TO_Chars (Convertir la chaîne de caractères en array of CHAR), 367
 - STRG_VAL (Convertir la chaîne de caractères en valeur numérique), 358
 - String
 - présentation des données chaînes de caractères, 357
 - Structure d'appel, 234
 - Structure du programme, 194
 - SUB (Soustraction), 269
 - SWAP (Permuter octets), 289
 - SWITCH (Branchement conditionnel), 316
 - Synchronisation
 - propriété Synchronisation de l'heure (PROFINET), 849
 - Synchronisation de l'heure, 189
 - Synchronisation des valeurs initiales des blocs de données, 229
 - Système IO, échange de données, 962
- ## T
- T_ADD (Additionner les temps), 348
 - T_COMBINE (Combiner temps), 349
 - T_CONFIG (configurer interface), 931
 - T_CONV (Convertir et extraire les temps), 347
 - T_DIAG, 908
 - T_DIFF (Différence de temps), 349
 - T_RESET, 906
 - T_SUB (Soustraire les temps), 348
 - Table de forçage permanent
 - adressage de la périphérie d'entrée, 1403
 - forçage permanent, 1403, 1404
 - Table de visualisation
 - déblocage des sorties à l'état ARRET, 1402
 - fonctionnement, 1400
 - forçage permanent, 232

- types de déclenchement, 1401
- visualisation, 1397
- Tableau comparatif
 - Modèles de CPU, 28
 - modules, 31
- Tableau de comparaison
 - Appareils IHM, 33
- Tableau, accès aux éléments, 301
- Tablette, accès au serveur Web, 1046
- TAN (Créer valeur tangente), 275
- Task cards
 - colonnes et en-têtes, 862, 876, 884, 895
- Task Cards
 - colonnes et en-têtes, 45, 1202, 1224, 1253, 1301, 1316, 1336
- TCON, 885
 - configuration, 833
 - ID de liaison, 854
 - paramètres de liaison, 857
- TCON, TDISCON, TSEND, et TRCV
 - versions, 884, 895
- TCON_Param, 857
- TCP
 - configuration des liaisons, 833, 833
 - ID de liaison, 854
 - mode ad hoc, 854
 - paramètres, 857
 - protocole, 851
- TDISCON, 885
- Téléchargement
 - certificat de sécurité Siemens pour le PC, 1054, 1126
 - mise à jour du firmware, 153
- Telecontrol, 1362
- TeleControl
 - processeurs de communication, 1358
- TeleService via GPRS, 1362
- Temporisation "Impulsion" (TP), 246
- Temporisation "Retard à la montée mémorisé" (TONR), 246
- Temporisation "Retard à la montée" (TON), 246
- Temporisation "Retard à la retombée" (TOF), 246
- Temporisation de surveillance du cycle (instruction RE_TRIGR), 321
- Temporisations
 - fonctionnement, 249
 - quantité, 30, 1445, 1459, 1474, 1489, 1506
 - RT (Réinitialiser temporisation), 246
 - taille, 30, 1445, 1459, 1474, 1489, 1506
 - TOF (Temporisation "Retard à la retombée"), 246
 - TON (Temporisation "Retard à la montée"), 246
 - TONR (Temporisation "Retard à la montée mémorisé"), 246
 - TP (Temporisation "Impulsion"), 246
- Temps
 - RD_SYS_T (Lire l'heure), 351
 - T_ADD (Ajouter les temps), 348
 - T_CONV (Convertir et extraire les temps), 347
 - T_SUB (Soustraire les temps), 348
 - WR_LOC_T (Régler l'heure locale), 351
- Temps d'attente, 106, 1138
- Temps de cycle
 - configuration, 112
 - présentation, 111
 - visualisation, 1392
- Temps de filtrage des entrées, 179
- Temps de filtrage des entrées TOR, 179
- Temps de filtre, 179
- Temps de performance, 1444, 1458, 1473, 1488, 1505
- Tensions nominales, 1437, 1437
- Test du programme, 232
- Thermocouples
 - compensation de soudure froide, 1562, 1612
 - principe de fonctionnement, 1562, 1612
 - SB 1231 AI 1 x 16 bits, 1610
 - tableau de sélection de filtre du module SM 1231
 - Thermocouple, 1563
 - tableau de sélection de filtre du SB 1231
 - Thermocouple, 1612
 - Tableau de sélection du filtre SB 1231, 1613
 - tableau de sélection du module SM 1231
 - Thermocouple, 1563
- TIA Portal, vue du portail et vue du projet, 39
- TimeTransformationRule pour l'heure d'été, 353
- TM_MAIL (envoyer e-mail), 1371
- TMAIL_C, 913
- Topologie
 - Anneau, 987
- Topologie en anneau, 987
- Trace, fonction, 1418
- Traitement image
 - état, 1398
 - visualisation, 1398
- Tranche (d'un type de données de variable), 139
- Transmission des données, déclenchement, 1173, 1291
- TRCV, 885
 - ID de liaison, 854
- TRCV (receive data via Ethernet (TCP))
 - mode ad hoc, 854
- TRCV (recevoir des données via Ethernet (TCP))
 - Configuration du paramètre, 954

TRCV_C
 mode ad hoc, 854
 TRCV_C (receive data via Ethernet (TCP))
 ID de liaison, 854
 TRCV_C (receive data via Ethernet (TCP))
 configuration, 833
 TRCV_V (recevoir des données via Ethernet (TCP)), 863
 paramètres de liaison, 857
 TRUNC (Former un nombre entier), 306
 TSAP (points d'accès au service transport), 835
 configuration des paramètres généraux, 950, 1027
 définition, 852
 instructions pour l'affectation aux appareils, 851
 TSAP et numéros de port limités, 944
 TSAP et numéros de port limités, 944
 TSEND, 885
 ID de liaison, 854
 TSEND_C (envoyer des données via Ethernet (TCP)), 863
 configuration de l'instruction, 952
 paramètres de liaison, 857
 TSEND_C (send data via Ethernet (TCP))
 configuration, 833
 ID de liaison, 854
 TSEND_C et TRCV_C
 versions, 862
 versions d'héritage, 876
 TURCV (receive data via Ethernet (UDP))
 configuration, 833
 TURCV (recevoir des données via Ethernet (UDP)), 924
 paramètres de liaison, 857
 TUSEND (envoyer des données via Ethernet (UDP)), 924
 paramètres, 857
 TUSEND (send data via Ethernet (UDP))
 configuration, 833
 Types de données, 129
 Bool, Byte, Word et DWord, 131
 caractères et chaînes de caractères, 135
 éditeur de type de données API, 138
 Real, LReal (réels à virgule flottante), 132
 Struc, 138
 tableaux, 137
 Time, Date, TOD (heure), DTL (date et heure long), 133
 USInt, SInt, UInt, Int, UInt, Dint (entiers), 132
 Variant (pointeur), 139
 Types Enum dans les pages Web personnalisées, 1089, 1090

U

UDP
 configuration des liaisons, 833
 paramètres, 857
 UDP, protocole, 851
 UFILL_BLK (Compléter zone contiguë), 287
 UMOVE_BLK (Copier zone contiguë), 277
 UPPER_BOUND (Lire la limite supérieure d'un ARRAY), 291

V

VAL_STRG (Convertir la valeur numérique en chaîne de caractères), 358
 valeurs booléennes ou de bit, 123
 Valeurs en retour
 instructions point à point, 1156
 instructions pour la communication ouverte, 944
 Variable
 ajout d'un type de données, 141
 forçage permanent, 1404
 surveillance de l'état ou de la valeur, 1397
 tranche, 139
 Variable, page Web État des variables, 1068
 Variables de bloc rémanentes
 chargement à l'état MARCHE, 1413
 Variables, visualisation et forçage à partir du serveur Internet, 1068
 VARIANT_TO_DB_ANY (Convertir VARIANT en DB_ANY), 311
 VariantGet (Lire la valeur de variable VARIANT), 296
 VariantPut (Ecrire la valeur de variable VARIANT), 297
 Vérification de la liaison, 908
 Versions des
 instructions, 45, 862, 876, 884, 895, 1202, 1224, 1253, 1301, 1316, 1336
 Visualisation
 acquisition et réinitialisation de valeurs de DB, 1398
 État CONT, 1398
 Etat CONT et utilisation de la table de visualisation, 1397
 forçage permanent, 1404
 table de forçage permanent, 1403
 table de visualisation, 1400
 temps de cycle, 1392
 utilisation de la mémoire, 1392
 Visualisation des variables à partir du serveur Internet, 1068
 Visualisation du programme, 232
 Visualisation, appareils IHM, 33
 Vitesse de transmission, 1138

Vitesses d'exécution des
instructions, 1444, 1444, 1458, 1458, 1473, 1473, 1488
, 1488, 1505, 1505
Vue du portail, 39
Vue du projet, 39, 40
Vue topologique, 39

W

WChar (type de données caractère de mot), 135
WHILE, SCL, 334
WR_LOC_T (Régler l'heure locale), 351
WR_SYS_T (Régler l'heure), 351
WRIT_DBL (écrire dans un bloc de données dans la
mémoire de chargement), 556
WRITE_BIG (Ecrire des données au format big
endian), 295
WRITE_LITTLE (Ecrire des données au format little
endian), 295
WRREC (écrire l'enregistrement), 395, 413
WString (type de données chaîne de mot), 135
WWW (synchronisation des pages Web
personnalisées), 1099

X

XON / XOFF, 1140

Z

Zone thermique, 55, 59
Zones de mémoire
 accès direct, 122
 adressage des valeurs booléennes ou de bit, 123
 mémoire image, 122
Zones de mémoire, visualisation avec le serveur
Web, 1056

