UNIVERSITÉ DE LORRAINE | IUT Saint-Dié-des-Vosges
Génie Electrique et Informatique Industrielle

LP SARII

AUTOM

Architecture et mise en œuvre d'un système automatisé

1

ORGANISATION DU MODULE

COURS MAGISTRAUX: 4 À 5 SÉANCES DE 2H

SÉANCES DE TRAVAUX DIRIGÉS : 10 SÉANCES DE 2H

SÉANCES DE TRAVAUX PRATIQUES : 8 SÉANCES DE 4H

INTERVENANTS : ÉRIC TERNISIEN-MICHEL RAVIER

Evaluation : 2 examens théoriques + 1 examen pratique

SOMMAIRE DU COURS

3

PARTIE I:GENERALITES/AUTOMATE/GRAFCET

I.1: GÉNÉRALITÉS SUR L'AUTOMATISME INDUSTRIEL
I.2: ARCHITECTURE D'UN SYSTÈME AUTOMATISÉ
I.3: DÉFINITIONS: CAPTEUR, PRÉ-ACTIONNEUR, ACTIONNEUR
I.4: CONCEPTION ET RÉALISATION D'UN SYSTÈME AUTOMATISÉ
I.5: PRÉSENTATION D'UN AUTOMATE INDUSTRIEL ET DES SOLUTIONS ALTERNATIVES
I.6: UNITÉ CENTRALE OU CPU – STRUCTURES ZONE DE MÉMOIRES
I.7: CYCLE API – CHIEN DE GARDE – INTERRUPTIONS DE PROGRAMME
I.8: GRAFCET: - ÉTAPES
- ACTIONS
- TRANSITIONS
- RÈGLES D'ÉVOLUTION
- DIVERGENCES

SOMMAIRE DU COURS

4

SOMMAIRE DU COURS

5

PARTIE I: Généralités/Automate/Grafcet

6

I.1: GÉNÉRALITÉS SUR L'AUTOMATISME INDUSTRIEL
I.2: ARCHITECTURE D'UN SYSTÈME AUTOMATISÉ
I.3: DÉFINITIONS: CAPTEUR, PRÉ-ACTIONNEUR, ACTIONNEUR
I.4: CONCEPTION ET RÉALISATION D'UN SYSTÈME AUTOMATISÉ
I.5: PRÉSENTATION D'UN AUTOMATE INDUSTRIEL ET DES SOLUTIONS ALTERNATIVES
I.6: UNITÉ CENTRALE OU CPU – STRUCTURES ZONE DE MÉMOIRES
I.7: CYCLE API – CHIEN DE GARDE – INTERRUPTIONS DE PROGRAMME
I.8: GRAFCET: - ÉTAPES
- ACTIONS
- TRANSITIONS
- RÈGLES D'ÉVOLUTION
- DIVERGENCES

I.1: Généralités sur l'automatisme industriel

Les exemples de systèmes automatisés de production sont nombreux. Les plus représentatifs et les plus gourmands en automates **sont les centres de fabrication d'automobiles**.

Les principales applications que l'on retrouve dans l'industrie :

❑**Commande de machines :**
- ❖Machines outils
- ❖Convoyage, stockage
- ❖Emballage
- ❖Machines de chantier, engin de levage

❑**Régulation de processus:**
- ❖Chimie, pétrochimie, pharmaceutique
- ❖Traitement des eaux
- ❖Thermique, fours, métallurgie

I.1: Généralités sur l'automatisme industriel

❑**Contrôle de systèmes:**
❖Production et distribution d'énergie (électricité, pétrole, gaz)
❖Transports
(chemin de fer, routier, marine)

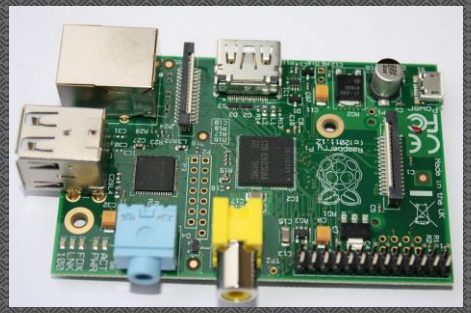
❑**Automatisme du bâtiment (ADB) :**
❖Chauffage, climatisation, sanitaire
❖Distribution électrique, éclairage
❖Sécurité, alarmes techniques

I.1: Généralités sur l'automatisme industriel

Parmi les applications industrielles présentées précédemment, les constructeurs de systèmes automatisés utilisent pour la plupart **du matériel industriel.**

- L'intelligence d'un système automatisé repose généralement sur **un automate programmable industriel A.P.I**

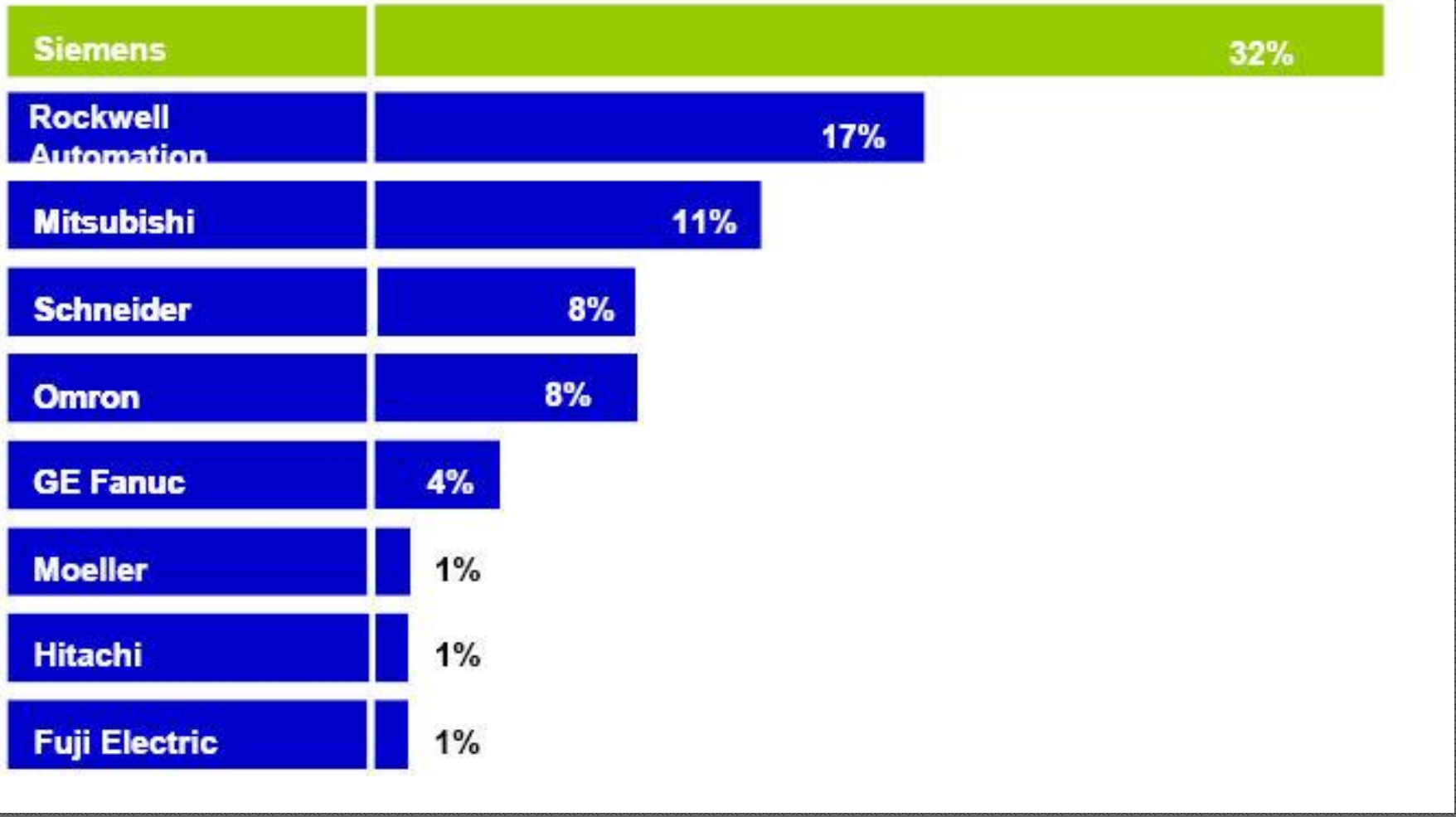
- L'A.P.I n'est pas l'unique moyen, d'autres solutions alternatives existent comme **le système à microprocesseur ou l'ordinateur (PC industriel).**



Nous reviendrons dans le chapitre I.5, pour présenter les différences entre les solutions A.P.I ou système à microcontrôleur.

I.1: Généralités sur l'automatisme industriel

L'A.P.I est souvent privilégié, voici **l'état du marché des automates dans le monde :**

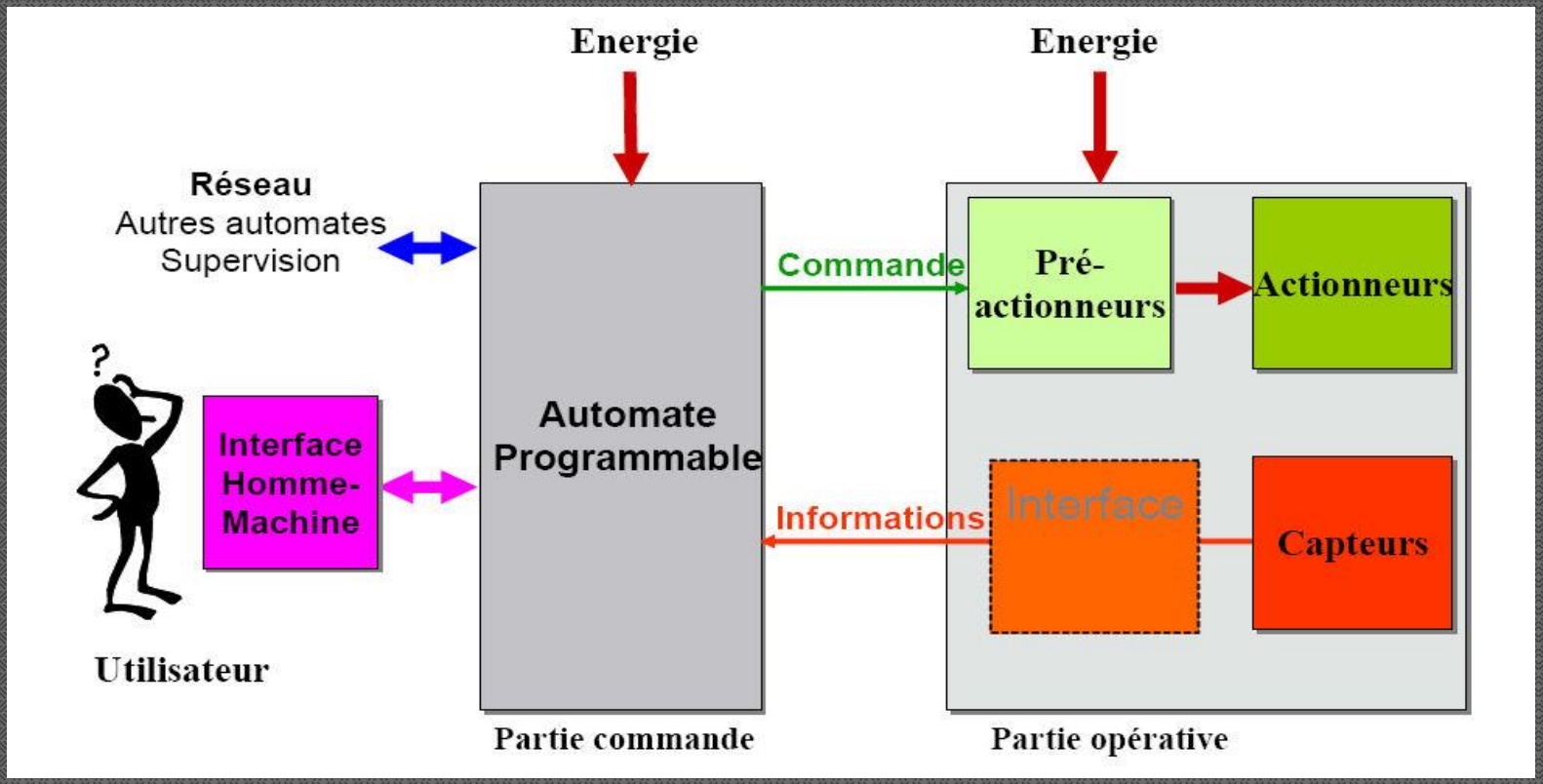


I.2: Architecture d'un système automatisé

Un système automatisé est **un ensemble d'éléments qui effectue des actions sans intervention de l'utilisateur.**

L'utilisateur ou l'opérateur se contente de donner des ordres de départ et si besoin d'arrêt.

Un système automatisé industriel peut en première description être schématisé par le synoptique suivant :



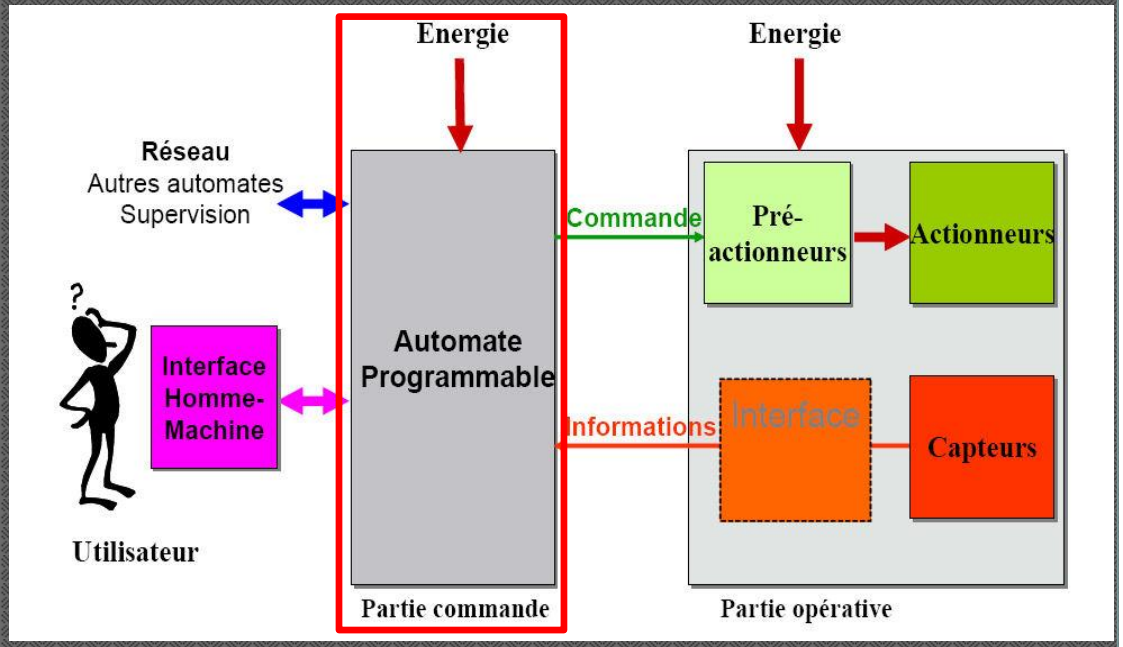
I.2: Architecture d'un système automatisé

Un système automatisé est composé de plusieurs parties :

La partie commande (PC):

Elle donne **les ordres** et reçoit **les informations** issues de la partie opérative.

Elle peut également effectuer des échanges avec l'extérieur via une interface homme-machine ou par un réseau industriel.



I.2: Architecture d'un système automatisé

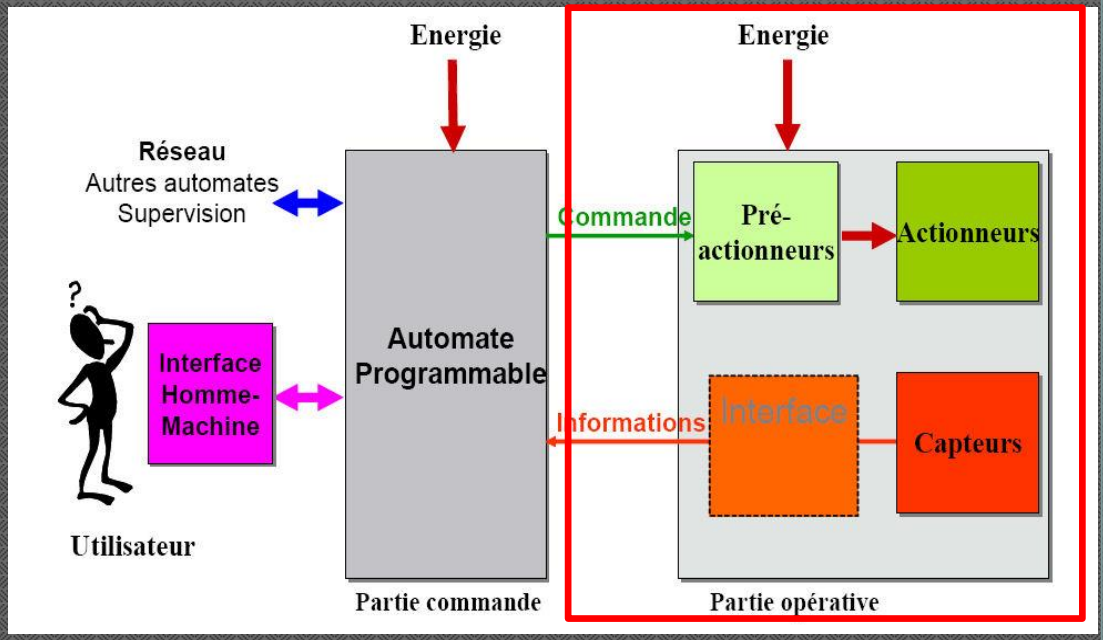
Un système automatisé comprend également :

La partie opérative (PO):

C'est la partie d'un système automatisé qui **effectue le travail.**
Autrement dit, c'est **la machine.**

C'est cette partie qui reçoit les ordres de la partie commande et qui les exécute.

Elle comporte:
- les pré-actionneurs et les actionneurs
- Les capteurs et leurs interfaces



I.2: Architecture d'un système automatisé

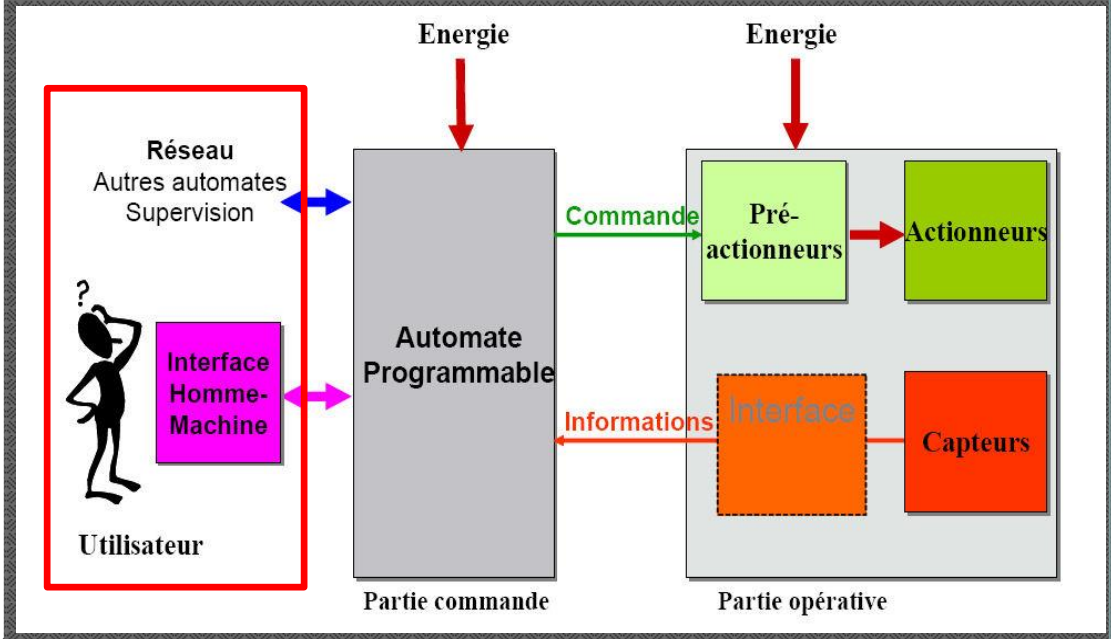
Enfin un système automatisé comprend également :

Une interface homme-machine et un réseau industriel :

L'interface homme-machine (I.H.M) est l'interface utilisateur qui relie l'opérateur au dispositif de commande d'un système industriel.

Exemples: boîte à boutons, voyants ou écran tactile.

La partie commande peut aussi échanger des informations avec d'autres systèmes (voir cours réseaux industriels)



Boîte à boutons, voyants et écran tactile

I.3: Définitions: Capteur - Pré-actionneur - Actionneur

Ces éléments font parties de la partie opérative, **ils sont indispensables à l'automatisation du système.**

Capteur et interface :

Un capteur est un élément de la partie opérative qui permet de recueillir des informations et de les transmettre à la partie commande. Les capteurs sont choisis en fonction des informations qui doivent être recueillies (température, son, lumière, déplacement, position)

Afin de pouvoir être traités par la partie commande, **les signaux issus de capteurs** placés sur le processus **sont parfois conditionnés par une électronique d'interface** (traitement d'image, mise en forme des signaux…) amplification…)

Codeur incrémental, caméra, détecteur inductif, détecteurs photoélectriques, détecteur de contact

I.3: Définitions: Capteur - Pré-actionneur - Actionneur

Exemples: capteur/interface/API :

Partie opérative (PO) | **Partie commande (PC)**

Capteur | **Interface**

Capteur de température de type PT100 (100Ω à 0°C)

Valeur ohmique en fonction de la température

Convertisseur PT100/4-20mA

Signal électrique de type 4-20mA

A.P.I (avec carte entrée analogique 4-20mA)

Capteur de pression de type différentiel

Signal électrique 0-10V en fonction de la pression

Convertisseur 0-10V/4-20mA

Signal électrique de type 4-20mA

A.P.I (avec carte entrée analogique 4-20mA)

I.3: Définitions: Capteur - Pré-actionneur - Actionneur

Actionneurs :
Un actionneur est un élément de la partie opérative qui est capable de produire **une action physique** tel qu'un déplacement, un dégagement de chaleur, une émission de lumière ou de son à partir de l'énergie qu'il a reçu.

Moteur électrique

Vérins pneumatiques

Thermoplongeur

Electrovanne

I.3: Définitions: Capteur - Pré-actionneur - Actionneur

Pré-actionneurs :

Les actionneurs **sont commandés par des systèmes électriques de commande intermédiaires appelés « pré-actionneurs ».**

Ils sont indispensables pour **les adaptions électriques** (tension, intensité). En effet, un API ne peut mettre en œuvre un moteur de 400kw triphasé seul, il faut un pré-actionneur entre l'API et ce moteur.

Ces pré-actionneurs mettent en œuvre des circuits d'électronique de puissance, d'électronique du signal analogique et d'électronique numérique.

Un pré-actionneur a pour fonction de transformer l'énergie issue d'une source (réseau électrique, compresseur pneumatique,...) en une énergie adaptée à l'actionneur pour un mouvement précis.

Contacteur et variateur de vitesse

Distributeur électropneumatique

I.3: Définitions: Capteur - Pré-actionneur - Actionneur

Exemples: API/pré-actionneur/actionneur :

Partie commande (PC)

Partie opérative (PO)

Pré-actionneur

Actionneur

A.P.I
(avec carte sortie analogique 0-10V)

Commande 0-10V

Variateur de vitesse
(électronique de puissance convertisseur de fréquence de type DC/AC)

Commande 0-400V tri à fréquence variable

Moteur asynchrone
(transforme l'énergie électrique en énergie mécanique)

A.P.I
(avec carte sortie analogique 0-10V)

Commande 0-10V

Gradateur de puissance
(électronique de puissance convertisseur de type AC/AC)

Commande 0-230V mono à valeur efficace variable

Thermoplongeur
(transforme l'énergie électrique en énergie thermique)

I.4: Conception et réalisation d'un système automatisé

La **conception et la réalisation** d 'un système automatisé comporte d'importants **enjeux techniques et économiques** dont il convient de limiter les risques.

Historiquement, pour la conception et la réalisation d'un système automatisé, les automaticiens adoptaient un schéma dit « **cycle en V** ». (voir transparent suivant)

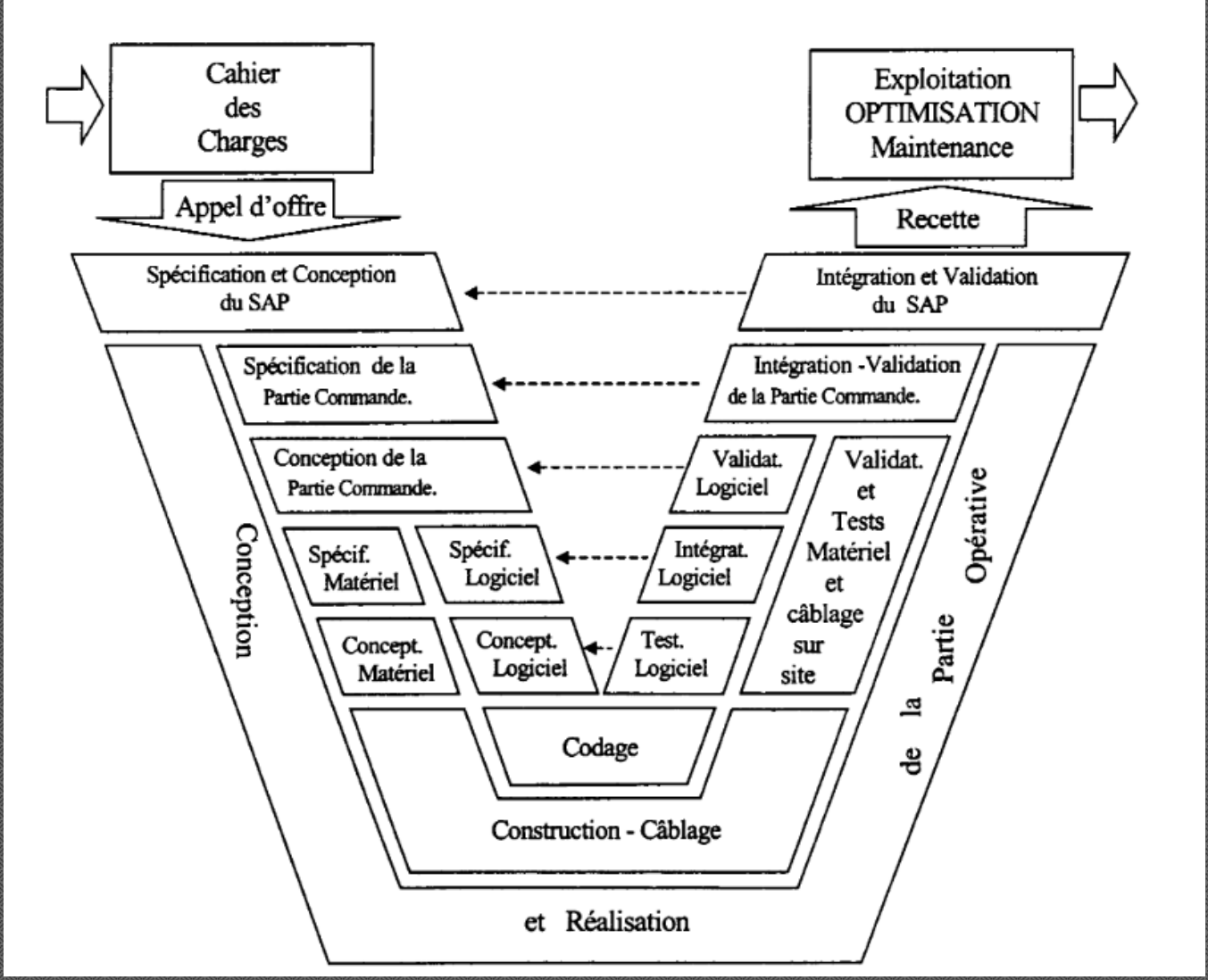
Le développement d'un système comportait différentes phases de :

- **Spécification**
- **Conception**
- **Réalisation**
- **Intégration**
- **Validation**

Toutes les entreprises du secteur de l'automatisation ont leur propre méthode de développement. Mais globalement, nous retrouverons des phases communes.

Nous essayerons dans cette partie de cours de cerner les principales tâches qui incombent au métier d'automaticien.

I.4: Conception et réalisation d'un système automatisé



Cycle en V

I.4: Conception et réalisation d'un système automatisé

La conception d'un système automatisé complexe comme une unité de production automatisé ne concerne pas que les automaticiens.

En effet, dans ce type de projet participeront des corps de métier comme:

- **des thermiciens**
- **des physiciens**
- **des électriciens**
- **des chimistes**
- **des hydrauliciens**
- **des mécaniciens**
- **etc…**



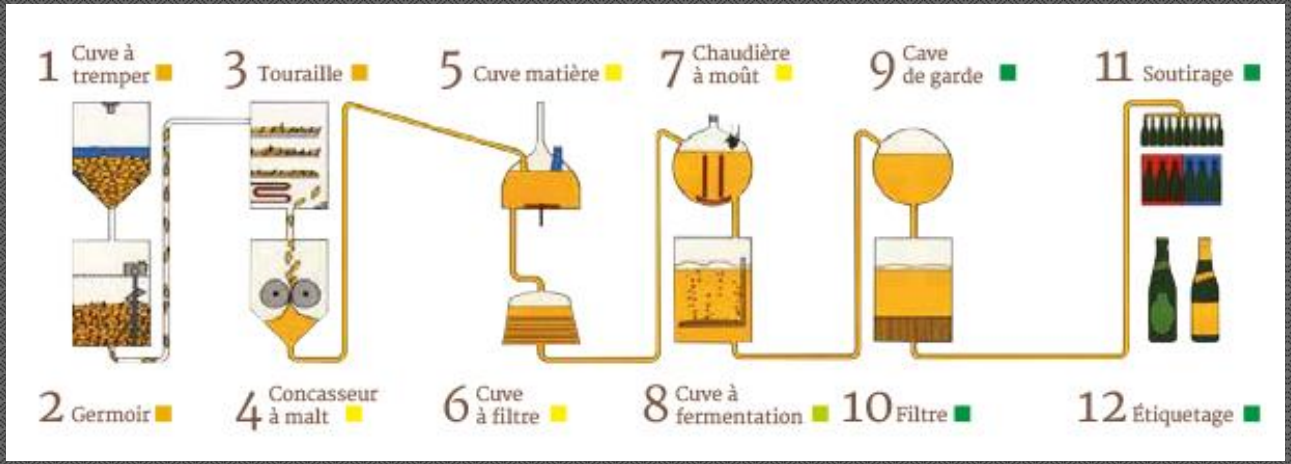
Le rôle de l'automaticien se limitera à l'étude et à l'intégration des systèmes de contrôle-commande (automate, système de supervision, instrumentation etc..).
Il travaillera donc en étroite collaboration avec les autres corps de métiers pendant l'élaboration du projet.

Le monde de la machine spéciale fonctionne à peu près tous de la même manière.

Le bureau d'étude électricité et automatisme devra travailler avec les autres départements comme le département mécanique, informatique ou encore R&D.

I.4: Conception et réalisation d'un système automatisé

Exemple: Une brasserie industrielle :



Connaissances du métier BRASSEUR
A partir de céréales d'orge, fabriquer une bonne bière ??

L'automaticien devra travailler de concert avec les brasseurs pour automatiser les différentes phases de la fabrication jusqu'au conditionnement.

Lors de l'installation d'une nouvelle brasserie d'autres corps de métiers entreront également en action.

I.4: Conception et réalisation d'un système automatisé

Exemple: Une brasserie industrielle :

Ainsi dans le domaine de la brasserie industrielle, les systèmes automatisés vont se charger de:

- fabriquer la bière selon les recettes établies
- remplir automatiquement les bouteilles
- de les bouchonner
- d'assurer leur déplacement vers les zones de stockage et de prélèvement

Grâce aux systèmes automatisés, la production est décuplée.

Cuve matière automatisée

Poste de soutirage automatisé

Convoyage de bouteilles automatisé

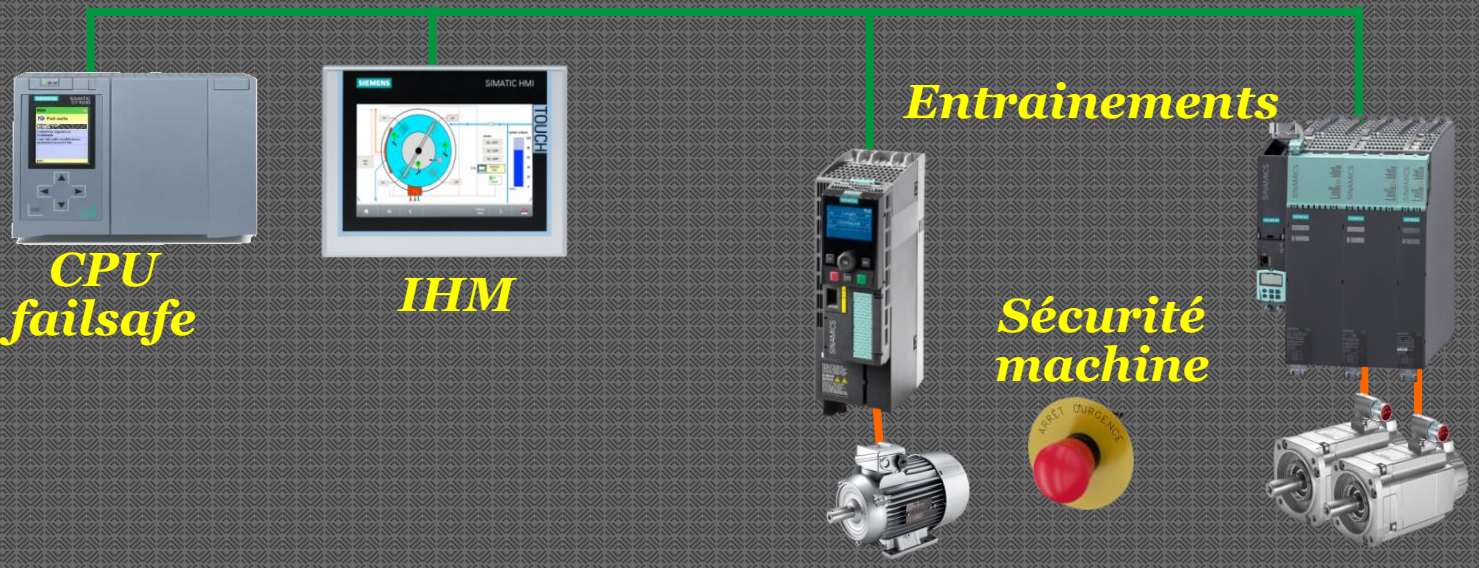
I.4: Conception et réalisation d'un système automatisé

L'automaticien dont le rôle est de donner souffle à la machine à concevoir suivra un cahier des charges pré-établi qui liste toutes les fonctionnalités que doit comporter celle-ci.

Il s'aidera d'autres documents techniques comme les documents d'analyse fonctionnelle.

Tous ces documents vont l'aider lors du choix technologique

- **choix du type de contrôleur**
- **système d'entrainement adéquat**
- **exigence du système en terme de sécurité**
- **etc...**



I.4: Conception et réalisation d'un système automatisé

Pour résumer, la création d'un système automatisé ou machine spéciale passera **par 3 grandes phases essentielles :**

- **La phase de conception** (élaboration des documents de conception, spécifications techniques du système, norme réglementaire et de sécurité)
- **La phase de réalisation** (fabrication des pièces mécaniques, assemblage, câblage, programmation des automates et superviseurs etc..)
- **La phase de mise en service** (dernier mise au point, test de la machine en situation de fonctionnement normale)

Phases	Place de l'automaticien
CONCEPTION	Choix de l'API et de l'ensemble du matériel d'automatisation
REALISATION	Programmation de l'API et mise en œuvre du matériel d'automatisation
MISE EN SERVICE	Vérifications du matériel d'automatisation et du programme et validation du système dans les différents modes de fonctionnement définis par le cahier des charges

I.4: Conception et réalisation d'un système automatisé

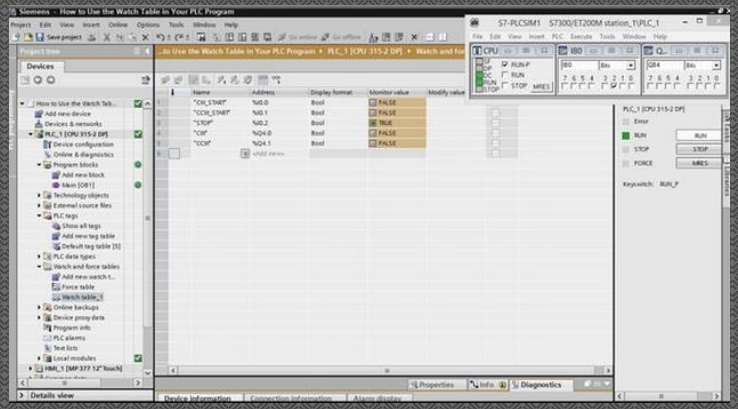
Mise en service pour l'automaticien :

Troisième phase du projet, elle est une des plus importantes pour l'automaticien.

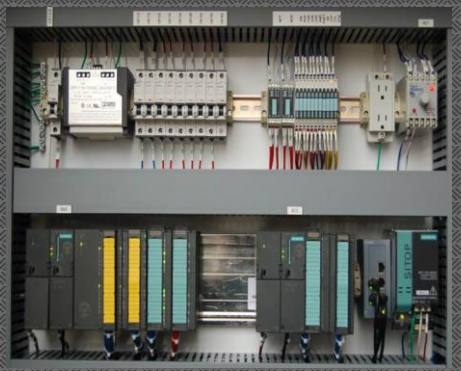
❖ **La vérification du matériel :**

Sur une machine automatisée, chaque capteur, interrupteur et bouton poussoir est connecté à une entrée spécifique de l'automate et chaque actionneur (vérin, moteur, voyant) à une sortie spécifique de celui-ci.

Il faudra donc veiller à bien vérifier que chaque capteur/actionneur soit connecté à la bonne entrée/sortie



Vérification des entrées/sorties par table de forçage



Vérification des LED(s) d'entrées TOR de l'API

Vérification par multimétrie

I.4: Conception et réalisation d'un système automatisé

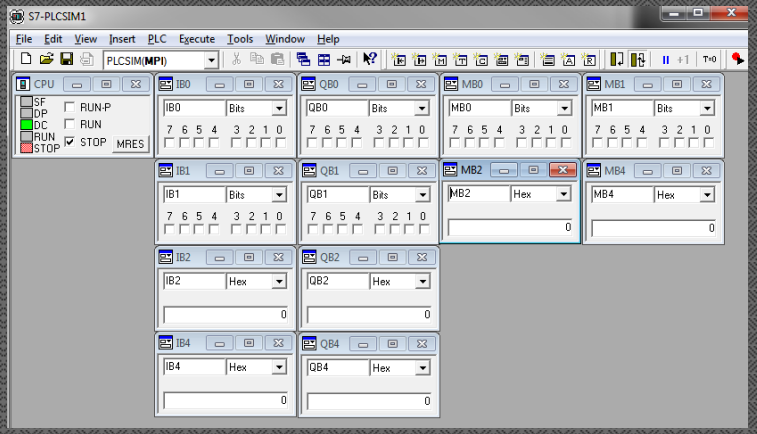
Mise en service pour l'automaticien :

❖ La vérification du programme :

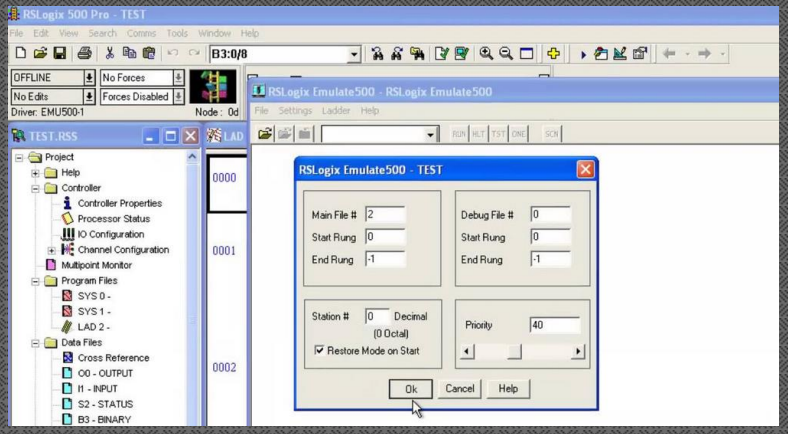
Pendant la phase de vérification du programme, l'automaticien pourra utilisé des outils de simulation afin de détecter tous les défauts émanants du programme.

Si vous utilisez un automate Siemens, vous pourrez utiliser l'application PLCSIM comme automate virtuel. Pour un automate Allen Bradley, vous pourrez utiliser RSEmulate.

Ces outils permettent de simuler le fonctionnement d'un automate et de faciliter le test du programme ainsi conçu.



PLCSIM



RSEmulate.

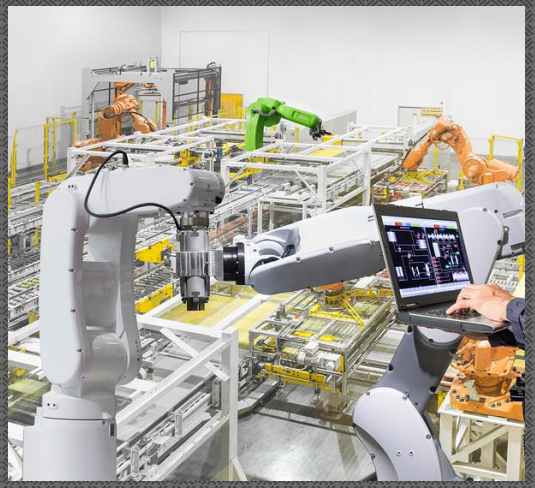
I.4: Conception et réalisation d'un système automatisé

Mise en service pour l'automaticien :

L'automaticien devra aussi **tester l'interface homme-machine utilisée, vérifier sa bonne configuration et s'assurer de la communication entre les différents dispositifs du système.**

Ceux-ci pouvant communiquer via des bus de terrain traditionnels comme le Profibus, le modbus, l'ASI ou encore via les réseaux d'Ethernet industriel.

Après cela, les parties individuelles du programme et les fonctions spéciales du système seront testées: fonctionnement en mode manuel et automatique, configuration et vérification des bases de données d'archivage etc..



I.5: Présentation d'un automate industriel et des solutions alternatives

Choix : API industriel /microcontrôleur/PC industriel :

Au moment de concevoir un système automatisé ou une machine spéciale, **on est souvent confronté à des choix technologiques à savoir le choix des capteurs, des actionneurs et des organes de contrôle/commande.**

VS !

ou

I.5: Présentation d'un automate industriel et des solutions alternatives

Cartes à microcontrôleur du marché:

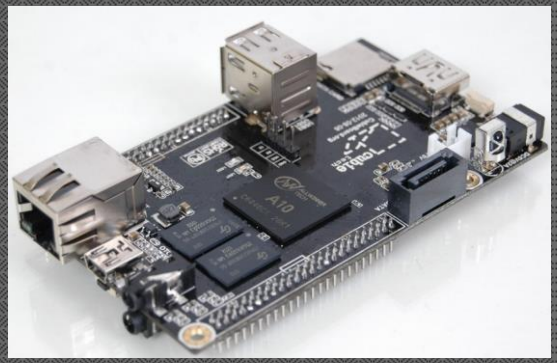
De nos jours avec **l'arrivée des SBC (single board computer)** comme le Raspberry Pi, l'Arduino, le CubieBoard, on n'a plus seulement un microcontrôleur séparé, on a un **ordinateur complet avec microcontrôleur, entrées/sorties numériques et analogiques, port d'alimentation etc..**

Ainsi, concevoir un système automatisé devient de plus en plus simple comparé à autrefois où on disposait seulement d'un PIC et on devait concevoir les cartes d'entrées/sorties.

Certaines entreprises vont également privilégier **le PC industriel comme organe de contrôle/commande.**

Raspberry Pi

Arduino



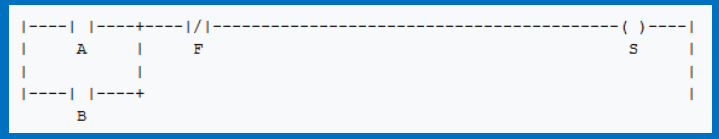
CubieBoard

I.5: Présentation d'un automate industriel et des solutions alternatives

Quelques éléments de comparaison :

Le prix : Les automates programmables sont souvent plus chères que les (microcontrôleurs + cartes) ou les ordinateurs monocartes ainsi pour de petites applications, il est plus intéressant de se tourner vers les microcontrôleurs.

La facilité de mise en œuvre et de programmation : Les automates sont programmés via des langages standard comme le ladder ou le grafcet facile à comprendre.



Comparé aux microcontrôleurs qui doivent utiliser pour des soucis de performance le langage assembleur, difficile pour un non initié.
Les microcontrôleurs comme certains APIs supportent de nos jours les langages évolués comme le C, cependant il faut avoir de bonnes bases en informatique.

I.5: Présentation d'un automate industriel et des solutions alternatives

Quelques éléments de comparaison :

La performance et la sécurité:

Les automates sont conçus pour les applications industrielles et peuvent fonctionner dans des milieux précaires (très basses ou hautes températures, milieux humides etc...).

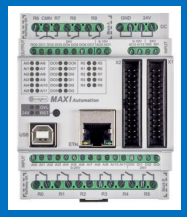
Ils sont aussi testés et approuvés pour répondre aux problèmes d'incompatibilités électromagnétiques. Comparées aux SBC qui sont pas assez durcis pour répondre aux besoins industriels.

Il existe néanmoins des solutions électroniques à base de microcontrôleurs conçues pour être utilisées en industrie.

Des versions industrielles de Arduino nommées Industruino ou Controllino tentent de répondre à l'intégration de système à base d'Arduino dans le milieu industriel.

Industrino

Controllino

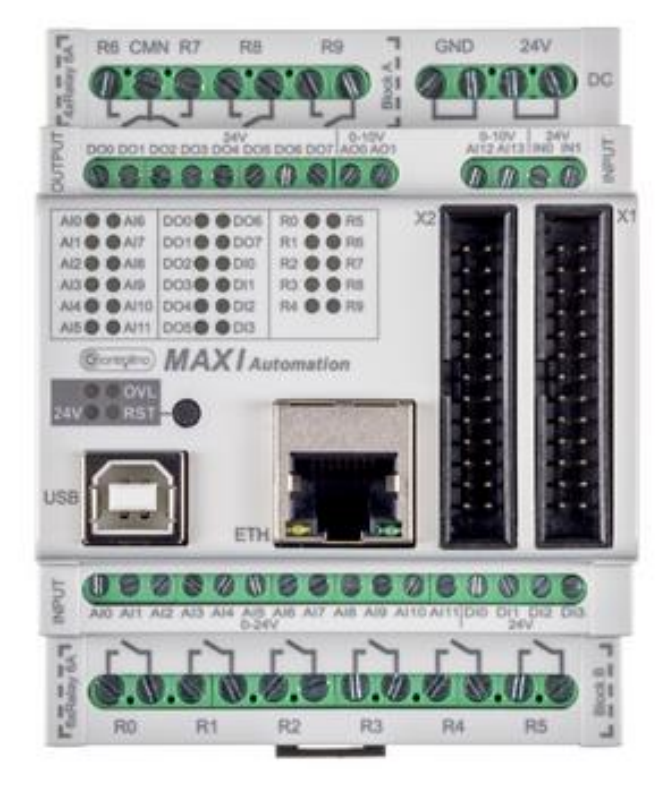


Pour conclure, le fait de choisir un automate par rapport à un microcontrôleur dépend principalement du type d'application, du budget et de la familiarisation avec les langages informatiques du programmeur.

I.5: Présentation d'un automate industriel

Exemple de solution microcontrôleur: CONTROLLINO

Une des solutions pour le choix de l'élément de contrôle commande peut être le Controllino: (une base sur carte Arduino avec des adaptations pour l'industrie)

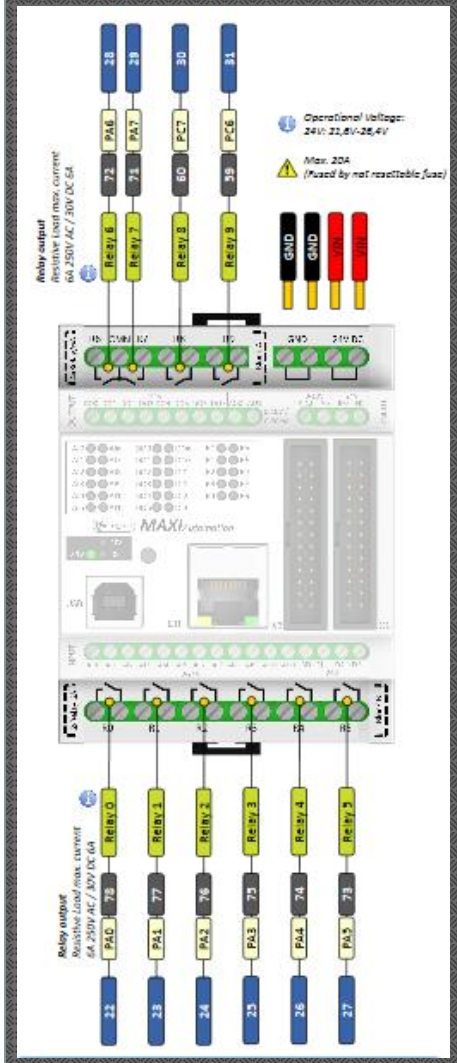


Quelques spécifications :

- Microcontroller: ATmega2560
- Ethernet Connector
- 2x serial Interface
- 1x I2C Interface
- 1x SPI Interface
- Input current Max. 20A
- 12x Analog/Digital Inputs 0-24V
- 2x Analog Inputs 0-10V
- 6x Digital Inputs (2x Interrupt)
- 8x Digital Outputs – 2A (PWM)
- 2x Analog Outputs – 0-10V (0-20mA)
- 10x Relays Outputs – 230V / 6A

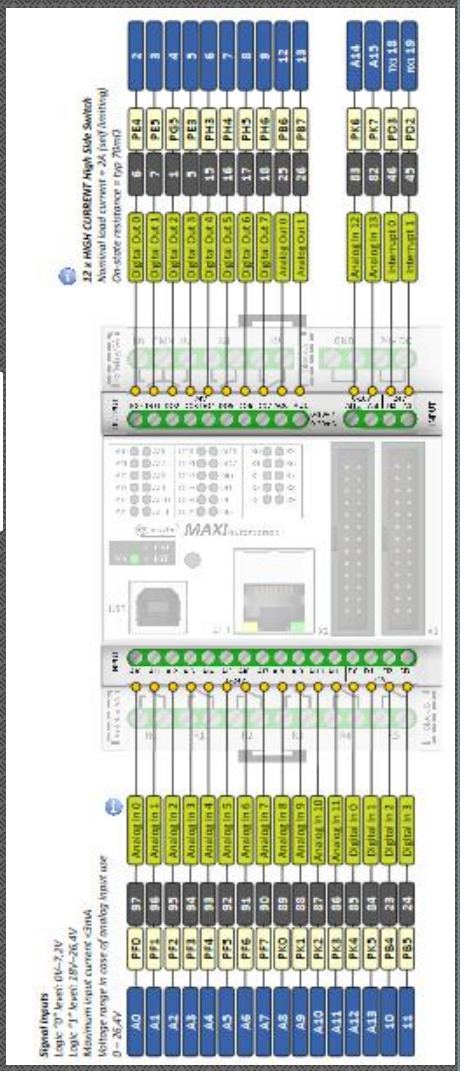
I.5: Présentation d'un automate industriel et des solutions alternatives

Exemple de solution microcontrôleur: CONTROLLINO



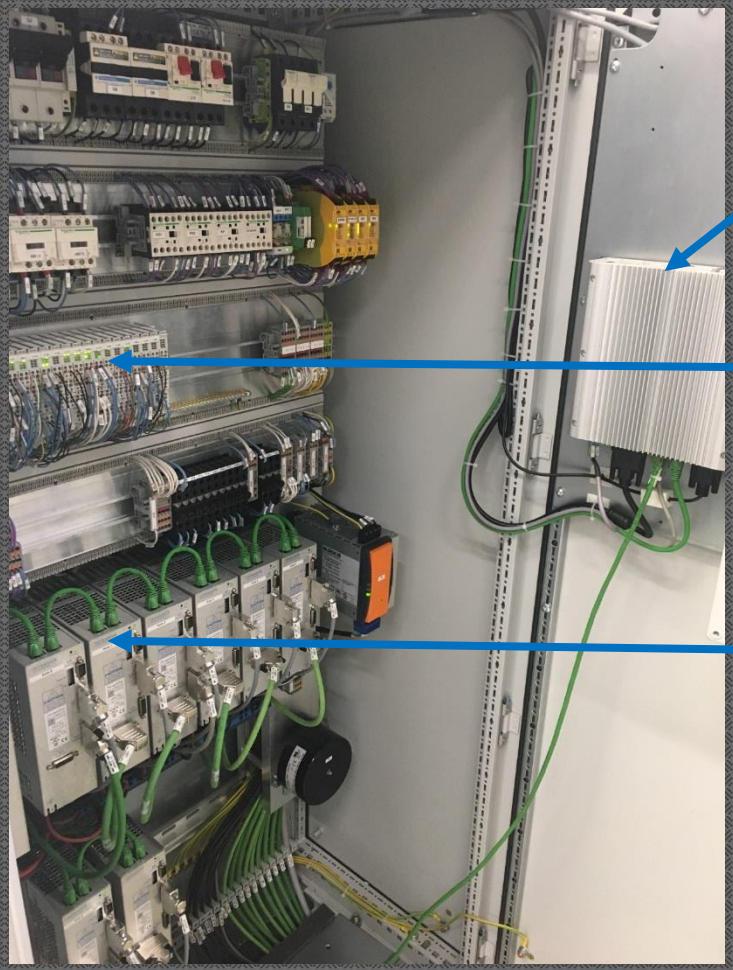
Toutes les informations techniques concernant ce produit (datasheet, caractéristiques techniques) :

https://controllino.biz

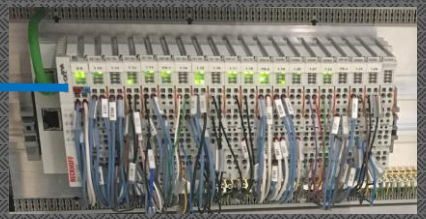


I.5: Présentation d'un automate industriel et des solutions alternatives

Exemple de solution: PC industriel



PC industriel :
- ❖ **organe de commande**



Bloc d'entrées/sorties déporté :
- ❖ **matériel industriel**
- ❖ **communication avec le PC**

Variateurs industriels :
- ❖ **matériel industriel**
- ❖ **communication avec le PC**

Le PC industriel peut être une solution de contrôle/commande. Néanmoins, il nécessite **une maitrise des langages informatiques.**

I.5: Présentation d'un automate industriel et des solutions alternatives

Solution la plus retenue API industriel :

La solution la plus utilisée en industrie reste l'API industriel.

Nous reviendrons en détail dans la présentation des différents modèles présents sur le marché.

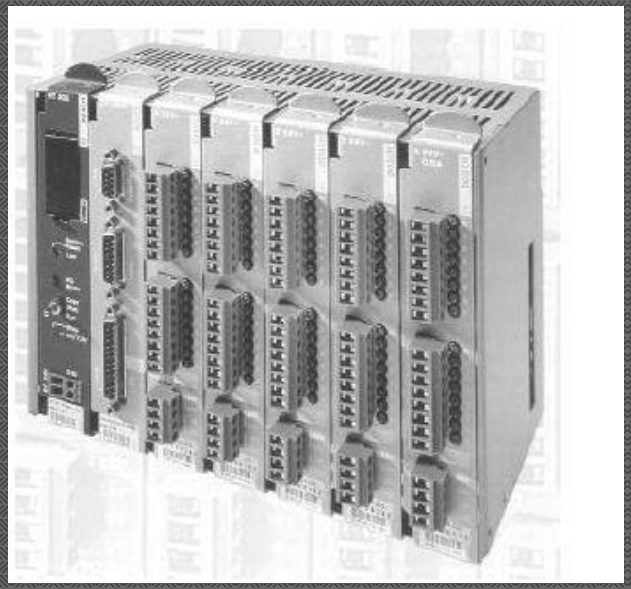
I.5: Présentation d'un automate industriel et des solutions alternatives

Historique de l'API :

C'est **MODICOM** qui créa en 1968, aux USA, le premier automate programmable.

Son succès donna naissance à une industrie mondiale qui s'est considérablement développé depuis.

L'automate programmable représente aujourd'hui l'intelligence des machines et des procédés automatisés dans l'industrie, des infrastructures et du bâtiment.



I.5: Présentation d'un automate industriel et des solutions alternatives

Evolution de l'API :

❑Dans les années 80, les automates deviennent de plus en plus gros :
- ❖ De plus en plus d'entrées/sorties
- ❖Un automate commande plusieurs machines

Architecture centralisée

❑Dans les années 90, on utilise des automates plus petits reliés entre eux par des réseaux.

Architecture décentralisée

❑De nos jours, les entrées/sorties sont aussi décentralisées, on parle de modules déportés.
❑L'automate possède de plus en plus de ports de communication et de moins en moins d'entrées/sorties en local.

I.5: Présentation d'un automate industriel et des solutions alternatives

Présentation de l'API :

***API* :**
Automate programmable industriel
Ou
***PLC* :**
Programmable Logic Controller

- Sont des appareils électroniques de traitement de l'information (remplacement de ce que l'on l'appelait autrefois : « la logique à relais »)

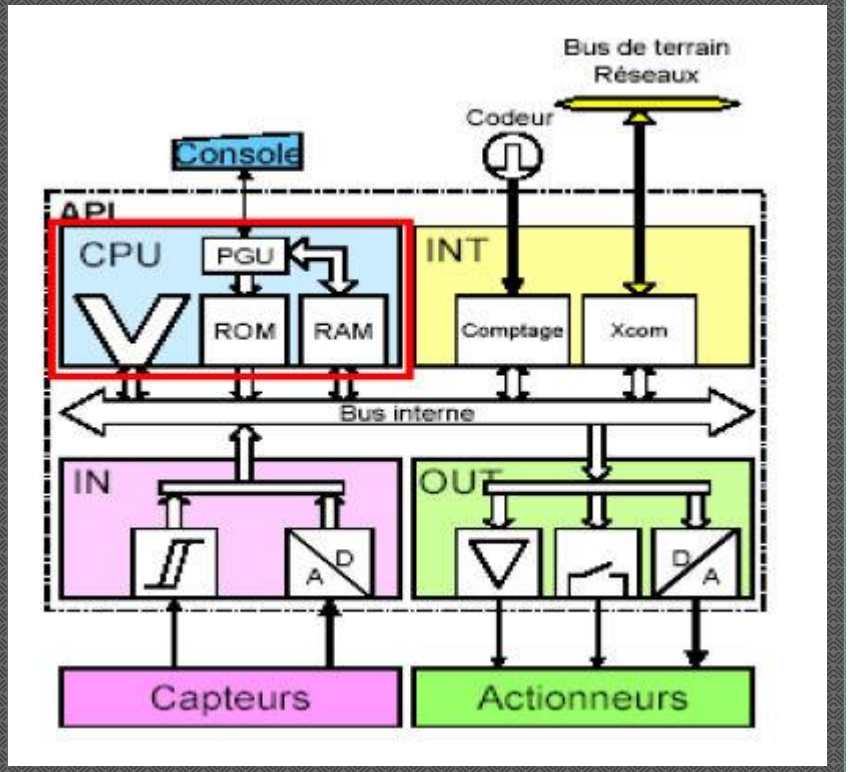
- Effectuent des fonctions d'automatisme programmées telles que :
  - **Logique combinatoire**
  - **Séquencement**
  - **Temporisation**
  - **Comptage**
  - **Calculs numériques**
  - **Asservissement, régulation**
  - **Communication (réseaux de terrain)**

- Permettent de commander, mesurer et contrôler au moyen de signaux d'entrées et de sorties (numériques ou analogiques) toutes machines et processus, dans un environnement industriel.

I.6: Unité centrale - CPU - Structures zone de mémoires

Architecture interne de l'API :

- Un automate programmable industriel se compose :
  - **D'une unité de traitement (CPU + Mémoire)**
  - **De modules d'entrées et de sorties**
  - **De modules de communication.**
  - **D'un module d'alimentation**
  - **D'éventuels modules spécifiques comme le comptage rapide selon les modèles.**
  - **D'un bus interne**



Nous reviendrons dans le chapitre II pour présenter les différentes cartes d'entrées/sorties que proposent les différents constructeurs du marché.

I.6: Unité centrale - CPU - Structures zone de mémoires

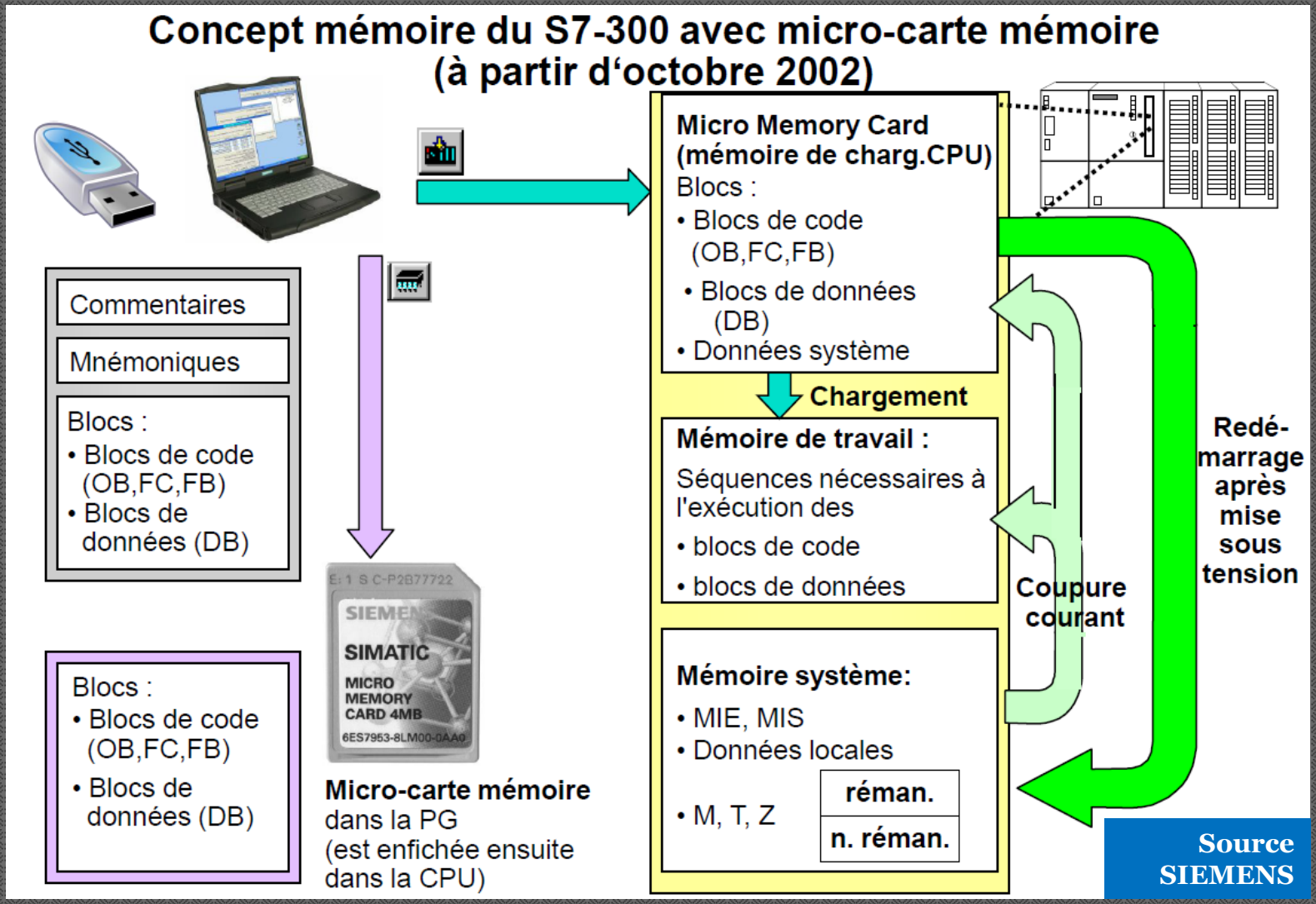
CPU SIEMENS- S7300

- **La CPU comprend:**
  - **Le processeur qui effectue les opérations logique, de temporisation, de comptage ou de calcul.**

- **En règle générale la mémoire d'une CPU se compose :**
  - **D'une zone de mémoire programme.**
    - Dans la phase d'étude et de mise au point cette mémoire doit être reprogrammable (RAM, EAPROM Electrically Alterable)
    - Dans la phase d'exploitation la mémoire doit être sauvegardée par batterie ou être non volatile (EEPROM, Flash)
  - **D'une zone de mémoire de données.**
    - Pour sauvegarder l'état des E/S, compteurs, variables internes…
    - Attention, il peut y avoir plusieurs zones. Bien regarder les spécificités indiquées par le constructeur. Il faut connaitre le comportement de cette zone en cas de coupure de l'énergie.

I.6: Unité centrale - CPU - Structures zone de mémoires

Exemple de la structure de mémoire d'un API SIEMENS-S7300 :



I.6: Unité centrale - CPU - Structures zone de mémoires

Carte MMC :
Cette MMC sert de mémoire de chargement de la CPU. Elle permet d'enregistrer les blocs de code et de données système (configuration matérielle, liaisons de communication, etc. Elle peut même stocker l'ensemble du projet. Le contenu de la MMC est par nature rémanent. Lorsqu'un bloc ou l'ensemble du programme utilisateur est chargé dans la CPU via le PC, l'enregistrement est réalisé sur la MMC. Toutes les séquences nécessaires à l'exécution des blocs sont automatiquement transférées dans la mémoire de travail (RAM).
Le chargement d'un bloc n'est possible que si la MMC est enfichée.
Un effacement général est requis après chaque retrait ou enfichage de la MMC.

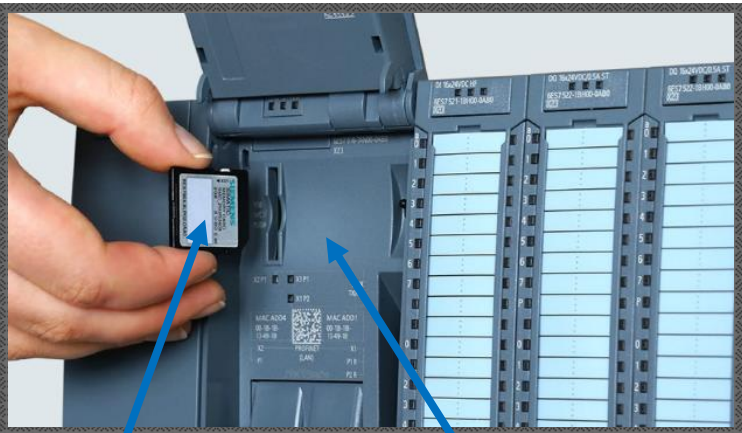
Mémoire de travail :
La mémoire de travail (RAM) est intégrée à la CPU et contient uniquement les données requises pour la durée de l'exécution (pour les blocs de données, uniquement les valeurs effectives et non les valeurs initiales).

Mémoire système:
La mémoire système contient les zones mémoire pour : MIE et MIS, les mémentos, les temporisations, les compteurs, les données locales.

Rémanence:
On qualifie de rémanentes toutes les données qui sont sauvegardées ou ne perdent pas leur contenu en cas de coupure de courant. Il s'agit de toutes les données de la mémoire de travail ainsi que des mémentos, temporisations et compteurs déclarés rémanents dans la configuration matérielle. La rémanence signifie qu'après coupure de courant, les données concernées sont sauvegardées sur la MMC et rechargées au démarrage lorsque la tension est rétablie.

I.6: Unité centrale - CPU - Structures zone de mémoires

Exemple de la structure de mémoire d'un API SIEMENS-S71500 :



Zone mémoire sur carte SD

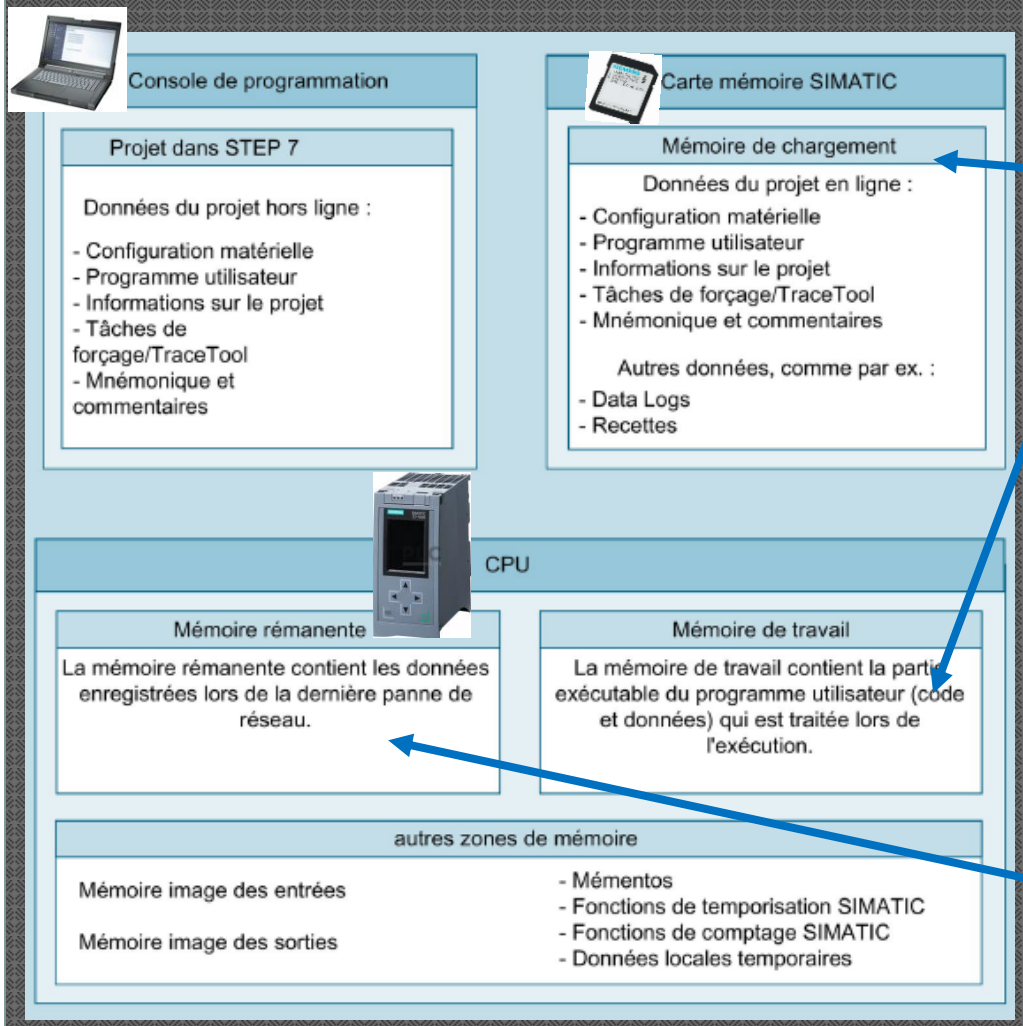
Zones mémoire dans CPU

	Objets	Mémoire de charg	Mémoire de travail	Mémoire de travail	Mémoire rémanente
1		19 %	1 %	8 %	23 %
2					
3	Totale:	0 kB	1048576 octets	5242880 octets	464000 octets
4	Occupée(s):	0 kB	15293 octets	411034 octets	110592 octets
5	Détails	2 MB			
6	DB	4 MB	469 octets		
7	FC	12 MB	410 octets		
8	FB	24 MB	14414 octets		
9	DB	256 MB		411034 octets	110592 octets
10	Types de données	2 GB 6681 octets	-		
11	Variables API	32 GB			0 octets

Plusieurs tailles de cartes SD

I.6: Unité centrale - CPU - Structures zone de mémoires

Exemple de la structure de mémoire d'un API SIEMENS-S71500 :



Mémoire de chargement :
La mémoire de chargement est une mémoire non volatile pour blocs de code, blocs de données, objets technologiques et configuration matérielle. La mémoire de chargement se trouve sur la carte SD (donc indispensable).

Mémoire de travail :
La mémoire de travail est une mémoire volatile qui contient les blocs de code et de données. La mémoire de travail est intégrée à la CPU et ne peut pas être étendue. Elle n'est utilisée que lorsque la CPU est en cours de fonctionnement. 2 zones (code, données/OT)

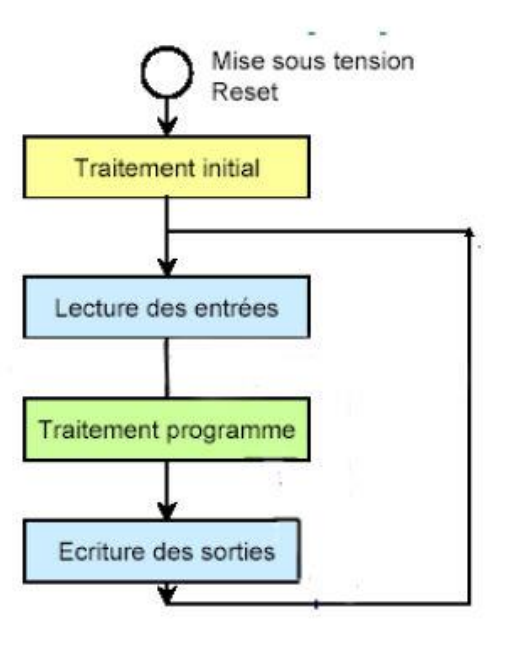
Mémoire rémanente :
La mémoire rémanente est une mémoire non volatile pour la sauvegarde d'une quantité limitée de données en cas de défaillance de tension.
Les actions suivantes suppriment le contenu de la mémoire rémanente :

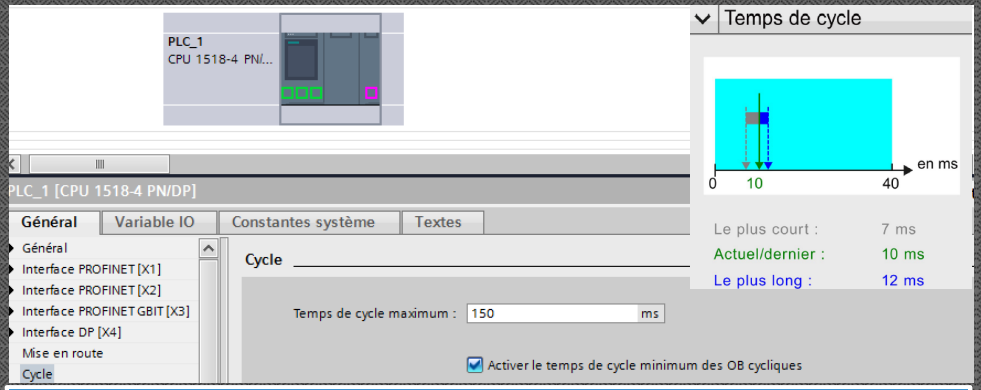
- Effacement général
- Restauration aux paramètres d'usine

I.7: Cycle API - Chien de garde - Interruptions de programme

- **Un API exécute son programme de manière cyclique.**
  - **Lecture des entrées**
  - **Traitement du programme**
  - **Écriture des sorties**

- Le temps d'exécution d'un cycle est contrôlé par une surveillance du temps appelée **chien de garde.** Si le temps de cycle dépasse le chien de garde, la CPU se mettra en défaut.





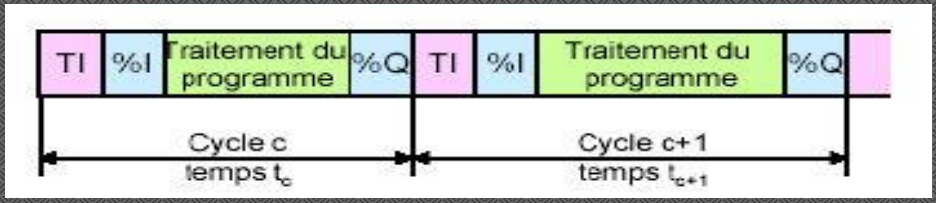
Sur certains modèles d'automates, le temps de surveillance de cycle est réglable.

I.7: Cycle API - Chien de garde - Interruptions de programme

On dissocie deux types de fonctionnement dans les API industriels :

❑Fonctionnement cyclique :

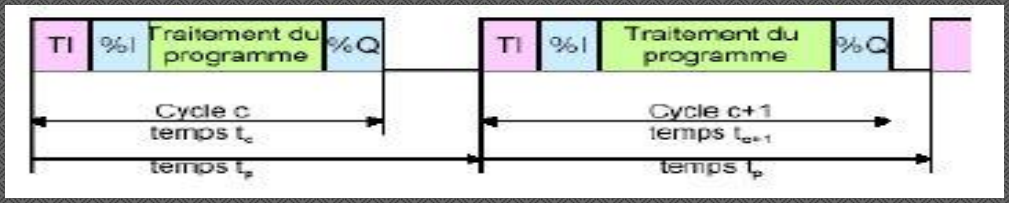
❖Les cycles s'enchaînent les uns après les autres, le temps de cycle dépend de la durée d'exécution du programme. Cette durée dépend du nombre de calculs et de la puissance de traitement de la CPU.



Attention, le temps de cycle doit être compatible avec l'application. Il est appelé également « cycle de scrutation ».

❑ Fonctionnement périodique:

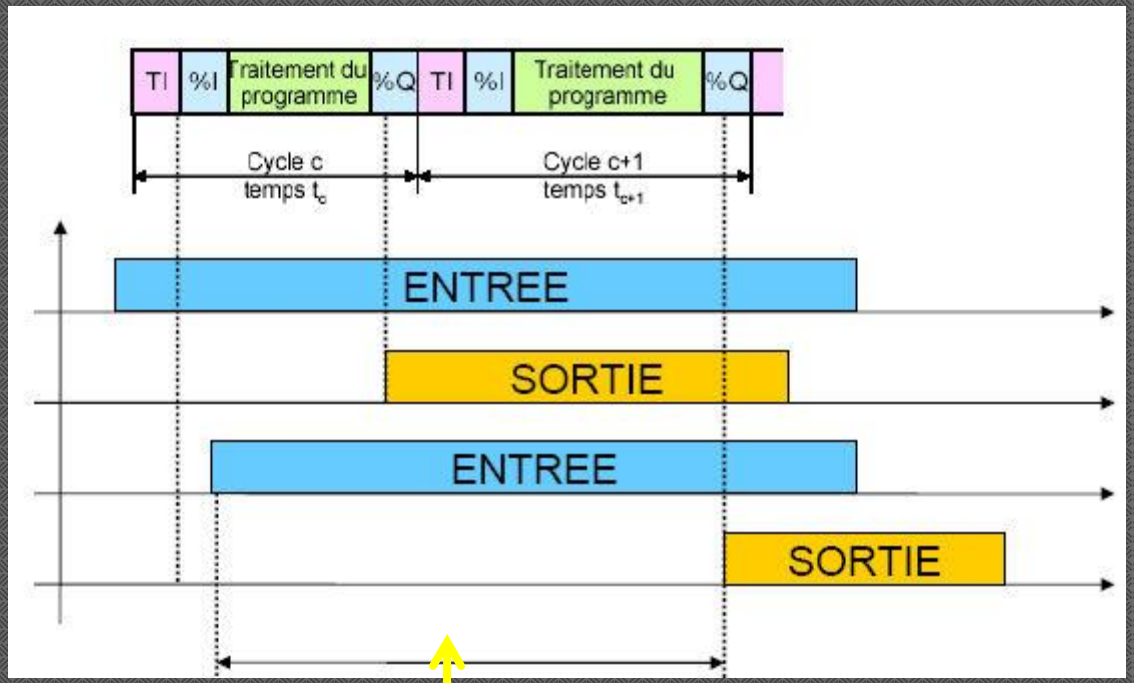
❖ Il est parfois possible de fixer le temps de cycle pour certaines parties du programme. Dans le jargon informatique, on nomme ce fonctionnement par le terme: « Interruption de programme »



Intéressant pour des cas de régulation.

I.7: Cycle API - Chien de garde - Interruptions de programme

Lorsque l'évolution d'une entrée doit entraîner l'activation d'une sortie, le temps de retard entre les 2 évènements dépend du temps de cycle.



Au maximum, 2 x le temps de cycle (2 x cycle de scrutation) en générale de l'ordre de quelques dizaines de millisecondes.

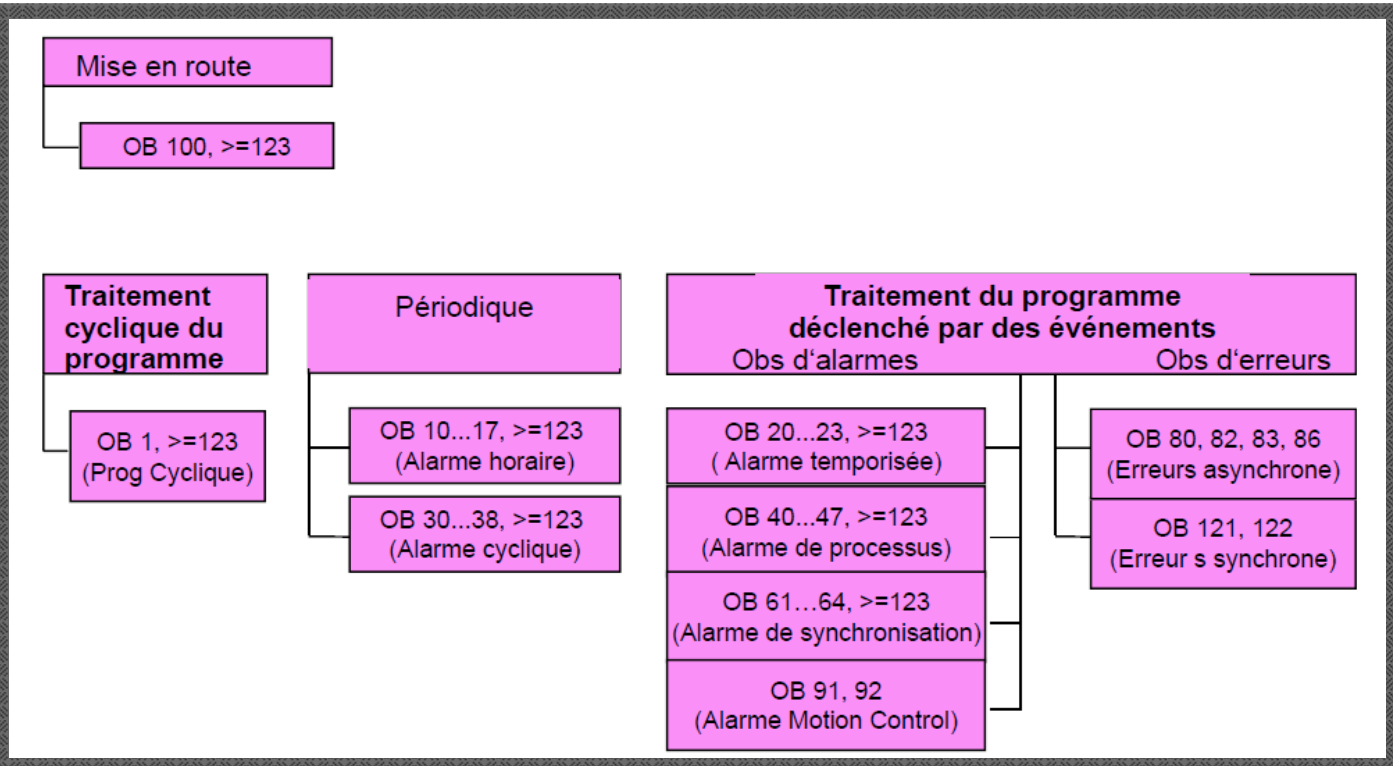
I.7: Cycle API - Chien de garde - Interruptions de programme

Exemple de fonctionnement d'un API SIEMENS :

Dans un API SIEMENS nous retrouvons également ce concept de fonctionnements cyclique et périodique. Chez SIEMENS, **ces différents fonctionnements se nomment des OB (blocs d'organisation)**.

Chez SIEMENS, OB1 (Main) pour le fonctionnement cyclique de l'API.

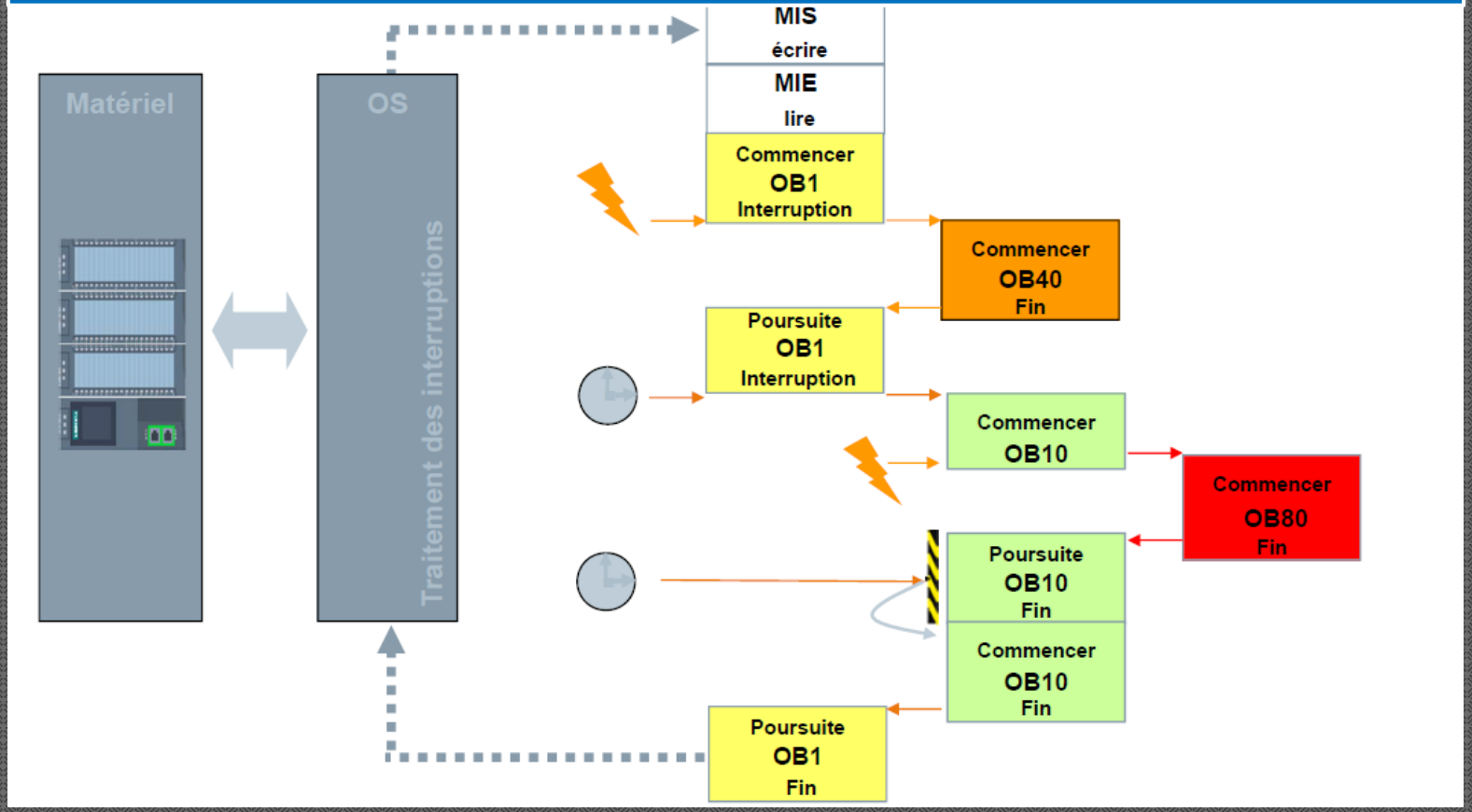
D'autres OB internes sont également mis à disposition pour l'utilisateur. (OB de démarrage, d'alarmes ou d'erreurs)



I.7: Cycle API - Chien de garde - Interruptions de programme

Exemple de fonctionnement d'un API SIEMENS :

L'exemple suivant démontre un exemple de cycle avec des interruptions.



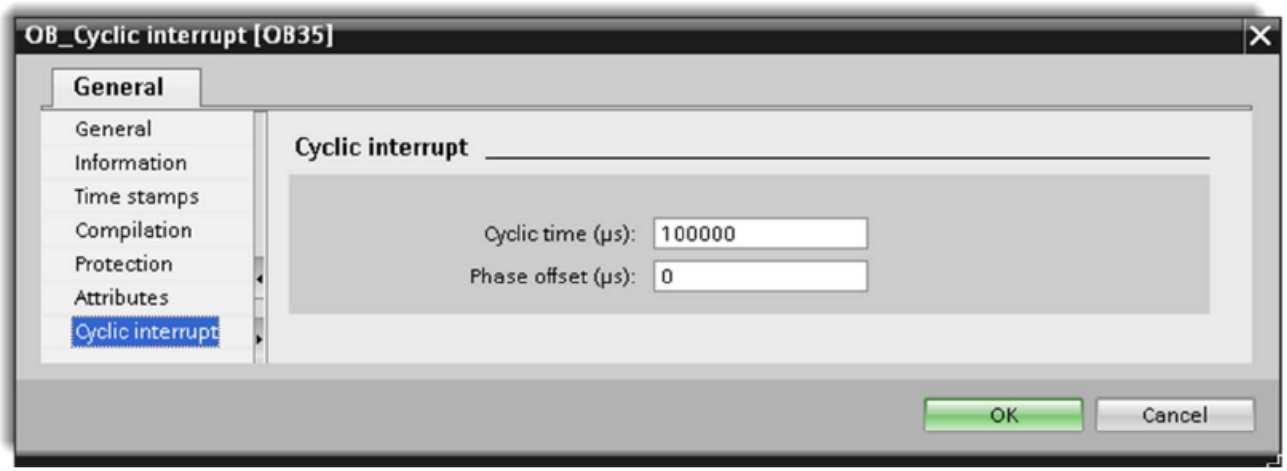
Le temps de cycle augmentera avec le nombre d'interruptions. C'est pour cela qu'il conviendra de choisir l'API en fonction du besoin de l'application.

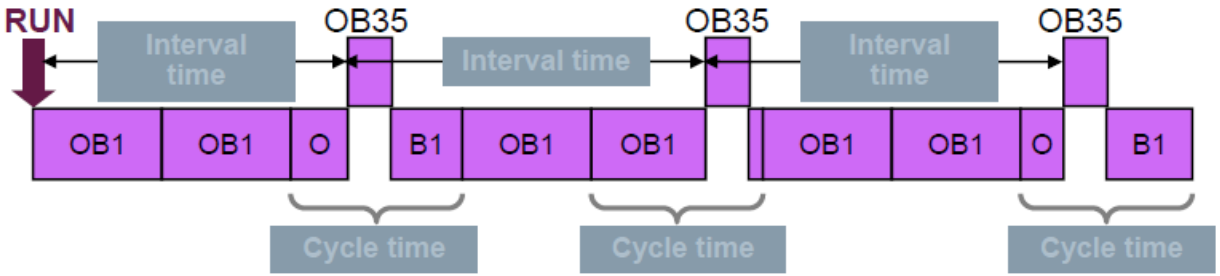
I.7: Cycle API - Chien de garde - Interruptions de programme

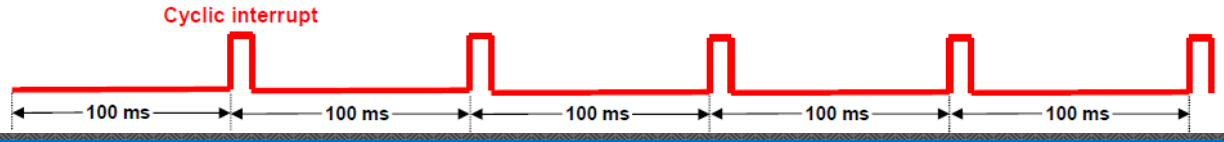
Exemple de fonctionnement d'un API SIEMENS :

Les interruptions les plus couramment utilisés sont :

- **L'interruption cyclique**
- **L' interruption sur un module d'entrées/sorties sur un seuil**





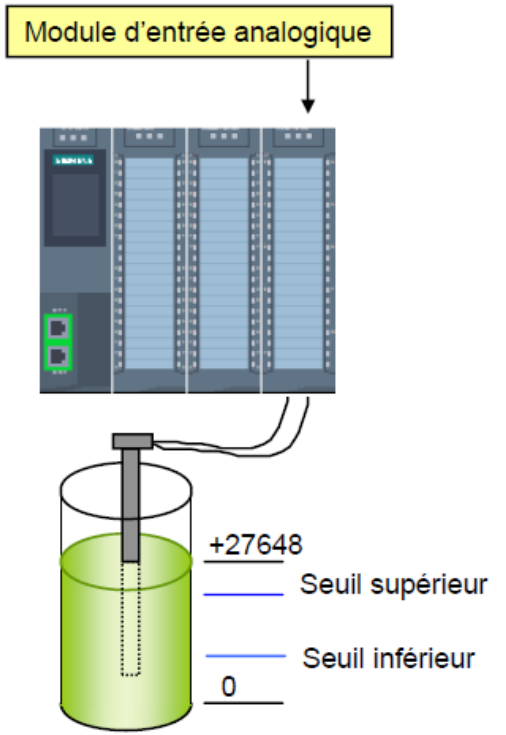


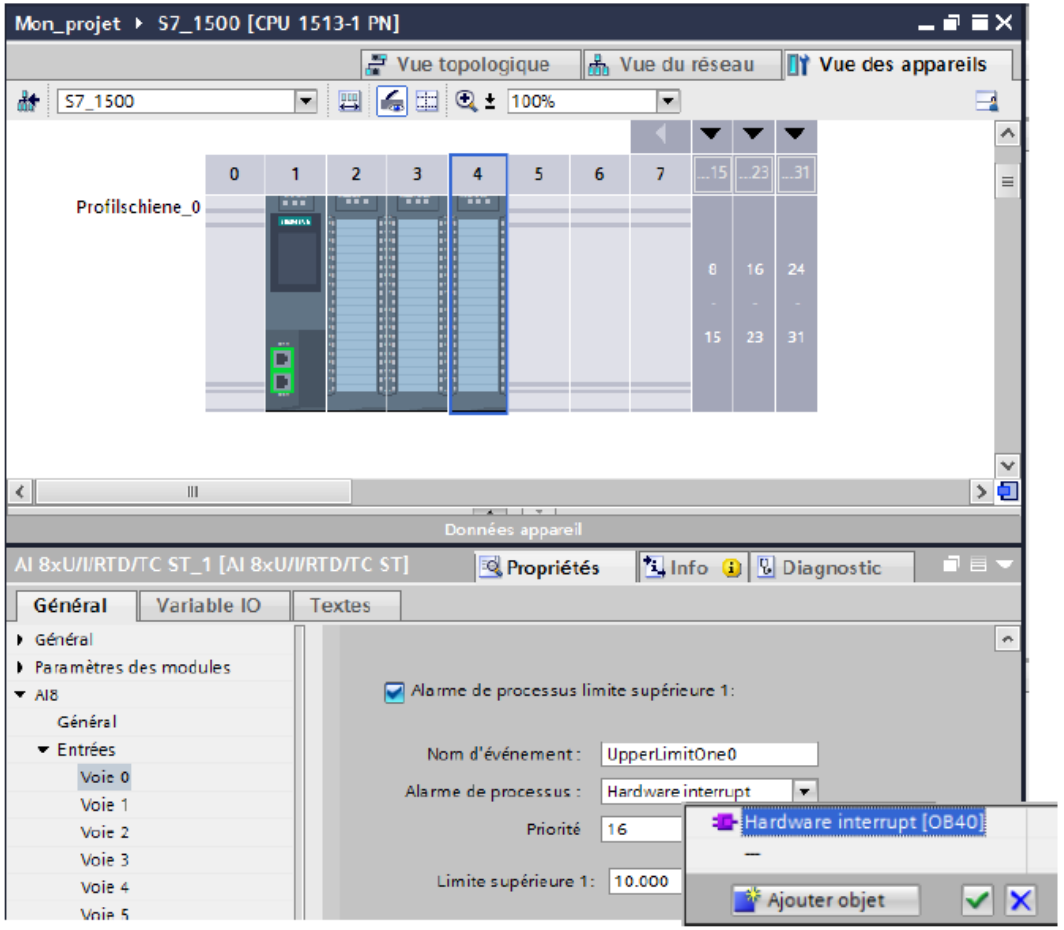
I.7: Cycle API - Chien de garde - Interruptions de programme

Exemple de fonctionnement d'un API SIEMENS :

Les interruptions les plus couramment utilisés sont :

- L'interruption cyclique
- **L' interruption sur un module d'entrées/sorties sur un seuil**

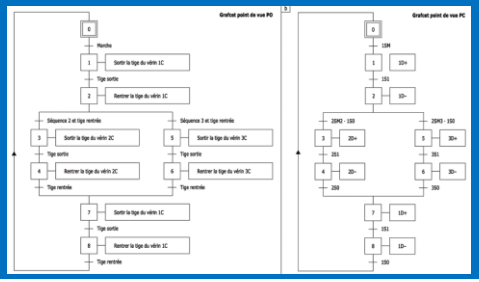




I.8: GRAFCET

Le Grafcet est un organigramme spécialisé, avec graphisme et règles particulières, utilisé pour décrire le cycle logique des système automatisés séquentiels.

GRAFCET
- GRA: GRAphe
- F: Fonctionnel de
- C: Commande
- E: Etapes
- T: Transition



Certains constructeurs proposent une programmation en Grafcet en respectant la norme: ce langage se nomme SFC

SFC
- S: Sequential
- F: Fonction
- C: Chart

I.8: GRAFCET

Le GRAFCET présente certains avantages et apprécié par les automaticiens:

❑Moyen de communication entre l'automaticien et son client, le GRAFCET est un outil utilisé pour la rédaction du cahier des charges d'un automatisme.

❑ Il est utilisé pour spécifier et concevoir le comportement souhaité de la partie commande et/ou de la partie opérative d'un système.

C'est un langage de spécification.

❑Un des points forts du GRAFCET est la facilité de passer du modèle l'implantation technologique de celui-ci dans un automate programmable industriel.

Le GRAFCET est alors un langage d'implémentation utilisé pour la réalisation de l'automatisme.

I.8: GRAFCET

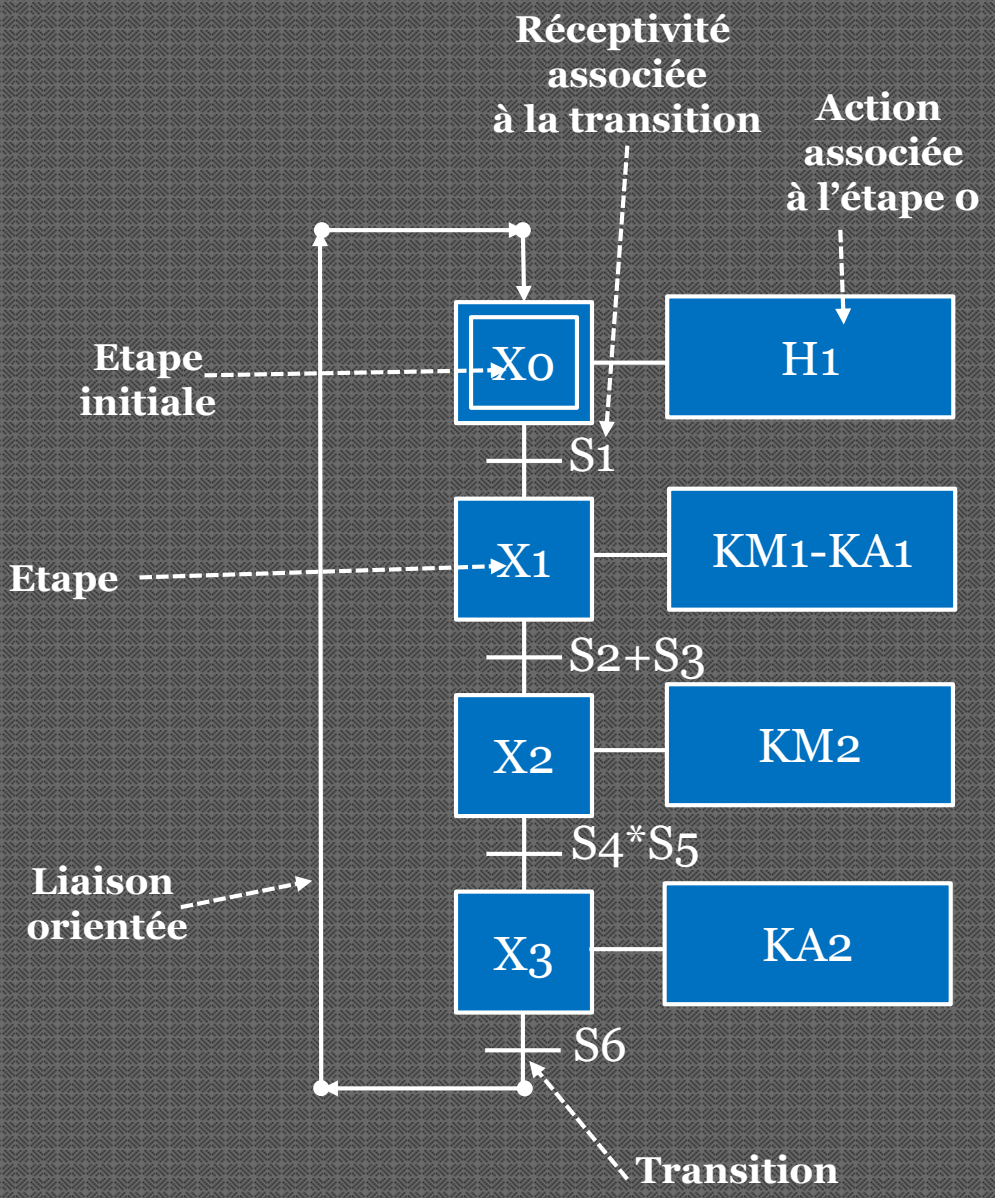
❑**Un GRAFCET est constitué :**

❖D' **ÉTAPES**, traduisant le séquencement.

❖D'**ACTIONS** qui agissent sur les actionneurs du système.

❖De **TRANSITIONS**, auxquelles sont associées des réceptivités sous forme de variables ou d'équations .

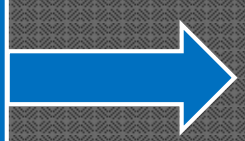
❖ De **LIAISONS ORIENTÉES**, reliant les étapes aux transitions, les transitions aux étapes et donnant un déroulement du cycle dans le sens vertical de haut en bas.



I.8: GRAFCET

❑ **ÉTAPE :**

❖Une étape correspond à un comportement stable du système. Les étapes sont numérotées dans l'ordre croissant. À chaque étape peuvent correspondre une ou plusieurs actions. Une étape est soit **active** soit **inactive**.



❑ **ÉTAPE INITIALE :**

❖La (ou les) étape(s) initiale(s) caractérisent l'état du système au début du fonctionnement. Elle se caractérise par le double carré.

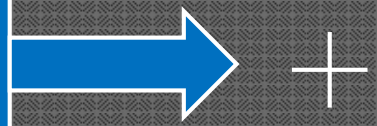
❑ **ACTIONS:**

❖**Les actions sont réalisées** lorsque **l'étape associée à l'action est active**. Il existe différents types d'actions.

I.8: GRAFCET

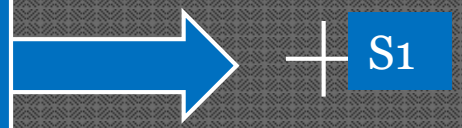
❑ **TRANSITION :**

❖Les transitions indiquent les possibilités d'évolution, à chaque transition est associée une réceptivité.



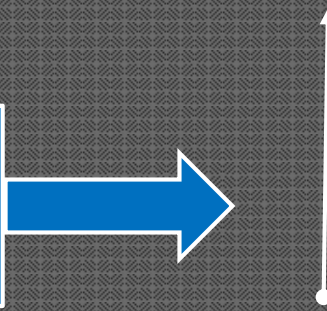
❑ **RÉCEPTIVITÉ :**

❖La réceptivité est la condition logique qui permet l'évolution. Si la réceptivité est vraie (=1), le cycle peut évoluer. Les réceptivités sont des comptes-rendus en provenance de la partie opérative ou des consignes en provenance du pupitre ou d'une IHM.



❑ **LIAISONS ORIENTÉES:**

❖**Un GRAFCET se lit de haut en bas.** Pour remonter, il est nécessaire d'indiquer le sens par une flèche.

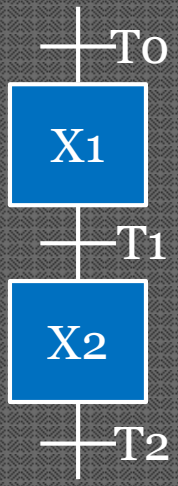


I.8: GRAFCET

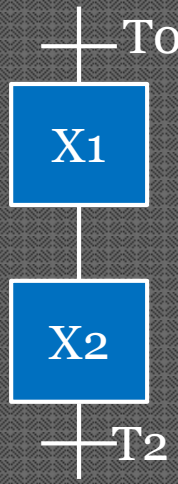
Un GRAFCET est un langage graphique de programmation, **les utilisateurs doivent par conséquent respecter une syntaxe.**

- L'alternance étape-transition doit être respectée quel que soit le chemin du graphe parcouru.
- Deux étapes ne doivent jamais être reliées directement.
- Deux transitions ne doivent jamais être reliées directement.

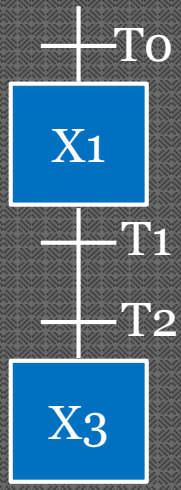
Oui, la syntaxe est correcte



Non, il manque une transition entre X1 et X2



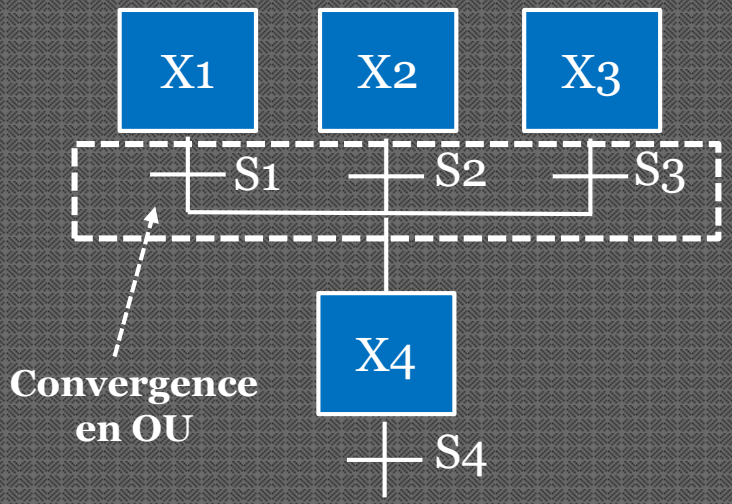
Non, il manque une étape entre T1 et T2

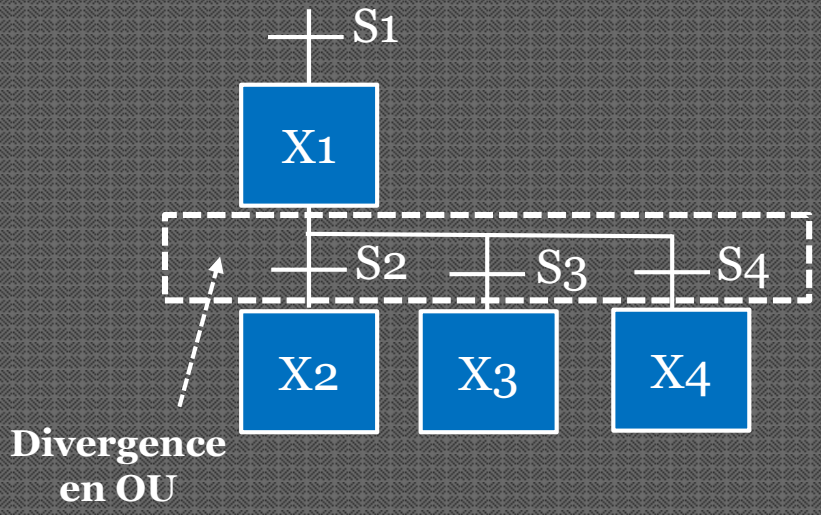


I.8: GRAFCET

Convergences et divergences :

OU:
On appellera un simple trait horizontal un OU logique. S'il est avant l'étape on dira qu'il réalise une **convergence en OU** et s'il est en sortie on dira qu'il réalise une **divergence en OU**





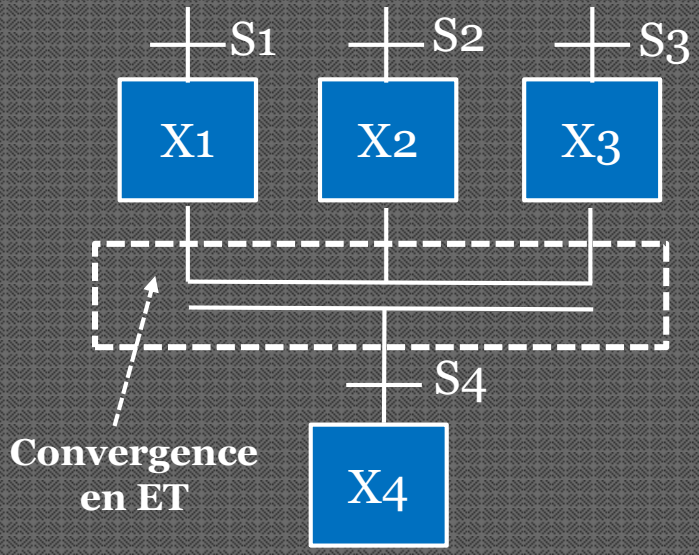
I.8: GRAFCET

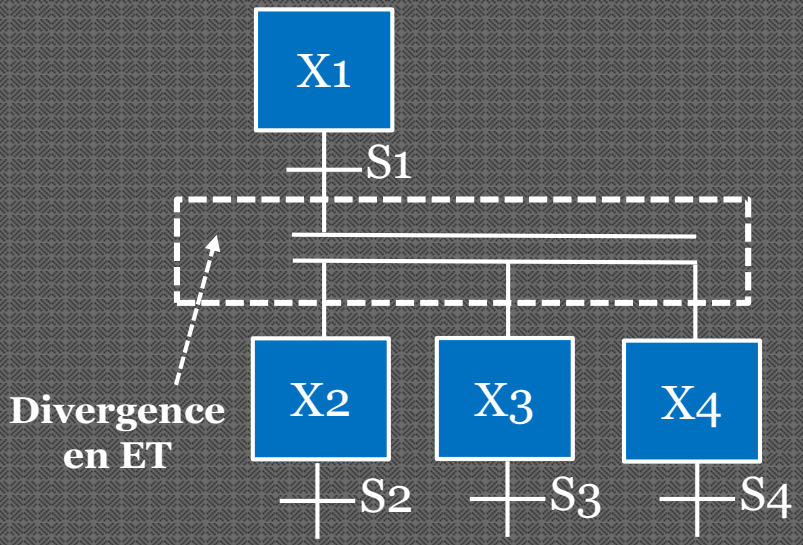
Convergences et divergences :

ET:

Une transition peut être liée en amont à un double trait horizontal qui s'appelle alors une **convergence en ET**. Si c'est en aval que l'on a un double trait horizontal, on parle d'une **divergence en ET**.

- Lorsque l'on a des doubles traits horizontaux, la terminologie associée est le **parallélisme structural.**
- Une transition est validée lorsque toutes les étapes immédiatement précédentes reliées à cette transition sont actives





I.8: GRAFCET

Règles d 'évolution :

Le GRAFCET est un outil normalisé qui respecte 5 règles d'évolution.

Les constructeurs qui implémentent le langage SFC (sequential fonction chart) dans leur cible matériel et outil logiciel doivent tenir compte de ces règles d'évolution.

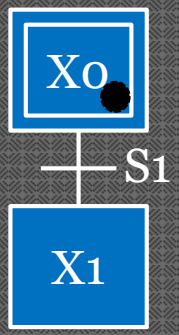
Remarque: Nombre d'industriels n'utilisent pas le langage SFC, ils transforment le GRAFCET dans d'autres langages comme: **le Ladder, le Logigramme, le langage structure ou encore le List**. Néanmoins, cette adaptation de langage doit respecter les 5 règles d'évolution.

Nous reviendrons dans le chapitre III.9, pour présenter différences méthodes de programmation d'un GRAFCET dans des langages normalisés.

I.8: GRAFCET

❑ Règle n°1 : Situation initiale

- ❖Elle caractérise le comportement initial de la partie commande vis-à-vis de la partie opérative, de l'opérateur et/ou des éléments extérieurs.
- ❖Elle correspond aux étapes actives au début du fonctionnement.
- ❖Du point de vue programmation, toutes les étapes initiales doivent s'activer automatiquement.



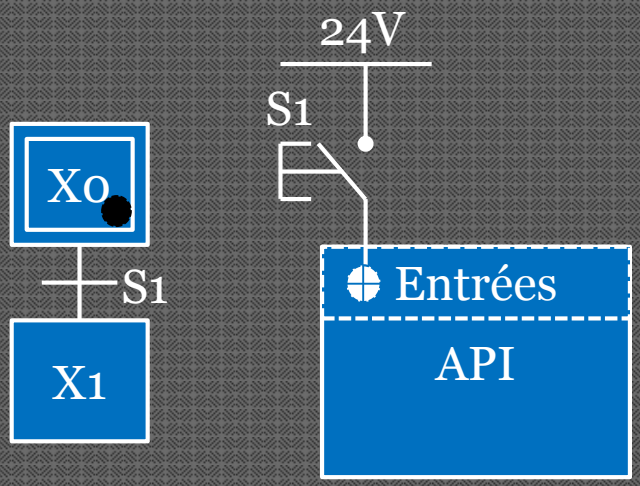
Au chargement du GRAFCET dans l'automate, l'étape initiale X0 est active. Cela est illustré par le jeton● positionné sur l'étape.

I.8: GRAFCET

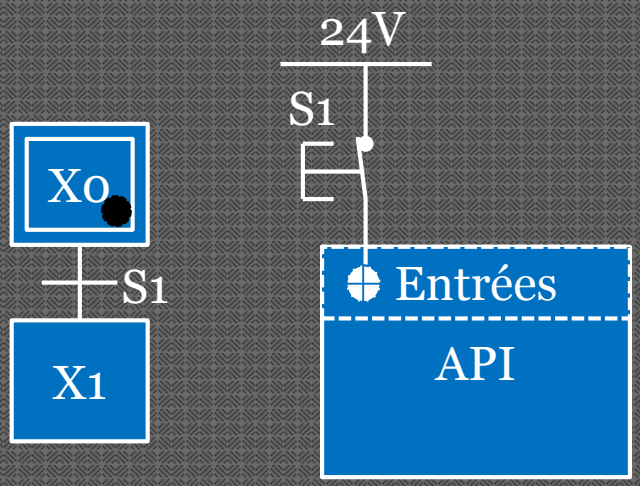
- **Règle 2 : l'évolution de la situation d'un grafcet correspondant au franchissement d'une transition ne peut se faire :**

  - que lorsque cette transition est validée
  - et que la réceptivité associée à cette transition est vraie.

Lorsque ces deux conditions sont réunies, la transition devient franchissable et elle est obligatoirement franchie.



S1=0, la réceptivité est fausse, la transition n'est pas franchissable

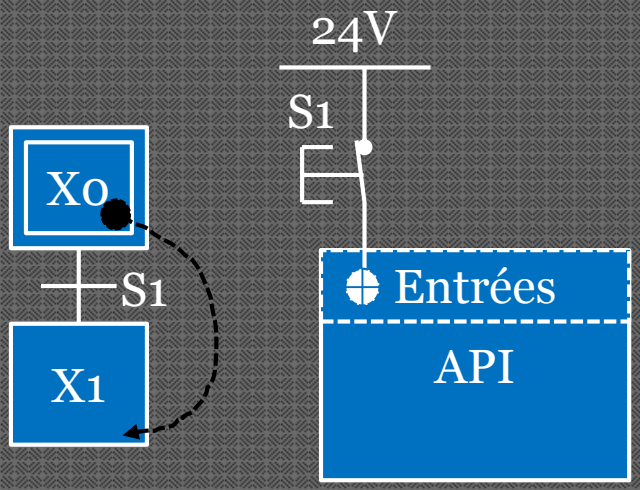


S1=1, la réceptivité est vraie, la transition devient franchissable

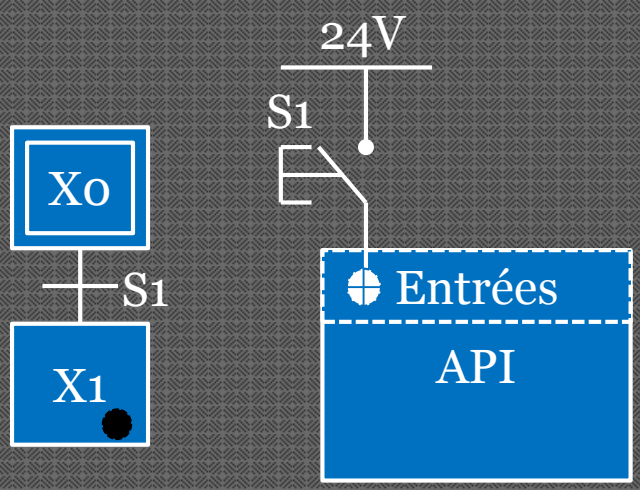
I.8: GRAFCET

- **Règle 3 : Le franchissement d'une transition provoque :**

  - la désactivation de toutes les étapes immédiatement précédentes reliées à cette transition.
  - l'activation de toutes les étapes suivantes reliées à cette transition



S1=1, le jeton passe de X0 à X1



X0 se désactive X1 s'active

I.8: GRAFCET

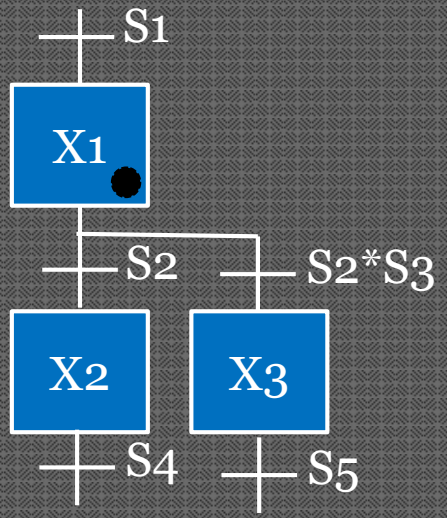
❑ **Règle n°4 :**

❖ Les transitions simultanément franchissables sont simultanément franchies.

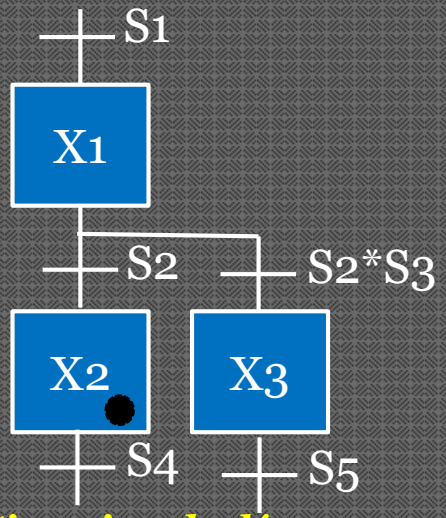
Exemple

❖ Si la condition « S2 » devient vraie sans que la condition « S3» le soit, l'étape X2 est activée.

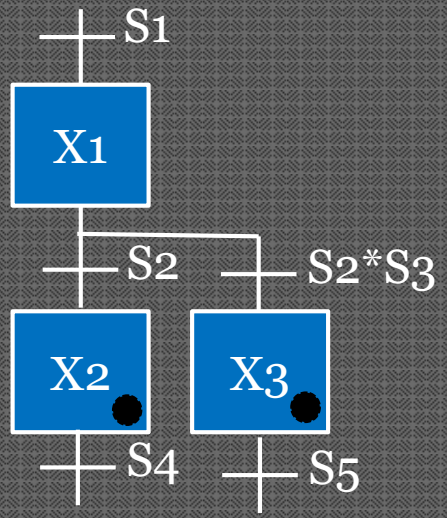
❖ Si la condition « S3 » est préalablement vraie avant que la condition «S2» le devienne, les deux étapes X2 et X3 sont activée simultanément.



Situation de départ X1=1 S2=0 et S3=0 pas d'évolution



Situation de départ X1=1 S2=0 et S3=0 puis S2=1 alors X1 se désactive et X2 s'active

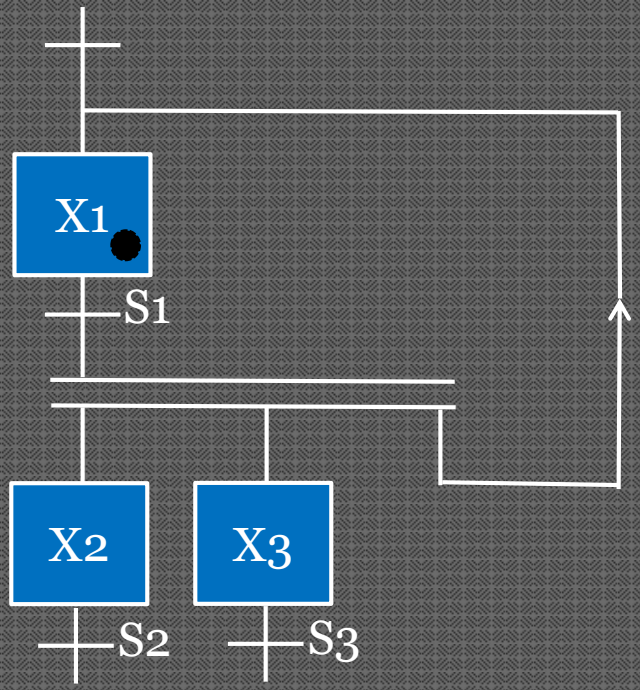


Situation de départ X1=1 S2=0 et S3=1 puis S2=1 alors X1 se désactive et X2 et X3 s'activent

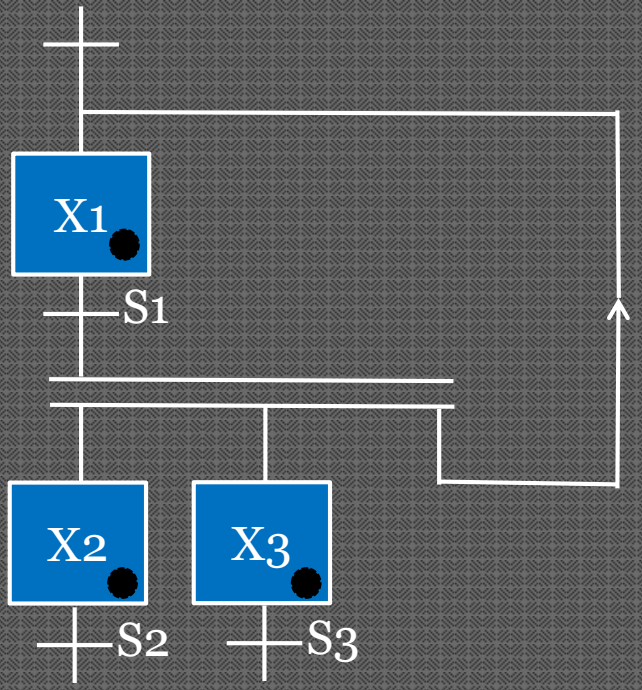
I.8: GRAFCET

❑ Règle n° 5 : Activation et désactivation simultanée d'une étape

❖Si au cours du fonctionnement, la même étape est simultanément activée et désactivée, elle reste active.



Situation de départ
X1=1 S1=0
Pas d'évolution



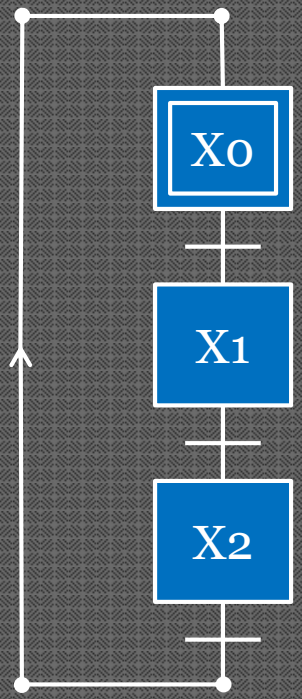
Situation de départ
X1=1 S1=0
puis S1=1
alors X1 reste active (règle 5)
et X2 et X3 s'activent également

I.8: GRAFCET

Structures de base :

❑ Séquence unique ou linéaire :

❖Une séquence unique est composée d'une suite d'étapes pouvant être activées les unes après les autres. Chaque étape n'est suivie que par une seule transition et chaque transition n'est validée que par une seule étape.

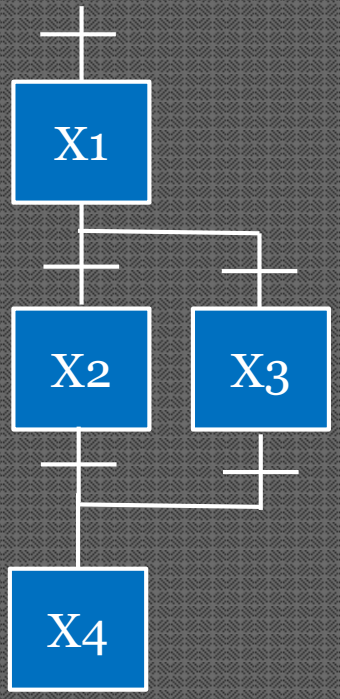


I.8: GRAFCET

Structures de base :

❑ Sélection de séquence :

- ❖Une étape peut être reliée à plusieurs transitions en amont (convergence) ou en aval (divergence).
- ❖Le « **OU** » permet de prendre en compte un choix, un "aiguillage " entre deux séquences.

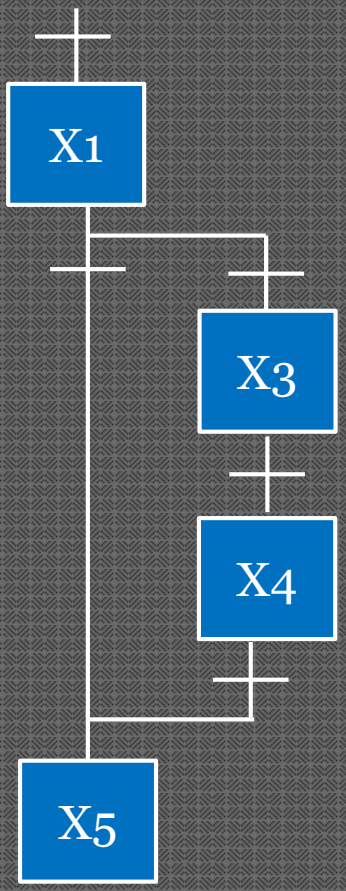


I.8: GRAFCET

Structures de base :

❑ Saut d'étape(s) :

❖Un saut d'étapes va permettre de shunter une partie du Grafcet.

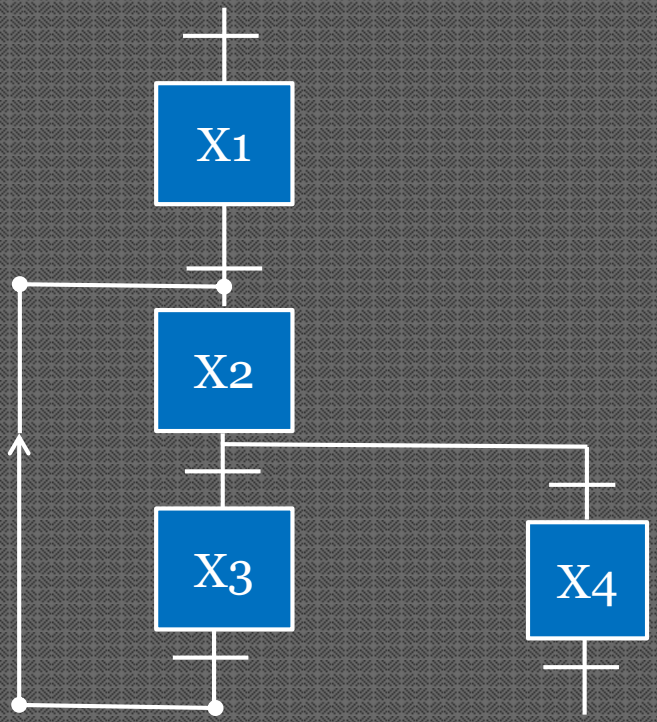


I.8: GRAFCET

Structures de base :

❑ Reprise de séquence

❖Une partie du grafcet peut être exécutée à plusieurs reprises.



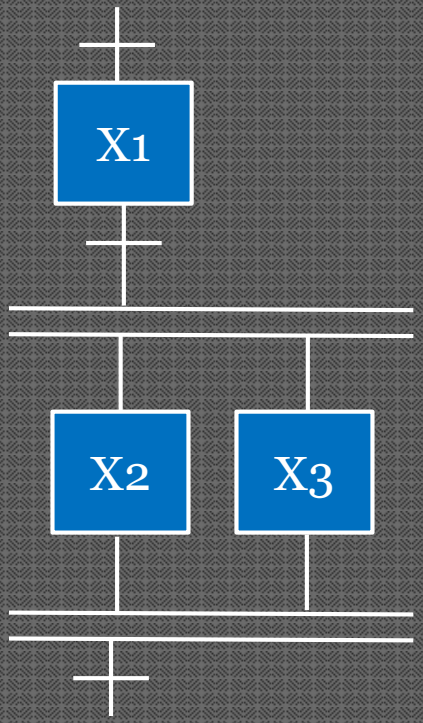
I.8: GRAFCET

Structures de base :

❑ Séquence simultanée :

❖Une transition peut supporter plusieurs étapes en amont et plusieurs étapes en aval :c'est la notion de « **ET** ».

❖Le franchissement d'une transition conduit à activer plusieurs séquences en même temps, ces séquences sont dites séquences simultanées et évoluent à leur rythme.

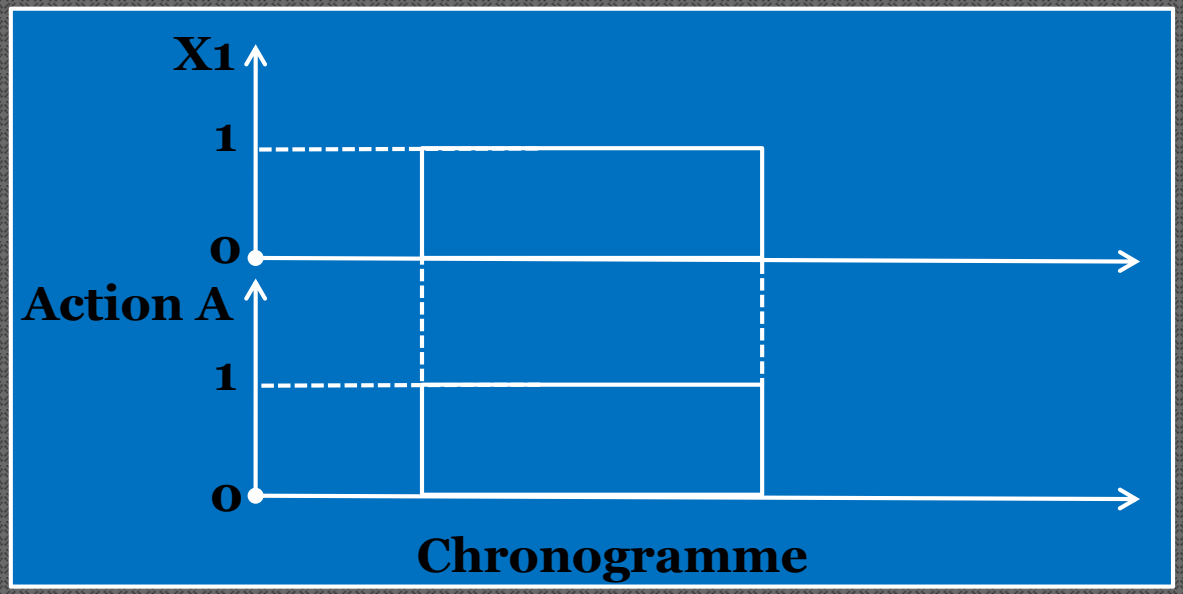


I.8: GRAFCET

Classification des actions :

❑ Action non mémorisée :

❖L'exécution de l'action se poursuit tant que l'étape à laquelle elle est associée reste active.

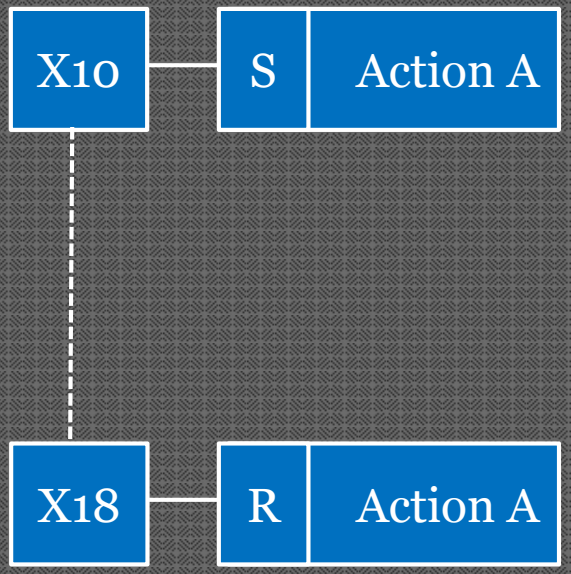


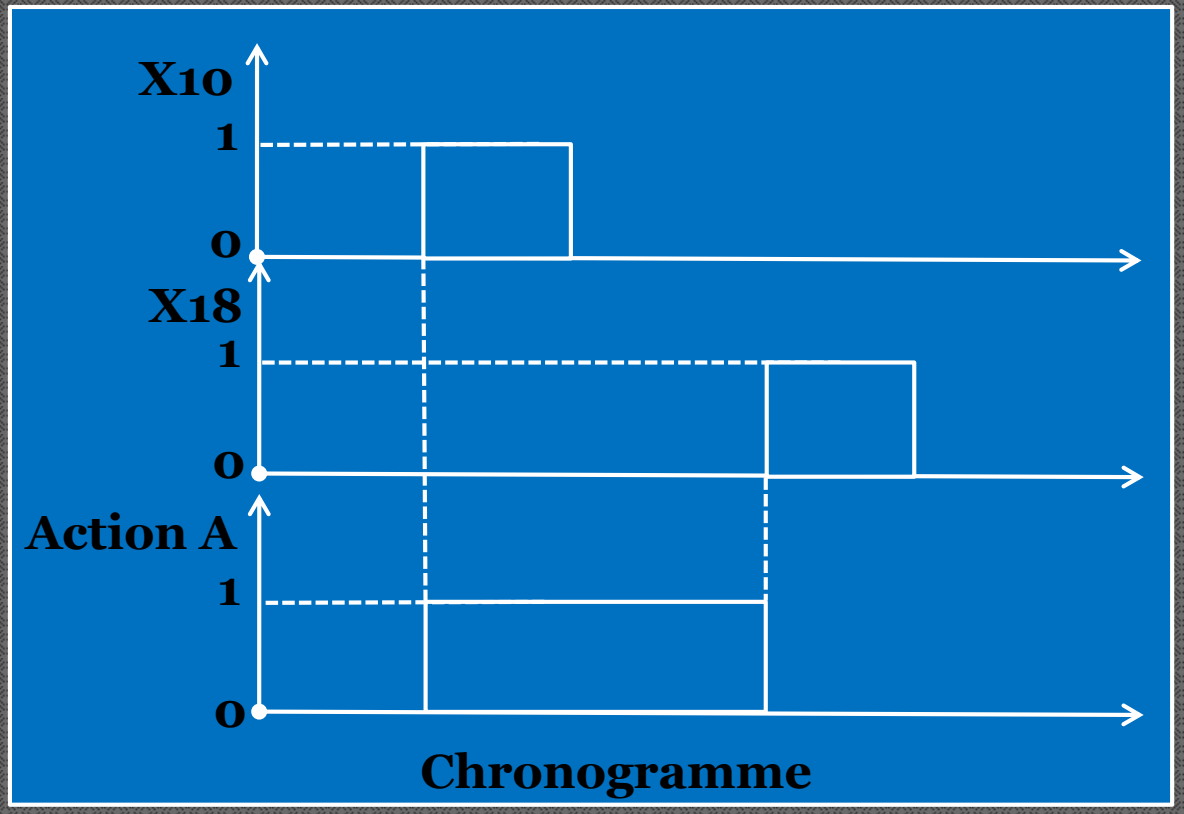
I.8: GRAFCET

Classification des actions :

❑ Action mémorisée :

❖Elle est associée à deux étapes, une étape de déclenchement : **opérateur S** comme « **set** », une étape d'arrêt : **opérateur R** comme « **reset** » . L'action mémorisée est exécutée dès que l'opérateur S est activé et reste active jusqu'à l'opérateur R.



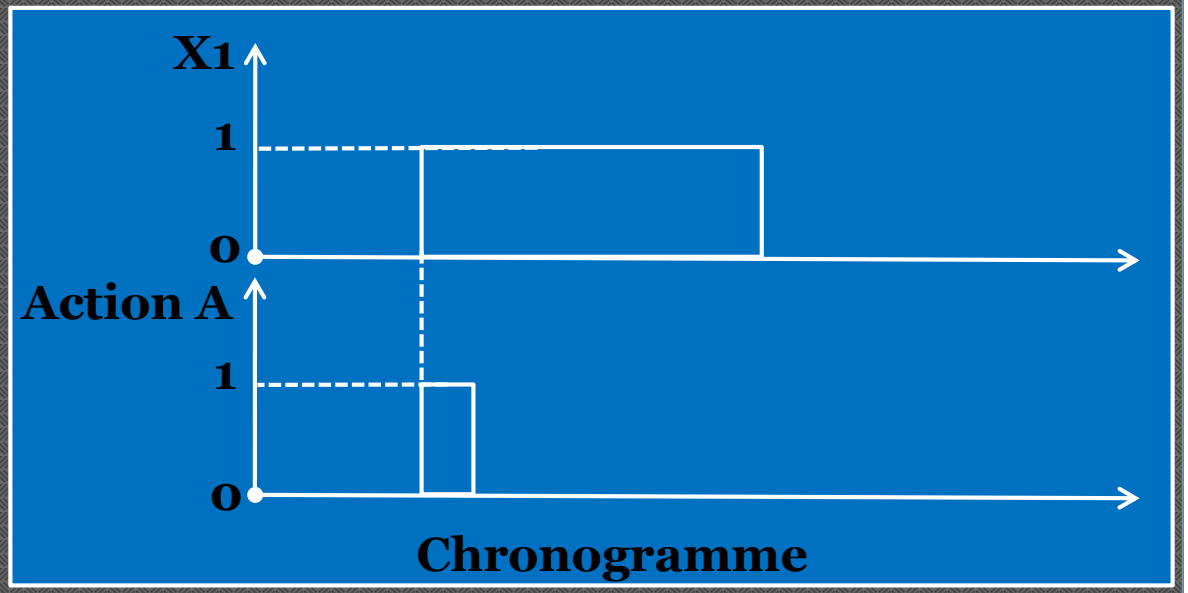


I.8: GRAFCET

Classification des actions :

❑ **Action impulsionnelle :**

❖C'est une action de durée très petite dont la valeur est sans importance mais suffisante à priori pour obtenir l'effet souhaité

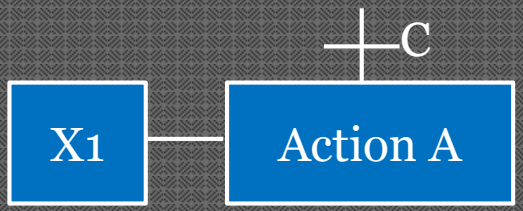


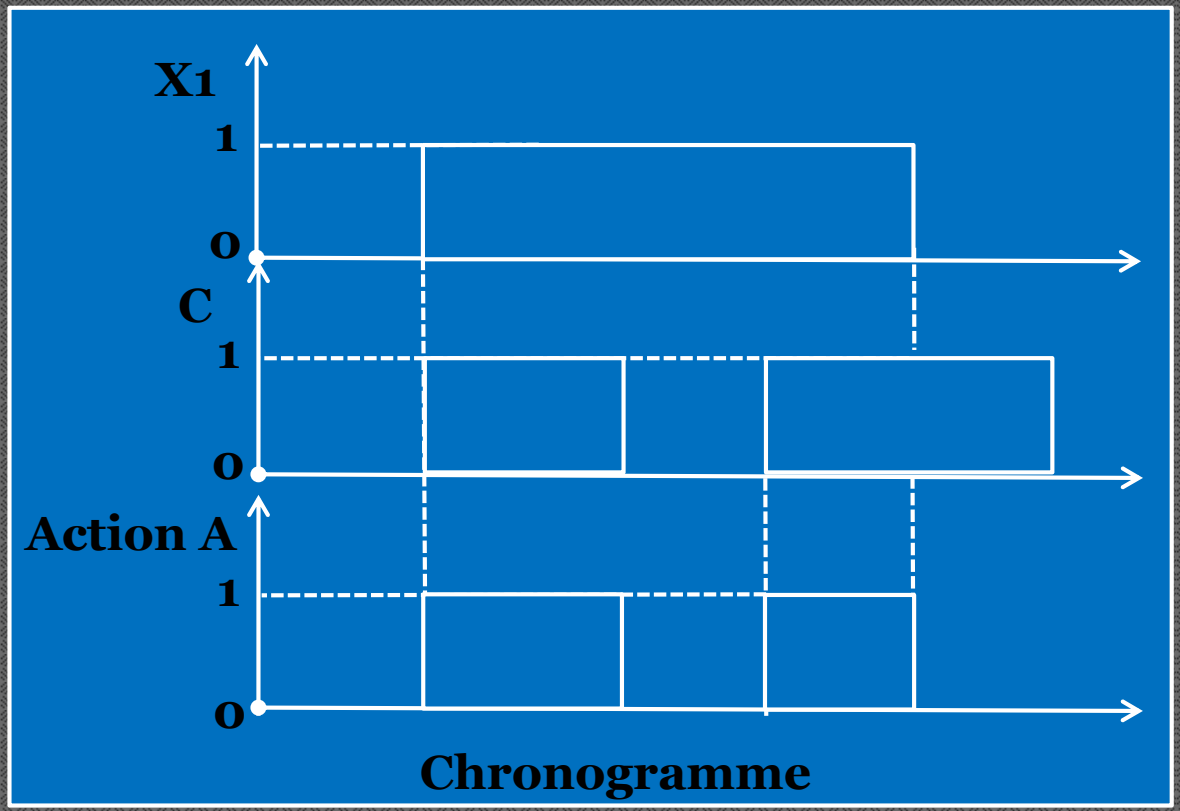
I.8: GRAFCET

Classification des actions :

❑ **Action conditionnelle:**

❖C'est une action non mémorisée dont l'exécution est soumise à une condition logique.



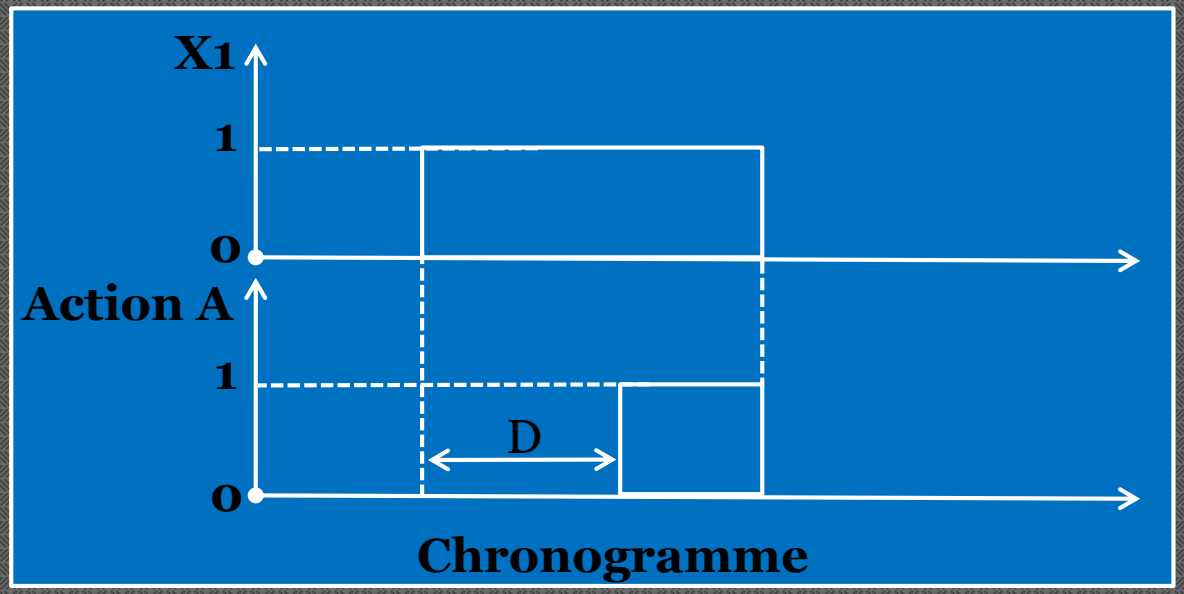


I.8: GRAFCET

Classification des actions :

- **Action retardée :**

  ❖C'est une action toujours associée à une étape mais qui est vrai à l'issu d'un temps défini par l'utilisateur.

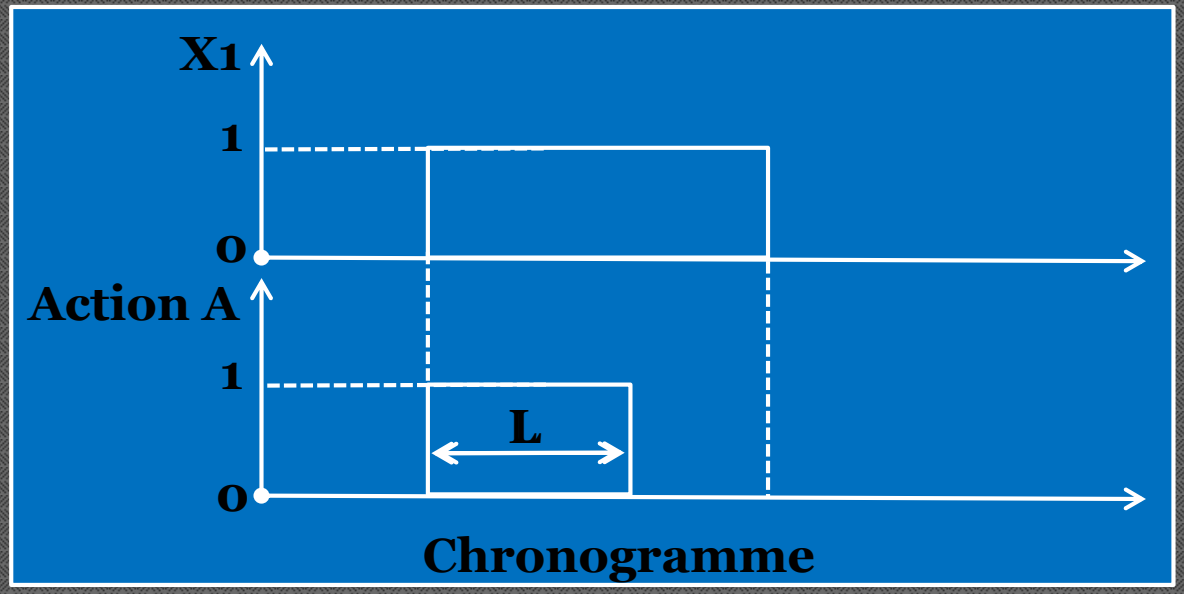


I.8: GRAFCET

Classification des actions :

- **Action limitée dans le temps :**

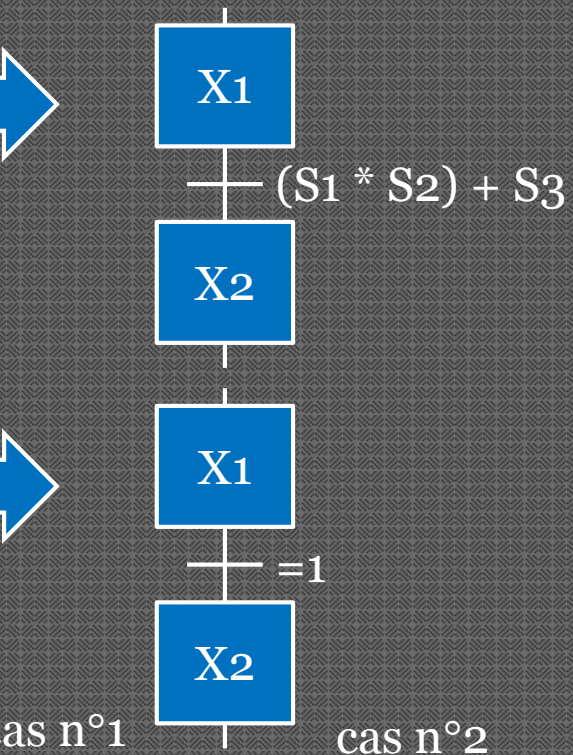
  ❖C'est une action toujours associée à une étape mais qui est vrai que pendant un temps défini par l'utilisateur.



I.8: GRAFCET

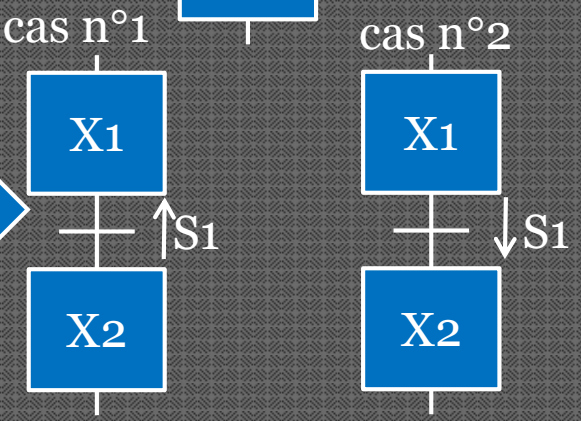
Classification des transitions :

- **Réceptivité (équation logique):**
  La transition X1 – X2 est franchie si l'équation (S1 * S2) + S3 est vraie.



- **Réceptivité (toujours vraie) :**
  La transition X1 – X2 est franchie dès que le l'étape X1 est active.

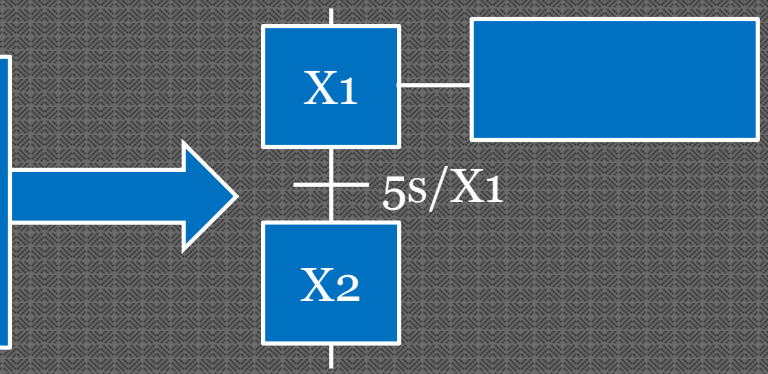
- **Réceptivité (événements fronts):**
  La transition X1 – X2 est franchie:
  - lors d'un **front montant** sur S1 (cas n°1),
  - ou lors d'un **front descendant** sur S1 (cas n°2).



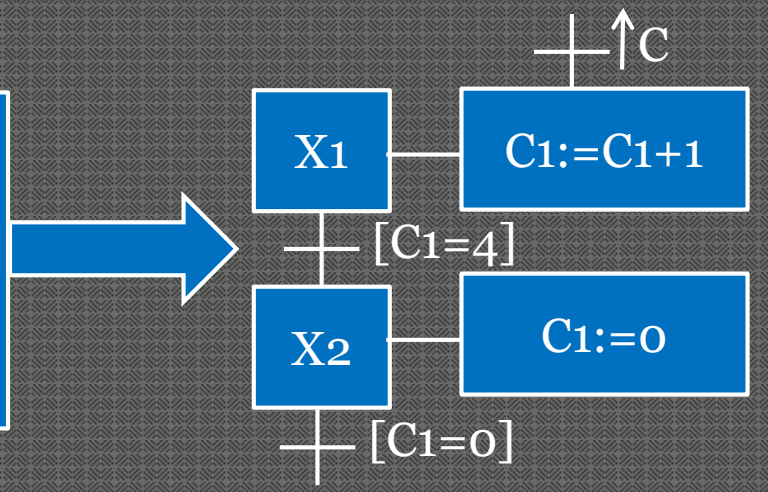
I.8: GRAFCET

Classification des transitions :

❑ **Réceptivité (comparaison d'un temps associé à une temporisation):**
La transition X1 – X2 est franchie lorsque la temporisation, démarrée à l'étape X1 est écoulée, soit au bout de 5s.
Remarque: **la temporisation doit être également programmée en action.**



❑ **Réceptivité (comparaison d'une valeur associée à du comptage):**
La transition X1- X2 est franchie
si le compteur C1 a atteint la valeur 4.
Remarque: le compteur est incrémenté de 1 sur l'étape X1 à condition d'avoir le front montant C.



I.8: GRAFCET

Grafcet(s) hiérarchisés :

Les Systèmes Automatisés de production sont de plus en plus complexes, afin de simplifier l'étude, la mise en œuvre et la maintenance du système, il est nécessaire de **structurer la partie commande et la partie opérative.**

L'objectif essentiel de la structuration est de permettre une approche progressive du fonctionnement d'un système automatisé, tant au niveau de l'analyse qu'au niveau de la représentation.

Dans l'analyse structurée, **le grafcet global est décomposé en modules**, chacun de ces modules correspond à une fonction du système (Sécurité, modes de marche, etc.) ou à une sous partie de la Partie Opérative (Poste 1, Poste 2, Poste 3, etc)

La structuration est :

- soit Hiérarchique (**GRAFCET Maître, GRAFCET Esclave**)
- soit sans hiérarchie (communication entre 2 postes).

Les commandes de **forçage** et **figeage** de grafcet, sont des moyens supplémentaires qui permettent de **préciser la hiérarchie des différents grafcets.**

I.8: GRAFCET

Grafcet(s) hiérarchisés :

Les principaux GRAFCETS que l'on retrouve souvent :

- **GRAFCET de surveillance (de sécurité) :** Ce GRAFCET décrit l'ensemble des procédures de sécurité du système, c'est le GRAFCET hiérarchiquement le plus important. L'arrêt d'urgence et les procédures de mise en route sont décrits dans ce GRAFCET.

- **GRAFCET de conduite (ou GRAFCET des Modes de Marches) :** Ce GRAFCET décrit l'ensemble des procédures de Marches (auto, Cycle/Cycle, Manuel,...) et des arrêts normaux.

- **GRAFCET de maintenance** : Précise les procédures d'intervention de l'opérateur et de réglage de la partie opérative.

- **GRAFCET de Production** : Ce GRAFCET est le niveau de description du fonctionnement normal de l'automatisme. Ce GRAFCET est en général décomposé en plusieurs taches représentant les différentes fonctions de l'automatisme.

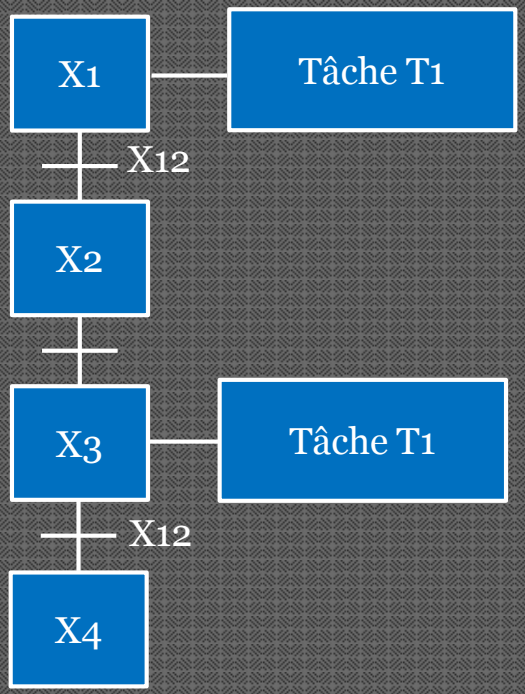
I.8: GRAFCET

Grafcet(s) hiérarchisés :

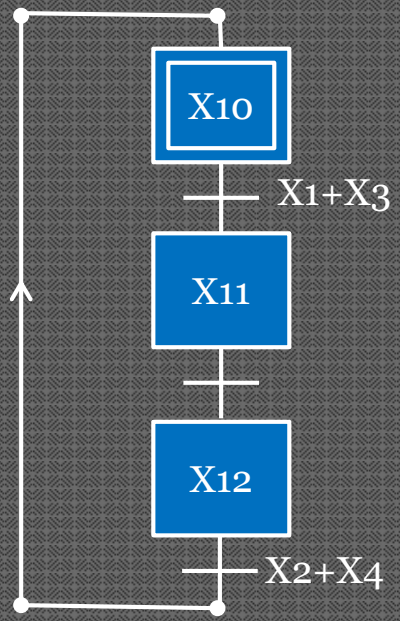
Comme évoqué précédemment, les GRAFCET(s) hiérarchisés forment une structure **de type maître/esclave** dans laquelle le **GRAFCET maître** donne des ordres à un ou plusieurs **GRAFCET esclaves.**

On parle alors de **GRAFCET de tâches** ou de sous-programmes GRAFCET.

Les GRAFCET esclaves renvoient un accusé d'exécution en fin de tâche.



GRAFCET maître



GRAFCET de tâche T1

I.8: GRAFCET

Grafcet(s) hiérarchisés :

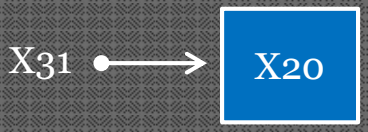
Ordre de figeage ou de forçage

- Il faut définir des graphes de niveau hiérarchique différents
- Un graphe peut figer ou forcer un autre dans une situation donnée si il est de niveau hiérarchique supérieur.

Forçage :

Grafcet G3

Grafcet G2



A l'étape X31 du GRAFCET G3, il y a forçage du GRAFCET G2 à l'étape X20. Plus d'évolution sur G2 tant que X31=1

Figeage :

Grafcet G3

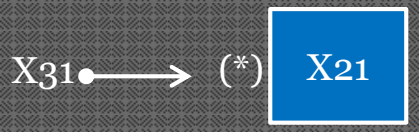
Grafcet G2

L'activation de l'étape X31 du GRAFCET G3 fige le GRAFCET G2 dans sa situation courante. Plus d'évolution sur G2 tant que X31=1

Figeage :

Grafcet G3

Grafcet G2



L'activation de l'étape X31 du GRAFCET G3 figera le GRAFCET G2 sur l'étape X21. par rapport au forçage, le grafcet G2 continue d'évoluer mais se boquera sur l'étape X21

PARTIE II:
Matériel & outils logiciels

II.1: Structure de contrôle de type API (petites applications)

Dans cette partie, nous présenterons les différentes gammes proposées par les constructeurs du marché.

Nous observerons 3 constructeurs du marché :

- **ALLEN-BRADLEY**

https://ab.rockwellautomation.com/

- **SCHNEIDER**

https://www.schneider-electric.fr/fr/

- **SIEMENS**

https://support.industry.siemens.com/cs/start?lc=fr-FR

Comme exemples plus détaillés, nous présenterons les produits SIEMENS n°1 mondial:
CPU, Technologie des cartes d'E/S, etc.

II.1: Structure de contrôle de type API (petites applications)

Pour les petites applications industrielles simples, les constructeurs proposent des petits automates (relais intelligent compact)

MicroLogix

Micro800

Plusieurs modèles : MicroLogix/Micro800
Module programmable compact avec possibilité d'extension
Programmation par soft:
Connected Components Workbench
Langages:
Ladder/logigramme/ST

Zélio Logic

ZELIO LOGIC
Module programmable compact avec possibilité d'extension
Programmation par soft:
Zélio Soft
Langages:
Ladder/logigramme

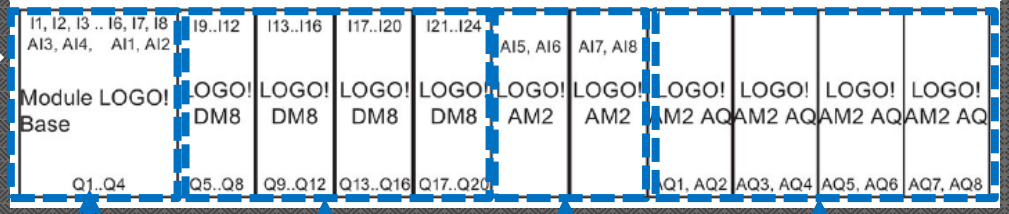
Logo

LOGO
Module programmable compact avec possibilité d'extension
Programmation par soft:
LOGO! Soft Comfort
Langage:
Ladder/logigramme

II.1: Structure de contrôle de type api (petites applications)

API SIEMENS LOGO : Exemple de configuration et caractéristiques

Exemple d'architecture avec des modules d'ES en extension



CPU compact
8 E-TOR
2 E-ANA
4 S-TOR

4 modules
4 E-TOR
4 S-TOR

2 modules
2 E-ANA

4 modules
2 S-ANA

LOGO est un module logique intelligent, utilisé pour la réalisation de solutions de contrôle/commande dans le cadre de petits projets d'automatisation.

Quelques avantages: facilité de montage, simplicité du câblage et programmation aisée.

Programmation par clavier ou par logiciel, il permet de réaliser rapidement de nombreuses solutions pour des machines et des installations simples en domotique ou pour diverses applications, y compris dans le domaine privé.

Fonctions intégrées:
- Serveur Web intégré
- Communication Ethernet
- Micro carte SD (sauvegarde de données)

Programmation Logo SOFT
Ladder et logigramme

Type de CPU	LOGO
Compact	●

Type de cartes	LOGO
DI/DQ	●
AI/AQ	●

II.1: Structure de contrôle de type api (petites applications)

API SIEMENS LOGO : Exemples de caractéristiques PLC et modules d'E/S

Exemples: 2 modèles de LOGO

Caractéristiques techniques

Numéro d'article	6ED1052-1CC01-0BA8 LOGO! 24CE, 8E(4EA)/4S TOR, 400 BLOCS	6ED1052-1MD00-0BA8 LOGO!12/24RCE, 8E(4EA)/4S TOR, 400 BLOCS
Avec afficheur	Oui	Oui
Type de configuration/Fixation		
Montage	sur rail DIN sym. 35 mm, largeur de 4 unités de châssis	sur rail DIN sym. 35 mm, largeur de 4 unités de châssis
Tension d'alimentation		
Valeur nominale (CC)		
• 12 V CC		Oui
• 24 V CC	Oui	Oui
• 115 V CC		
• 230 V CC		
Plage admissible, limite inférieure (CC)	20,4 V	10,8 V
Plage admissible, limite supérieure (CC)	28,8 V	28,8 V
Valeur nominale (CA)		
• 24 V CA		
• 115 V CA		
• 230 V CA		
Heure		
Minuteries		
• Nombre	190	190
• Réserve de marche	480 h	480 h
Entrées TOR		
Nombre d'entrées TOR	8; dont 4 utilisables en analogique (0 à 10 V)	8; dont 4 utilisables en analogique (0 à 10 V)
Sorties TOR		
Nombre de sorties TOR	4; Transistor	4; Relais
Protection contre les courts-circuits	Oui; électrique (1 A)	Non; protection externe requise
Courant de sortie		
• pour état log. "1" plage admissible pour 0 à 55 °C, maxi	0,3 A	10 A
Sorties relais		
Pouvoir de coupure des contacts		
- pour charge inductive, maxi		3 A
- pour charge résistive, max.		10 A

Référence et modèle

Spécificités E/S intégrées à la CPU

Exemples: 2 modèles d'extension E/S TOR

Caractéristiques techniques (suite)

Numéro d'article	6ED1055-1CB10-0BA2 LOGO! DM16 24 MOD. EXT., 4UL, 8E/8S TOR	6ED1055-1NB10-0BA2 LOGO! DM16 24R MOD. EXT. 4UL, 8E/8S TOR
Type de configuration/Fixation		
Montage	sur rail DIN sym. 35 mm, largeur de 4 unités de châssis	sur rail DIN sym. 35 mm, largeur de 4 unités de châssis
Tension d'alimentation		
Valeur nominale (CC)		
• 24 V CC	Oui	Oui
• 115 V CC		
• 230 V CC		
Plage admissible, limite inférieure (CC)	20,4 V	20,4 V
Plage admissible, limite supérieure (CC)	28,8 V	28,8 V
Valeur nominale (CA)		
• 24 V CA		Non
• 115 V CA		
• 230 V CA		
Fréquence réseau		
• Plage admissible, limite inférieure		
• Plage admissible, limite supérieure		
Entrées TOR		
Nombre d'entrées TOR	8	8
Tension d'entrée		
• Type de tension d'entrée	DC	DC
• pour état log. "0"	< 5 V CC	< 5 V CC
• pour état log. "1"	> 12 V CC	> 12 V CC
Courant d'entrée		
• pour état log. "0", max. (courant de repos admissible)	0,85 mA	0,85 mA
• pour état log. "1", typ.	3,5 mA	2 mA
Retard d'entrée (pour valeur nominale de la tension d'entrée)		
pour entrées standard		
- pour "0" vers "1", maxi	1,5 ms	1,5 ms
- pour "1" vers "0", maxi	1,5 ms	1,5 ms
Sorties TOR		
Nombre de sorties TOR	8	8; Relais
Protection contre les courts-circuits	Oui	Non
Activation d'une entrée TOR	Oui	Oui
Pouvoir de coupure des sorties		
• pour charge de lampes, maxi		1 000 W
Montage en parallèle de deux sorties		
• pour augmentation de puissance	Non	Non
Fréquence de commutation		
• pour charge résistive, max.	10 Hz	2 Hz
• pour charge inductive, maxi	0,5 Hz	0,5 Hz
• mécanique, maxi		10 Hz
Sorties relais		
Pouvoir de coupure des contacts		
- pour charge inductive, maxi		3 A
- pour charge résistive, max.		5 A

Spécificités E/S TOR sur module d'extension

Exemples: 1 modèle d'extension E ANA

Numéro d'article	6ED1055-1MA00-0BA2 LOGO! AM2 MOD. EXT., 12/24V, 2EA,
Type de configuration/Fixation	
Montage	sur rail DIN sym. 35 mm, largeur de 2 unités de châssis
Tension d'alimentation	
Valeur nominale (CC)	
• 12 V CC	Oui; 10,8 V CC à 28,8 V CC
• 24 V CC	Oui; 10,8 V CC à 28,8 V CC
Entrées analogiques	
Nombre d'entrées analogiques	2
Etendues d'entrée	
• Tension	Oui
• Courant	Oui
• Thermomètres à résistance	Non
Etendues d'entrée (valeurs nominales), tensions	
• 0 à +10 V	Oui
Etendues d'entrée (valeurs nominales), courants	
• 0 à 20 mA	Oui; 0 mA ou 4 mA à 20 mA
Etendues d'entrée (valeurs nominales), thermomètres à résistance	
• Pt 100	Non
CEM	
Emission de perturbations radioélectriques selon EN 55 011	
• Classe de valeur limite B, pour l'emploi dans les zones résidentielles	Oui
Degré et classe de protection	
Degré de protection selon EN 60529	
• IP20	Oui

Spécificités E ANA sur module d'extension

II.1: Structure de contrôle de type API (petites applications)

Pour les petites applications industrielles d'entrées de gamme, les constructeurs proposent des automates compacts (avec possibilités d'extension). Ils sont en général dédiés à la petite machine.

CompactLogix

COMPACTLOGIX
API compact avec E/S intégrés
Possibilité d'extension de cartes
Communication industrielle:
Ethernet IP-DeviceNet
Programmation par soft :
RSLogix 5000
Langages:
Ladder/logigramme/ST/SFC

M221

M241

M251

M258

Plusieurs modèles:
MODICOM:
M221/M241/M251/M258
API compact avec E/S intégrés
Possibilité d'extension de cartes
Communication industrielle:
Ethernet IP-Modbus-Canopen
Programmation par soft :
Somachine
Langages:
Ladder/logigramme/ST/SFC

S71200

S71200
API compact avec E/S intégrés
Possibilité d'extension de cartes
Communication industrielle: Ethernet IP-Profinet-Profibus
Programmation par soft :
Step7 TIA-PORTAL
Langages: Ladder/logigramme/ST

II.1: Structure de contrôle de type api (petites applications)

API SIEMENS S71200 : Exemple de caractéristiques

Appareil :

Descriptif d'une CPU S71200: modèle 1214

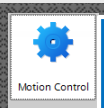
CPU 1214C DC/DC/DC

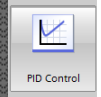
N° d'article : 6ES7 214-1AG40-0XB0

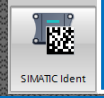
Version : V4.2

Description :

Mémoire de travail 100 Ko ; alimentation DC24V avec DI14 x DC24V SINK/SOURCE, DQ10 x DC24V et AI2 intégrées ; 6 compteurs rapides et 4 sorties d'impulsions intégrées ; extension des E/S intégrées par Signal Board ; jusqu'à 3 modules de communication pour communication série ; jusqu'à 8 modules d'entrées-sorties pour extension des E/S ; 0,04 ms/k instructions ; interface PROFINET pour programmation, communication IHM et API-API







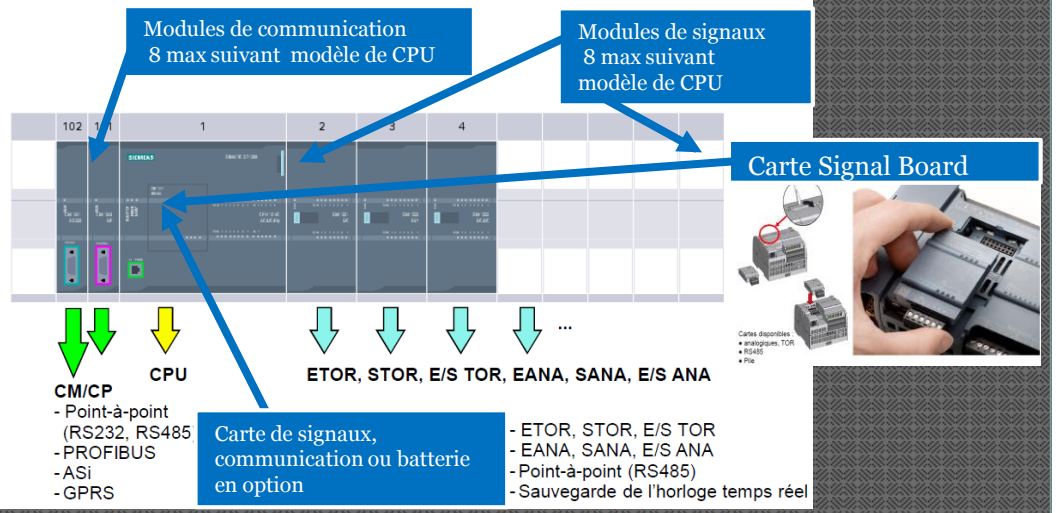
Fonctions technologiques intégrées:

Motion control: Axe en asservissement de position

PID control: régulation

Simatic ident : système RFID- Lecteur optique

Présentation de l'architecture S71200



Type de CPU	S71200
Standard	●
Sécurité	●
Compact	●
Technologie	●
Haute performance	●

Type de cartes	S71200
DI/DQ	●
AI/AQ	●
F-DI/FDQ	●
F-AI	●

II.1: Structure de contrôle de type API (petites applications)

API SIEMENS s71200 : Exemples de caractéristiques

Référence et modèle

Spécificités Zones de mémoires

Langages de programmation

Spécificités des entrées/sorties

Spécificités réseau industriel

Spécificités fonctions technologiques

Caractéristiques techniques

Numéro d'article	6ES7214-1BG40-0XB0 CPU 1214C, CA/CC/REL, 14ETOR/10STOR/2EA	6ES7214-1AG40-0XB0 CPU 1214C, CC/CC/CC, 14ETOR/10STOR/2EA
Informations générales		
Désignation du type de produit	CPU 1214C AC/DC/Relay	CPU 1214C DC/DC/DC
Ingénierie avec		
• Pack de programmation	à partir de STEP 7 V14	à partir de STEP 7 V14
Tension d'alimentation		
Valeur nominale (CC)		
• 24 V CC		Oui
Valeur nominale (CA)		
• 120 V CA	Oui	
• 230 V CA	Oui	
Alimentation des capteurs		
Alimentation des capteurs 24 V		
• 24 V	20,4 à 28,8 V	L+ moins 4 V CC min.
Puissance dissipée		
Puissance dissipée, typ.	14 W	12 W
Mémoire		
Mémoire de travail		
• Intégré	100 kbyte	100 kbyte
Mémoire de chargement		
• Intégré	4 Mbyte	4 Mbyte
• enfichable (SIMATIC Memory Card), maxi.	Carte mémoire SIMATIC	Carte mémoire SIMATIC
Sauvegarde		
• sans pile	Oui	Oui
Temps de traitement CPU		
pour opérations sur bits, typ.	0,085 µs; / instruction	0,085 µs; / instruction
pour opérations sur mots, typ.	1,7 µs; / instruction	1,7 µs; / instruction
pour opérations à virgule flottante, typ.	2,3 µs; / instruction	2,3 µs; / instruction
Zones de données et leur rémanence		
Mémentos		
• Nombre, maxi	8 kbyte; Taille de la zone de mémentos	8 kbyte; Taille de la zone de mémento
Mémoire image du processus		
• Entrées, réglables	1 kbyte	1 kbyte
• Sorties, réglables	1 kbyte	1 kbyte
Heure		
Horloge		
• Horloge matérielle (horloge temps réel)	Oui	Oui

Caractéristiques techniques (suite)

Numéro d'article	6ES7214-1BG40-0XB0 CPU 1214C, CA/CC/REL, [illegible]	6ES7214-1AG40-0XB0 CPU 1214C, CC/CC/CC, 14ETOR/10STOR/2EA
Configuration		
Programmation		
Langage de programmation		
- CONT	Oui	Oui
- LOG	Oui	Oui
- SCL	Oui	Oui
Dimensions		
Largeur	110 mm	110 mm
Hauteur	100 mm	100 mm
Profondeur	75 mm	75 mm
Poids		
Poids approx.	455 g	415 g

Caractéristiques techniques (suite)

Numéro d'article	6ES7214-1BG40-0XB0 CPU 1214C, CA/CC/REL, 14ETOR/10STOR/2EA	6ES7214-1AG40-0XB0 CPU 1214C, CC/CC/CC, 14ETOR/10STOR/2EA
Entrées TOR		
Nombre d'entrées TOR	14; intégré	14; intégré
• dont entrées utilisables pour les fonctions technologiques	6; HSC (compteur rapide)	6; HSC (compteur rapide)
Sorties TOR		
Nombre de sorties TOR	10; Relais	10
• dont les sorties rapides		4; Sortie de trains d'impulsions 100 KHz
Entrées analogiques		
Nombre d'entrées analogiques	2	2
Etendues d'entrée		
• Tension	Oui	Oui
Sorties analogiques		
Nombre de sorties analogiques	0	0
1. Interface		
Type d'interface	PROFINET	PROFINET
Physique	Ethernet	Ethernet
Fonctionnalité		
• Contrôleur PROFINET IO	Oui	Oui
• Périphérique PROFINET IO	Oui	Oui
• Communication SIMATIC	Oui	Oui
• Communication IE ouverte	Oui	Oui
• Serveur Web	Oui	Oui
• Redondance des média	Non	Non
Fonctions de communication		
Communication S7		
• pris en charge	Oui	Oui
Communication IE ouverte		
• TCP/IP	Oui	Oui
• ISO-on-TCP (RFC1006)	Oui	Oui
• UDP	Oui	Oui
Serveur Web		
• pris en charge	Oui	Oui
Nombre de liaisons		
• total	16; dynamique	16; dynamique
Fonctions intégrées		
Nombre de compteurs	6	6
Fréquence de comptage (compteurs), maxi	100 kHz	100 kHz
Fréquencemètre	Oui	Oui
Positionnement en boucle ouverte	Oui	Oui
Nombre d'axes de positionnement asservis, max.	8	8
Nombre de axe de positionnement via interface impulsion-direction	jusqu'à 4 avec SB 1222	4; avec sorties intégrées
Régulateur PID	Oui	Oui
Nombre d'entrées d'alarme	4	4
Nombre de sorties impulsionnelles		4
Fréquence limite (impulsion)		100 kHz

II.2: Structure de contrôle de type api (applications standards)

Pour les applications industrielles de haut de gamme et les applications process.
les constructeurs proposent des automates compacts (avec possibilités d'extension). Ils sont en général dédiés à la grosse machine ou l'automatisation de process industriels de grande envergure.

ControlLogix 5580

COMTROLLOGIX
API standard avec modules E/S
Communication industrielle:
EthernetIP-ControlNet-DeviceNet
Programmation soft:
Studio 5000 Logix Designer
Langages:
Ladder/logigramme/ST/SFC

M340

M580

Prenium

Quantum (contrôle de process)

Plusieurs modèles:
MODICOM:
M340/M580/Prenium/Quantum
API compact avec E/S intégrés
Possibilité d'extension de cartes
Communication industrielle:
Ethernet IP-Modbus-Canopen
Programmation par soft :
UNITY PRO
Langages:
Ladder/logigramme/ST/SFC

S7300

S71500

S7400 (contrôle de process)

S71500/s7300/s7400
API standard avec modules E/S
Communication industrielle:
EthernetIP-Profinet-Profibus
Programmation par soft :
Step7 TIA-PORTAL
Langages: Ladder/logigramme/ST/
LIST/SFC

II.2: Structure de contrôle de type api (applications standards)

API SIEMENS S71500 : Exemple de caractéristiques

Appareil :

CPU 1518-4 PN/DP

Descriptif d'une CPU S71500: modèle 1518

N° d'article : 6ES7 518-4AP00-0AB0

Version : V2.1

Description :

CPU avec écran ; mémoire de travail 4 Mo code et 20 Mo données ; temps d'opération sur bits 1 ns ; concept de sécurité à 4 niveaux, fonctions technologiques intégrées : Motion Control, Régulation, Comptage&Mesure ; traçabilité intégrée ; première interface : contrôleur PROFINET IO, prise en charge de RT/IRT, Performance Upgrade PROFINET V2.3, 2 ports, périphérique I, MRP, MRPD, protocole de transport TCP/IP, secure Open User Communication, communication S7, serveur Web, client DNS, OPC UA Server Data Access, équidistance, routage ; deuxième interface : contrôleur PROFINET IO, prise en charge RT, périphérique I, protocole de transport TCP/IP, secure Open User Communication, communication S7, serveur Web, client DNS, OPC UA Server Data Access, routage ; troisième interface : interface Gigabit, services de base PROFINET, protocole de transport TCP/IP, secure Open User Communication, communication S7, serveur Web, client DNS, OPC UA Server Data Access, routage ; quatrième interface : maître PROFIBUS DP, communication S7. équidistance, routage ; options Runtime, firmware V2.1

Fonctions technologiques intégrées:

Motion control: Axe en asservissement de position

PID control: régulation

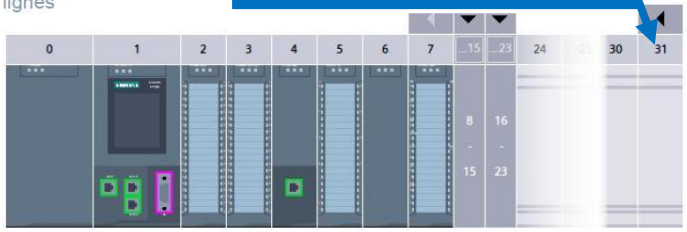
Simatic ident : système RFID- Lecteur optique

Comptage rapide: HSC – Mesures sur codeurs externes

Présentation de l'architecture S71500

Montage sur une seule ligne
ET200MP décentralisé sur plusieurs lignes

Modules centralisés
32 max



PS/PM (option)

CPU

ETOR, STOR, EAN, SANA

CM/CP :
- Point-à-point (RS232, RS485)
- PROFIBUS
- PROFINET

PS nouveau segment d'alimentati

TM :
- Comptage
- Détection de position

2 segments d'alimentation par châssis. Alimentation pour bus de fond de panier des modules périphériques suivants

Type de CPU	S71500
Standard	●
Sécurité	●
Compact	●
Technologie	●
Haute performance	●

Type de cartes	S71500
DI/DQ	●
AI/AQ	●
F-DI/FDQ	●
F-AI	●

II.2: Structure de contrôle de type api (applications standards)

API SIEMENS s71500 : Modèles de CPU

C : Compact

F: Failsafe (sécurité)

T : Technologique

MFP : Multifunctional Platform

	Compact CPUs		Standard-CPUs						Technology CPUs				MFP
CPU types	1511C-1 PN	1512C-1 PN	1511F-1 PN	1513F-1 PN	1515F-2 PN	1516F-3 PN/DP	1517F-3 PN/DP	1518F-4 PN/DP	1511TF-1 PN	1515TF-2 PN	1516TF-3 PN/DP	1517TF-3 PN/DP	1518F-4 PN/DP MFP
Interfaces	1 1	1 1	1 1	1 1	1 2 1	1 2 1	1 2 1	1 2 1 3	1 1	1 2 1	1 2 1	1 2 1	1 2 1 3
Program/ data storage	175 KB 1 MB	250 KB 1 MB	150/ 225 KB 1 MB	300/ 450 KB 1.5 MB	500/ 750 KB 3 MB	1/ 1.5 MB 5 MB	2/3 MB 8 MB	4/6 MB 20 MB	225/ 225 KB 1 MB	750/ 750 KB 3 MB	1.5/ 1.5 MB 5 MB	3/3 MB 8 MB	4/6 MB 20 MB 50 MB[1]
Bit-performance	60 ns	48 ns	60 ns	40 ns	30 ns	10 ns	2 ns	1 ns	60 ns	30 ns	10 ns	2 ns	1 ns
Max. number of connections	96	128	96	128	192	256	320	384	96	192	256	320	384
Positioning axes • Typical[2] • Maximum[2]	 5 10	 5 10	 5 10	 5 10	 7 30	 7 30	 70 128	 128 128	 5 10	 7 30	 65 80	 70 128	 128 128
Width	85 mm	110 mm	35 mm	35 mm	70 mm	70 mm	175 mm	175 mm	35 mm	70 mm	175 mm	175 mm	175 mm
											New		New

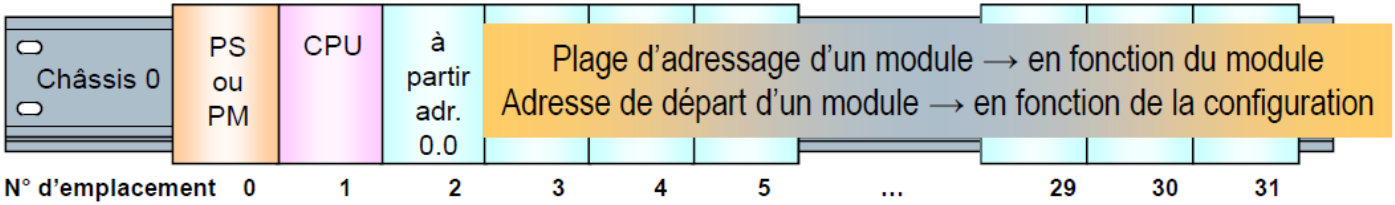
1 Additional 50 MB memory for ODK applications; 2 For 4ms Servo/IPO cycle

ODK : Open Development Kit

Mémoire travail: Programme/Données

II.2: Structure de contrôle de type api (applications standards)

API SIEMENS s71500 : Plan d'adressage des modules

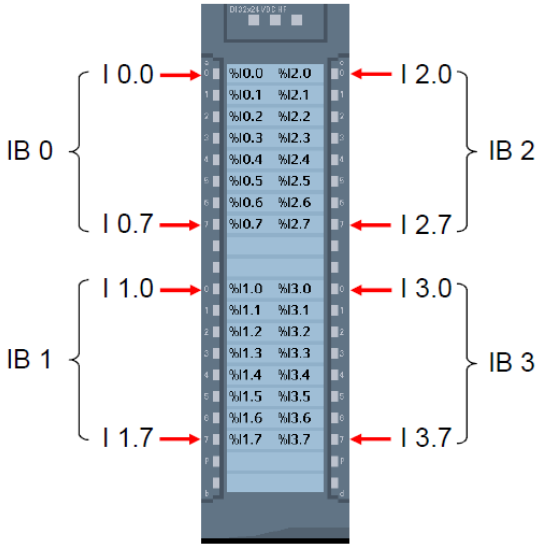


Adresses par défaut des modules périphériques :

- Adressage en fonction de l'emplacement
- Adressage à partir de l'adresse = 0
- Les adresses sont attribuées en continu dans l'ordre dans lequel les modules périphériques ont été configurés

Ex : paramétrage du module d'E/S de 0,,3

Ex. **DI32**



II.2: Structure de contrôle de type api (applications standards)

API SIEMENS S71500 : Exemples de caractéristiques

Référence et modèle

Spécificités Zones de mémoires

Mémoire image entrées/sorties

Spécificités réseau industriel

Caractéristiques techniques		
Numéro d'article	**6ES7511-1AK01-0AB0** CPU 1511-1 PN, 150KO PROG., 1MO DONN.	**6ES7513-1AL01-0AB0** CPU 1513-1 PN, 300KO PROG., 1.5MO DONN.
Informations générales		
Désignation du type de produit	CPU 1511-1 PN	CPU 1513-1 PN
Ingénierie avec		
• STEP 7 TIA Portal configurable/ intégré à partir de la version	V14 SP1 (FW V2.1) / à partir de V13 SP1, mise à jour 4 (FW V1.8)	V14 SP1 (FW V2.1) / à partir de V13 SP1, mise à jour 4 (FW V1.8)
Ecran		
Diagonale d'écran [cm]	3,45 cm	3,45 cm
Tension d'alimentation		
Type de tension d'alimentation	24 V CC	24 V CC
Puissance dissipée		
Puissance dissipée, typ.	5,7 W	5,7 W
Mémoire		
Mémoire de travail		
• intégré (pour programme)	150 kbyte	300 kbyte
• intégré (pour données)	1 Mbyte	1,5 Mbyte
Mémoire de chargement		
• enfichable (SIMATIC Memory Card), max.	32 Gbyte	32 Gbyte
Temps de traitement CPU		
pour opérations sur bits, typ.	60 ns	40 ns
pour opérations sur mots, typ.	72 ns	48 ns
pour opérations à virgule fixe, typ.	96 ns	64 ns
pour opérations à virgule flottante, typ.	384 ns	256 ns
Compteurs, temporisations et leur rémanence		
Compteurs S7		
• Nombre	2 048	2 048
Compteurs CEI		
• Nombre	illimité (limitation uniquement par mémoire de travail)	illimité (limitation uniquement par mémoire de travail)
Temporisations S7		
• Nombre	2 048	2 048
Temporisateurs CEI		
• Nombre	illimité (limitation uniquement par mémoire de travail)	illimité (limitation uniquement par mémoire de travail)

Zones de données et leur rémanence		
Mémentos		
• Nombre, maxi	16 kbyte	16 kbyte
Plage d'adresses		
Plage d'adresses de périphérie		
• Entrées	32 kbyte; toutes les entrées se trouvent dans la mémoire image du processus	32 kbyte; toutes les entrées se trouvent dans la mémoire image du processus
• Sorties	32 kbyte; toutes les sorties se trouvent dans la mémoire image du processus	32 kbyte; toutes les sorties se trouvent dans la mémoire image du processus
Heure		
Horloge		
• Type	Horloge matérielle	Horloge matérielle
1. Interface		
Réalisation physique de l'interface		
• Nombre de ports	2	2
• Commutateur intégré	Oui	Oui
• RJ 45(Ethernet)	Oui; X1	Oui; X1
Fonctionnalité		
• Protocole IP	Oui; IPv4	Oui; IPv4
• Contrôleur PROFINET IO	Oui	Oui
• Périphérique PROFINET IO	Oui	Oui
• Communication SIMATIC	Oui	Oui
• Communication IE ouverte	Oui	Oui
• Serveur Web	Oui	Oui
• Redondance des média	Oui	Oui
Contrôleur PROFINET IO		
Services		
- Communication PG/OP	Oui	Oui
- Routage S7	Oui	Oui
- Mode synchrone	Oui	Oui
- Communication IE ouverte	Oui	Oui
- IRT	Oui	Oui
- MRP	Oui; en tant que gestionnaire de la redondance MRP et/ou client MRP ; nombre max. de périphériques dans l'anneau : 50	Oui; en tant que gestionnaire de la redondance MRP et/ou client MRP ; nombre max. de périphériques dans l'anneau : 50
- MRPD	Oui; Condition : IRT	Oui; Condition : IRT
- PROFIenergy	Oui	
- Démarrage prioritaire	Oui; max. 32 appareils PROFINET	Oui; max. 32 appareils PROFINET
- Nombre de périphériques IO raccordables, max.	128; au total, il est possible de raccorder max. 256 périphériques décentralisés via AS-i, PROFIBUS ou PROFINET	128; au total, il est possible de raccorder max. 512 périphériques décentralisés via AS-i, PROFIBUS ou PROFINET
- dont périphériques d'E/S avec IRT, max.	64	64
- Nombre de périphériques d'E/S raccordables pour RT, maxi	128	128
- dont en ligne, maxi	128	128
- Nombre de périphériques IO activables/désactivables simultanément, maxi	8; au total sur toutes les interfaces	8; au total sur toutes les interfaces
- Nombre de périphériques d'E/S par outil, maxi	8	8
- Temps de rafraîchissement	La valeur minimale du temps d'actualisation dépend aussi du temps paramétré pour la communication PROFINET IO, du nombre de périphériques IO et du nombre de données utiles configurées	La valeur minimale du temps d'actualisation dépend aussi du temps paramétré pour la communication PROFINET IO, du nombre de périphériques IO et du nombre de données utiles configurées

Principales caractéristiques, pour plus de détails (voir catalogue)
Configurateur de solution:
TIA Selection Tool

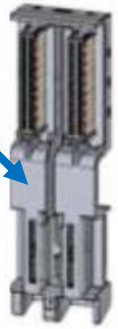
II.2: Structure de contrôle de type api (applications standards)

API SIEMENS s71500 : Montage et positions de montage

Profil support
+
rail DIN symétrique intégré

Connecteur de BUS
liaison CPU aux modules périphériques

en l'absence de connexion sur fond de panier

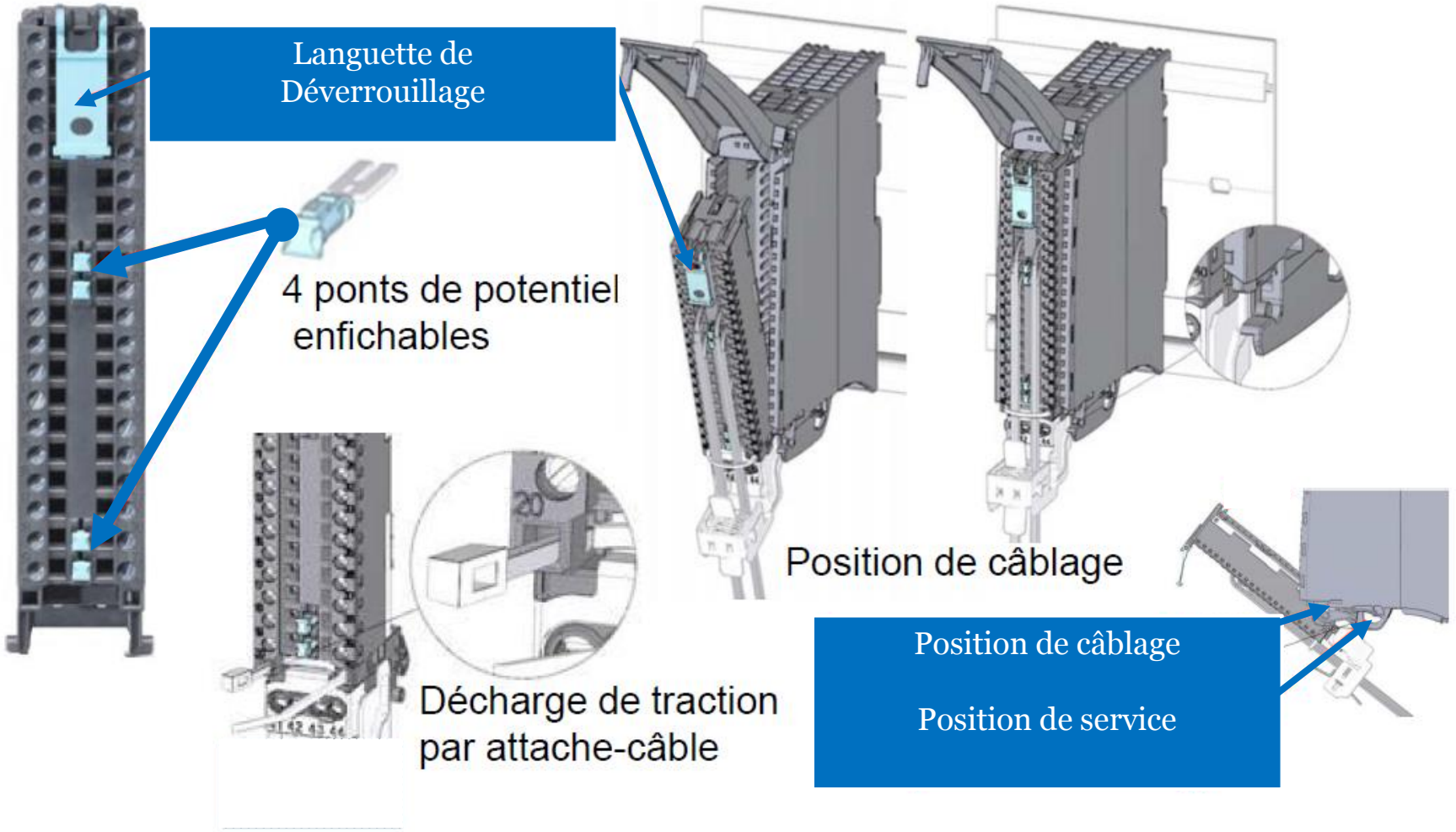


Montage vertical

Montage horizontal

II.2: Structure de contrôle de type api (applications standards)

API SIEMENS s71500 : Technique de raccordement- Connecteur frontal



II.2: Structure de contrôle de type api (applications standards)

API SIEMENS s71500 : Présentation de l'afficheur

- Toutes les CPU S7-1500 sont livrées avec un afficheur
- Deux tailles en fonction du type de CPU
  1,36“ jusqu'à CPU1513
  2,4“ à partir de CPU1516
- Possèdent leur propre MLFB
  → commande possible comme pièce de rechange
- La CPU est exploitable sans afficheur (autre volet frontal)
- Débrochage et enfichage possibles en cours de fonctionnement → la CPU reste en RUN
- Affichage en 3 langues (menus et messages d'erreur/signalisation)
- Changement de langue possible en cours de fonctionnement
- Acquittement des alarmes
- Backup/Restore
- Formatage de la carte mémoire

II.2: Structure de contrôle de type api (applications standards)

API SIEMENS s71500 : Menus de l'afficheur

Options du menu principal et leur signification :

- **Aperçu** (état de la CPU : nom, type, MLFB, version)
- **Diagnostic** (affichage tampon de diagnostic, alarmes)
- **Réglages** (paramétrage de la CPU : adresses, date/heure, mode de fonctionnement, RAZ CPU, niveaux de protection, déverrouillage afficheur si mot de passe)
- **Modules** (information d'état des modules dans le système)
- **Ecran** (paramétrage de l'afficheur : luminosité, langue, mode économie d'énergie, MLFB, version)

Couleur des informations d'état et leur signification :

vert	Marche de la CPU, Marche avec alarme
jaune	Stop ou arrêt de la CPU
rouge	Erreur
blanc	Etablissement de la connexion ou connexion interrompue avec la CPU

II.2: Structure de contrôle de type api (applications standards)

API SIEMENS s71500 : Menus de l'afficheur

II.3: Périphérie décentralisée

Pour les applications industrielles de haut de gamme et les applications process, les automaticiens intègrent souvent **des entrées/sorties déportées que l'on nomme « périphérie décentralisée ».**
Les échanges avec les CPU se font par réseau industriel.


CompactBlock LDX I/O


Modules POINT I/O


Modules d'E/S FLEX 5000


Modules d'E/S FLEX 1794

Plusieurs modèles :
- **CompactBlock LDX I/O**
- **Modules POINT I/O**
- **Modules d'E/S FLEX 5000**
- **Modules d'E/S FLEX 1794**
- **…**

Tout type d'E/S-Différents indices de protection-Différentes interfaces réseaux-E/S de sécurité-modèles ATEX


OTB


TM5


TM7


X80


Advantys STB


Advantys ETB

Plusieurs modèles :
API logic contrôler :
- **OTB**
- **TM5-IP20**
- **TM7-IP67**

API M340-M580-Quantum:
- **X80**
- **Advantys STB-IP20**
- **Advantys ETB-IP67**

Tout type d'E/S-Différents indices de protection-Différentes interfaces réseaux-E/S de sécurité-modèles ATEX


ET200SP


ET200MP


ET200M


ET200ISP


ET200pro

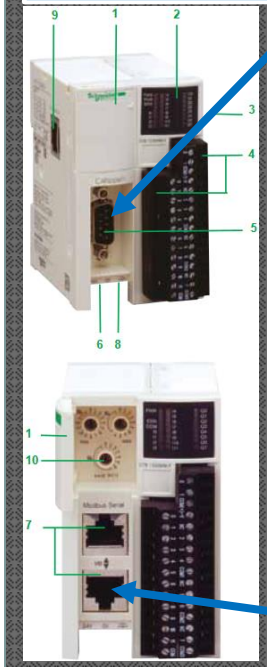
Plusieurs modèles :
En armoire (IP20)
- **ET200SP**
- **ET200MP (format S71500)**
- **ET200M (format S7300)**
- **ET200ISP (ATEX - IP30)**

Hors armoire:
- **ET200pro (IP65/67)**
- **Etc.**

Tout type d'E/S-Différents indices de protection-Différentes interfaces réseaux-E/S de sécurité-modèles ATEX

II.3: Périphérie décentralisée

Périphérie décentralisée/SCHNEIDER OTB IP20 : Exemple de caractéristiques



OTB interface Canopen

OTB interfaces Modbus-série

Description

Les modules interface Modicon **OTB1●0 DM9LP** *(1)* comprennent :

1 Une porte d'accès pivotante.
2 Un bloc de visualisation de :
- □ l'état du module d'interface et de sa communication (PWR, RUN, ERR, COM, STAT, 10T, 100T selon modèle)
- □ l'état des entrées et des sorties (IN● et OUT●).

3 Un connecteur pour les modules d'extension (face latérale droite).
4 Deux borniers à vis débrochables pour le raccordement des capteurs d'entrées et des préactionneurs de sorties.

Selon modèle :
5 Un connecteur type SUB-D 15 contacts pour le raccordement du bus CANopen avec interface **OTB1C0DM9LP**.
6 Un connecteur type RJ45 pour le raccordement du réseau Ethernet Modbus/TCP, avec interface Ethernet Modbus/TCP **OTB1E0DM9LP**.
7 Deux connecteurs type RJ45 en parallèle pour le raccordement de la liaison série Modbus avec interface **OTB1S0DM9LP**.

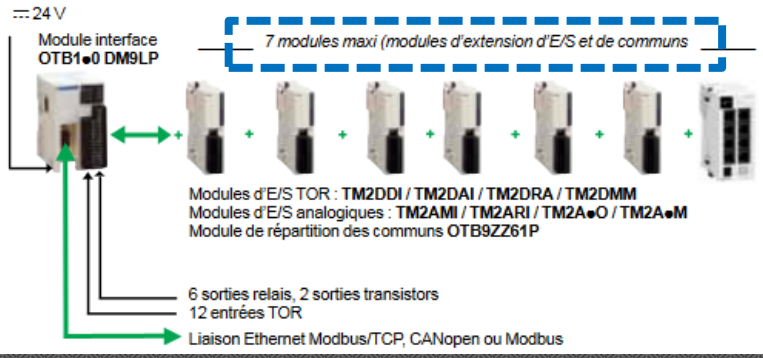
8 Un bornier à vis pour le raccordement de l'alimentation secteur ⎓ 24 V.
9 Un connecteur type RJ45 destiné à la mise à jour par téléchargement du système d'exploitation interne du module.

Avec accès par la porte pivotante 1
10 Deux ou trois roues codeuses (selon modèle) pour le réglage de l'adresse de l'îlot OTB et de son débit binaire sur le réseau, le bus ou la liaison série.

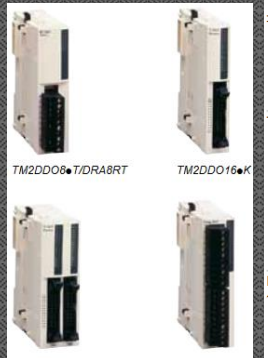
Bus de terrain CAN	Réseau local RS 485
ISO 11898 (connecteur SUB-D 9 contacts)	RS 485 (2 connecteurs RJ 45 en parallèle)
CSMA-MA, multimaître	Maître-esclave
10…1000 Kbit/s selon distance	1,2…38,4 kbauds
Double paire torsadée blindée	Double paire torsadée
127 esclaves	32 esclaves par segment
30 m *(9843 ft)* (1 Mbit/s)…1000 m *(3280 ft)* (> 10 Kbit/s)	Jusqu'à 1000 m *(3280 ft)*
20 E/S	
12 entrées ⎓ 24 V sink/source (PNP ou NPN)	
6 sorties à relais et 2 sorties ⎓ 24 V transistors source (PNP)	
Borniers à vis débrochables	
7 modules d'entrées/sorties TOR, analogiques ou accessoires de connexion	
Incluant les E/S TOR du module d'interface : - 132 avec extension d'E/S TOR à bornier à vis - 164 avec extension d'E/S TOR à bornier à ressort - 228 avec extension d'E/S TOR à connecteur type HE10 - E/S analogiques à bornier à vis : jusqu'à 7 x 8 E, ou 7 x 2 S, ou 7 x (4 E/2 S)	
2 voies 32 bits (0…4 294 967 295 points) - entrées TOR dédiées - comptage/décomptage avec présélection	
2 voies 32 bits (0…4 294 967 295 points) - entrées/sorties TOR dédiées - comptage/décomptage, comptage, décomptage, fréquencemètre	
2 voies fonction PWM (sortie à modulation de largeur d'impulsion) et fonction PLS (sortie générateur d'impulsions)	

Caractéristiques Interface CANopen et MODBUS série

Configuration des modules interface



Exemples de modules de sorties TOR



Modules de sorties "Tout ou Rien"

Type de sortie	Nb de voies	Nb de points communs	Raccordement	Référence	Masse kg / lb
Transistors ⎓ 24 V	8, sink 0,3 A	1	Par bornier débrochable à vis (fourni)	**TM2DDO8UT**	0,085 / 0,187
	8, source 0,5 A	1	Par bornier débrochable à vis (fourni)	**TM2DDO8TT**	0,085 / 0,187
Transistors ⎓ 24 V	16, sink 0,1 A	1	Par connecteur type HE 10	**TM2DDO16UK**	0,070 / 0,154
	16, source 0,4 A	1	Par connecteur type HE 10	**TM2DDO16TK** *(1)*	0,070 / 0,154
	32, sink 0,1 A	2	Par connecteur type HE 10	**TM2DDO32UK**	0,105 / 0,231
	32, source 0,4 A	2	Par connecteur type HE 10	**TM2DDO32TK** *(1)*	0,105 / 0,231
Relais 2 A (Ith) **~ 230 V/⎓ 30 V**	8 (contact NO)	2	Par bornier débrochable à vis (fourni)	**TM2DRA8RT**	0,110 / 0,243
	16 (contact NO)	2	Par bornier débrochable à vis (fourni)	**TM2DRA16RT**	0,145 / 0,320

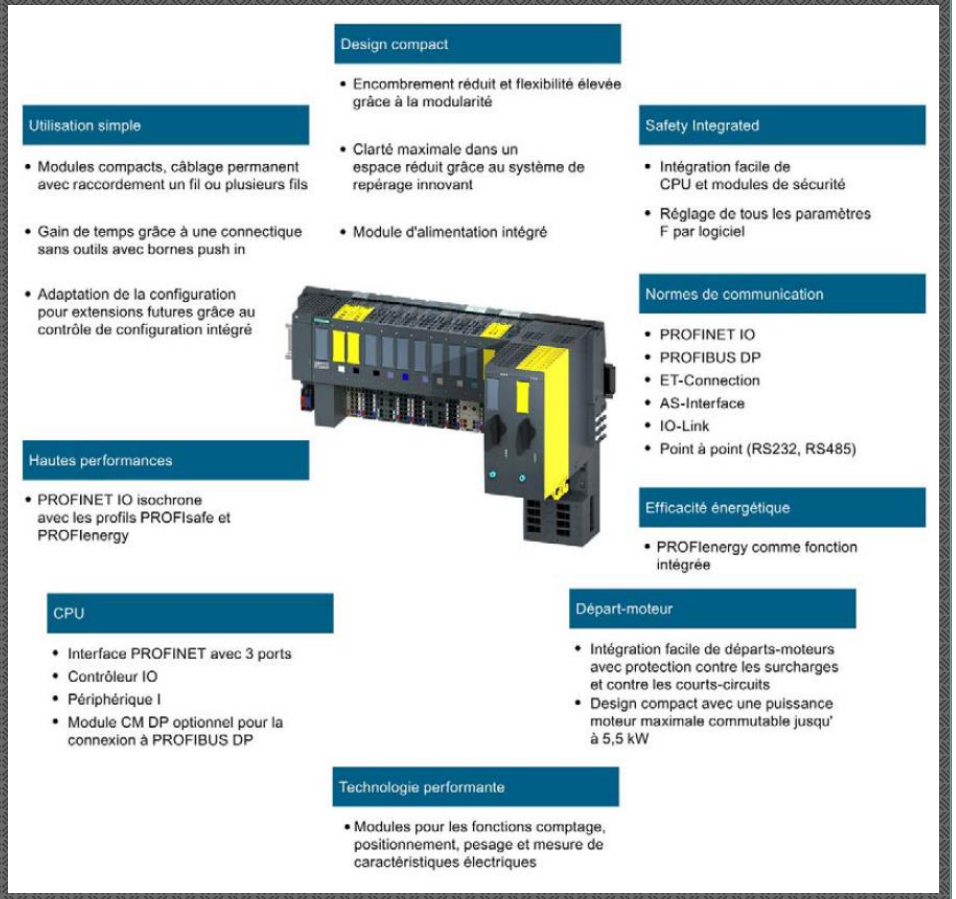
II.3: Périphérie décentralisée

Périphérie décentralisée SIEMENS ET200SP: Exemple de caractéristiques

SIMATIC ET 200SP est un système de périphérie décentralisée modulable permettant de coupler les signaux du processus à un automate de niveau supérieur via un bus de terrain.

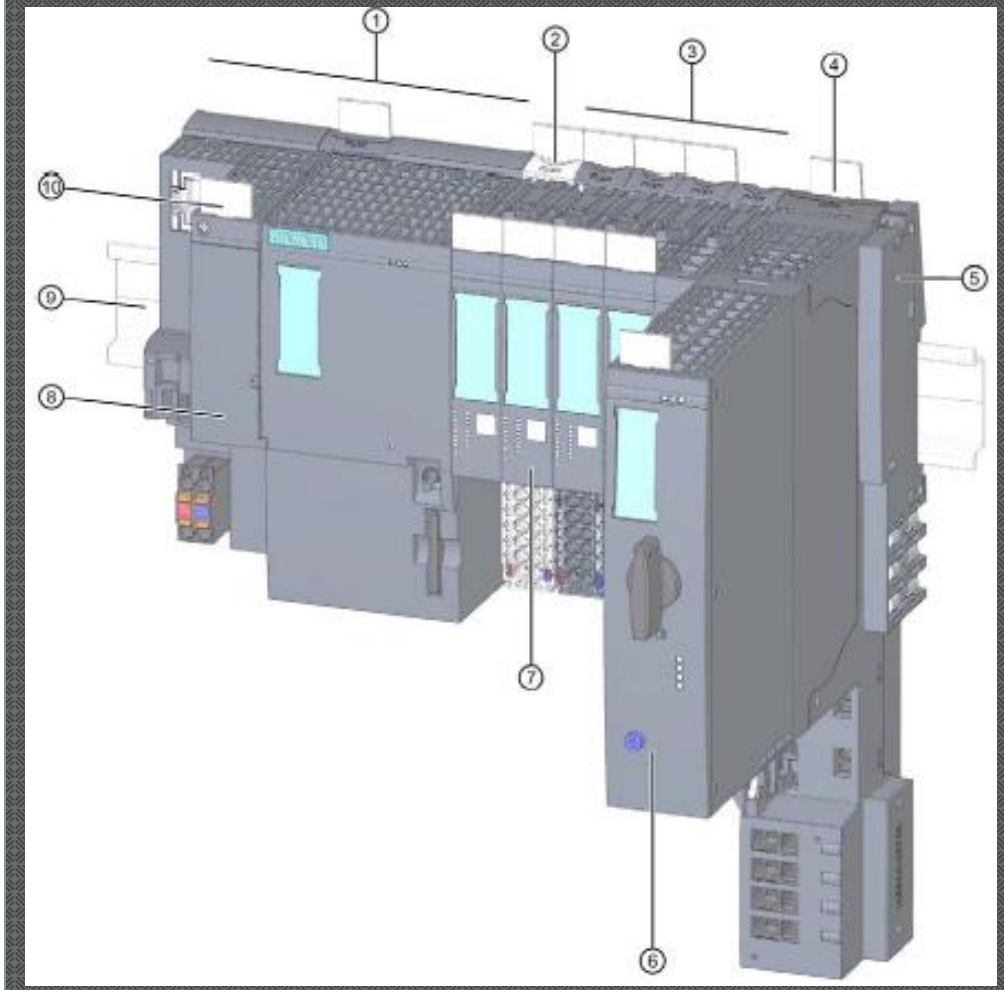
Domaine d'utilisation :
Multifonctionnel, le système de périphérie décentralisée SIMATIC ET 200SP convient à l'utilisation dans différents domaines d'application :

- **Structure modulable**
- **Modèles avec CPU**
- **Modules d'interface PROFINET IO ou PROFIBUS DP**
- **Applications pour la technique de sécurité**
- **Large éventail de modules de périphérie (64 modules max) (31 départs-moteurs)**
- **Indice de protection IP 20**
- **Montage encastré dans armoire électrique (rail DIN)**



II.3: Périphérie décentralisée

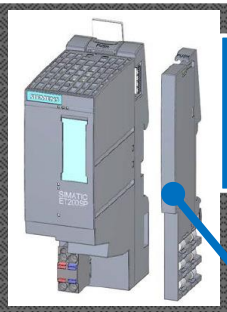
Périphérie décentralisée SIEMENS ET200SP: Exemple de configuration



① CPU/module d'interface
② BaseUnit BU..D claire avec arrivée de l'alimentation
③ BaseUnits BU..B foncées pour le prolongement du groupe de potentiel
④ BaseUnit pour départs-moteurs
⑤ Module serveur (compris dans la fourniture de la CPU/du module d'interface)
⑥ Départ-moteur ET 200SP
⑦ Module de périphérie
⑧ BusAdapter
⑨ Profilé support
⑩ Etiquette de repérage

II.3: Périphérie décentralisée

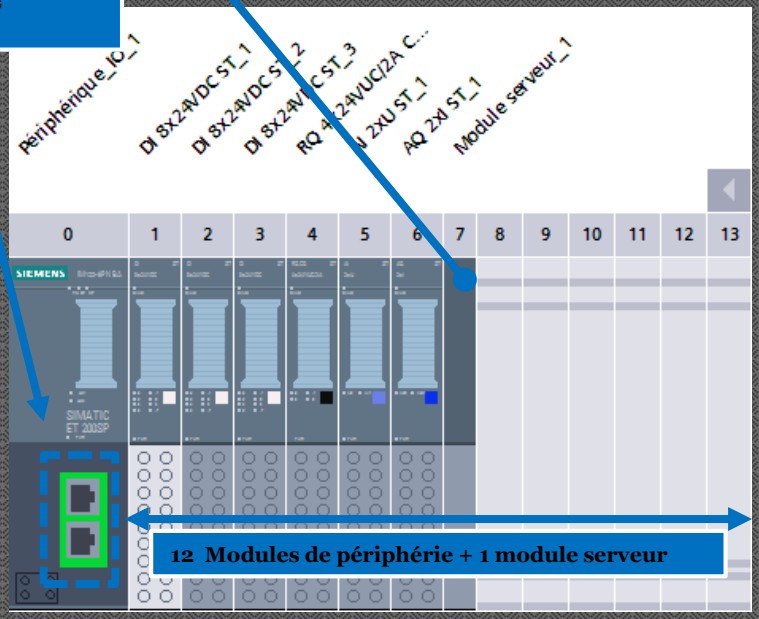
Périphérie décentralisée SIEMENS ET200SP : Exemple de module d'interface PROFINET



ET 200SP- Module d'interface IM 155-6 PN BA (6ES7155-6AR00-0AN0)
– Connecte le système de périphérie décentralisée ET 200SP avec PROFINET IO
– Tension d'alimentation DC 24 V

Module d'interface avec 2 ports PROFINET

Le module serveur est livré avec le module d'interface et doit toujours être placé en fin de montage



Caractéristiques techniques

Numéro d'article	**6ES7155-6AR00-0AN0**
	ET 200SP, IM155-6PN BASIC
Informations générales	
Désignation du type de produit	IM 155-6 PN BA avec 2x ports RJ45 et module serveur
Fonction du produit	
• Données I&M	Oui; I&M0 à I&M3
Ingénierie avec	
• STEP 7 TIA Portal configurable/ intégré à partir de la version	V13 SP1
• STEP 7 configurable/intégré à partir de la version	à partir de V5.5 SP4
• PROFIBUS à partir de la version/ révision GSD	
• PROFINET à partir de la version/ révision GSD	V2.3 / -
Tension d'alimentation	
Type de tension d'alimentation	
Valeur nominale (CC)	24 V
Protection contre l'inversion de polarité	Oui
Temps de maintien sur panne réseau/d'alimentation	
• Temps de maintien sur panne réseau/d'alimentation	5 ms
Configuration matérielle	
Profilé-support	
• Modules par châssis, maxi	12
Cartouches	
• Nombre de sous-modules par station, max.	
Interfaces	
Nombre d'interfaces PROFINET	1; 2 ports (commutateur)
Nombre d'interfaces PROFIBUS	

Référence et modèle

Versions logicielles compatibles

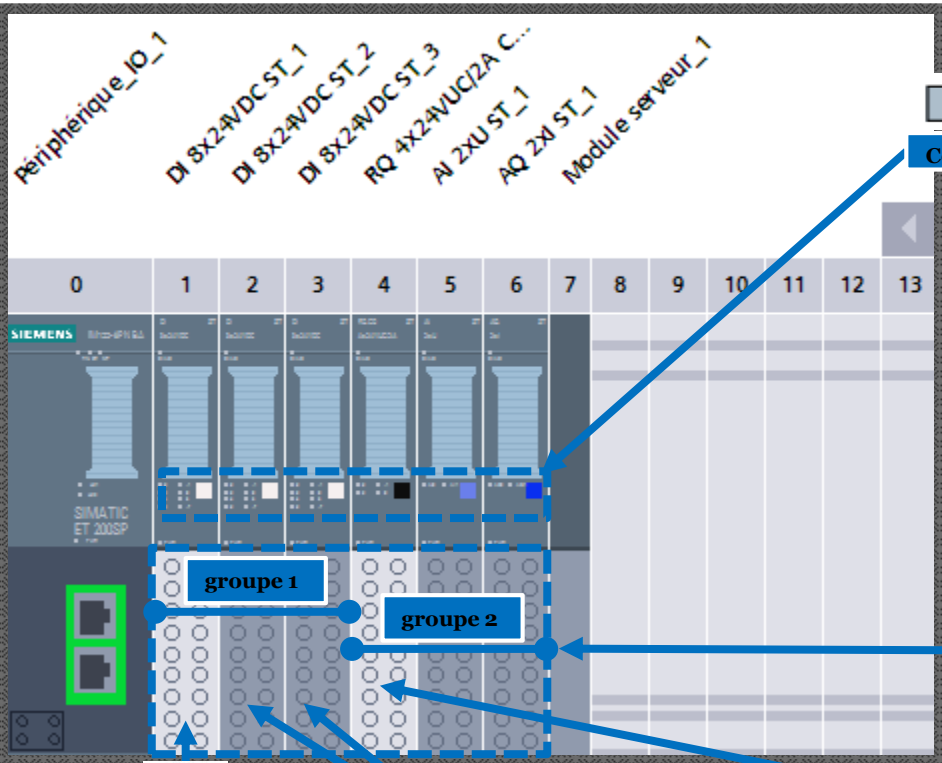
Alimentation 24V DC

Nombre de modules périphériques

Nombre de ports PROFINET

II.3: Périphérie décentralisée

Périphérie décentralisée SIEMENS ET200SP : Modules d'entrées/Sorties



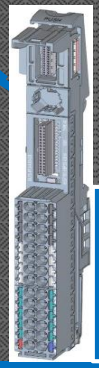
DI DQ AI AQ

Code couleur associé aux types de modules

Les BaseUnit assurent la liaison électrique et mécanique des modules de l'ET 200SP.

Pour chaque type de module, il existe un type de BaseUnit

BU de type A0, variante claire
Elle crée un nouveau groupe de potentiel limité à 10A

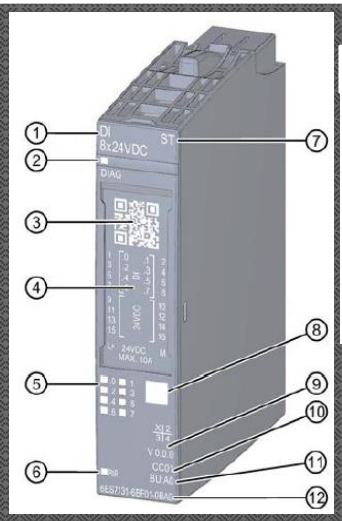


BU de type A0, variante foncée
Elle poursuit le groupe de potentiel précédent

Création d'un nouveau groupe de potentiel

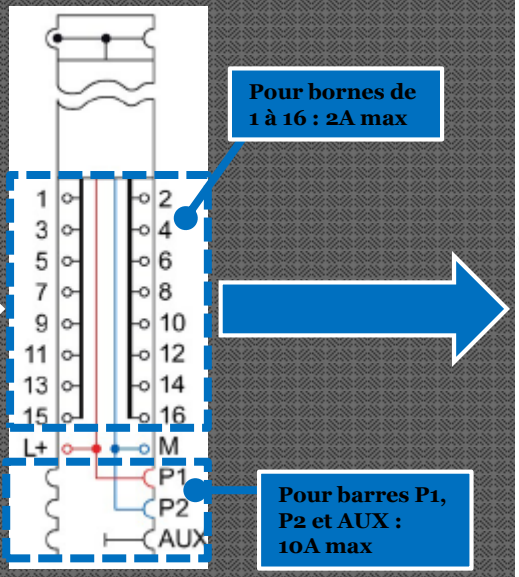
II.3: Périphérie décentralisée

Périphérie décentralisée SIEMENS ET200SP : Module d'entrée 6ES7 131-6BF00-0BA0

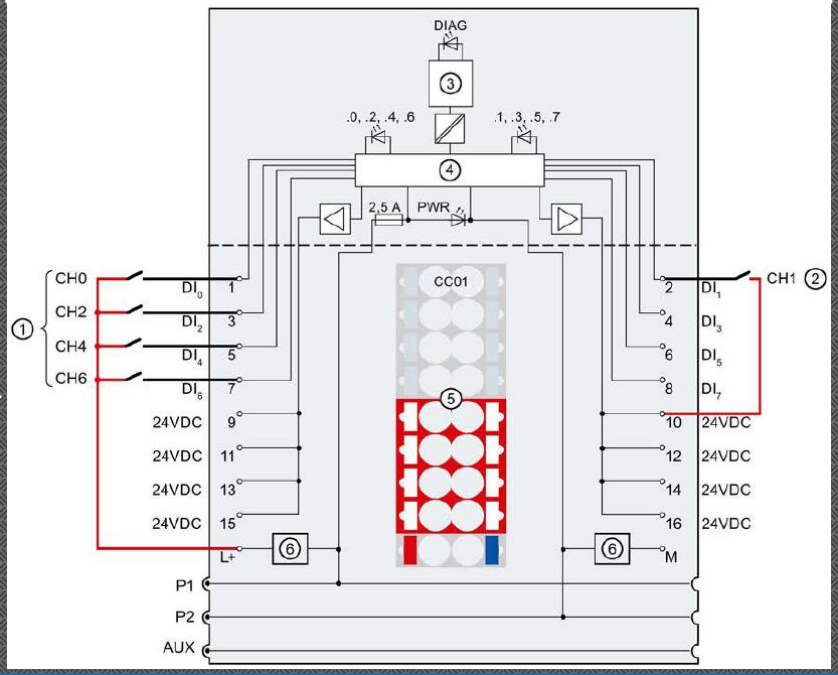


Présentation du module

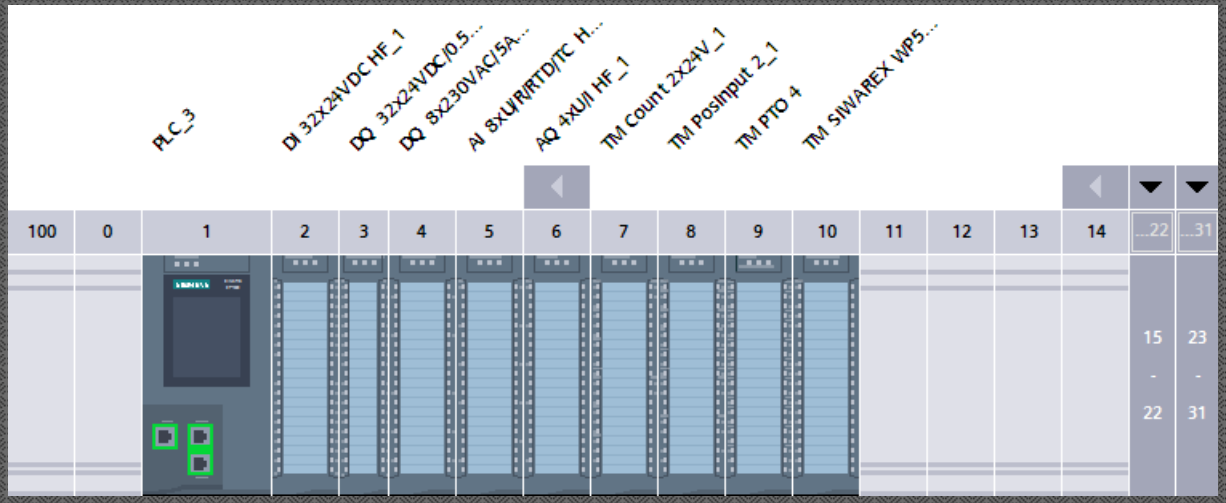
① Type et désignation du module	⑦ Classe de fonction
② LED de diagnostic	⑧ Code couleur du type de module
③ Code matriciel 2D	⑨ Version des fonctions et du firmware
④ Schéma de raccordement	⑩ Code couleur pour le choix des étiquettes de repérage couleur
⑤ LED d'état de la voie	⑪ Type de BU
⑥ LED pour la tension d'alimentation	⑫ N° d'article



Exemple de schéma de raccordement du module



Dans cette partie II.4, nous allons présenter les différents modules d'entrées /sorties proposées par les constructeurs.
Nous illustrerons cette partie avec une architecture SIEMENS s71500.



Description du modèle :

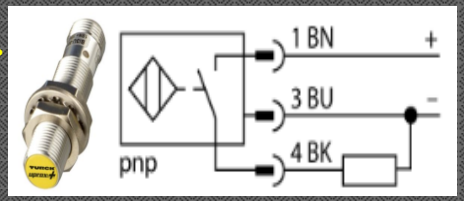
- **1: CPU 1515-2PN**
- **2: Module 32DI (6ES7 521-1BL00-0AB0)**
- **3: Module 32DQ (6ES7 522-1BL10-0AA0)**
- **4: Module 8DQ à relais (6ES7 522-5HF00-0AB0)**
- **5: Module 8AI (6ES7 531-7PF00-0AB0)**
- **6: Module 4AQ (6ES7 532-5ND00-0AB0)**
- **7: Module de comptage rapide TM Count 2x24V (6ES7 550-1AA00-0AB0)**
- **8: Module de détection de position TM PosInput 2 (6ES7 551-1AB00-0AB0)**
- **9: Module de commande pour entraînements pas-à-pas TM PTO 4 (6ES7 553-1AA00-0AB0)**
- **10: Module de pesage TM SIWAREX WP521 ST (7MH4 980-1AA01)**

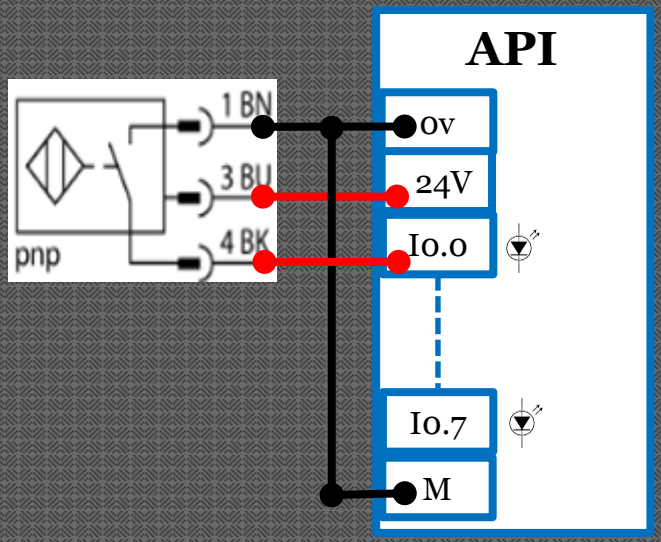
II.4: Cartes d'entrées/sorties TOR

Les cartes d'entrées TOR

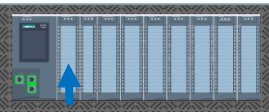
- ❑Souvent alimentés en 24V continu
- ❑Deux câblages possibles:
  - ❖Logique positive ou logique négative
- ❑ Les entrées sont généralement isolées électriquement de l'unité centrale et peu sensibles aux perturbations (filtrage – temps de retard)
- ❑ Lorsqu'une entrée TOR est active (état d'un capteur, état d'un BP, etc.) une LED s'allume sur la CPU.

Capteur inductif 3 fils



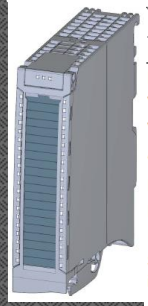


II.4: Cartes d'entrées/sorties TOR



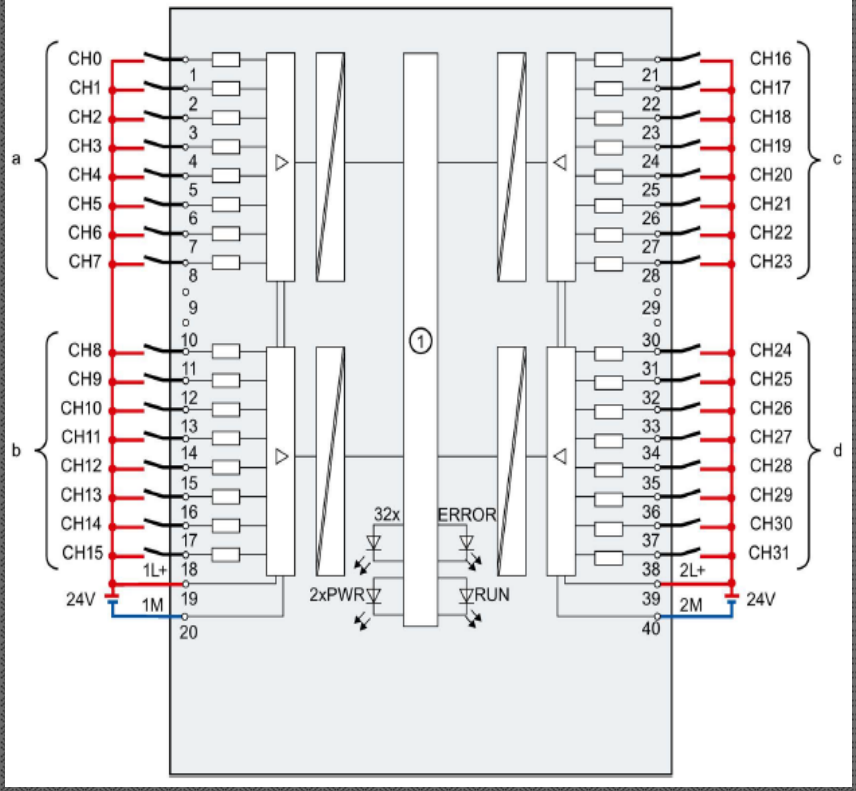
Module d'entrées TOR à transistors
Modèle à 32 entrées

❖ Module 32DI (6ES7 521-1BL00-0AB0)



	6ES7521-1BL00-0AB0
Informations générales	
Désignation de type du produit	DI 32x24VDC HF
Version fonctionnelle du matériel	FS01
Version de firmware	V2.1.0
• Mise à jour du firmware possible	Oui
Fonction du produit	
Données I&M	Oui ; I&M0 à I&M3
Ingénierie	
STEP 7 TIA Portal configurable / intégré à partir de la version	V13 SP1 / -
configurable avec STEP 7 / intégrée à partir de la version	V5.5 SP3 / -
PROFIBUS à partir de la version GSD/ révision GSD	V1.0 / V5.1
PROFINET à partir de la version GSD/ révision GSD	V2.3 / -
Mode de fonctionnement	
DI	Oui
Compteur	Oui
MSI	Oui
Tension d'alimentation	
Valeur nominale (CC)	24 V
plage admissible, limite inférieure (CC)	20,4 V
plage admissible, limite supérieure (CC)	28,8 V
Protection contre l'inversion de polarité	Oui
Courant d'entrée	
Consommation, max.	40 mA ; 20 mA par groupe avec une alimentation de 24 V CC
Puissance	
Consommation sur le bus interne	1,1 W
Puissance dissipée	
Puissance dissipée, typ.	4,2 W

Exemple de schéma de raccordement du module



II.4: Cartes d'entrées/sorties TOR

Les cartes de sorties tout ou rien

- Elles permettent de raccorder à l'automate, les différents pré-actionneurs tels que :
  - Distributeurs, vannes, contacteurs, voyants, etc.
- Les tensions de sorties usuelles sont de 5 volts en continu ou de 24, 48, 110, 220 volts en continu ou en alternatif.
- Les courants vont de quelques milliampères à quelques ampères.
- Ces cartes possèdent le plus souvent des transistors ou des relais et parfois des triacs.
- L'état de chaque sortie est visualisée par une diode électroluminescente, lorsque la sortie est active dans le programme.

Electrovanne Bobine: 230vAC/2A

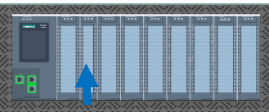
Contacteur Bobine: 24V DC/0,1A

Ce sont les caractéristiques de pré-actionneurs qui nous permettront de choisir entre un module de sorties à transistors ou un module de sorties à relais.

- Exemple de carte à transistor: 24V/0,5A par voie
- Exemple de carte à relais : contact sec tenues électriques: 230V/5A

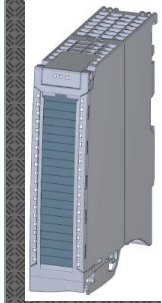
Ainsi pour l'électrovanne, on choisira une carte à relais et pour le contacteur les deux modèles de cartes peuvent convenir. Attention une carte à relais coûte plus chère qu'une carte à transistors.

II.4: Cartes d'entrées/sorties TOR



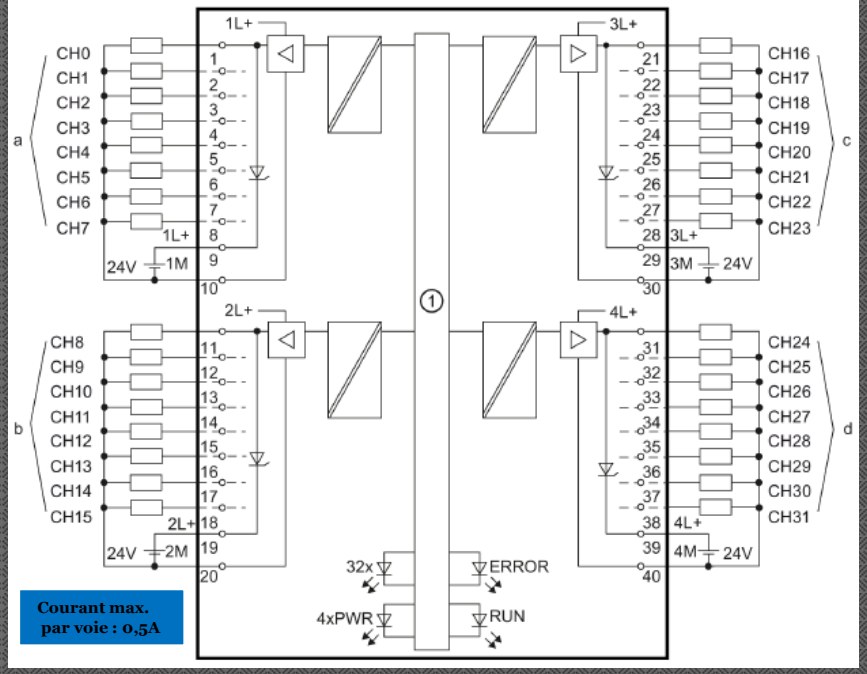
Module de sorties TOR à transistors
Modèle à 32 sorties

❖ Module 32DQ (6ES7 522-1BL10-0AA0)



	6ES7522-1BL10-0AA0
Désignation de type du produit	DQ 32x24VDC/0.5A BA
Informations générales	
Version du matériel	E01
Version de firmware	V1.0.0
Fonction du produit	
Données I&M	Oui
Ingénierie	
configurable avec STEP 7 TIA Portal/intégrée à partir de la version	V13 / V13
configurable avec STEP 7/intégrée à partir de la version	V5.5 SP3 / -
Mode de fonctionnement	
MSO	Oui
Tension d'alimentation	
Type de la tension d'alimentation	CC
Valeur nominale (CC)	24 V
Plage admissible, limite inférieure (CC)	20,4 V
Plage admissible, limite supérieure (CC)	28,8 V
Protection contre l'inversion de polarité	Oui ; via une protection par fusibles interne de 7 A par groupe
Courant d'entrée	
Consommation, max.	60 A
Puissance	
Consommation sur le bus interne	1,15 W
Puissance dissipée	
Puissance dissipée, typ.	3,8 W
Sorties TOR	
Type de sortie TOR	Transistor
Nombre de sorties	32
type PNP	Oui
Sorties TOR, paramétrables	Non
Protection contre les courts-circuits	Oui
• Seuil de réponse, typ.	1 A
Limitation de la tension de coupure inductive à	L+ (-53 V)
Commande d'une entrée TOR	Oui

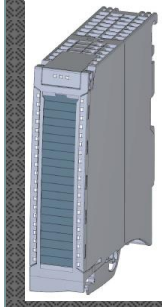
Exemple de schéma de raccordement du module



II.4: Cartes d'entrées/sorties TOR

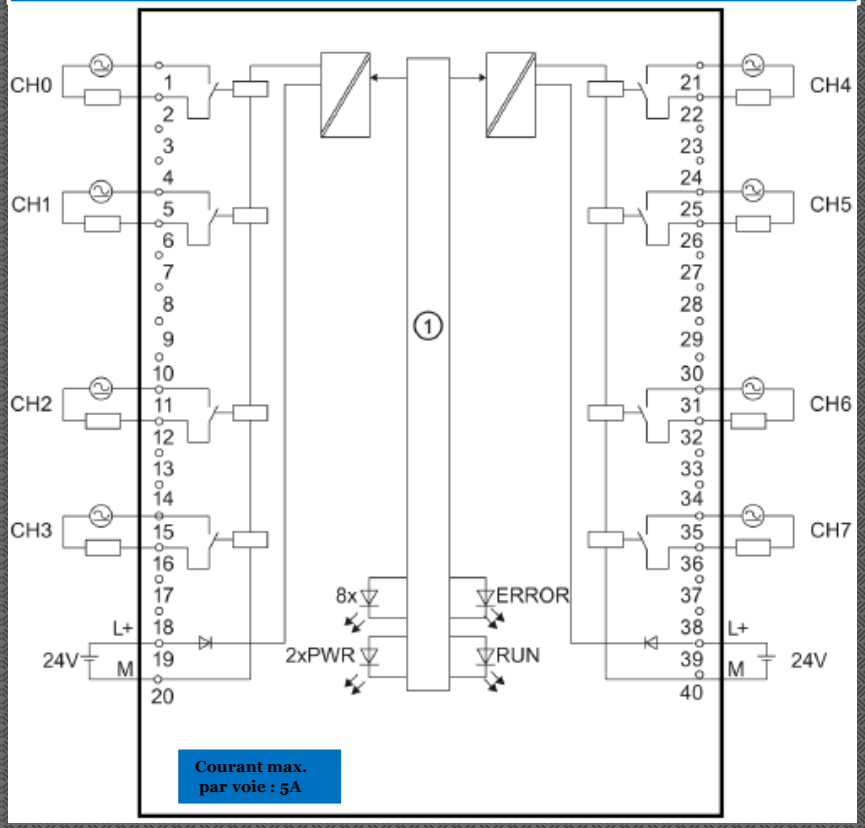
Module de sorties TOR à relais
Modèle à 8 sorties

❖ Module 8DQ à relais (6ES7 522-5HF00-0AB0)



	6ES7522-5HF00-0AB0
Désignation de type du produit	DQ 8x230VAC/5A ST (Relay)
Informations générales	
Version matérielle	E01
Version de firmware	V2.0.0
Fonction du produit	
Données I&M	Oui ; IM0 à IM3
Ingénierie	
configurable avec STEP 7 TIA Portal / intégrée à partir de la version	V12.0 / V12.0
configurable avec STEP 7 / intégrée à partir de la version	V5.5 SP3 / -
Mode de fonctionnement	
MSO	Oui
Tension d'alimentation	
Type de la tension d'alimentation	CC
Valeur nominale (CC)	24 V
plage admissible, limite inférieure (CC)	20,4 V
Plage admissible, limite supérieure (CC)	28,8 V
Protection contre l'inversion de polarité	Oui
Courant d'entrée	
Consommation, max.	80 mA
Puissance	
Consommation sur le bus interne	0,8 W
Puissance dissipée	
Puissance dissipée, typ.	5 W
Sorties TOR	
Type de sortie TOR	Relais
Nombre de sorties	8
à commutation M	Oui
à commutation P	Oui
Sorties TOR, paramétrables	Oui
Protection contre les courts-circuits	Non
Commande d'une entrée TOR	possible

Exemple de schéma de raccordement du module



II.5: Cartes d'entrées/sorties ANA

Introduction:

Les automates ne peuvent traiter des valeurs analogiques que sous forme de configurations binaires. Des capteurs de mesure raccordables au module analogique acquièrent des grandeurs physiques, par ex. pression ou température.

Cette valeur analogique est mesurée sous forme de courant, tension ou résistance par le module d'entrées analogiques.

Afin que la CPU puisse traiter la valeur de courant ou de tension acquise, un convertisseur analogique-numérique (CAN) intégré au module d'entrées analogiques la convertit en un nombre entier de 16 bits.

Les capteurs de mesure suivants peuvent être utilisés en fonction du type de mesure :

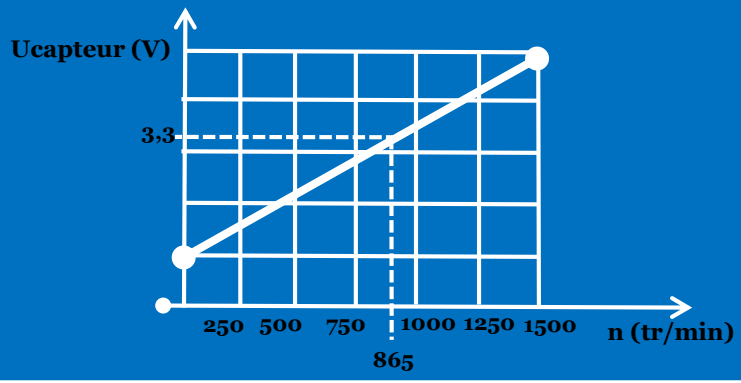
- **Capteurs de tension**
- **Capteurs de courant**
  - Transducteur de mesure 2 fils
  - Transducteur de mesure 4 fils
- **Capteurs résistifs**
  - Montage 4 fils
  - Montage 3 fils
  - Montage 2 fils
- **Thermocouples**

Exemple :

Un capteur de mesure convertit une plage de vitesse **(n)** de 0 à 1500 tr/min en une plage de tension **(U*capteur*)** de 1 à 5 V.

Pour une vitesse de rotation mesurée de 865 tr/min, le capteur de mesure fournit une valeur de tension de 3,3 V.

$$Ucapteur = \frac{(5-1)}{1500} * n + 1$$



II.5: Cartes d'entrées/sorties ANA

Processus

Grandeur physique

Signal analogique normalisé

Dynamo

Transducteur

Capteur

- Pression
- Température
- Débit
- Vitesse
- Valeur pH
- etc.

± 500mV
± 1V
± 5V
± 10V
± 20mA
4 à 20mA
etc.

Moteur

Variateur

Grandeur physique

Pré-actionneur normalisé

Actionneur

Module d'entrées analogiques API

Mémorisation du résultat

CAN

IW.....
IW.....
⋮
IW.....

Module de sorties analogiques API

Conversion et mémorisation du résultat

CNA

QW.....
QW.....
⋮
QW.....

Programme utilisateur

Lecture le module d'entrée analogique

accès directement au module d'I :P

accès via la MIE rafraîchie en début de cycle programme

Programme CPU: Mise à l'échelle des signaux

accès directement au module d'O :P

accès via la MIS rafraîchie en fin de cycle programme

Ecriture sur le module de sortie analogique

II.5: Cartes d'entrées/sorties ANA

CAN et CNA : Un CAN (conversion analogique numérique) est un dispositif électronique permettant la conversion d'un signal analogique en un signal numérique (Entrée ANA sur API).
Un CNA (conversion numérique analogique) permet la conversion d'un signal numérique en signal analogique (Sortie ANA sur API).
Leur résolution dépend d'une nombre de bits. Pour un CAN ou CNA à N bits, le nombre d'états possibles en sortie est 2^N, ce qui permet d'exprimer des signaux numériques de 0 à 2^N-1 en code binaire naturel.

Exemple chez SIEMENS:
Représentation des valeurs analogiques et résolution

N° de bit		Unités		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Poids du bit		Dec.	Hex.	VZ	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Reso-lution in bits + sign	8	128	80	*	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0
	9	64	40	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0	0
	10	32	20	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0
	11	16	10	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0
	12	8	8	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0
	13	4	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0
	14	2	2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0
	15	1	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

* = **0** or **1**

II.5: Cartes d'entrées/sorties ANA

Exemple chez SIEMENS :
Représentation des valeurs analogiques pour différentes plages de mesure. Le format de variable pour une grandeur analogique reste le le nombre entier sur un format de 16 bits (IW…), une valeur signée correspond donc à : $2^{16} - 1 = 65535$ ***soit une plage signée entre*** -32768 à $+32767$

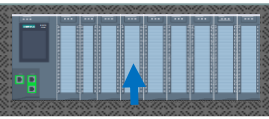
Plage	Tension Par ex:		Courant Par ex:		Resistance Par ex:		Température ex: Pt100 (Standard)	
	Plage ± 10V	Unités	Plage 4 à 20mA	Unités	Plage 0 à 300Ohm	Unités	Plage -200 à +850ºC	Unités
Débordement	>= 11.759	32767	>= 22.815	32767	>=352.778	32767	>= 1000.1	32767
Domaine de dépassement	11.7589 : 10.0004	32511 : 27649	22.810 : 20.0005	32511 : 27649	352.767 : 300.011	32511 : 27649	1000.0 : 850.1	10000 : 8501
Plage nominale	10.00 7.50 : -7.5 -10.00	27648 20736 : -20736 -27648	20.000 16.000 : : 4.000	27648 20736 : : 0	300.000 225.000 : : 0.000	27648 20736 : : 0	850.0 : : : -200.0	8500 : : : -2000
Domaine de dépassement	- 10.0004 : - 11.759	- 27649 : - 32512	3.9995 : 1.1852	- 1 : - 4864	Valeurs Négatives impossible	- 1 : - 4864	- 200.1 : - 243.0	- 2001 : - 2430
Débordement	<= - 11.76	- 32768	<= 1.1845	- 32768		- 32768	<= - 243.1	- 32768

II.5: Cartes d'entrées/sorties ANA

Exemple chez SIEMENS :
Représentation des valeurs par les sorties analogiques

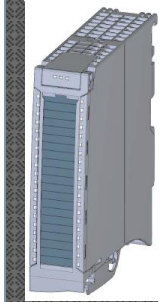
Plage	Unité	Tension			Courant		
		Plage de sortie			Plage de sortie		
		0 à 10V	1 à 5V	± 10V	0 à 20mA	4 à 20mA	± 20mA
Débordement	>=32767	0	0	0	0	0	0
Domaine de dépassement	32511 : 27649	11.7589 : 10.0004	5.8794 : 5.0002	11.7589 : 10.0004	23.515 : 20.0007	22.81 : 20.005	23.515 : 20.0007
Plage nominale	27648 : 0 : - 6912	10.0000 : 0	5.0000 : 1.0000	10.0000 : 0 : : : : : : : -10.0000	20.000 : 0	20.000 : 4.000	20.000 : 0 : : : : : : : -20.000
	- 6913 : : : - 27648	0	0.9999 0		0	3.9995 0	
			0			0	
Domaine de dépassement	- 27649 : - 32512			- 10.0004 : - 11.7589			- 20.007 : - 23.515
Débordement	<=- 32513			0			0

II.5: Cartes d'entrées/sorties ANA



Module d'entrées analogiques

❖ Module 8AI (6ES7 531-7PF00-0AB0)



	6ES7531-7PF00-0AB0
Informations générales	
Désignation de type du produit	AI 8xU/R/RTD/TC HF
Version du matériel	FS01
Version de firmware	V1.1.0
• Mise à jour du firmware possible	Oui
Fonction du produit	
Données I&M	Oui ; I&M0 à I&M3
Plage de mesure échelonnable	Oui
Valeurs de mesure évolutives	Non
Adaptation de la plage de mesure	Non
Ingénierie	
configurable avec STEP 7 TIA Portal/intégrée à partir de la version	V14 / -
configurable avec STEP 7 / intégrée à partir de la version	V5.5 SP3 / -
PROFIBUS à partir de la version GSD/révision GSD	V1.0 / V5.1
PROFINET à partir de la version GSD/révision GSD	V2.3 / -
Mode de fonctionnement	
Suréchantillonnage	Non
MSI	Oui
Configuration en RUN (CiR)	
Reparamétrage en mode RUN possible	Oui
Calibrage en RUN possible	Oui
Tension d'alimentation	
Valeur nominale (CC)	24 V
plage admissible, limite inférieure (CC)	20,4 V
Plage admissible, limite supérieure (CC)	28,8 V
Protection contre l'inversion de polarité	Oui
Courant d'entrée	
Consommation, max.	55 mA ; pour une alimentation de 24 V CC
Puissance	
Consommation sur le bus interne	0,85 W

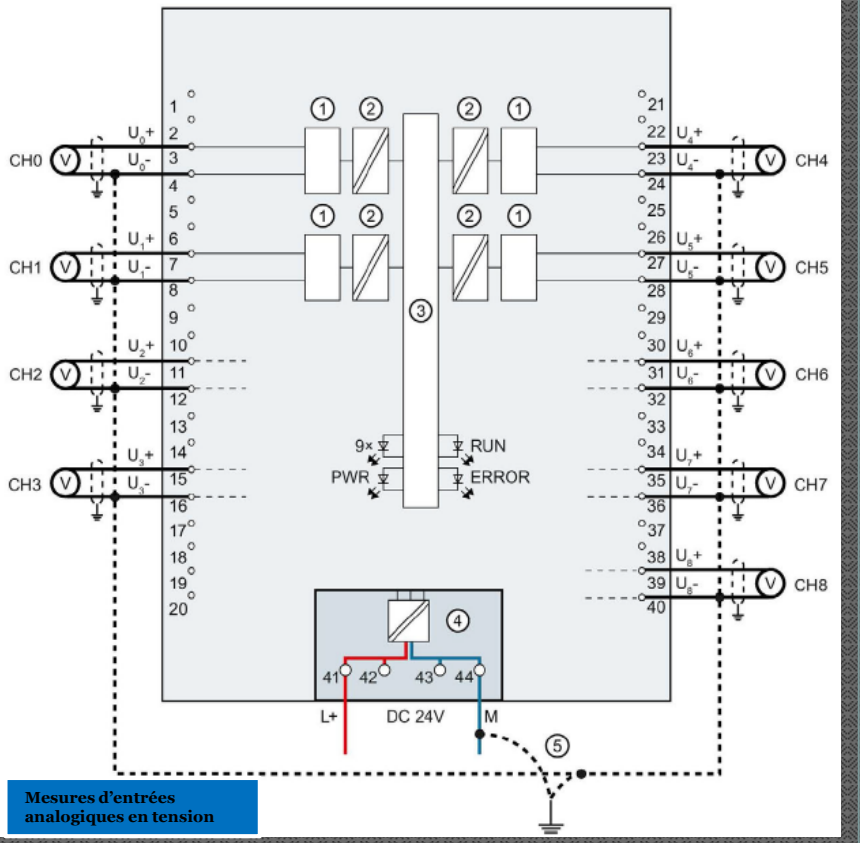
Le type de mesure est paramétrable pour chaque voie avec la suite logiciel TIA PORTAL, associé bien entendu au câblage adéquat

Totally Integrated Automation PORTAL

Ce module possède les caractéristiques techniques suivantes :

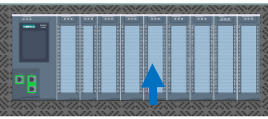
- ❑ **9 entrées analogiques avec séparation galvanique**
- ❑ **Types de mesure paramétrables pour chaque voie :**
- ❖ **Tension**
- ❖ **Résistance**
- ❖ **Thermomètre à résistance (RTD)**
- ❖ **Thermocouple (TC)**

Exemple de schéma de raccordement du module



Mesures d'entrées analogiques en tension

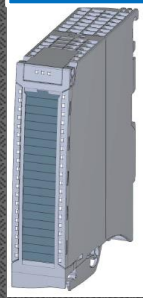
II.5: Cartes d'entrées/sorties ANA



Module de sorties analogiques

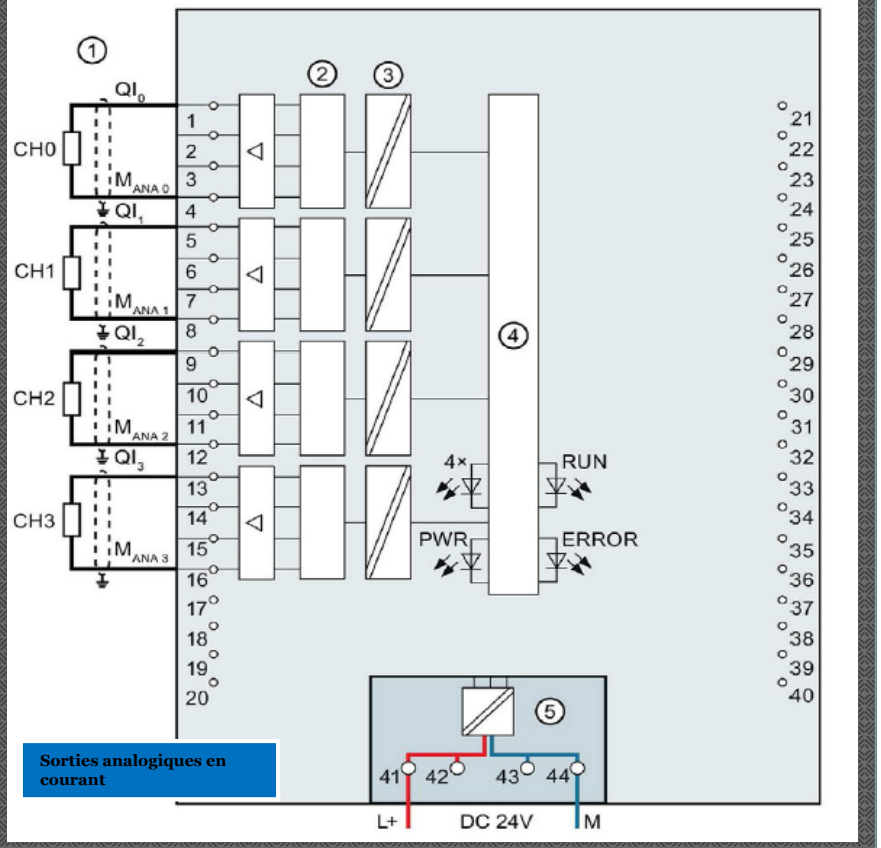
Le module possède les caractéristiques techniques suivantes :
- ❑ **4 sorties analogiques avec séparation galvanique**
- ❖ **Sortie de tension sélectionnable voie par voie**
- ❖ **Sortie de courant sélectionnable voie par voie**

❖ Module 4AQ (6ES7 532-5ND00-0AB0)



	6ES7532-5ND00-0AB0
Informations générales	
Désignation de type du produit	AQ 4xU/I HF
Version du matériel	FS01
Version de firmware	V1.1.0
• Mise à jour du firmware possible	Oui
Fonction du produit	
Données I&M	Oui ; I&M0 à I&M3
Ingénierie	
configurable avec STEP 7 TIA Portal /intégrée à partir de la version	V14 / -
configurable avec STEP 7 / intégrée à partir de la version	V5.5 SP3 / -
PROFIBUS à partir de la version GSD/révision GSD	V1.0 / V5.1
PROFINET à partir de la version GSD/révision GSD	V2.3 / -
Mode de fonctionnement	
Suréchantillonnage	Non
MSO	Oui
Configuration en RUN (CiR)	
Reparamétrage en mode RUN possible	Oui
Calibrage en RUN possible	Oui
Tension d'alimentation	
Valeur nominale (CC)	24 V
plage admissible, limite inférieure (CC)	20,4 V
Plage admissible, limite supérieure (CC)	28,8 V
Protection contre l'inversion de polarité	Oui
Courant d'entrée	
Consommation, max.	160 mA ; pour une alimentation de 24 V CC
Puissance	
Consommation sur le bus interne	0,95 W
Puissance dissipée	
Puissance dissipée, typ.	5 W

Exemple de schéma de raccordement du module



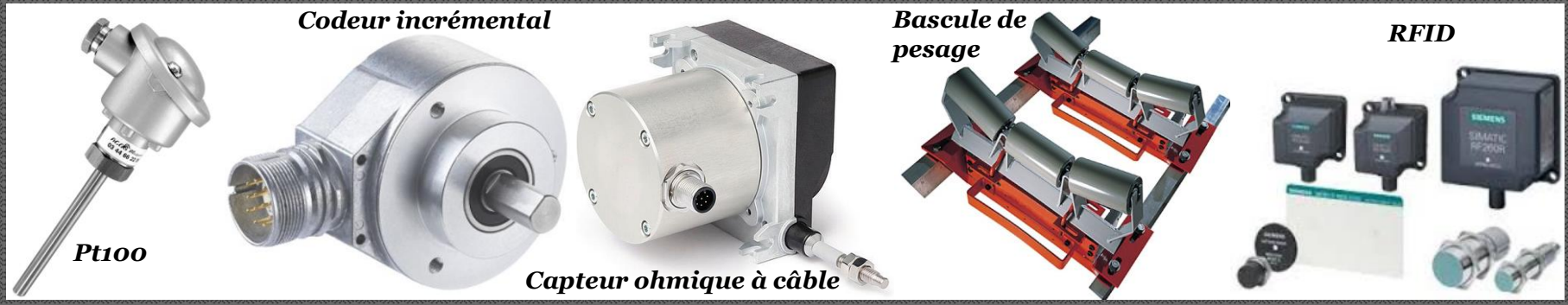
Le type de sortie est paramétrable pour chaque voie avec la suite logiciel TIA PORTAL, associé bien entendu au câblage adéquat

Totally Integrated Automation PORTAL

II.6: Cartes spécifiques

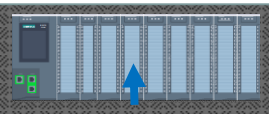
❑ Dans l'automatisme industriel, il existe un certains nombres de capteurs et de pré-actionneurs spécifiques à certaines applications.
Parmi ces applications, on retrouve souvent comme:

- **La mesure de température (exemples: PT100, PT1000, Thermocouple, etc.)**
- **Le comptage rapide (exemples : codeur incrémental, capteur inductif, etc.)**
- **La mesure de positions (exemples : capteur ohmique à câble, codeur absolu, etc.**
- **La commande d'axe (exemple: commande PTO pour asservissement en position , etc.)**
- **Les systèmes de pesage (exemple: bascule de pesage avec jauges de contrainte, etc.)**
- **Les systèmes sans contact (exemples: RFID, système d'identification optique, etc.)**
- **Etc…**



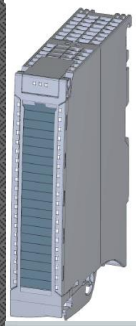
❑ Pour répondre à ces besoins, les constructeurs proposent des cartes spécifiques entièrement dédiées à ces applications. Ceci permet une gestion directe par l'API.

II.6: Cartes spécifiques



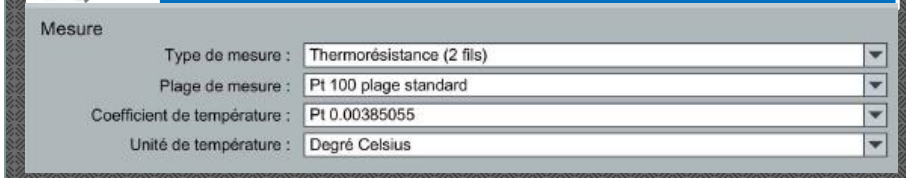
Module d'entrées analogiques pour la mesure de température

- ❖ **Module 8AI (6ES7 531-7PF00-0AB0)**



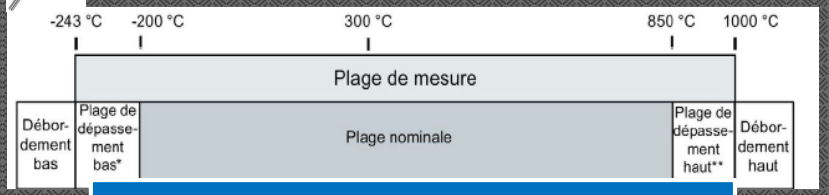
Ce module d'entrées analogiques présentée en page 121 est polyvalent, (entrée ANA en tension), il permet également de **mesurer des températures via des capteurs de type PT100, PT1000 (technologies 2 fils, 3 fils et 4 fils) et également des thermocouples.**
La configuration s'effectue avec l'outil logiciel TIA-PORTAL.

Totally Integrated Automation PORTAL



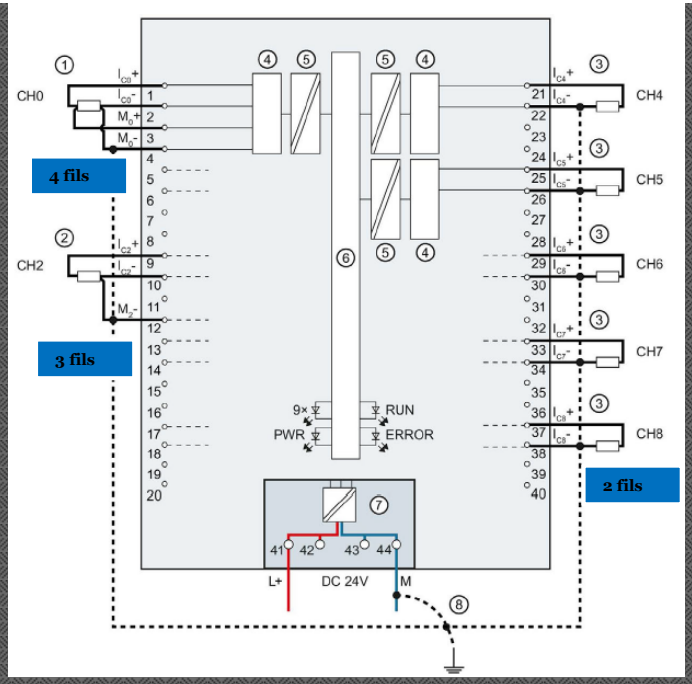
- ❖ **Thermomètre à résistance (RTD)**
- ❖ **Thermocouple (TC)**

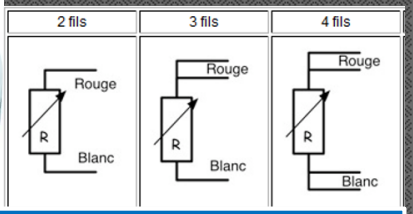
Ex: Pt100



Exemple d'un plage de mesure d'unePT100

Exemple de schéma de raccordement du module



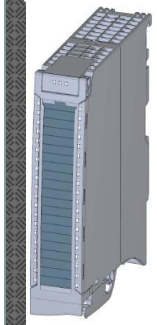


3 montages industriels de PT100

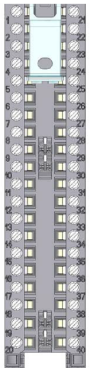
II.6: Cartes spécifiques

Module de comptage rapide

❖ TM Count 2x24V (6ES7 550-1AA00-0AB0)



Le comptage consiste à acquérir et totaliser des événements.
Les compteurs du module technologique acquièrent des signaux de capteur et des impulsions et les évaluent de manière appropriée. **(attention à regarder la fréquence max des impulsions en fonction de la carte)**
Le sens de comptage peut être prédéfini par des signaux de capteur ou d'impulsion appropriés.
Les entrées TOR permettent de commander les processus de comptage. Vous pouvez commuter les sorties TOR exactement aux valeurs de comptage définies indépendamment du programme utilisateur.

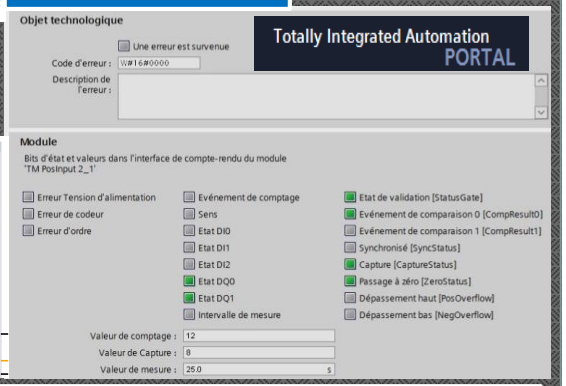


	Nom de signal	Désignation				
		Codeur incrémental 24 V		Générateur d'impulsions 24 V		
		avec signal N	sans signal N	avec signal de sens	sans signal de sens	comptage/ décomptage
	Voie de comptage 0					
1	CH0.A	Signal de capteur A		Signal de comptage A		Signal de comptage A
2	CH0.B	Signal de capteur B		Signal de sens B	—	Signal de décomptage B
3	CH0.N	Signal de capteur N	—			
4	DI0.0	Entrée TOR DI0				
5	DI0.1	Entrée TOR DI1				
6	DI0.2	Entrée TOR DI2				
7	DQ0.0	Sortie TOR DQ0				
8	DQ0.1	Sortie TOR DQ1				
	Alimentation des capteurs et masse pour les deux voies de comptage					
9	24VDC	Alimentation de capteur 24 V				
10	M	Masse pour l'alimentation des capteurs, les entrées TOR et les sorties TOR				
	Voie de comptage 1					
11	CH1.A	Signal de capteur A		Signal de comptage A		Signal de comptage A
12	CH1.B	Signal de capteur B		Signal de sens B	—	Signal de décomptage B
13	CH1.N	Signal de capteur N	—			
14	DI1.0	Entrée TOR DI0				
15	DI1.1	Entrée TOR DI1				
16	DI1.2	Entrée TOR DI2				
17	DQ1.0	Sortie TOR DQ0				
18	DQ1.1	Sortie TOR DQ1				
19 - 40	—	—				

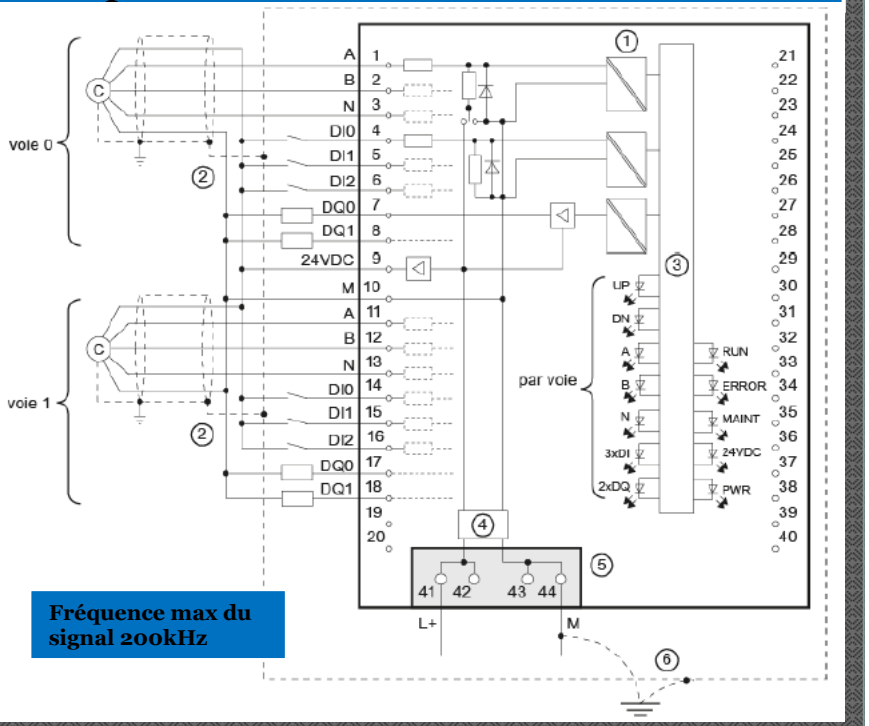
Ex: Codeur incrémental

+1 Comptage et mesure

Fonction technologique avec TIA PORTAL



Exemple de schéma de raccordement du module

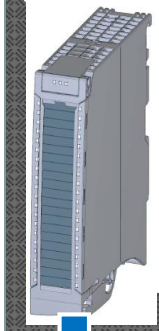


Fréquence max du signal 200kHz

II.6: Cartes spécifiques

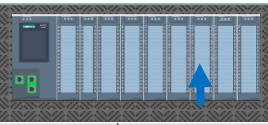
Module de détection de position

❖ TM PosInput 2 (6ES7 551-1AB00-0AB0)



le module technologique TM PosInput 2 peut être associé à un codeur absolu SSI pour détecter la position.

Le module technologique lit la valeur de position via une interface série synchrone du codeur absolu SSI et la met à disposition de la commande.
Possibilité de raccorder un codeur incrémental ou un dispositif générant des impulsions

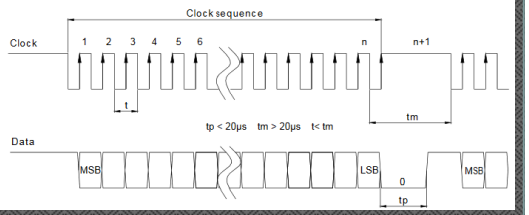


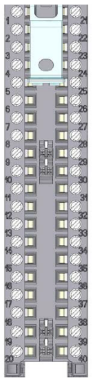
Ex: Codeur absolu SSI
Interface Série Synchrone

	Connecteur 12 broches. Numéro des contacts				Câble couleur des fils	
Interface	SL	S1 / S4	S2	S3	SL	S1
Clock -	1	1	1	1	Yellow / (Jaune)	Yellow / (Jaune)
Clock +	2	2	2	2	Green / (Vert)	Green / (Vert)
Data +	3	3	3	3	Grey / Gris	Grey / Gris
Data -	4	4	4	4	Pink / (Rose)	Pink / (Rose)
Sens ou Complement	8	8	8	8	Red / (Rouge)	Red / (Rouge)
+ U_b= 10-30 V	11	11	11	11	Brown / (Brun)	Brown / (Brun)
GND	12	12	12	12	White / (Blanc)	White / (Blanc)

L'interface SSI se compose de 2 paires de fils:
Une paire pour la transmission des signaux d'horloge du maître
et l'autre paire pour transmettre les données de l'esclave (codeur).
Une troisième paire de fils permet en option l'alimentation du codeuR

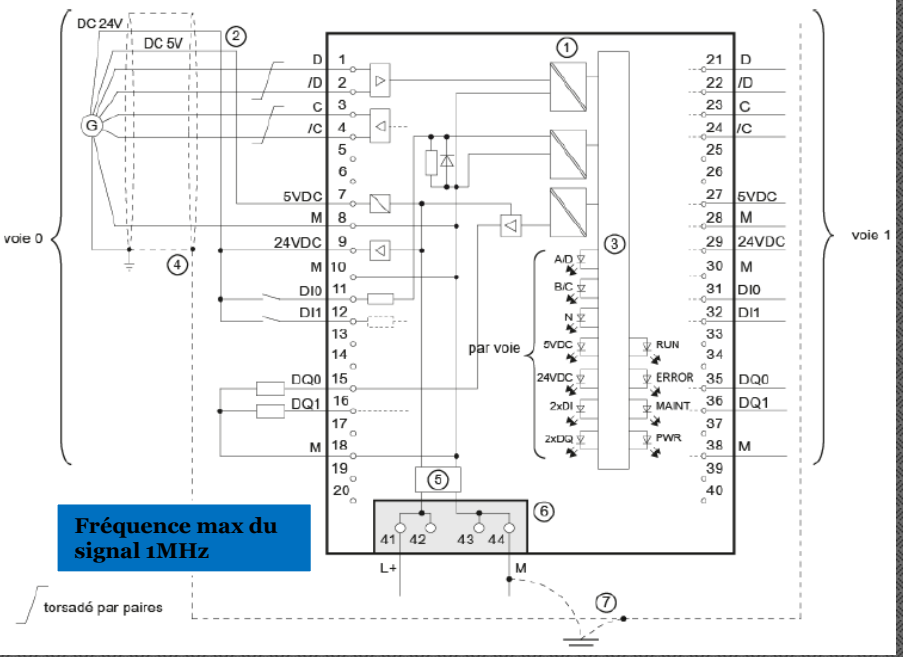
Séquence de transmission





	Nom de signal	Désignation					
		Codeur incrémental RS422/TTL		Générateur d'impulsions RS422/TTL			Codeur absolu SSI
		avec signal N	sans signal N	avec signal de sens	sans signal de sens	comptage/ décomptage	
voie 0							
1	CH0.A ou CH0.D	Signal de capteur A		Signal de comptage A		Signal de comptage A	Signal de données SSI DAT
2	/CH0.A ou /CH0.D	Signal de capteur /A (uniquement RS422)		Signal de comptage/A (uniquement RS422)		Signal de comptage/A (RS422 uniquement)	Signal de données SSI /DAT
3	CH0.B ou CH0.C	Signal de capteur B		Signal de sens B	—	Signal de décomptage B	Signal de cycle CLK SSI
4	/CH0.B ou /CH0.C	Signal de capteur /B (uniquement RS422)		Signal de sens/B (RS422 unique-ment)		Signal de décomptage /B (RS422 uniquement)	Signal de cycle SSI /CLK
5	CH0.N	Signal de capteur N	—				
6	/CH0.N	Signal de capteur/N (RS422 unique-ment)	—				
7	5VDC	Alimentation de capteur 5 V					
8	M	Masse pour l'alimentation des capteurs et les entrées TOR					
9	24VDC	Alimentation de capteur 24 V					
10	M	Masse pour l'alimentation des capteurs et les entrées TOR					
11	DI0.0	Entrée TOR DI0					
12	DI0.1	Entrée TOR DI1					
13	—	—					
14	—	—					
15	DQ0.0	Sortie TOR DQ0					
16	DQ0.1	Sortie TOR DQ1					
17	—	—					
18	M	Masse pour les sorties TOR					
19 - 20	—	—					

Exemple de schéma de raccordement du module codeur de type absolu SSI

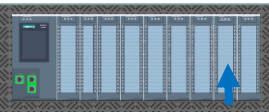


Fréquence max du signal 1MHz

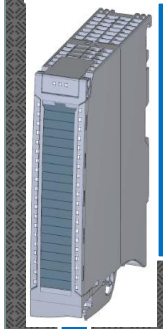
II.6: Cartes spécifiques

Module de commande pour entrainements PTO pour moteur pas à pas

Ex: Entrainement Servo moteur Interface 24V asymétrique ou RS422



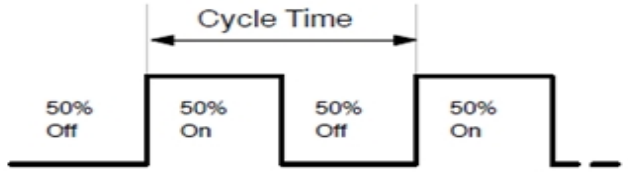
❖ TM PTO 4 (6ES7 553-1AA00-0AB0)

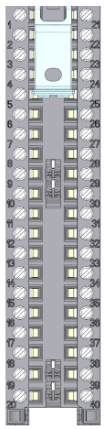


Pulse Train Output est une interface simple et universelle entre une commande et un entraînement.
PTO est prise en charge par de nombreux entraînements pas à pas et servomoteurs .
PTO est également désignée comme une interface impulsions-direction.

Fonction technologique avec TIA PORTAL

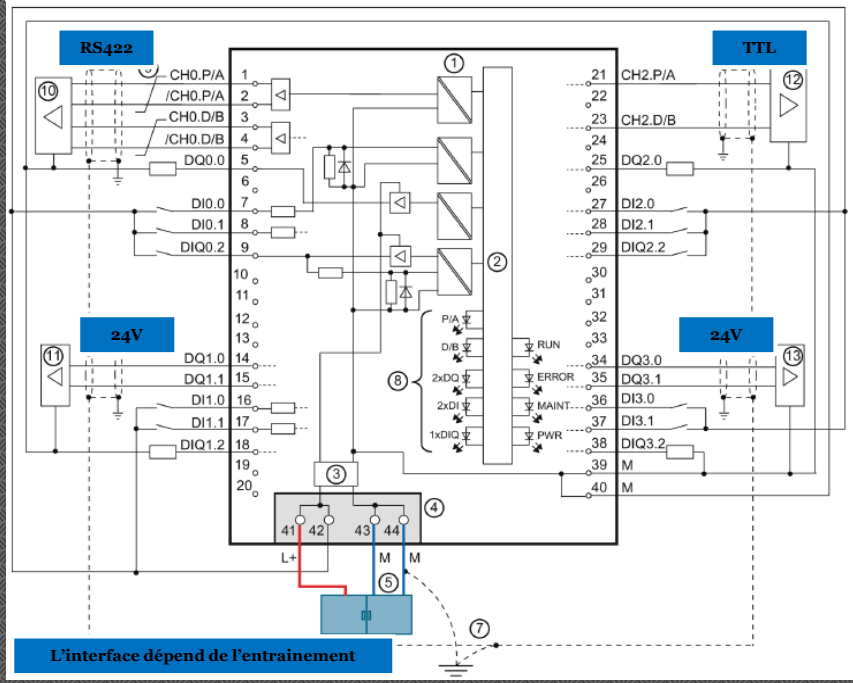
Pulse-Train-Output (PTO) est une technique de modulation dédiée à l'asservissement de position d'axes ou en contrôle de vitesse.
La fréquence est réglable et le rapport cyclique fixe à 50%:





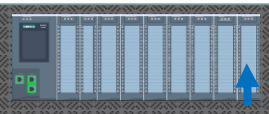
	Nom de signal	Désignation		
		24 V, asymétrique	RS422, symétrique	TTL (5 V), asymétrique
	Voie 0			
1	CH0.P/A	—	Signal d'impulsion P/A	Signal d'impulsion P/A
2	/CH0.P/A		Signal d'impulsion P/A inversé	—
3	CH0.D/B		Signal d'impulsion D/B	Signal d'impulsion D/B
4	/CH0.D/B		Signal d'impulsion D/B inversé	—
5	DQ0.0	Signal d'impulsion P/A	Sortie TOR DQ0	Sortie TOR DQ0
6	DQ0.1	Signal d'impulsion D/B	—	—
7	DI0.0	Entrée TOR DI0		
8	DI0.1	Entrée TOR DI1		
9	DIQ0.2	Entrée/sotie TOR DIQ2		
	Voie 1			
10	CH1.P/A	—	Signal d'impulsion P/A	Signal d'impulsion P/A
11	/CH1.P/A		Signal d'impulsion P/A inversé	—
12	CH1.D/B		Signal d'impulsion D/B	Signal d'impulsion D/B
13	/CH1.D/B		Signal d'impulsion D/B inversé	—
14	DQ1.0	Signal d'impulsion P/A	Sortie TOR DQ0	Sortie TOR DQ0
15	DQ1.1	Signal d'impulsion D/B	—	—
16	DI1.0	Entrée TOR DI0		
17	DI1.1	Entrée TOR DI1		
18	DIQ1.2	Entrée/sotie TOR DIQ2		
19	—	—		
20				

Exemple de schéma de raccordement du module



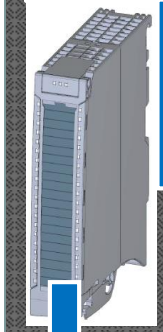
L'interface dépend de l'entrainement

II.6: Cartes spécifiques



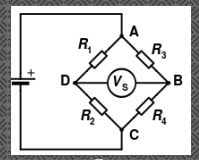
Module de pesage

❖ **TM SIWAREX WP521 ST (7MH4 980-1AA01)**



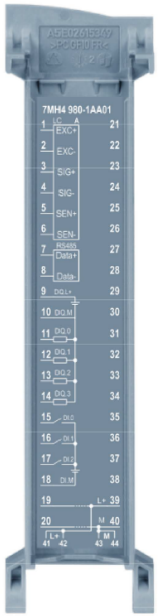
La tâche primaire de l'électronique de pesage consiste aux mesures et enregistrements de valeurs de poids avec grande précision.
Grâce à l'intégration dans les solutions d'automatisme, il est possible de traiter le poids directement dans l'API.

Bascule de pesage



Jauge de contrainte sur pont de Wheatstone

SIWATOOL : Logiciel de configuration pour systèmes de pesage

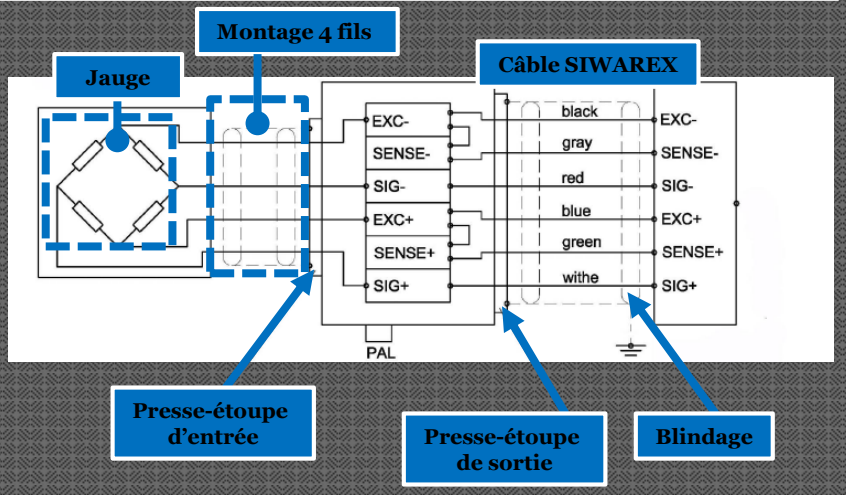


Brochage du connecteur WP521

WP521 ST					
Peson EXC+	1	21			
Peson EXC-	2	22			
Peson SIG+	3	23			
Peson SIG-	4	24			
Peson SEN+	5	25			
Peson SEN-	6	26			
RS485, D+	7	27			
RS485, D-	8	28			
DQ.L+ (24V DQ)	9	29			
DQ.M (0V DQ)	10	30			
DQ.0	11	31			
DQ.1	12	32	(Broches 21 ... 38 inutilisable)		
DQ.2	13	33			
DQ.3	14	34			
DI.0	15	35			
DI.1	16	36			
DI.2	17	37			
DI.M (0V DI)	18	38			
L+ (si pontté)	19	39	L+ (de 41, 42)	Broches 19 et 39 pontables	
M (si pontté)	20	40	M (de 43, 44)	Broches 20 et 40 pontables	
	41 42	43 44			
	L+	M			

Exemple de schéma de raccordement du module

Raccordement de pesons à jauge extensiométrique en montage 4 ou 6 fils (par voie)



II.6: Cartes spécifiques

Systèmes RFID

LA RFID, technologie sans contact permet le transfert de données rapide et sûr entre des transpondeurs et un API.

Applications: Gestion des flux des matières, suivi de production, flexibilité des lignes, etc.

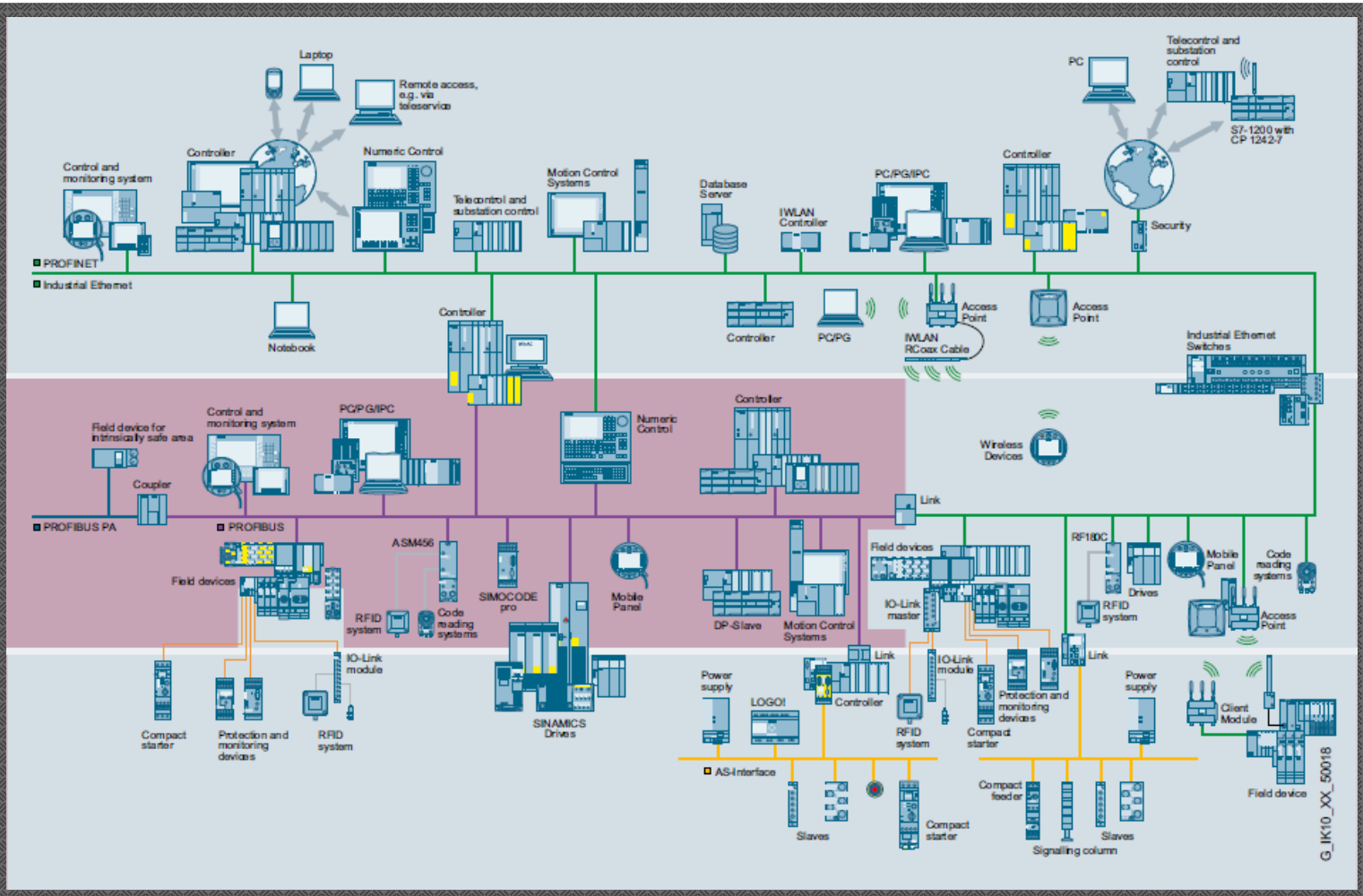
Systèmes d'identification optiques

Les systèmes d'identification optiques assurent une lecture de codes 1D/2D, reconnaissance de texte (OCR) ainsi que la reconnaissance d'objets.
Applications: Optimiser la fiabilité de production, tests de qualités, traçabilité des produits, etc.

Intégration de ces solutions dans les API.

II.6: Cartes spécifiques

- **Les automates possèdent de plus en plus de ports de communications.**
  - Pour la programmation, le réglage ou la maintenance (PC, console de programmation, etc.)
  - Pour la supervision, le dialogue inter-automates (Ethernet – TCP/IP – PROFIBUS, etc.)
  - Pour la communication avec des capteurs et les pré-actionneurs (bus ASI, IO/Link, etc.)



II.7: Présentation des suites logicielles

Si nous reprenons les produits présentés précédemment (SIEMENS et SCHNEIDER), voici leur logiciel :

SIEMENS

S7300

S71500

S7400
(contrôle de process)

Schneider Electric

M340

M580

Prenium

Quantum
(contrôle de process)

❑Les automates SIEMENS

- ❖S7300
- ❖S71500
- ❖S7400

❑Les automates SCHNEIDER

- ❖M340
- ❖M580
- ❖Prenium
- ❖Quantum

❑ TIA PORTAL

❑ UNITY PRO

II.7: Présentation des suites logicielles

❑ UNITY PRO

Unity Pro est le logiciel commun de programmation, mise au point et exploitation des gammes d'automates Modicon M340, M580, M580S, Premium et Quantum.

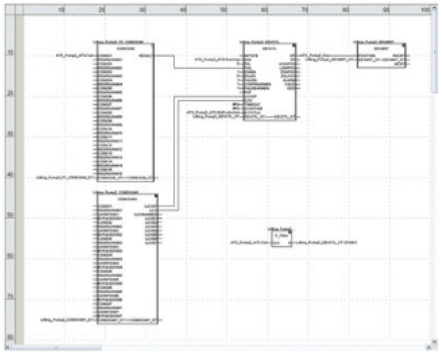
Unity Pro est un logiciel multitâche qui offre les fonctionnalités suivantes :

- **5 langages de programmation IEC 61131-3**
- **Possibilité de migration d'anciennes gammes Modicom (LL984)**
- **Intégration des équipements dans la norme FDT/DTM**
- **Librairie de blocs fonctions (DFB) intégrée et personnalisable**
- **Simulateur automate sur PC pour valider votre programme avant installation**
- **Tests intégrés et diagnostic**
- **Cybersécurité**

Principales	
Gamme de produits	Logiciel Unity Pro
Fonction produit	Logiciel
Langue utilisateur	Chinois Anglais Français Allemand Italien Espagnol
Format	DVD-ROM
Système d'exploitation	Windows 7 32 bit Windows 7 64 bits Windows Server 2012 R2 Windows 10 32 bits Windows 10 64 bits

II.7: Présentation des suites logicielles

Unity Pro



Editeur langage FBD

❑ LANGAGES API

Langages de programmation

Les cinq langages conformes à la norme IEC 61131-3

Les cinq langages de type graphique ou textuel du logiciel Unity Pro permettent la programmation des plates-formes d'automatisme Modicon M340, Modicon M580, Modicon M580 Safety, Modicon Momentum, Premium et Quantum.

Les 3 langages graphiques sont :
- Langage à contacts (LD),
- Function Block Diagram (FBD),
- Langage diagramme fonctionnel en séquence (SFC) ou Grafcet.

Les 2 langages textuels sont :
- Littéral structuré (ST),
- Liste d'instructions (IL).

Pour ces 5 langages, l'utilisation du jeu d'instructions de base conforme à la norme IEC 61131-3 permet de créer des applications portables d'une plate-forme sur une autre. De plus, le logiciel Unity Pro apporte des extensions à ce jeu d'instructions de base. Ces extensions spécifiques aux plates-formes d'automatisme Modicon M340, Modicon M580, Modicon M580 Safety, Modicon Momentum, Premium et Quantum autorisent le développement d'applications plus complexes et permettent ainsi de tirer profit des spécificités de chacune de ces plates-formes.

Langage LL984

Le langage LL984 (Ladder Logic 984) permet la migration des anciennes gammes Modicon. Il est utilisé pour la programmation des plates-formes d'automatisme Modicon M580, Modicon M580 Safety, Modicon M340, Momentum et Quantum.

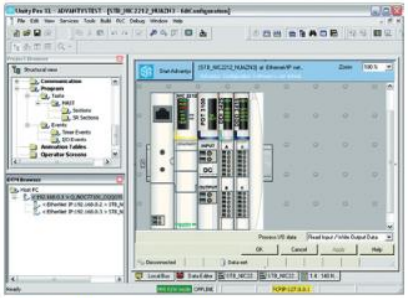
❑ INTEGRATION MATERIEL TIERS

Fonction FDT/DTM

Unity Pro facilite l'intégration d'architectures de bus de terrain dans les systèmes de contrôle d'ingénierie à l'aide de la technologie FDT/DTM :
- FDT *(Field Device Tool)* est le conteneur qui permet d'accueillir les DTM des équipements.
- DTM *(Device Type Manager)* est l'outil de configuration d'un équipement qui intègre ses propres interfaces graphiques. Il regroupe les propriétés qui définissent cet équipement.

Outre la norme FDT/DTM, Unity Pro utilise des informations spécifiques du DTM maître créé pour le module Profibus Remote Master (PRM), le module HART, le module de pesage Premium ISPY101, l'intégration des variateurs de vitesse Altivar Process et les modules réseau Modbus/TCP et EtherNet/IP BMXNOC0401 et BMENOC03●1.

L'utilisation du Master DTM permet à Unity Pro d'effectuer les actions suivantes :
- Gérer la scrutation des E/S de l'automate,
- Créer les variables application grâce à la description des objets process, disponibles à partir des équipements DTM connectés,
- Gérer la synchronisation avec la configuration de l'automate,
- Créer un DTM générique à partir des fichiers de description (GSD ou EDS).

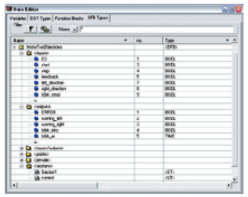


Editeur DTM (îlot Modicon STB)

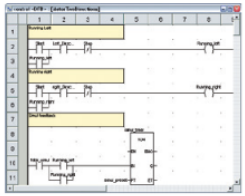
Catalogue matériel DTM

II.7: Présentation des suites logicielles

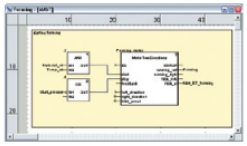
Unity Pro



Conception



Création du code



Utilisation dans le programme

❑ BLOCS FONCTIONS DFB

Blocs fonctions utilisateur DFBs

Le logiciel Unity Pro offre à l'utilisateur la possibilité de créer ses propres blocs fonctions selon les spécificités de ses applications pour les plates-formes Modicon M340, Modicon M580, Modicon M580 Safety, Modicon Momentum, Premium et Quantum.

Une fois créés et mis en bibliothèque, ces blocs fonctions utilisateur pourront être réutilisés avec la même facilité que les blocs fonctions élémentaires EFBs.

Ces blocs fonctions utilisateur permettent de structurer une application. Ils seront utilisés dès qu'une séquence de programme se trouve répétée à plusieurs reprises dans l'application ou pour figer une programmation standard. Ils peuvent être protégés en lecture seule ou en lecture/écriture. Ils peuvent être distribués vers les autres applications Unity Pro.

L'utilisation d'un bloc fonction DFB dans une ou plusieurs applications permet :

- de simplifier la conception et la saisie des programmes,
- d'accroître la lisibilité et la compréhension du programme,
- de faciliter sa mise au point (toutes les variables manipulées par le bloc fonction DFB sont identifiées grâce à l'éditeur de données),
- d'utiliser des variables privées spécifiques aux DFB, donc indépendantes de l'application.

La mise en œuvre d'un bloc fonction DFBs s'effectue en différentes phases :

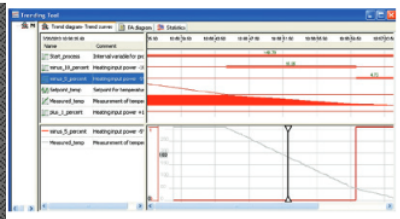
- La conception du DFBs se composant d'un nom, d'un ensemble de paramètres (entrées, sorties, variables internes publiques et privées) et d'un commentaire via l'éditeur de données,
- La création du code dans une ou plusieurs sections de programme avec, selon les besoins, le choix des langages : littéral structuré, liste d'instructions, à contacts ou blocs fonctionnels (ST, IL, LD ou FBD),
- Son rangement éventuel avec un numéro de version associée dans une bibliothèque,
- La création d'une instance DFBs dans l'éditeur de données ou lors de l'appel de la fonction dans l'éditeur de programme,
- L'utilisation de cette instance dans le programme de façon identique à un bloc fonction élémentaire EFB (la création de l'instance peut se faire à partir du programme).

❑ SIMULATION - DIAGNOSTIC

Panneau de commande simulateur

Simulateur d'automate

Le simulateur intégré au logiciel Unity Pro permet, à partir d'un terminal PC, de tester le programme application des plates-formes d'automatisme Modicon M340, M580, M580S, Momentum, Premium ou Quantum, sans faire appel à une connexion au processeur. Les fonctions offertes par les outils de mise au point sont disponibles pour la mise au point des tâches maître, rapide et auxiliaires.
Le simulateur ne gérant pas les entrées/sorties de l'automate, l'utilisation des tables d'animation permet de simuler par forçage à 1 ou à 0 l'état des entrées.
Le simulateur peut être connecté à des applications tierces via un serveur OPC avec logiciel OFS *(OPC Factory Server)*.



Panneau de commande de l'outil d'analyse des tendances

Outils d'analyse des tendances

L'outil d'analyse des tendances permet un contrôle simple des variables en détectant les problèmes d'exploitation ou en améliorant les performances du process. Vous pouvez sélectionner toutes les variables de votre application et lancer une acquisition, effectuer des enregistrements et analyser les enregistrements à l'aide d'outils intégrés ou d'Excel. Jusqu'à 16 variables peuvent être scrutées au temps de cycle MAST de l'automate.

II.7: Présentation des suites logicielles

Unity Pro

Cybersécurité

Schneider Electric a toujours veillé à la sécurité de ses système. Des conseils sur la sécurité sont mis à la disposition de nos clients afin d'assurer que leurs systèmes sont protégés des attaques potentielles.

Avec les plates-formes d'automatisme Modicon M340, M580, M580S, Premium, Momentum et Quantum :

- Protection contre les modifications de programmation réalisées à distance via un mot de passe,
- Option pour activer ou désactiver les services HTTP ou FTP.

Avec la plate-forme d'automatisme Modicon M580 :

- Historique des événements de sécurité dans la base de données SYSLOG,
- Gestion des services Ethernet étendus (DHCP, etc.) attribuables à chaque utilisateur dans la liste de contrôle d'accès,
- Sécurisation renforcée de la communication IPSec entre Unity Pro ou SCADA et l'automate.

❑ **Cybersécurité**

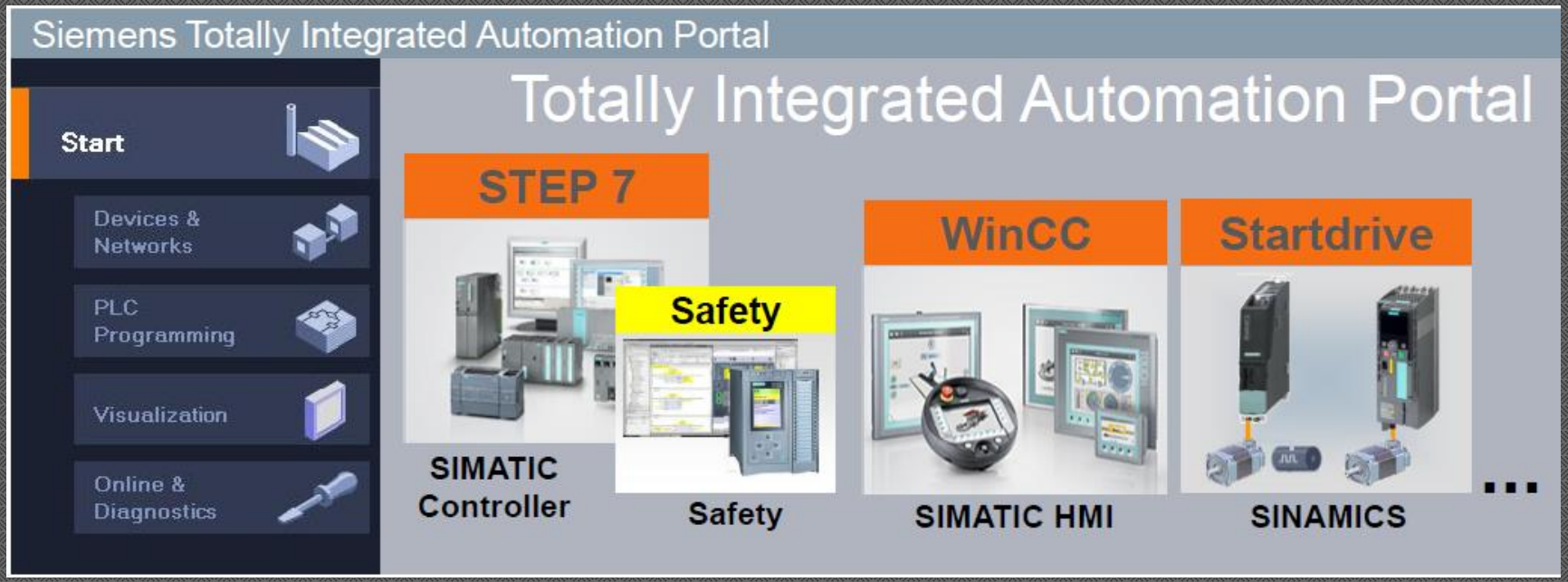
Voir tuto plus complet site :.....

II.7: Présentation des suites logicielles

❑ TIA PORTAL

SIEMENS

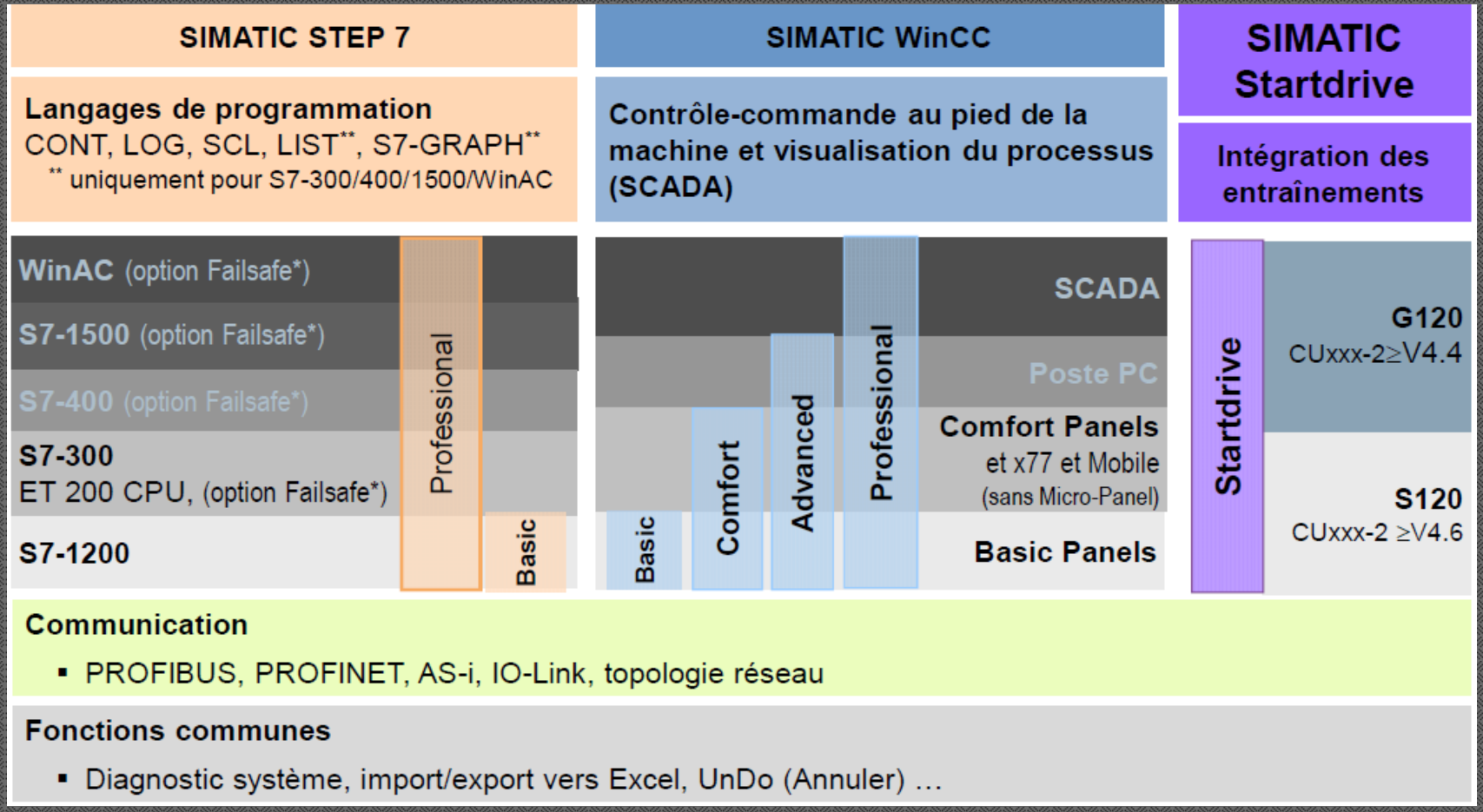
Totally Integrated Automation Portal est un **environnement d'ingénierie** qui est le cadre d'une ingénierie cohérente pour…
· …La programmation des systèmes automatisés
· …La visualisation des processus



II.7: Présentation des suites logicielles

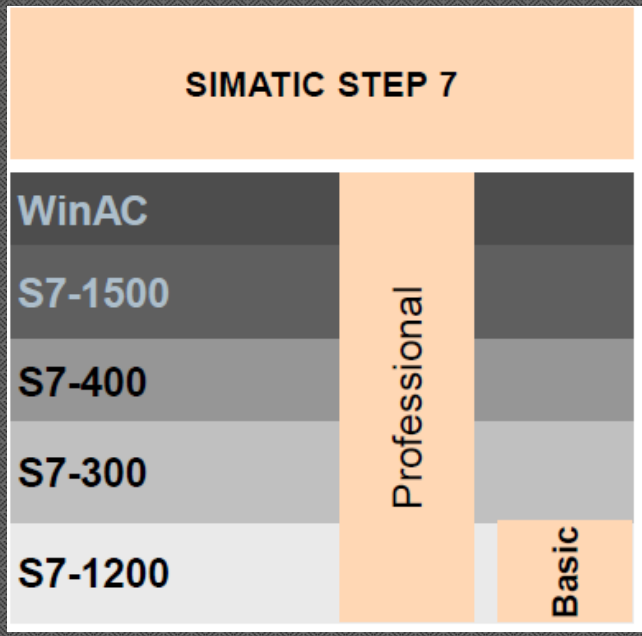
Totally Integrated Automation PORTAL

TIA Portal : Gamme de produits et fonctionnalités



II.7: Présentation des suites logicielles

TIA Portal: STEP 7 Gamme de produits



STEP 7 Basic pour S7-1200
- LAD, FBD, SCL

STEP 7 Professional
pour S7-1200, S7-1500, S7-300, S7-400, WinAC
- LAD, FBD, SCL
- STL, GRAPH pour S7-1500 / 300 / 400 / WinAC

WinCC Basic is always part of STEP 7

Option packages
- STEP 7 Safety
- PID Professional pour ET200CPU, S7-300 / 400 / WinAC
- STEP 7 Easy Motion pour ET200CPU / S7-300 / 400 / WinAC

II.7: Présentation des suites logicielles

TIA Portal : Gamme de progiciels WinCC

SIMATIC WinCC pour un usage au pied de la machine

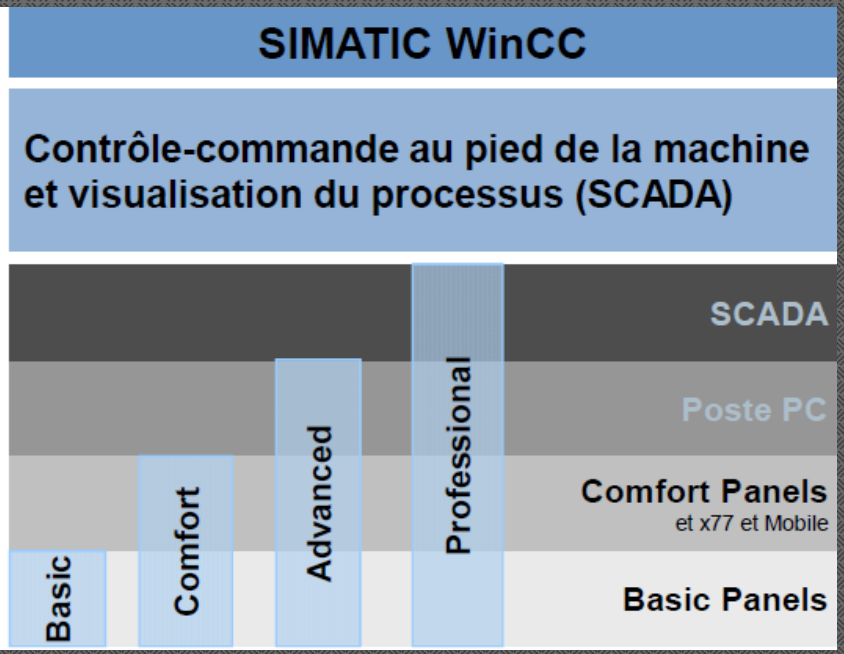
- Fonctionnalités comme WinCC flexible 2008 SP2

Fonctionnalités SIMATIC WinCC SCADA

- Fonctionnalités de base comme WinCC V7.0 SP3, diverses options RT (exécutif)
- Projets de taille moyenne : Capacité jusqu'à 64 k de Powertags (variables de process)

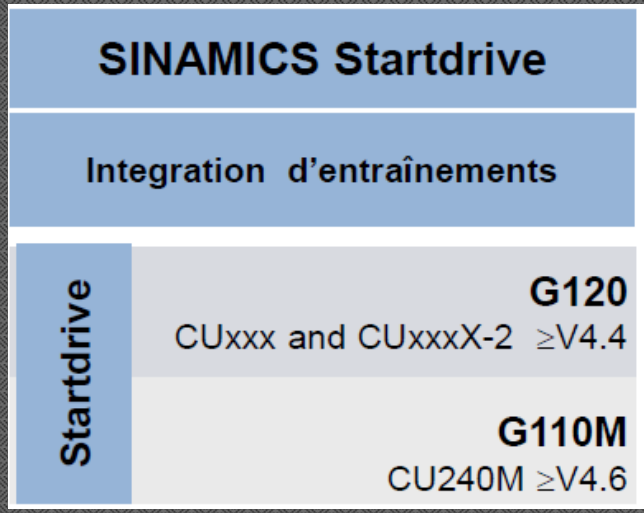
Fonctionnalités étendues

- F(x)-Control, OPC UA Client Serveur, pilotes tiers



II.7: Présentation des suites logicielles

TIA Portal: Gamme de produits Startdrive



Fonctions:

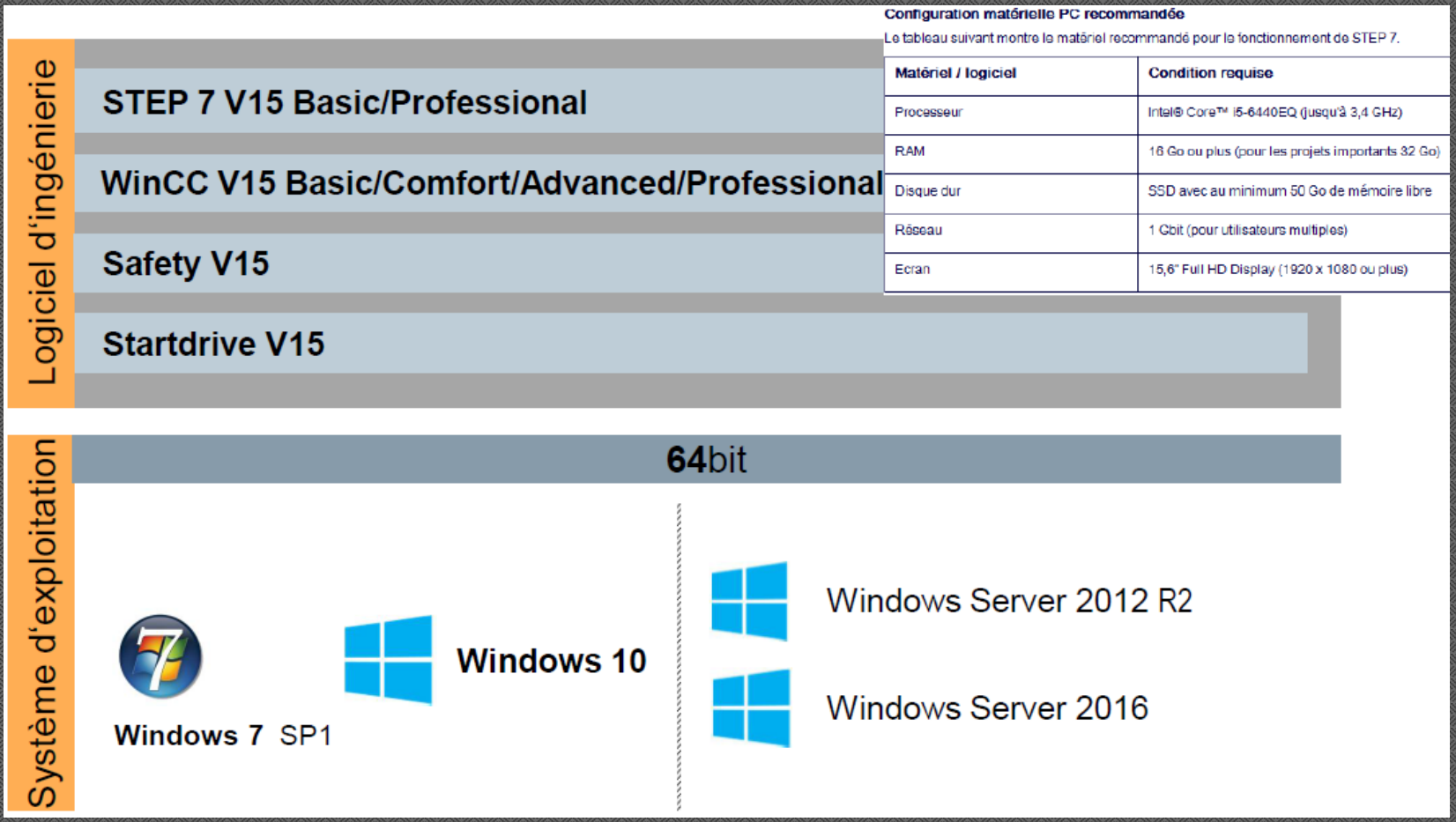
- Paramétrage des entraînements
- Mise en service
- Test

Gammes :

- SINAMICS G120
- SINAMICS G110M
- SINAMICS S120

II.7: Présentation des suites logicielles

TIA Portal : Systèmes d'exploitation pour PC/PG



Configuration matérielle PC recommandée

Le tableau suivant montre le matériel recommandé pour le fonctionnement de STEP 7.

Matériel / logiciel	Condition requise
Processeur	Intel® Core™ i5-6440EQ (jusqu'à 3,4 GHz)
RAM	16 Go ou plus (pour les projets importants 32 Go)
Disque dur	SSD avec au minimum 50 Go de mémoire libre
Réseau	1 Gbit (pour utilisateurs multiples)
Ecran	15,6" Full HD Display (1920 x 1080 ou plus)

PARTIE III:
Langage & programmation

142

III.1: PRÉSENTATION DES LANGAGES NORMALISÉS CEI61131
III.2: TYPE DE FORMAT DE DONNÉES
III.3: DÉCLARATION DES VARIABLES ET PLAN D'ADRESSAGE
III.4: OPÉRATIONS BINAIRES ET COMBINATOIRES
III.5: OPÉRATIONS NUMÉRIQUES
III.6: TEMPORISATIONS
III.7: COMPTEURS-DÉCOMPTEURS
III.8: STRUCTURES DES PROGRAMMES
III.9: PROGRAMMER UN GRAFCET EN LANGAGES NORMALISÉS CEI61131

La partie de ce cours sera orientée « SIEMENS »

III.1: Présentation des langages normalisés CEI61131

La CEI 61131-3, intitulée Automates programmables - Partie 3 : Langages de programmation, est une norme industrielle de la Commission électrotechnique internationale (CEI) définissant **cinq langages de programmation à utiliser pour les automates programmables**.
Elle a été publiée la première fois en 1993, et une seconde édition a vu le jour en 2003.

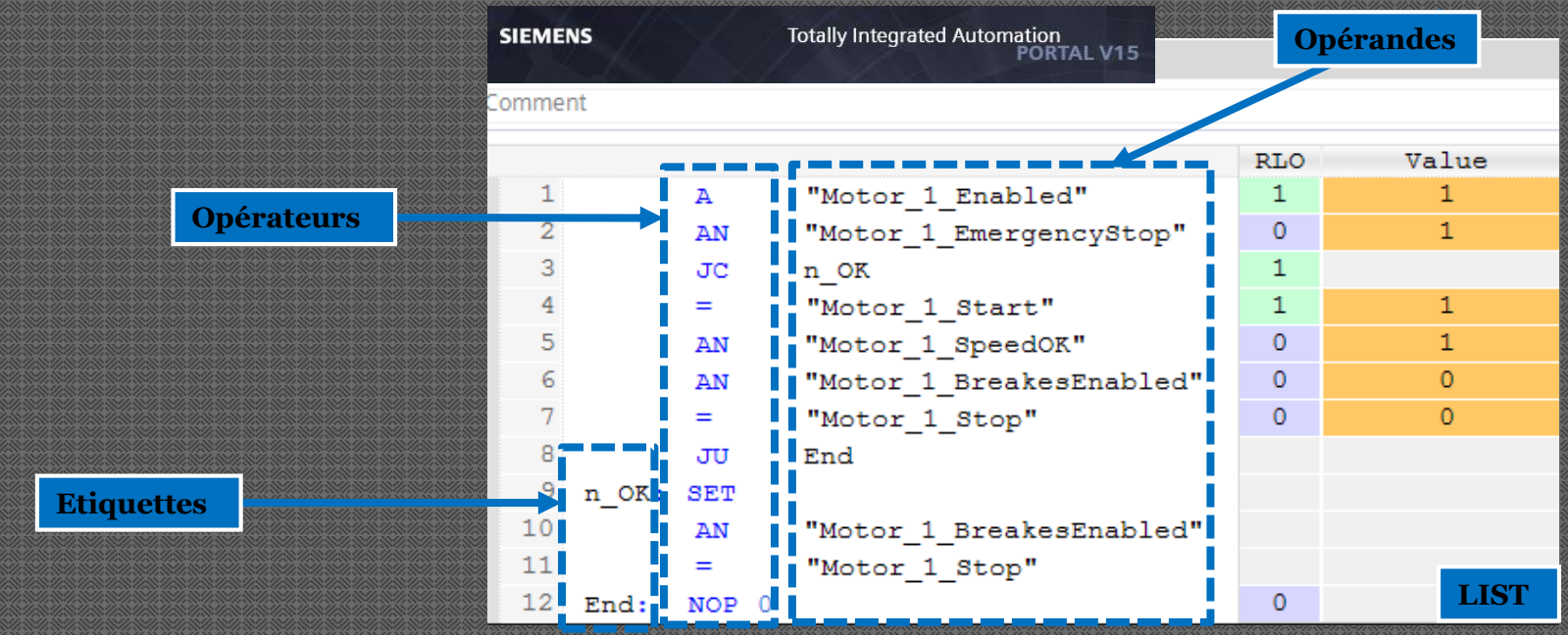
- **Les langages littéraux :**
  - IL LISTE D'INSTRUCTIONS - pseudo-assembleur
  - ST TEXTE STRUCTURE - langage littéral structuré

- **Les langages graphiques :**
  - LD LADDER - langage à contact
  - FBD LOGIGRAMME - langage à blocs fonctionnels
  - **SFC** SEQUENTIAL FUNCTION CHART - grafcet

III.1: Présentation des langages normalisés CEI61131

❑ IL: liste d 'instructions

❖ Un programme IL est une liste d'instructions.
❖ Chaque instruction doit commencer par une nouvelle ligne, et doit contenir un opérateur, un ou plusieurs opérandes, séparés par des guillemets ("....").
❖Une étiquette suivie de deux points (:) peut précéder l'instruction.
❖Des lignes vides peuvent être insérées entre des instructions.
❖Un commentaire peut être posé sur une ligne sans instruction.



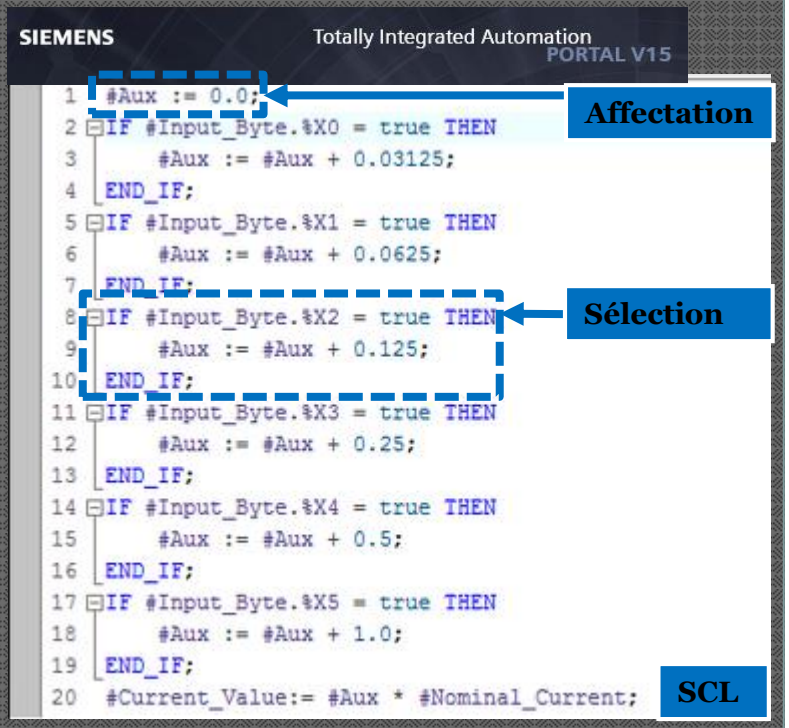
III.1: Présentation des langages normalisés CEI61131

❑ ST: Langage littéral structuré

❖ Un programme ST est une suite d'énoncés.
❖ Chaque énoncé est terminé par un point-virgule (« ; »).
❖ Les noms utilisés dans le code source (identificateurs de variables, constantes, mots clés du langage…) sont délimités par des séparateurs passifs ou des séparateurs actifs, qui ont un rôle d'opérateur.
❖ Des commentaires peuvent être librement insérés dans la programmation.

❑ Le langage littéral structuré ST utilise :

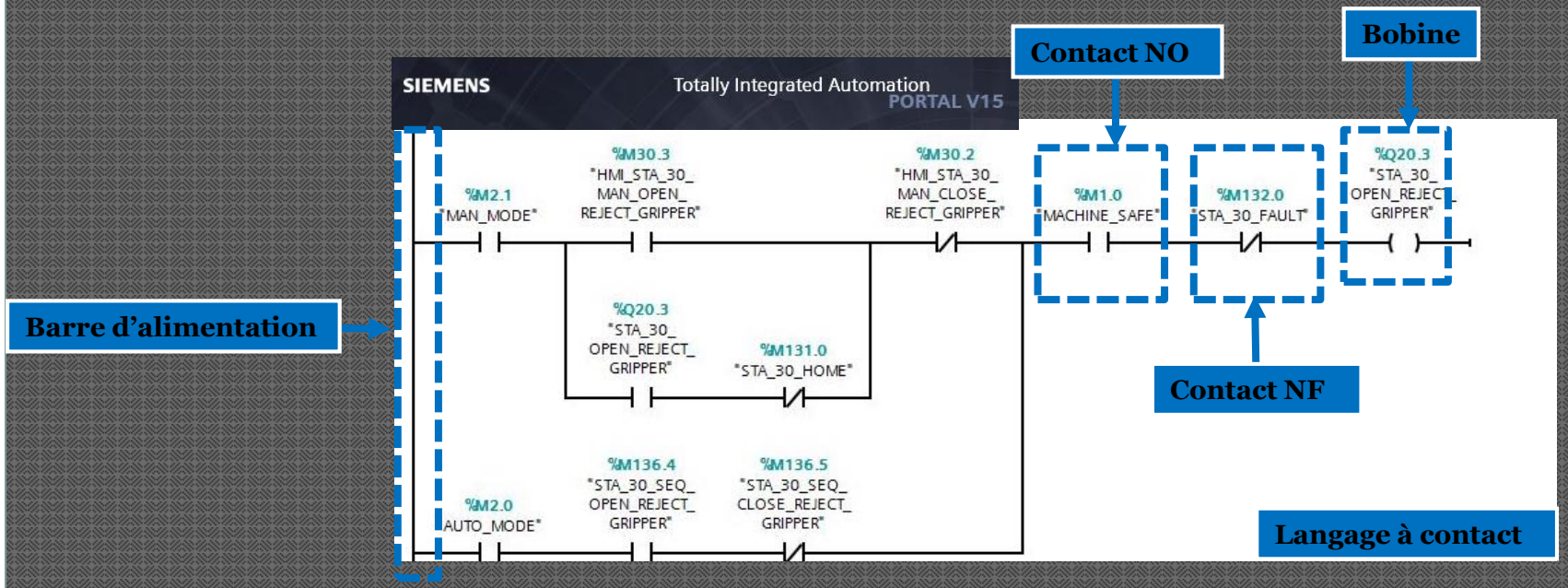
- ❖ **des expressions (E<F) AND NOT C;**
- ❖ **des énoncés :**
  - ➢ **les énoncés d 'affectation: C:=C+1;**
  - ➢ **les énoncés de sélection IF…THEN…END IF;**
  - ➢ **les énoncés d 'itération FOR…TO…. DO… END FOR; Etc.**



III.1: Présentation des langages normalisés CEI61131

❑ LD: LADDER ou langage à contacts

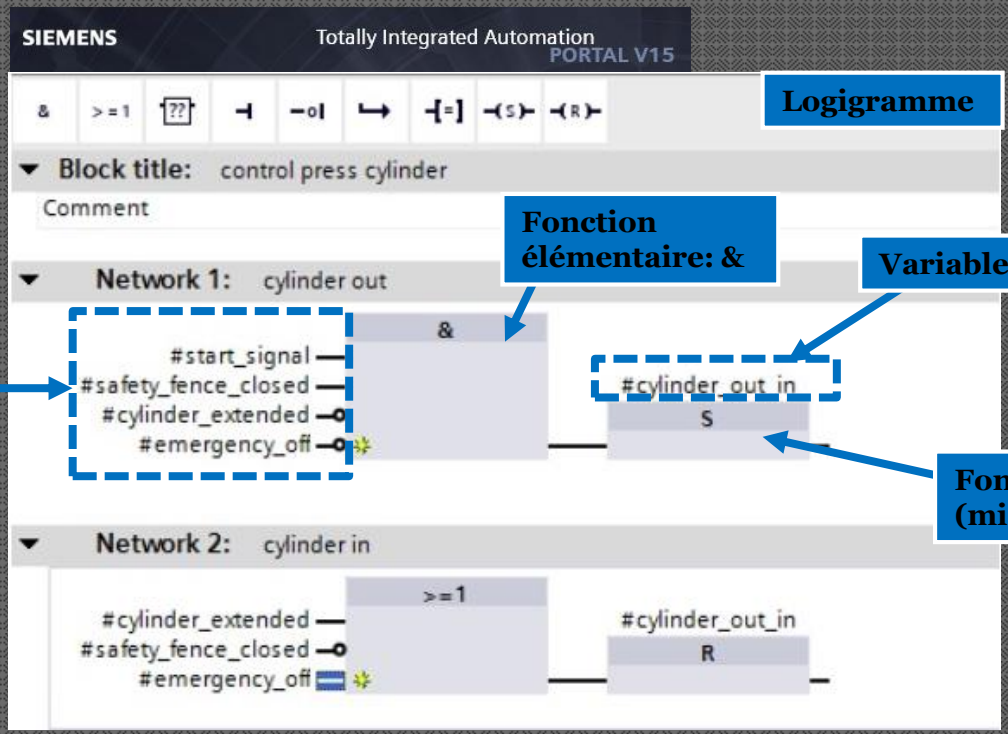
❖ Le langage LD (ladder) est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des bobines (en sortie).
❖ Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts.
❖ Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation.



III.1: Présentation des langages normalisés CEI61131

❑ FBD: LOGIGRAMME ou Langage à blocs fonctionnels

- ❖ Le diagramme FBD décrit une fonction entre des variables d'entrée et des variables de sortie.
- ❖ Une fonction est décrite comme un réseau de fonctions élémentaires.
- ❖ Les variables d'entrée et de sortie sont connectées aux boîtes fonctions par des liaisons.
- ❖ Une sortie d'une boîte peut être connectée sur une entrée d'une autre boîte.

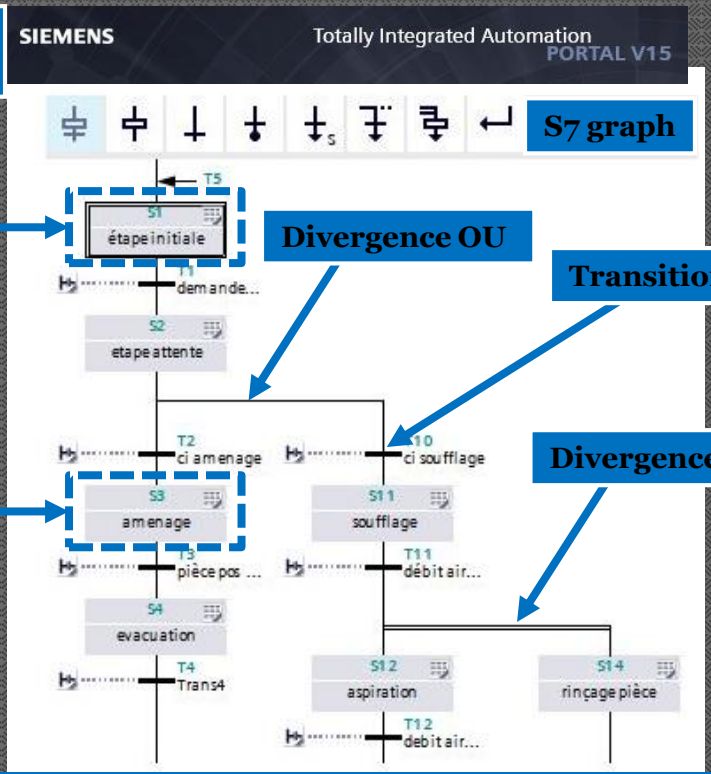


III.1: Présentation des langages normalisés CEI61131

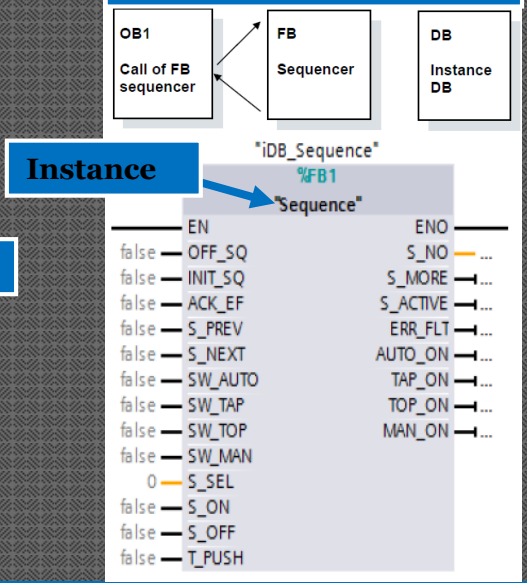
❑ SFC: Sequential Function Chart ou GRAFCET

❖ Un programme SFC est un réseau graphique d'étapes et de transitions, reliées par des liaisons orientées.
❖ Les liens de connexion multiples sont représentés par des divergences et des convergences.
❖Les principales règles graphiques sont :
- ➢ une étape ne peut pas être suivie d'une autre étape
- ➢ une transition ne peut pas être suivie d'une autre transition

Mêmes règles que la norme GRAFCET



Un programme S7 Graph est toujours instancié.



III.1: Présentation des langages normalisés CEI61131

- **La norme CEI 61131-3 répond à une attente des utilisateurs concernant les langages de programmation des API :**
  - **Harmonisation des vocabulaires utilisés**
  - **Notions et concepts de base s'appuyant sur une norme**
  - **Syntaxe et sémantique des langages les plus indépendants possibles d'un constructeur d'API donné**
  - **Facilité de mise en œuvre de principes tels que structuration et modularité des programmes**
  - **Possibilité de définir ses propres blocs fonctionnels « utilisateur »**

- **Chaque programmeur utilisera le langage selon ses connaissances. Néanmoins, le langage peut être parfois imposé par le client final.**
  - **Langage Ladder (électriciens)**
  - **Logigramme (électroniciens)**
  - **ST ou LIST (informaticiens)**
  - **SFC (automaticiens)**

- **Chaque langage a ses spécificités, il convient de choisir le langage le plus approprié en fonction du programme que l'on souhaite obtenir :**
  - **Opérations numériques, traitement de données : privilégier le ST**
  - **Opérations binaires ou combinatoires : logigramme ou ladder**

III.2: Type de format de données

- Un programme API comporte systématiquement des données à traiter pour les opérations de :
  - **Calculs booléens**
  - **Calculs numériques**
  - **Traitements horodatés**
  - **Traitements sur tableaux**
  - **Traitements sur chaîne de caractères**
  - **Etc.**

- Chez SIEMENS comme chez tous les constructeurs on dissocie 2 types de données:
  - **Données élémentaires**
  - **Données Complexes**

- Chaque donnée sera associée à une variable ce qui impose :
  - **Une taille normalisée de la donnée**
  - **Les opérations possibles liées à cette donnée**
  - **Une syntaxe d'écriture**

- Exemple d'une donnée de type binaire:
  - **Opération : calcul booléen**
  - **Type : donnée élémentaire**
  - **Taille : 1 bit**

Entrée I, test booléen, donnée élémentaire, 1 bit

III.2: Type de format de données

❑ Type des données : exemple chez SIEMENS

SIEMENS Totally Integrated Automation PORTAL V15

	Type	Type de données
Type de données élémentaires	Binaire	BOOL
	Bit	BYTE; WORD; DWORD; LWORD
	Integer	SINT; USINT; INT; UNIT; DINT; UDINT; LINT ULINT
	Nombre à virgule flottantes	REAL; LREAL
	Timers	S5TIME; TIME; LTIME
	Date, Time-of-day	DATE; TIME_OF_DAY; LTIME_OF_DAY; LDT(DATE_AND_LTIME);
	Caractères	CHAR; WCHAR
Type de données complexes	Date, Time-of-day	DT(DATE_AND_TIME); DTL;
	Chaine de caractères	STRING; WSTRING
	Tableau	ARRAY […] of <Datatype>
	Structure	STRUCT
	Type de données utilisateurs	PLC-data type (User Defined Data Type)

III.2: Type de format de données

❑ Taille des données : exemple chez SIEMENS

SIEMENS Totally Integrated Automation PORTAL V15

	Description	Taille(Bit)	S7-1200	S7-1500	Exemple
Types de données binaire	BOOL BYTE WORD DWORD	1 8 16 32	✓	✓	TRUE B#16#F5 W#16#F0F0 DW#16#F0F0FF0F
	LWORD	64	✗	✓	LW#16#5F52DE8B
Types de données numériques	SINT USINT	8	✓	✓	50 20
	INT UINT	16			-23 64530
	DINT UDINT REAL	32			DINT# -2133548520 UDINT#435676 1.0
	LREAL	64			LREAL#-1.0e-5
	LINT ULINT	64	✗	✓	LINT#1543258759 ULINT#154316159

III.2: Type de format de données

❑ Taille des données : exemple chez SIEMENS

SIEMENS Totally Integrated Automation PORTAL V15

Type de données	Longueur (Bits)	S7-1200	S7-1500	Exemple
DT (DATE_AND_TIME)	64	✗	✓	DT#2008-10-25-08:12:34.567
DTL	96	✓	✓	DTL#1976-12-16-20:30:20.250
STRING	8 * (Nombre de caractères+2)	✓	✓	´C'est une chaine´ max. 254 caractères en format ASCII
WSTRING (Wide Character String)	16 * (Nombre de caractères+2)	✓	✓	WSTRING#´STRING dans le format UNICODE Supérieur à 16382 caractères dans le format unicode

III.2: Type de format de données

❑ Taille des données : exemple chez SIEMENS

SIEMENS Totally Integrated Automation PORTAL V15

Type de données	Longueur (Bits)	S7-1200	S7-1500	Exemple	
ARRAY	Définie par l'utilisateur	✓	✓	Valeurs de mesures : ARRAY[1..20] of INT;	
STRUCT	Définie par l'utilisateur	✓	✓	Moteur: STRUCT Vitesse : REAL; Consigne : UINT; Erreur : BOOL; Activation : BOOL; END_STRUCT	
WSTRING (Wide Character String)	Définie par l'utilisateur	✓	✓	**Définie** (dans le dossier "Type de données utils.") PLC-DT1 : STRUCT vitesse : REAL; Activation: BOOL; Defaut: BOOL; END_STRUCT	**Utilisation** (dans les blocs de données et interface de blocs) Moteur1 : PLC-DT1; Moteur2 : PLC-DT1; Moteur3 : PLC-DT1; :

III.2: Type de format de données

❑ Taille des données : exemple chez SIEMENS

SIEMENS Totally Integrated Automation PORTAL V15

Type de données	Longueur (Bits)	S7-1200	S7-1500	Exemple
ARRAY	Définie par l'utilisateur	✓	✓	Valeurs de mesures : ARRAY[1..20] of INT;
STRUCT	Définie par l'utilisateur	✓	✓	Moteur: STRUCT Vitesse : REAL; Consigne : UINT; Erreur : BOOL; Activation : BOOL; END_STRUCT
WSTRING (Wide Character String)	Définie par l'utilisateur	✓	✓	**Définie** (dans le dossier "Type de données utils.") PLC-DT1 : STRUCT vitesse : REAL; Activation: BOOL; Defaut: BOOL; END_STRUCT **Utilisation** (dans les blocs de données et interface de blocs) Moteur1 : PLC-DT1; Moteur2 : PLC-DT1; Moteur3 : PLC-DT1; :

III.3: Déclaration des variables et plan d'adressage

- **Chaque constructeur propose son plan d'adressage (mémoire CPU) et la syntaxe associée :**
  - **Allocation de la zone mémoire**
  - **Syntaxe : Lettre de la zone, taille et adresse**

- **Lors d'une déclaration de la variable, 2 étapes :**
  - **Création**
  - **Définition du nom et du type de données**

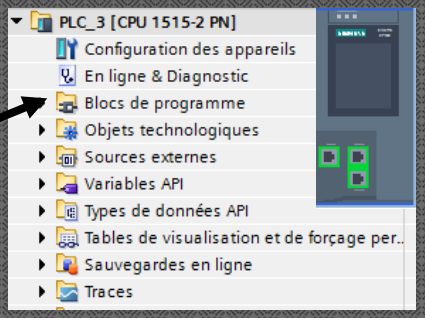
- **Une variable peut être :**
  - **Globale**
  - **Locale**

III.3: Déclaration des variables et plan d'adressage

❑ Variables globales

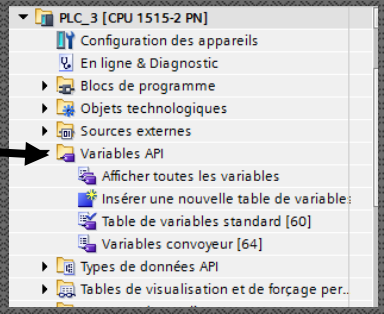
SIEMENS Totally Integrated Automation PORTAL V15

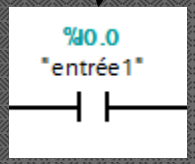
Domaine de validité	❖ Valables dans l'ensemble de la CPU et utilisables dans tous les blocs de programmation ❖ Le nom symbolique de ces variables doit être univoque.
Opérandes	❖ Entrées ❖ Sorties ❖ Mémentos ❖ Variables dans un bloc de données ❖ Temporisations/compteurs
Lieu de déclaration	❖ Table de variables API ❖ Blocs de données globaux
Représentation	❖ Représentation entre des guillemets Exemple: "entrée 1"

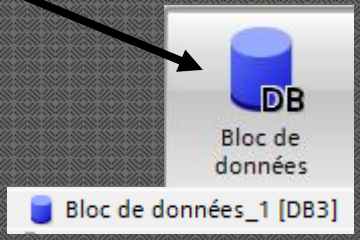


Zone processus images E/S : MIE/MIS

Zone mémoire interne CPU







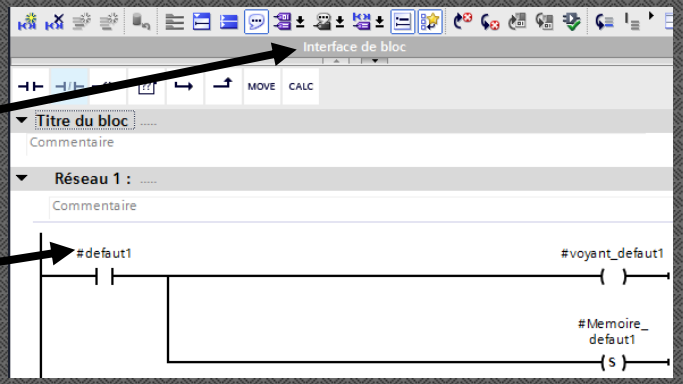
III.3: Déclaration des variables et plan d'adressage

❑ Variables locales

SIEMENS Totally Integrated Automation PORTAL V15

Domaine de validité	❖ Valables uniquement dans le bloc dans lequel elles ont été déclarées ❖ Le nom doit être univoque uniquement dans le bloc.
Opérandes	❖ Variables IN ❖ Variables OUT ❖ Variables IN/OUT ❖ Variables TEMPORAIRES ❖ Variables STATIQUES (uniquement dans FB)
Lieu de déclaration	❖ Interface du bloc
Représentation	❖ Représentation avec un # Exemple: # defaut1

	Nom	Type de données
1	Input	
2	defaut1	Bool
3	Output	
4	voyant_defaut1	Bool
5	InOut	
6	Static	
7	Memoire_defaut1	Bool
8	Temp	
9	Constant	

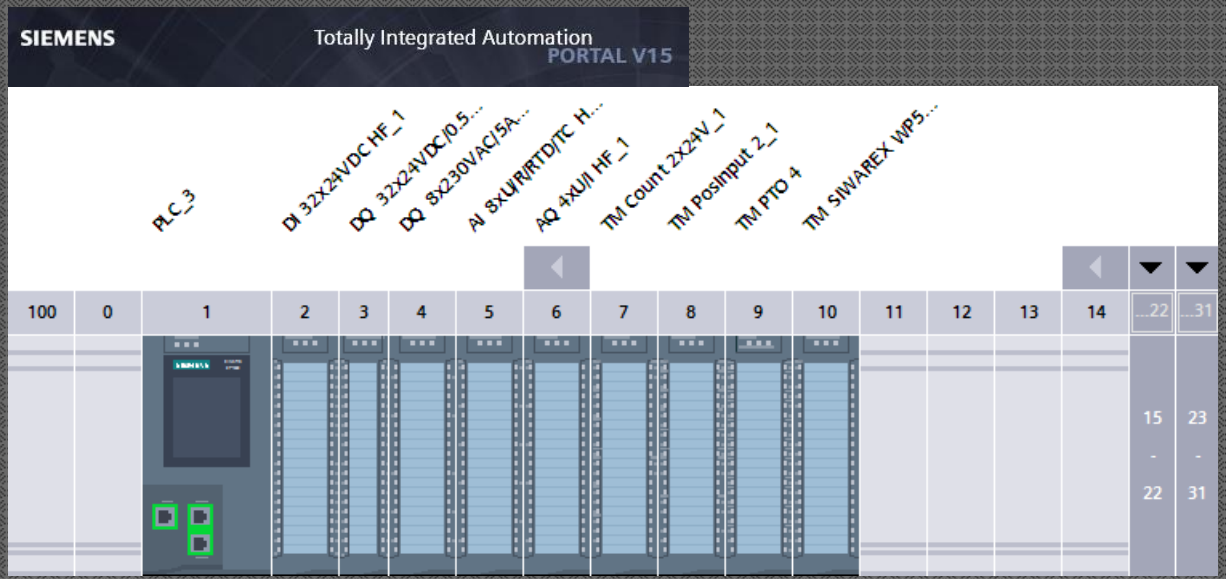


Dièse pour les  comme moi

Hashtag pour le jeunes comme vous

III.3: Déclaration des variables et plan d'adressage

❑ Plan d'adressage des E/S automate

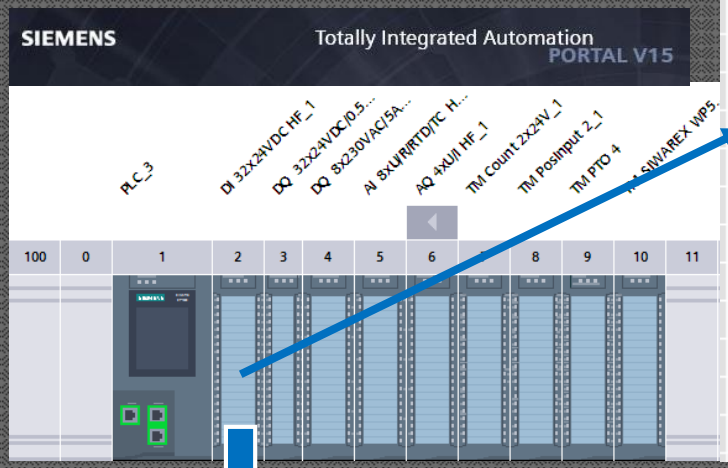


Description du modèle :

- **1: CPU 1515-2PN**
- **2: Module 32DI (6ES7 521-1BL00-0AB0)**
- **3: Module 32DQ (6ES7 522-1BL10-0AA0)**
- **4: Module 8DQ à relais (6ES7 522-5HF00-0AB0)**
- **5: Module 8AI (6ES7 531-7PF00-0AB0)**
- **6: Module 4AQ (6ES7 532-5ND00-0AB0)**
- **7: Module de comptage rapide TM Count 2x24V (6ES7 550-1AA00-0AB0)**
- **8: Module de détection de position TM PosInput 2 (6ES7 551-1AB00-0AB0)**
- **9: Module de commande pour entraînements pas-à-pas TM PTO 4 (6ES7 553-1AA00-0AB0)**
- **10: Module de pesage TM SIWAREX WP521 ST (7MH4 980-1AA01)**

III.3: Déclaration des variables et plan d'adressage

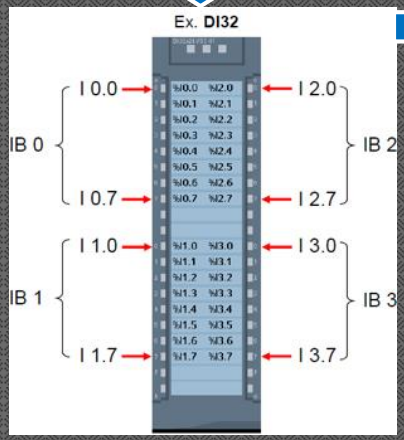
❑ Plan d'adressage des E/S automate



Adressage basé sur l'octet

Module 32 entées

PLC_3	0	1		
▸	0	1 X1		
▸	0	1 X2		
DI 32x24VDC HF_1	0	2	0...3	
DQ 32x24VDC/0.5A BA_1	0	3		0...3
DQ 8x230VAC/5A ST_1	0	4		4
AI 8xU/R/RTD/TC HF_1	0	5	4...21	
AQ 4xU/I HF_1	0	6		5...12
TM Count 2x24V_1	0	7	22...53	13...36
TM PosInput 2_1	0	8	54...85	37...60
TM PTO 4	0	9	86...157	61...100
TM SIWAREX WP521 ST_1	0	10	158...189	158...189



Module 32 entrées TOR : (0...3) //4 octets// (8x4 = 32 entrées)

I0.0...I0.7 - I1.0...I1.7 - I2.0...I0.7 - I3.0...I3.7

- **Module 32 sorties TOR : (0...3) //4 octets// (8x4 = 32 sorties) Q0.0 à Q3.7**
- **Module 8 sorties TOR : (4) //1 octet// (8x1 = 8 sorties) Q4.0 à Q4.7**
- **Module 9 entrées ANA : (4...21) //18 octets// (18/2 = 9 entrées) IW4 - IW6 - IW8 - IW10...IW20**
- **Module 4 sorties ANA : (5...12) //8 octets// (8/2= 4 sorties) QW5 - QW7 - QW9 - QW11**
- **Etc.**

III.3: Déclaration des variables et plan d'adressage

❑ Plan d'adressage

SIEMENS Totally Integrated Automation PORTAL V15

Entrées I (lecture dans la Mémoire Image d'Entrées MIE)
- ❖ **I y.x** désigne une entrée y est le numéro de voies (0 à etc.), x sa position (0 à 8)
- ❖ **IB y** désigne un octet d'entrées
- ❖ **IW y** désigne un mot d'entrées (16 bits)
- ❖ **ID y** désigne un double mots d'entrées (32 bits).

Les mêmes termes suivis de :P accèdent directement à la périphérie.

Entrées Q (lecture dans la Mémoire Image de sorties MIS)
- ❖ **Q y.x** désigne une entrée y est le numéro de voies (0 à etc.), x sa position (0 à 8)
- ❖ **QB y** désigne un octet d'entrées
- ❖ **QW y** désigne un mot d'entrées (16 bits)
- ❖ **QD y** désigne un double mots d'entrées (32 bits).

Les mêmes termes suivis de :P accèdent directement à la périphérie.

Mémentos M(lecture dans la Mémoire interne CPU)
- ❖ **M y.x** désigne une entrée y est le numéro de voies (0 à etc.), x sa position (0 à 8)
- ❖ **MB y** désigne un octet d'entrées
- ❖ **MW y** désigne un mot d'entrées (16 bits)
- ❖ **MD y** désigne un double mots d'entrées (32 bits).

Données DB (nécessite la création d'un DB global)
- ❖ **DB y.x** désigne une entrée y est le numéro de voies (0 à etc.), x sa position (0 à 8)
- ❖ **DBB y** désigne un octet d'entrées
- ❖ **DBW y** désigne un mot d'entrées (16 bits)
- ❖ **DBD y** désigne un double mots d'entrées (32 bits).

III.3: Déclaration des variables et plan d'adressage

❑ Ordonnancement de la mémoire :

SIEMENS Totally Integrated Automation PORTAL V15

BIT X,0	BIT X,1	BIT X,2	BIT X,3	BIT X,4	BIT X,5	BIT X,6	BIT X,7	BYTE X	MOT X modulo 2	MOT X modulo 2	DOUBLE MOT Modulo 4	DOUBLE MOT Modulo 4	DOUBLE MOT Modulo 4	DOUBLE MOT Modulo 4
M0.0	M0.1	M0.2	M0.3	M0.4	M0.5	M0.6	M0.7	MB0	MW0					
M1.0	M1.1	M1.2	M1.3	M1.4	M1.5	M1.6	M1.7	MB1		MW1	MD0			
M2.0	M2.1	M2.2	M2.3	M2.4	M2.5	M2.6	M2.7	MB2	MW2			MD1		
M3.0	M3.1	M3.2	M3.3	M3.4	M3.5	M3.6	M3.7	MB3		MW3			MD2	
M4.0	M4.1	M4.2	M4.3	M4.4	M4.5	M4.6	M4.7	MB4	MW4		MD4			MD3
M5.0	M5.1	M5.2	M5.3	M5.4	M5.5	M5.6	M5.7	MB5		MW5		MD5		
M6.0	M6.1	M6.2	M6.3	M6.4	M6.5	M6.6	M6.7	MB6	MW6				MD6	
M7.0	M7.1	M7.2	M7.3	M7.4	M7.5	M7.6	M7.7	MB7		MW7				MD7
M8.0	M8.1	M8.2	M8.3	M8.4	M8.5	M8.6	M8.7	MB8	MW8		MD8			
M9.0	M9.1	M9.2	M9.3	M9.4	M9.5	M9.6	M9.7	MB9		MW9		MD9		
M10.0	M10.1	M10.2	M10.3	M10.4	M10.5	M10.6	M10.7	MB10	MW10				MD10	
M11.0	M11.1	M11.2	M11.3	M11.4	M11.5	M11.6	M11.7	MB11		MW11				MD11
M12.0	M12.1	M12.2	M12.3	M12.4	M12.5	M12.6	M12.7	MB12	MW12		MD12			
M13.0	M13.1	M13.2	M13.3	M13.4	M13.5	M13.6	M13.7	MB13		MW13		MD13		
M14.0	M14.1	M14.2	M14.3	M14.4	M14.5	M14.6	M14.7	MB14	MW14				MD14	
M15.0	M15.1	M15.2	M15.3	M15.4	M15.5	M15.6	M15.7	MB15		MW15				MD15
M16.0	M16.1	M16.2	M16.3	M16.4	M16.5	M16.6	M16.7	MB16	MW16		MD16			
M17.0	M17.1	M17.2	M17.3	M17.4	M17.5	M17.6	M17.7	MB17		MW17		MD17		
M18.0	M18.1	M18.2	M18.3	M18.4	M18.5	M18.6	M18.7	MB18	MW18				MD18	
M19.0	M19.1	M19.2	M19.3	M19.4	M19.5	M19.6	M19.7	MB19		MW19				MD19
M20.0	M20.1	M20.2	M20.3	M20.4	M20.5	M20.6	M20.7	MB20	MW20					
M21.0	M21.1	M21.2	M21.3	M21.4	M21.5	M21.6	M21.7	MB21		MW21				
M22.0	M22.1	M22.2	M22.3	M22.4	M22.5	M22.6	M22.7	MB22						

Exemple :

D	MD0			
W	MW0		MW2	
B	MB0	MB1	MB2	MB3
BIT	M0.7 M0.6 M0.5 M0.4 M0.3 M0.2 M0.1 M0.0	M1.7 M1.6 M1.5 M1.4 M1.3 M1.2 M1.1 M1.0	M2.7 M2.6 M2.5 M2.4 M2.3 M2.2 M2.1 M2.0	M3.7 M3.6 M3.5 M3.4 M3.3 M3.2 M3.1 M3.0
BINAIRE	1 1 1 1	1 1 1 0	1 1 0 1	1 1 0 0	1 0 1 1	1 0 1 0	1 0 0 1	1 0 0 0
HEXA	F E	D C	B A	9 8
HEXA	F E D C		B A 9 8	
HEXA	FEDCBA98			

La connaissance du plan d'adressage permet d'éviter le chevauchement
Ce type de numérotation s'applique à tout les types de variables (Q.., I.., DB..,)

III.4: Opérations binaires et combinatoires

Introduction

- Pour nombre de tâches de commande, il est nécessaire de programmer l'API en utilisant des opérations de type binaires et combinatoires. **Dans de nombreux cas, les actions doivent être déclenchées lorsqu'une certaine combinaison de conditions est vérifiée.**

- Ces opérations sont programmables dans l'ensemble des langages normalisés, mais en règle générale, les automaticiens privilégient le LADDER (langage à contact) ou le FBD (logigramme) pour répondre à ce besoin de programmation.

- Pour faciliter le travail de programmation, les API disposent d'instructions élémentaires intégrées dans des bibliothèques.

- Les opérations binaires et combinatoires font appel en général aux fonctions logiques suivantes:
  - ET
  - OU
  - OU EXCLUSIF
  - NON ET
  - NON OU
  - ETC.

III.4: Opérations binaires et combinatoires

❑ Instructions sur bits: LADDER ou FBD

SIEMENS Totally Integrated Automation PORTAL V15

Instructions en FBD

Opérations logiques sur ...	
&	Opération logique ET [Maj+F2]
>=1	Opération logique OU [Maj+F3]
x	Opération logique OU EXCLUSIF
–[=]	Affectation [Maj+F7]
–[/=]	Négation de l'affectation
–[R]	Mise à 0 sortie
–[S]	Mise à 1 sortie
SET_BF	Mise à 1 champ de bits
RESET_BF	Mise à 0 champ de bits
SR	Bascule 'mise à 1/mise à 0'
RS	Bascule 'mise à 0/mise à 1'
–|P|–	Interroger front montant d'un opérande
–|N|–	Interroger front descendant d'un opérande
–[P]–	Mise à 1 de l'opérande si front montant du signal
–[N]–	Mise à 1 de l'opérande si front descendant du signal
P_TRIG	Interroger front montant du RLO
N_TRIG	Interroger front descendant du RLO
R_TRIG	Détecter front montant du signal
F_TRIG	Détecter front descendant du signal

Instructions en LADDER

Opérations logiques sur bits	
–||–	Contact à fermeture [Maj+F2]
–|/|–	Contact à ouverture [Maj+F3]
–|NOT|–	Inverser RLO
–( )–	Affectation [Maj+F7]
–(/)–	Négation de l'affectation
–(R)	Mise à 0 sortie
–(S)	Mise à 1 sortie
SET_BF	Mise à 1 champ de bits
RESET_BF	Mise à 0 champ de bits
SR	Bascule 'mise à 1/mise à 0'
RS	Bascule 'mise à 0/mise à 1'
–|P|–	Interroger front montant d'un opérande
–|N|–	Interroger front descendant d'un opérande
–(P)–	Mise à 1 de l'opérande si front montant du signal
–(N)–	Mise à 1 de l'opérande si front descendant du signal
P_TRIG	Interroger front montant du RLO
N_TRIG	Interroger front descendant du RLO
R_TRIG	Détecter front montant du signal
F_TRIG	Détecter front descendant du signal

III.4: Opérations binaires et combinatoires

SIEMENS Totally Integrated Automation PORTAL V15

- Le langage de base de l'automate est le LIST. C'est le langage "assembleur" de l'automate.
- Chacune des instructions prend un certain temps CPU (voir transparent 95) ou voir manuel de référence de la CPU retenue.
- Ce langage utilise un bit logique (RLG). Toute nouvelle instruction logique met la valeur de l'opérande dans le bit RLG.
- Ensuite les opérations logiques sont exécutées entre l'opérande et le RLG. L'attribution (=) copie le résultat se trouvant dans RLG et l'attribue à l'opérande qui la suit.
- Chez SIEMENS le RLG se nomme VKE

Opérateurs	Modificateurs *	Opérandes	Descriptions
A	N (	booléen	AND
O	N (	booléen	OR
X	N (	booléen	XOR
NOT		booléen	Inverser le résultat logique courant
S		booléen	Forçage à TRUE
R		booléen	Forçage à FALSE
=		booléen	Affecter le résultat logique courant
)		booléen	Fin d'une opération différée

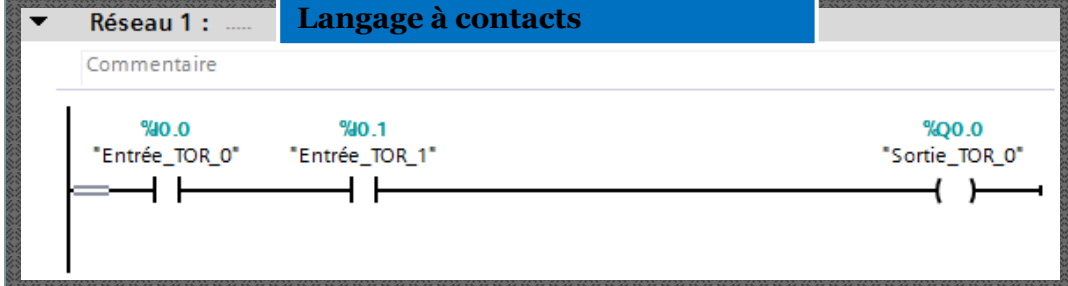
* **N : Not** **(: Opération différée**

III.4: Opérations binaires et combinatoires

❑ Fonction logique : ET

SIEMENS Totally Integrated Automation PORTAL V15

Langage à contacts

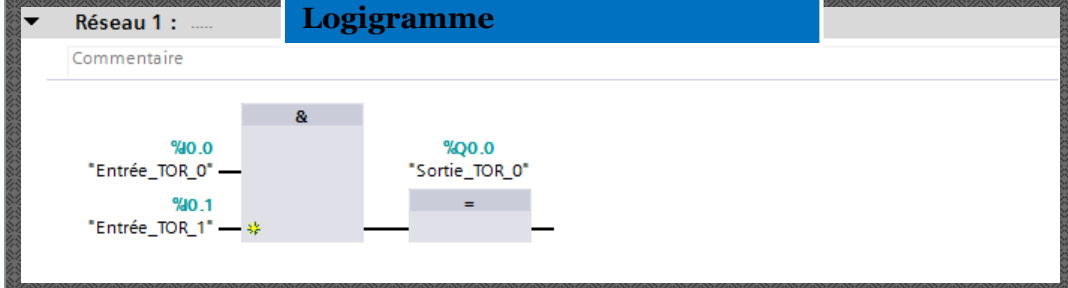


S=A.B

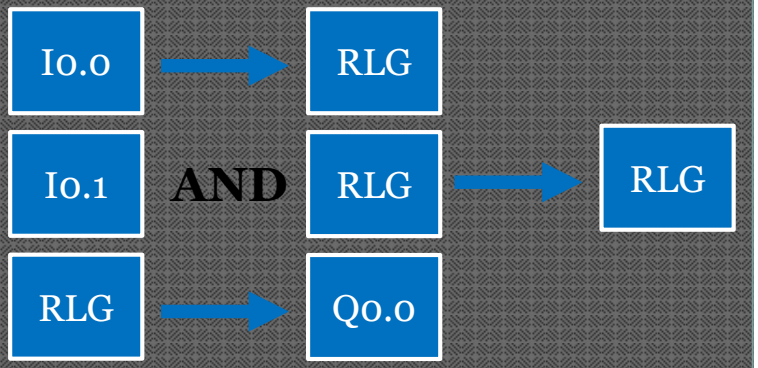
%Q0.0 = %I0.0 . %I0.1

La sortie est mise à 1 si les 2 entrées sont à 1

Logigramme



Interprétation par la CPU



Texte structuré (SCL)

```
1  //Fonction ET
2  "Sortie_TOR_0" := "Entrée_TOR_0" AND "Entrée_TOR_1";
```

Liste d'instructions (LIST)

```
1  // Fonction ET
2        A     "Entrée_TOR_0"
3        A     "Entrée_TOR_1"
4        =     "Sortie_TOR_0"
5
```

III.4: Opérations binaires et combinatoires

❑ Fonction logique : OU

SIEMENS Totally Integrated Automation PORTAL V15

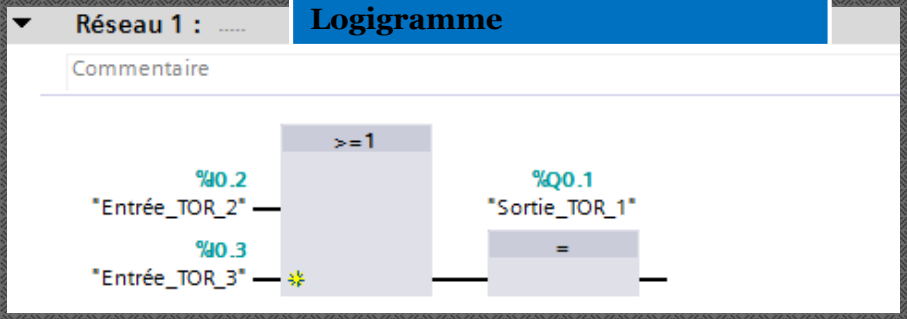
Langage à contacts

S=A+B

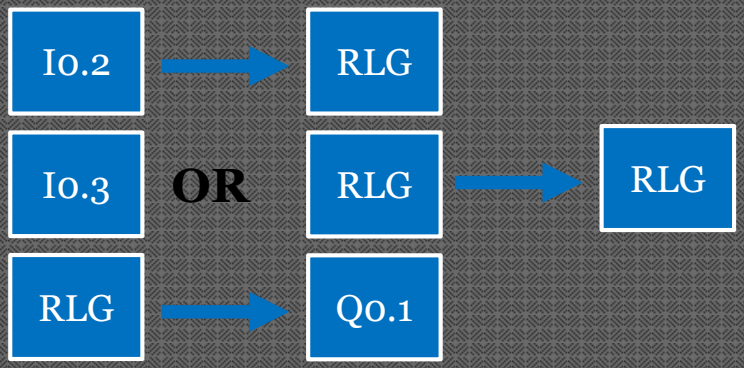
%Q0.1 = %I0.2 . %I0.3

La sortie est mise à 1 si au moins une des 2 entrées est à 1

Logigramme



Interprétation par la CPU



Texte structuré (SCL)

```
1  //Fonction OU
2  "Sortie_TOR_1" := "Entrée_TOR_2" OR "Entrée_TOR_3";
3
```

Liste d'instructions (LIST)

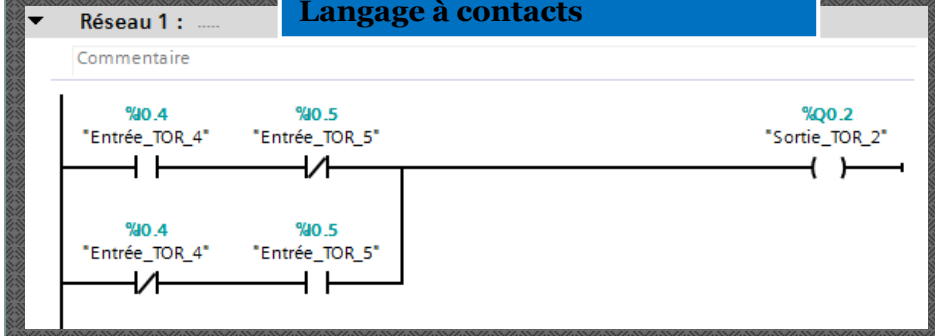
```
7   //Fonction OU
8         O     "Entrée_TOR_2"
9         O     "Entrée_TOR_3"
10        =     "Sortie_TOR_1"
11
```

III.4: Opérations binaires et combinatoires

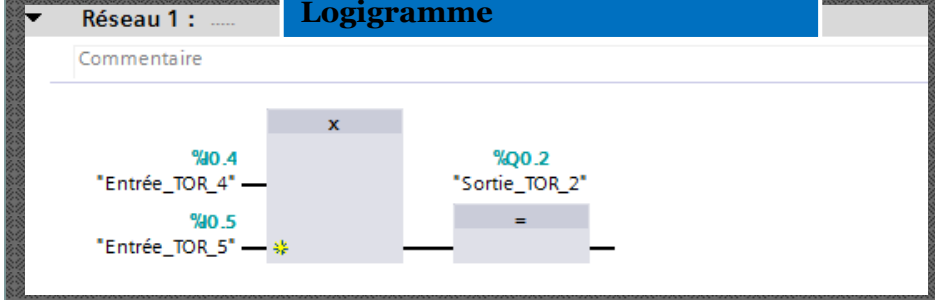
❑ Fonction logique : OU EXCLUSIF

SIEMENS Totally Integrated Automation PORTAL V15

Langage à contacts



Logigramme



Texte structuré (SCL)

```
1  //Fonction OU exclusif
2  "Sortie_TOR_2" := "Entrée_TOR_4" XOR "Entrée_TOR_5";
```

Liste d'instructions (LIST)

```
12 //Fonction OU exclusif
13       X     "Entrée_TOR_4"
14       X     "Entrée_TOR_5"
15       =     "Sortie_TOR_2"
16
```

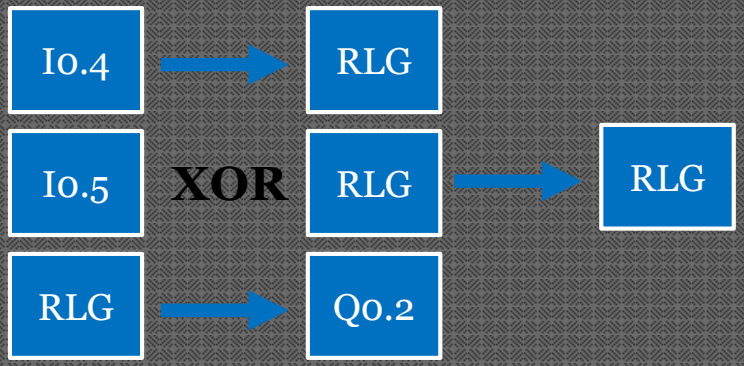
Dans le langage à contact, le OU exclusif ne peut être réalisé qu'à l'aide de contacts à ouverture et de contacts à fermeture. Par contre le OU exclusif existe dans les autres langages.

S=(A./B)+(/A.B)

%Q0.2 = (%I0.4 . /%I0.3) + (/%I0.4 . %I0.3)

La sortie est mise à 1 une entrée est à 1 et l'autre à 0

Interprétation par la CPU

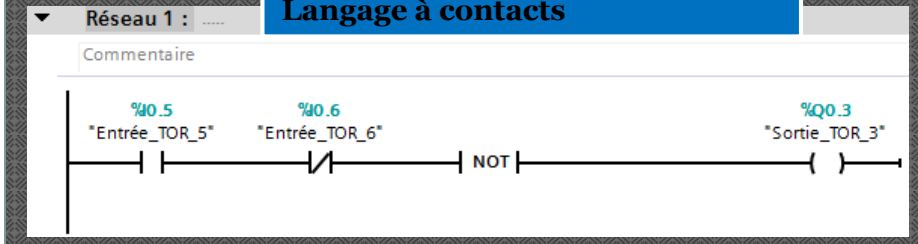


III.4: Opérations binaires et combinatoires

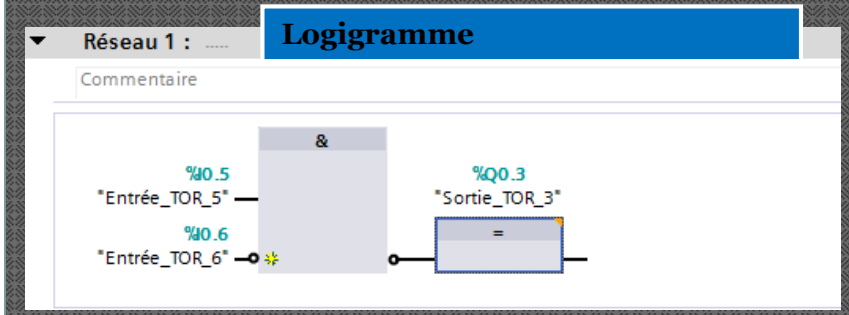
❑ Fonction logique : INVERSER

SIEMENS Totally Integrated Automation PORTAL V15

Langage à contacts



Logigramme



Texte structuré (SCL)

```
1  //Fonction NOT
2  "Sortie_TOR_3" := NOT ("Entrée_TOR_5" AND NOT "Entrée_TOR_6");
```

Liste d'instructions (LIST)

```
17  //Fonction NOT
18        A     "Entrée_TOR_5"
19        AN    "Entrée_TOR_6"
20        NOT
21        =     "Sortie_TOR_3"
```

Les opérations ET NON, OU NON et OU exclusif NON (AN, ON et XN) permettent de faire un ET, OU , OU Exclusif en inversant l'opérande qui suit l'opération (en langage à contact c'est un contact NC Normalement Connecté, en logigramme c'est un petit rond qui désigne cette inversion) .

En LIST ou en SCL, pour exécuter un NON ET, NON OU ou NON OU Exclusif, ces opérations n'existent pas en tant que tel, il est nécessaire d'inverser le résultat de l' opération logique en utilisant le mot clé NOT.

S=/(A./B) =/A+B

%Q0.3 = (%I0.5 . /%I0.6)

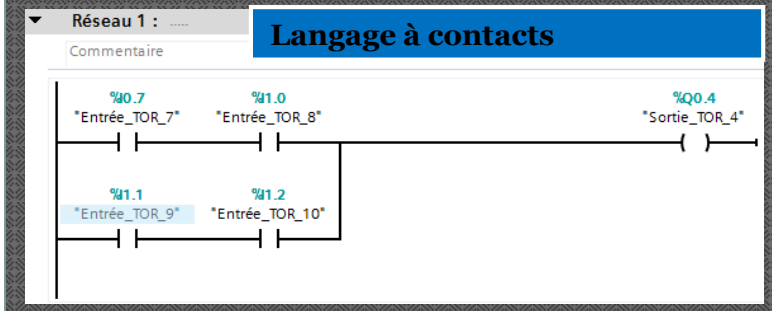
La sortie est mise à 1 une entrée est à 1 et l'autre à 0

III.4: Opérations binaires et combinatoires

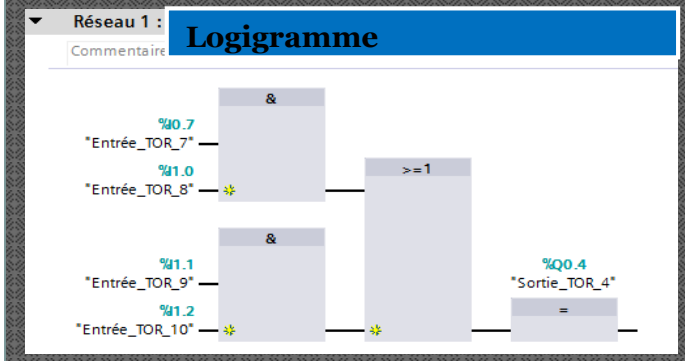
❑ Opérations combinatoires différés :

SIEMENS Totally Integrated Automation PORTAL V15

Langage à contacts



Logigramme



Il est possible de faire des combinaisons d'opérations logiques.

Pour cela, il a été prévu l'utilisation des opérations combinatoires différées, grâce à parenthèses associées aux fonctions logiques.

Exemple ci-contre:
%Q0.4 = (%I0.7 . %I1.0) + (%I1.1 . %I1.2)

Texte structuré (SCL)

```
1  //Fonction différée
2  "Sortie_TOR_4" := ("Entrée_TOR_7" AND "Entrée_TOR_8") OR ("Entrée_TOR_9" AND "Entrée_TOR_10");
```

Liste d'instructions (LIST)

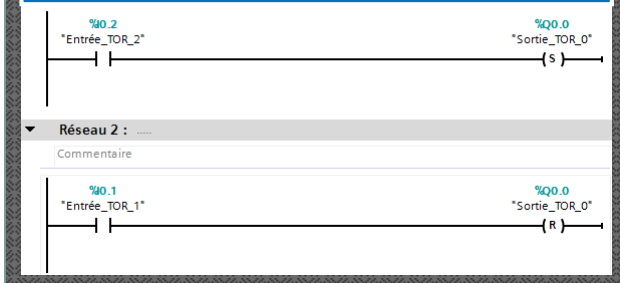
```
23  //Fonction différée
24        A     "Entrée_TOR_7"
25        A     "Entrée_TOR_8"
26        O(
27        A     "Entrée_TOR_9"
28        A     "Entrée_TOR_10"
29        )
30        =     "Sortie_TOR_4"
```

III.4: Opérations binaires et combinatoires

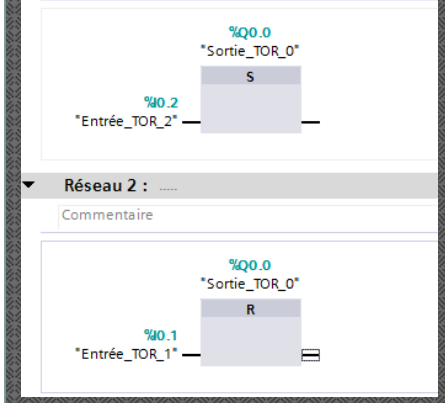
❑ Fonctions SET et RESET :

SIEMENS Totally Integrated Automation PORTAL V15

Langage à contacts



Logigramme



Texte structuré (SCL)

```
1  // Fonctions SET et RESET
2  IF "Entrée_TOR_1" THEN
3      "Sortie_TOR_0" := 0;
4  ELSIF "Entrée_TOR_2" THEN
5      "Sortie_TOR_0" := 1;
6  END_IF;
7
```

Liste d'instructions (LIST)

```
32  //Fonctions Set et Reset
33        A     "Entrée_TOR_2"
34        S     "Sortie_TOR_0"
35        A     "Entrée_TOR_1"
36        R     "Sortie_TOR_0"
37
```

La fonction SET met à 1 une variable et maintient cette valeur tant que la fonction RESET est à 0.

Normalement, les fonctions SET et RESET ne doivent pas passer en même temps à 1.

La priorité est toujours placée sur la dernière instruction exécutée. Ainsi dans l'exemple ci-dessous c'est l'instruction R qui est exécutée en dernier et donc priorité est donnée à l'entrée I0.1 avec commande de la sortie %Q0.0 à 0.

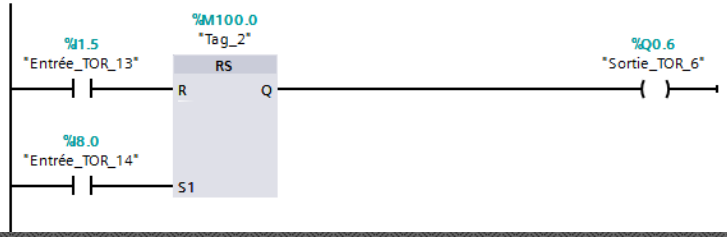
Le langage SCL demande l'utilisation d'une structure de contrôle IF pour arriver au même résultat. Comme pour le langage LIST, nous avons fixé une priorité sur l'entrée %I0.1 mais cette fois-ci c'est avec le test sur %I0.1 qui est fait en premier , ce qui fixe la priorité sur cette entrée et donc si les 2 entrées %I0.1 et %I0.2 sont à 1, la sortie %Q0.0 sera forcée à 0.

III.4: Opérations binaires et combinatoires

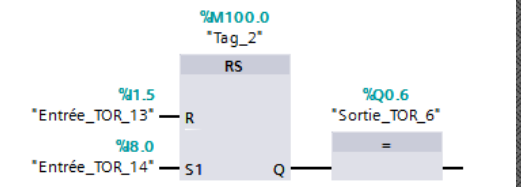
❑ Bascules RS :

SIEMENS Totally Integrated Automation PORTAL V15

Langage à contacts



Logigramme



Texte structuré (SCL)

```
// Bascule RS
IF "Entrée_TOR_13" THEN
    "Tag_2" := 0;
ELSIF "Entrée_TOR_14" THEN
    "Tag_2":= 1;
END_IF;
"Sortie_TOR_6":= "Tag_2";

```

Liste d'instructions (LIST)

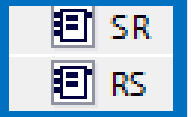
```
//Bascule RS
      A     "Entrée_TOR_13"
      S     "Tag_2"
      A     "Entrée_TOR_14"
      R     "Tag_2"
      A     "Tag_2"
      =     "Sortie_TOR_6"
```

Une fonction mémoire RS est représentée comme un rectangle avec l'entrée d'initialisation S et celle de réinitialisation R. Un état bref de signal à 1 à l'entrée d'initialisation initialise la fonction mémoire.

Normalement dans une bascule RS, les deux entrées R et S ne doivent pas passer en même temps à 1.

Il existe deux types de bascule RS

- SR: Priorité du l'entrée R (R1)
- RS: Priorité du l'entrée S (S1)



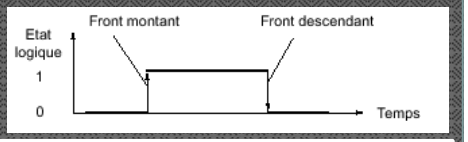
Dans notre exemple la mémorisation de l'état de la bascule est affecté à la variable %M100.0.

En SCL, même méthode de programmation que pour les fonctions SET et RESET (voir transparent 171).

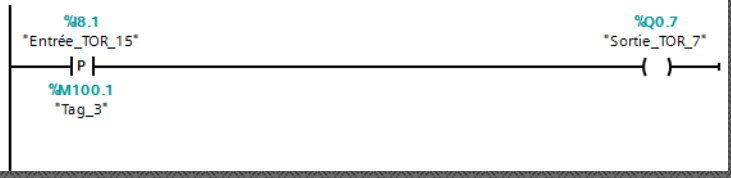
III.4: Opérations binaires et combinatoires

❑ Opérations sur front : Front montant

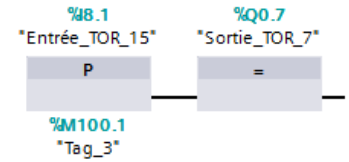
SIEMENS Totally Integrated Automation PORTAL V15



Langage à contacts



Logigramme



Texte structuré (SCL)

```
1  //front montant
2  IF "Entrée_TOR_15"= 1 AND "Tag_3" = 0
3  THEN
4      "Sortie_TOR_7" := 0;
5  ELSE
6      "Sortie_TOR_7" := 1;
7  END_IF;
8  "Tag_3" := "Entrée_TOR_15";
9
```

Liste d'instructions (LIST)

```
63  //front montant
64        A     "Entrée_TOR_15"
65        FP    "Tag_3"
66        =     "Sortie_TOR_7"
67
```

Les opérations sur front détectent la variation de niveau du signal par exemple celle d'une entrée, contrairement aux opérations détectant un état du signal statique (0 ou 1).

Dans l'exemple ci-contre, si un front montant (P: pente positive) traduisant le passage de '0' à '1' est reconnu sur %I8.1, alors %Q 0.7 est mise à 1 pour un cycle API.

La reconnaissance du front montant est enregistré dans un bit de mémoire interne %M100.1, et sa valeur est comparée avec celle du cycle précédent.

En SCL, on utilise la structure IF avec mise à jour dans un bit de mémoire interne%M100.1 après la structure de contrôle afin de préparer le prochain test au cycle suivant.

En LIST, on utilise l'instruction FP associée à un bit de mémoire interne %M100.1.

III.4: Opérations binaires et combinatoires

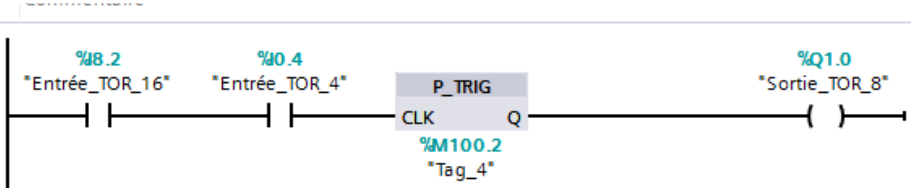
❑ Opérations sur front : Front montant

SIEMENS Totally Integrated Automation PORTAL V15

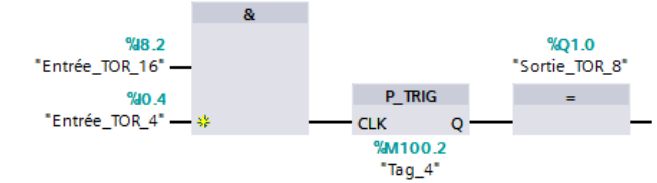
Remarque : L'adresse du bit de front ne doit pas être utilisée plusieurs fois dans le programme, car sa valeur serait alors écrasée. Il faut que l'adresse soit univoque.

Le langage à contact et logigramme proposent également un bloc (P_TRIG). Cette instruction permet de tester le front montant d'une équation. Dans l'exemple ci-dessous, on détecte le front montant de l'équation (%I8.2 . %I0.4). Ce bloc génère une instruction BLD 100 (graphisme du bloc) en langage LIST.

Langage à contacts



Logigramme



Liste d'instructions (LIST)

```
57  //Front montant
58        A     "Entrée_TOR_16"
59        A     "Entrée_TOR_4"
60        BLD   100                    //bloc Ptrig
61        FP    "Tag_4"
62        =     "Sortie_TOR_8"
63
```

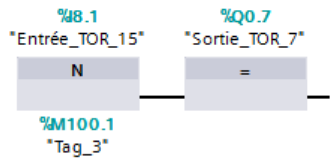
III.4: Opérations binaires et combinatoires

❑ Opérations sur front : Front descendant

SIEMENS Totally Integrated Automation PORTAL V15

Langage à contacts

Logigramme



Texte structuré (SCL)

```
1  //front descendant
2  IF "Entrée_TOR_15"=0 AND "Tag_3" = 1
3  THEN
4      "Sortie_TOR_7" := 0;
5  ELSE
6      "Sortie_TOR_7" := 1;
7  END_IF;
8  "Tag_3" := "Entrée_TOR_15";
9
```

Liste d'instructions (LIST)

```
52  //front descendant
53        A     "Entrée_TOR_15"
54        FN    "Tag_3"
55        =     "Sortie_TOR_7"
56
```

Dans l'exemple ci-contre, si un front descendant (N: pente négative) (Passage de '1'à '0') est reconnu sur %I8.1, alors %Q0.7 est mise à 1 pour un cycle API.

La reconnaissance de front descendant est enregistré dans un bit de mémoire interne de front %M100.1, et sa valeur est comparée avec celle du cycle précédent.

En SCL, on utilise la structure IF avec mise à jour dans un bit de mémoire interne %M100.1 après la structure de contrôle afin de préparer le prochain test au cycle suivant.

En LIST, on utilise l'instruction FN associée à un bit de mémoire interne %M100.1.

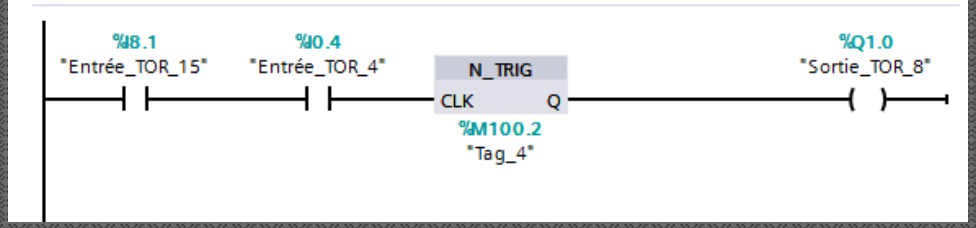
III.4: Opérations binaires et combinatoires

❑ Opérations sur front : Front descendant

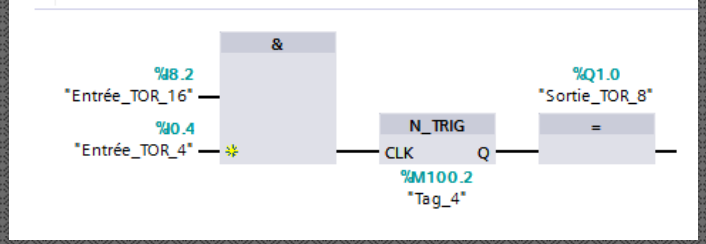
SIEMENS Totally Integrated Automation PORTAL V15

Le langage à contact et logigramme proposent également un bloc (N_TRIG). Cette instruction permet de tester le front descendant d'une équation. Dans l'exemple ci-dessous, on détecte le front descendant de l'équation (%I8.1 . %I0.4). Ce bloc génère une instruction BLD 100 (graphisme du bloc) en langage LIST.

Langage à contacts



Logigramme



Liste d'instructions (LIST)

```
46  //Front descendant
47        A     "Entrée_TOR_16"
48        A     "Entrée_TOR_4"
49        BLD   100                    //bloc Ntrig
50        FN    "Tag_3"
51        =     "Sortie_TOR_7"
52
```

III.5: Opérations numériques

Introduction

- Pour nombre de tâches de commande, il est nécessaire de programmer l'API en utilisant des opérations numériques.

- Ces opérations sont programmables dans l'ensemble des langages normalisés, mais en règle générale, les automaticiens privilégient souvent le SCL (langage à contact) ou le LIST (logigramme) pour répondre à ce besoin de programmation.

- Pour faciliter le travail de programmation, les API disposent d'instructions élémentaires intégrées dans des bibliothèques.

- Le type d'opération numérique est liée à la variable (type et format). Ainsi les API proposent des variables de type INT, DINT, REAL etc. Le format dépendra du type d'opération. Exemple: déclaration d'un DINT en format real pour une opération de type sinus.

- Les opérations numériques font le plus souvent appel aux instructions suivantes :
  - Chargement et transfert de valeur
  - Opérations arithmétiques
  - Comparaisons
  - Conversions
  - ETC.

III.5: Opérations numériques

❑ Chargement-conversion-comparaison :

SIEMENS Totally Integrated Automation PORTAL V15

Transfert	
MOVE	Copier valeur
Deserialize	Désérialiser
Serialize	Sérialiser
MOVE_BLK	Copier zone
MOVE_BLK_VARIANT	Copier zone
UMOVE_BLK	Copier zone contiguë
FILL_BLK	Compléter zone
UFILL_BLK	Compléter zone contiguë
SCATTER	Décomposer une suite de bits en ses bits constitutifs
SCATTER_BLK	Décomposer les éléments d'un ARRAY d'une suite de bits en ses bits constitutifs
GATHER	Composer une suite de bits à partir de bits individuels
GATHER_BLK	Composer plusieurs éléments d'un ARRAY d'une suite de bits à partir de bits individuels
SWAP	Permutation
Conversion	
CONVERT	Convertir valeur
ROUND	Arrondir nombre
CEIL	Arrondir à l'entier supérieur
FLOOR	Arrondir à l'entier inférieur
TRUNC	Former un nombre entier
SCALE_X	Mise à l'échelle
NORM_X	Normaliser
Comparaison	
CMP ==	Egal à
CMP <>	Différent de
CMP >=	Supérieur ou égal à
CMP <=	Inférieur ou égal à
CMP >	Supérieur à
CMP <	Inférieur à
IN_Range	Valeur dans la plage
OUT_Range	Valeur en dehors de la plage
–|OK|–	Contrôler validité
–|NOT_OK|–	Contrôler invalidité

III.5: Opérations numériques

❑ Fonctions mathématiques :

SIEMENS Totally Integrated Automation PORTAL V15

Fonctions mathématiques	
CALCULATE	Calculer
ADD	Addition
SUB	Soustraction
MUL	Multiplication
DIV	Division
MOD	Calculer le reste de la division
NEG	Créer le complément à 2
INC	Incrémenter
DEC	Décrémenter
ABS	Valeur absolue
MIN	Calculer le minimum
MAX	Calculer le maximum
LIMIT	Définir une limite
SQR	Carré
SQRT	Racine carrée
LN	Logarithme népérien
EXP	Fonction exponentielle
SIN	Sinus
COS	Cosinus
TAN	Tangente
ASIN	Arc sinus
ACOS	Arc cosinus
ATAN	Arc tangente
FRAC	Décimales
EXPT	Elever à la puissance

III.5: Opérations numériques

❑ Opérations dur mots-décalage et rotation :

SIEMENS Totally Integrated Automation PORTAL V15

Opérations logiques sur mots	
AND	Opération logique ET
OR	Opération logique OU
XOR	Opération logique OU EXCLUSIF
INVERT	Former le complément à 1
DECO	Décoder
ENCO	Encoder
SEL	Sélectionner
MUX	Multiplexeur
DEMUX	Démultiplexeur

Décalage et rotation	
SHR	Décaler à droite
SHL	Décaler à gauche
ROR	Rotation à droite
ROL	Rotation à gauche

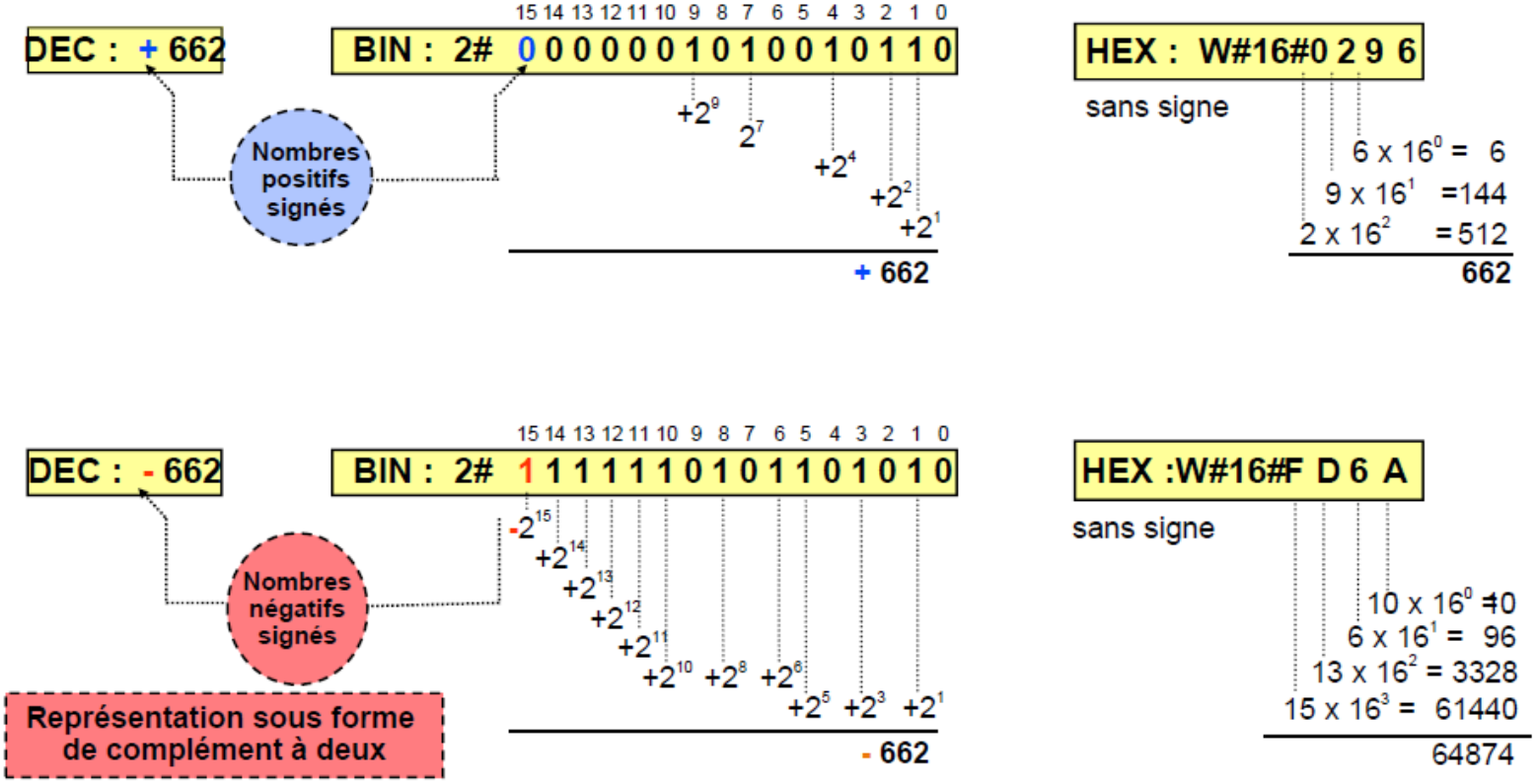
III.5: Opérations numériques

❑ Format Entier

Plage de valeurs - 32768 ... + 32767
(sans signe : 0 ... 65535)

Opérations arithmétiques : par ex. + I, * I, <I, ==I

Formats de représentation :



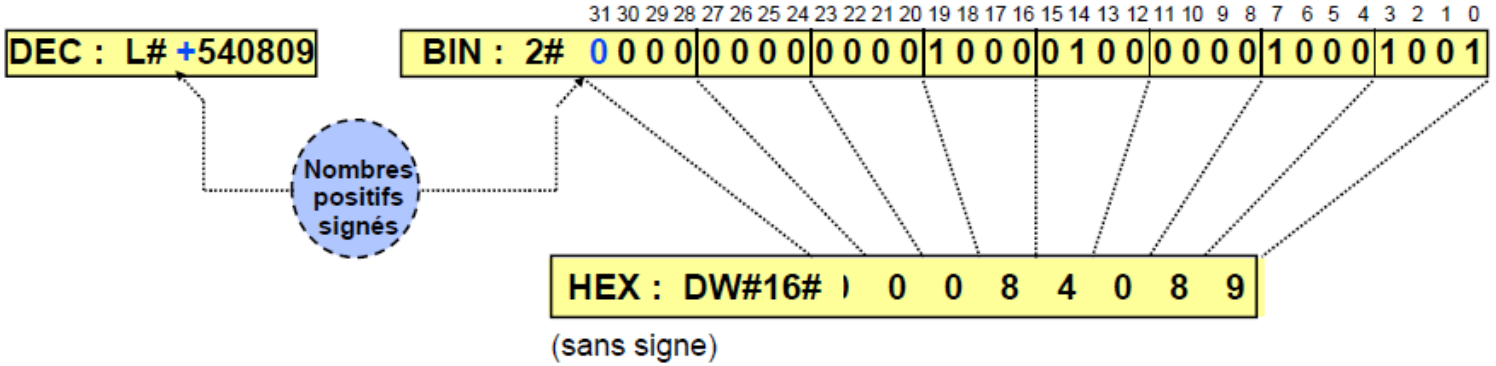
III.5: Opérations numériques

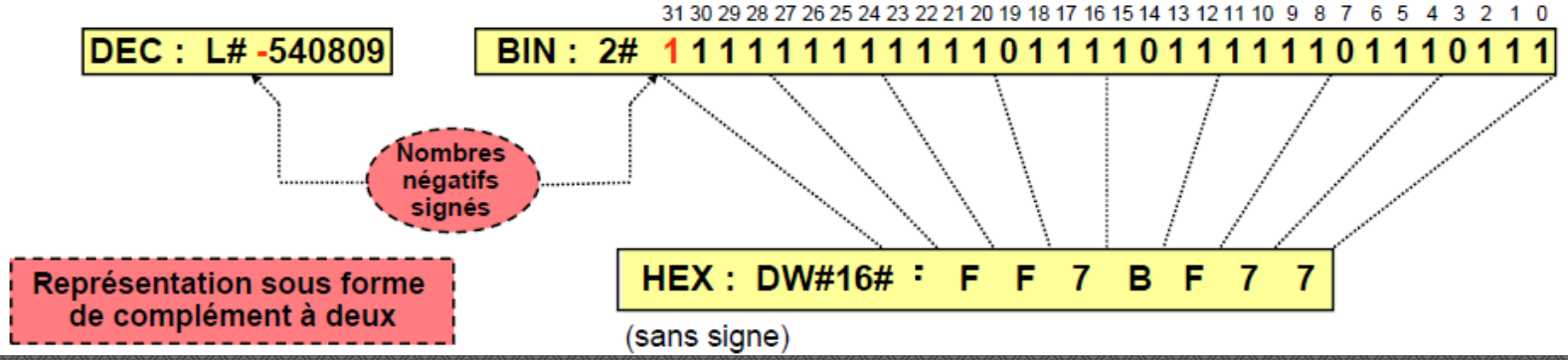
❑ Format Double - Entier

Plage de valeurs L# -2147483648 ... L#+2147483647
(sans signe : 0 ... 4294967295)

Opérations : par ex. + D, * D, <D, ==D

Formats de représentation :





III.5: Opérations numériques

❑ Format Réel

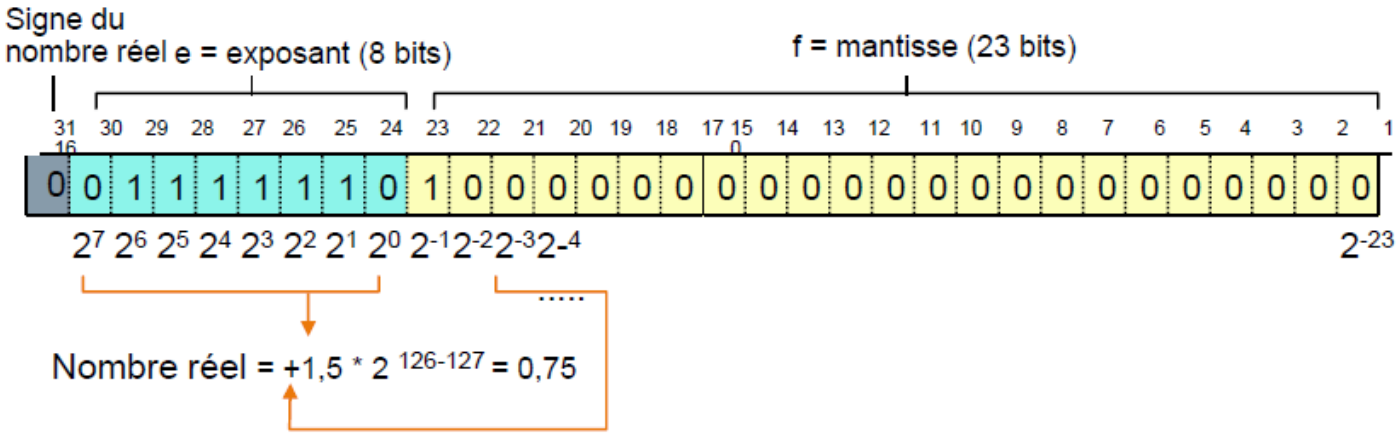
Plage de valeurs $- 3.402823 \cdot 10^{+38} \ldots - 1.175495 \cdot 10^{-38}$, 0.0, $+ 1.175495 \cdot 10^{-38} \ldots + 3.402823 \cdot 10^{+38}$

Opérations : par ex. **+ R, * R, <R, ==R**
sin, acos, ln, exp, SQR

Format général d'un nombre réel = (signe) • (1.f) • (2^{e-127})

Exemple : 0.75

Attention: Pour la syntaxe, la saisie d'une virgule est un point.



III.5: Opérations numériques

❑ Rappel du plan d'adressage

SIEMENS Totally Integrated Automation PORTAL V15

Le plan d'adressage SIEMENS est basé sur l'octet (voir transparent 162) .
Dans l'exemple ci-dessous, il a été déclaré un double mot %QD120 (format 32 bits).
Une table de visualisation permet de consulter la valeur de ce double mot dans des formats différents: (hexadécimal, décimal, binaire).

// SORTIE QD120 : AFFICHAGE HEXADECIMAL				
	"Tag_6"	%QD120	Hexa	16#AF31_1B82
		%QB120	Hexa	16#AF
		%QB121	Hexa	16#31
		%QB122	Hexa	16#1B
		%QB123	Hexa	16#82
// SORTIE QD120 : AFFICHAGE DECIMAL				
	"Tag_6"	%QD120	DEC	2_939_231_106
		%QB120	DEC	175
		%QB121	DEC	49
		%QB122	DEC	27
		%QB123	DEC	130
// SORTIE QD120 : AFFICHAGE BINAIRE				
	"Tag_6"	%QD120	Bin	2#1010_1111_0011_0001_0001_1011_1000_0010
		%QB120	Bin	2#1010_1111
		%QB121	Bin	2#0011_0001
		%QB122	Bin	2#0001_1011
		%QB123	Bin	2#1000_0010
		%Q120.7	BOOL	TRUE
		%Q123.0	BOOL	FALSE

Octet de poids fort de %QD120

Octet de poids faible de %QD120

Bit de poids fort de %QD120 (2^{31})

Bit de poids faible de %QD120 (2^{0})

// SORTIE QD120 : descendre au niveau du bit				
	"Tag_6"	%QD120	Hexa	16#AF31_1B82
		%QB120	Hexa	16#AF
		%QB120	Bin	2#1010_1111
		%Q120.0	BOOL	TRUE
		%Q120.1	BOOL	TRUE
		%Q120.2	BOOL	TRUE
		%Q120.3	BOOL	TRUE
		%Q120.4	BOOL	FALSE
		%Q120.5	BOOL	TRUE
		%Q120.6	BOOL	FALSE
		%Q120.7	BOOL	TRUE

Bit de poids faible de %QB120 (2^{0})

Bit de poids fort de %QB120 (2^{7})

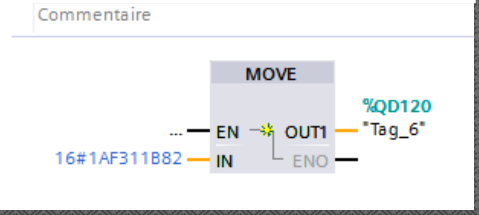
III.5: Opérations numériques

❑ Opérations de chargement et de transfert :

SIEMENS Totally Integrated Automation PORTAL V15

Langage à contacts

Logigramme



Texte structuré (SCL)

```
1  #calcul := ((4 / 1500) * "vitesse(tr/min)") + 1.0; // calcul en réel
2  "Ucapteur(V)" := #calcul; // transfert du résultat dans la variable Ucapteur(V)
3
```

Liste d'instructions (LIST)

```
1        A     "marche_arrêt"
2        JCN   saut1
3        L     16#47F
4        T     "CMD_VARIATEUR"
5   saut1: NOP 0
6
7
8        AN    "marche_arrêt"
9        JCN   saut2
10       L     16#47E
11       T     "CMD_VARIATEUR"
12  saut2: NOP 0
13
```

L'opération **" MOVE "** permet d'affecter une valeur dans un octet, un mot, un double mot.

Cette valeur d'affectation peut être une constante ou une valeur d'un autre mot.

Dans le cas d'affectation d'un mot a un autre de format diffèrent soit les octets de poids fort sont perdus soit ils sont rempli par des zéros.

L'opération MOVE n'effectue aucune vérification sur la compatibilité des formats.

En LIST, cette instruction se compose de deux parties:
L: LOAD (charger)
T: TRANSFER (transférer).

III.5: Opérations numériques

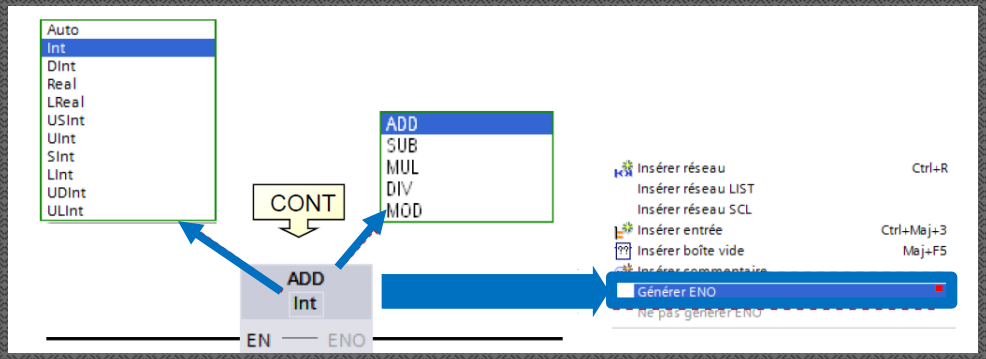
❑ Calculs :

SIEMENS Totally Integrated Automation PORTAL V15

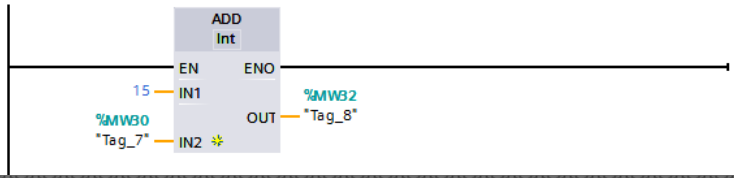
Différentes " **fonctions mathématiques** " sont intégrées dans la bibliothèque d'instruction.
Chaque instruction de calcul est entièrement paramétrable:

- ❖ type d'opération et format des variables.

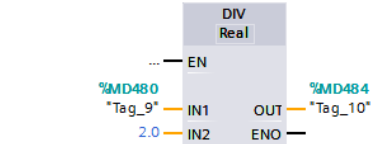
Il convient de choisir les bonnes variables en fonction de l'opération souhaitée.



Langage à contacts



Logigramme



Texte structuré (SCL)

```
1  // Calcul d'un signal sinus
2  "Tag_11":= 311*SIN(314 *#t);
3
```

Liste d'instructions (LIST)

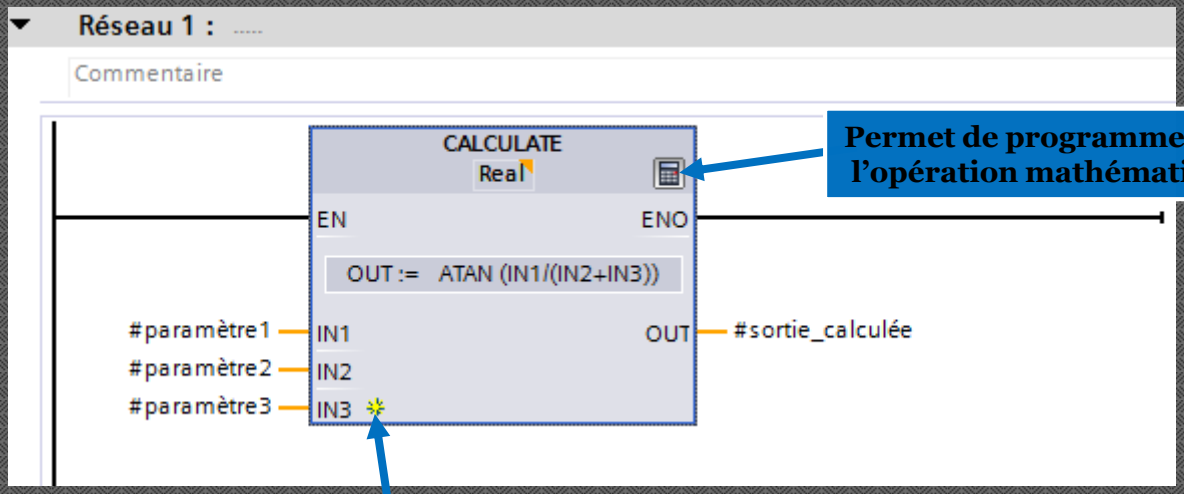
```
1        L     "Tag_6"
2        L     +15
3        -I
4        T     "Tag_7"
5
```

III.5: Opérations numériques

❑ Calculs :

SIEMENS Totally Integrated Automation PORTAL V15

Une nouvelle instruction " **CALCULATE** " est intégrée dans TIAPortal, elle permet de programmer plusieurs opérations mathématiques dans le même bloc.



Permet de programmer l'opération mathématique souhaitée

Permet d'ajouter des paramètres d'entrée

III.5: Opérations numériques

❑ Comparaisons :

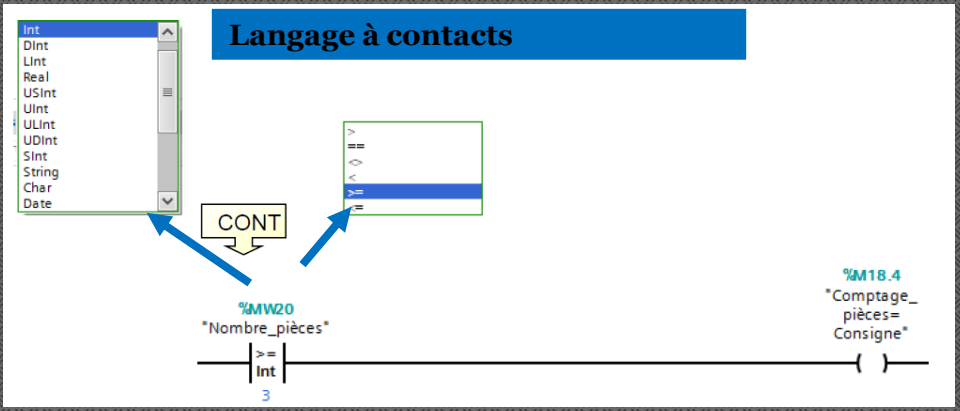
SIEMENS Totally Integrated Automation PORTAL V15

Différentes " **fonctions de comparaison** " sont intégrées dans la bibliothèque d'instruction.
Chaque instruction de comparaison est entièrement paramétrable:

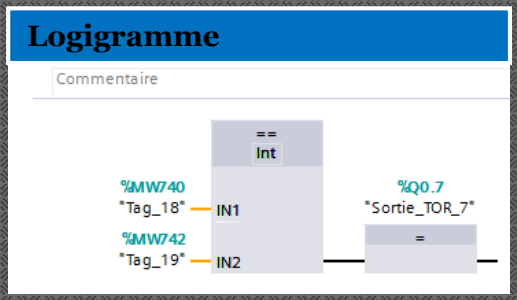
- ❖ type de comparaison et format des variables.

Il convient de choisir les bonnes variables en fonction de la comparaison souhaitée.
Lorsque l'opération de comparaison est vérifiée le résultat logique est mis à 1.

Langage à contacts



Logigramme



Texte structuré (SCL)

```
1  IF "Tag_18" <> "Tag_19"
2  THEN
3      "Tag_12" := 16#FFFF;
4  END_IF;
5
```

Liste d'instructions (LIST)

```
1        L     "Tag_7"
2        L     "Tag_6"
3        >=I
4        S     "Sortie_TOR_5"
5
```

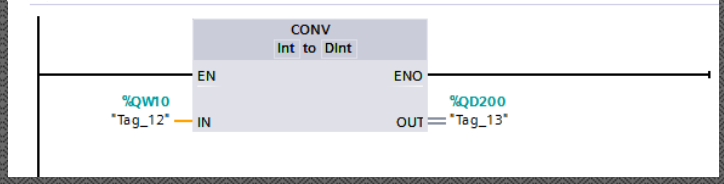
III.5: Opérations numériques

❑ Conversions :

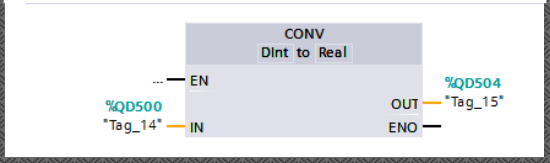
SIEMENS Totally Integrated Automation PORTAL V15

L'instruction **"Convertir valeur"** permet de programmer des conversions explicites. Lorsque l'instruction est insérée, la boîte de dialogue "Convertir valeur" s'ouvre. Vous y entrez le type de données source et le type de données cible de la conversion. La valeur source est lue et convertie dans le type de données cible indiqué.

Langage à contacts



Logigramme



Texte structuré (SCL)

```
1  "Tag_16" := INT_TO_BCD16("Tag_17");
2
```

Liste d'instructions (LIST)

```
1      L     "Tag_7"
2      ITD
3      T     "Tag_8"
4
```

D'autres opérations de conversion

III.5: Opérations numériques

❑ Mises à l'échelle :

SIEMENS Totally Integrated Automation PORTAL V15

Dans la plupart des systèmes automatisés, on effectue des mises à l'échelle.
Elaboration d'une consigne de vitesse d'un variateur, affichage de la valeur d'un capteur d'instrumentation sur une IHM, mise à l'échelle d'une sortie calculée par un bloc PID, etc.
Pour effectuer ces mises à l 'échelle, des outils sont à disposition : **"NORM_X " et " SCALE _X"**

Plage	Courant Par ex:	
	Plage 4 à 20mA	Unités
Débordement	>= 22.815	32767
Domaine de dépassement	22.810 : 20.0005	32511 : 27649
Plage nominale	20.000 16.000 : : 4.000	27648 20736 : : 0
Domaine de dépassement	3.9995 : 1.1852	- 1 : - 4864
Débordement	<= 1.1845	- 32768

Rappel de la plage normalisée des grandeurs analogiques.

Dans l'exemple qui suivra on prendra un capteur analogique 4-20 mA qui mesure le poids de pièces.

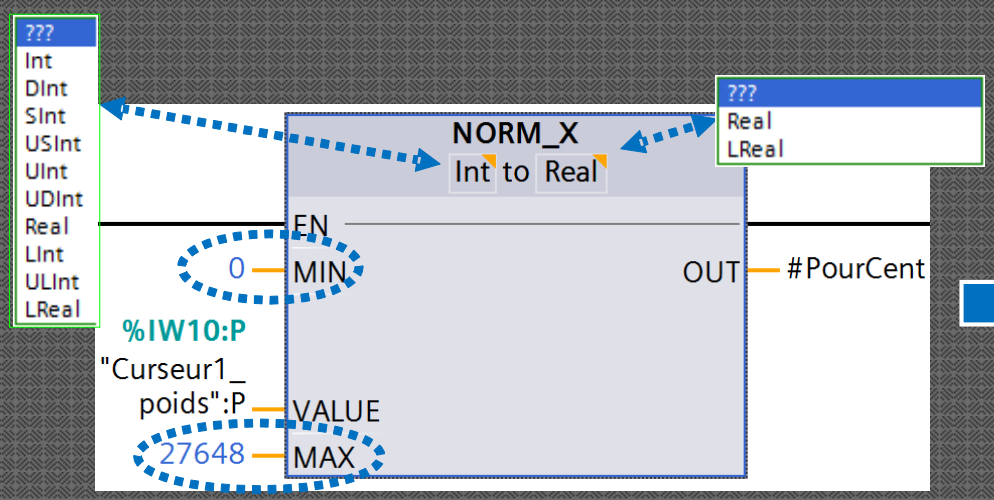
On considère que l'on obtient l'étendue de l'échelle du capteur soit:

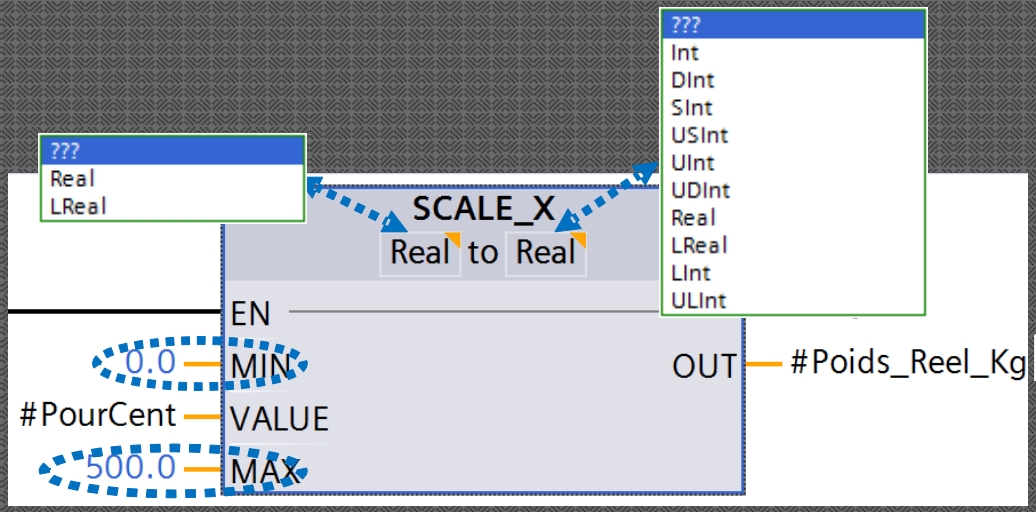
- **0 numérique pour 4mA**
- **27648 numérique pour 20mA**

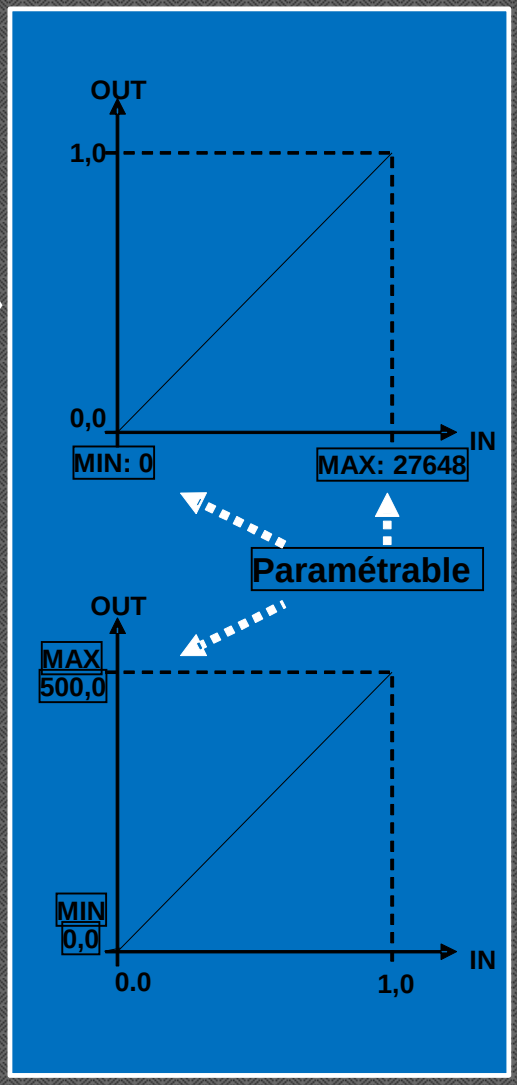
III.5: Opérations numériques

❑ Mises à l'échelle :

SIEMENS Totally Integrated Automation PORTAL V15





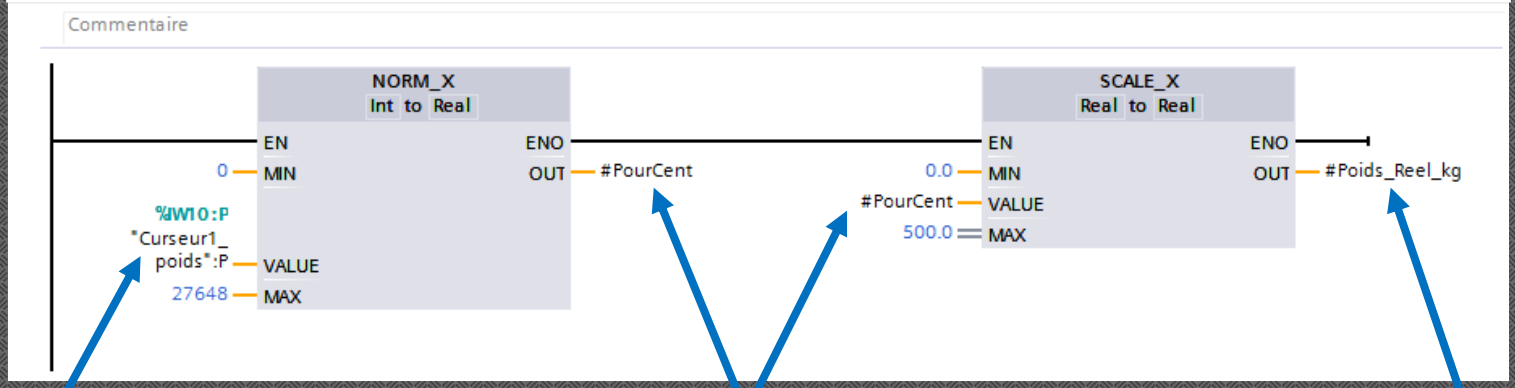


III.5: Opérations numériques

❑ Mises à l'échelle :

SIEMENS Totally Integrated Automation PORTAL V15

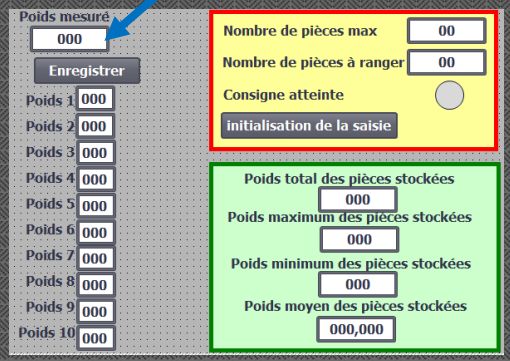
Langage à contacts



Capteur de poids 4-20mA
La valeur numérique du CNA : (0-27648)

Variable temporaire format « real » : (0.0 - 1.0)

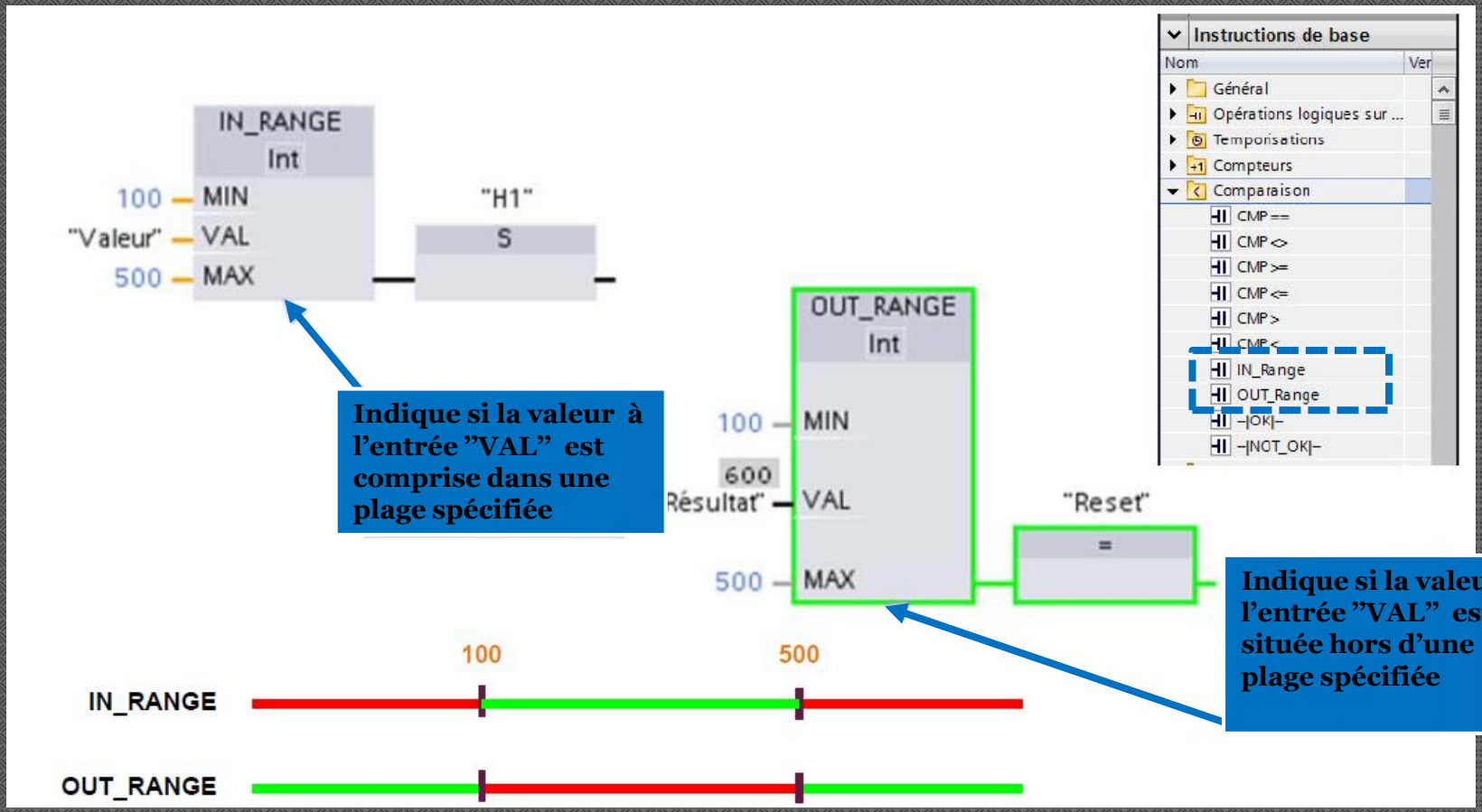
Poids mis à l'échelle (0.0kg – 500.0kg)



III.5: Opérations numériques

❑ Autres opérations utiles:

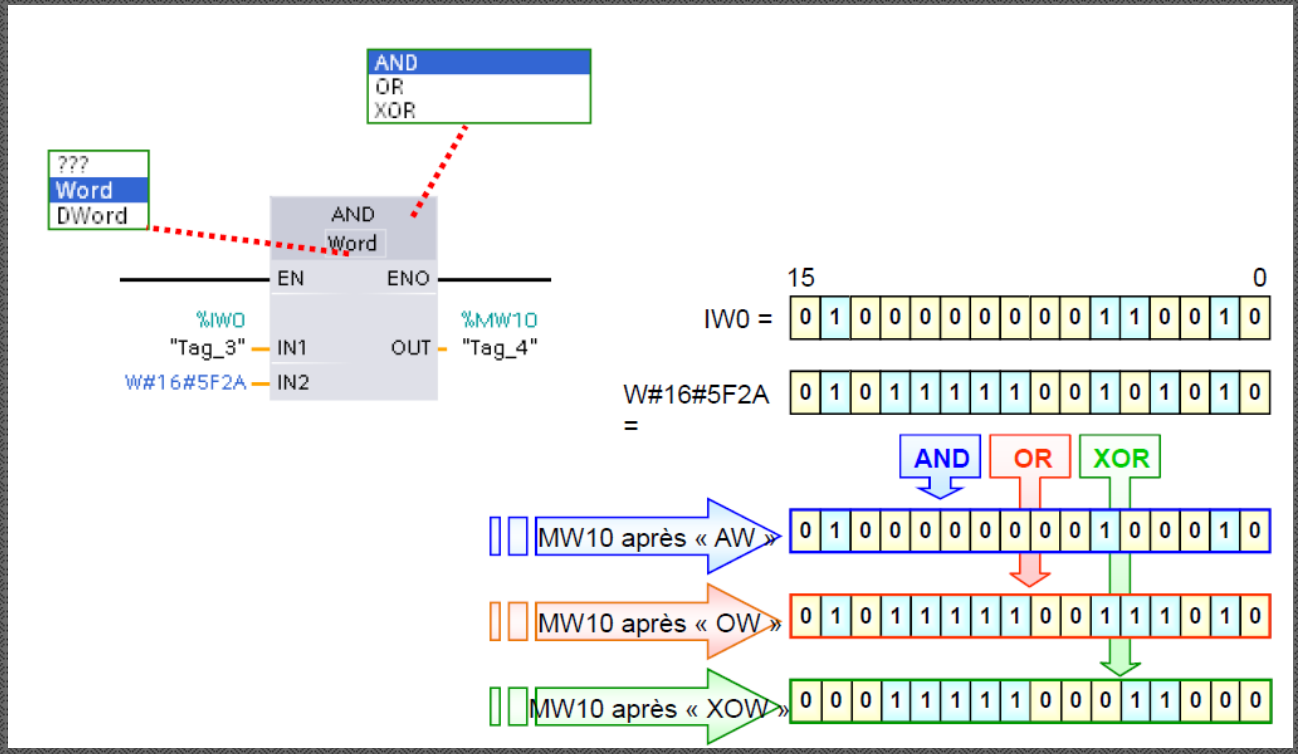
SIEMENS Totally Integrated Automation PORTAL V15



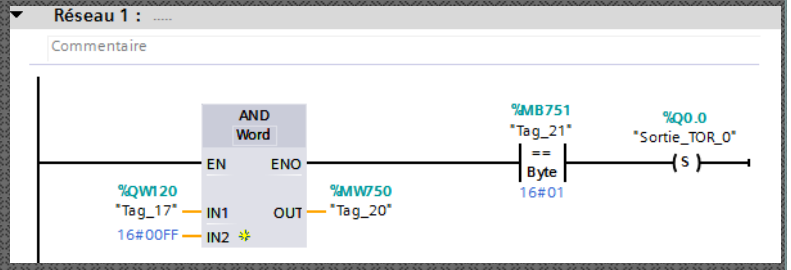
III.5: Opérations numériques

❑ Autres opérations utiles:

SIEMENS Totally Integrated Automation PORTAL V15



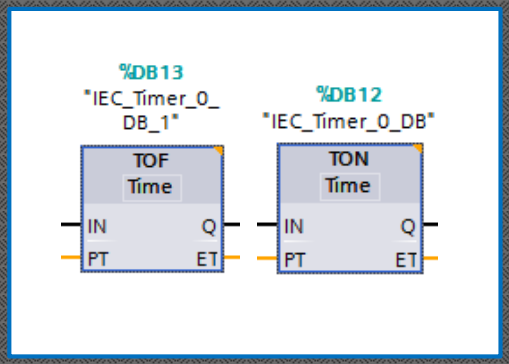
Exemple d'un masque en ET :
Extraire l'octet de poids faible d'un mot simple



III.6: Temporisations

Introduction

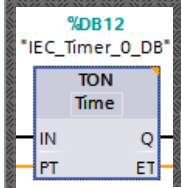
- Pour nombre de tâches de commande, il est nécessaire de contrôler le temps. Par exemple, un moteur ou une pompe peuvent avoir besoin de fonctionner pendant une durée précise ou être mise en marche après un certain temps.
- Les API disposent de temporisateurs intégrés.
- Les temporisateurs comptent les secondes ou les fractions de seconde en utilisant l'horloge interne de la CPU.
- Tous les constructeurs intègrent des temporisateurs dans leur API. Ils répondent également à une norme CEI.
- Les temporisateurs les plus utilisés sont :
  - TON: retard à la montée
  - TOF: retard à la retombée



III.6: Temporisations

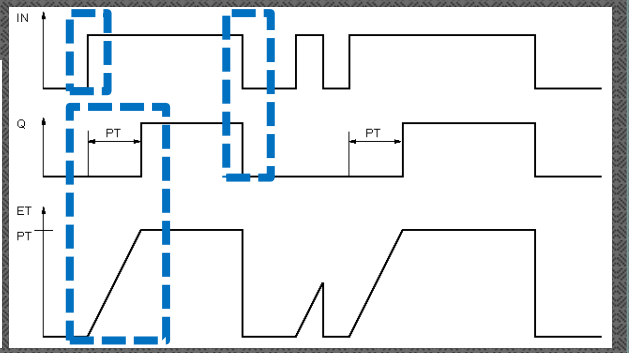
❑ TON : Retard à la montée

SIEMENS Totally Integrated Automation PORTAL V15



4 paramètres

Paramètres	Déclaration	Type de données		Zone de mémoire		Description
		S7-1200	S7-1500	S7-1200	S7-1500	
IN	Input	BOOL	BOOL	I, Q, M, D, L	I, Q, M, D, L, P	Entrée de démarrage
PT	Input	TIME	TIME, LTIME	I, Q, M, D, L ou constante	I, Q, M, D, L, P ou constante	Durée d'un retard à la montée La valeur du paramètre PT doit être positive.
Q	Output	BOOL	BOOL	I, Q, M, D, L	I, Q, M, D, L, P	Sortie mise à 1 après écoulement de la durée PT.
ET	Output	TIME	TIME, LTIME	I, Q, M, D, L	I, Q, M, D, L, P	Valeur de temps actuelle



Description

L'instruction "Retard à la montée" vous permet de retarder la mise à 1 de la sortie Q de la durée programmée PT.

- ❖ L'instruction est démarrée lorsque l'entrée IN passe de "0" à "1" (front montant).
- ❖ La durée PT programmée débute au démarrage de l'instruction.
- ❖ Une fois la durée PT écoulée, la sortie Q fournit l'état logique "1". La sortie Q reste à 1 tant que l'entrée In reste à"1".
- ❖ Lorsque l'entrée IN passe de "1" à "0", la sortie Q est remise à 0.
- ❖ La valeur de temps actuelle peut être demandée à la sortie ET. La valeur de temps débute à T#0s et se termine lorsque la durée PT est atteinte. La sortie ET est remise à 0 dès que l'état logique à l'entrée IN passe à "0".

Il faut associer à chaque appel de l'instruction "Retard à la montée" une temporisation CEI (instance) dans laquelle les données de la temporisation sont stockées.

IEC_Timer_0_DB

	Nom	Type de données	Valeur de départ
1	Static		
2	PT	Time	T#0ms
3	ET	Time	T#0ms
4	IN	Bool	false
5	Q	Bool	false

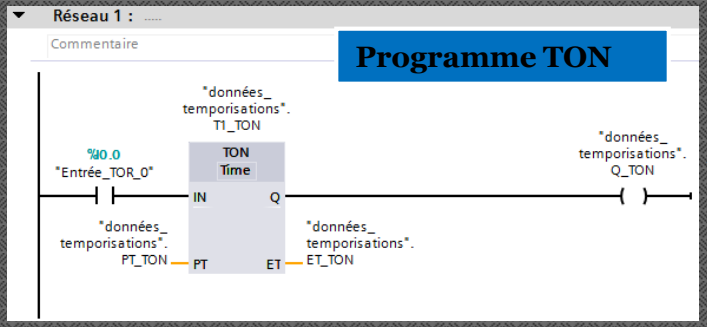
Remarque: le paramètre PT peut être associé à une variable au format DINT (ms)

III.6: Temporisations

❑ TON : exemple de programmation

SIEMENS Totally Integrated Automation PORTAL V15

Programme TON

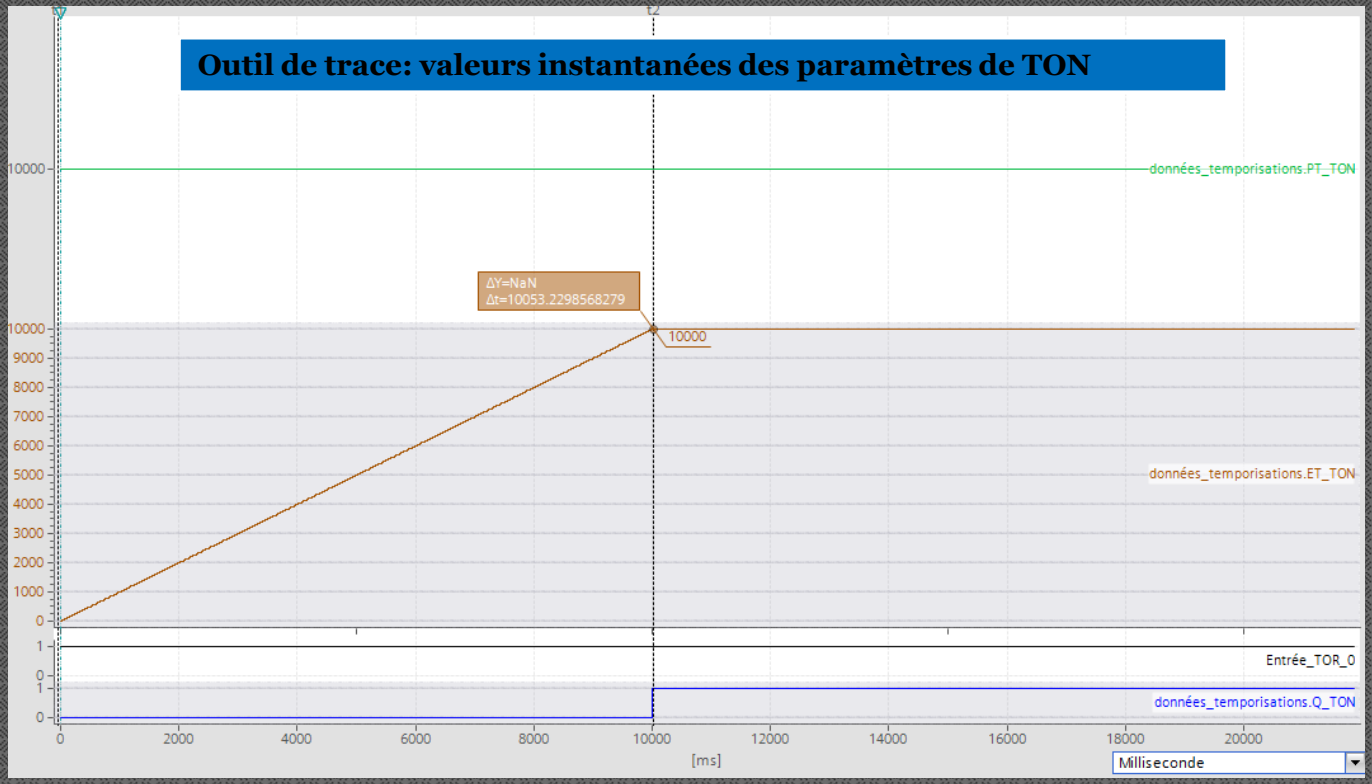


Instance TON

données_temporisations

	Nom	Type de données	Valeur de départ
1	Static		
2	T1_TON	IEC_TIMER	
3	PT	Time	T#0ms
4	ET	Time	T#0ms
5	IN	Bool	false
6	Q	Bool	false
7	T1_TOF	IEC_TIMER	
8	PT_TON	DInt	10000
9	ET_TON	DInt	0
10	Q_TON	Bool	false
11	PT_TOF	DInt	10000
12	ET_TOF	DInt	0
13	Q_TOF	Bool	false

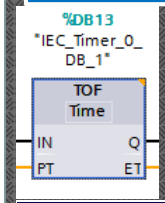
Outil de trace: valeurs instantanées des paramètres de TON



III.6: Temporisations

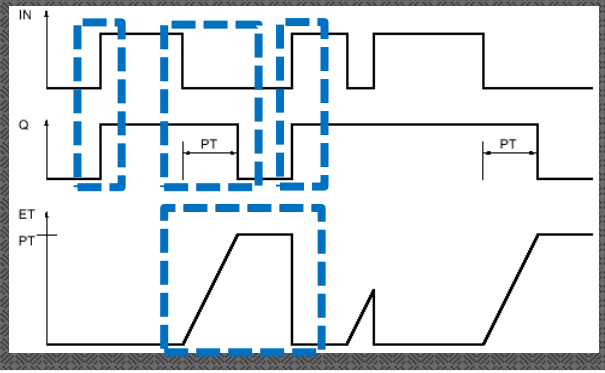
❑ TOF : Retard à la retombée

SIEMENS Totally Integrated Automation PORTAL V15



4 paramètres

Paramètres	Déclaration	Type de données		Zone de mémoire		Description
		S7-1200	S7-1500	S7-1200	S7-1500	
IN	Input	BOOL	BOOL	I, Q, M, D, L	I, Q, M, D, L, P	Entrée de démarrage
PT	Input	TIME	TIME, LTIME	I, Q, M, D, L ou constante	I, Q, M, D, L, P ou constante	Durée d'un retard à la retombée La valeur du paramètre PT doit être positive.
Q	Output	BOOL	BOOL	I, Q, M, D, L	I, Q, M, D, L, P	Sortie mise à 0 après écoulement de la durée PT.
ET	Output	TIME	TIME, LTIME	I, Q, M, D, L	I, Q, M, D, L, P	Valeur de temps actuelle



Description

L'instruction "Retard à la montée" vous permet de retarder la mise à 0 de la sortie Q de la durée programmée PT.

- La sortie Q est mise à 1 lorsque à l'entrée IN passe de "0" à "1" (front montant).
- Lorsque l'état logique à l'entrée IN repasse à "0", la durée programmée PT démarre. La sortie Q reste à 1 tant que la durée PT s'écoule.
- Une fois la durée PT écoulée, la sortie Q est remise à 0.
- La valeur de temps actuelle peut être demandée à la sortie ET. La valeur de temps débute à T#0s et se termine lorsque la durée PT est atteinte.. Si l'entrée IN passe à "1" avant que la durée PT soit écoulée, la sortie ET est remise à la valeur T#0s.

Il faut associer à chaque appel de l'instruction "Retard à la retombée" une temporisation CEI dans laquelle les données de l'instruction sont stockées.

IEC_Timer_0_DB

	Nom	Type de données	Valeur de départ
1	Static		
2	PT	Time	T#0ms
3	ET	Time	T#0ms
4	IN	Bool	false
5	Q	Bool	false

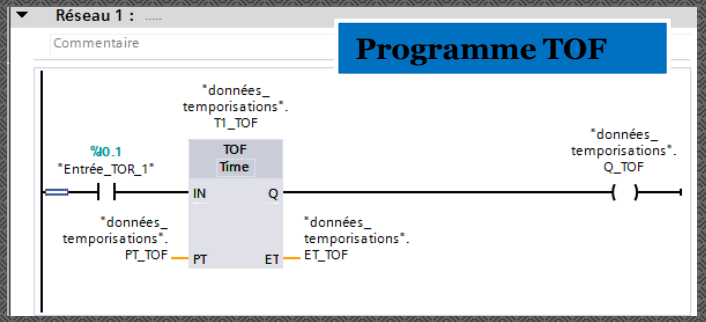
Remarque: le paramètre ET peut être associé à une variable au format DINT (ms)

III.6: Temporisations

❑ TOF : Retard à la retombée

SIEMENS Totally Integrated Automation PORTAL V15

Programme TOF

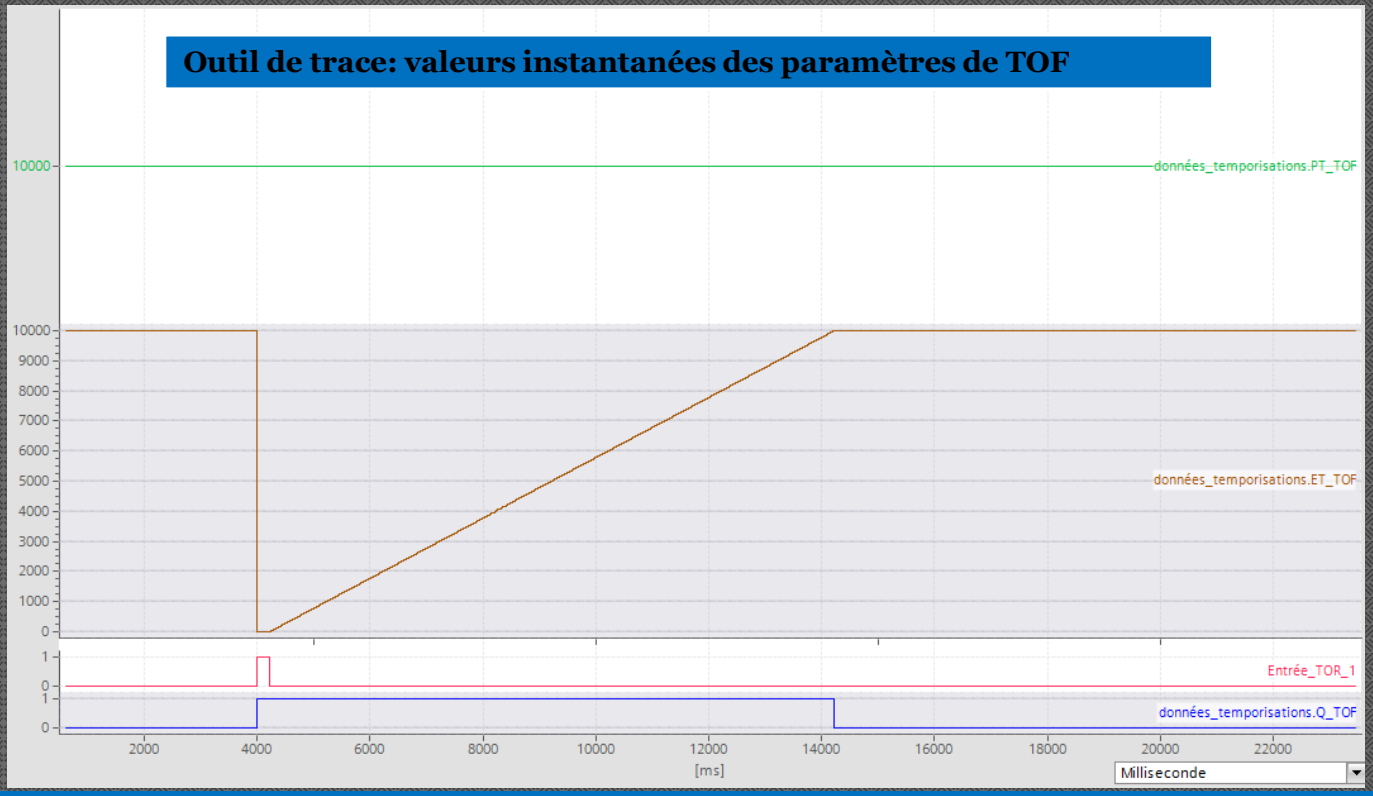


Instance TOF

données_temporisations

	Nom	Type de données	Valeur de départ
1	Static		
2	T1_TON	IEC_TIMER	
3	T1_TOF	IEC_TIMER	
4	PT	Time	T#0ms
5	ET	Time	T#0ms
6	IN	Bool	false
7	Q	Bool	false
8	PT_TON	DInt	10000
9	ET_TON	DInt	0
10	Q_TON	Bool	false
11	PT_TOF	DInt	10000
12	ET_TOF	DInt	0
13	Q_TOF	Bool	false

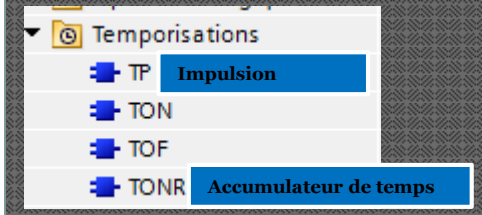
Outil de trace: valeurs instantanées des paramètres de TOF



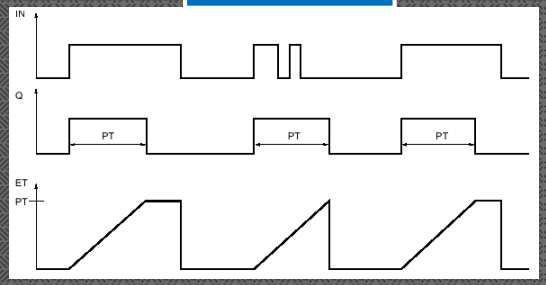
III.6: Temporisations

❑ Autres temporisations

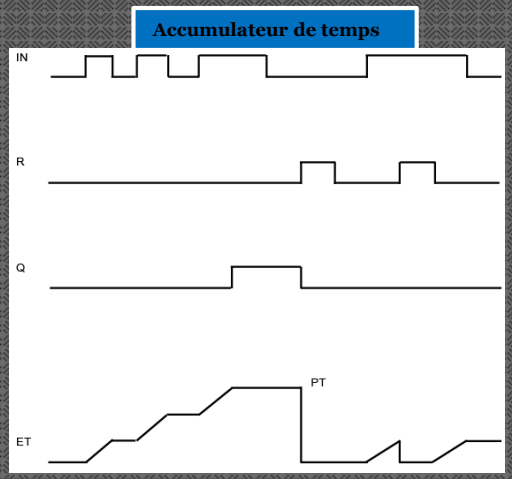
SIEMENS Totally Integrated Automation PORTAL V15



Impulsion

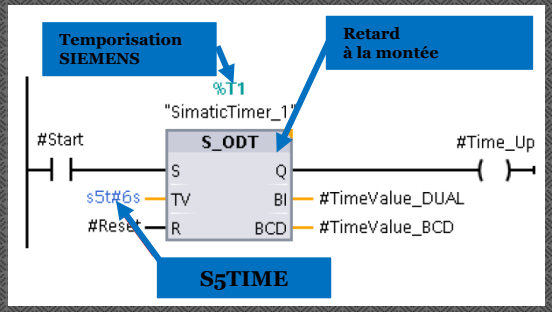


Accumulateur de temps

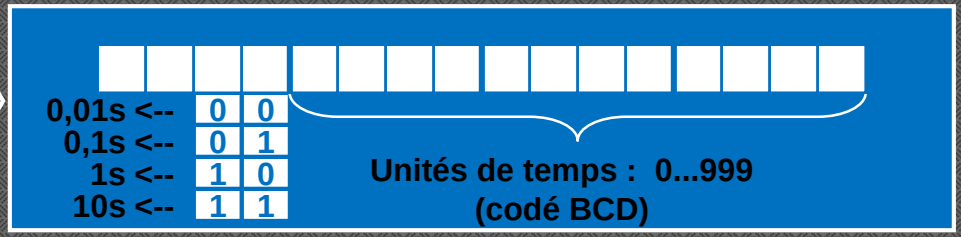


Siemens propose également ses propres temporisateurs. La variable de réglage du temps est au format S5TIME

S_PULSE	Impulsion
S_PEXT	Impulsion prolongée
S_ODT	Retard à la montée
S_ODTS	Retard à la montée mémorisé
S_OFFDT	Retard à la retombée



« S5TIME »



Temps réglable entre 10ms à 2h46min30s
Un temps de 10ms donne :
0000 0000 0000 0001
Un temps de 2h46min30s donne :
0011 1001 1001 1001

III.6: Temporisations

❑ Temporisations en SCL ou LIST

SIEMENS Totally Integrated Automation PORTAL V15

```
1  //temporisation IEC
2  "IEC_Timer_0_DB_2".TON(IN:="X10",
3                         PT:=T#10s,
4                         Q=>"ft1",
5                         ET=>#tps_en_cours);
6
7
```

SCL

Paramètres
IN: entrée de temporisation
PT: Temps de réglage
Q: sortie de temporisation
ET: Valeur courant de la temporisation

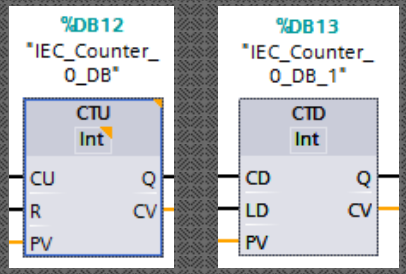
```
1
2       CALL  TON , "IEC_Timer_0_DB"
3           Time
4           IN :="x10"
5           PT :=T#10s
6           Q  :="ft1"
7           ET :=#temps_en_cours
8
```

LIST

III.7: Compteurs -Décompteurs

Introduction

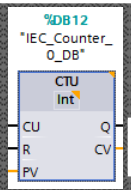
- Dans les API , les compteurs sont disponibles sous formes d'éléments intégrés et permettent de compter le nombre d'occurrences de signaux d'entrée.
- Cela peut servir à compter des produits lorsqu'ils passent sur une bande transporteuse.
- Tous les constructeurs intègrent des compteurs/décompteurs dans leurs API. Ils répondent également à une norme CEI.
- Les temporisateurs les plus utilisés sont :
  - CTU: count UP (compteur)
  - CTD: count DOWN (décompteur)



III.7: Compteurs -Décompteurs

❑ CTU: Compteur

SIEMENS Totally Integrated Automation PORTAL V15



5 paramètres

Paramètre	Déclaration	Type de données	Zone de mémoire		Description
			S7-1200	S7-1500	
CU	Input	BOOL	I, Q, M, D, L ou constante	I, Q, M, D, L ou constante	Entrée de comptage
R	Input	BOOL	I, Q, M, D, L, P ou constante	I, Q, M, D, L, T, C, P ou constante	Entrée de remise à zéro
PV	Input	Nombres entiers	I, Q, M, D, L, P ou constante	I, Q, M, D, L, P ou constante	Valeur pour laquelle la sortie Q est mise à 1.
Q	Output	BOOL	I, Q, M, D, L	I, Q, M, D, L	Etat du compteur
CV	Output	Nombres entiers, CHAR, WCHAR, DATE	I, Q, M, D, L, P	I, Q, M, D, L, P	Valeur de comptage actuelle

Instance CTU

IEC_Counter_0_DB

	Nom	Type de données	Valeur de départ
1	Static		
2	CU	Bool	false
3	CD	Bool	false
4	R	Bool	false
5	LD	Bool	false
6	QU	Bool	false
7	QD	Bool	false
8	PV	Int	0
9	CV	Int	0

Description :

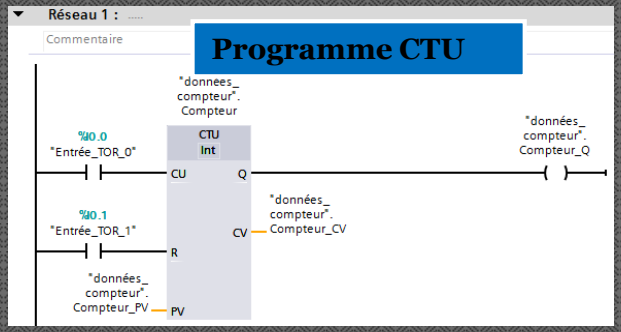
- ❑ Avec l'instruction "Comptage", on incrémente la valeur à la sortie CV.
- ❑ Quand l'état logique passe de "0" à "1" (front montant) à l'entrée CU, l'instruction est exécutée et la valeur de comptage actuelle à la sortie CV est augmentée de un. La valeur de comptage est incrémentée à chaque détection d'un front montant jusqu'à ce qu'elle atteigne la valeur limite supérieure du type de données spécifié à la sortie CV. Une fois la valeur limite supérieure atteinte, l'état logique à l'entrée CU n'a plus d'influence sur l'instruction.
- ❑ On peut interroger l'état du compteur à la sortie Q. L'état logique à la sortie Q est déterminé par le paramètre PV. Quand la valeur de comptage actuelle est supérieure ou égale à la valeur du paramètre PV, la sortie Q est mise à l'état logique "1". Dans tous les autres cas, l'état logique à la sortie Q est "0".
- ❑ La valeur à la sortie CV est remise à zéro lorsque l'état logique à l'entrée R passe à "1". Tant que l'entrée R présente l'état logique "1", l'état logique à l'entrée CU n'a pas d'effet sur l'instruction.

III.7: Compteurs -Décompteurs

❑ CTU : exemple de programmation

SIEMENS Totally Integrated Automation PORTAL V15

Programme CTU

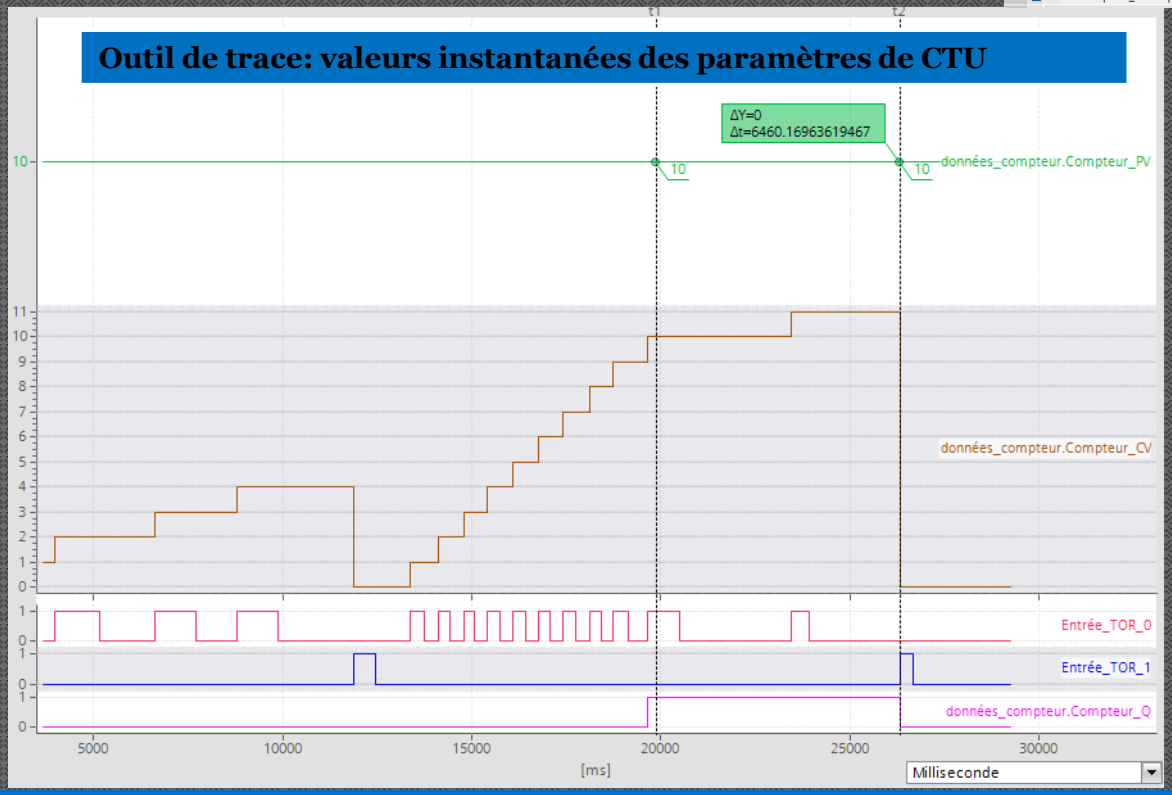


Instance CTU

données_compteur

	Nom	Type de données	Valeur de départ
1	Static		
2	Compteur	IEC_COUNTER	
3	CU	Bool	false
4	CD	Bool	false
5	R	Bool	false
6	LD	Bool	false
7	QU	Bool	false
8	QD	Bool	false
9	PV	Int	0
10	CV	Int	0
11	Decompteur	IEC_COUNTER	
12	compteur_decompteur	IEC_COUNTER	
13	Compteur_CV	Int	0
14	Compteur_PV	Int	10
15	Compteur_Q	Bool	false
16	Décompteur_CV	Int	0
17	Décompteur_PV	Int	10
18	Décompteur_Q	Bool	false
19	Compteur_décompte...	Int	10
20	Compteur_décompte...	Int	0
21	Compteur_décompte...	Bool	false
22	Compteur_décompte...	Bool	false

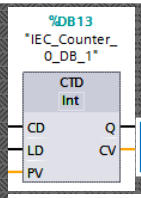
Outil de trace: valeurs instantanées des paramètres de CTU



III.7: Compteurs -Décompteurs

❑ CTD: Décompteur

SIEMENS Totally Integrated Automation PORTAL V15



5 paramètres

Paramètre	Déclaration	Type de données	Zone de mémoire		Description
			S7-1200	S7-1500	
CD	Input	BOOL	I, Q, M, D, L ou constante	I, Q, M, D, L ou constante	Entrée de comptage
LD	Input	BOOL	I, Q, M, D, L, P ou constante	I, Q, M, D, L, T, C, P ou constante	Entrée de chargement
PV	Input	Nombres entiers	I, Q, M, D, L, P ou constante	I, Q, M, D, L, P ou constante	Valeur que prend la sortie CV lorsque LD = 1.
Q	Output	BOOL	I, Q, M, D, L	I, Q, M, D, L	Etat du compteur
CV	Output	Nombres entiers, CHAR, WCHAR, DATE	I, Q, M, D, L, P	I, Q, M, D, L, P	Valeur de comptage actuelle

Instance CTD

IEC_Counter_0_DB_1

	Nom	Type de données	Valeur de départ
1	Static		
2	CU	Bool	false
3	CD	Bool	false
4	R	Bool	false
5	LD	Bool	false
6	QU	Bool	false
7	QD	Bool	false
8	PV	Int	0
9	CV	Int	0

Description :

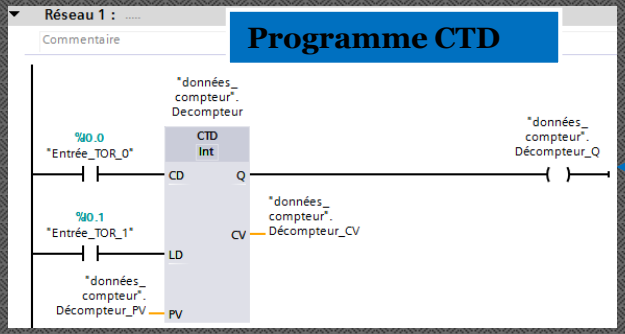
- ❑ Avec l'instruction "Décomptage", on peut compter à rebours la valeur à la sortie CV. Quand l'état logique passe de "0" à "1" (front montant) à l'entrée CD, l'instruction est exécutée et la valeur de comptage actuelle à la sortie CV est diminuée de un.
- ❑ La valeur de comptage est décrémentée chaque fois qu'un front montant est détecté, jusqu'à ce que la valeur limite inférieure du type de données spécifié soit atteinte. Une fois la valeur limite inférieure atteinte, l'état logique à l'entrée CD n'a plus d'influence sur l'instruction.
- ❑ On peut interroger l'état du compteur à la sortie Q. Quand la valeur de comptage actuelle est inférieure ou égale à zéro, la sortie Q est mise à l'état logique "1". Dans tous les autres cas, l'état logique à la sortie Q est "0".
- ❑ La valeur de la sortie CV prend la valeur du paramètre PV quand l'état logique de l'entrée LD passe à 1. Tant que l'entrée LD présente l'état logique "1", l'état logique à l'entrée CD n'a pas d'effet sur l'instruction.

III.7: Compteurs -Décompteurs

❑ CTU : exemple de programmation

SIEMENS Totally Integrated Automation PORTAL V15

Programme CTD

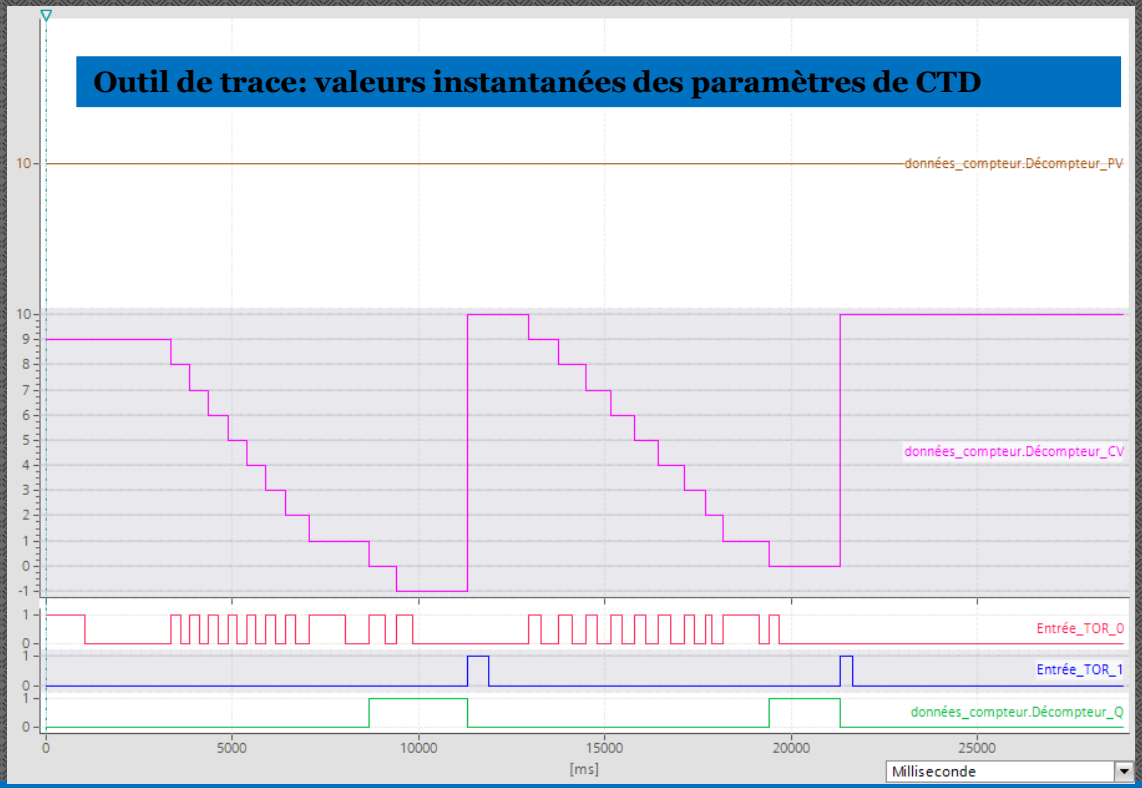


Instance CTD

données_compteur

	Nom	Type de données	Valeur de départ
1	Static		
2	Compteur	IEC_COUNTER	
3	Decompteur	IEC_COUNTER	
4	CU	Bool	false
5	CD	Bool	false
6	R	Bool	false
7	LD	Bool	false
8	QU	Bool	false
9	QD	Bool	false
10	PV	Int	0
11	CV	Int	0
12	compteur_decompteur	IEC_COUNTER	
13	Compteur_CV	Int	0
14	Compteur_PV	Int	10
15	Compteur_Q	Bool	false
16	Décompteur_CV	Int	0
17	Décompteur_PV	Int	10
18	Décompteur_Q	Bool	false
19	Compteur_décompte...	Int	10
20	Compteur_décompte...	Int	0
21	Compteur_décompte...	Bool	false
22	Compteur_décompte...	Bool	false

Outil de trace: valeurs instantanées des paramètres de CTD



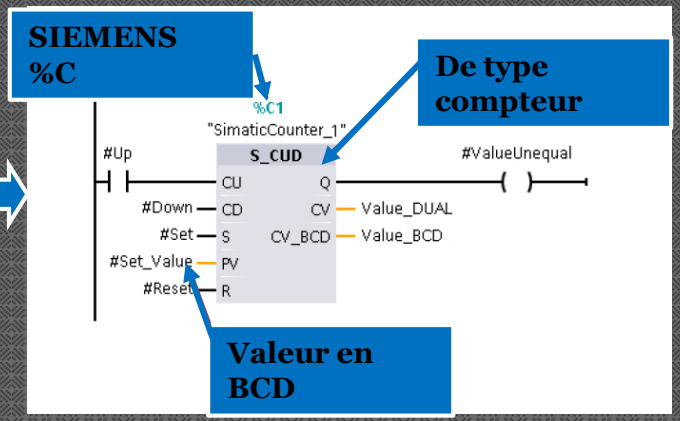
III.7: Compteurs -Décompteurs

❑ Autres compteurs/décompteurs

SIEMENS Totally Integrated Automation PORTAL V15

Ce bloc associe le CTU et le CTD

Siemens propose également ses propres compteurs/décompteurs. La variable de pré-sélection est au format BCD.



Les compteurs peuvent également être réalisés par de instructions arithmétiques : C:=C+1

III.7: Compteurs -Décompteurs

❑ Compteurs en SCL ou LIST

SIEMENS Totally Integrated Automation PORTAL V15

```
1  //Compteur-décompteur CEI
2  "IEC_Counter_0_DB_1".CTUD(CU:="S0",
3                            CD:="S1",
4                            R:="S2",
5                            LD:="S3",
6                            PV:=10,
7                            QU=>"H1",
8                            QD=>"H2",
9                            CV=>"C2");
10
```

SCL

Paramètres:
CU: entrée de comptage
CD: entrée de décomptage
R:entrée de RAZ du compteur
LD: entrée de pré-chargement
PV: valeur de pré-chargement (CV=PV si LD) =1
QU: sortie=1 si CV>=PV
QD: sortie=1 si CV<0
CV: valeur courante du compteur

```
1     CALL  CTUD , "IEC_Counter_0_DB"
2           Int
3           CU :="s0"
4           CD :="s1"
5           R  :="s2"
6           LD :="s3"
7           PV :=10
8           QU :="h1"
9           QD :="h2"
10          CV :="c45"
11
```

LIST

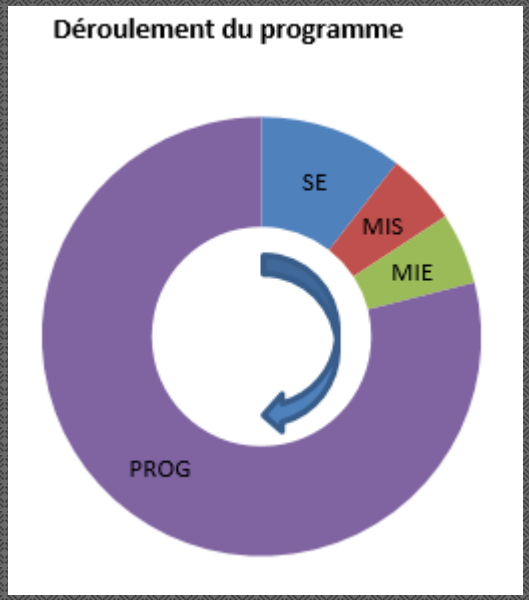
```
12  //COMPTEUR
13  //front montant sur S4
14  "Tag_39" := "S4" AND NOT "Tag_38";
15  "Tag_38" := "S4";
16  //incrémentation
17  IF "Tag_39" = 1
18  THEN
19      "C3" := "C3" + 1;
20  END_IF;
21  //RAZ compteur
22  IF "S5"
23  THEN
24      "C3" := 0;
25  END_IF;
26
```

SCL

Autre méthode

III.8: Structures des programmes

❑ Rappel du cycle API



Une fois que les tâches internes du **S**ystème d'**E**xploitation (**SE**) ont été réalisées, la **M**émoire **I**mage des **S**orties (**MIS**) est inscrite dans les sorties des modules et l'état des entrées et lu dans la **M**émoire **I**mage des **E**ntrées (**MIE**).

Puis a lieu le traitement du programme utilisateur avec tous les blocs qui y sont appelés.

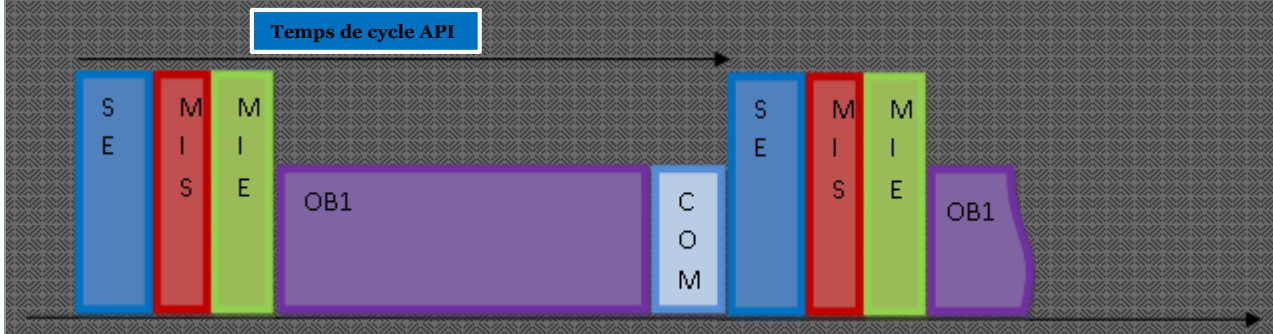
L'écriture de la mémoire image des sorties (MIS) sur les sorties des modules et la lecture de la mémoire image des entrées sont réalisées automatiquement par le système d'exploitation.

Avantages de la mémoire image du processus
L'accès à la mémoire image du processus offre, par rapport à l'accès direct aux modules d'entrées et de sorties, l'avantage que la CPU dispose d'une mémoire image des signaux du processus cohérente pendant la durée du traitement de programme cyclique. Si un état de signal change sur un module d'entrées pendant le traitement du programme, l'état de signal dans la mémoire image est conservé jusqu'à la mise à jour de la mémoire image du processus dans le cycle suivant. L'interrogation répétée d'un signal d'entrée dans un programme utilisateur permet de garantir la cohérence de l'information d'entrée.

III.8: Structures des programmes

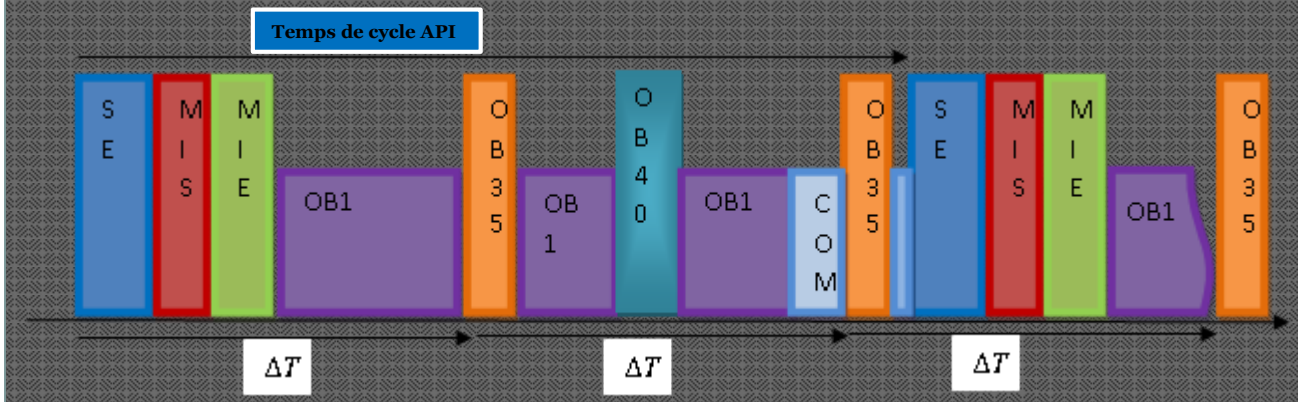
❑ Exemple de déroulement d'un programme sans interruption

SIEMENS Totally Integrated Automation PORTAL V15

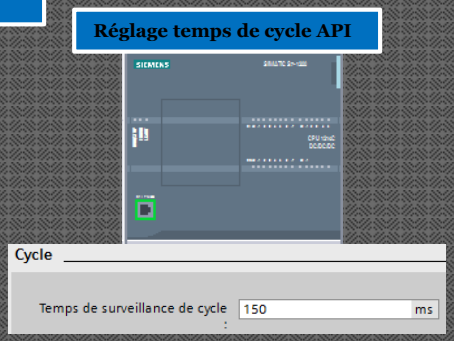


OB1 : traitement cyclique
MIE: Mémoire image entrées
MIS: Mémoire image sorties
SE: Système Exploitation
COM: Communication
OB35: Interruption cyclique
OB40:Interruption Hardware

❑ Exemple de déroulement d'un programme avec interruption



Réglage temps de cycle API



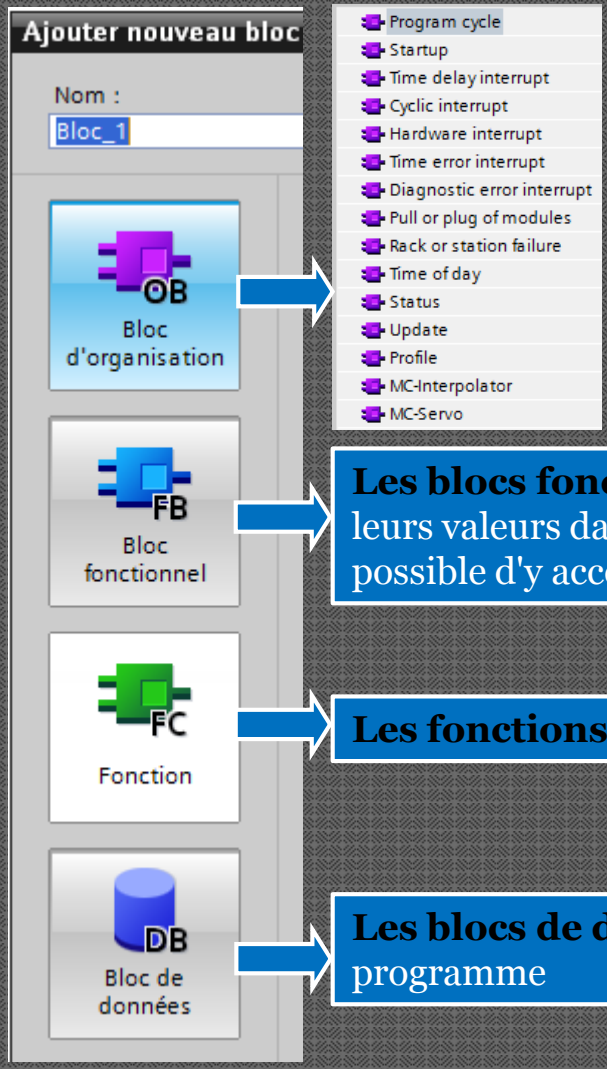
L'OB1 est interrompu par l'OB35, OB cyclique de période ΔT et par l'OB40 pour une alarme (le temps de cycle est rallongé dans ce cas de la durée de traitement de 2 fois la durée de l'OB35 et de la durée de l'OB 40). On rappelle les différents OB de programmation (voir transparent 50)

Si le temps de cycle est supérieur à la valeur indiquée dans la case temps de surveillance du cycle dans la configuration, la CPU passe à l'arrêt

III.8: Structures des programmes

❑ Eléments constitutifs d'un programme

SIEMENS Totally Integrated Automation PORTAL V15



Les blocs d'organisation diffèrent selon le modèle de CPU. Un programme dispose toujours d'un OB1 (program cycle)

Les blocs fonctionnels sont des blocs de code qui sauvegardent en permanence leurs valeurs dans **des blocs de données d'instance (DBI)** afin qu'il soit possible d'y accéder même après le traitement du bloc.

Les fonctions sont des blocs de code sans mémoire.

Les blocs de données globales (DBG) servent à sauvegarder les données du programme

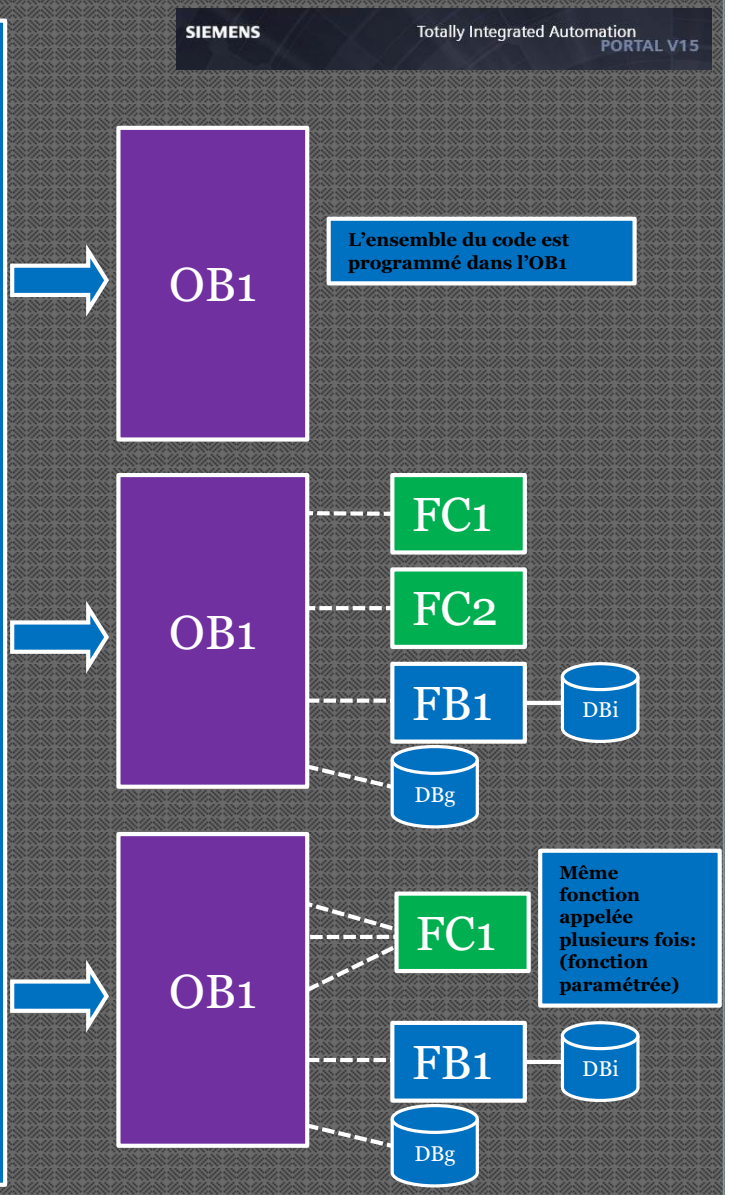
III.8: Structures des programmes

Il existe plusieurs types de programmation :

- **La programmation linéaire** : C'est une suite de codes inscrite dans un bloc OB1.Le programme est une suite d'instructions en langage LIST, de réseaux en échelles pour le langage à contact CONT ou de logigrammes dans le langage LOG. Ce type de programmation est à proscrire car il ne permet pas de différencier les différentes parties d'un programme ou uniquement par commentaires.

- **La programmation segmentée :** Dans ce cas le programme est divisée en blocs qui peuvent correspondre à une tache ou une unité dans une machine. Ces blocs doivent être appelés dans l'OB1, l'ordre d'appel définit l'ordre d'exécution. La lisibilité du programme est accrue et un des avantages est que pour les dépannages ou les mises au point, la navigation ne s'effectue que dans la partie concernée.

- **La programmation structurée :** C'est une programmation segmentée qui utilise en plus des blocs paramétrables. Lors de l'appel d'un bloc qui possède des paramètres, il faut les renseigner en affectant des opérandes globaux.

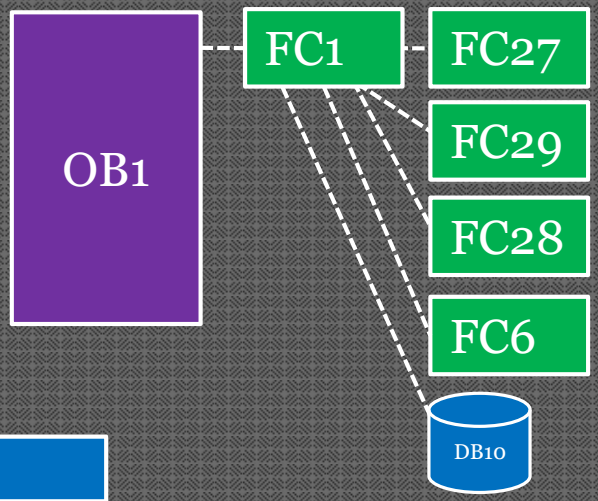


III.8: Structures des programmes

❑ Exemple de structure d'une partie d'un programme

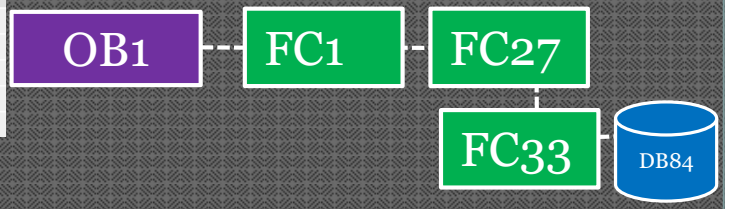
SIEMENS Totally Integrated Automation PORTAL V15

Main	OB1	
CE_APP_2W	FC1	Main Ré9 (———— Appel bloc CE APP ————)
CE_APP_2X	FC27	CE_APP_2W Ré10 (Appels des blocs combinatoire d'entrée
CE_APP_3X	FC29	CE_APP_2W Ré10 (Appels des blocs combinatoire d'entrée
CE_APP_5X	FC28	CE_APP_2W Ré10 (Appels des blocs combinatoire d'entrée
CE_APP_8X	FC6	CE_APP_2W Ré10 (Appels des blocs combinatoire d'entrée
DG_2W	DB10	CE_APP_2W Ré7 (BIT ARRET D'URGENCE)
DG_2W	DB10	CE_APP_2W Ré4 (MISE EN SERVICE GROUPE D'ENERGIE PNE
DG_2W	DB10	CE_APP_2W Ré5 (MISE EN SERVICE GROUPE D'ENERGIE PNE
DG_2W	DB10	CE_APP_2W Ré2 (MISE EN SERVICE GROUPE DE SECURITE)

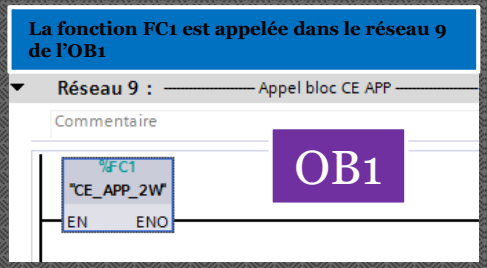


❑ Détails de FC27

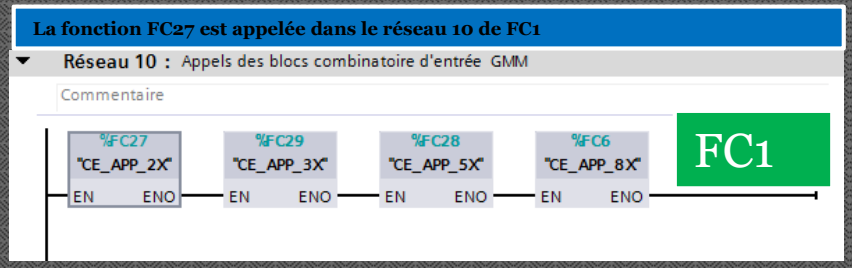
Main	OB1	
CE_APP_2W	FC1	Main Ré9 (———— Appel bloc CE APP ————)
CE_APP_2X	FC27	CE_APP_2W Ré10 (Appels des blocs combinatoire d'entrée GMM)
CE_APP_26	FC33	CE_APP_2X Ré9 (Appel CE Application Entité)
DG_2X	DB84	CE_APP_2X Ré12 (MANU => Fontion validée)
DG_2X	DB84	CE_APP_2X Ré7 (BIT ARRET D'URGENCE)



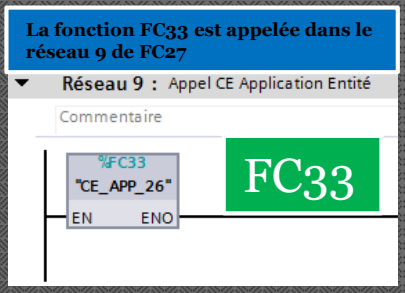
La fonction FC1 est appelée dans le réseau 9 de l'OB1



La fonction FC27 est appelée dans le réseau 10 de FC1



La fonction FC33 est appelée dans le réseau 9 de FC27



III.8: Structures des programmes

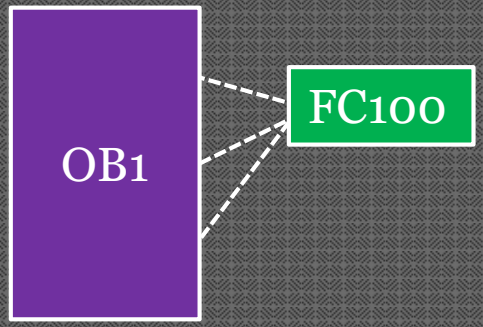
❑ Exemple d'une fonction paramétrée

SIEMENS Totally Integrated Automation PORTAL V15

Nous avons vu (transparent 210) que le programmation structurée nécessitait l'utilisation de fonctions ou de blocs fonctionnels paramétrés.

Cette méthode de programmation permet de créer des fonctions ou des blocs fonctionnels entièrement paramétrables.

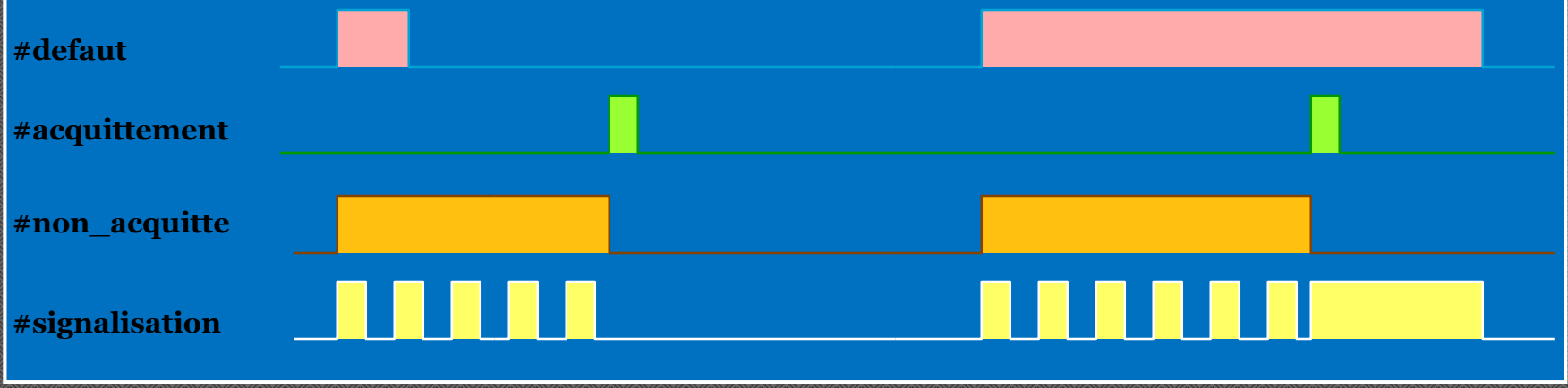
Très utile lorsque la fonction est répétée plusieurs fois dans le même programme

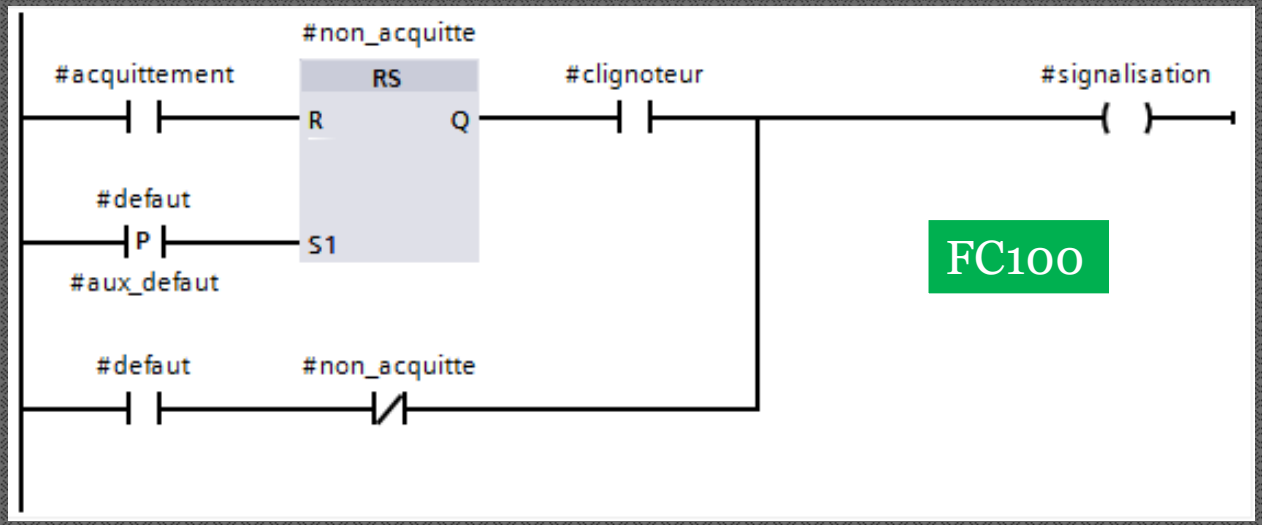


Pour illustrer, l'utilisation des fonctions paramétrées, nous prendrons comme exemple la fonction FC100.

Cette fonction permet de gérer des défauts d'une machine. Pour notre cas trois types de défaut peuvent survenir, c'est pour cette raison que la fonction est appelée trois fois dans l'OB1

Chronogramme d'un défaut



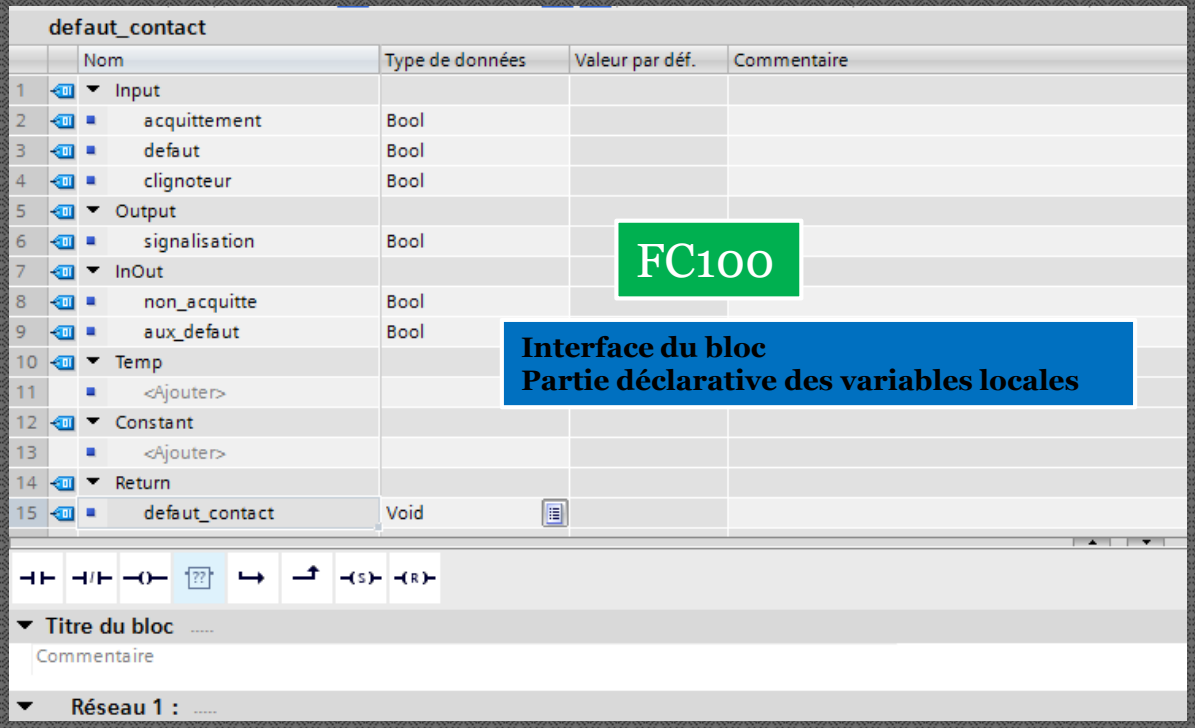


III.8: Structures des programmes

❑ Exemple d'une fonction paramétrée

SIEMENS Totally Integrated Automation PORTAL V15

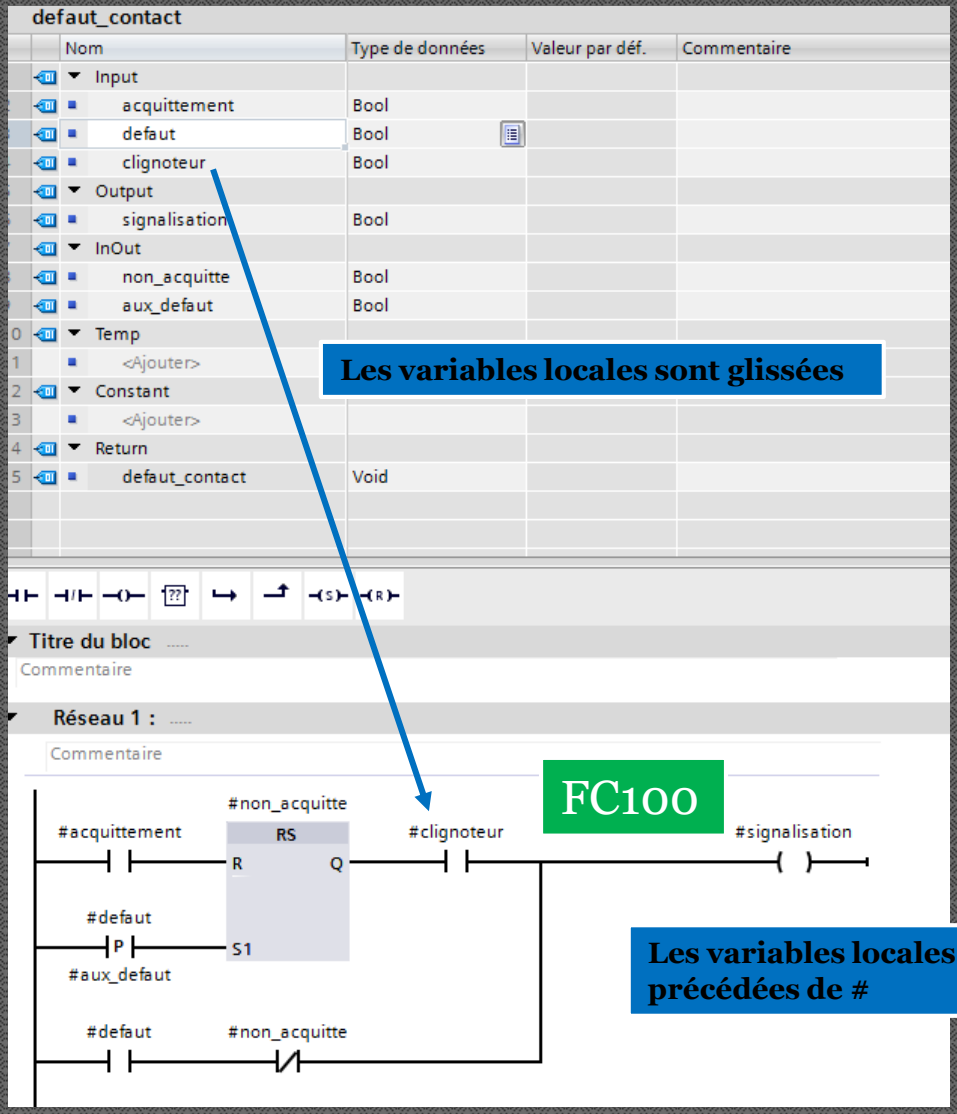
Type de paramètre	Déclaration	Utilisation
Paramètre entrant	Paramètre entrant	lecture
Paramètre sortant	Paramètre sortant	écriture
Paramètre entrant et sortant	Paramètre entrant et sortant	Lecture t écriture



III.8: Structures des programmes

❑ Exemple d'une fonction paramétrée

SIEMENS Totally Integrated Automation PORTAL V15

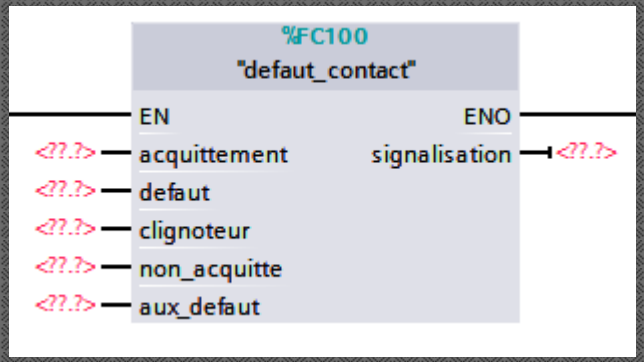


III.8: Structures des programmes

❑ Exemple d'une fonction paramétrée

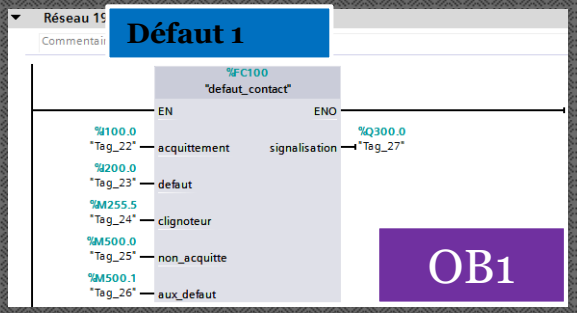
SIEMENS Totally Integrated Automation PORTAL V15

OB1 → FC100



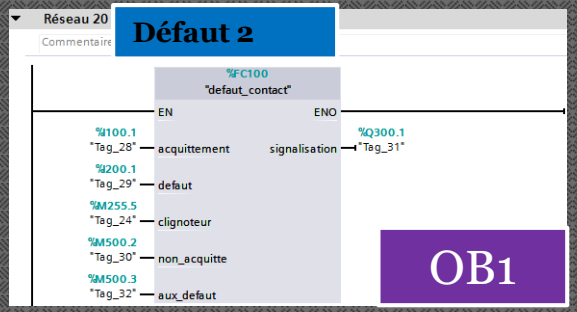
Il faut maintenant associé pour les trois types de défaut les variables globales (E/S et M) de l'API

Défaut 1



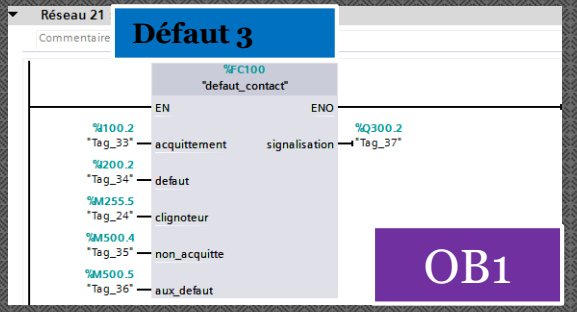
OB1

Défaut 2



OB1

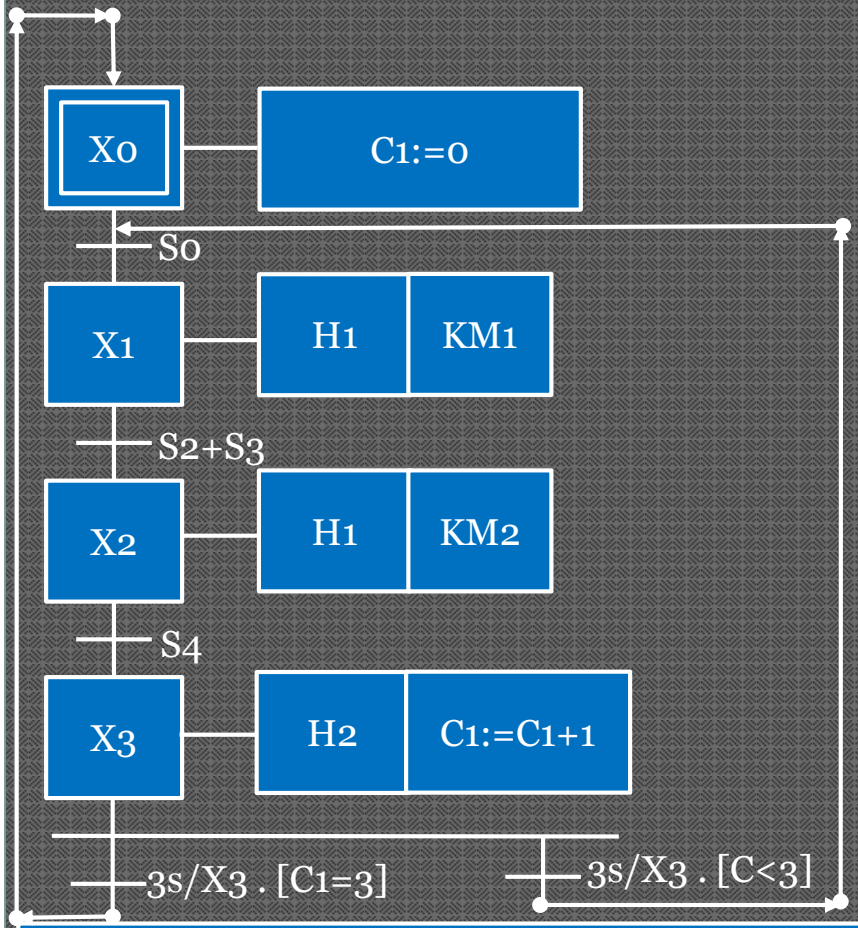
Défaut 3



OB1

Si d'autres défauts à gérer, on peut utiliser la même fonction.

III.9: Programmer un GRAFCET en langages normalisés CEI61131

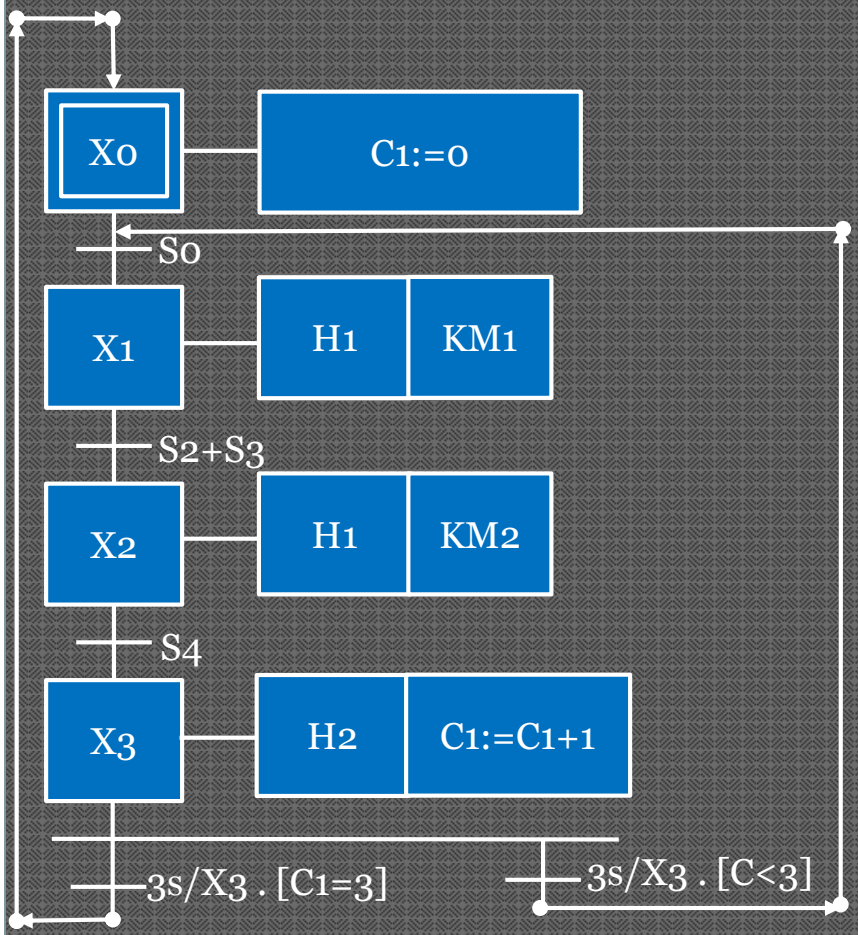


Pour cette partie de cours, nous prendrons un GRAFCET que nous coderons dans 4 langages :

- **LADDER**
- **LOGIGRAMME**
- **SCL**
- **LIST**

Remarque: Les méthodes de programmation de GRAFCET proposées dans les transparents suivants sont utilisées par des industriels. Néanmoins, il existe d'autres solutions pour arriver au même résultat. L'automaticien doit savoir s'adapter en fonction des méthodes de programmation imposées par son entreprise ou demandées par le client.

III.9: Programmer un GRAFCET en langages normalisés CEI61131



La méthode de travail consiste dans un premier temps à écrire les équations d'activation et de désactivation des étapes en respectant les règles d'évolution du GRACFET.
On notera l'activation d'une étape par Xi(1) et la désactivation par Xi(0), i désignant le numéro de l'étape.

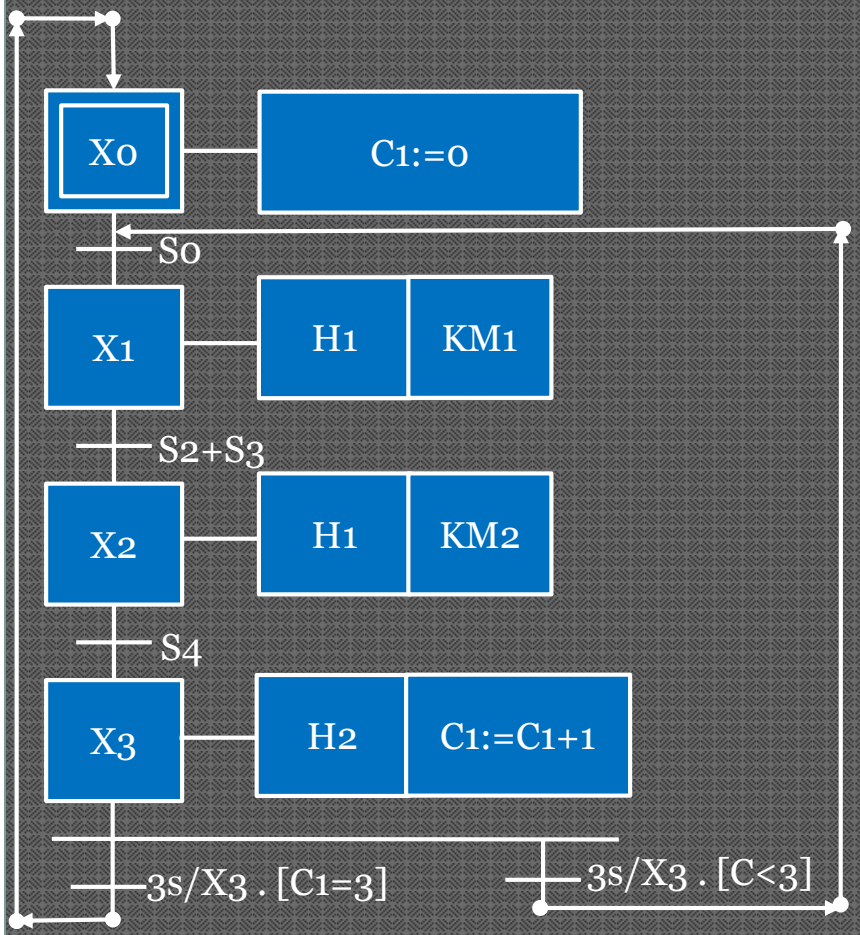
Activation et désactivation de X0 :
X0(1) = X3 . (3s/X3) . [C1=3]
X0(0)= X1

Activation et désactivation de X1 :
X1(1) = (X3 . (3s/X3) . [C1<3]) + (X0 . S0)
X1(0)= X2

Activation et désactivation de X2 :
X2(1) = X1 . (S2 + S3)
X2(0)= X3

Activation et désactivation de X3 :
X3(1) = X2 . S4
X3(0)= X0 + X1

III.9: Programmer un GRAFCET en langages normalisés CEI61131

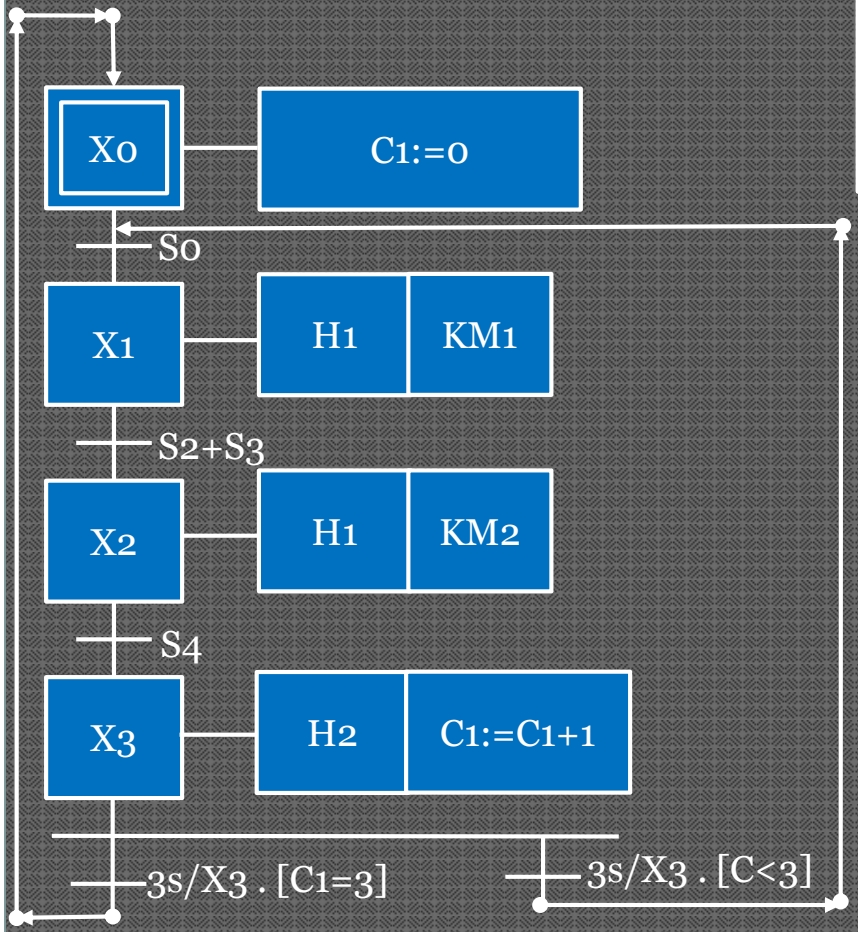


La seconde phase consiste à écrire les équations des actions.
Attention, dans un programme séquentiel, lorsqu'une action est validé dans plusieurs étapes, il convient de paralléliser les équations (OU logique) au risque de dysfonctionnement.
Deux types opérations : binaire ou numérique
Opérations binaires :H1, KM1, KM2 et H2
Opérations numériques : Compteur C1 (CEI) et temporisation T1 (CEI), le compteur peut être également traité par opération arithmétique.

Programme des actions:
Opérations binaires :
KM1= X1(1)
KM2= X2(1)
H1= X1(1) + X2(1)
H2= X3(1)
Opérations numériques :
Affecter 0 au compteur C1 sur X0
Incrémenter de 1 le compteur C1 sur chaque front montant de X3
Démarrer temporisation T1 sur X3

III.9: Programmer un GRAFCET en langages normalisés CEI61131

SIEMENS Totally Integrated Automation PORTAL V15



L'état d'une étape (0-1) peut être programmé sur l' état booléen d'un bit, mais on peut également travailler sur une valeur entière d'un mot simple (16bits) ou d'un mot double (32bits)

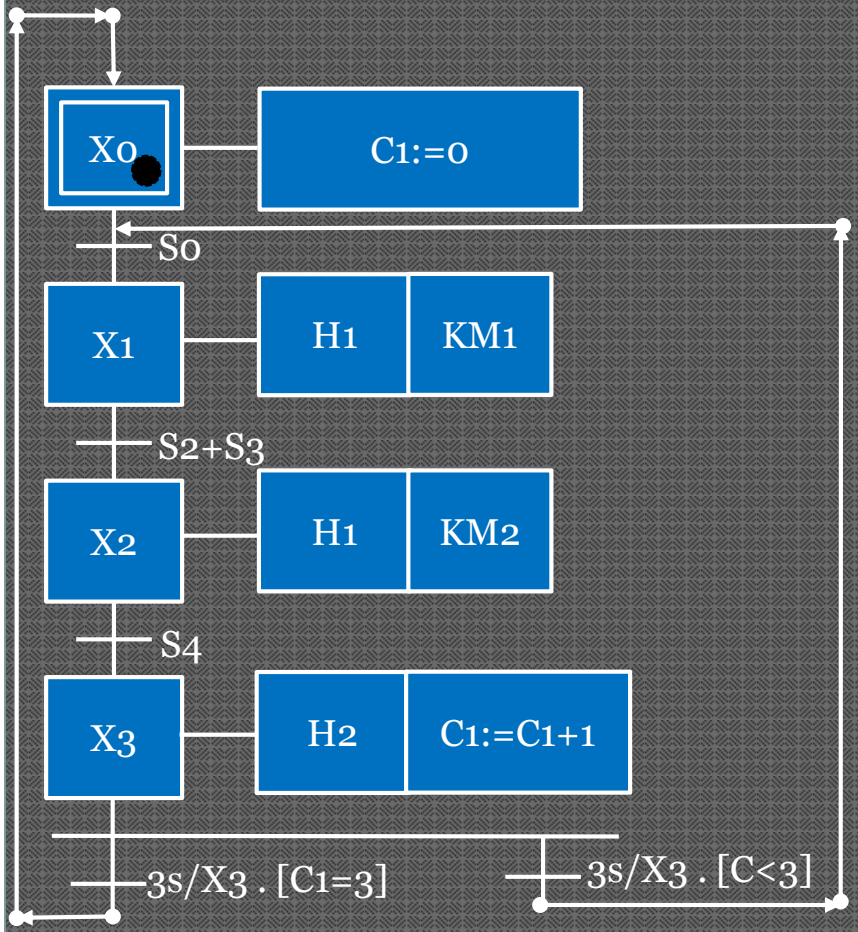
Utilisation d'un bit :
L'étape X0 affecté à %M605.0
Si X0 activée alors %M605.0 = 1
Si X0 désactivée alors %M605.0 = 0
X1: %M605.1, X2:%M605.2, etc.

Utilisation d'un MOT :
L'ensemble des étapes est associé à un mot :%MW604
Si %MW604 =0 alors X0 activée
Si %MW604 <>0 alors X0 désactivée

- si %MW604 =1 alors X1(1)
- si %MW604 =2 alors X2(1)
- si %MW604 =3 alors X3(1)

III.9: Programmer un GRAFCET en langages normalisés CEI61131

SIEMENS Totally Integrated Automation PORTAL V15



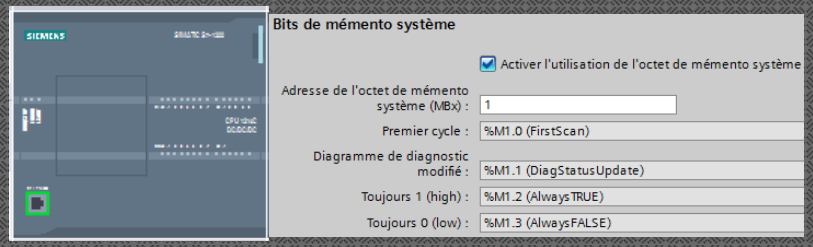
Le respect des règles d'évolution du GRAFCET nous impose également la position du jeton sur l'étape initiale au chargement du programme dans l'API (règle n°1).
Si on reprend l'équation d'activation de l'étape initiale il convient d'ajouter cette contrainte.

Deux solutions:

- Coder l'initialisation dans un OB de démarrage (OB100), c'est un bloc particulier appelé par la CPU uniquement sur le premier cycle API.

Ou

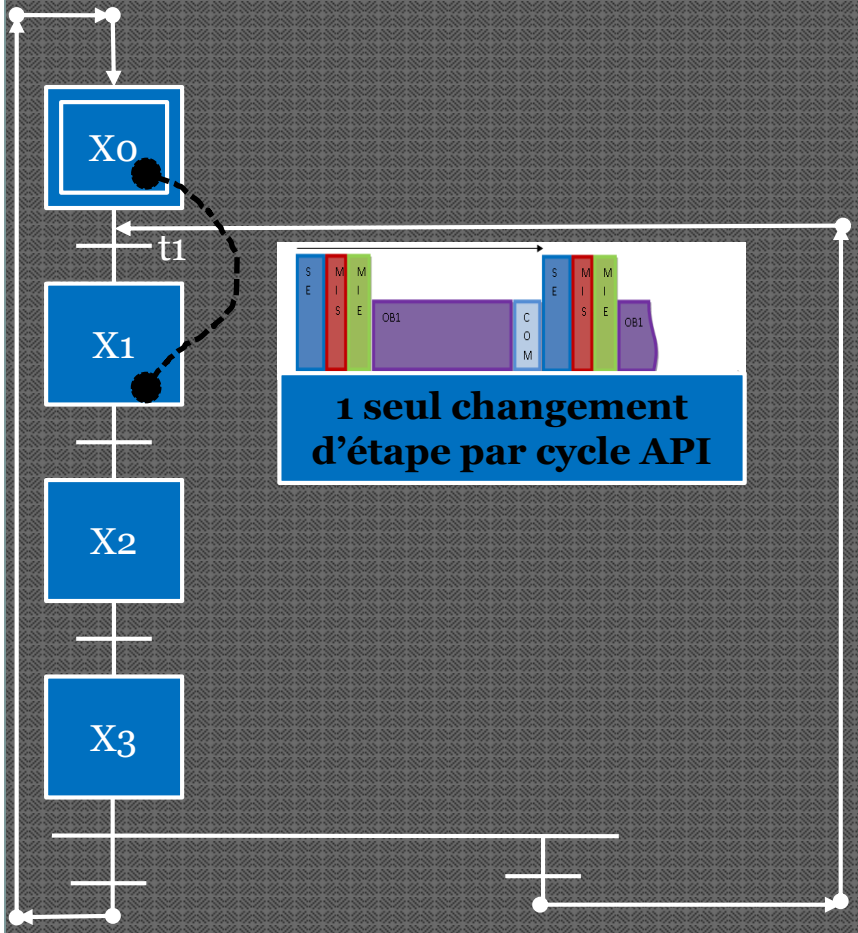
- Utiliser un bit système CPU appelé FirstScan (équivalent OB100), ce bit système doit néanmoins être déclaré dans les propriétés de la CPU



Modification de l'activation de X0 :
X0(1) = X3 . (3s/X3) . [C1=3] + FirstScan

III.9: Programmer un GRAFCET en langages normalisés CEI61131

SIEMENS Totally Integrated Automation PORTAL V15



L'évolution des étapes du GRAFCET impose un seul changement d'étape par cycle API.
Cette contrainte impose également d'adapter la programmation pour respecter cette règle.

Solution différente selon le langage utilisé:

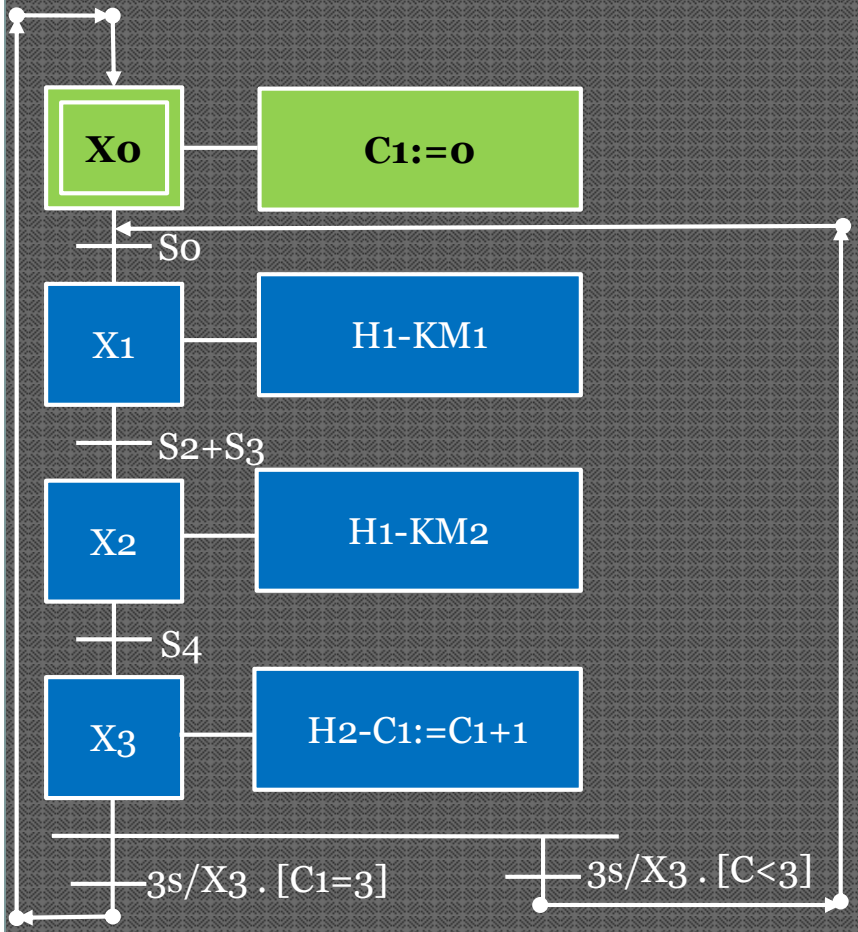
- En langage à contact ou en logigramme on peut utiliser un bit mémoire bloquant l'évolution de deux étapes ou plus sur le même cycle API.
  - Variable booléenne : « pass »
    « pass » est mis à 1 lors d'une évolution du GRAFCET(changement d'étape) et remis à zéro en fin de programmation.

Exemple:
Mise à 1 de « pass » si : X0(1) . t1 . /pass
Mise à 0 de « pass » à chaque fin de cycle API

- En SCL et en LIST on peut travailler sur deux variables représentant l'état d'une étape sur le cycle n et sur le cycle n+1
  - Variable cycle n: etat_courant
  - Variable cycle n + 1 : etat_futur

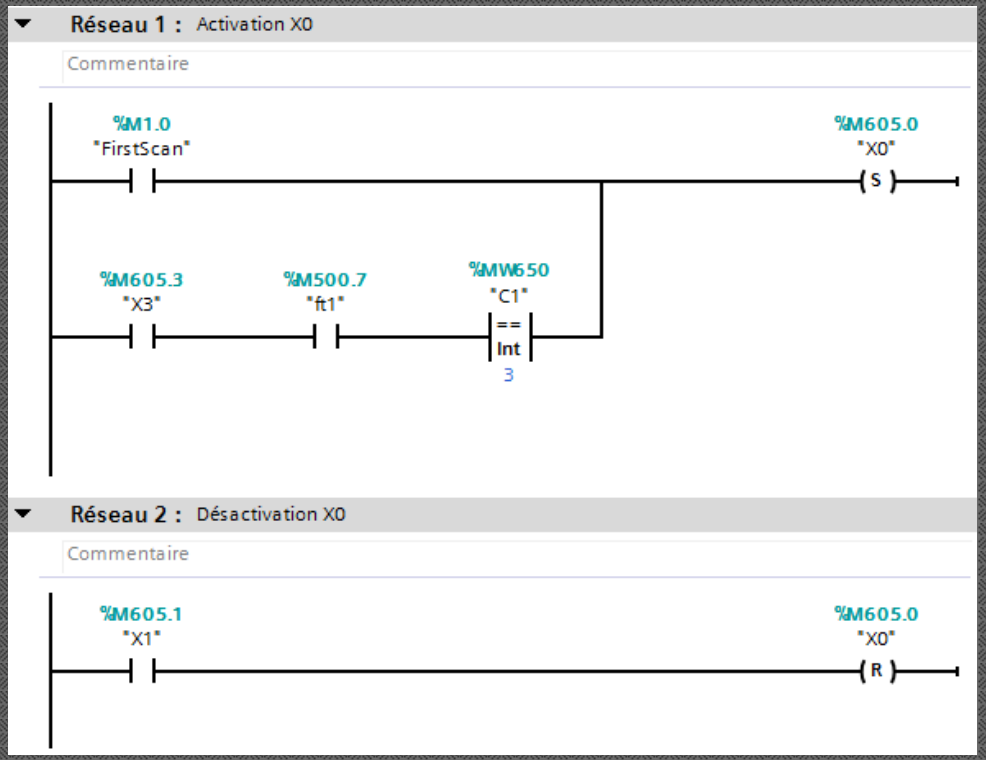
III.9: Programmer un GRAFCET en langages normalisés CEI61131

SIEMENS Totally Integrated Automation PORTAL V15



Programme du GRAFCET en langage à contact

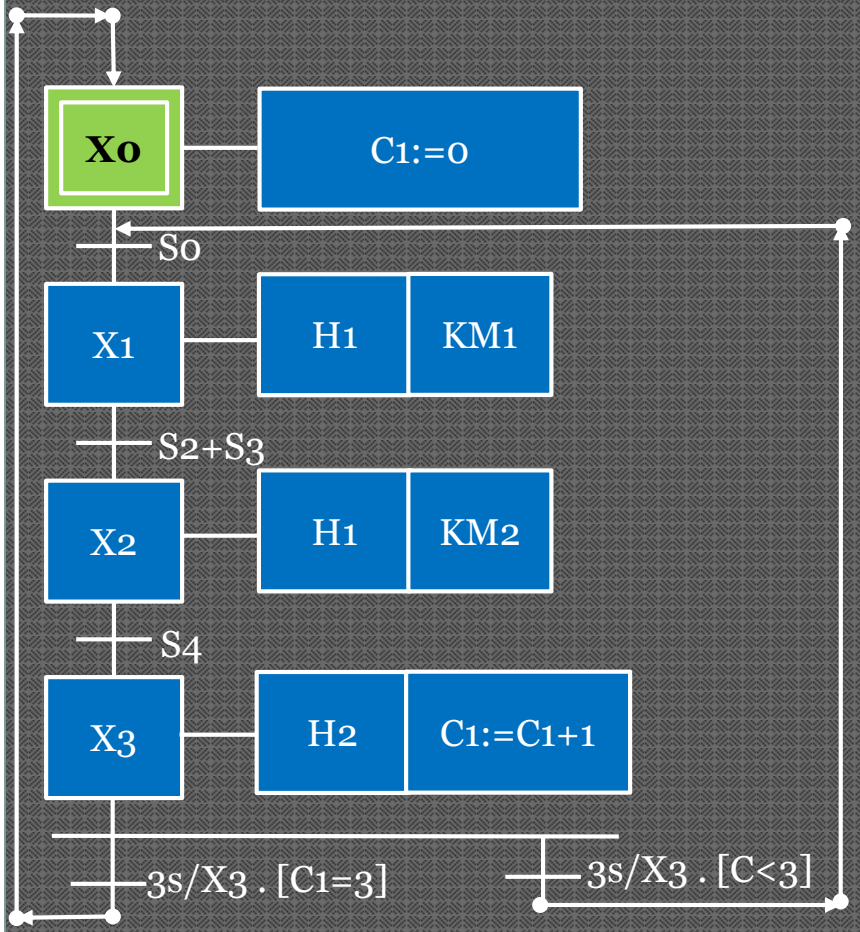
Activation et désactivation de X0 :
X0(1) = X3 . (3s/X3) . [C1=3] + FirstScan
X0(0)= X1



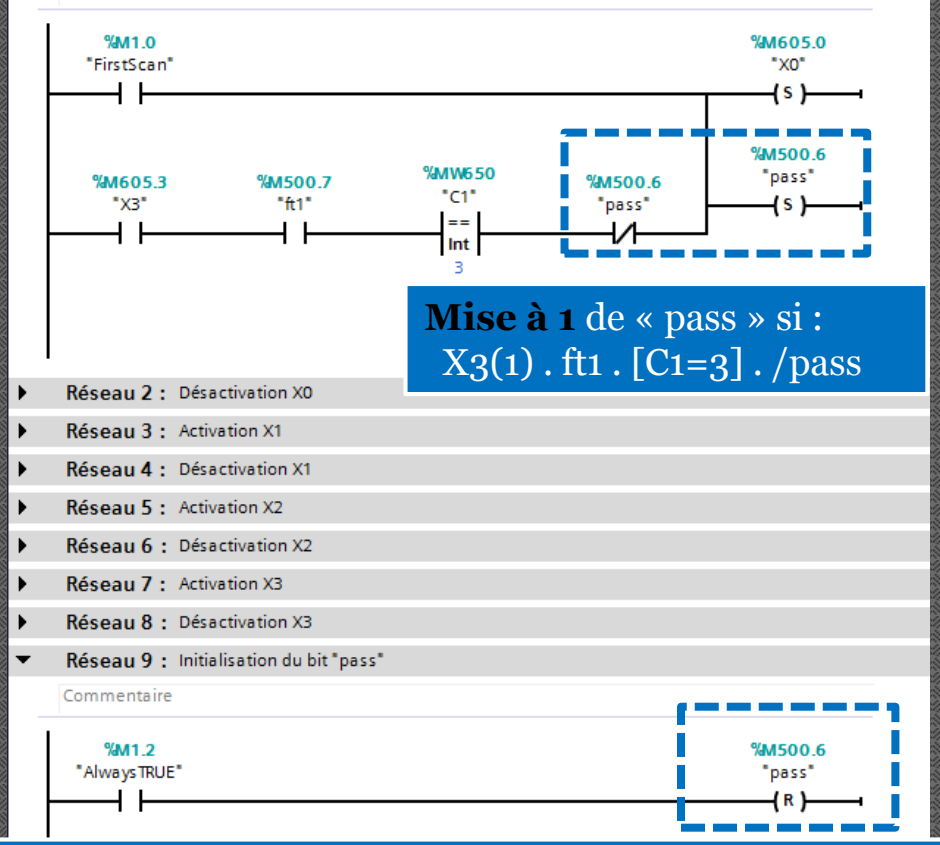
III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en langage à contact

SIEMENS Totally Integrated Automation PORTAL V15



Intégration du bit « pass »



Mise à 1 de « pass » si :
X3(1) . ft1 . [C1=3] . /pass

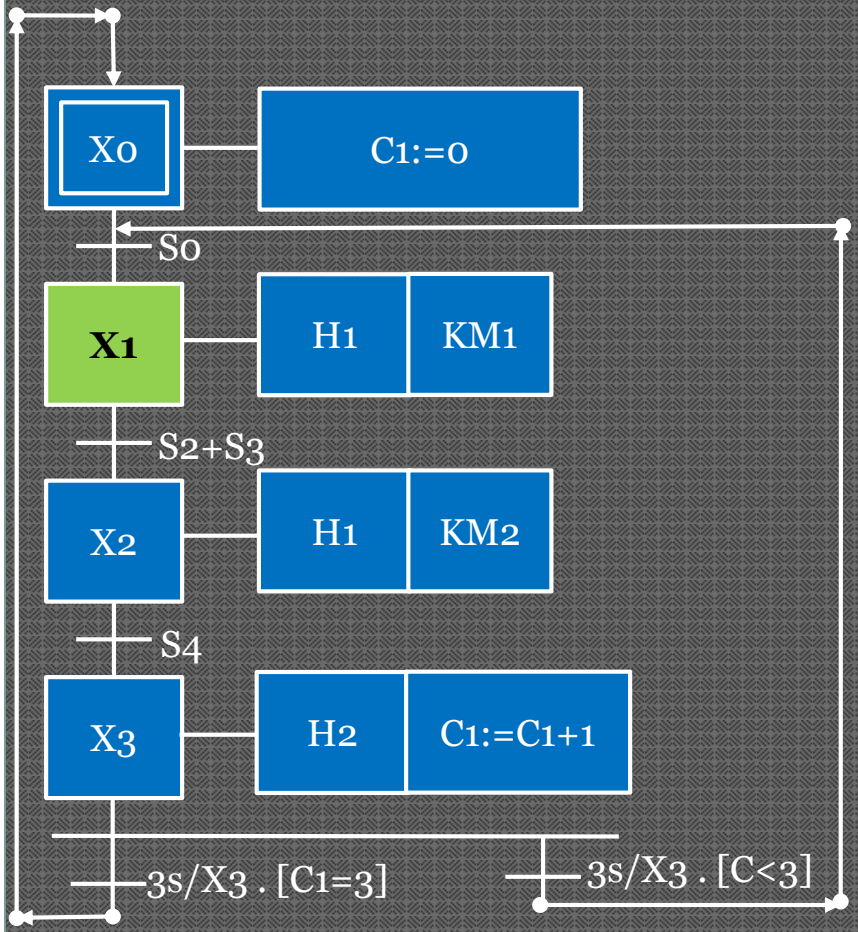
Mise à 0 de « pass » sur le réseau 9 garantissant un seul changement d'étape par tour de cycle API

III.9: Programmer un GRAFCET en langages normalisés

CEI61131

Programme du GRAFCET en langage à contact

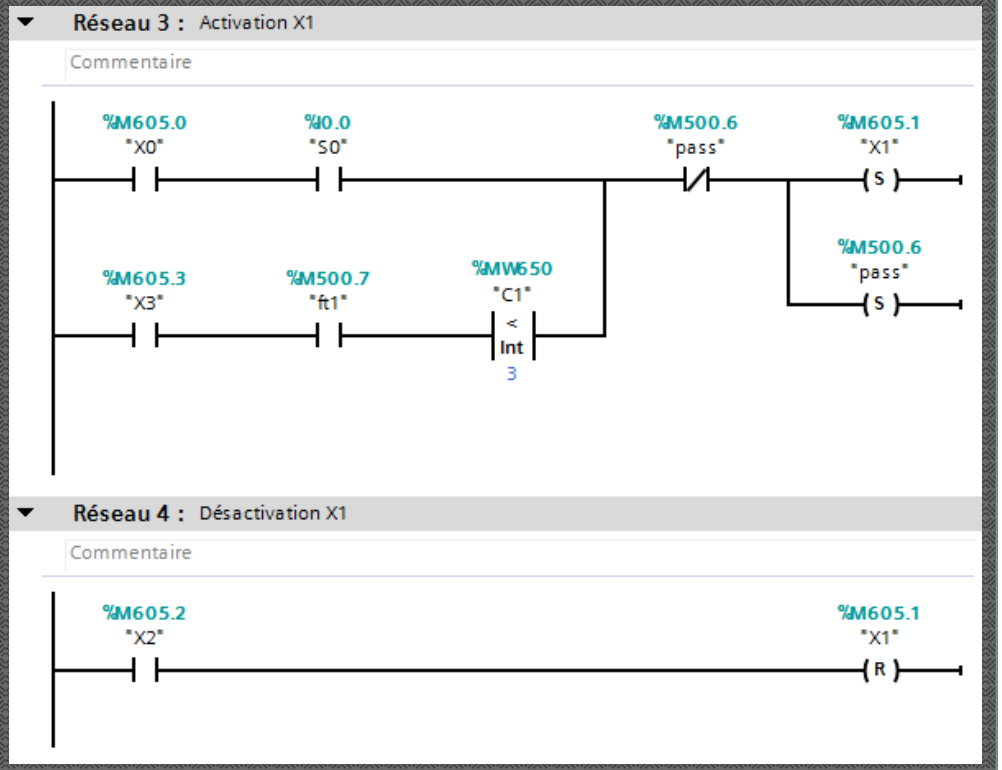
SIEMENS Totally Integrated Automation PORTAL V15



Activation et désactivation de X1 :

X1(1) = (X3 . (3s/X3) . [C1<3]) + (X0 . S0)

X1(0)= X2

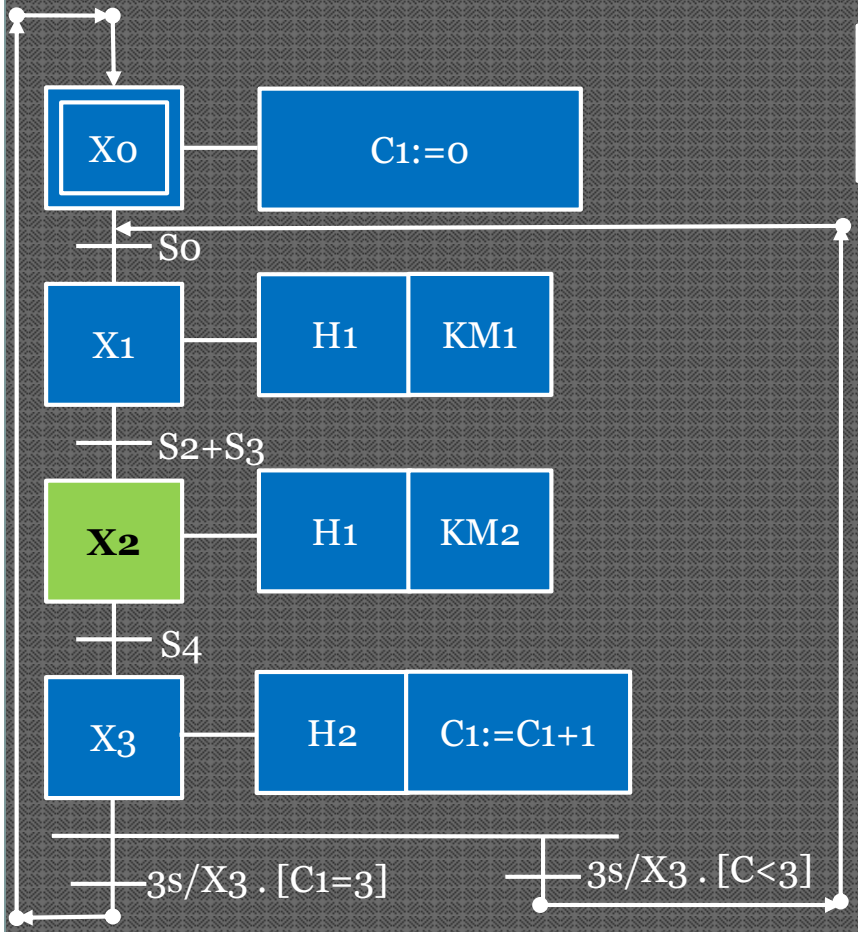


III.9: Programmer un GRAFCET en langages normalisés

CEI61131

Programme du GRAFCET en langage à contact

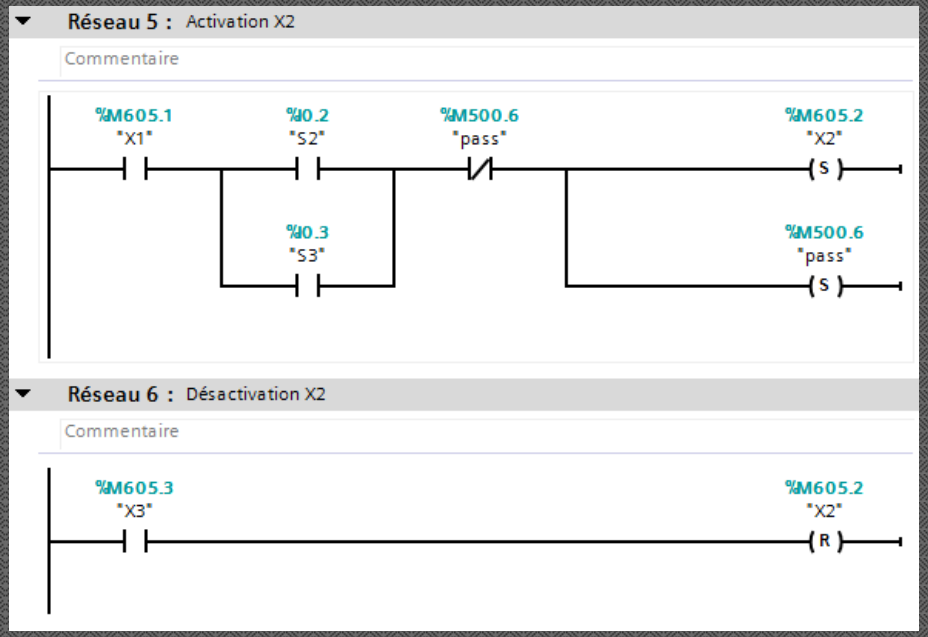
SIEMENS Totally Integrated Automation PORTAL V15



Activation et désactivation de X2 :

X2(1) = X1 . (S2 + S3)

X2(0)= X3

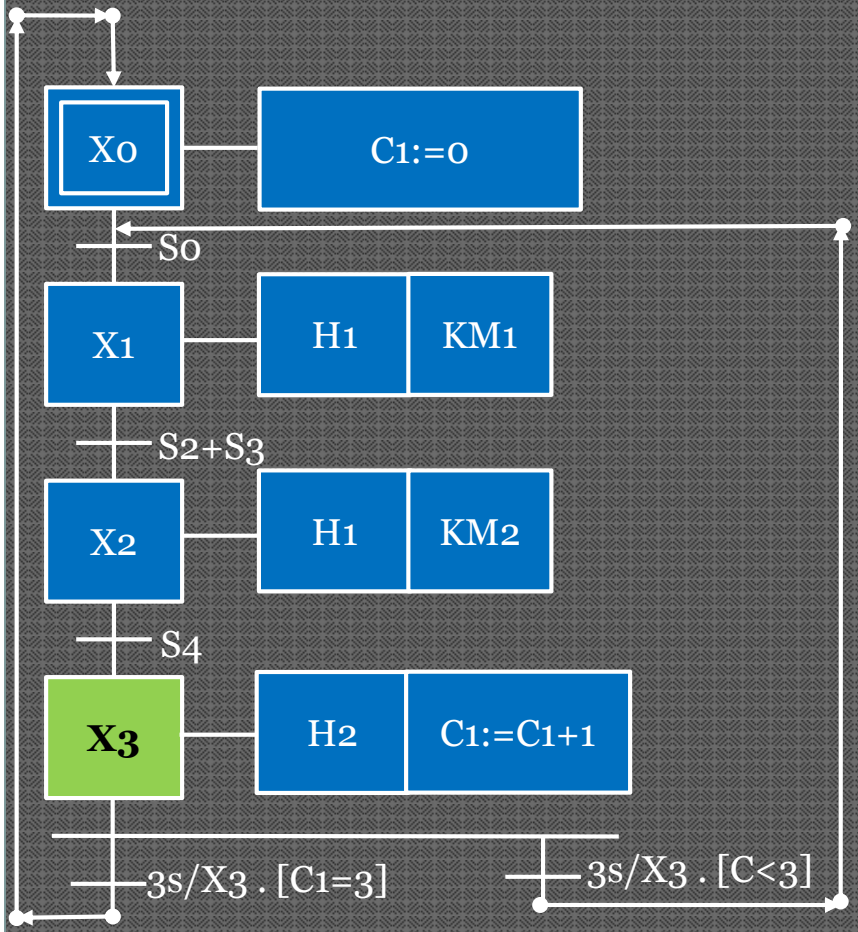


III.9: Programmer un GRAFCET en langages normalisés

CEI61131

Programme du GRAFCET en langage à contact

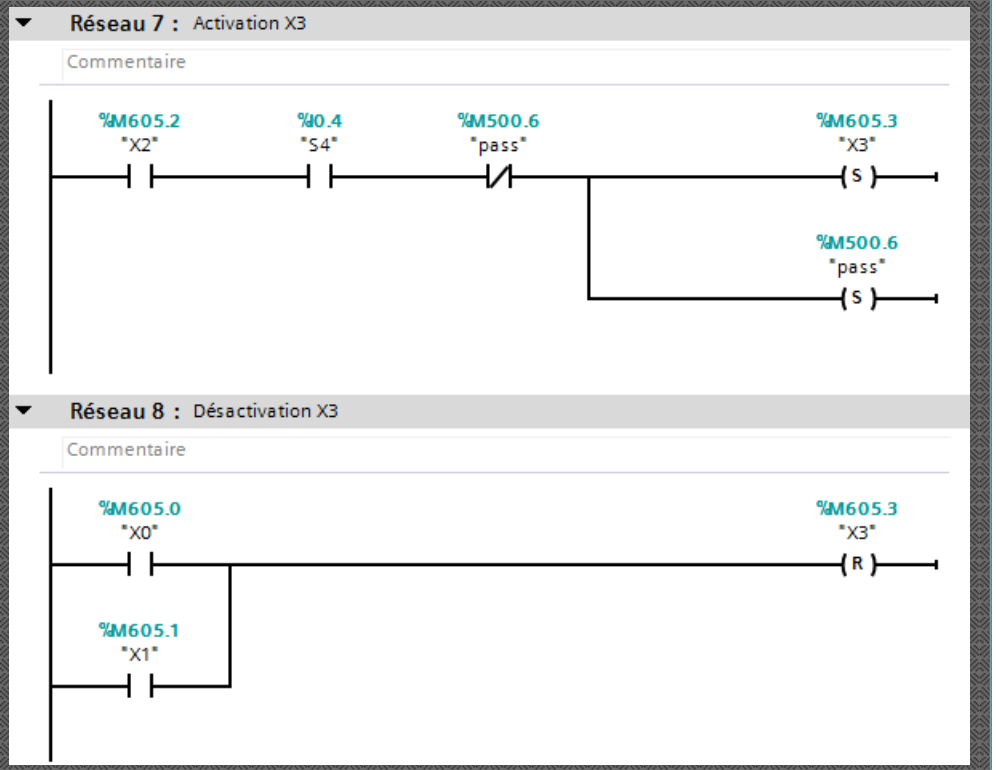
SIEMENS Totally Integrated Automation PORTAL V15



Activation et désactivation de X3 :

X3(1) = X2 . S4

X3(0)= X0 + X1

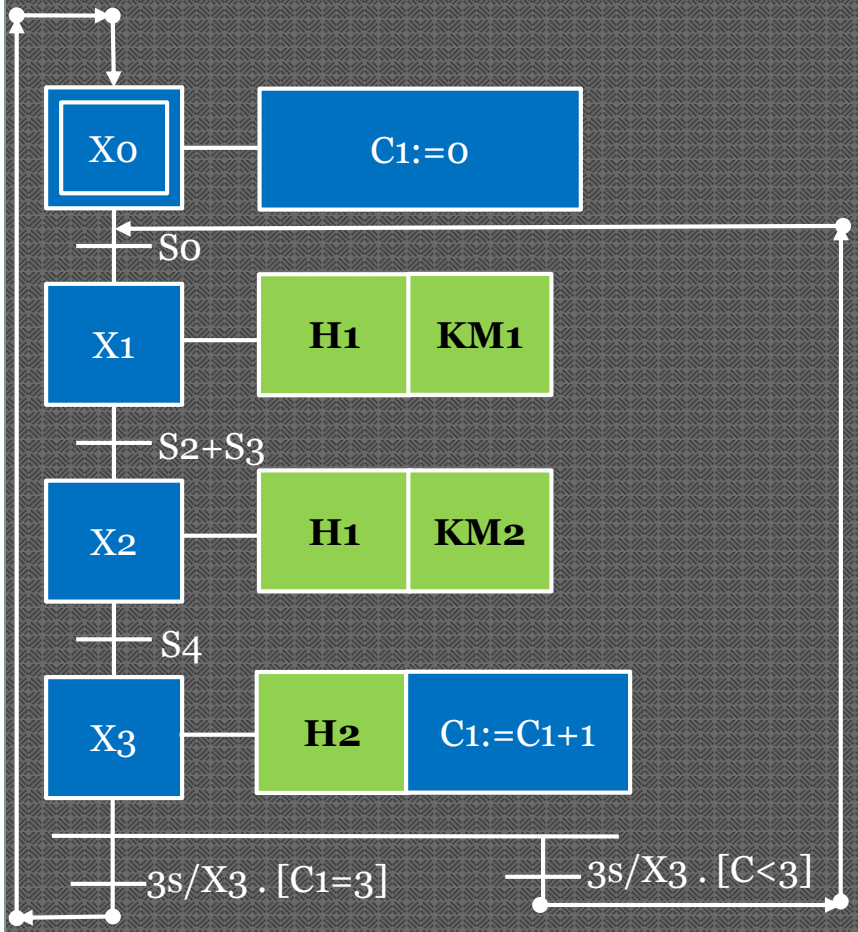


III.9: Programmer un GRAFCET en langages normalisés

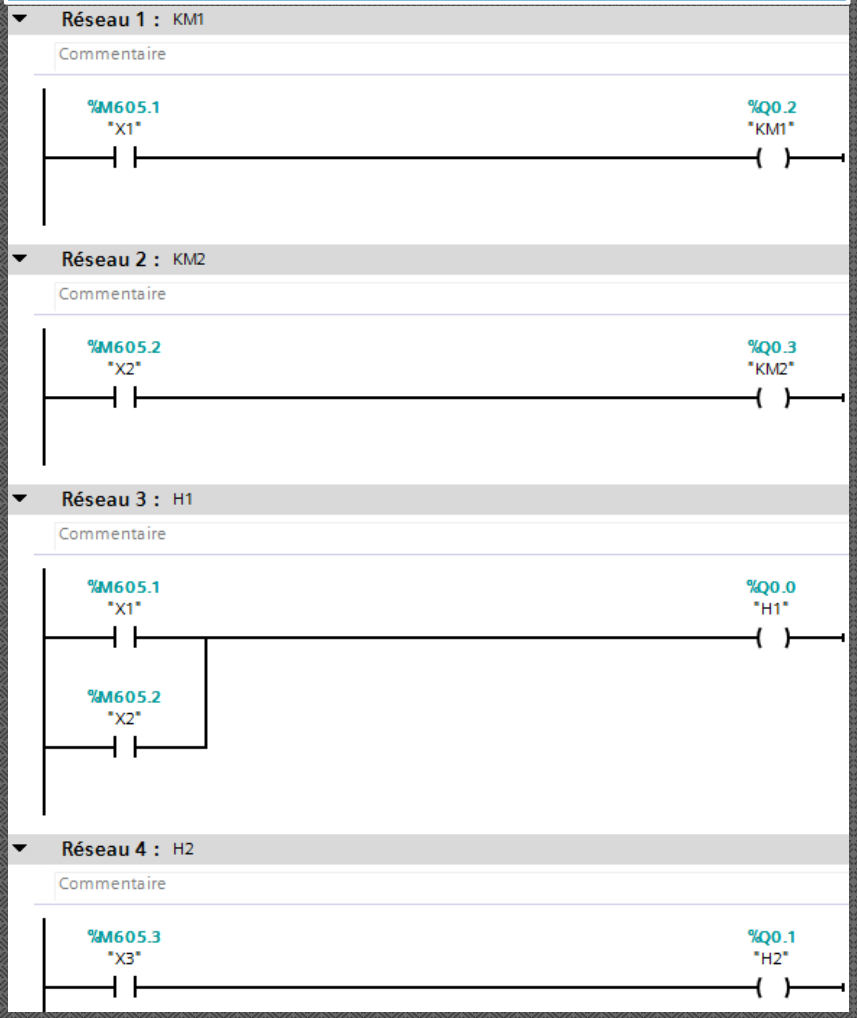
CEI61131

Programme du GRAFCET en langage à contact

SIEMENS Totally Integrated Automation PORTAL V15



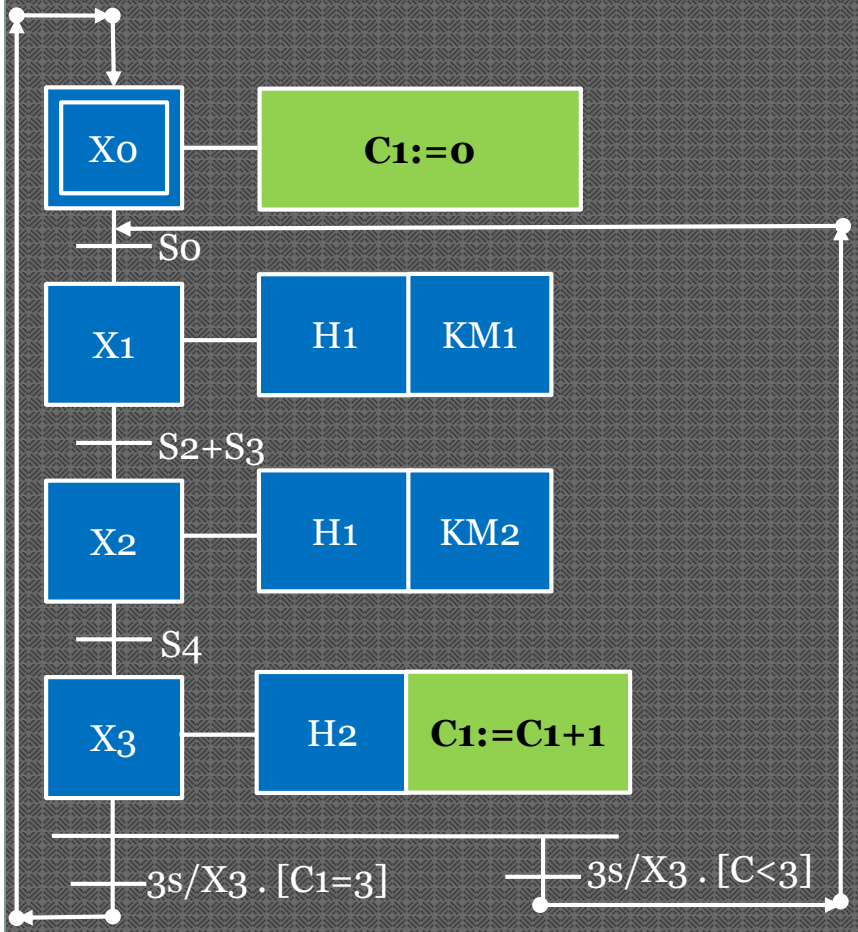
Actions: Opérations binaires



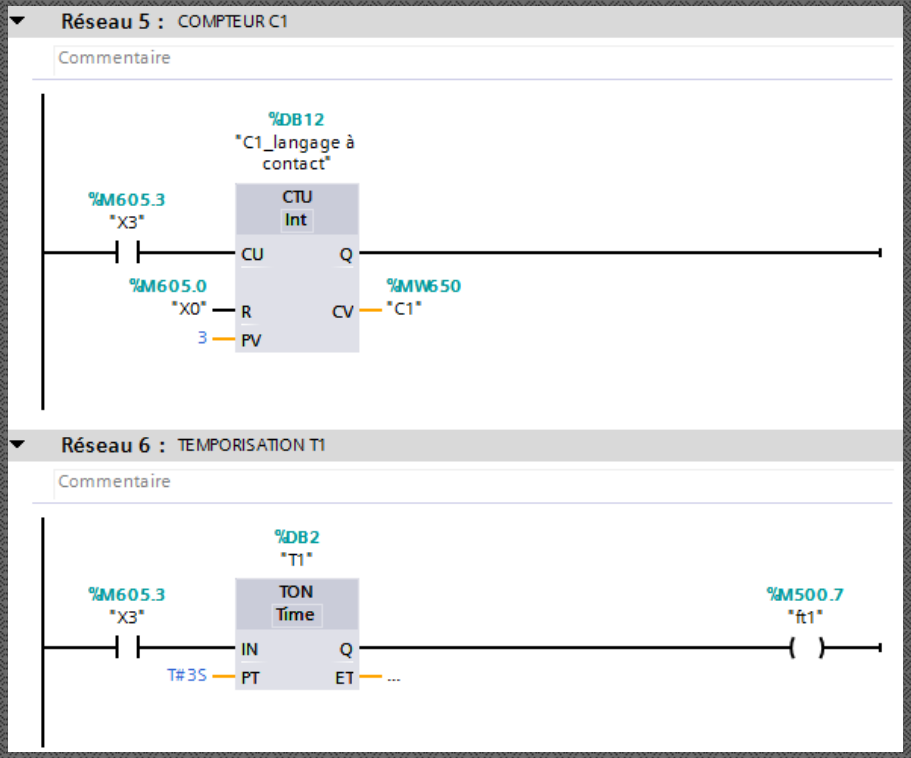
III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en langage à contact

SIEMENS Totally Integrated Automation PORTAL V15



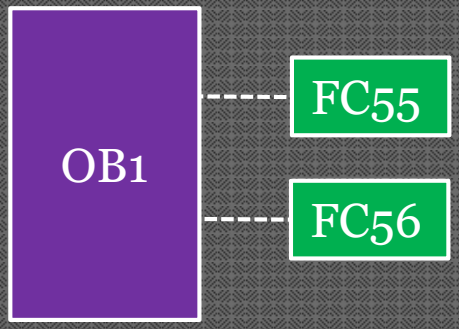
Actions: Opérations numériques

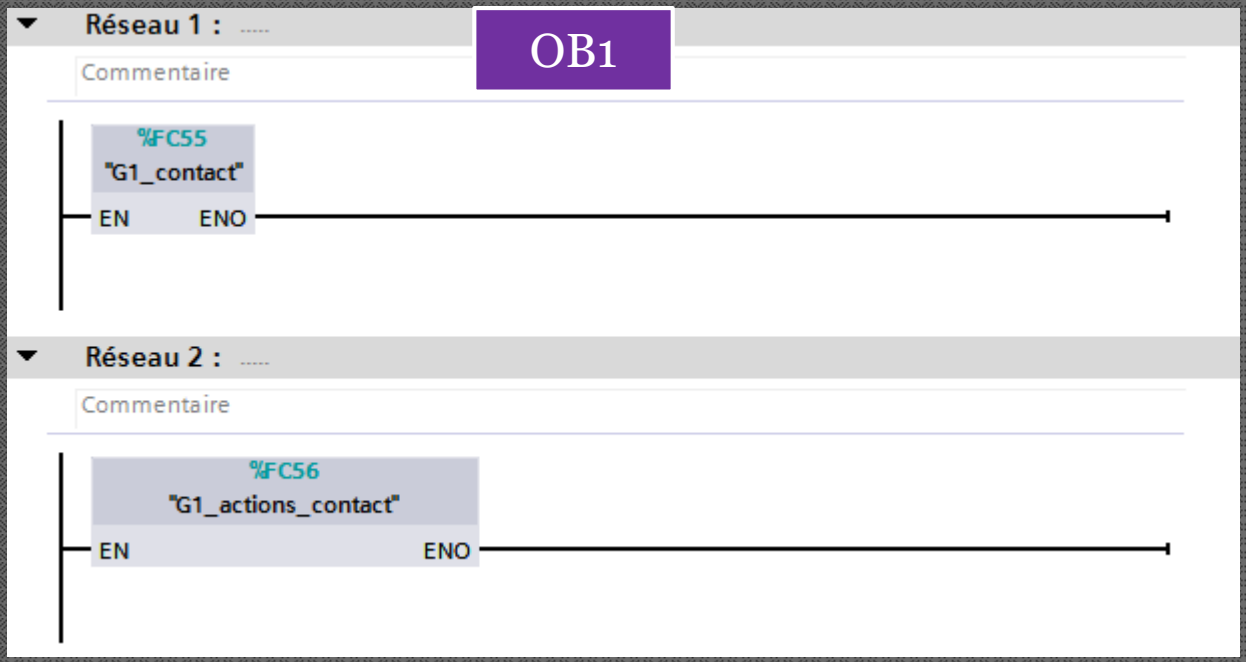


Programme du GRAFCET en langage à contact

SIEMENS Totally Integrated Automation PORTAL V15

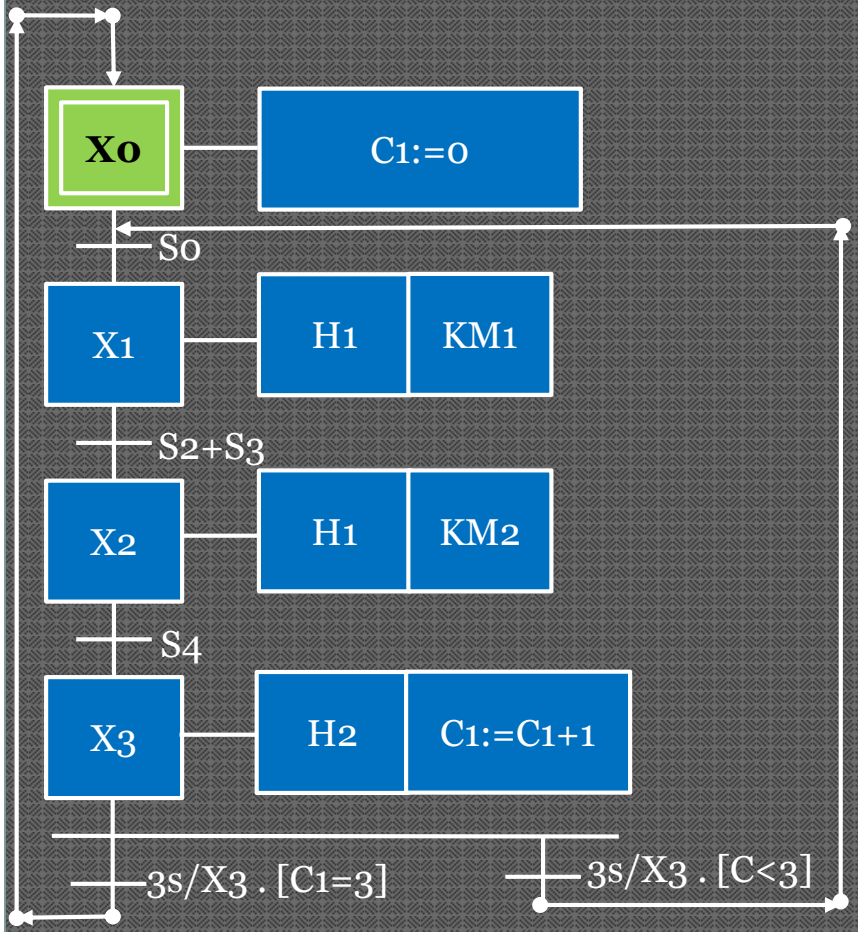
Structure d'appel dans l'OB1



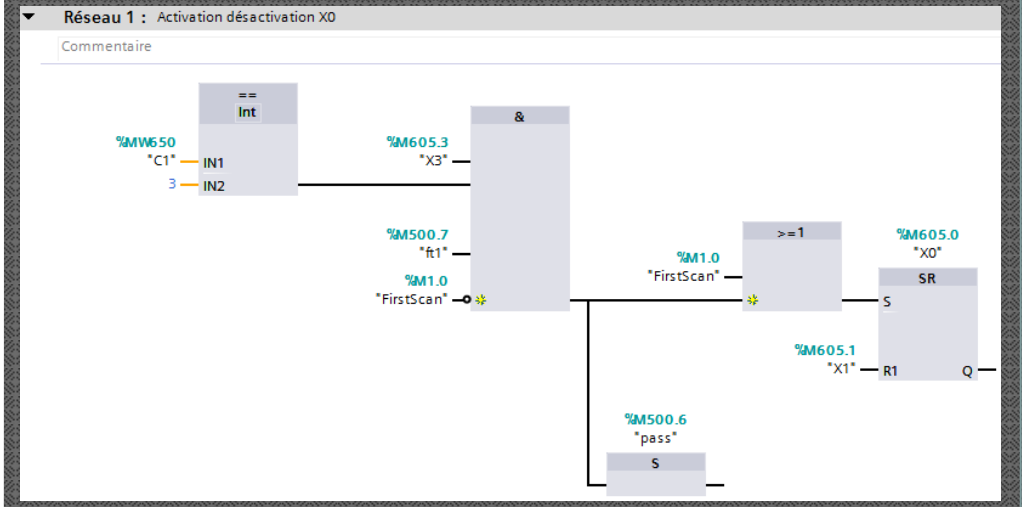


Programme du GRAFCET en logigramme

SIEMENS Totally Integrated Automation PORTAL V15



Activation et désactivation de X0 :
X0(1) = X3 . (3s/X3) . [C1=3] + FirstScan
X0(0)= X1

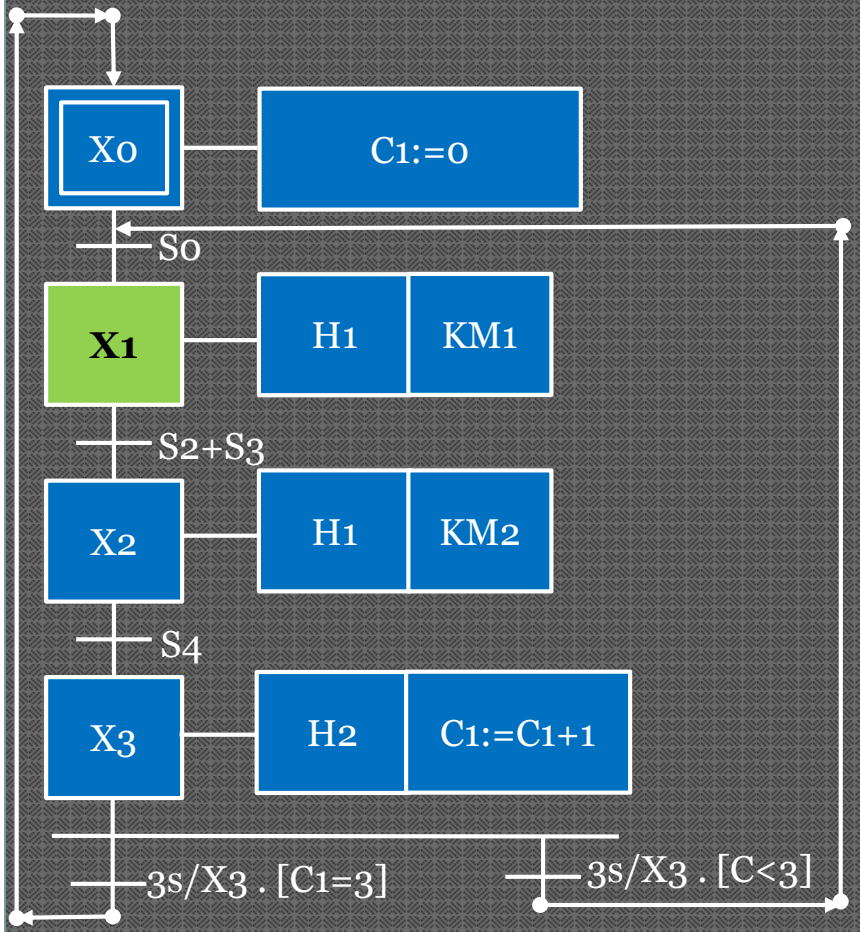


III.9: Programmer un GRAFCET en langages normalisés

CEI61131

Programme du GRAFCET en logigramme

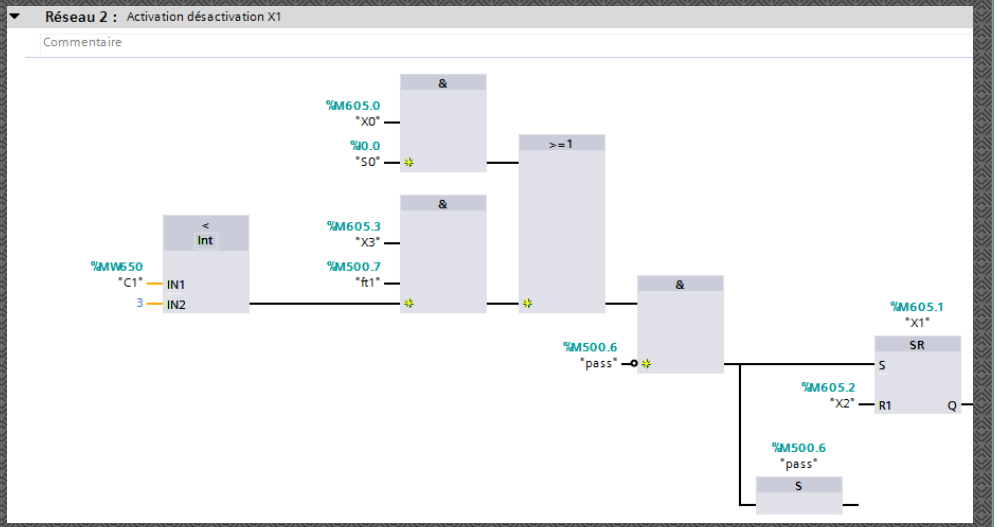
SIEMENS Totally Integrated Automation PORTAL V15

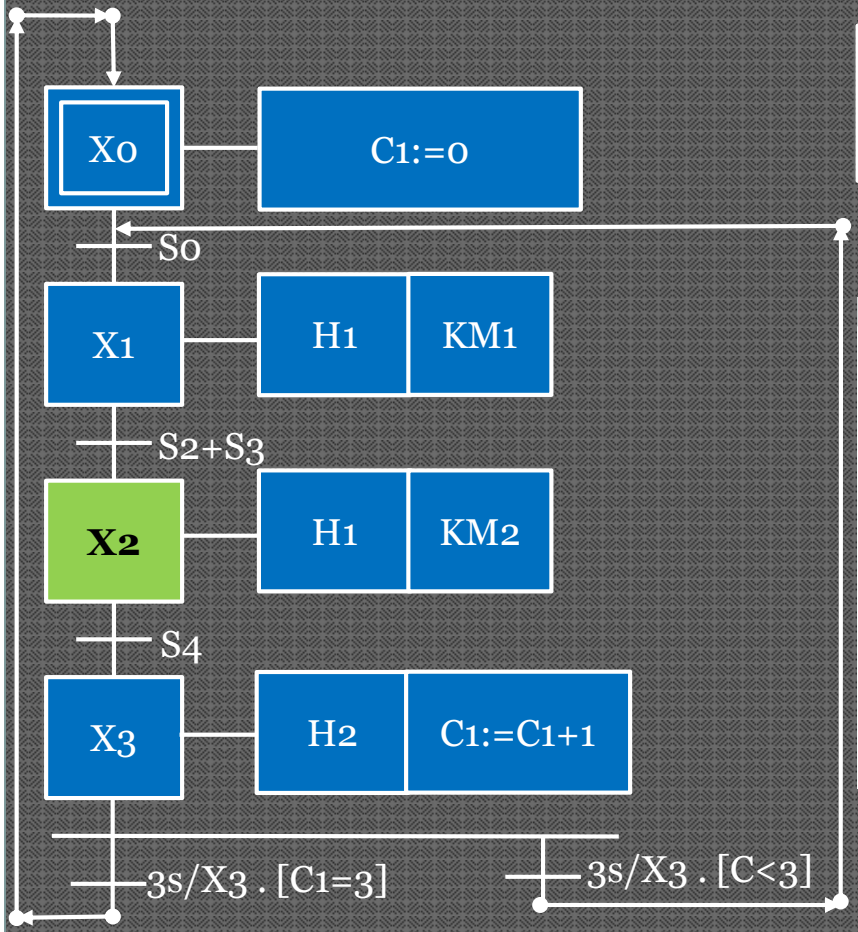


Activation et désactivation de X1 :

X1(1) = (X3 . (3s/X3) . [C1<3]) + (X0 . S0)

X1(0)= X2

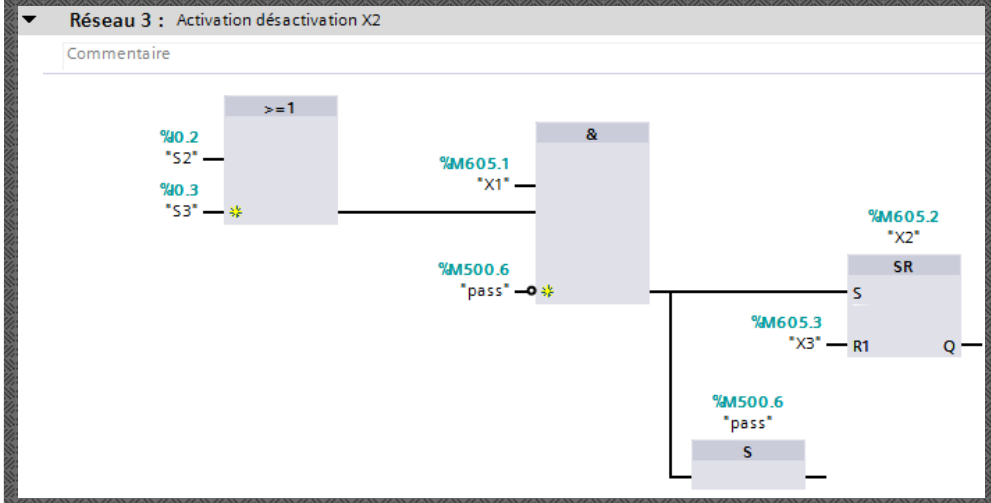


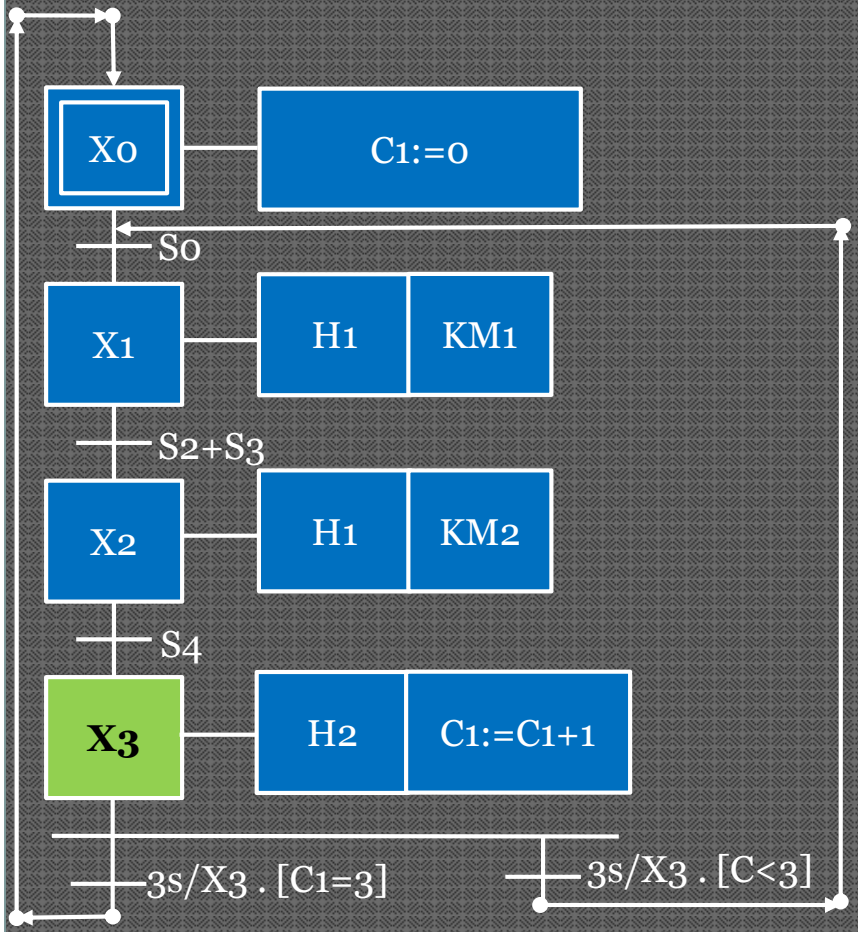


Activation et désactivation de X2 :

X2(1) = X1 . (S2 + S3)

X2(0)= X3

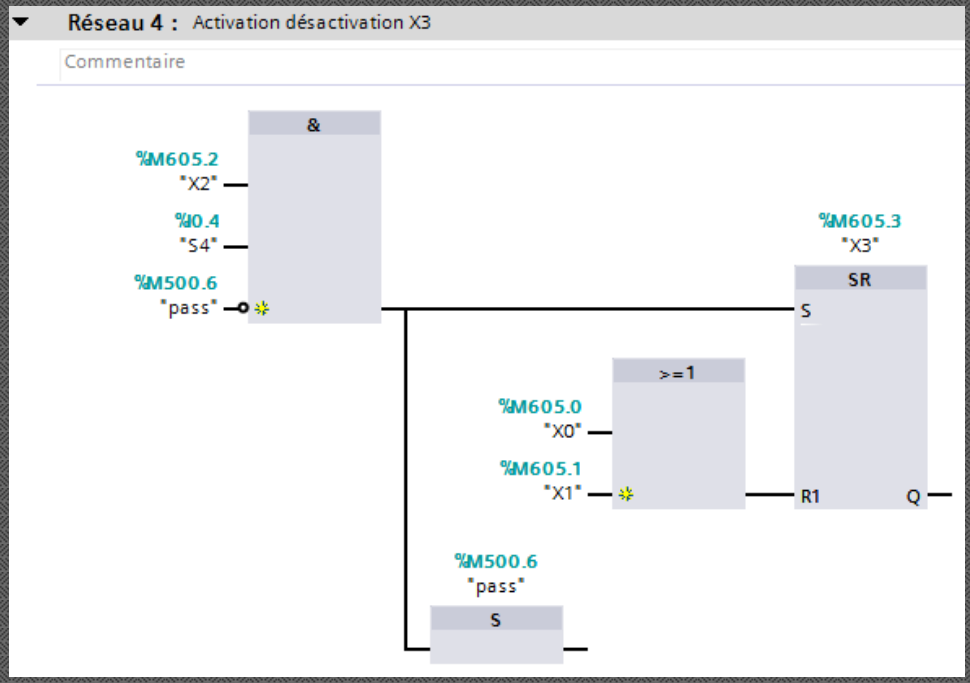




Activation et désactivation de X3 :

X3(1) = X2 . S4

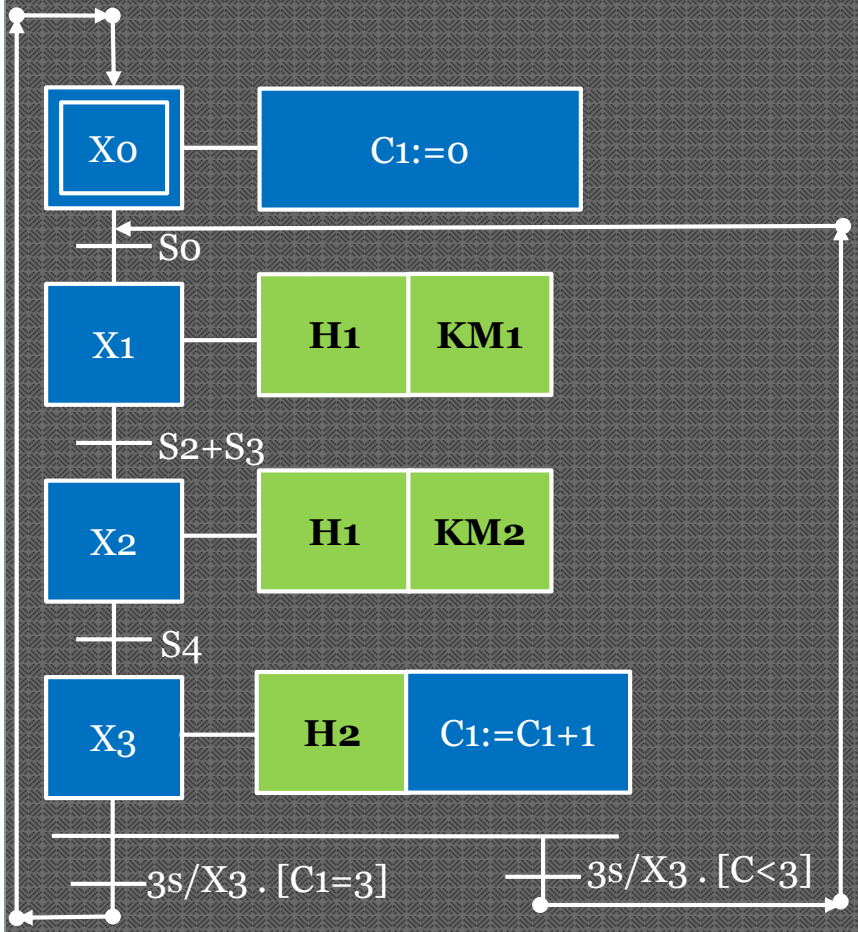
X3(0)= X0 + X1



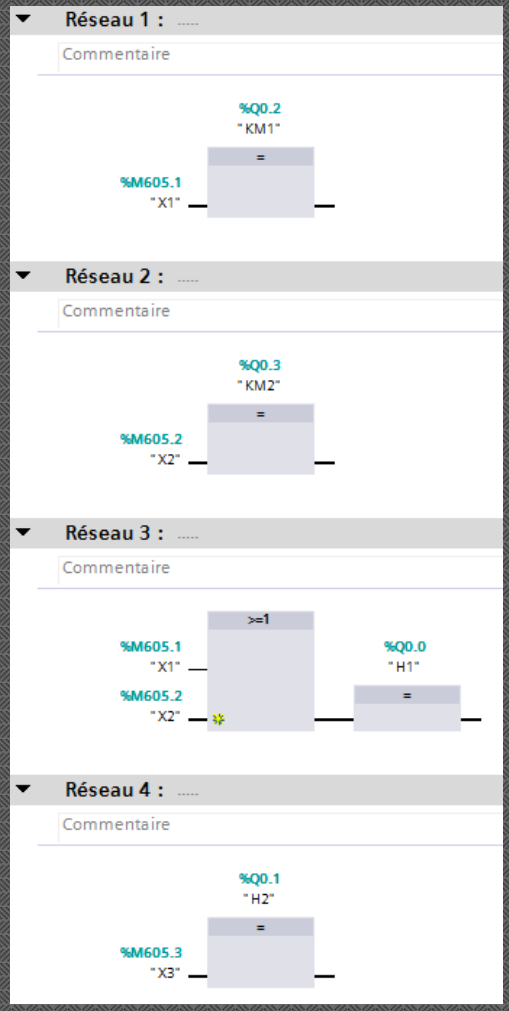
III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en logigramme

SIEMENS Totally Integrated Automation PORTAL V15



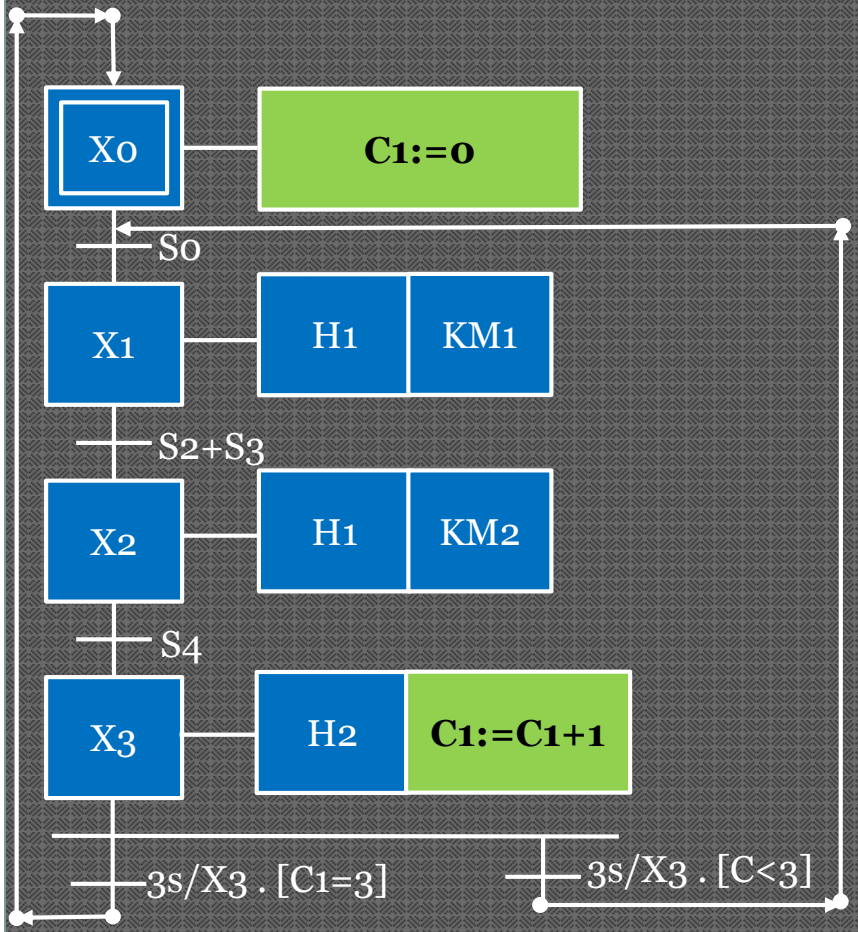
Actions: Opérations binaires



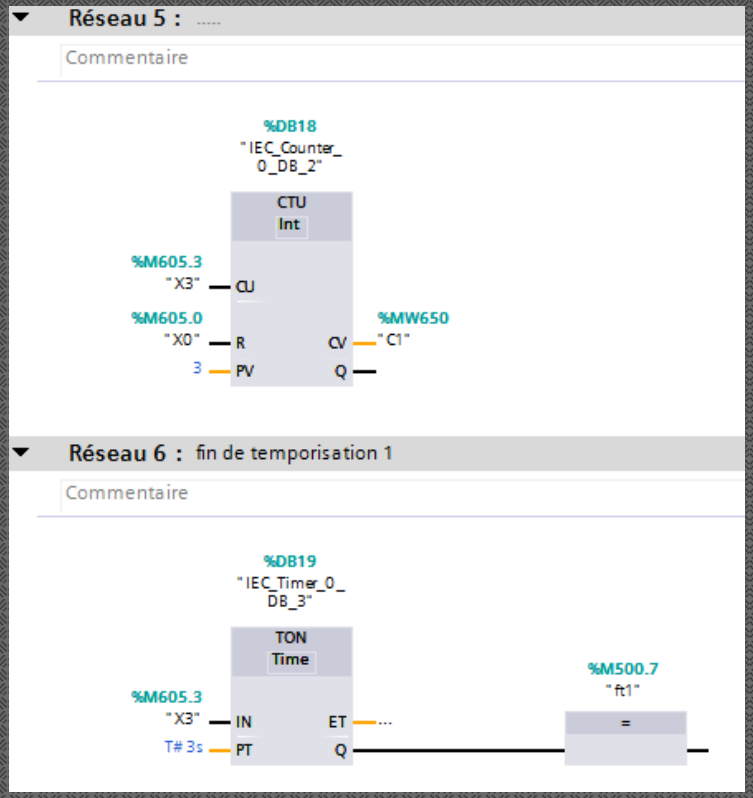
III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en logigramme

SIEMENS Totally Integrated Automation PORTAL V15



Actions: Opérations numériques

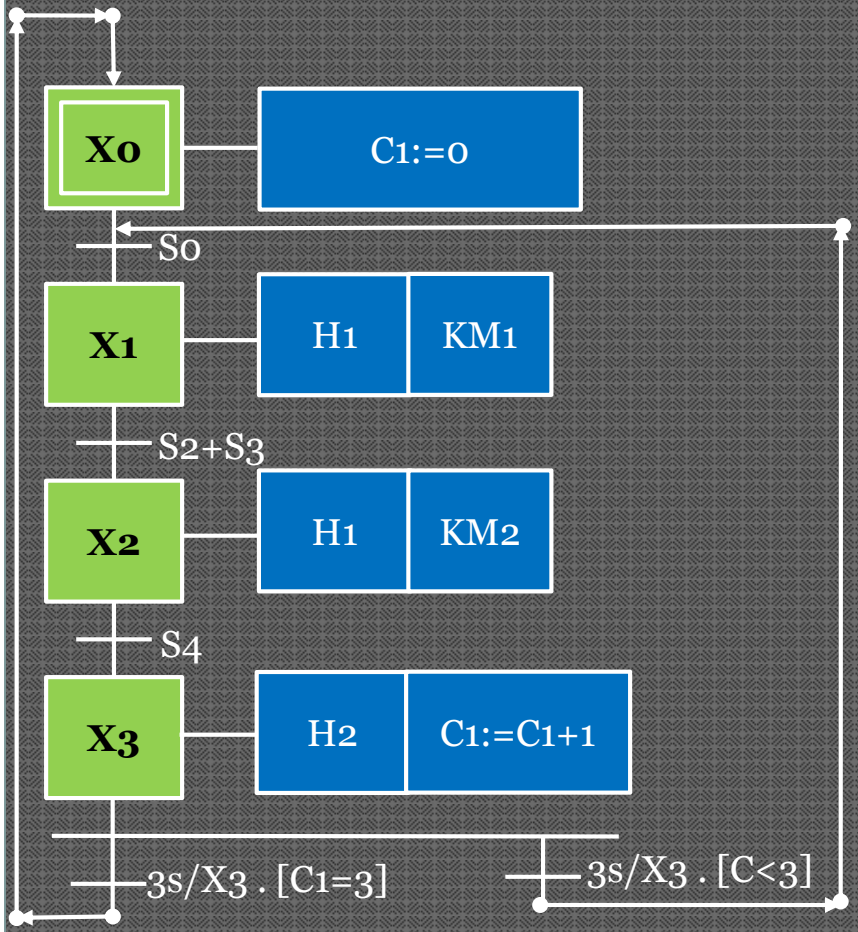


III.9: Programmer un GRAFCET en langages normalisés

CEI61131

Programme du GRAFCET en SCL

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

```
1  //Initialisation du GRAFCET
2  IF "FirstScan"
3  THEN
4      "etat_courant" := 0;
5      "etat_futur" := 0;
6  END_IF;
7
8  //etape-X0
9  IF "etat_courant" = 3 AND "ft1" AND "C1" = 3
10 THEN
11     "etat_futur" := 0;
12 END_IF;
13 //etape-X1
14 IF ("etat_courant" = 0 AND "S0") OR ("etat_courant" = 3 AND "ft1" AND "C1" < 3)
15 THEN
16     "etat_futur" := 1;
17 END_IF;
18 //etape-X2
19 IF "etat_courant" = 1 AND ("S2" OR "S3")
20 THEN
21     "etat_futur" := 2;
22 END_IF;
23 //etape-X3
24 IF "etat_courant" = 2 AND "S4"
25 THEN
26     "etat_futur" := 3;
27 END_IF;
28 //actualisation à chaque tour de cycle API
29 "etat_courant" := "etat_futur";
```

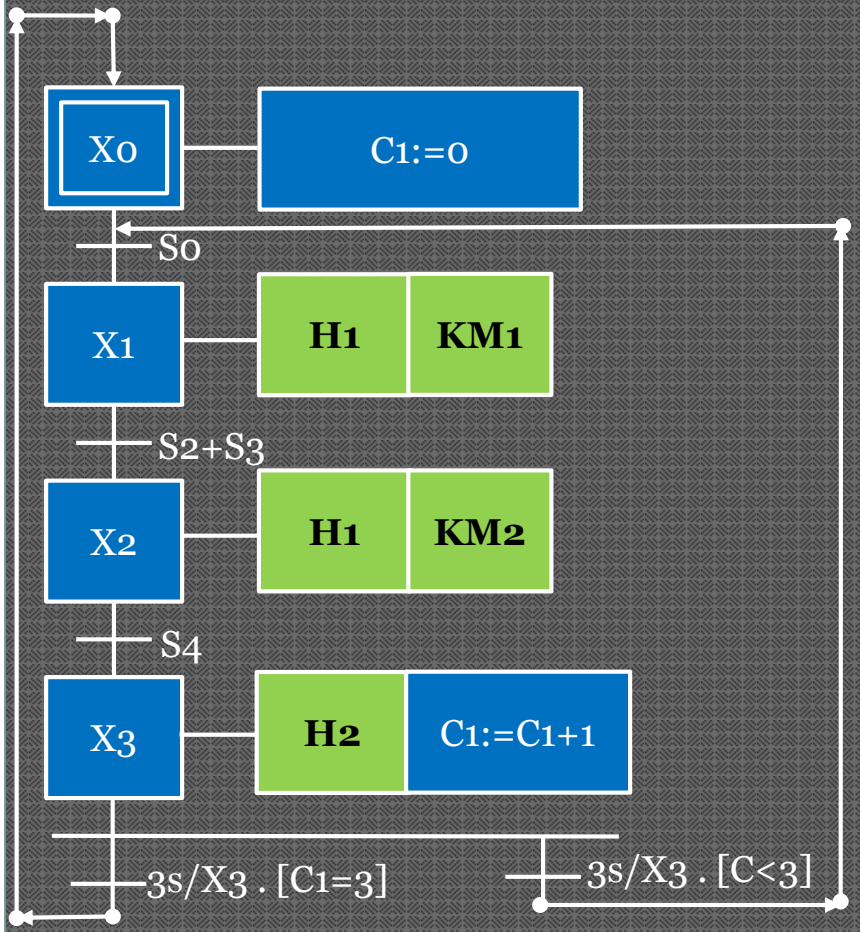
Remarque: la solution proposée n'est pas la solution, d'autres méthodes peuvent être utilisées. La meilleure solution est celle qui fonctionne !

III.9: Programmer un GRAFCET en langages normalisés

CEI61131

Programme du GRAFCET en SCL

SIEMENS Totally Integrated Automation PORTAL V15



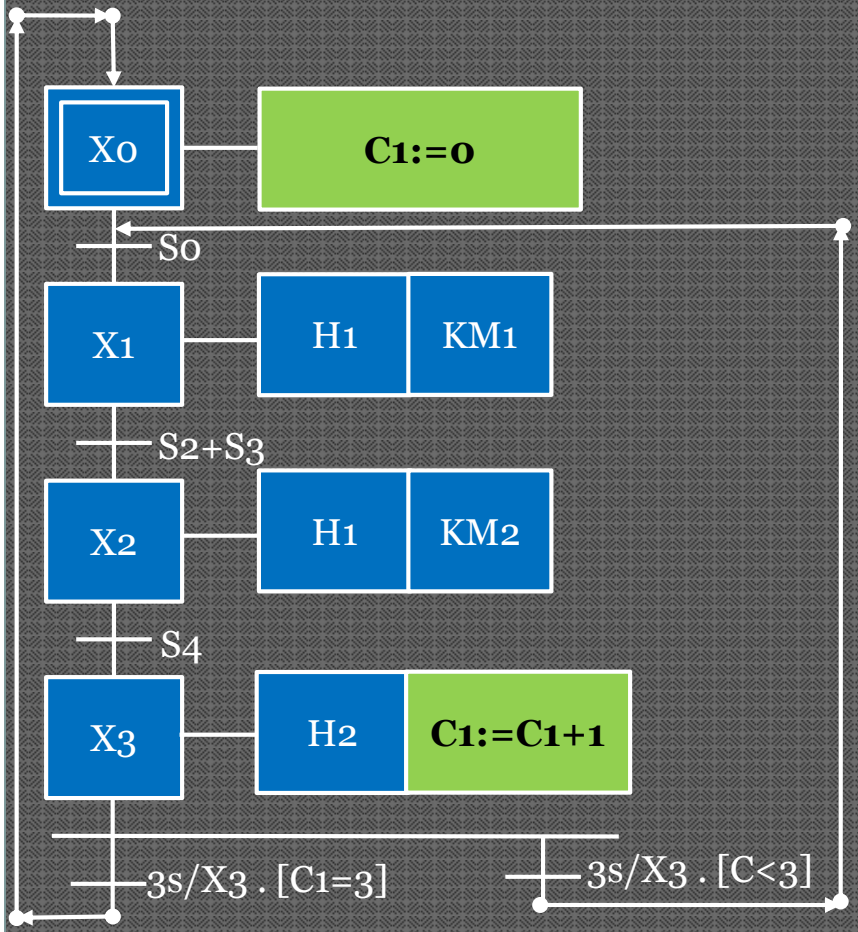
Actions: Opérations binaires

```
75  //ACTIONS BINAIRES
76 IF "etat_courant" = 1
77  THEN
78      "KM1" := 1;
79  ELSE
80      "KM1" := 0;
81  END_IF;
82 IF "etat_courant" = 2
83  THEN
84      "KM2" := 1;
85  ELSE
86      "KM2" := 0;
87  END_IF;
88 IF "etat_courant" = 1 OR "etat_courant" = 2
89  THEN
90      "H1" := 1;
91  ELSE
92      "H1" := 0;
93  END_IF;
94 IF "etat_courant" = 3
95  THEN
96      "H2" := 1;
97      "RUN_TEMPO_T1" := 1;//bit de démarrage temporsation T1
98      "INC_C1" := 1;//bit d'incrémentation compteur
99  ELSE
100     "H2" := 0;
101     "RUN_TEMPO_T1" := 0;
102     "INC_C1" := 0;
103 END_IF;
104 IF "etat_courant" = 0
105 THEN
106     "RAZ_compteur" := 1;//bit RAZ compteur
107 ELSE
108     "RAZ_compteur" := 0;
109 END_IF;
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en SCL

SIEMENS Totally Integrated Automation PORTAL V15



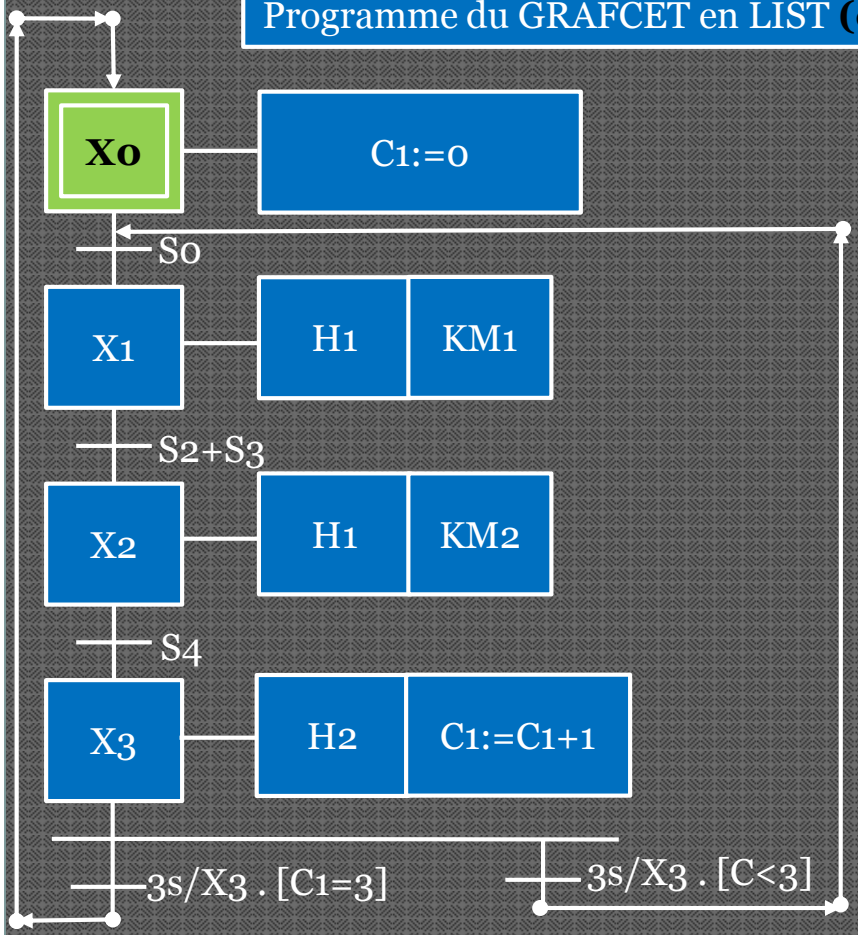
Actions: Opérations numériques

```
111  //ACTIONS NUMERIQUES
112  //COMPTEUR
113 "IEC_Counter_0_DB".CTU(CU := "INC_C1",
114                        R := "RAZ_compteur",
115                        PV := 3,
116                        CV => "C1");
117  //TEMPORISATION
118 "IEC_Timer_0_DB".TON(IN := "RUN_TEMPO_T1",
119                       PT := T#3s,
120                       Q => "ft1");
121
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au SCL)**

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

Réseau 1 : Initialisation GRAFCET

Commentaire

```
1  //initialisation du GRAFCET
2        A     "FirstScan"          %M1.0
3        JCN   saut0
4        L     0                    0
5        T     "etat_courant"       %MW600
6  saut0: NOP 0
7        A     "FirstScan"          %M1.0
8        JCN   saut1
9        L     0                    0
10       T     "etat_futur"         %MW602
11 saut1: NOP 0
```

Réseau 2 : Etape 0

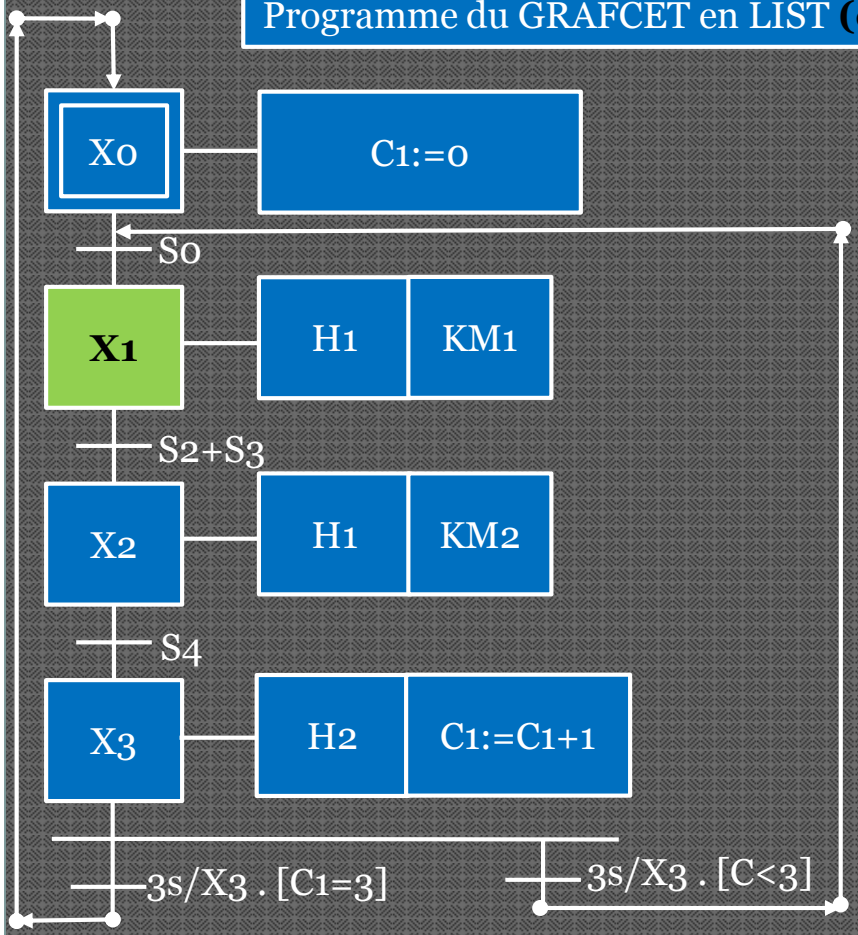
Commentaire

```
1  //etape X0
2        A(
3        L     "etat_courant"       %MW600
4        L     3                    3
5        ==I
6        )
7        A(
8        L     "C1"                 %MW650
9        L     3                    3
10       ==I
11       )
12       A     "ft1"                %M500.7
13       JCN   saut2
14       L     0                    0
15       T     "etat_futur"         %MW602
16 saut2: NOP 0
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au SCL)**

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

Réseau 3 : Etape 1

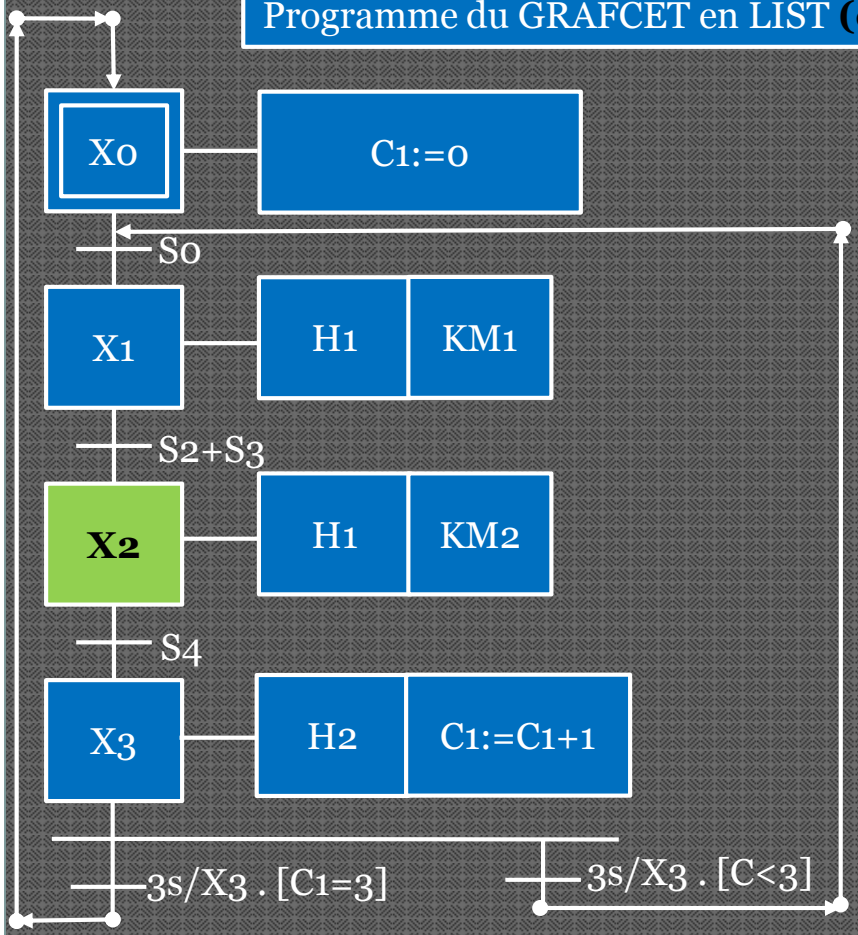
Commentaire

```
1  //etape X1
2        A(
3        A(
4        L     "etat_courant"          %MW600
5        L     0                       0
6        ==I
7        )
8        A     "S0"                    %I0.0
9        O
10       A(
11       L     "etat_courant"          %MW600
12       L     3                       3
13       ==I
14       )
15       A(
16       L     "C1"                    %MW650
17       L     3                       3
18       <I
19       )
20       A     "ft1"                   %M500.7
21       )
22       JCN   saut3
23       L     1                       1
24       T     "etat_futur"            %MW602
25 saut3: NOP 0
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au SCL)**

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

Réseau 4 : Etape 2

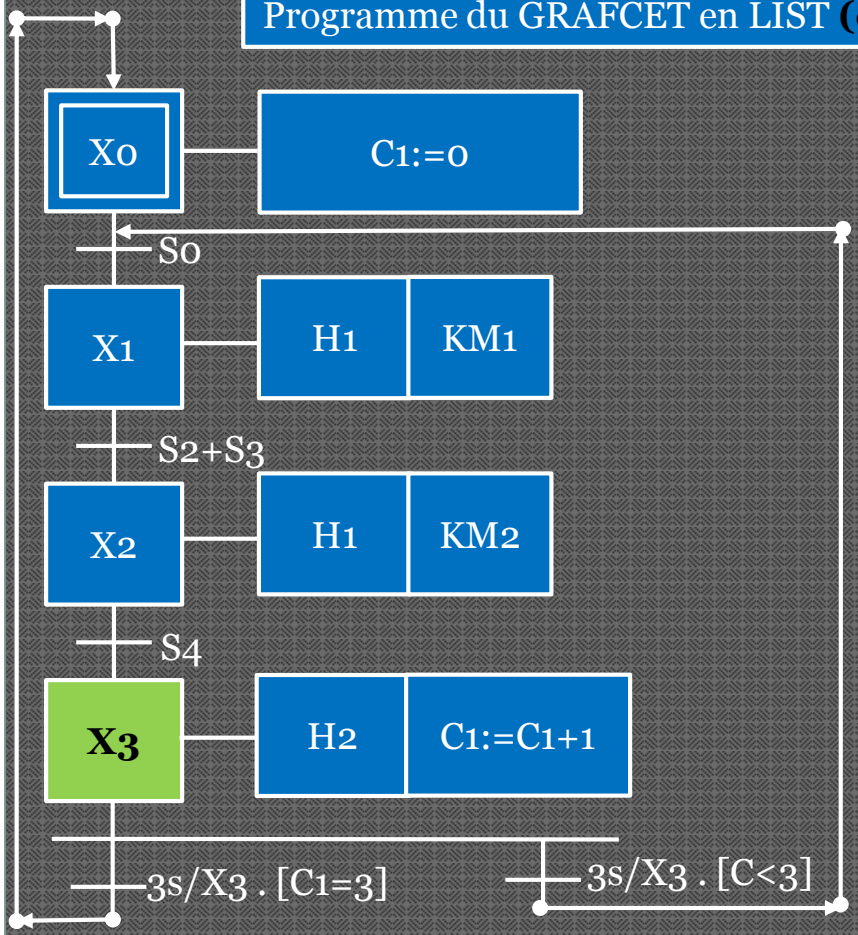
Commentaire

```
1  //etape X2
2        A(
3        L     "etat_courant"          %MW600
4        L     1                       1
5        ==I
6        )
7        A(
8        O     "S2"                    %I0.2
9        O     "S3"                    %I0.3
10       )
11       JCN   saut4
12       L     2                       2
13       T     "etat_futur"            %MW602
14 saut4: NOP 0
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au SCL)**

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

Réseau 5 : Etape 3

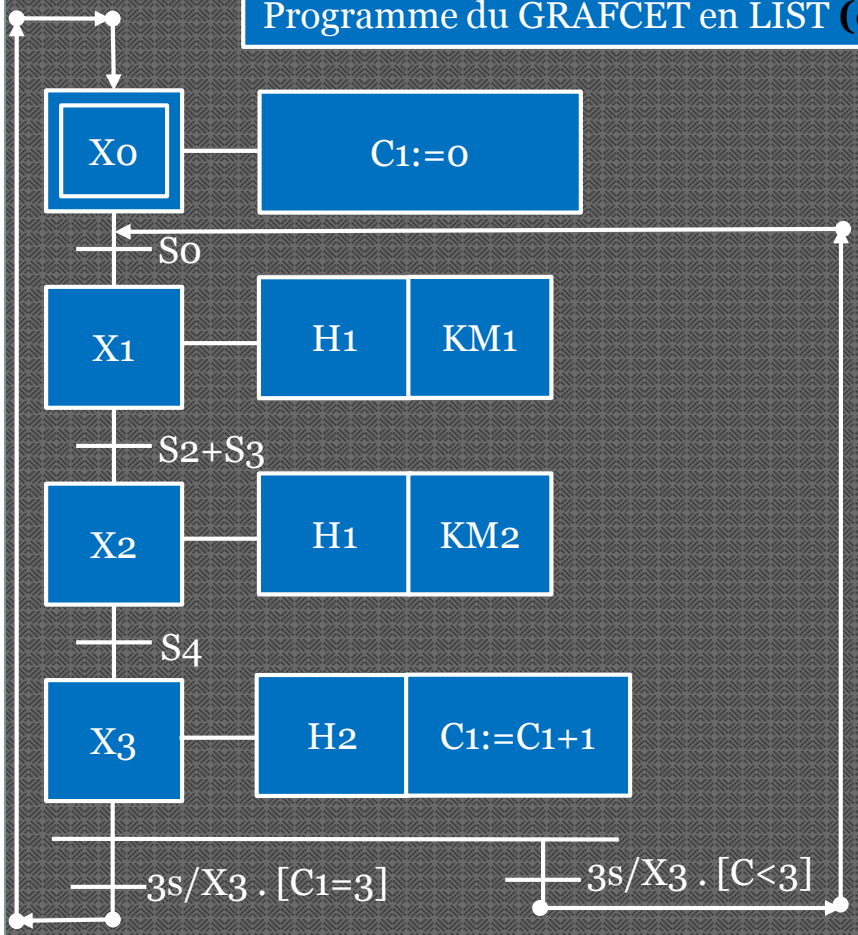
Commentaire

```
1  //etape X3
2        A(
         L     "etat_courant"          %MW600
4        L     2                       2
5        ==I
6        )
7        A     "S4"                    %I0.4
8        JCN   saut5
9        L     3                       3
10       T     "etat_futur"            %MW602
11 saut5: NOP 0
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au SCL)**

SIEMENS Totally Integrated Automation PORTAL V15



Actualisation variable sur tour de cycle API

Réseau 6 : Actuallisation à chaque tour de cycle API

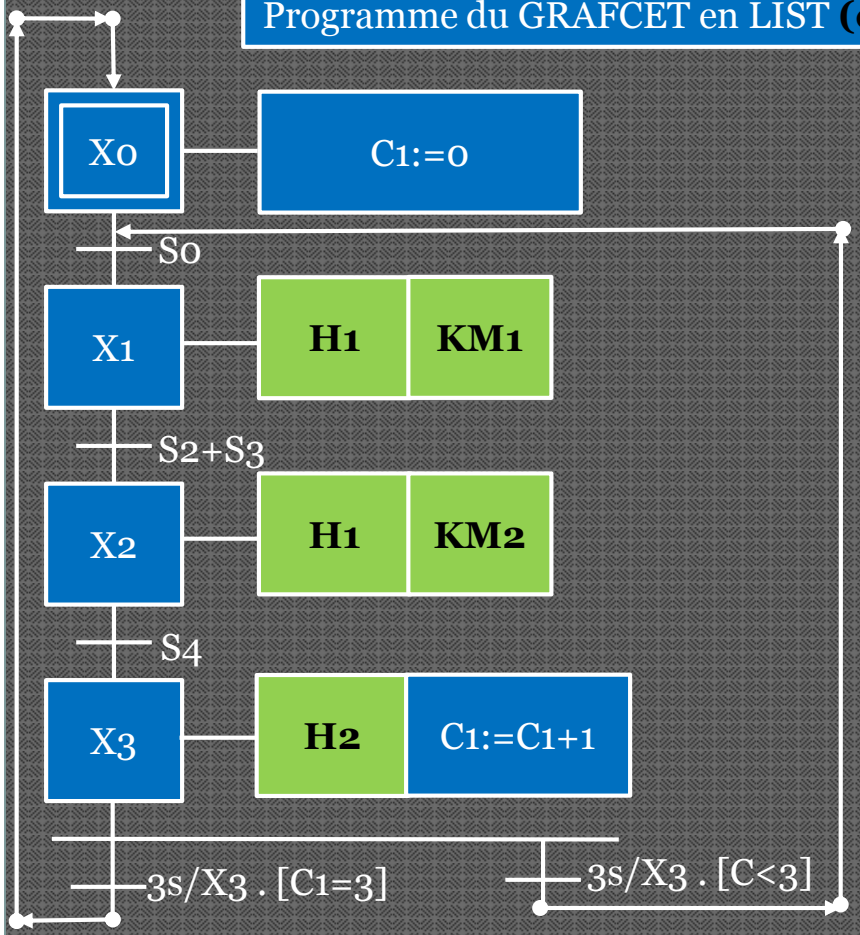
Commentaire

```
1    L    "etat_futur"      %MW602
2    T    "etat_courant"    %MW600
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au SCL)**

SIEMENS Totally Integrated Automation PORTAL V15



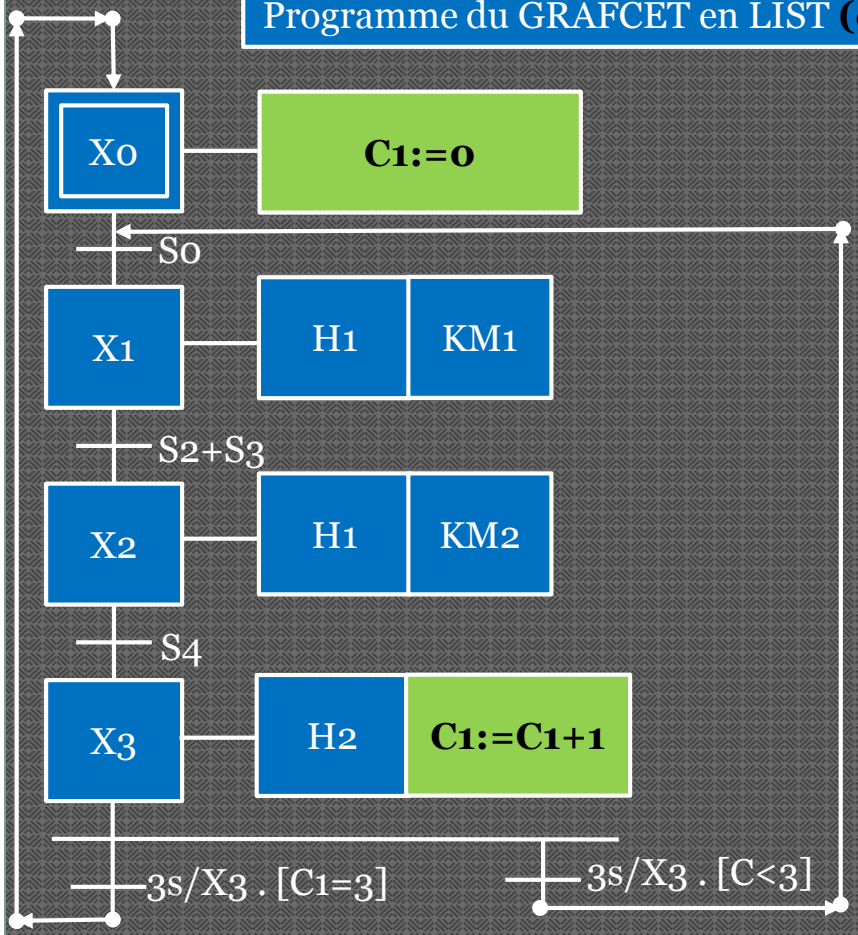
Actions: Opérations binaires

```
1     L     "etat_courant"                %MW600
2     L     0                             0
3     ==I
4     =     "RAZ_compteur"                %M504.1
5
6     O(
7     L     "etat_courant"                %MW600
8     L     1                             1
9     ==I
10    )
11    O(
12    L     "etat_courant"                %MW600
13    L     2                             2
14    ==I
15    )
16    =     "H1"                          %Q0.0
17
18    L     "etat_courant"                %MW600
19    L     1                             1
20    ==I
21    =     "KM1"                         %Q0.2
22
23    L     "etat_courant"                %MW600
24    L     2                             2
25    ==I
26    =     "KM2"                         %Q0.3
27
28    L     "etat_courant"                %MW600
29    L     3                             3
30    ==I
31    =     "H2"                          %Q0.1
32
33    L     "etat_courant"                %MW600
34    L     3                             3
35    ==I
36    =     "RUN_TEMPO_T1"                %M504.0
37
38    L     "etat_courant"                %MW600
39    L     3                             3
40    ==I
41    =     "INC_C1"                      %M501.0
42
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au SCL)**

SIEMENS Totally Integrated Automation PORTAL V15



Actions: Opérations numériques

Réseau 8 : Compteur

Commentaire

```
1   CALL  CTU , "IEC_Counter_0_DB_1"        %DB3
2     Int
3     CU :="INC_C1"                         %M501.0
4     R  :="RAZ_compteur"                   %M504.1
5     PV :=3                                3
6     Q  :=
7     CV :="C1"                             %MW650
8
9
```

Réseau 9 : Temporisation

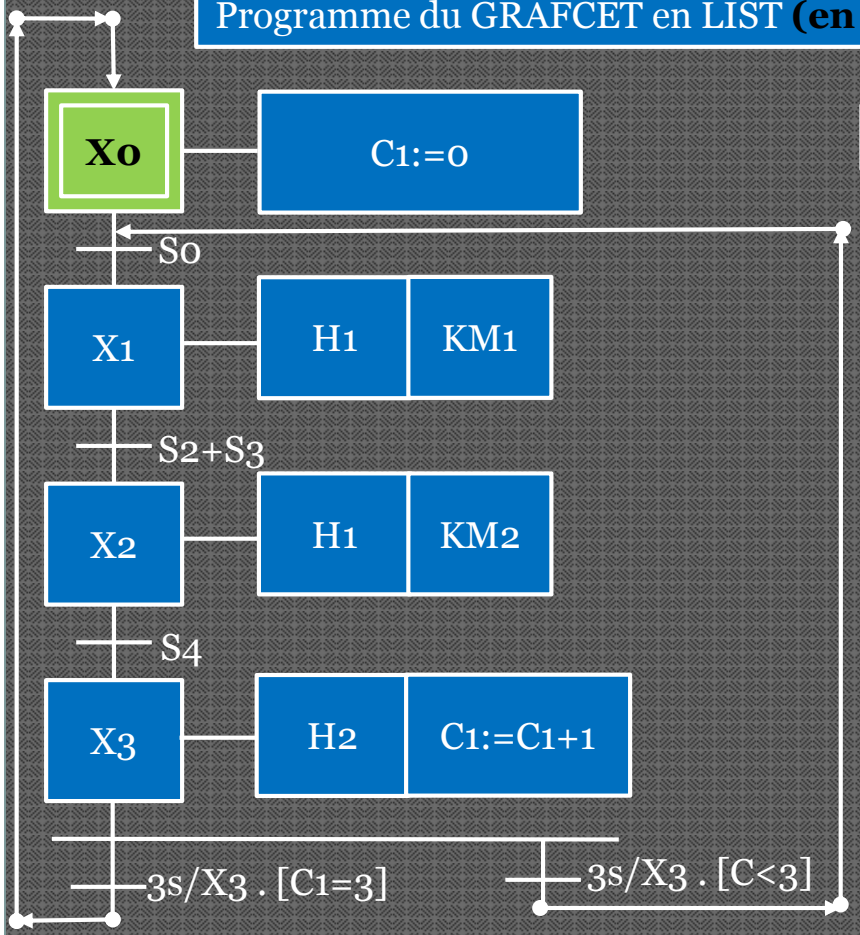
Commentaire

```
1
2   CALL  TON , "IEC_Timer_0_DB_1"          %DB4
3     Time
4     IN :="RUN_TEMPO_T1"                   %M504.0
5     PT :=T#3s                             T#3s
6     Q  :="ft1"                            %M500.7
7     ET :=
8
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au Ladder)**

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

Réseau 1 : Etape 0 Activation

Commentaire

```
1   A    "X3"                    %M605.3
2   A    "IEC_Timer_0_DB_1".Q
3   A(
4   L    "C1"                    %MW650
5   L    3                       3
6   ==I
7   )
8   AN   "pass"                  %M500.6
9   O    "FirstScan"             %M1.0
10  S    "X0"                    %M605.0
11  S    "pass"                  %M500.6
12
```

Réseau 2 : Etape 0 Désactivation

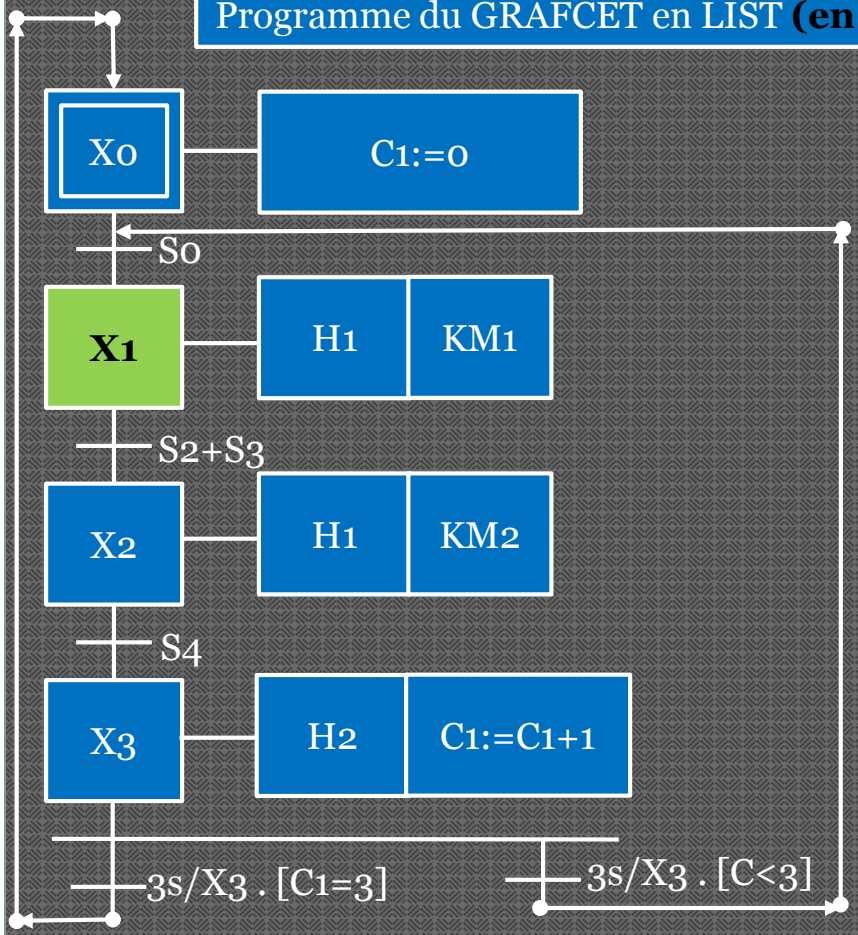
Commentaire

```
1   A    "X1"                    %M605.1
2   R    "X0"                    %M605.0
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au Ladder)**

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

Réseau 3 : Etape 1 Activation

Commentaire

```
1   A(
2   A    "X3"                    %M605.3
3   A    "IEC_Timer_0_DB_1".Q
4   A(
5   L    "C1"                    %MW650
6   L    3                       3
7   <I
8   )
9   O
10  A    "X0"                    %M605.0
11  A    "MS0"                   %M504.2
12  )
13  AN   "pass"                  %M500.6
14  S    "X1"                    %M605.1
15  S    "pass"                  %M500.6
```

Réseau 4 : Etape 1 Désactivation

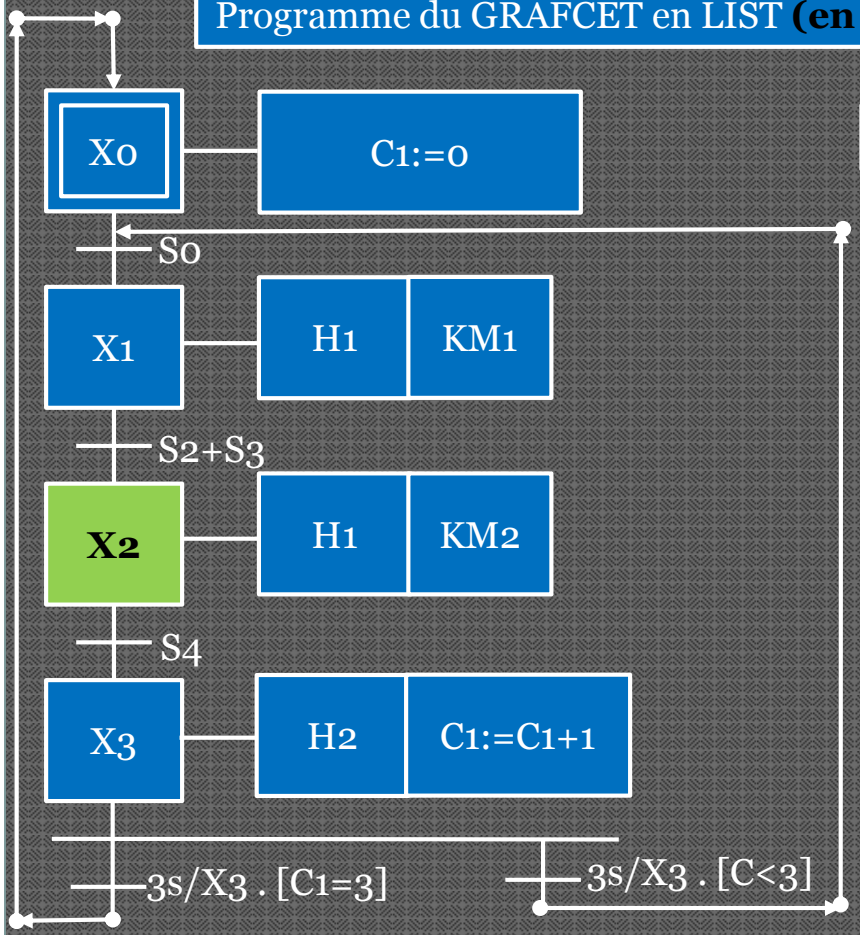
Commentaire

```
1   A    "X2"                    %M605.2
2   R    "X1"                    %M605.1
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au Ladder)**

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

Réseau 5 : Etape 2 Activation

Commentaire

```
1   A    "X1"      %M605.1
2   A(
3   O    "MS2"     %M504.4
4   O    "MS3"     %M504.5
5   )
6   AN   "pass"    %M500.6
7   S    "X2"      %M605.2
8   S    "pass"    %M500.6
9
```

Réseau 6 : Etape 2 Désactivation

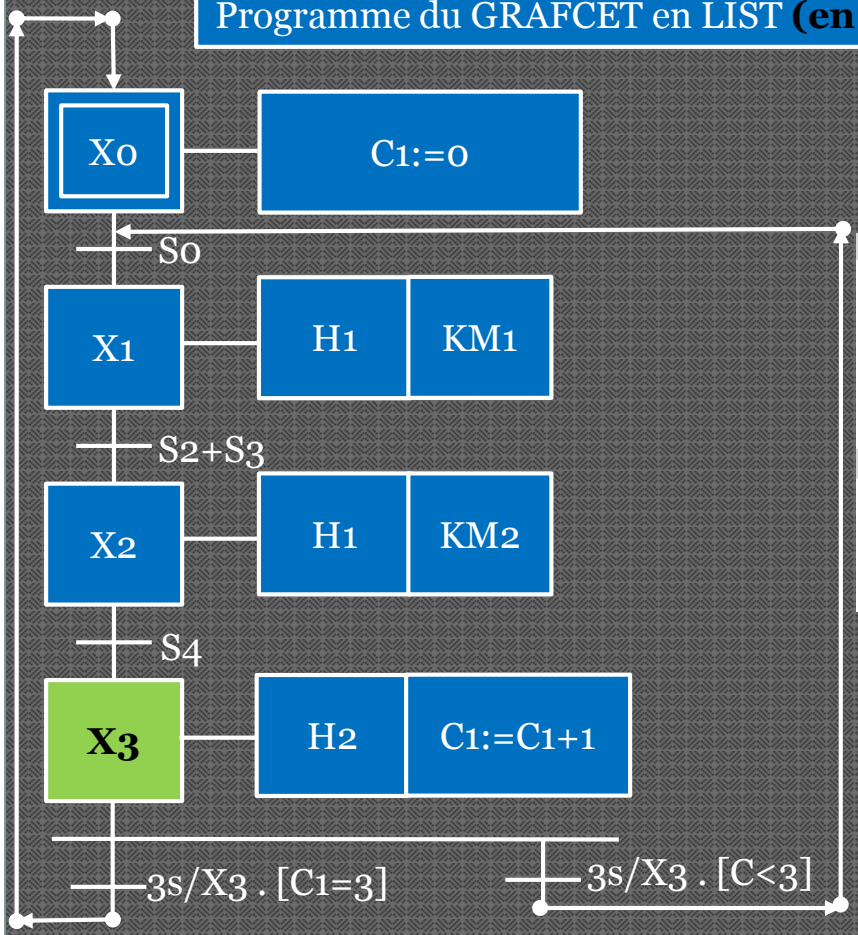
Commentaire

```
1   A    "X3"      %M605.3
2   R    "X2"      %M605.2
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au Ladder)**

SIEMENS Totally Integrated Automation PORTAL V15



Programmation des étapes du GRAFCET

Réseau 7 : Etape 3 Activation

Commentaire

```
1   A    "X2"      %M605.2
2   A    "MS4"     %M504.6
3   AN   "pass"    %M500.6
4   S    "X3"      %M605.3
5   S    "pass"    %M500.6
```

Réseau 8 : Etape désactivation

Commentaire

```
1   O    "X0"      %M605.0
2   O    "X1"      %M605.1
3   R    "X3"      %M605.3
```

Actualisation du bit « pass » à chaque tour de cycle API

Réseau 9 : Actuallisation à chaque tour de cycle API

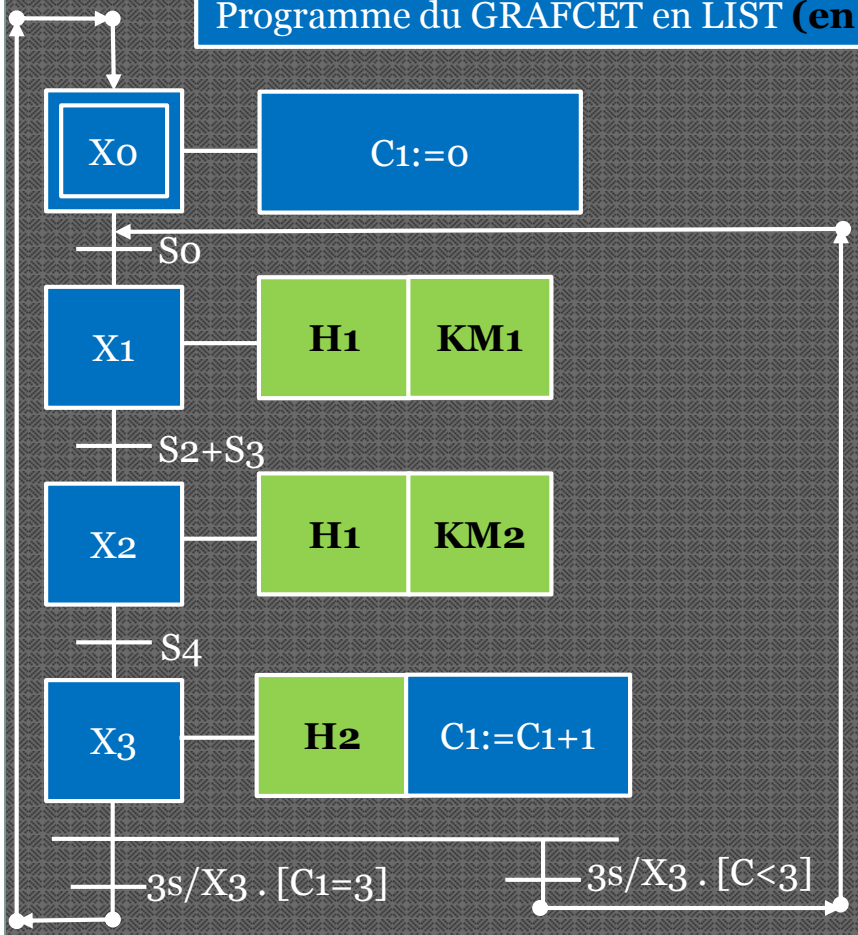
Commentaire

```
1   A    "AlwaysTRUE"   %M1.2
2   R    "pass"         %M500.6
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au Ladder)**

SIEMENS Totally Integrated Automation PORTAL V15



Actions: Opérations binaires

Réseau 10 : Actions opérations binaires

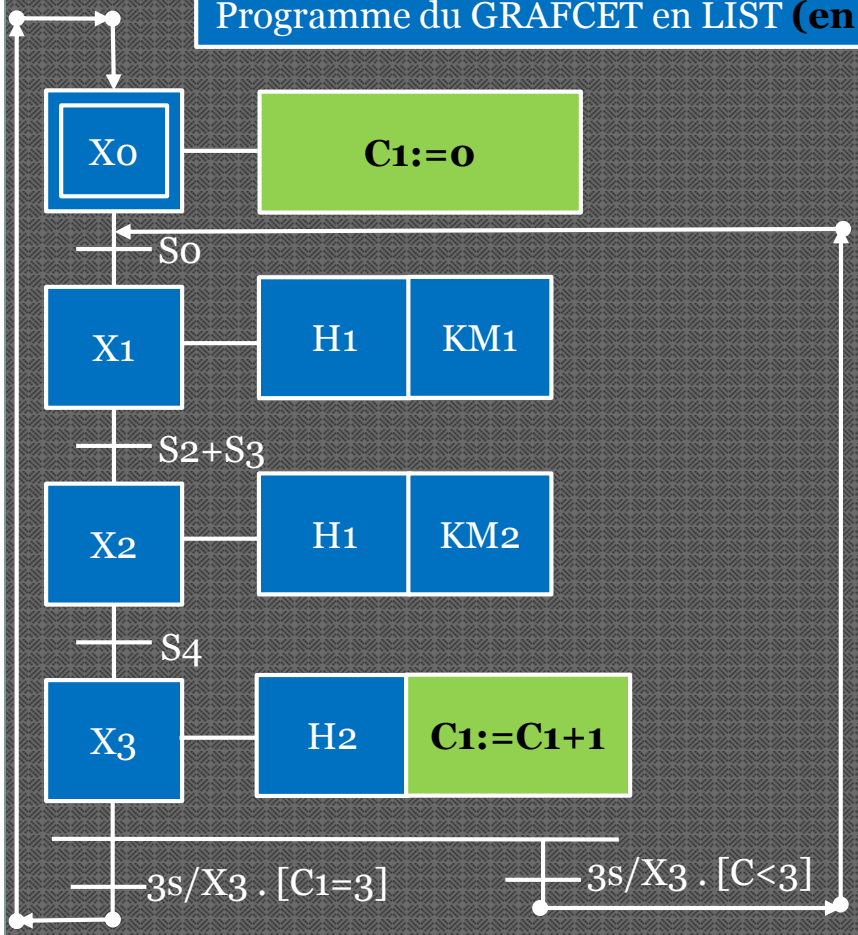
Commentaire

```
1    O    "X1"    %M605.1
2    O    "X2"    %M605.2
3    =    "H1"    %Q0.0
4
5    A    "X1"    %M605.1
6    =    "KM1"   %Q0.2
7
8    A    "X2"    %M605.2
9    =    "KM2"   %Q0.3
10
11   A    "X3"    %M605.3
12   =    "H2"    %Q0.1
```

III.9: Programmer un GRAFCET en langages normalisés CEI61131

Programme du GRAFCET en LIST **(en référence au Ladder)**

SIEMENS Totally Integrated Automation PORTAL V15



Actions: Opérations numériques

Réseau 11 : Compteur

Commentaire

```
1   CALL  CTU , "IEC_Counter_0_DB_1"        %DB3
2       Int
3       CU :="X3"                           %M605.3
4       R  :="X0"                           %M605.0
5       PV :=3                              3
6       Q  :=
7       CV :="C1"                           %MW650
8
9
```

Réseau 12 : Temporisation

Commentaire

```
1
2   CALL  TON , "IEC_Timer_0_DB_1"          %DB4
3       Time
4       IN :="X3"                           %M605.3
5       PT :=T#3s                           T#3s
6       Q  :=
7       ET :=
8
```

FIN

C'était pour arrondir à 1 OCTET de transparents

Rappel: $11111111_{\#2}$=$FF_{\#16}$ = $255_{\#10}$