

# AUTOMATISME

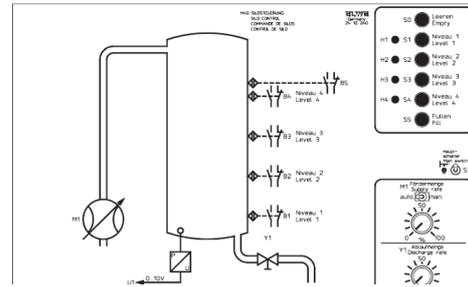
## Module AUTOM- TP3

24/09/2018

### Commande de silo



S71200



Commande de silo

### Présentation :

Pour ce TP, l'objectif sera d'aborder les mises en œuvre de :

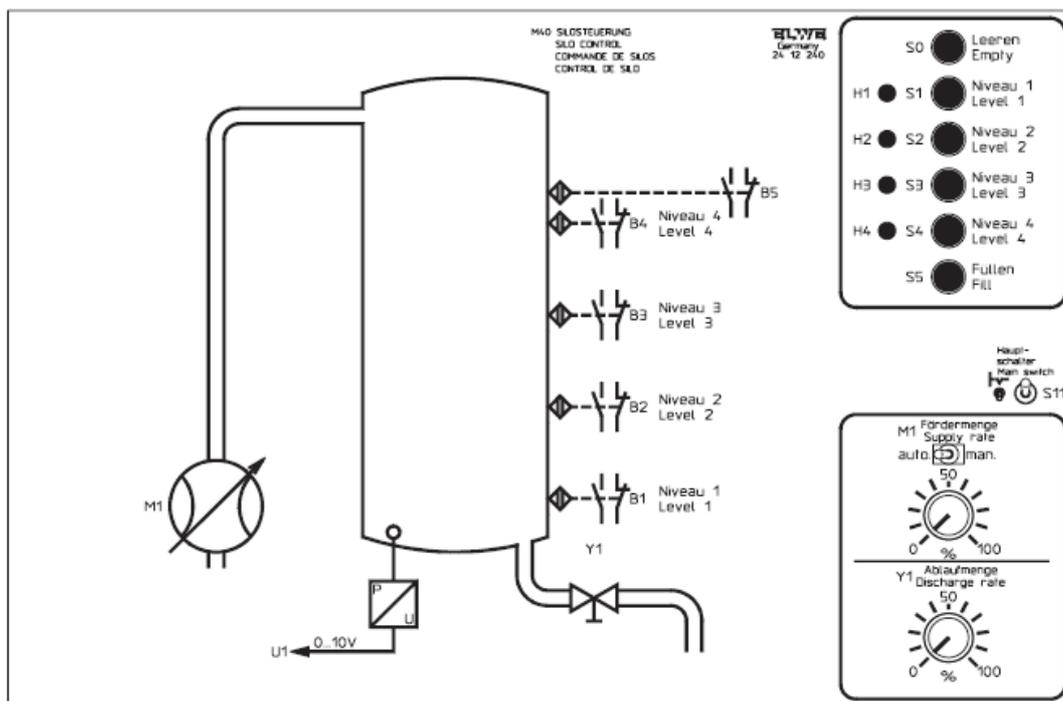
- Bloc fonctionnel avec passage de paramètres.
- OB40 : bloc d'organisation d'alarme de processus (interruption de programme par hardware).
- OB100 : bloc d'organisation de démarrage.
- OB35 : bloc d'organisation d'alarme cyclique (interruption de programme cyclique).

Le développement se fera avec TIA PORTAL V13 SP2

L'API sera un S71200 V3.0, modèle compact disposant de : 14 entrées TOR - 10 sorties TOR - 2 entrées ANA (0-10V) - 1 sortie ANA (0-10V).

Le langage CEI utilisé sera : LADDER

### Vue du système dans son ensemble :



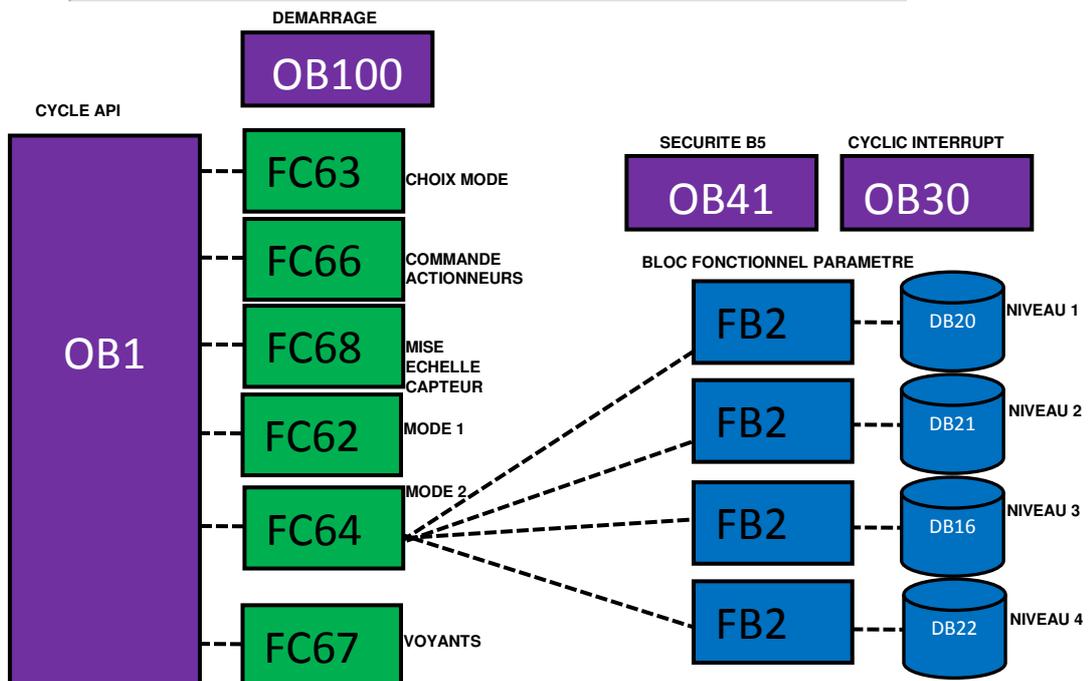


**Adressage des entrées-sorties câblées sur l'API :**

Nom	Type de données	Adresse
CONSIGNE_M1	Int	%QW80
S0	Bool	%I0.0
S1	Bool	%I0.1
S2	Bool	%I0.2
S3	Bool	%I0.3
S4	Bool	%I0.4
S5	Bool	%I0.5
B1	Bool	%I0.6
B2	Bool	%I0.7
B3	Bool	%I1.0
B4	Bool	%I1.1
B5	Bool	%I1.2
S11	Bool	%I1.3
NIVEAU_MESURE	Int	%IW64
Y1	Bool	%Q0.0
M1	Bool	%Q0.5
H1	Bool	%Q1.0
H2	Bool	%Q1.1

**Structure du programme :**

Main	OB1
▶ Choix_mode	FC63
▶ Commande_actionneurs	FC66
▶ Mise_echelle_capteur	FC68
▶ Mode 1	FC62
▶ Mode 2	FC64
▶ Bloc_fonctionnel_paramétré_mode2, NIVEAU1	FB2, DB20
▶ Bloc_fonctionnel_paramétré_mode2, NIVEAU2	FB2, DB21
▶ Bloc_fonctionnel_paramétré_mode2, NIVEAU3	FB2, DB16
▶ Bloc_fonctionnel_paramétré_mode2, NIVEAU4	FB2, DB22
▶ Voyants	FC67
▶ Cyclic interrupt	OB30
▶ Sécurité_B5	OB41
▶ Démarrage	OB100



## **Description fonctionnelle :**

### **Choix mode (FC63)**

Cette fonction permet de gérer 2 modes de fonctionnement mode 1 et mode 2.

Les descriptions du mode 1 (FC62) et du mode 2 (FC64) sont évoquées ci-dessous.

Pour obtenir le mode 1, il faut que l'interrupteur S11 soit à 1, que le système ne soit pas en mode 2, et que les actions du mode 2 soient terminées (Pompe de remplissage ou vanne de vidange).

De même que pour obtenir le mode 2, il faut que l'interrupteur S11 soit à 0, que le système ne soit pas en mode 1, et que les actions du mode 1 soient terminées (Pompe de remplissage ou vanne de vidange).

### **Mode 1 (FC62)**

Si le mode 1 est validée, l'opérateur doit pouvoir choisir :

- Un remplissage du silo en appuyant sur le bouton-poussoir S5. Le silo doit alors se remplir au maximum jusqu'au changement d'état du capteur B4 grâce à la pompe M1.
- Une vidange du silo en appuyant sur le bouton-poussoir S0. Le silo doit alors se vider au minimum jusqu'au changement d'état du capteur B1 grâce à la vanne Y1.

Remarque : Les processus de remplissage et de vidange ne peuvent fonctionner simultanément.

### **Mode 2 (FC64 et FB2)**

Le niveau de remplissage doit être réglé suivant le niveau sélectionné par l'action d'un des boutons poussoir S1, S2, S3 ou S4.

- S1 correspondant à une consigne de niveau fixée à 25% de la contenance totale du silo.
- S2 correspondant à une consigne de niveau fixée à 50% de la contenance totale du silo.
- S3 correspondant à une consigne de niveau fixée à 75% de la contenance totale du silo.
- S4 correspondant à une consigne de niveau fixée à 100% de la contenance totale du silo.

La mesure du niveau du silo (0-100%) est calculée dans la fonction FC68 (description ci-dessous)

En fonction de la comparaison effectuée entre la mesure de niveau et la consigne sélectionnée par un des boutons poussoirs (S1 à S4), le silo se remplira ou se vidangera jusqu'au niveau demandé.

Le programme doit pouvoir traiter qu'une seule demande de consigne à la fois. Si l'opérateur appuie sur plusieurs consignes de niveau, seule la dernière sera prise en compte.

Ce fonctionnement doit être programmé dans un bloc fonctionnel paramétré (FB2). Ce bloc fonctionnel sera ensuite appelé à 4 reprises dans la fonction (FC64) correspondant ainsi aux 4 consignes possibles.

### **Mise à l'échelle capteur de niveau (FC68)**

Cette fonction doit permettre de mettre à l'échelle le capteur de niveau analogique. Le silo vide devra correspondre à la valeur 0% et le silo plein (capteur B4) devra correspondre à la valeur 100%.

### **Commande des actionneurs (FC66)**

Cette fonction doit permettre de commander les deux sorties TOR du système : la pompe M1 et la vanne Y1.

Les différentes équations combinatoires permettant de piloter ces deux actionneurs doivent se trouver exclusivement dans cette fonction. On rappelle que M1 et Y1 ne peuvent pas être actionnées simultanément.

### **Fonctionnement des voyants (FC67)**

Deux voyants sont présents sur le pupitre opérateur ;

Le voyant H1 est réservé au fonctionnement de la pompe M1, et H2 pour Y1.

Voyant H1 :

- Si la pompe est en marche selon le mode 1, le voyant H1 clignotera à 0.5Hz.
- Si la pompe est en marche selon le mode 2, le voyant H1 clignotera à 10Hz.

Voyant H2 :

- Si la vanne est en marche selon le mode 1, le voyant H2 clignotera à 0.5Hz.
- Si la vanne est en marche selon le mode 2, le voyant H2 clignotera à 0.5Hz.

### **Niveau de sécurité B5 (OB41)**

Lors d'un remplissage si le niveau venait atteindre le capteur B5, la pompe devrait aussitôt être mise hors service.

La pompe devrait pouvoir redémarrer uniquement si le capteur B5 est à nouveau libéré.

Cette condition de fonctionnement devra être gérée dans un OB d'interruption de programme par hardware.

Cette interruption se fera sur le capteur B5.

### **Rampe consigne moteur de pompe M1 (OB30)**

La pompe M1 doit fonctionner selon l'option 2 (AUTO). Par conséquent, l'API doit fournir une commande TOR (traitée dans FC66) et une consigne de vitesse analogique 0-100%.

On souhaite élaborer une rampe de mise en vitesse.

La consigne de vitesse doit passer de 0% à 100% en 3s, par incrément de 0.2%.

Cette fonctionnalité devra être programmée dans un OB d'interruption de programme cadencée.

### **Procédure de démarrage API (OB100)**

Lors du démarrage de la, CPU (passage RUN-STOP), les deux modes de fonctionnement « mode 1 » et « mode 2 » doivent être désactivés.

La consigne de vitesse de la pompe M1 doit être à 0%.

## Travail demandé :

### PROJET + CONFIGURATION

1. Créer un projet TIA PORTAL.
2. Effectuer la configuration matérielle et réseau PN de l'API S71200 de votre poste de travail.

**Attention à bien vérifier le firmware de la CPU et l'adresse IP attribuée au coupleur Profinet.**

3. Déclarer les entrées sorties câblées de l'API au regard du système à piloter.
4. Compiler, charger et tester
5. Tester vos entrées sorties grâce à une table de visualisation.

### FC63

1. Créer la fonction (LADDER)

 Choix\_mode [FC63]

2. Déclarer de nouvelles variables nécessaires à la programmation de cette fonction.
3. Programmer cette fonction.
4. Compiler, charger et tester.

**Remarque : cette fonction sera susceptible d'être modifiée au fur et à mesure de l'avancement de la programmation**

### FC62

1. Créer la fonction (LADDER)

 Mode1 [FC62]

2. Déclarer de nouvelles variables nécessaires à la programmation de cette fonction.
3. Programmer cette fonction.

**Remarque : cette fonction sera susceptible d'être modifiée au fur et à mesure de l'avancement de la programmation**

### FC66

1. Créer la fonction (LADDER)

 Commande\_actionneurs [FC66]

2. Déclarer de nouvelles variables nécessaires à la programmation de cette fonction.
3. Programmer cette fonction.
4. Compiler, charger et tester. (FC62 et FC66). Pour cet essai, option1 : (interrupteur M1 sur MAN et consigne par potentiomètre

**Remarques : Cette fonction sera susceptible d'être modifiée au fur et à mesure de l'avancement de la programmation.**

### FC68

Vous utiliserez les fonctions NORM\_X et SCALE\_X pour la mise à l'échelle.

1. Créer la fonction (LADDER)

 Mise\_echelle\_capteur [FC68]

2. Dans une table de visualisation, déterminer la valeur numérique correspondant à 100% et à 0% ;
3. Déclarer de nouvelles variables nécessaires à la programmation de cette fonction.
4. Programmer cette fonction.
5. Compiler, charger et tester.

## FB2 et FC64

1. Créer le bloc fonctionnel et la fonction (LADDER)

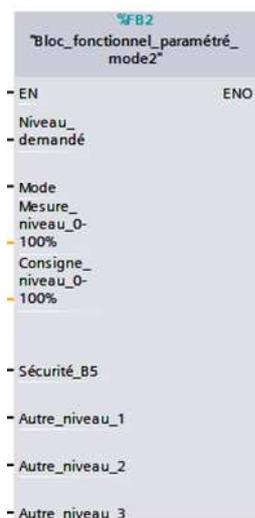
 Bloc\_fonctionnel\_paramétré\_mode2 [FB2]

 Mode2 [FC64]

2. Déclarer les variables locales de FB2 (interface du bloc)

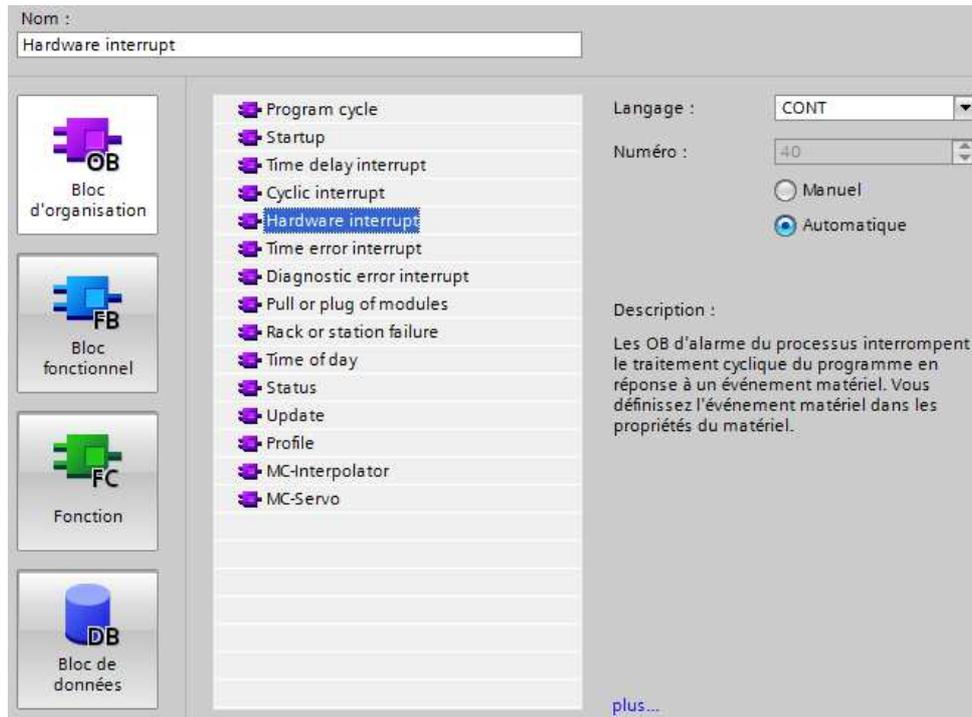
Nom	Type de données
▼ Input	
■ Niveau_demandé	Bool
■ Mode	Bool
■ Mesure_niveau_0-100%	Real
■ Consigne_niveau_0-1...	Real
■ Sécurité_B5	Bool
■ Autre_niveau_1	Bool
■ Autre_niveau_2	Bool
■ Autre_niveau_3	Bool
▼ Output	
■ <Ajouter>	
▼ InOut	
■ <Ajouter>	
▼ Static	
■ valeur_haute	Real
■ valeur_basse	Real
■ copie_consigne_0-10...	Real
■ Auxiliaire_M1_mode2	Bool
■ Auxiliaire_Y1_mode2	Bool
▼ Temp	
■ <Ajouter>	
▼ Constant	
■ <Ajouter>	

3. Programmer ce bloc fonctionnel en associant les variables locales (#). Voir annexe (algorithme FB2)
4. Appeler un FB2 dans la fonction FC2 et associer les variables globales pour chaque niveau de consigne.  
Chaque DB d'instance seront nommés respectivement : NIVEAU1, NIVEAU 2, NIVEAU 3, NIVEAU 4.

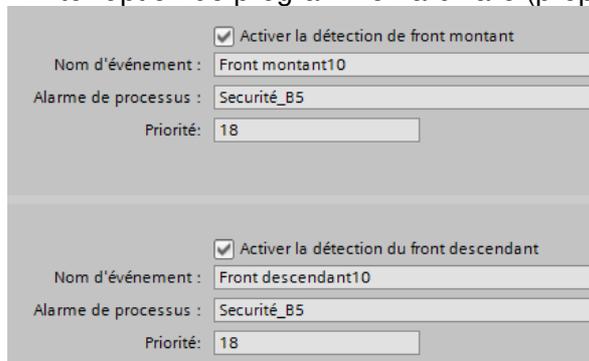


5. Modifier FC63 (verrouillages mode 1) et FC66 (activation des actionneurs).
6. Compiler, charger et tester.

## OB41 alarme de processus (interruption de programme par hardware) :

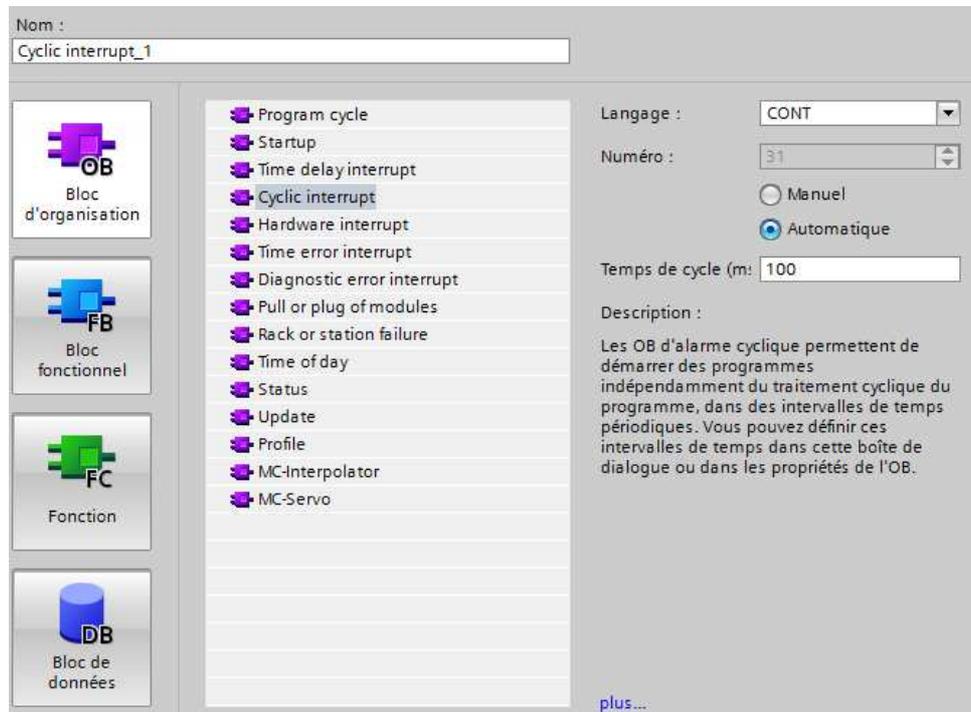


1. Créer le bloc d'organisation OB41 et la fonction (LADDER)  
 Sécurité\_B5 [OB41]
2. Configurer l'interruption de programme hardware (propriétés de la CPU)

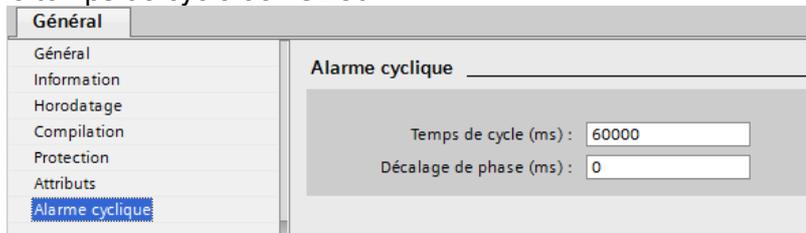


3. Charger la configuration matérielle.
4. Programmer ce bloc d'organisation. Voir annexe (algorithme OB41).
5. Modifier FC63 et FC64 (désactivation des auxiliaires pompe M1).
6. Compiler, charger et tester.

## OB30 alarme cyclique (interruption de programme cadencée) :

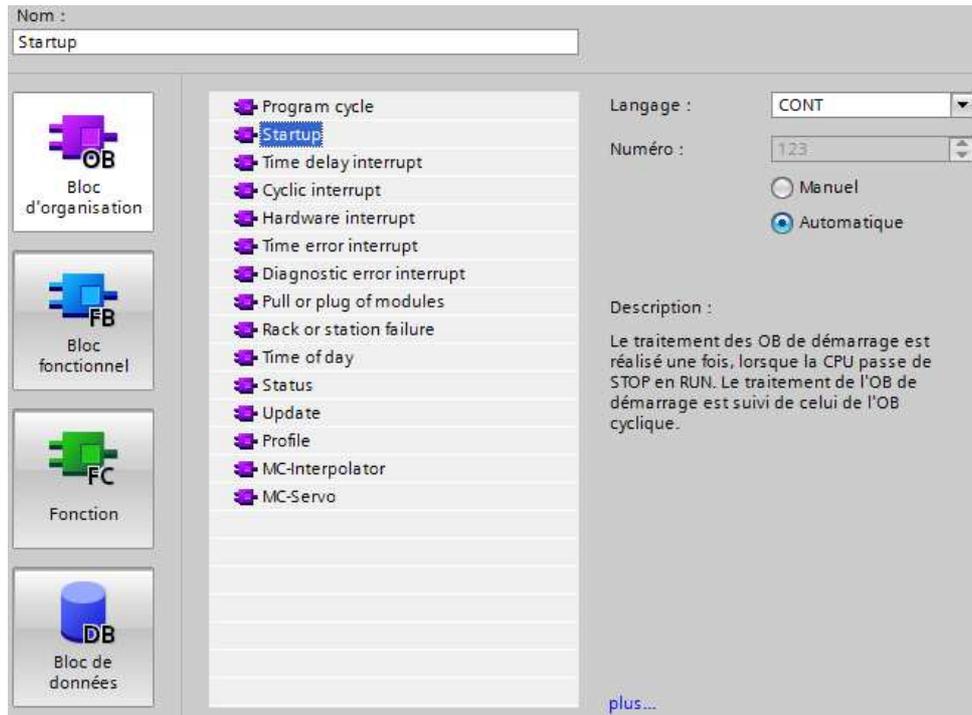


1. Créer le bloc d'organisation OB30 et la fonction (LADDER)  
 Cyclic interrupt [OB30]
2. Calculer le temps de cycle d'appel de l'OB30, afin que la consigne de vitesse doit passer de 0% à 100% en 3s, par incrément de 0.2%.
3. Paramétrer le temps de cycle de l'OB30 :



4. Programmer ce bloc d'organisation. Voir annexe (algorithme OB30)
5. Désactiver l'option 1 : MAN et activer l'option 2 : AUTO
6. Compiler, charger et tester.

## OB30 bloc d'organisation de démarrage :



1. Créer le bloc d'organisation OB100 et la fonction (LADDER)  
 Démarrage [OB100]
2. Programmer ce bloc d'organisation selon la description fonctionnelle.
3. Compiler, charger et tester.