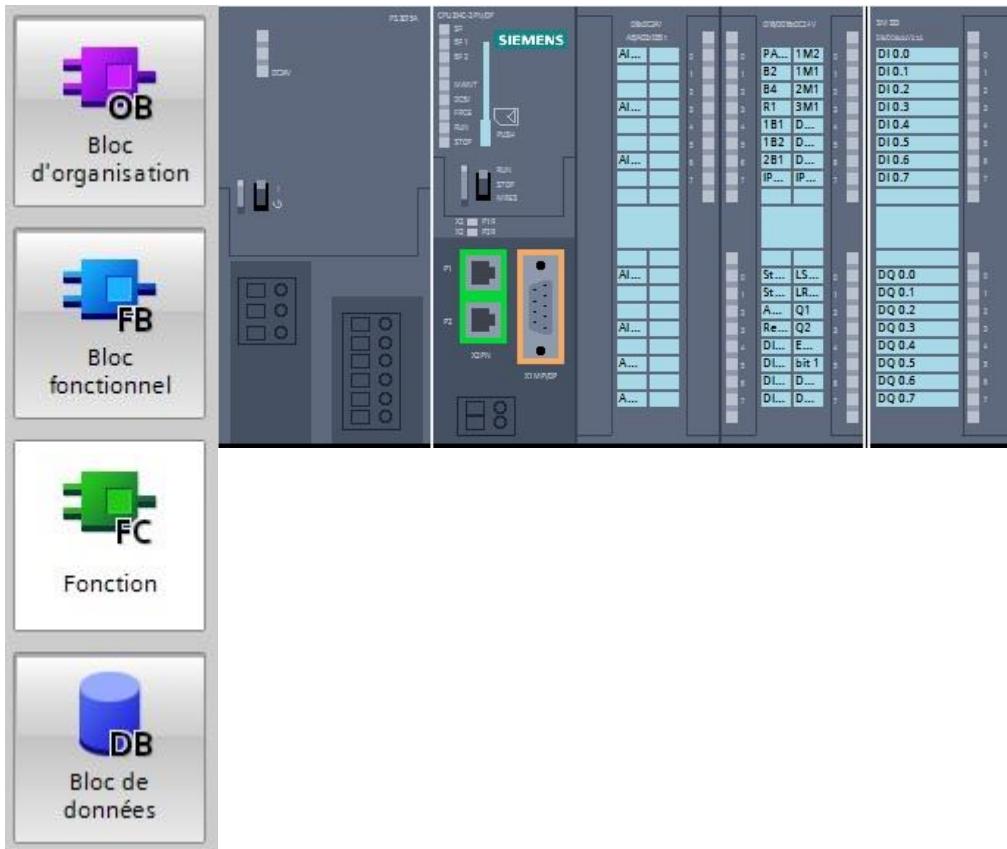


# Atelier C – TIA Portal



## Programmation des automates S7-300 – Introduction au logiciel TIA Portal



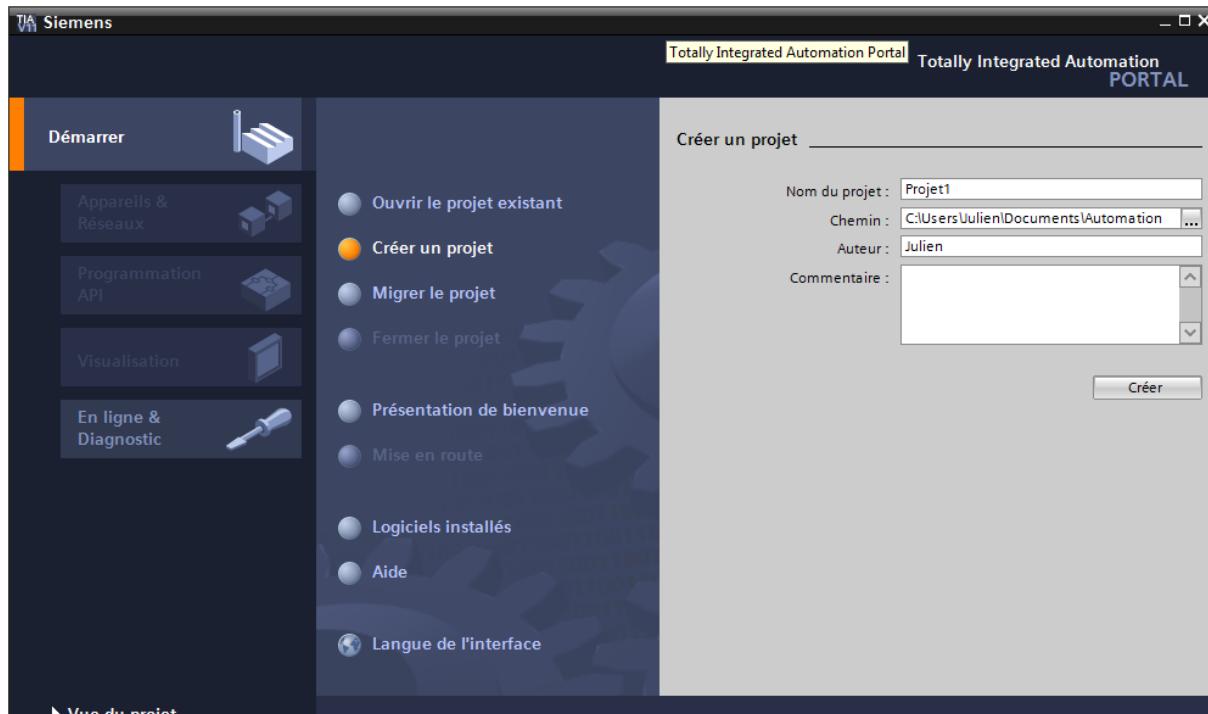
## 1. Introduction

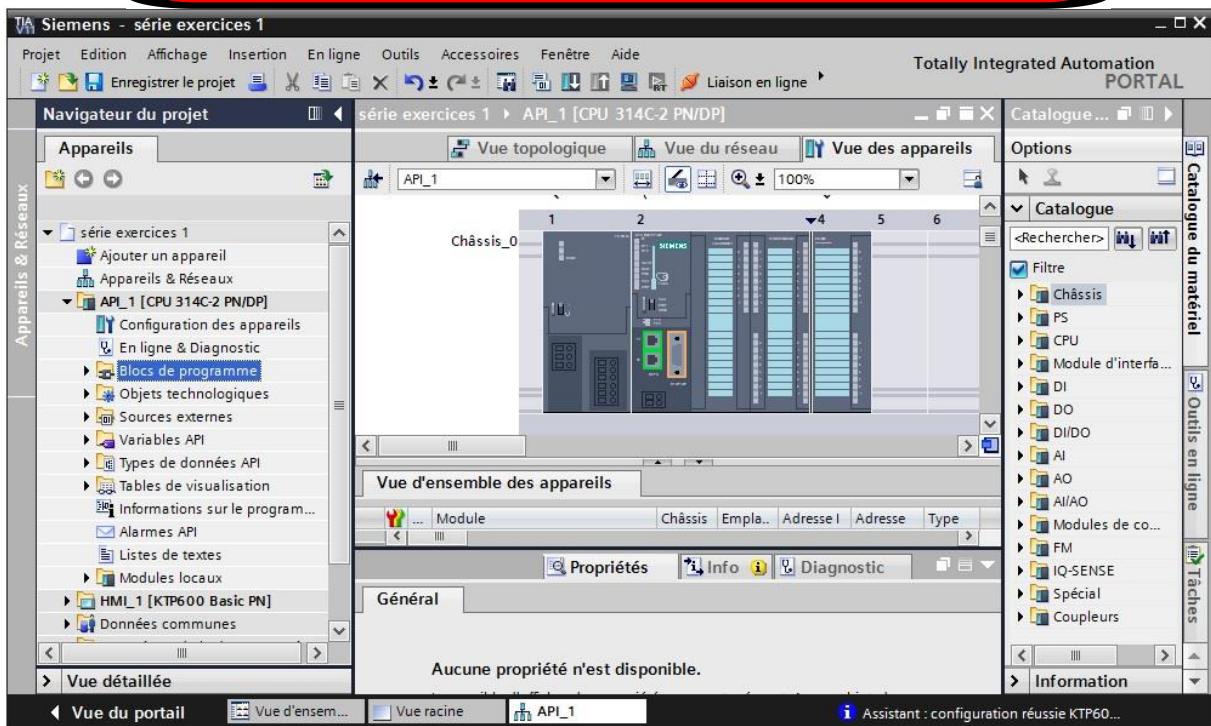
La plateforme Totally Integrated Automation Portal est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant les logiciels SIMATIC STEP 7 V11 et SIMATIC WinCC V11 (dans la version du programme disponible au CTA de Virton).

## 2. Vue du portail et vue du projet

Lorsque l'on lance TIA Portal, l'environnement de travail se décompose en deux types de vue :

- ***La vue du portail*** : elle est axée sur les tâches à exécuter et sa prise en main est très rapide.
- ***La vue du projet*** : elle comporte une arborescence avec les différents éléments du projet. Les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue.

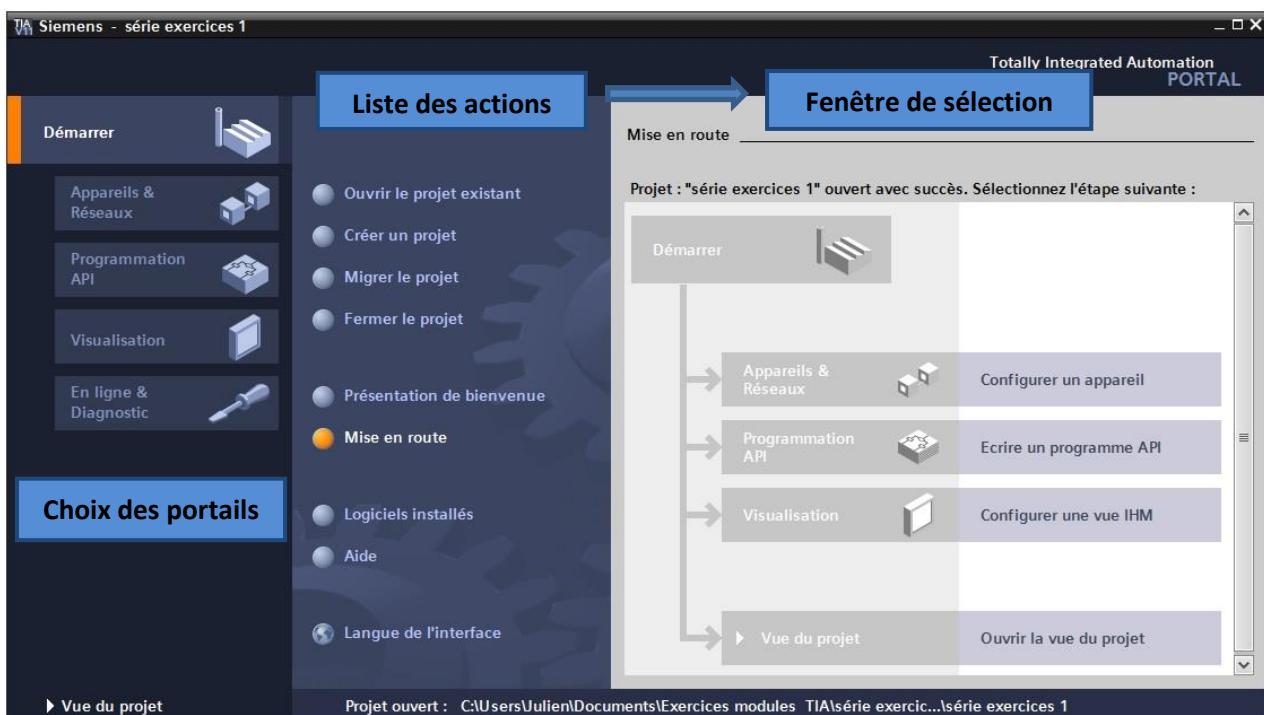




**Vue du projet**

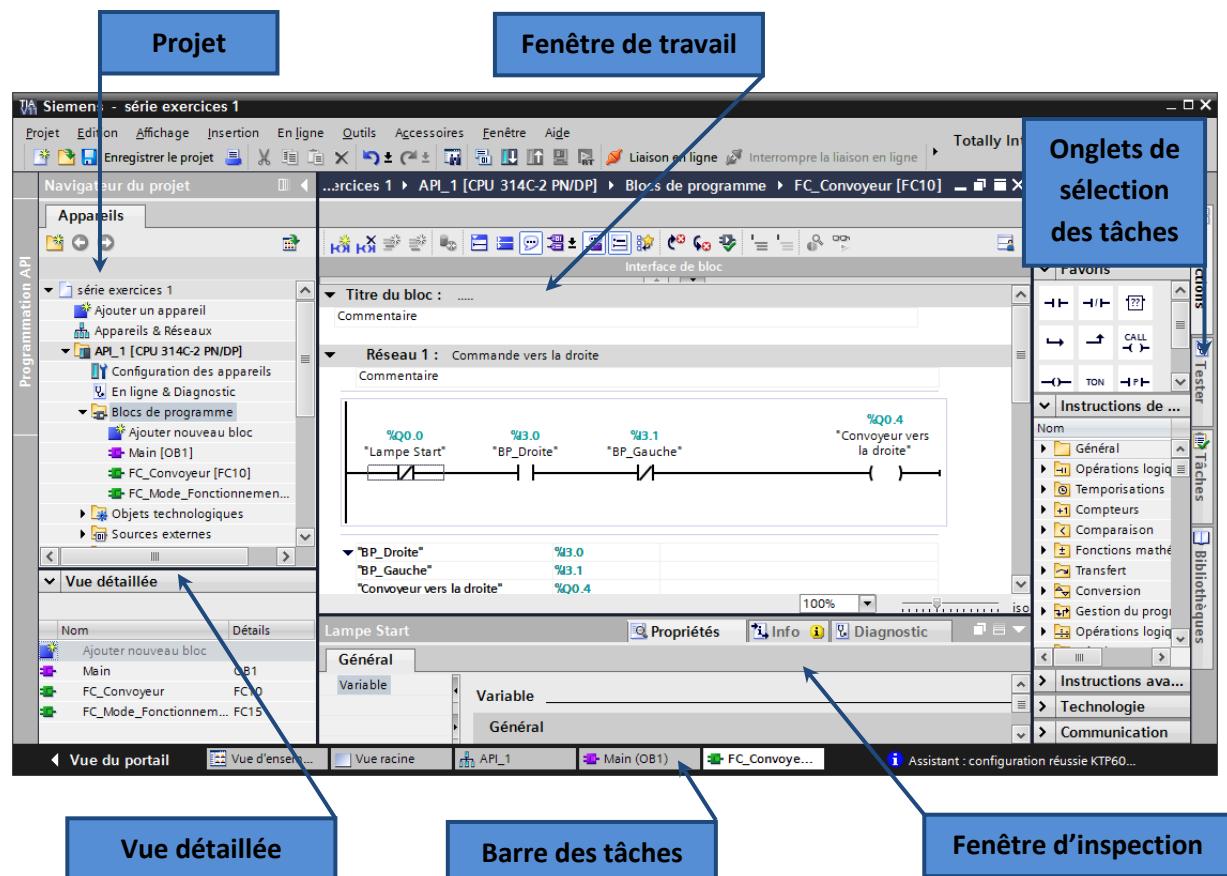
## 2.1 Vue du portail

Chaque portail permet de traiter une catégorie de tâche (actions). La fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionnée.



## 2.2 Vue du projet

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée.



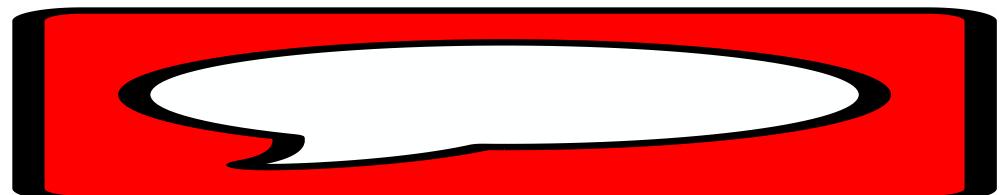
La **fenêtre de travail** permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des HMI,...

La **fenêtre d'inspection** permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme,...).

Les **onglets de sélection de tâches** ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle → bibliothèques des composants, bloc de programme → instructions de programmation).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas.

Il est également possible de redimensionner, réorganiser, désancker les différentes fenêtres.



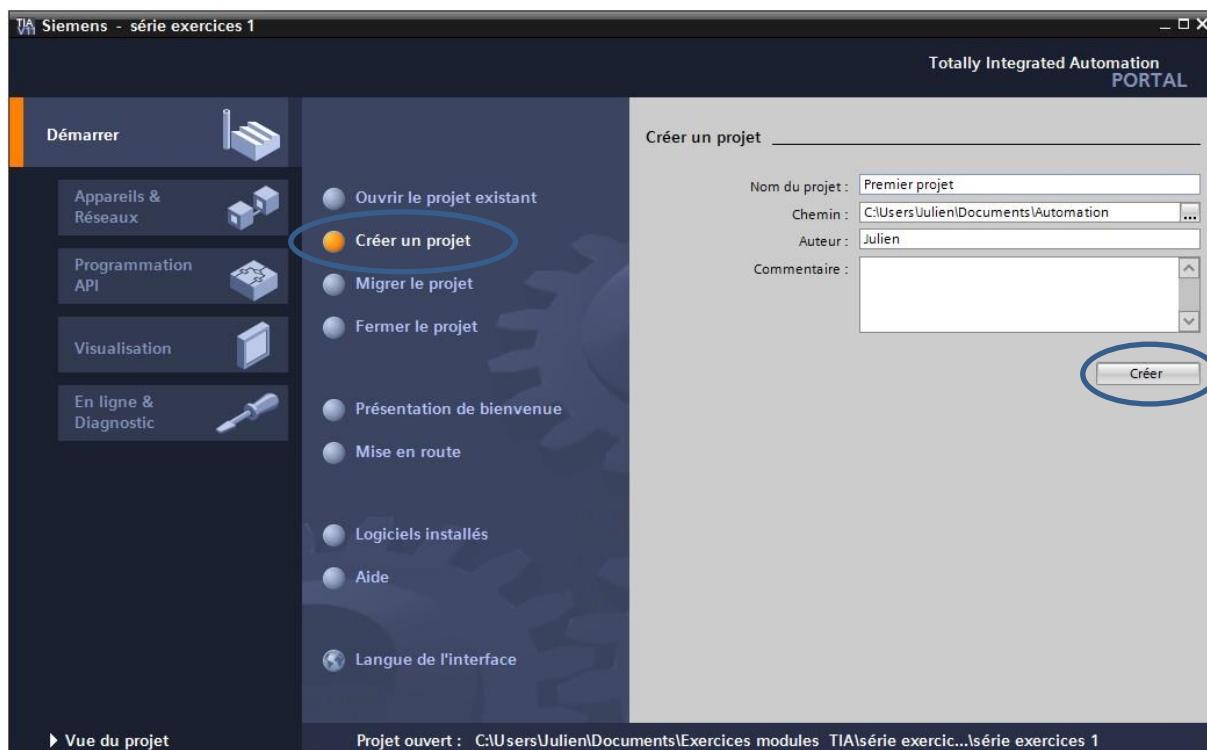
### 3. Création d'un projet et configuration d'une station de travail

#### 3.1 Crédit d'un projet

Pour créer un projet dans la vue du portail, il faut sélectionner l'action « **Créer un projet** ».

On peut donner un nom au projet, choisir un chemin où il sera enregistré, indiquer un commentaire ou encore définir l'auteur du projet.

Une fois que ces informations sont entrées, il suffit de cliquer sur le bouton « **créer** »

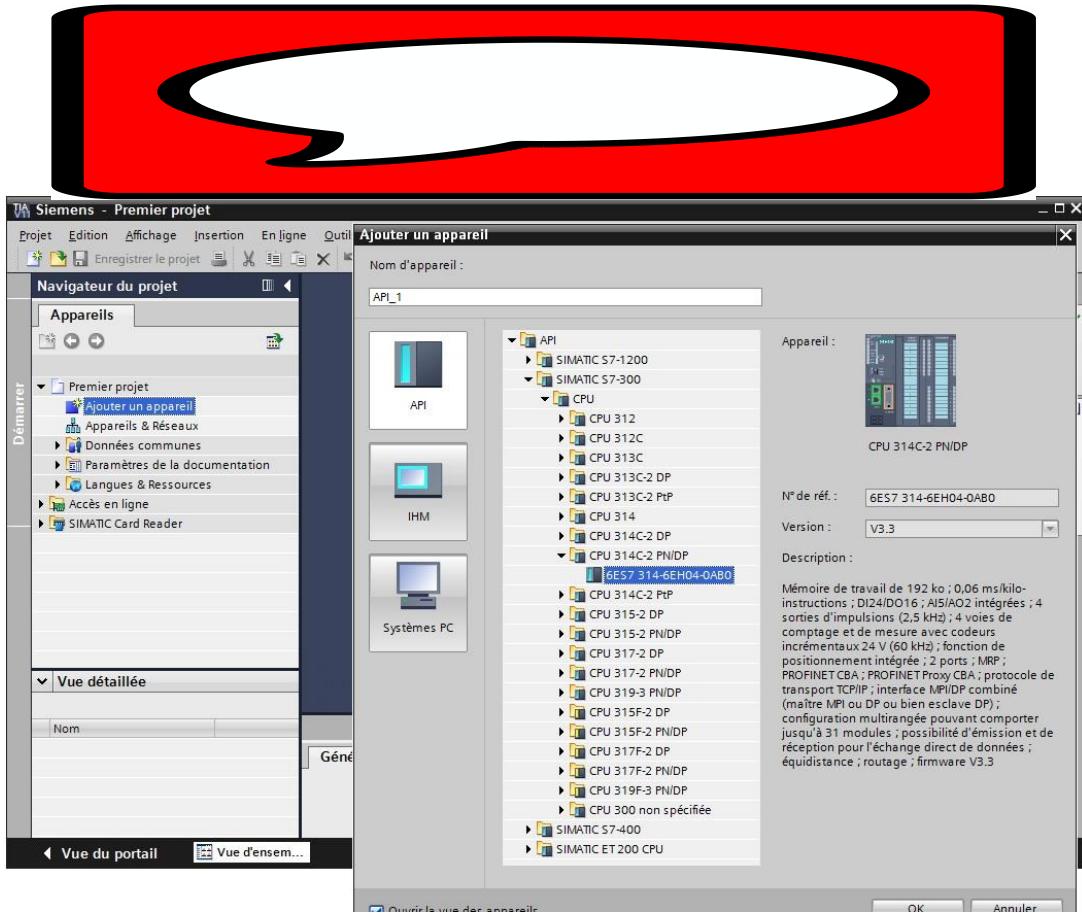


#### 3.2 Configuration et paramétrage du matériel

Une fois votre projet créé, on peut configurer la station de travail.

La première étape consiste à définir le matériel existant. Pour cela, on peut passer par la **vue du projet** et cliquer sur « **ajouter un appareil** » dans le navigateur du projet.

La liste des éléments que l'on peut ajouter apparaît (API, HMI, système PC). On commencera par faire le choix de notre CPU pour ensuite venir ajouter les modules complémentaires (alimentation, E/S TOR ou analogiques, module de communication AS-i,...).



Les modules complémentaires de l'API peuvent être ajoutés en utilisant le catalogue. Si l'on veut ajouter un écran ou un autre API, il faut repasser par la commande « ajouter un appareil » dans le navigateur du projet.

Lorsque l'on sélectionne un élément à insérer dans le projet, une description est proposée dans l'onglet information.

**Contient tous les éléments que l'on peut ajouter à la CPU**

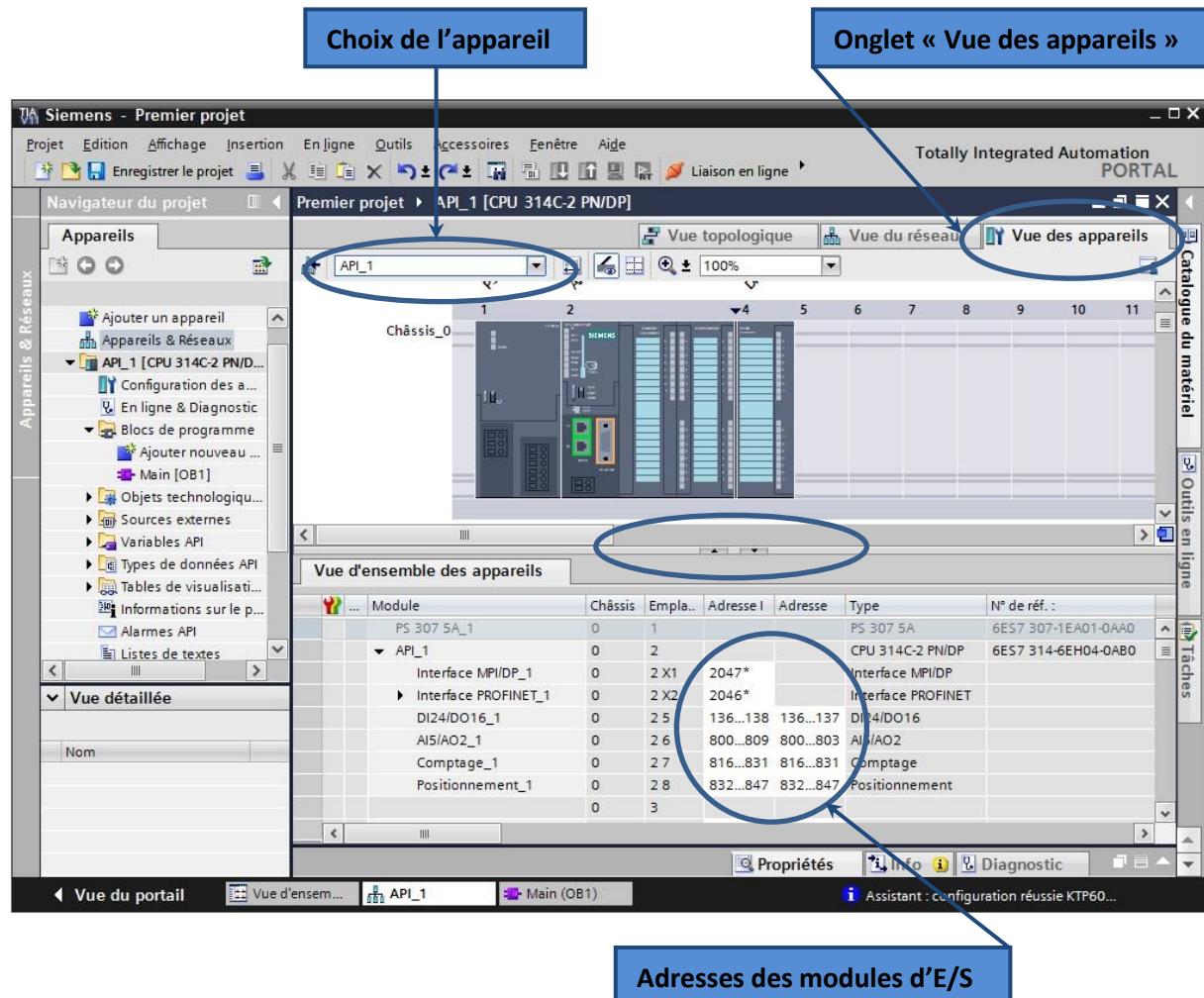
Permet d'avoir une information sur le matériel sélectionné



### 3.3 Adressage des E/S

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matériel, il faut aller dans « **appareil et réseau** » dans le navigateur du projet.

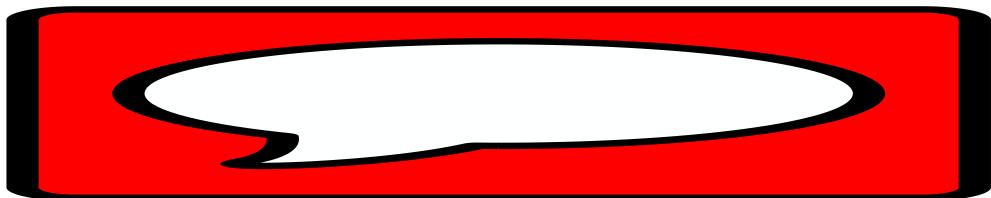
Dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet « **Vue des appareils** » et de sélectionner l'appareil voulu.



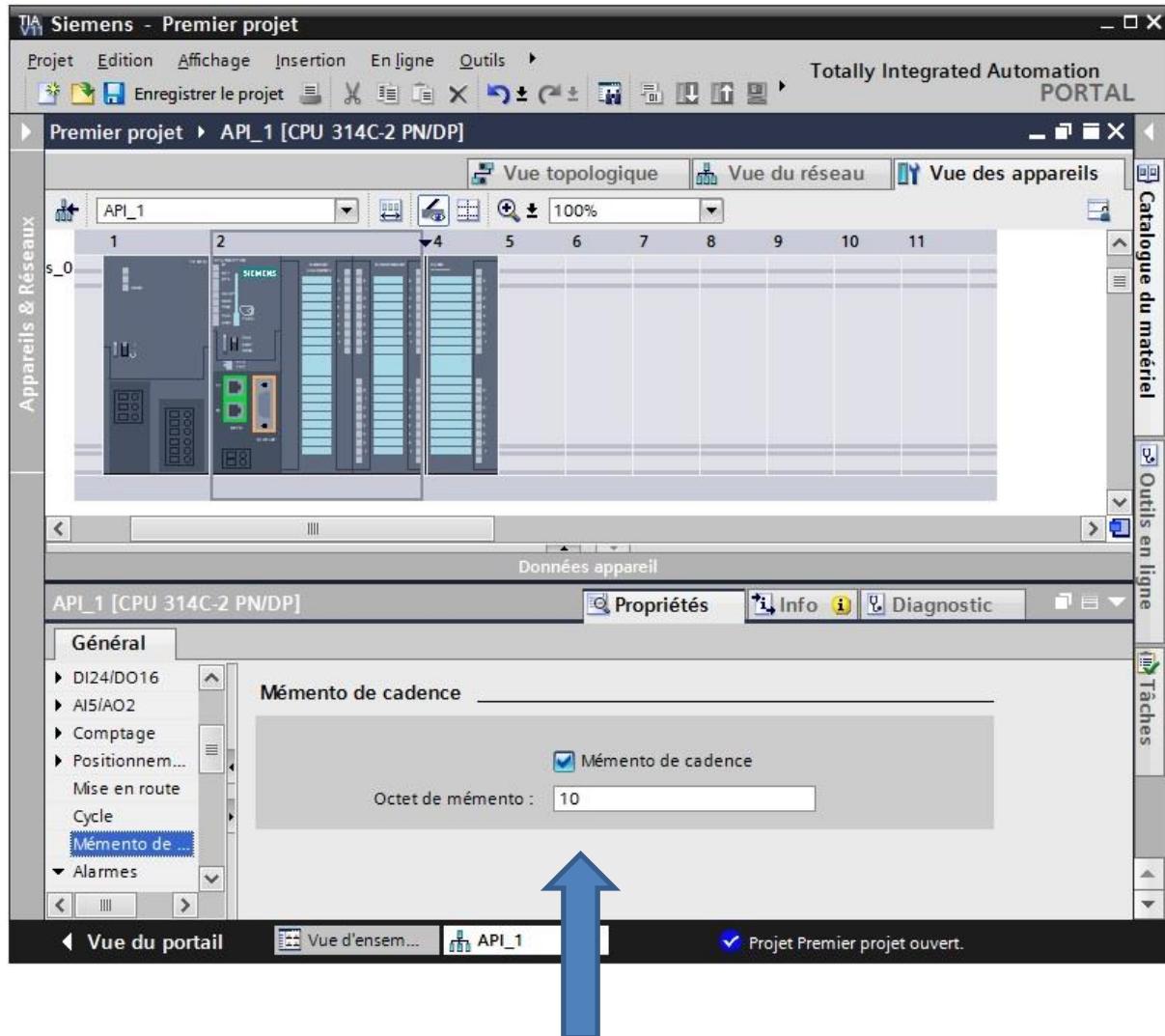
On sélectionne la CPU puis à l'aide des deux petites flèches (voir figure), on fait apparaître l'onglet « **Vue d'ensemble des appareils** » (n'hésitez pas à masquer certaines fenêtres et à en réorganiser d'autres). Les adresses des entrées et sorties apparaissent. Vous pouvez les modifier en entrant une nouvelle valeur dans la case correspondante.

### 3.4 Memento de cadence

Une fois la CPU déterminée, on peut définir le memento de cadence. Pour cela, on sélectionne la CPU dans la fenêtre « **Vue des appareils** » et l'onglet « **propriété** » dans la fenêtre d'inspection.



Dans le menu « **Général** », choisir l'option « **Memento de cadence** », cocher la case « **Memento de cadence** » et choisir l'octet du memento de cadence que l'on va utiliser.

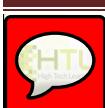


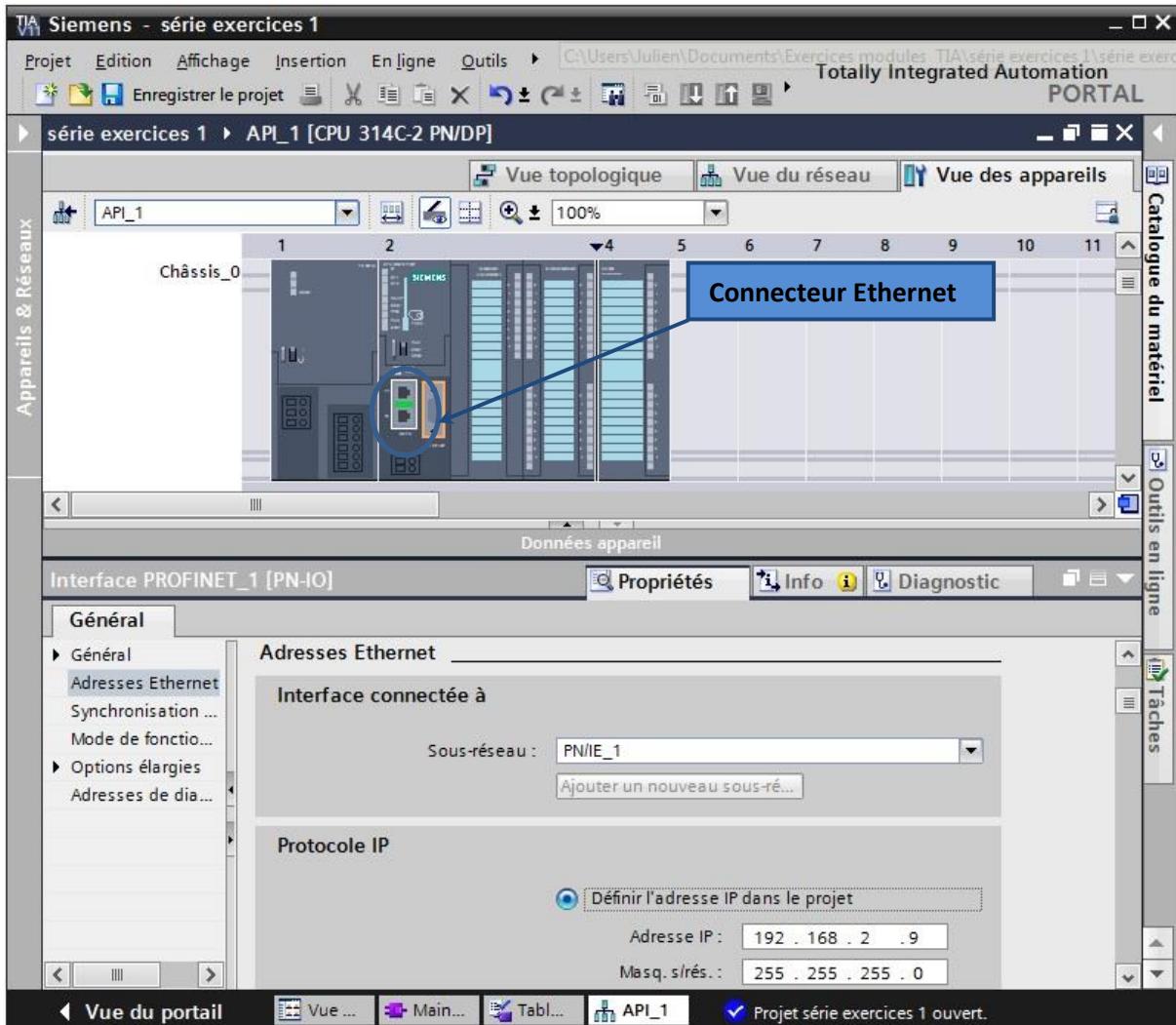
Bit de l'octet de mémento de cadence	7	6	5	4	3	2	1	0
Période [s]	2	1.6	1	0.8	0.5	0.4	0.2	0.1
Fréquence [Hz]	0.5	0.625	1	1.25	2	2.5	5	10

### 3.5 Adresse Ethernet de la CPU

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet. Un double clic sur le connecteur Ethernet de la station fait apparaître la fenêtre d'inspection permettant de définir ses propriétés.

Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même réseau. On utilisera comme adresse pour l'automate 192.168.2.n°de l'automate.





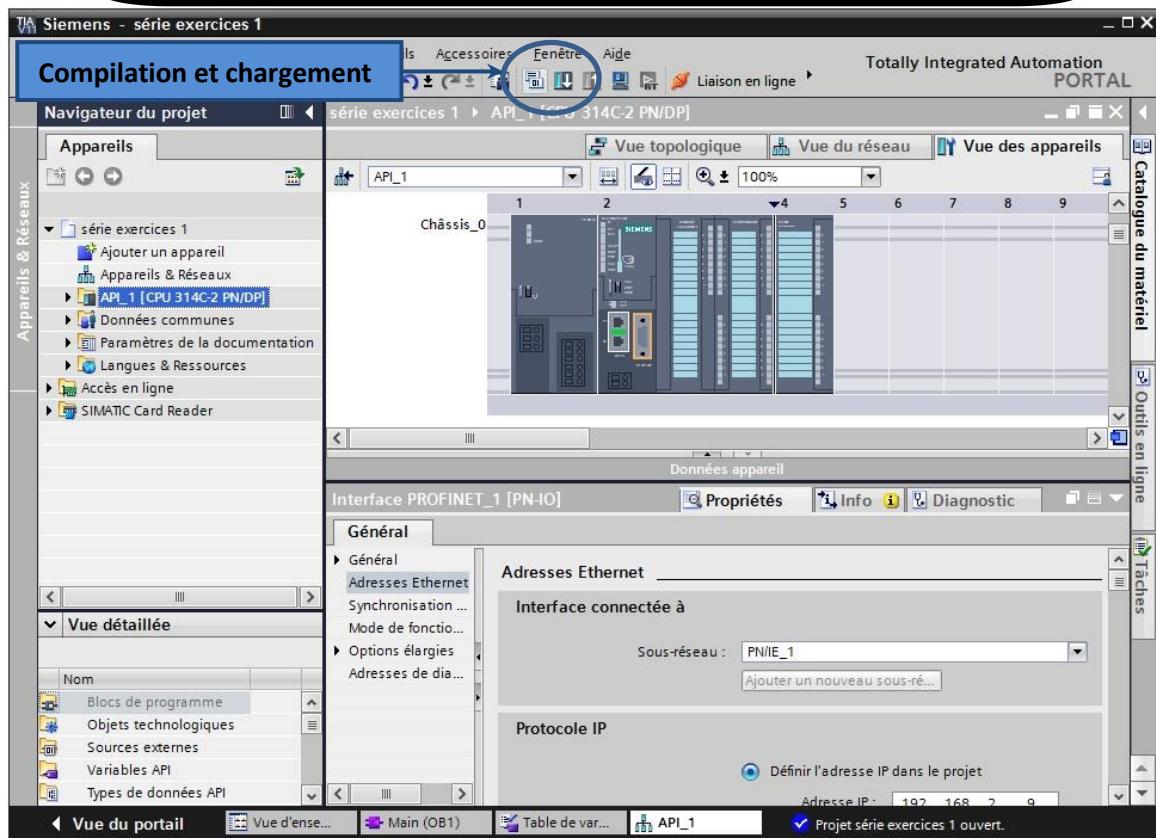
### 3.6 Compilation et chargement de la configuration matérielle

Une fois la configuration matérielle réalisée, il faut la compiler et la charger dans l'automate.

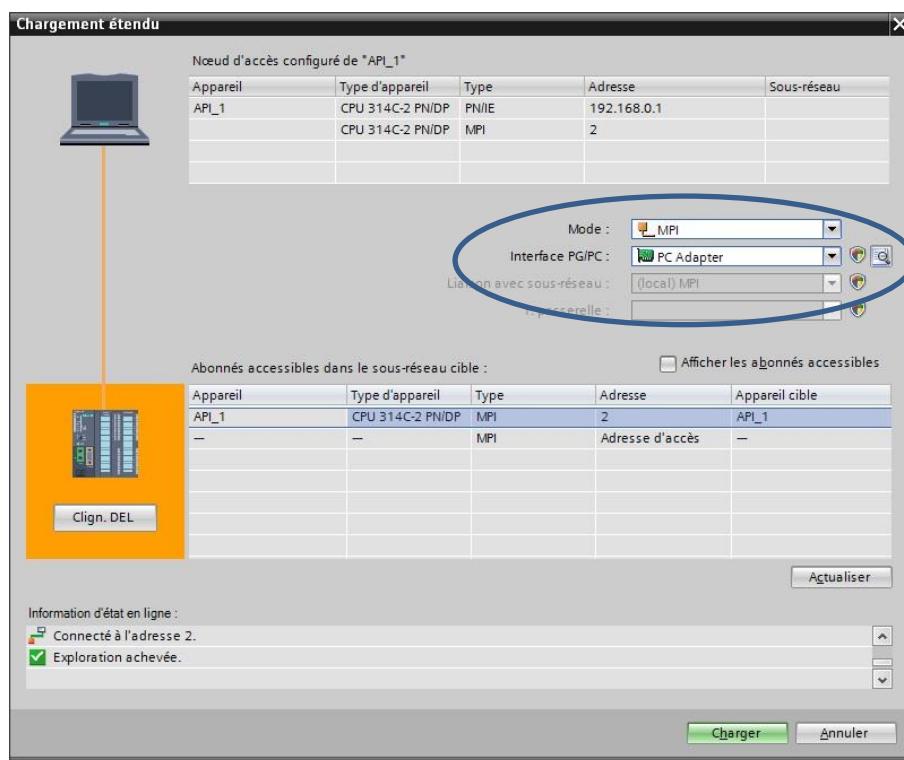
La compilation se fait à l'aide de l'icône « **compiler** » de la barre de tâche. On sélectionne l'API dans le projet puis cliquer sur l'icône « **compiler** ».

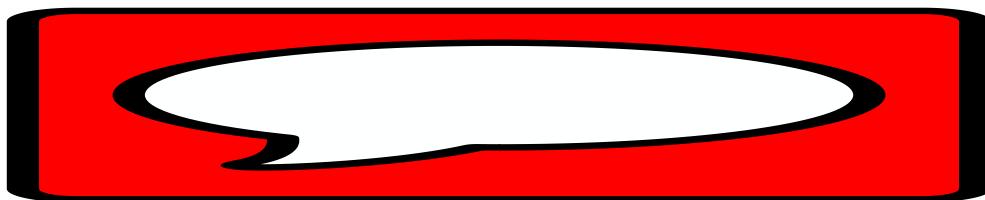
En utilisant cette manière, on effectue une compilation matérielle et logicielle.

Une autre solution pour compiler est de faire un clic droit sur l'API dans la fenêtre du projet et de choisir l'option « Compiler → Configuration matérielle ».



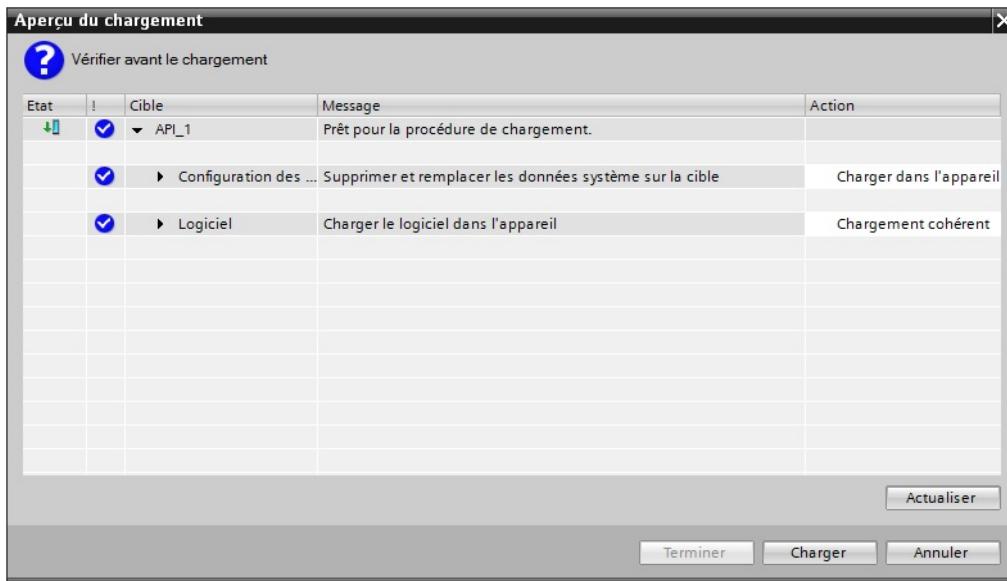
Pour charger la configuration dans l'automate, on effectue un clic sur l'icône « **charger dans l'appareil** ». La fenêtre ci-dessous s'ouvre et vous devez faire le choix du mode de connexion (PN/IE, Profibus, MPI). Si vous choisissez le mode PN/IE, l'API doit posséder une adresse IP.



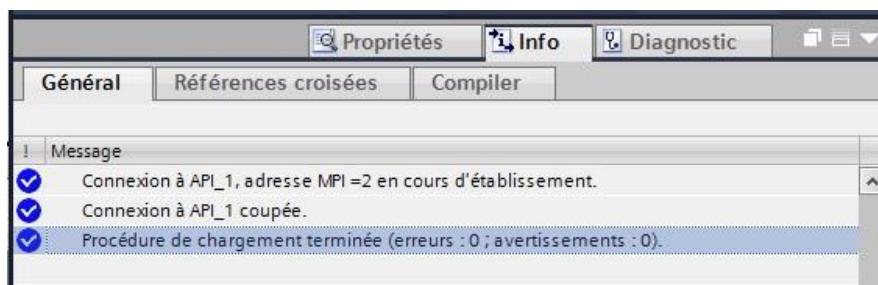


Pour une première connexion ou pour charger l'adresse IP désirée dans la CPU, il est plus facile de choisir le mode de connexion MPI et de relier le PC à la CPU via le « **PC Adapter** ».

Si le programme trouve un appareil, il figure dans la liste en bas de la fenêtre. La touche « **Clign. DEL** » permet de faire clignoter une LED sur la face avant de l'appareil afin de s'assurer que l'on est connecté à l'appareil désiré.



Une fois la configuration terminée, on peut charger le tout dans l'appareil. Des avertissements / confirmations peuvent être demandés lors de cette opération. Si des erreurs sont détectées, elles seront visibles via cette fenêtre. Le programme ne pourra pas être chargé tant que les erreurs persistent.

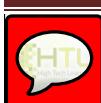


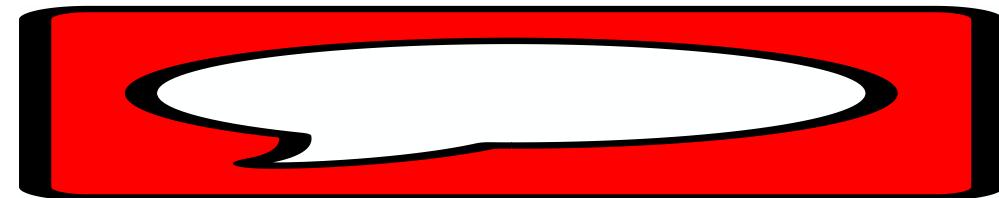
### 3.7 Exercice : configuration d'une station de travail

Lancer TIA Portal et créer un projet nommé « Configuration station ». Réaliser la configuration matérielle de l'automate mis à votre disposition.

En plus de la configuration matérielle, on demande :

- De nommer votre station « API\_X » où X est le numéro de votre automate.
- De définir un memento de cadence (octet 10)
- De faire débuter les adresses des E/S liées à la CPU à 0
- De faire débuter les adresser des E/S du module DI/DO supplémentaire à 3





## 4. Les variables API

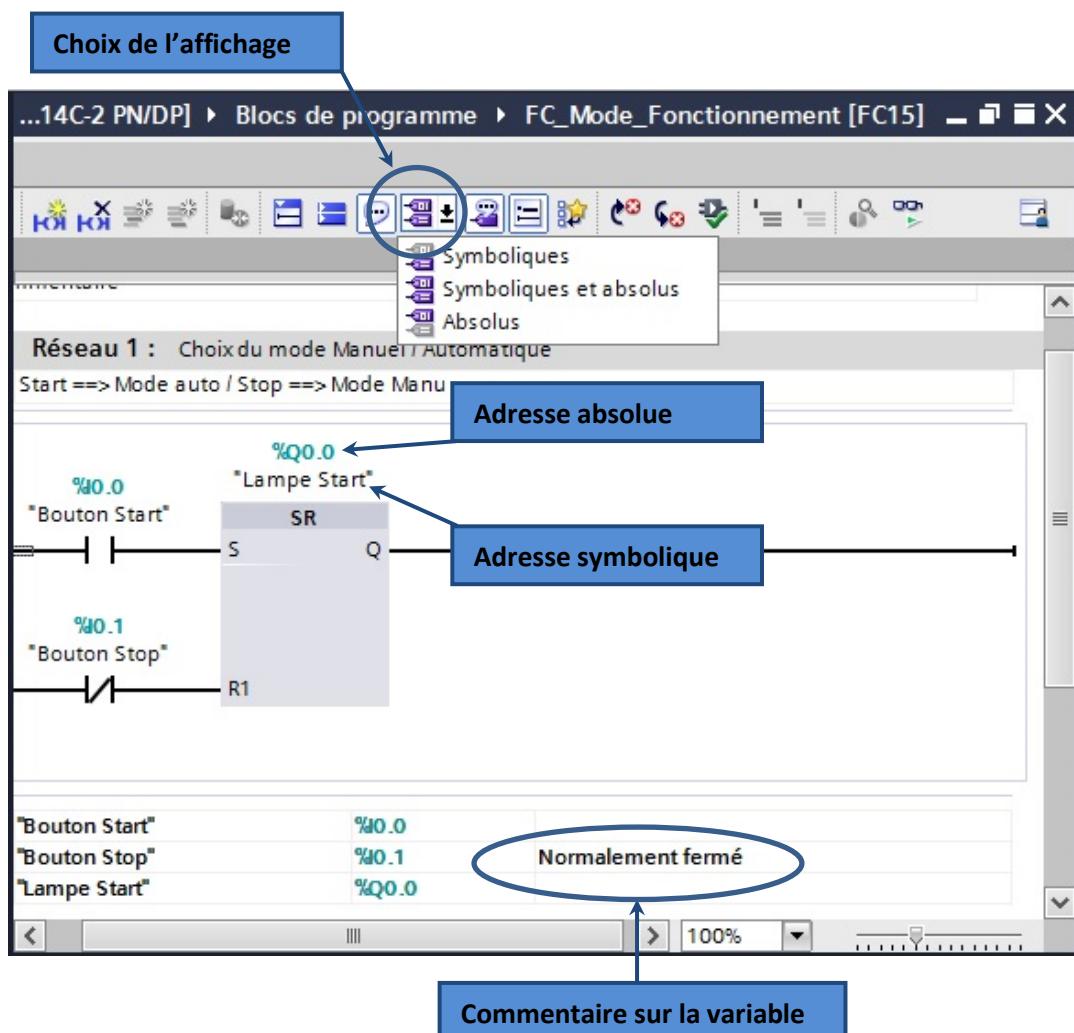
### 4.1 Adresses symbolique et absolue

Dans TIA Portal, toutes les variables globales (entrées, sorties, mémentos,...) possède une **adresse symbolique** et une **adresse absolue**.

- **L'adresse absolue** représente l'identificateur d'opérande (I, Q, M,...) et son adresse et numéro de bit.
- **L'adresse symbolique** correspond au nom que l'utilisateur a donné à la variable (ex : Bouton\_Marche).

Le lien entre les adresses symbolique et absolue se fait dans **la table des variables API**.

Lors de la programmation, on peut choisir d'afficher les adresses absolues, symboliques ou encore les deux simultanément.



## 4.2 Table des variables API

C'est dans la table des variables API que l'on va pouvoir déclarer toutes les variables et les constantes utilisées le programme.

Lorsque l'on définit une variable API, il faut définir :

- Un nom : c'est l'adressage symbolique de la variable.
- Le type de donnée : BOOL, INT,...
- L'adresse absolue : par exemple Q1.5

On peut également insérer un commentaire qui nous renseigne sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable.

The screenshot shows the SIMATIC Manager software interface. On the left, there's a navigation tree under 'Programmation API' for project 'API\_1 [CPU 314C-2 PN/DP]'. The 'Variables API' node is expanded, showing various variables like 'Lampe Start', 'Lampe Reset', 'Lampe Q1', 'Lampe Q2', etc. In the center, a table titled 'Table de variables standard' lists these variables with columns for Nom, Type de données, Adresse, Réma..., Visible..., Acces..., and Commentaire. The 'Lampe Q2' row is selected. A detailed view of 'Lampe Q2' is shown on the right, with tabs for 'Général' and 'Variable'. The 'Général' tab displays the variable's name as 'Lampe Q2', type as 'Bool', address as '%Q0.3', and comment as 'Sur face avant de la station'. The 'Variable' tab shows the variable's name again and its data type as 'Bool'. At the bottom, there are tabs for 'Vue du portail', 'Vue d'ensem...', 'Table de vari...', 'Main (OB1)', 'FC\_Convoye...', and 'FC\_Mode\_Fo...'. A red oval is drawn over the top right corner of the window.

Remarques :

- La commande copier / coller permet de copier une table des variable sur Excel facilement.
- En sélectionnant le coin inférieur droit d'une cellule d'adresse ou d'un nom et en le faisant glisser vers le bas, on peut créer des variables automatiquement (comme dans Excel).
- Visible dans IHM : seules les variables cochées peuvent être visibles lors de la configuration HMI.
- Accessible dans HMI (uniquement S7 – 1200) : permet au pupitre HMI l'accès en ligne aux variables API sélectionnées.



### 4.3 Signalisation des erreurs dans la table des variables

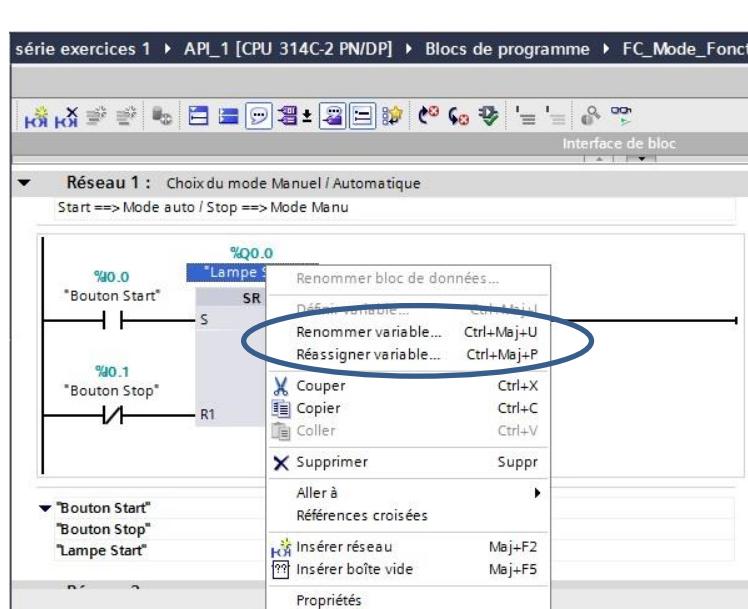
Lorsqu'il y a une erreur de syntaxe dans la table des variables API, celle-ci est signalée en rouge ou en orange. Lorsque l'on sélectionne la case colorée, un message signalant le type d'erreur apparaît.

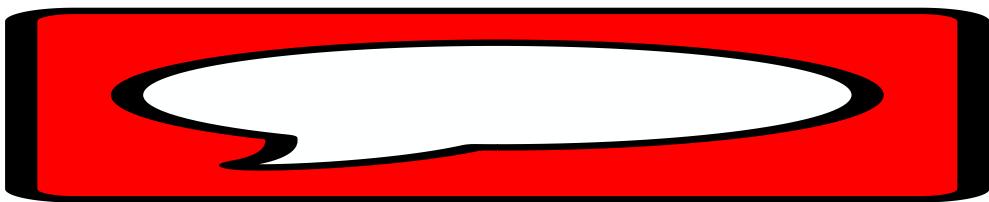
Une table des variables contenant des erreurs peut être enregistrée mais ne pourra pas être compilée et chargée dans l'automate.

	Nom	Type de données	Adresse	Réma...	Visible dans IHM	Accessible depuis IHM	Commentaire
1	Lampe Start	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Lampe Reset	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Lampe Q1	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Lampe Q2	Bool	<b>%Q0.3</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Convoyeur vers la droite	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Convoyeur vers la gauche	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Déplacement pince vers le haut	Int	<b>%Q1.0</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sur face avant de la station
8	Déplacement pince vers le bas	Bool	%Q1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Ouverture pince	Bool	<b>%Q0.3</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Fermeture pince	Bool	%Q1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Rotation pince position 0°	Bool	%Q1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Rotation pince position 180°	Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Lampe Q2(1)	Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Bouton Start	Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Bouton Stop	Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Interrupteur Auto / Manu	Bool	%I0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	Bouton Reset	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Normalement fermé

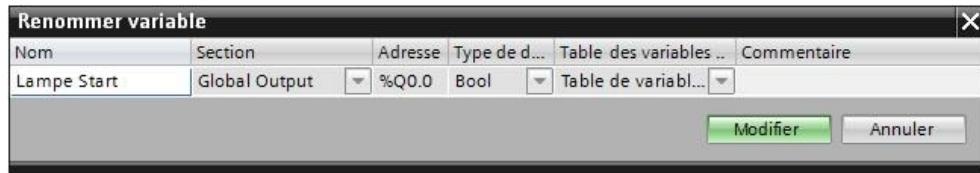
### 4.4 Renommer / réassigner des variables

Il est possible de renommer et réassigner les variables API dans la fenêtre de programmation. Pour cela, il faut faire un clic droit sur la variable en question et choisir l'option renommer ou réassigner.





La fenêtre si dessous s'ouvre.

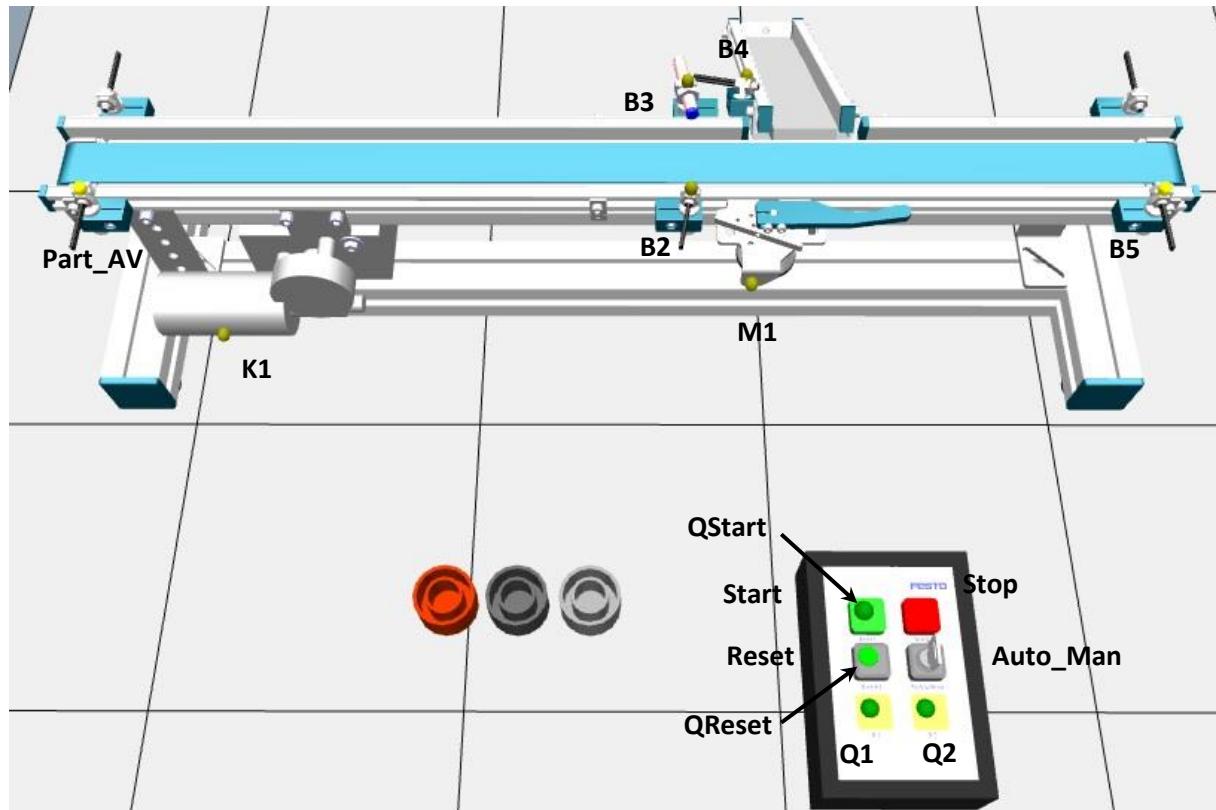


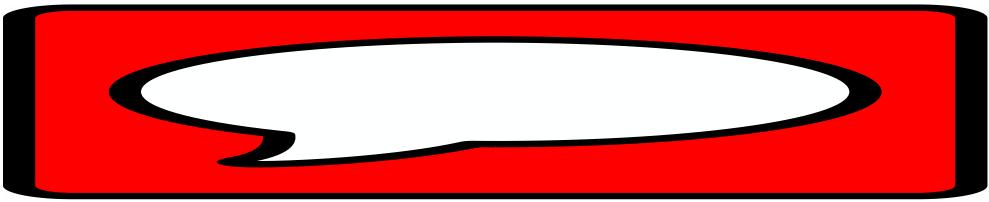
- **Renommer la variable** donne l'accès aux paramètres Nom et Commentaire
- **Réassigner la variable** donne l'accès aux paramètres Section, Adresse, Type de données, Table des variables API et Commentaire.

#### 4.5 Exercice : réalisation d'une table des variables API

Les exercices qui vont suivre seront réalisés à l'aide du logiciel CIROS Mechatronics. Ce logiciel permet de simuler le fonctionnement de stations modulaires FESTO.

Le modèle utilisé se trouve dans la catégorie « Système de transfert » et se nomme « Module de projet Ligne de convoyage 4 »





Symbole	Adresse	Type	Commentaire
K1	Q 0.0	Bool	Marche moteur convoyeur
M1	Q 0.1	Bool	Vérin rotatif
QStart	Q 1.0	Bool	Voyant bouton Start
QReset	Q 1.1	Bool	Voyant bouton Reset
Q1	Q 1.2	Bool	Voyant Q1
Q2	Q 1.3	Bool	Voyant Q2
Part_AV	I 0.0	Bool	Présence pièce entrée convoyeur (NF)
B2	I 0.1	Bool	Présence pièce vérin rotatif
B3	I 0.2	Bool	Détecteur pièce métallique
B4	I 0.3	Bool	Magasin plein
B5	I 0.4	Bool	Présence pièce sortie convoyeur (NF)
Start	I 1.0	Bool	Bouton Start
Stop	I 1.1	Bool	Bouton Stop (NF)
Reset	I 1.3	Bool	Bouton Reset
Auto_Man	I 3.0	Bool	Commutateur Auto / Manu / Vient de PLC Sim

La simulation se fera par l'intermédiaire de PLC Sim. Il n'est donc pas nécessaire de faire une configuration matérielle de toutes les E/S, il faut juste faire le choix d'une CPU.

On vous demande de créer un nouveau projet intitulé « première série d'exercices ». La CPU utilisée dans ce projet sera une CPU 314C-2 PN/DP.

Réaliser une table des variable API reprenant les éléments du tableau ci-dessus en ajoutant à cette table les données suivantes :

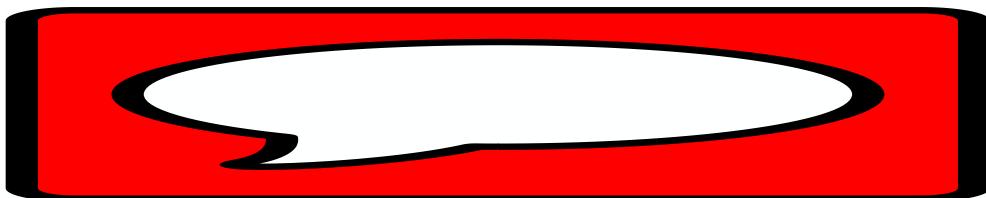
- Un bit de cadence à 1 Hz
- Un bit de cadence à 2 Hz
- Un bit de cadence à 5 Hz

## 5. Les opérations binaires

Les opérations binaires sont les opérations réalisées sur les variables de type « Bool ». On retrouve dans ces opérations les fonctions logiques (ET, OU, OU Exclusif), les bascules RS, les détections de fronts,...

Plusieurs langages de programmation sont disponibles : le schéma à contact (LADDER), les logigrammes (LOG) ou le langage LIST (instructions). Seul le langage Ladder sera utilisé dans ce cours.





## 5.1 Interrogation à 1 et à 0

Les symboles d'interrogation à 1 et à 0 sont très utilisés en Ladder. C'est à partir de ces symbole que l'on va pouvoir réaliser des fonctions logiques, lancer des temporisations, des compteurs,...

- Interrogation à 1 ou Contact NO (à fermeture). Il fournit le résultat d'interrogation 1 lorsque l'opérande interrogé est à l'état 1.
- Interrogation à 0 ou Contact NF (à ouverture). Il fournit le résultat d'interrogation 1 lorsque l'opérande interrogé est à l'état 0.

Le résultat de l'interrogation va être différent en fonction du type de détecteur interrogé. Le tableau ci-dessous montre les différents cas qui peuvent être rencontrés.

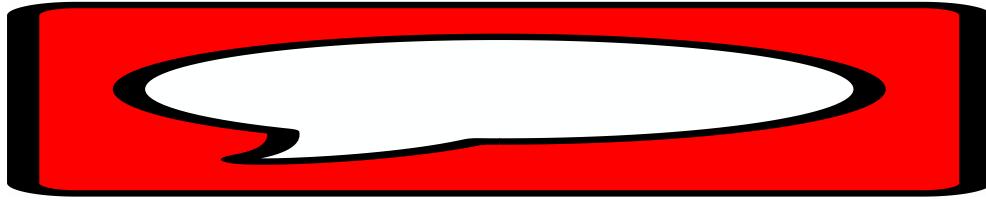
Processus			Evaluation du programme dans l'automate				
Le capteur est un ...	Le capteur est ...	La tension à l'entrée est ...	Etat du signal à l'entrée	Interrogation à "1"		Interrogation à "0"	
				Symbol / instruction	Résultat interrogation	Symbol / instruction	Résultat interrogation
contact NO	activé	présente	1		« oui » 1		« non » 0
	inactivé	absente	0	CONT :	« non » 0	CONT :	« oui » 1
contact NF	activé	absente	0	contact dit « NO »	« non » 0	contact dit « NF »	« oui » 1
	inactivé	présente	1		« oui » 1		« non » 0

## 5.2 Affection / Mise à 1 et mise à 0

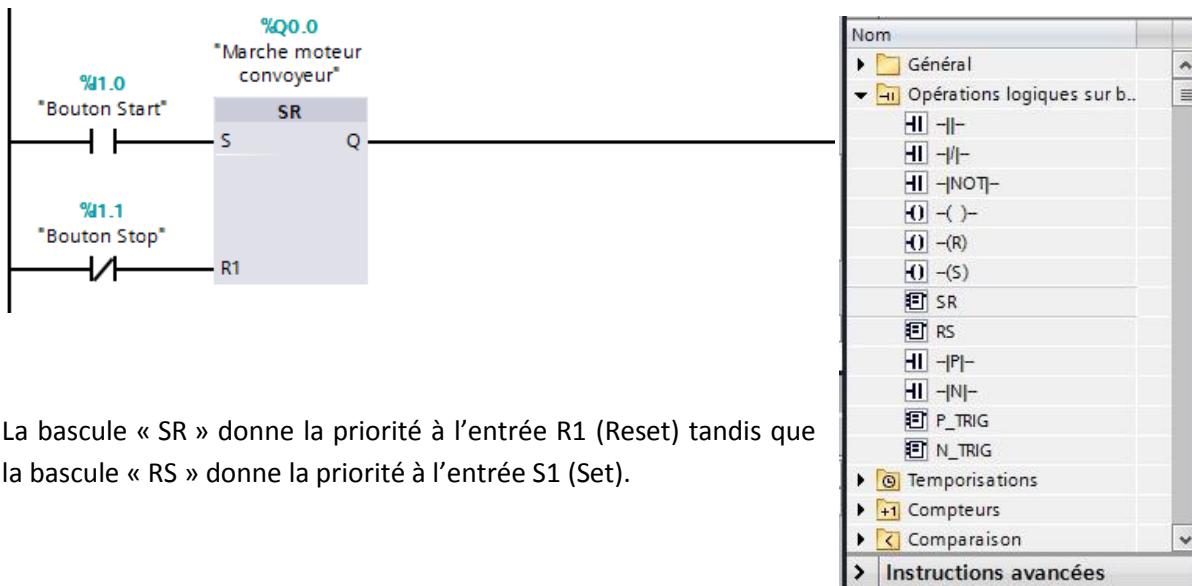
→ L'affection transmet le résultat logique à l'opérande indiquée. Ce résultat affecté reste disponible après l'affection et peut être affecté à un autre opérande ou relié à d'autres opérations

→ Mise à 1 → Si le résultat logique = « 1 », l'opérande indiqué est mis à « 1 » (SET). Si le résultat = 0, l'opérande reste inchangé.

→ Mise à 0 → Si le résultat logique = « 1 », l'opérande indiqué est mis à « 0 » (RESET). Si le résultat = 0, l'opérande reste inchangé.



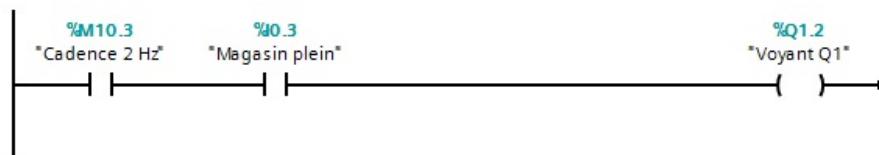
Les instructions SET et RESET peuvent être combinées en utilisant les bascules « SR » et « RS ».



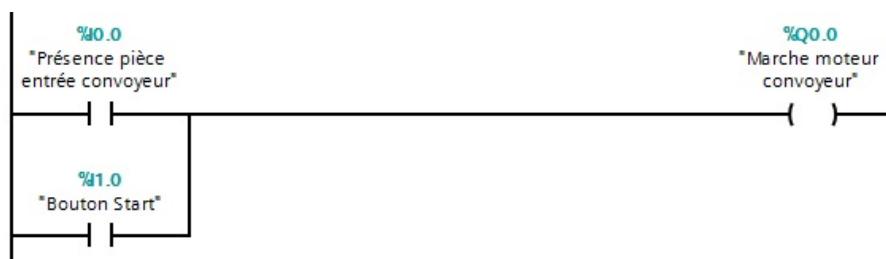
La bascule « SR » donne la priorité à l'entrée R1 (Reset) tandis que la bascule « RS » donne la priorité à l'entrée S1 (Set).

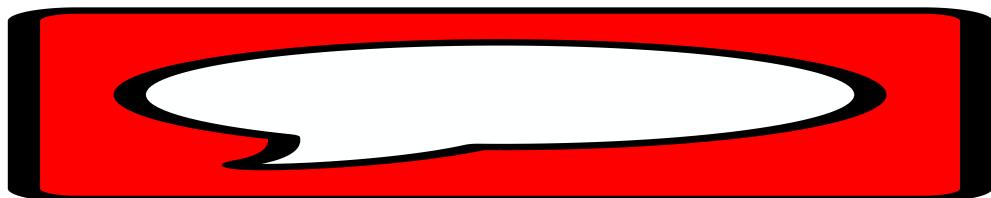
### 5.3 Fonctions logiques ET, OU, OU Exclusif

**Fonction ET**: le résultat logique est égal à « 1 » lorsque toutes les entrées sont à l'état « 1 ». En langage Ladder, cela revient à mettre les contacts en série.

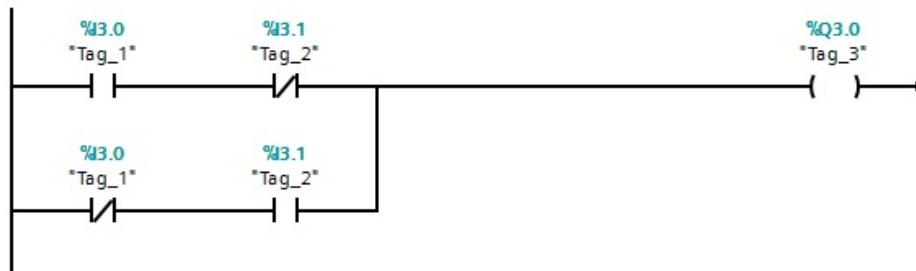


**Fonction OU**: le résultat logique est égal à « 1 » lorsqu'au moins une des entrées est à l'état « 1 ». En langage Ladder, cela revient à mettre les contacts en parallèle.





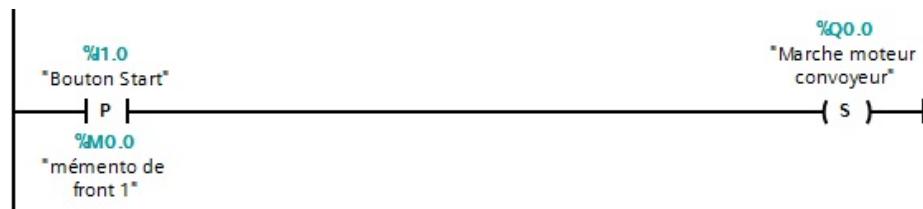
**Fonction OU Exclusif:** le résultat logique est égal à « 1 » lorsqu'une seule des deux entrées est à l'état 1 (cas de deux entrées). En langage Ladder, l'instruction se programme comme indiqué dans la figure ci-dessous.



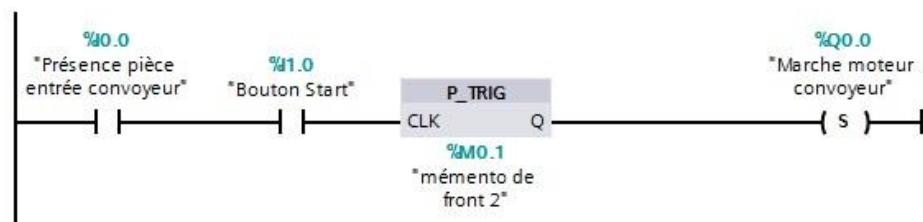
#### 5.4 Détection de front

Il est possible d'évaluer le front d'un signal. Cela permet de détecter le passage de « 0 » à « 1 » (front montant) ou le passage de « 1 » à « 0 » (front descendant) d'un opérande.

Les symboles d'interrogation des fronts montant (P) ou descendant (N) d'un opérande doivent toujours être associées à un memento. Pour chaque détection de front, le memento doit être différent (même pour le même opérande).



La réponse au front d'un résultat logique, comme par exemple les fonctions ET ou OU, se fait à l'aide des instructions « P\_TRIG » et « N\_TRIG ».

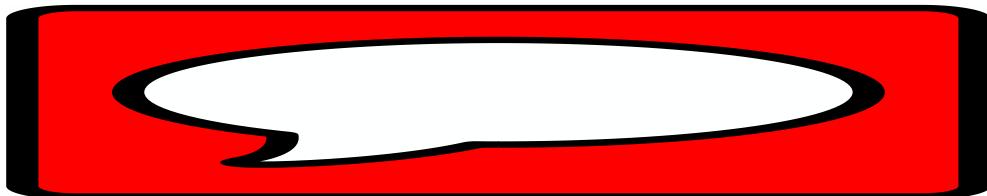


#### 5.5 Exercices : commande de la ligne de convoyage

**Exercice 1 :** lorsque l'on appuie sur le bouton « Start », le convoyeur se met en marche et lorsque l'on appuie sur le bouton « Reset », le vérin rotatif sort.

**Exercice 2 :** une impulsion sur le bouton « Start » démarre le convoyeur et une impulsion sur le bouton « Stop » l'arrête. Le convoyeur ne peut démarrer que si une pièce est présente à l'entrée de celui-ci.





**Exercice 3 :** Le convoyeur démarre lorsqu'une pièce est présente à l'entrée et que l'on appuie sur le bouton Start. Il s'arrête lorsque la pièce franchi la cellule B5 (présence pièce sortie convoyeur).

Lorsque la pièce est présente à l'entrée du convoyeur, le voyant Q1 est allumé. Lorsque le convoyeur est en marche, il clignote à la fréquence de 2 Hz.

**Exercice 4 :** on va distinguer le mode manuel et le mode automatique.

Lorsque le commutateur « Auto\_Man » est sur Manuel (0), le convoyeur et le vérin sont commandés à l'aide des boutons « Start » et « Reset » (impulsions).

Lorsque le commutateur est sur Auto, le convoyeur :

- Démarre si une pièce est présente à l'entrée du convoyeur, que le magasin n'est pas plein et que l'on appuie sur le bouton « Start »
- S'arrête lorsque la pièce a franchi la barrière B5 ou que le mode automatique est désactivé.

De plus, il faut prévoir une signalisation lumineuse :

- Lorsque l'on est en mode automatique, le voyant du bouton Start s'allume.
- Lorsque la pièce est présente à l'entrée du convoyeur, le voyant Q1 est allumé. Lorsque le convoyeur est en marche, il clignote à la fréquence de 2 Hz.
- Lorsque le magasin est plein, Q1 et Q2 clignotent à 1 Hz.

# Atelier C - TIA Portal



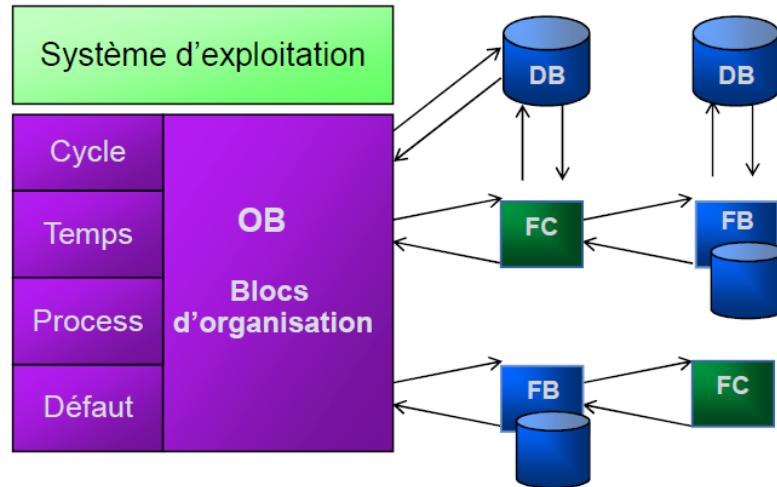
programmation des automates S7-300  
Blocs d'organisation

The screenshot shows the TIA Portal interface for programming S7-300 automation. On the left, there are two icons: a purple one labeled "OB Bloc d'organisation" and a blue one labeled "FB Bloc fonctionnel". The main workspace displays a ladder logic diagram for an organization block (OB). The diagram includes a coil for "lancement OB alarme" (OB call), a normally open contact for "%M15.0 \*Front\*", and a normally closed contact for "T#5S". The output coil is labeled "SRT\_DINT". The contacts are connected to the inputs of the SRT\_DINT block. The block's parameters are set to "OB\_NR = 20" and "DTIME = #000:00:00". The output RET\_VAL is assigned to the value "20". The right side of the screen shows the hardware rack configuration for a Siemens CPU 314C-2 DP, detailing its digital input (DI) and digital output (DO) assignments.

API\_1 [CPU 314C-2 PN/DP]

- Configuration des appareils
- En ligne & Diagnostic
- Blocs de programme
  - Ajouter nouveau bloc
  - Main [OB1]
  - COMPLETE RESTART [OB100]
  - TOD\_INTO [OB10]
  - DEL\_INTO [OB20]
  - HW\_INTO [OB40]
  - CYC\_INT5 [OB35]

## 1. Types de blocs de programme



### Légende :

OB = Bloc d'organisation  
FB = Bloc fonctionnel  
FC = Fonction  
DB = Bloc de données



Le degré d'imbrication maximum des blocs dépend de la CPU.

L'automate met à disposition différents types de blocs qui contiennent le programme et les données correspondantes. Selon les exigences et la complexité du processus, il est possible de structurer le programme en différents blocs : OB, FB et FC.

### 1.1 Les blocs d'organisation - OB



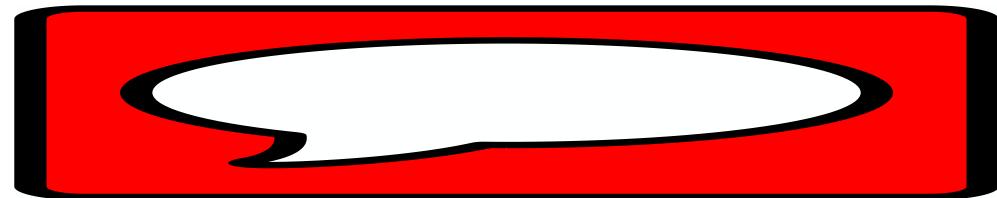
Ils constituent l'interface entre le système d'exploitation et le programme utilisateur.

Les OB sont programmables par l'utilisateur, ce qui permet de déterminer le comportement de la CPU.

Les OB sont appelés par le système d'exploitation en liaison avec les événements suivants :

- Comportement au démarrage
- Exécution cyclique du programme
- Exécution du programme déclenchée par des alarmes (cyclique, processus, diagnostic,...)
- Traitement des erreurs

Pour que le traitement du programme démarre, le projet doit posséder au moins un OB cyclique (par exemple l'OB 1)



## 1.2 Les fonctions – FC



Ce sont des blocs de code sans mémoire.

Les données des variables temporaires sont perdues après l'exécution de la fonction. Si on veut mémoriser ces données, il faut utiliser des opérandes globaux.

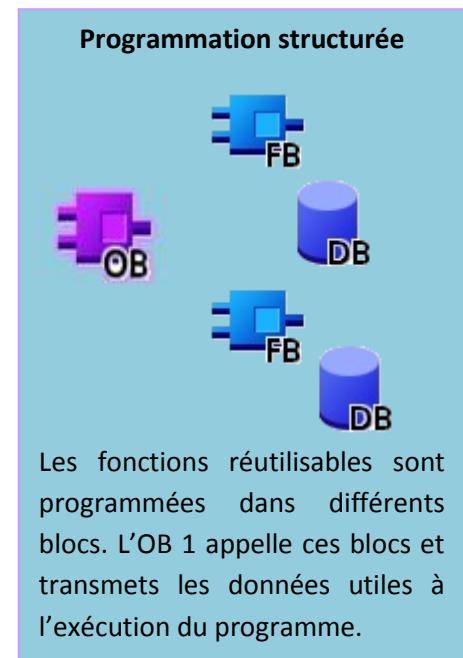
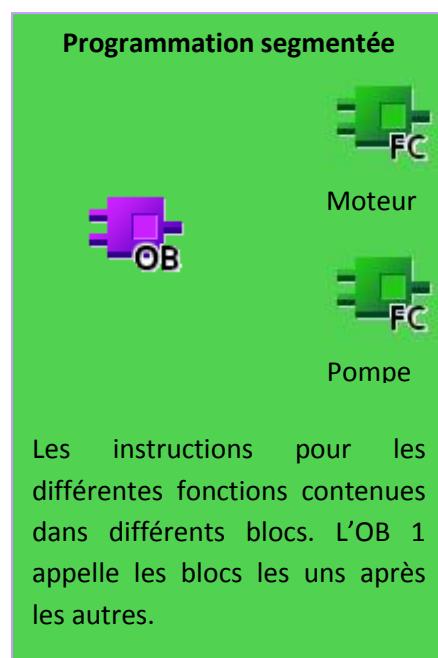
Elles sont utilisées pour la programmation de fonctions utilisées plusieurs fois. On simplifie de ce fait la programmation.

## 1.3 Les blocs fonctionnels – FB

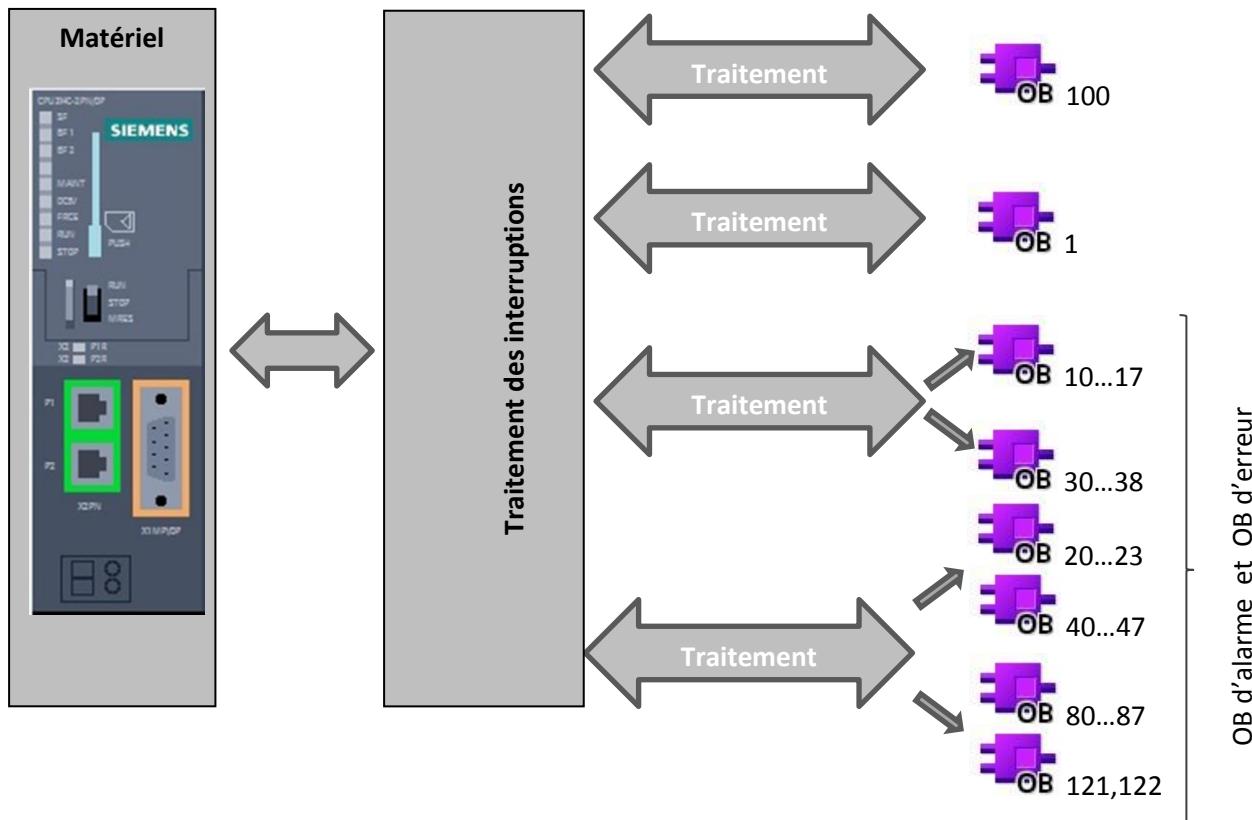


Ce sont des blocs de code qui mémorisent durablement leurs paramètres d'entrée, de sortie et d'entrée/sortie dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement de blocs.

L'utilisation de ces différents blocs permet plusieurs structures du programme.



## 2. Les Blocs d'Organisations OB



### 2.1 Les OB de démarrage (ex : OB 100)

Ils sont traités une seule fois, lorsque le mode de fonctionnement passe de STOP à RUN. Après le traitement de l'OB de démarrage, c'est le traitement de l'OB cyclique qui démarre (voir « type de démarrage dans S7 »).

### 2.2 Les OB cycliques (ex : OB 1)

Ils sont traités cycliquement. Les OB cycliques sont des blocs de code de niveau supérieur dans le programme, dans lesquels on peut appeler d'autres blocs.

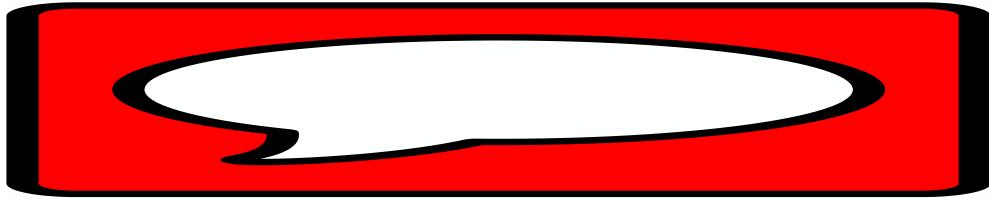
### 2.3 Les OB de traitement périodique

Le traitement cyclique du programme peut être interrompu par des OB de priorités supérieure.

#### Les OB d'alarmes horaires (OB 10)

Les alarmes horaires sont utilisées pour exécuter un programme donné, appelé dans l'OB 10, une seule fois à un moment précis ou périodiquement à partir de ce moment (toutes les minutes, toutes les heures, tous les jours, toutes les semaines, tous les mois, déclenchement annuel).

En fonction de la CPU, l'utilisateur dispose au maximum de huit alarmes horaires différentes.



## Les OB d'alarmes temporisées (OB 20, 21)

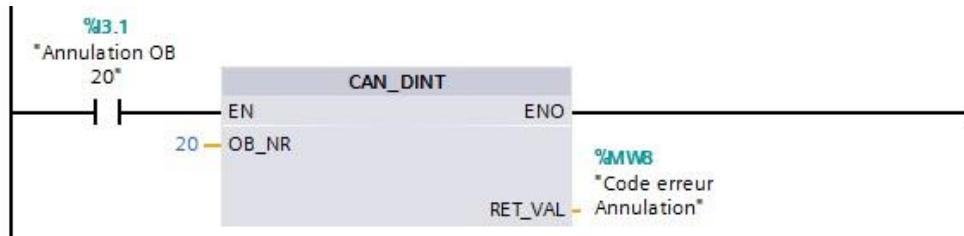
S7 met à disposition jusqu'à quatre OB (en fonction de la CPU) qui sont traités à chaque fois à la suite d'une temporisation paramétrable.

Le traitement du programme d'un OB d'alarme temporisée (OB20) est lancé avec retard après l'apparition d'un événement déterminé.

L'OB20 peut être activé uniquement par un appel de l'instruction « **SRT\_DINT** ». Cette instruction est également utilisée pour préciser la durée de la temporisation.



L'action peut être annulée par l'instruction « **CAN\_DINT** ».

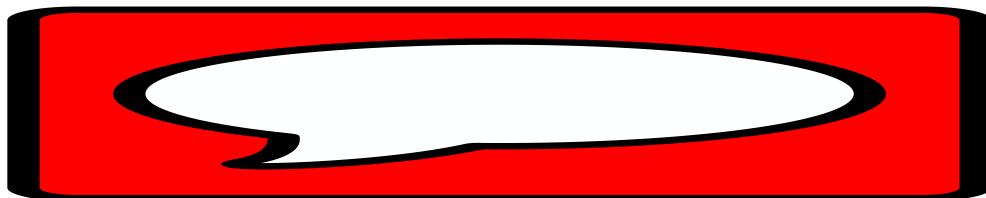


## Les OB d'alarmes cycliques (OB 30...37)

Une alarme cyclique permet de lancer le traitement d'un bloc à intervalles réguliers. Le S7-300 dispose de l'OB d'alarme cyclique OB35. Par défaut, l'intervalle de temps pour l'appel de l'OB est de 100 ms, la plage de réglage allant de 1 ms à 1 min.

En fonction de la CPU, l'utilisateur dispose au maximum de huit alarmes cycliques différentes.

Il faut veiller à ce que l'intervalle défini soit supérieur au temps nécessaire à l'exécution du contenu de l'OB 35. Si l'OB 35 est encore actif au moment où il est appelé, le système appelle l'OB 80 (erreur d'alarme cyclique).



## 2.4 Les OB d'alarmes processus

Le traitement du programme d'un OB d'alarme de processus (OB40) est lancé dès qu'un événement déterminé survient dans le processus.

Les alarmes de processus peuvent être déclenchées par différents signaux provenant des modules :

- **Sur les modules de signaux paramétrables** (DI, DO, AI, AO), le signal qui doit déclencher l'alarme de processus est défini avec l'outil de configuration matérielle.
- **Sur les modules CP** (processeur de communication) et FM (module de fonction), le comportement de l'alarme est défini à l'aide du logiciel de configuration du module correspondant.

En fonction de la CPU, l'utilisateur dispose au maximum de huit alarmes de processus différentes.

Il n'y a pas d'alarme processus disponible sur les automates du CTA.

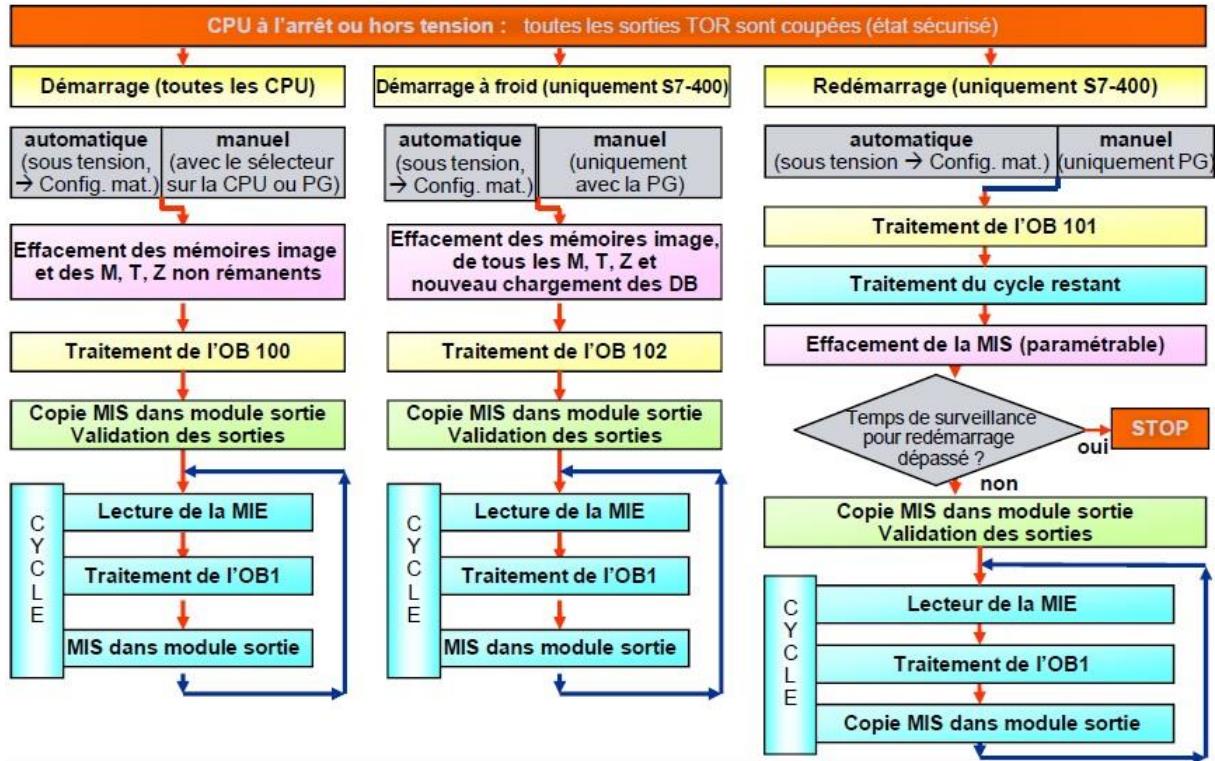
## 2.5 Les OB de traitement des erreurs asynchrones

Par définition, les erreurs asynchrones surviennent de manière asynchrone par rapport au traitement du programme et ne peuvent donc pas être imputées à un endroit précis du programme.

Le tableau ci-dessous nous montre les différents types d'erreurs asynchrones.

Type d'erreur	Exemple	OB d'erreur	Priorité
Alarme esclave DPV1	Alarme spéciale des esclave Profibus DPV1	OB 55 / 56 / 57	26 / 28 (également paramétrable selon la CPU)
Perte de redondance	Coupure de la CPU (uniquement S7-400 à disponibilité élevée)	OB 70 / 72	
Erreur de temps	Dépassement du temps de cycle maxi.	OB 80	
Défaut d'alimentation	Défaillance de la pile de sauvegarde	OB 81*	
Alarme de diagnostic	Rupture de fil sur module avec fonction de diagnostic	OB 82	
Alarme débrochage/enfichage	Retrait / Montage d'un module	OB 83	
Défaut matériel CPU	Niveau de signal erroné sur l'interface MPI	OB 84*	
Erreur d'exécution du programme	Erreur d'actualisation de la mémoire image	OB 85	
Défaut du châssis	Défaillance d'un esclave DP	OB 86	
Erreur de communication	Détection erronée d'un télégramme	OB 87	

### 3. Type de démarrage dans S7

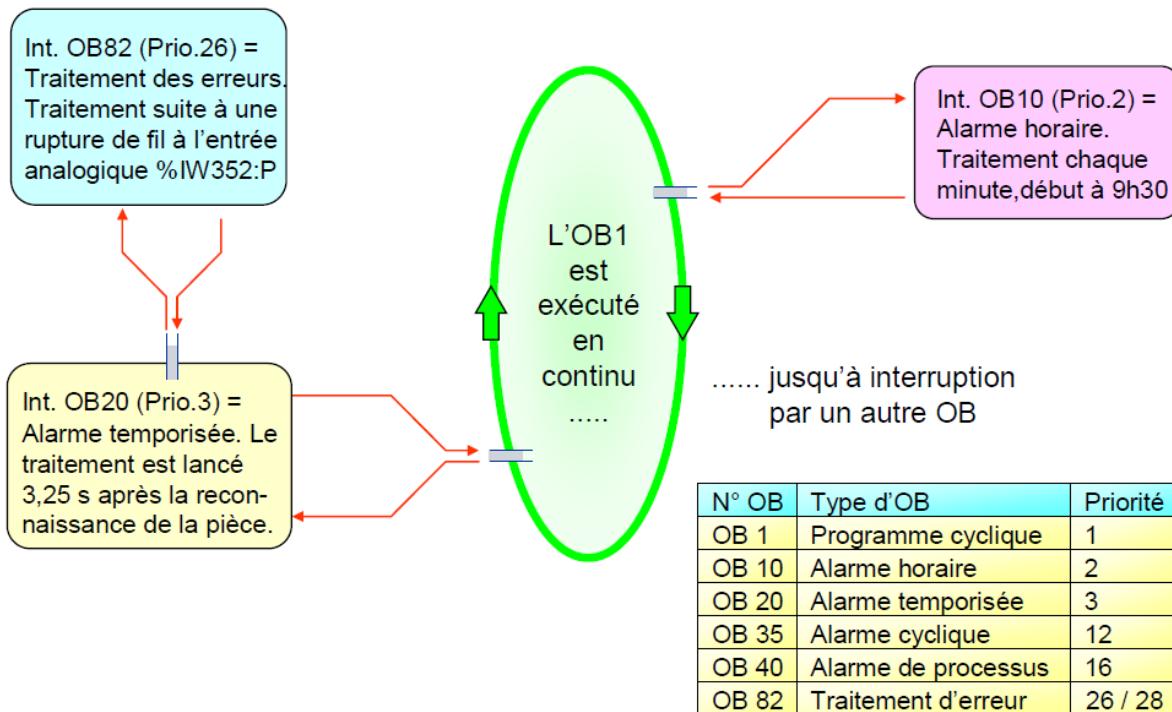


Il existe 3 modes de démarrage de la CPU, avec les trois OB de mise en route correspondants. Cela permet à l'utilisateur de programmer le comportement de l'automate au démarrage en fonction de la situation.

**En cas de démarrage**, les mémoires images du processus (Mémoire Internes des Entrées et Mémoires Interne des Sorties) ainsi que tous les mémentos, les temporisations et les compteurs non rémanents sont effacés. Les mémentos, les temporisations et les compteurs rémanents, ainsi que les valeurs effectives des blocs de données sont en revanche conservé si l'automate est doté d'une pile de sauvegarde (S7 – 300 également sans pile de sauvegarde, si l'automate est doté d'une EPROM et que la rémanence de la CPU a été paramétrée). **Le programme stocké dans l'OB 100 est exécuté une seule fois, puis le traitement cyclique du programme démarre.**

Les CPU des automates S7 – 400 et les modèles haut de gamme du S7 – 300 peuvent faire l'objet d'un redémarrage ou d'un démarrage à froid (voir tableau ci-dessus).

## 4. Diagramme de fonctionnement des OB



Les OB constituent l'interface entre le système d'exploitation de la CPU et le programme utilisateur. Ils sont appelés exclusivement par le système d'exploitation. Certains événements (alarmes horaire et de processus,...) entraînent le lancement d'un OB approprié.

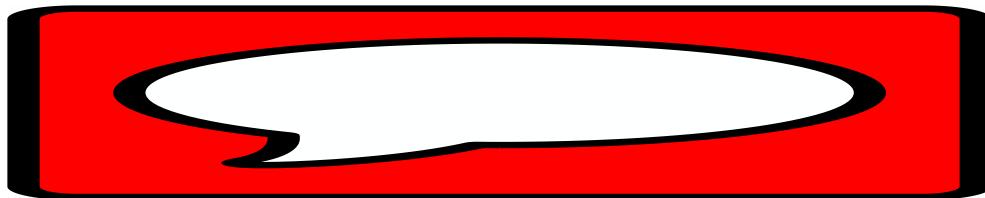
### 4.1 Interruption des OB

Lorsque le système d'exploitation appelle un OB autre que l'OB1, il interrompt le traitement cyclique du programme car l'OB1 est **celui qui a la priorité la plus faible**. Tout OB peut donc interrompre le programme principal et lancer l'exécution de son propre programme, le traitement de l'OB1 reprenant ensuite au point d'interruption.

Lorsque le système appelle un OB de priorité supérieure à celui déjà en cours d'exécution, l'interruption intervient après l'opération en cours de traitement. Le système d'exploitation sauvegarde alors la pile complète des registres du bloc interrompu. Les informations contenues dans ces registres sont restaurées lorsque le système d'exploitation reprend le traitement du bloc interrompu.

### 4.2 Priorité des OB

Le traitement d'un OB peut être interrompu aux limites d'une instruction par un événement (OB) de priorité supérieure. Les priorités s'échelonnent de 0 à 28, 0 étant la priorité la plus faible et 28 la priorité la plus forte.



L'OB 82 possède soit la priorité 26 lorsqu'il survient au cours du traitement de l'OB 1, soit la priorité 28 lorsqu'il survient au cours d'un démarrage.

Les OB de même priorité ne s'interrompent pas mutuellement, mais sont traités selon leur ordre d'occurrence.

## 5. Exercices

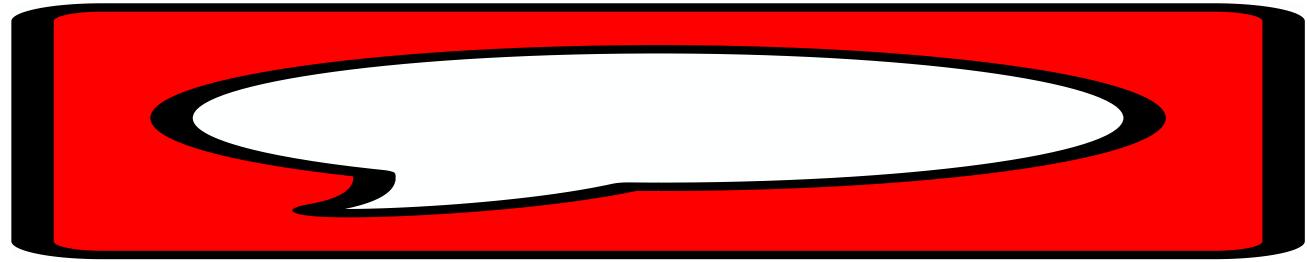
1. A l'aide de l'OB 100, réaliser un programme qui vous permet d'avoir en permanence un bit à 1 (M0.1) et un bit à 0 (%M0.0).
2. Vous souhaitez utiliser une fréquence de clignotement de 4 Hz. Cette fréquence n'est malheureusement pas disponible via le mémento de cadence de clignotement. Etablir une fréquence de clignotement dans le mémento M35.0 à l'aide de l'alarme cyclique.
3. Réaliser un programme dans lequel toutes les minutes, un voyant s'allume pendant 5 secondes.
4. Réaliser un programme dans lequel tous les jours à 16h, une sonnerie (%Q1.0) retenti jusqu'à acquittement via une impulsion sur les BP\_ACQ (%I0.1)

# Atelier C : TIA Portal



## Introduction à la programmation Grafcet sous Siemens TIA Portal





## **1 BUTS DE L'EXERCICE :**

Le but de ce module est de réaliser un programme en langage Grafcet pour piloter de façon basique la station Festo MPS Sorting et de le tester dans un Automate Siemens S7-300.

## **2 MISE EN SITUATION**

La programmation Grafcet est utilisée lors de la programmation de processus industriels séquentiels. Celle-ci est particulièrement bien adaptée à la programmation des stations MPS FESTO présente dans le laboratoire d'automation.

La plate-forme Siemens TIA (Totally Integrated Automation) Portal est la dernière évolution des logiciels de programmation Siemens. Cette plate-forme regroupe dans un seul logiciel la programmation des différents dispositifs d'une installation. On peut donc avec ce logiciel, programmer et configurer, en plus de l'automate, les dispositifs HMI, les variateurs, etc.

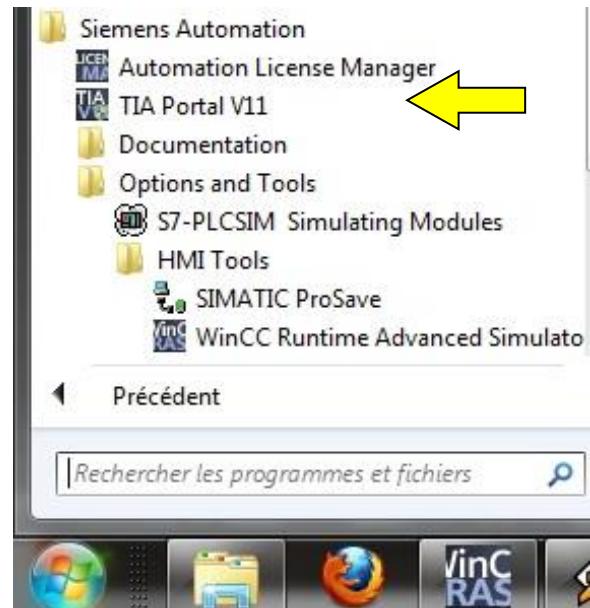
## **3 TRAVAIL A REALISER**

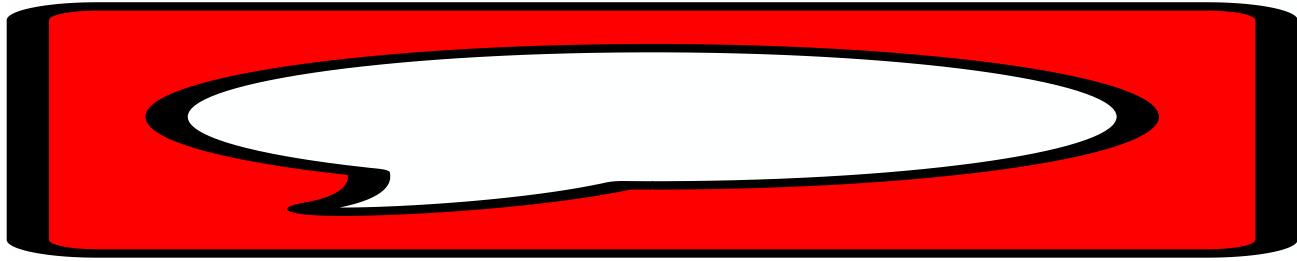
### **3.1 Création d'un projet et paramétrage de l'automate**

#### **3.1.1 Démarrage du logiciel**

Dans l'arborescence du Menu Démarrer, cliquez sur TIA PORTAL V11 qui se trouve dans le dossier Siemens Automation.

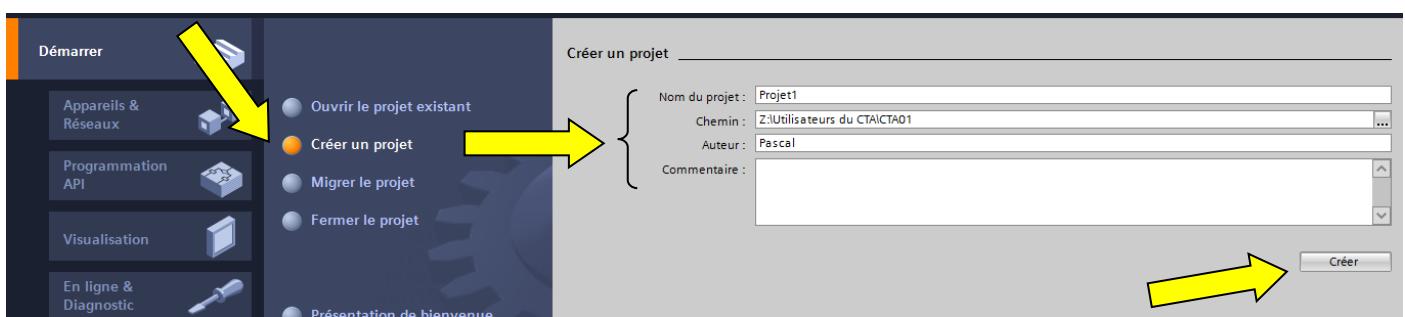
Vous pouvez également utiliser le raccourci présent sur le bureau.





### 3.1.2 Crédation d'un nouveau projet

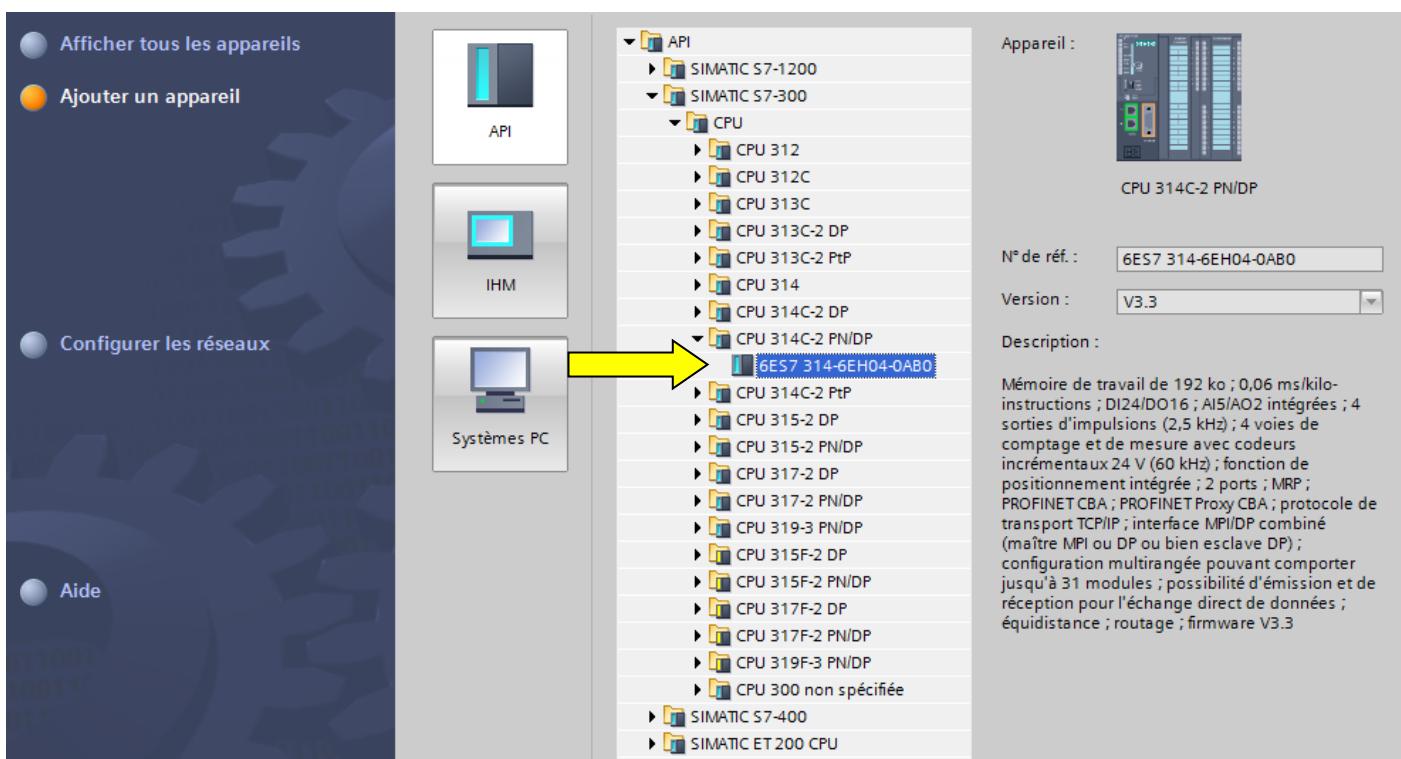
- a) Cliquez sur nouveau projet
- b) Complétez les données nécessaires
  - Nom du projet : Grafset 01
  - Chemin : Z:\Utilisateurs du CTA\CTA\$\$\Grafcet01  
\$\$ = numéro de votre PC
- c) Cliquez sur le bouton Créer

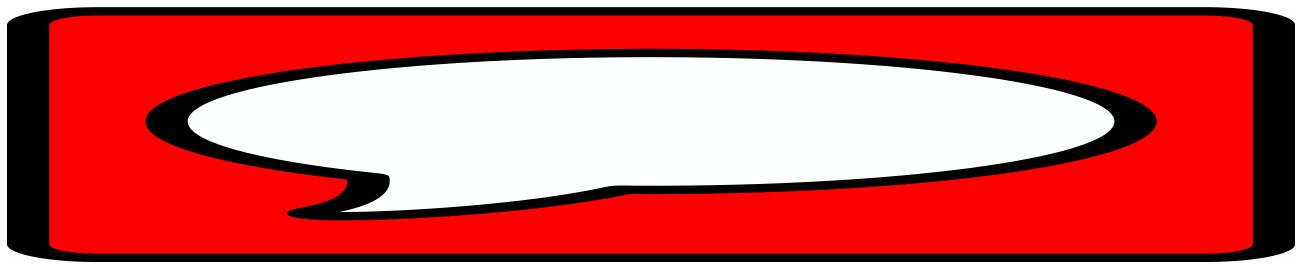


**ATTENTION À L'ENDROIT OÙ VOUS ENREGISTREZ VOTRE PROJET!!!!**

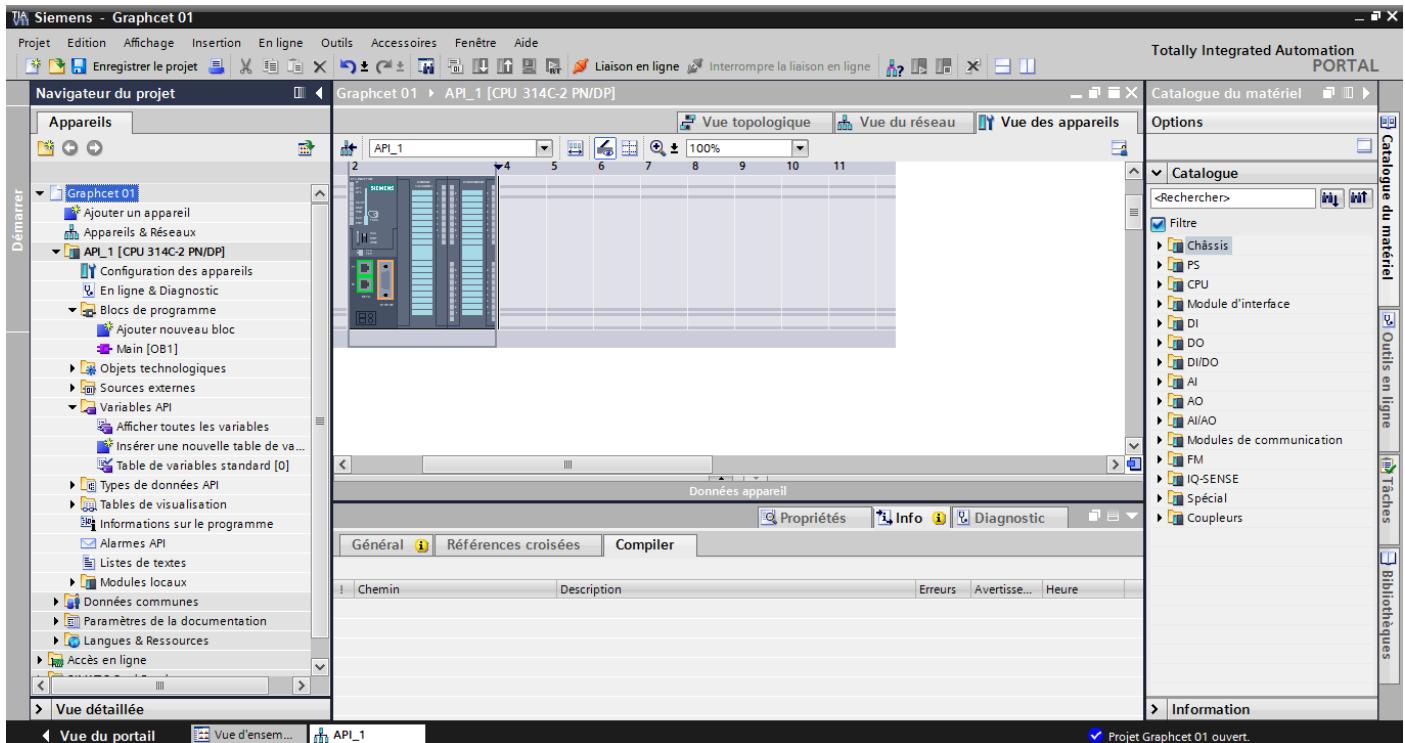
### 3.1.3 Ajout d'un automate

Dans la liste, sélectionnez l'automate S7 314C-2 PN/DP (6ES7 314-6EH04-0AB0) puis cliquez sur Ajouter.





### 3.1.4 Présentation de l'interface



- Navigateur de projet

Dans cette partie, on retrouve la liste de tous les composants du projet. Plusieurs automates peuvent s'y retrouver lorsque le projet contient plusieurs automates en réseau. On peut également y retrouver les appareils de supervision ou les régulateurs présents dans l'installation. Pour chaque appareil du projet, on retrouvera tous les éléments nécessaires à la configuration, programmation ou mise en service.

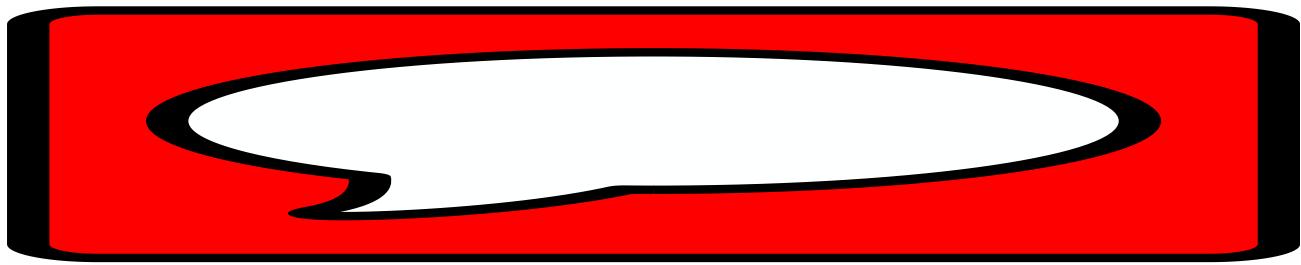
- Partie centrale

La partie centrale de la fenêtre permet d'afficher le contenu du menu sélectionné dans le navigateur de projet. On y retrouvera donc aussi bien les outils de programmation de l'automate que les menus de configuration des différents éléments du projet.

- Partie de droite

Le menu à droite de l'écran varie également en fonction de l'élément sélectionné dans le navigateur de projet. On y retrouvera, par exemple, les blocs à intégrer lorsqu'un bloc de programme est sélectionné.



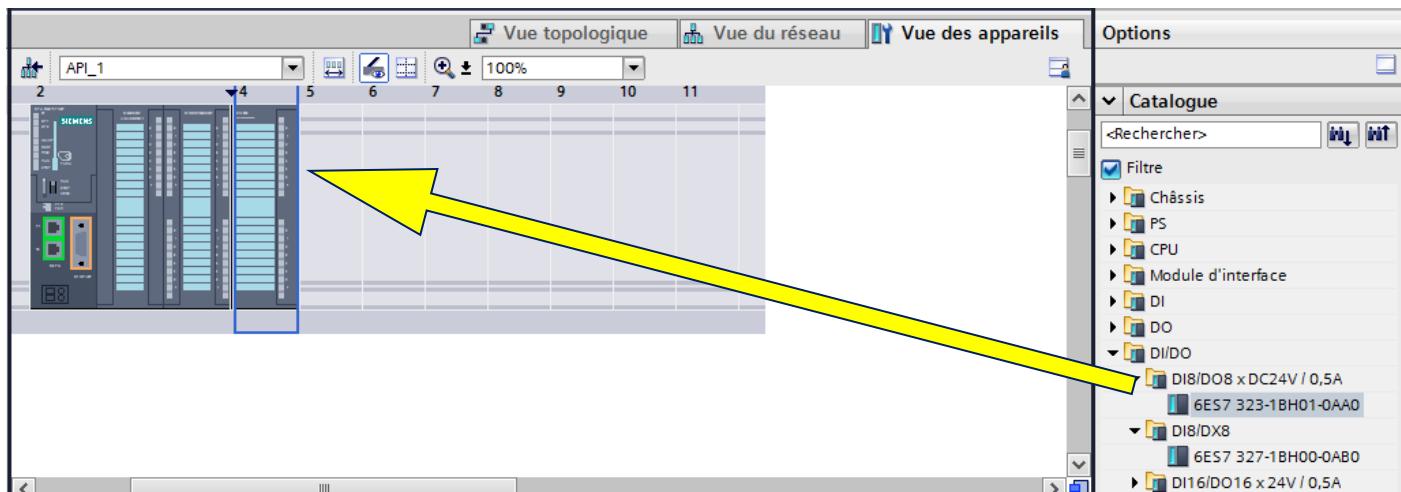


- Barre d'outils principale

Cette barre d'outils contient, en plus des traditionnelles icônes présentes dans la plupart des logiciels, des icônes permettant de compiler et de charger les programmes dans l'automate. On retrouve également des icônes permettant de contrôler l'automate connecté au PC.

### 3.1.5 Ajout du module de simulation

Un module de simulation est accouplé à l'automate. Ce module permet de simuler aux choix, 16 entrées, 16 sorties ou 8 entrées et 8 sorties. Dans cet exercice, nous utiliserons le module dans sa configuration 8 entrées et 8 sorties.

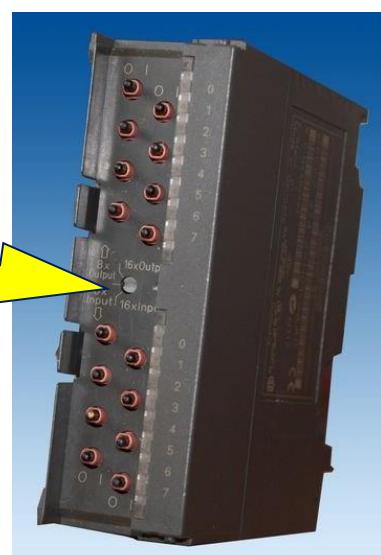


Après avoir sélectionné le menu « Vue des appareils » dans le navigateur de projet, ajoutez le module 6ES7 323-1BH01-0AA0 que l'on peut trouver dans l'arborescence en suivant ces différentes entrées :

**DI/DO → DI8/DO8 x DC24V/0,5A.**

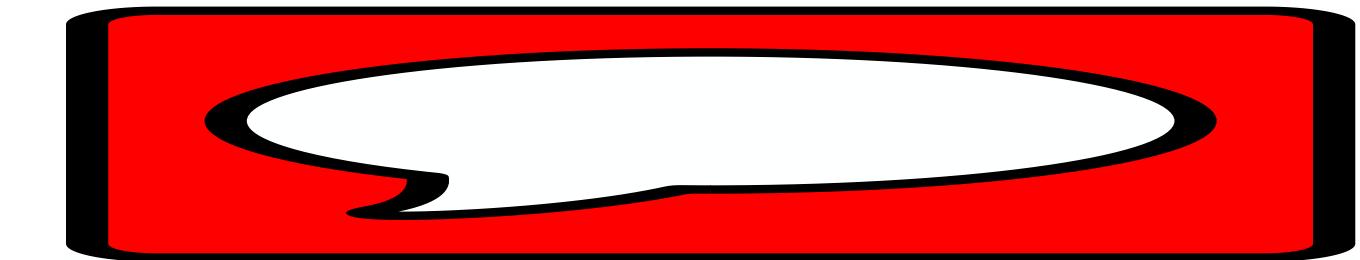
Pour ajouter le module, faites un glisser déposer du numéro du module vers l'emplacement 4 de la platine automate.

Vérifiez également que le commutateur sur le module est bien positionné sur position **8 x output – 8 x input**.



la

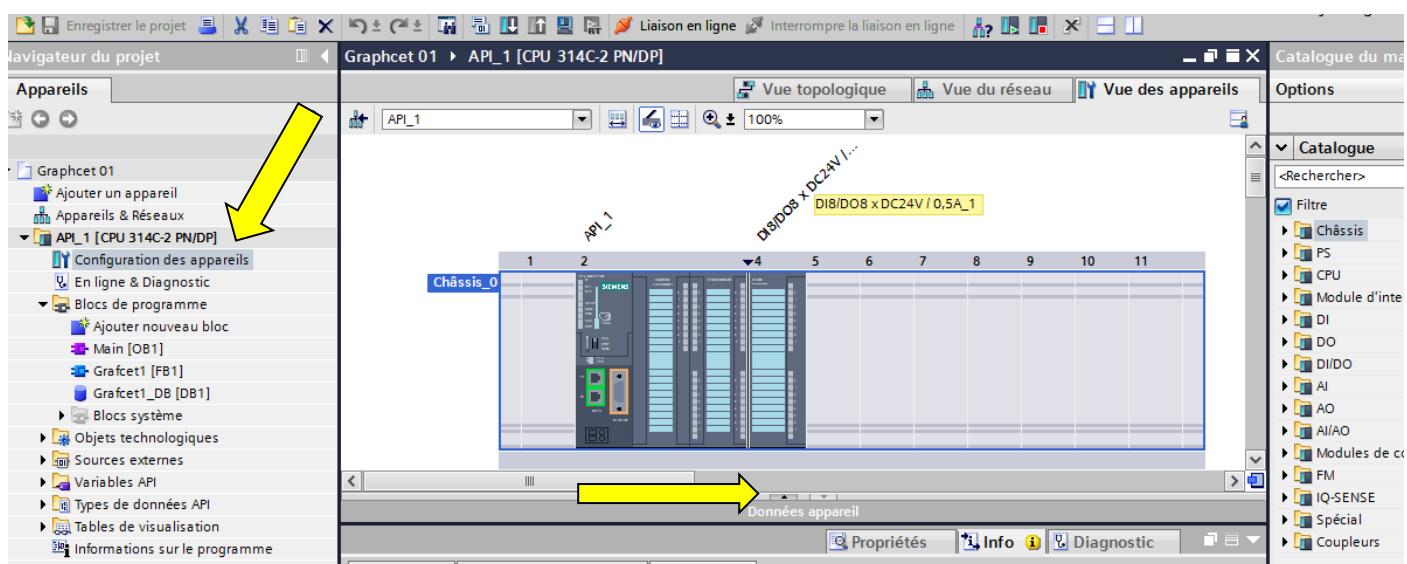




### 3.1.6 Relevé des adresses utilisées pour les entrées/sorties

Avant de pouvoir travailler, il nous reste à relever les adresses des différentes entrées/sorties de l'automate.

Après avoir sélectionné la vue « Configuration des appareils » dans le navigateur de projet, cliquez sur la flèche permettant d'afficher les données de l'appareil.



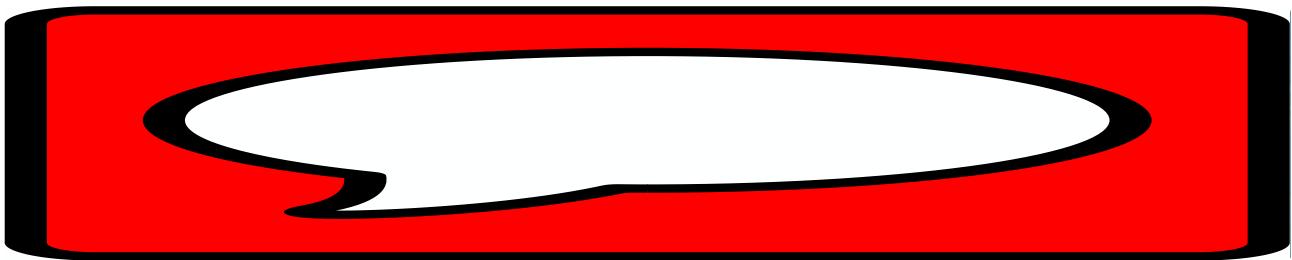
Le logiciel affiche alors les adresses utilisées par les différents modules de l'automate.

Vue d'ensemble des appareils							
Module	Châssis	Empla..	Adresse I	Adresse	Type	N° de réf. :	Firmw.
0	1						
▼ API_1	0	2			CPU 314C-2 PN/DP	6ES7 314-6EH04-0AB0	V3.3
Interface MPI/DP_1	0	2 X1	2047*		Interface MPI/DP		
► Interface PROFINET_1	0	2 X2	2046*		Interface PROFINET		
DI24/DO16_1	0	2 5	136...138	136...137	DI24/DO16		
AI5/AO2_1	0	2 6	800...809	800...803	AI5/AO2		
Comptage_1	0	2 7	816...831	816...831	Comptage		
Positionnement_1	0	2 8	832...847	832...847	Positionnement		
	0	3					
DI8/DO8 x DC24V / 0,5A_1	0	4	0	0	DI8/DO8 x DC24V / ...	6ES7 323-1BH01-0AA0	

Les adresses utilisées sont donc:

- 0.0 à 0.7 pour les entrées et les sorties du module de simulation
- 136.0 à 137.7 pour les entrées et les sorties intégrées à l'automate





### 3.1.7 Liaison avec l'automate

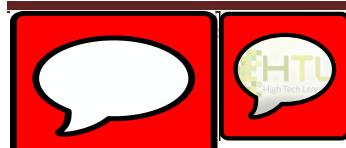
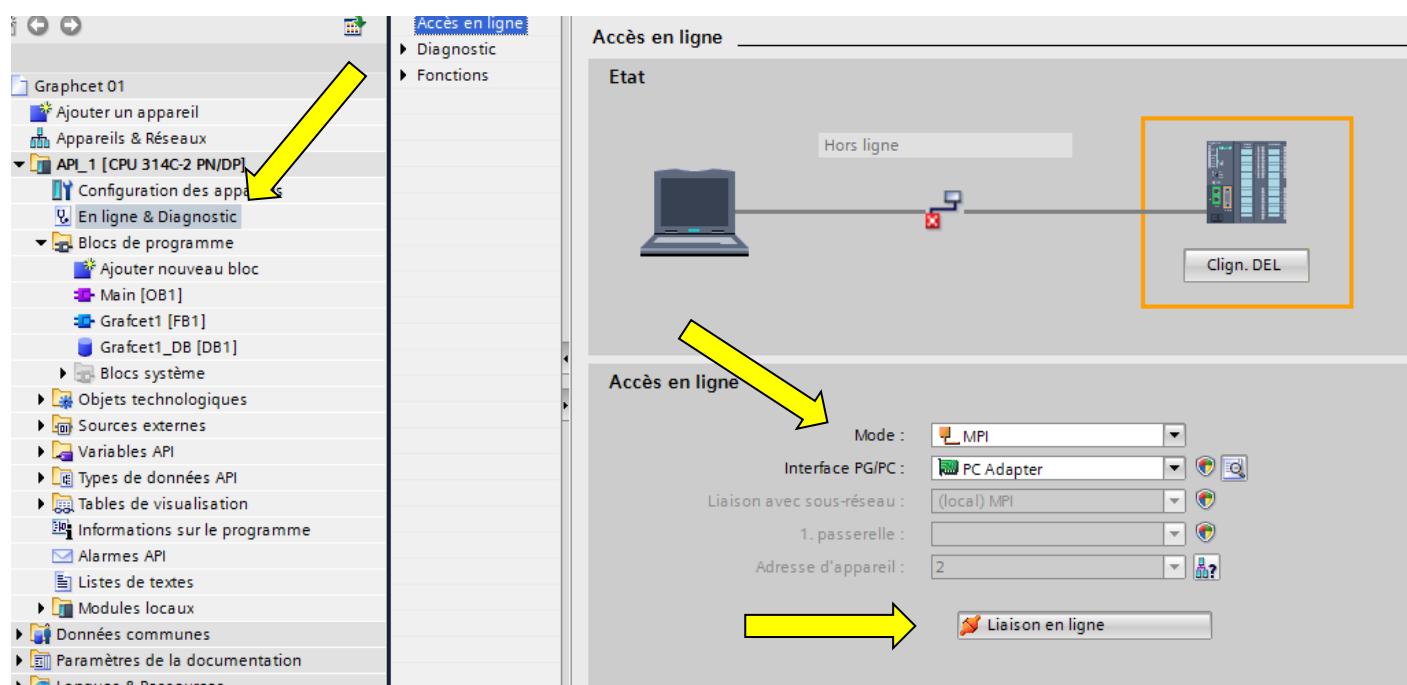
Il faut maintenant charger la configuration de l'automate dans celui-ci.

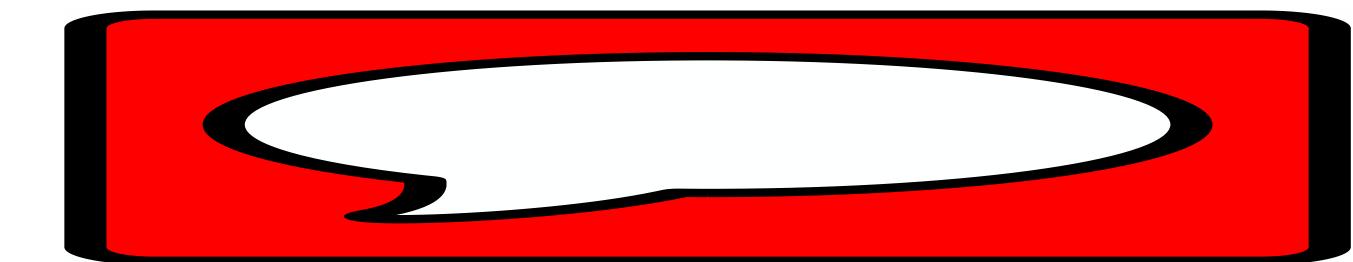
Pour cela, il faut tout d'abord connecter l'automate au PC en utilisant l'interface Simatic S7 PC USB adapter.

Ensuite, après avoir sélectionné la vue « En ligne et diagnostique », sélectionnez les options suivantes :

- Mode : MPI
- Interface PG/PC : Pc Adapter

Il ne reste plus qu'à cliquer sur le bouton « Liaison en ligne » pour vous connecter à l'automate.

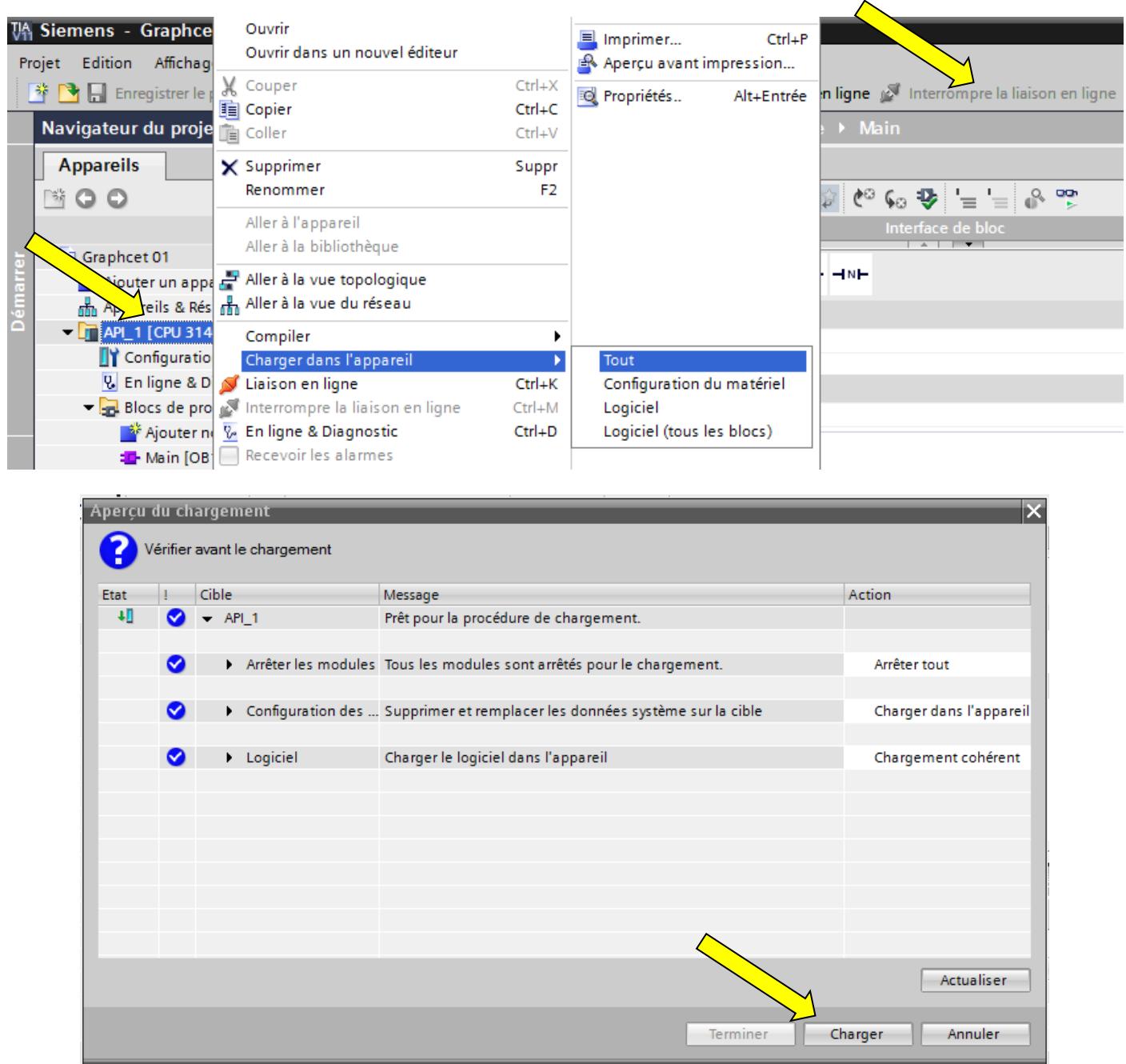


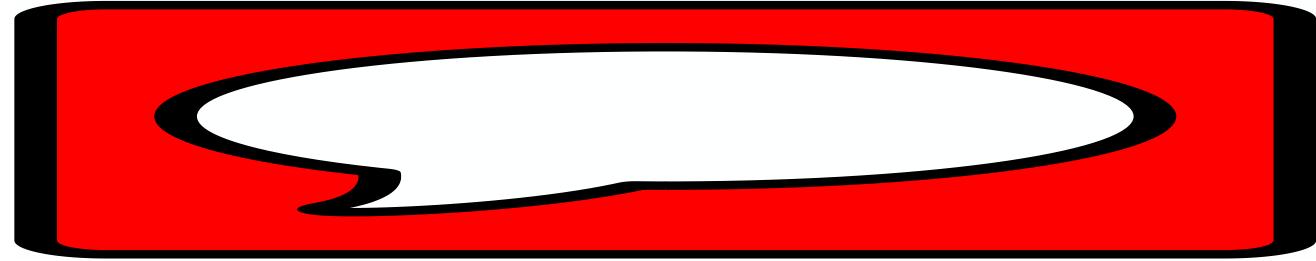


### 3.1.8 Chargement de la configuration dans l'automate

Maintenant que notre automate est correctement configuré dans le projet, nous allons charger cette configuration dans l'automate.

Après avoir interrompu la liaison en ligne, il suffit de faire un clic de droite sur l'automate dans le navigateur de projet puis de cliquer sur Charger dans l'appareil → Tout.





## 3.2 Programmation de l'automate

### 3.2.1 Crédation d'une table de variables

Afin de faciliter la programmation, il est intéressant de créer une table de variables.

	Nom	Table des variables	Type de données	Adresse	Réma...	Visible...	Acces...	Commentai...
1	IN0	Table de variables s..	Bool	%IO.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	IN1	Table de variables s..	Bool	%IO.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	IN2	Table de variables s..	Bool	%IO.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	IN3	Table de variables s..	Bool	%IO.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	IN4	Table de variables s..	Bool	%IO.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	IN5	Table de variables s..	Bool	%IO.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	IN6	Table de variables s..	Bool	%IO.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	IN7	Table de variables s..	Bool	%IO.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	OUT0	Table de variabl...	Bool	%I1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	<Ajouter>							

Pour créer une variable, il suffit d'indiquer le nom de la variable et son adresse.

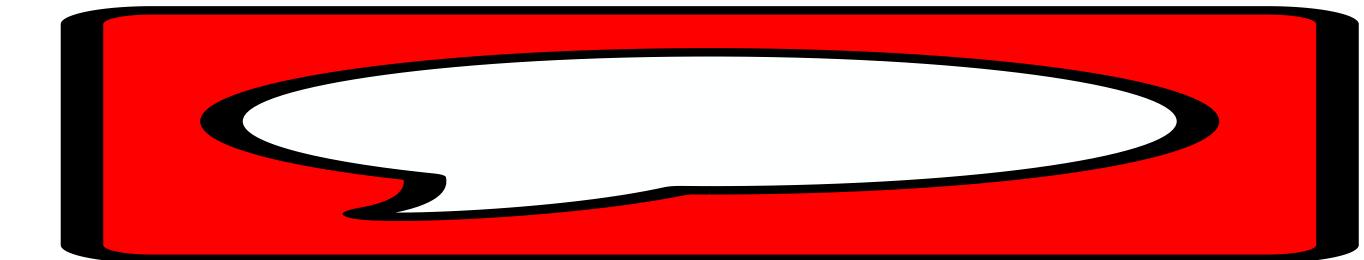
Pour gagner du temps, il est possible de copier les variables, comme on peut le faire dans un tableur, en tirant sur le coin inférieur droit d'une cellule.

26	1M2	Table de variables s.. Bool
27	1MB	Table de variables s.. Bool
28	LStart	Table de variables standard
29	<Ajouter>	

### 3.2.2 Types de variables

- Identificateurs d'opérande
  - Entrée (I)
  - Sortie (Q)
  - Memento (M)
- Types de Variables
  - Bool (1 bit) → 1 ou 0
  - Byte (8 bits)
  - Word (16 bits)
  - DWord (32bits)
  - ...





### 3.2.3 Tables de visualisation

Afin de visualiser les entrées/sorties de l'automate depuis TIA portal, il est nécessaire de construire une table de visualisation.

Pour créer une table de visualisation, cliquez sur ajouter une table de visualisation dans le navigateur de projet.

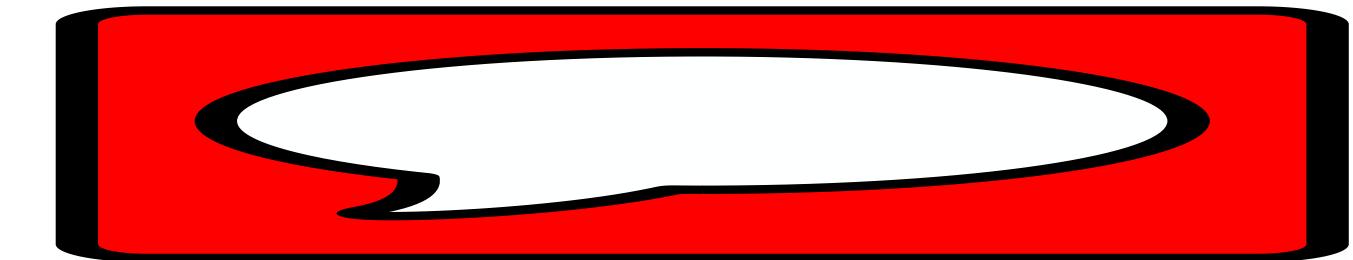
i	Nom	Adresse	Format d'affichage	Valeur visualisatio...	Valeur de forçage	...
1	"IN1"	%IO.3	Bool	<input checked="" type="checkbox"/> TRUE	<input type="checkbox"/>	
2	"IN2"	%IO.4	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
3	"IN3"	%IO.5	Bool	<input checked="" type="checkbox"/> TRUE	<input type="checkbox"/>	
4	"IN4"	%IO.6	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
5	"IN5"	%IO.7	Bool	<input checked="" type="checkbox"/> TRUE	<input type="checkbox"/>	
6	"IN6"	%Q0.0	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
7	"INT"	%Q0.1	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
8	"OUT0"	%Q0.2	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
9	"OUT1"	%Q0.3	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
10	"OUT2"	%Q0.4	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
11	"OUT3"	%Q0.5	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
12	"OUT4"	%Q0.6	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
13	"OUT5"	%Q0.7	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>	
14	"OUT6"					
15	"OUT7"					
16						

Tapez ensuite le nom des variables que vous voulez visualiser.

Pour voir les valeurs des variables, il est nécessaire de cliquer sur le bouton « visualiser tout ».

Il est également possible de forcer une variable. Faites un clic de droite sur une des variables, puis choisissez le type de forçage que vous voulez réaliser.

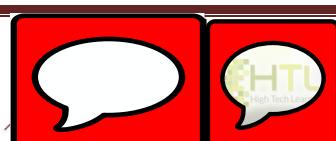
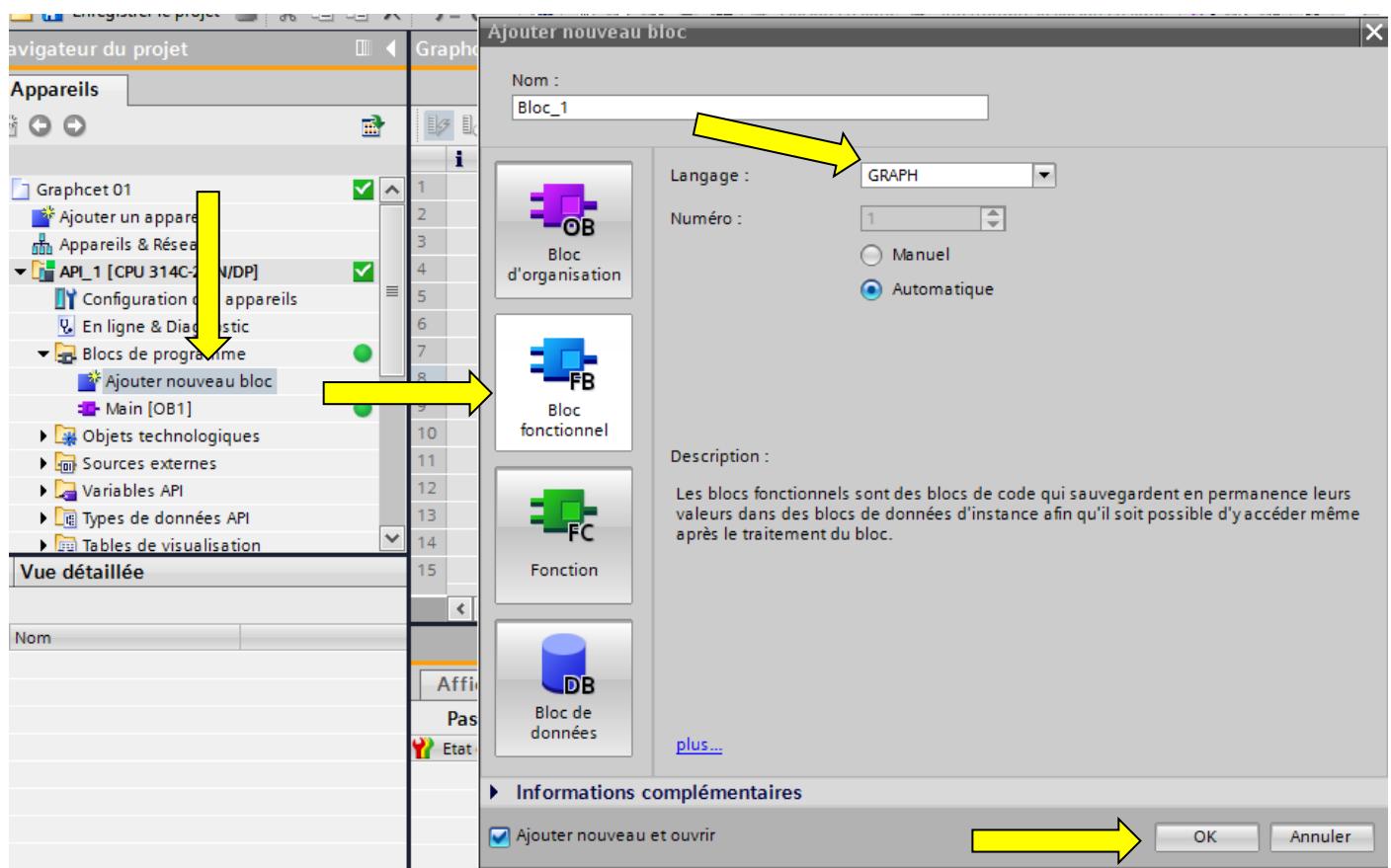




### 3.2.4 Ajout d'un Grafcet

Nous arrivons enfin à la programmation du Grafcet.

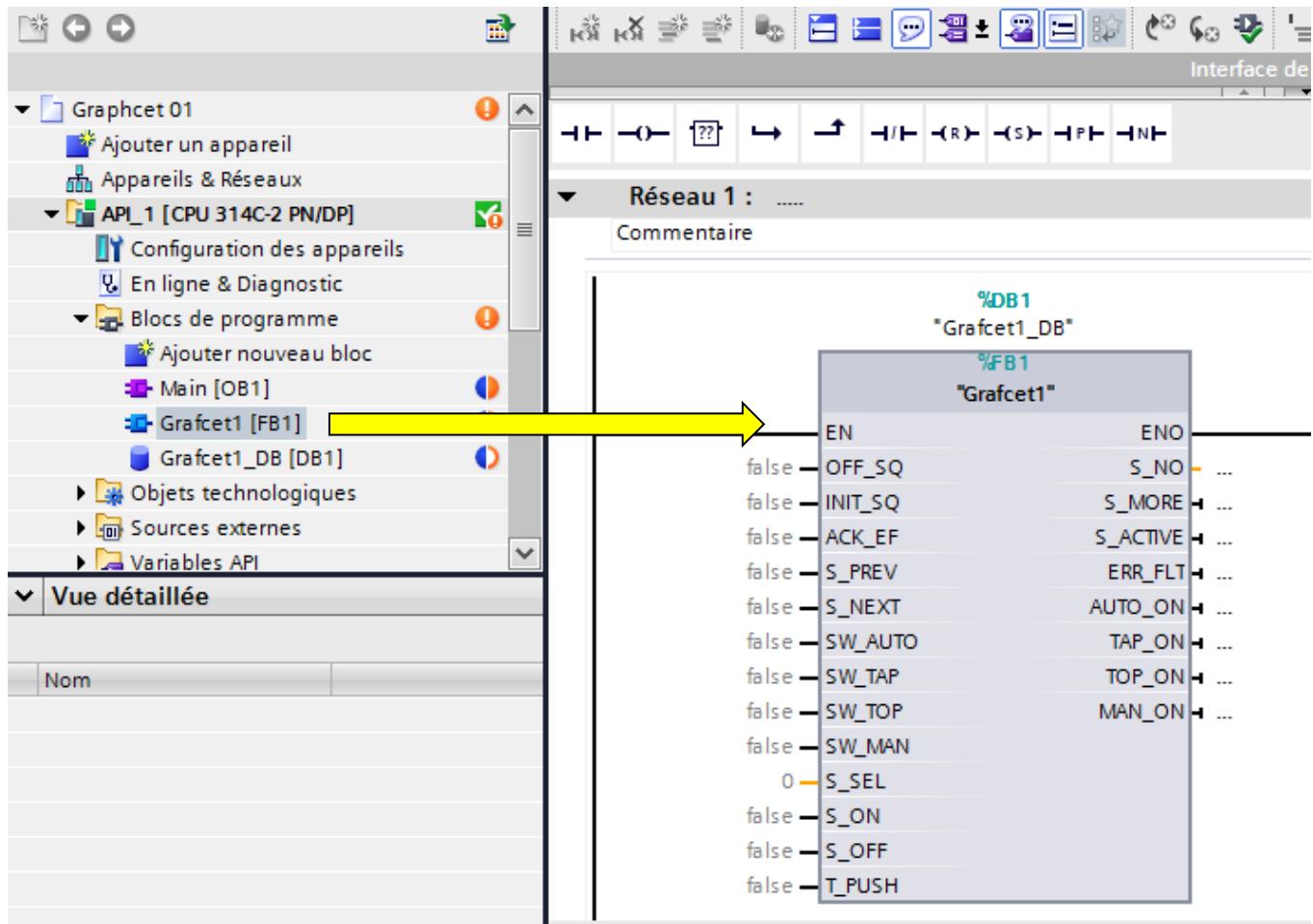
Pour ajouter un Grafcet dans le projet, il faut ajouter un bloc fonctionnel au programme de l'automate. N'oubliez pas de choisir le langage GRAPH avant d'ajouter le bloc.



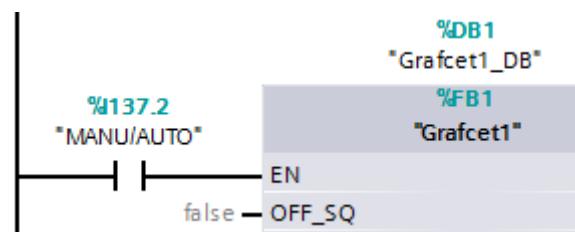
### 3.2.5 Appel du Grafset depuis le bloc de programme principal

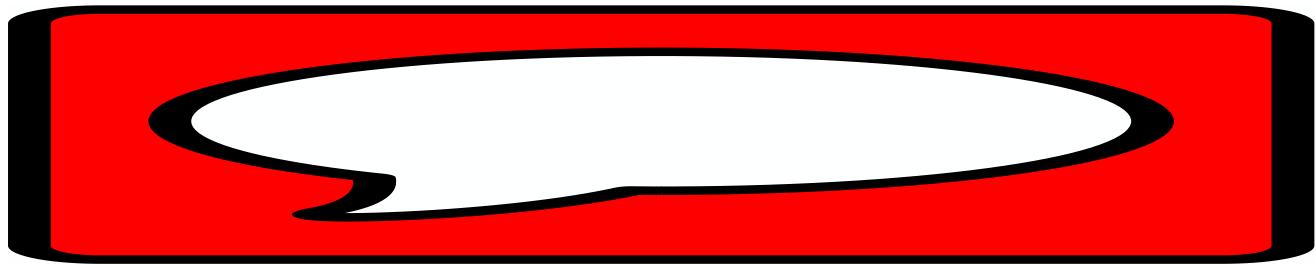
Pour que l'automate exécute le Grafset, il faut que le bloc fonctionnel contenant le Grafset soit appelé depuis le bloc de programme principal.

L'ajout du bloc de programme se fait simplement en glissant le bloc de programme fonctionnel dans le bloc de programme principal.



Il est possible de rendre conditionnelle l'exécution de ce Grafset en plaçant un contact sur l'entrée de celui-ci.



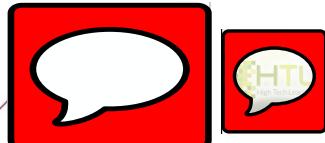
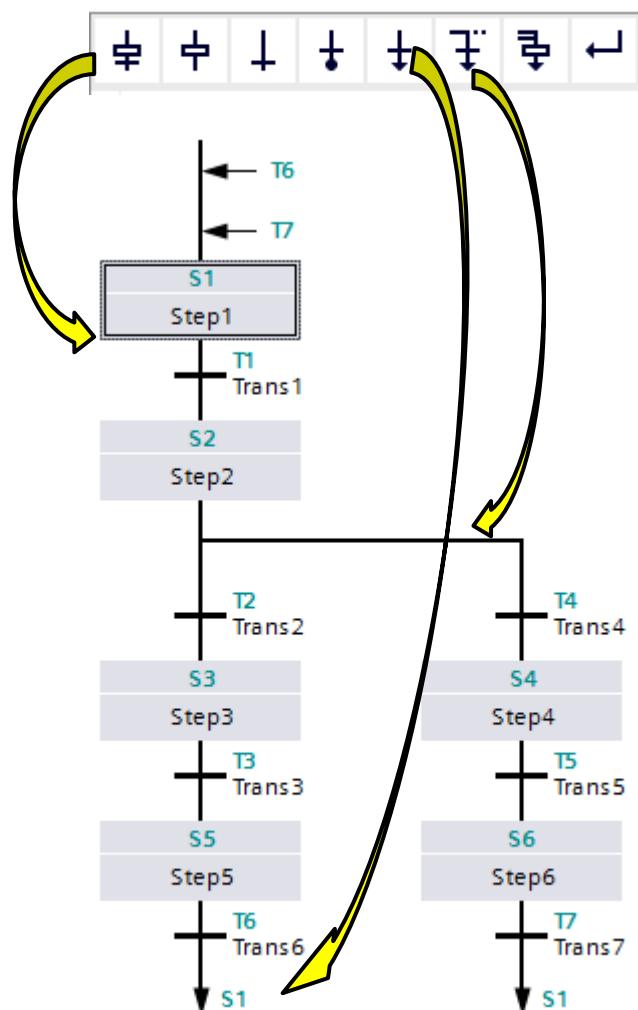


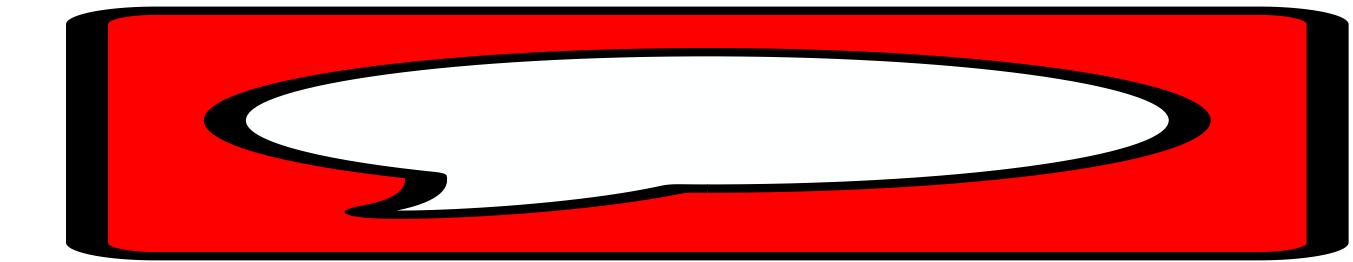
### 3.2.6 Création du Grafset

L'exercice proposé permet de commander de façon basique la station Festo MPS Sorting. Cette station permet de trier les pièces en fonction de leur couleur (rouge, noir, métallisé).

A la fin de l'exercice, la station fera le tri entre les pièces rouges et les pièces métallisées.

Pour créer le Grafset, il suffit de glisser les éléments désirés dans la fenêtre du Grafset. Des carrés verts apparaissent à l'endroit où le composant sélectionné pourra être déposé.





### 3.2.7 Variables utiles pour la programmation du Grafset

Ajoutez les variables suivantes à la table des variables :

→[!]	Tag_1	Table de variables s...	Bool	%M0.0
→[!]	Part_av	Table de variables s...	Bool	%I136.0
→[!]	B2	Table de variables s...	Bool	%I136.1
→[!]	B3	Table de variables s...	Bool	%I136.2
→[!]	B4	Table de variables s...	Bool	%I136.3
→[!]	Bstart	Table de variables s...	Bool	%I137.0
→[!]	Bstop	Table de variables s...	Bool	%I137.1
→[!]	Tapis	Table de variables s...	Bool	%Q136.0
→[!]	1M1	Table de variables s...	Bool	%Q136.1
→[!]	1M2	Table de variables s...	Bool	%Q136.2
→[!]	1M3	Table de variables s...	Bool	%Q136.3
→[!]	LStart	Table de variables s...	Bool	%Q137.0

### 3.2.8 Gracet

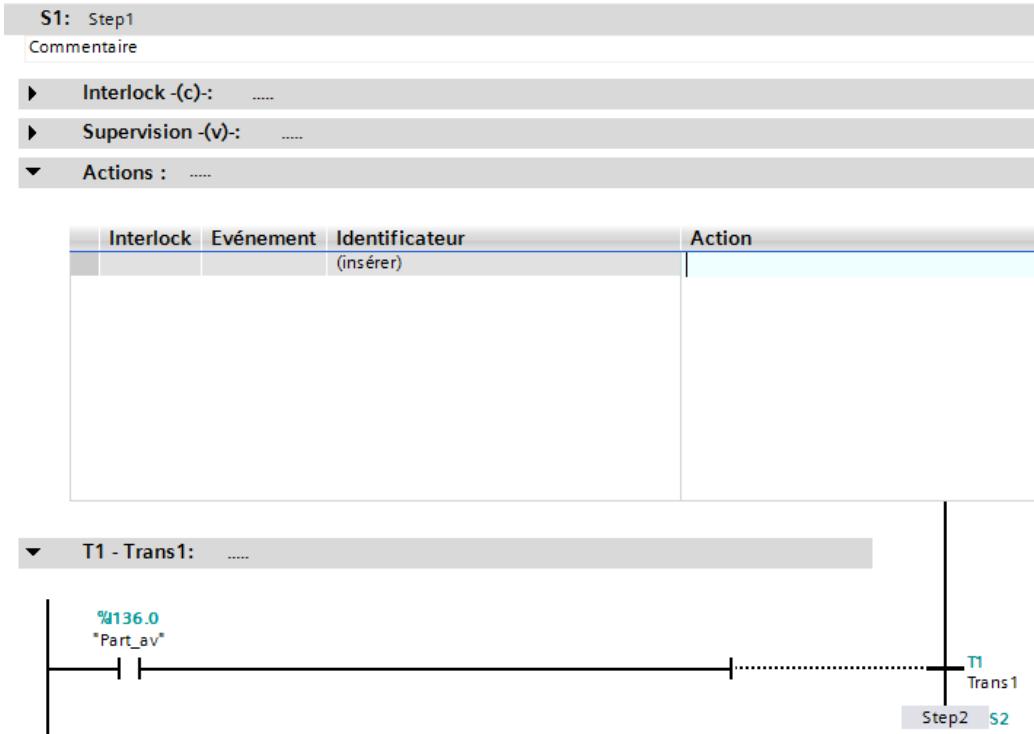
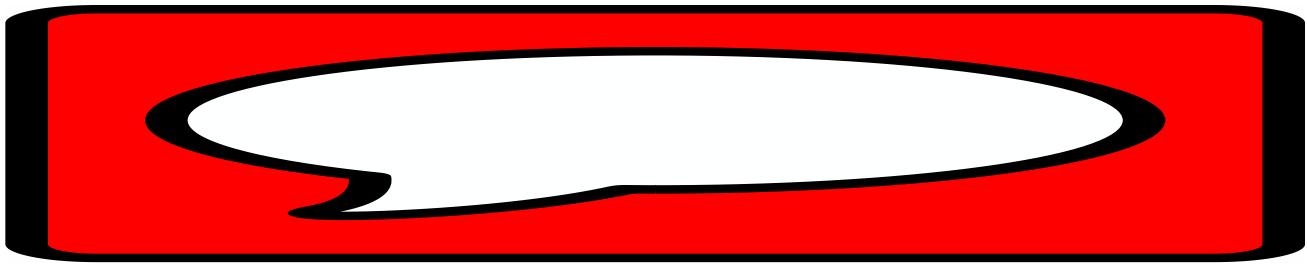
#### 3.2.8.A SI, TI

S1: /

T1: Part\_av = 1

- La condition est remplie si un cylindre est détecté à l'entrée.





### 3.2.8.B S2, T2 & T4

**S2: S Mettre à 1 Tapis**

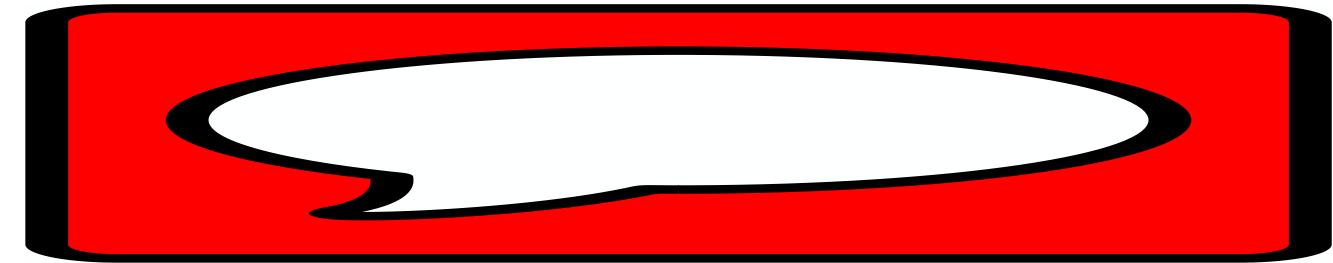
S Mettre à 1 pendant la durée de l'étape LStart

**T2: Bstart=1 X B2=0 X B3=1**

**T4: Bstart=1 X B2=1 X B3=1**

- En fonction des indications des capteurs, on choisira une branche du Grafcet.
- On allume la lumière du bouton Start pour signaler à l'opérateur qu'il doit intervenir.





## S2: Step2

Commentaire

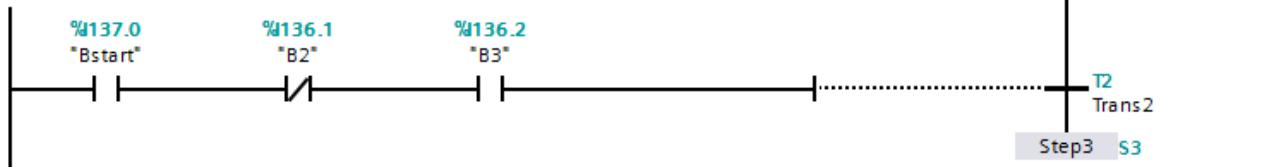
▶ Interlock -(c)-: ....

▶ Supervision -(v)-: ....

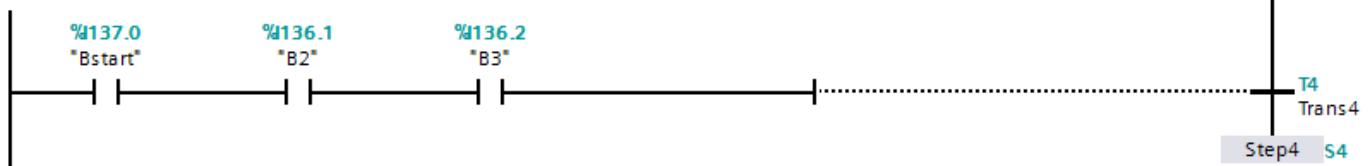
▼ Actions : ....

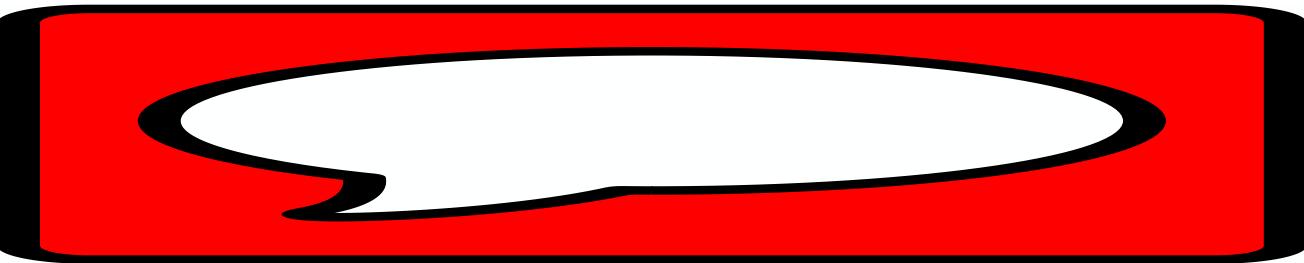
Interlock	Événement	Identificateur	Action
	S	- Mettre à 1	"Tapis"
	D	- Retard à la montée	"Tag_1", T#1S
	N	- Mise à 1 tant que l'étape est active (insérer)	"LStart"

▼ T2 - Trans2: ....



▼ T4 - Trans4: ....





### 3.2.8.C S3, T3

**S3:** N Mettre à 1 pendant la durée de l'étape 1M1 & 1M2

**T3:** B4 = 1

- On sort le premier bras et on retire la barrière qui retenait le cylindre.
- L'étape se termine lorsque la barrière optique détecte une pièce.

**S3:** Step3

Commentaire

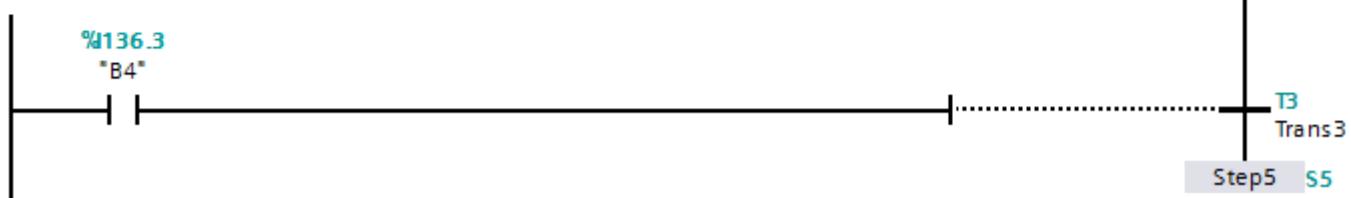
▶ Interlock -(c)-: ....

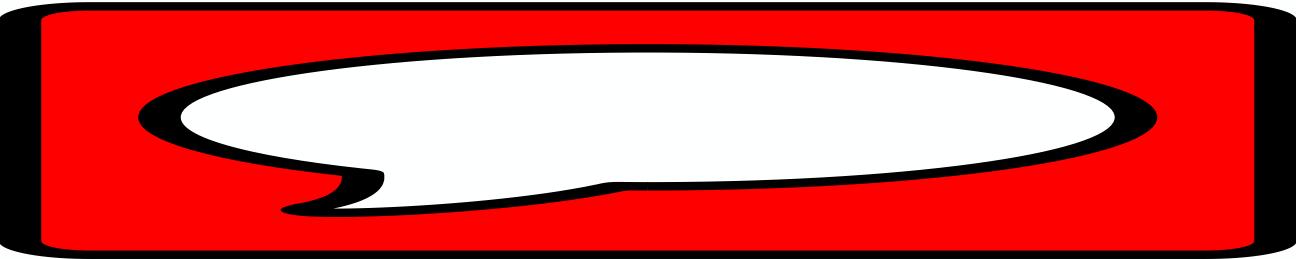
▶ Supervision -(v)-: ....

▼ Actions : ....

Interlock	Événement	Identificateur	Action
		N - Mise à 1 tant que l'étape est active	"1M1"
		N - Mise à 1 tant que l'étape est active	"1M3"
		(insérer)	

▼ T3 - Trans3: ....





### 3.2.8.D S5, T6

#### R Mettre à 0 Tapis

- On arrête le tapis puis on retourne à l'étape initiale

S5: Step5			
Commentaire			
▶ Interlock -(c)-: ....			
▶ Supervision -(v)-: ....			
▼ Actions : ....			

Interlock	Événement	Identificateur	Action
		R - Mettre à 0 (insérer)	"Tapis"

### 3.2.8.E S4, T5

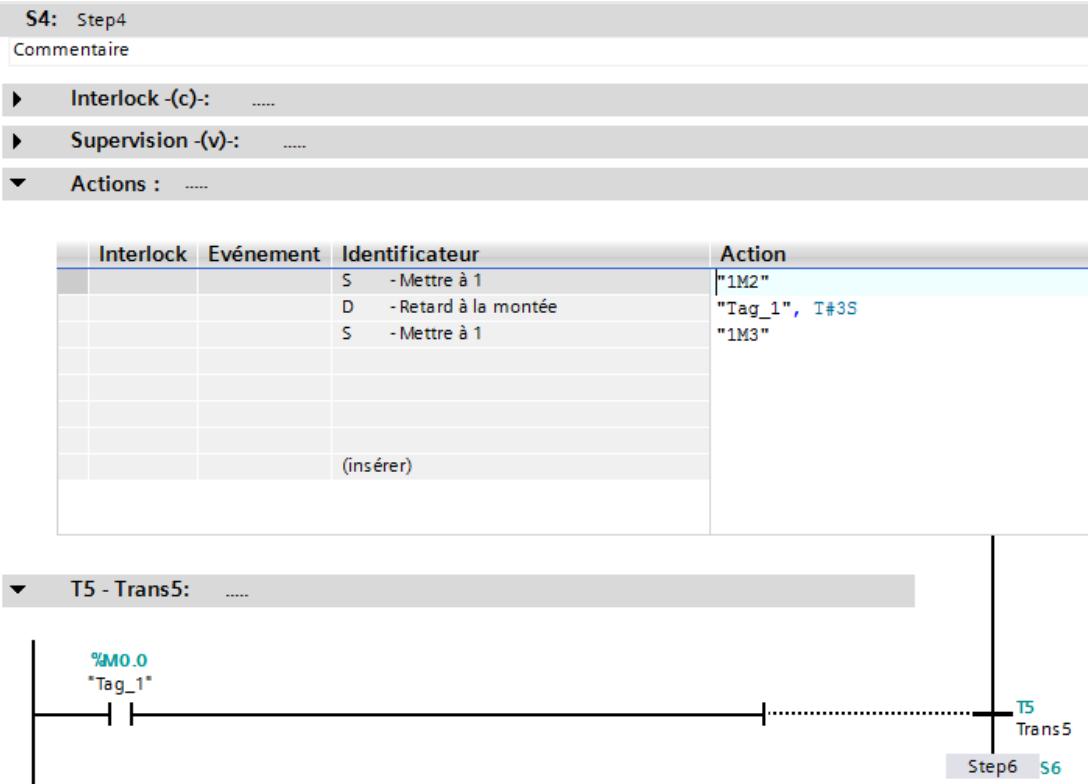
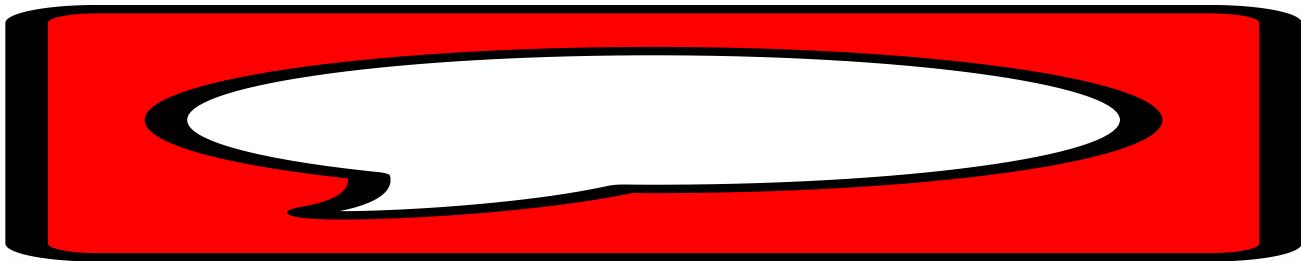
#### S Mettre à 1: 1M1 & 1M3

#### D Retard à la montée : M0.0 3s

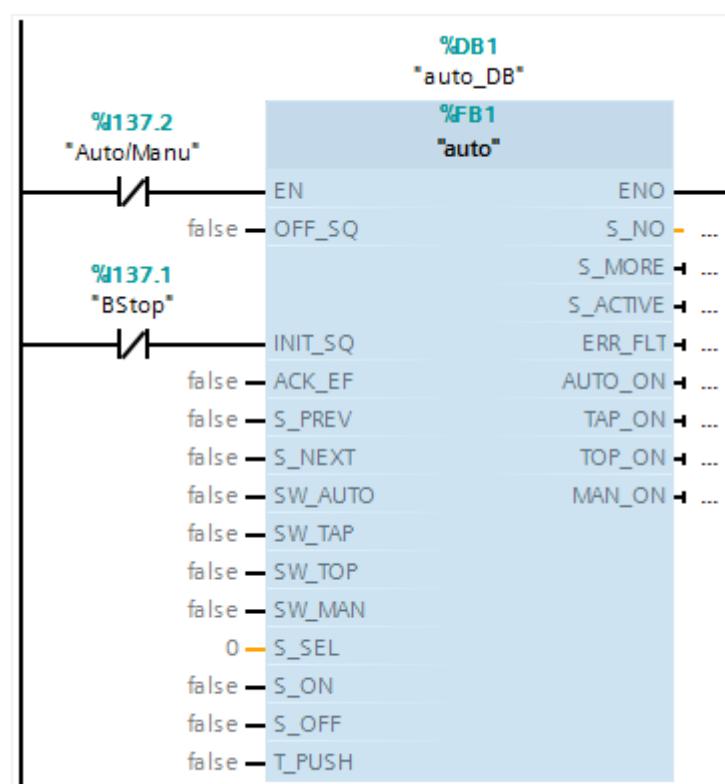
#### T5 : M0.0 = 1

- On sort le deuxième bras et on retire la barrière qui retenait le cylindre
- On démarre une temporisation de 3 secondes
- L'étape se termine lors de la fin de la temporisation





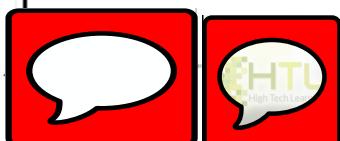
### 3.2.9 Remise à zéro du Grafcet

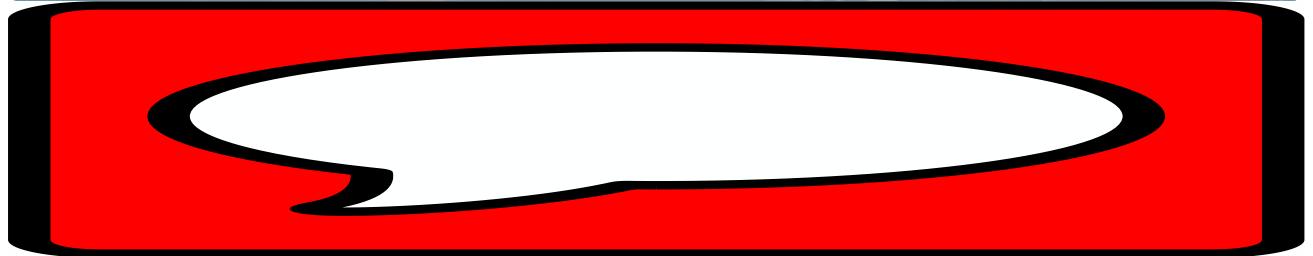


Lors d'un arrêt du cycle normal, il est intéressant de pouvoir revenir à l'étape initiale du Grafcet.

Pour réaliser cette action, il existe des entrées sur le bloc d'appel du Grafcet. Pour forcer le Grafcet à revenir à son étape initiale, il faut mettre à 1 l'entrée « INIT\_SQ ». Dans l'exemple ci-contre, c'est le bouton Stop qui est utilisé pour réaliser la remise à zéro du Grafcet.

**Attention, après avoir remis un Grafcet à zéro, il est nécessaire de replacer les composants du système dans leur position initiale.**





#### **4 POUR ALLER PLUS LOIN...**

**4.1 Modifiez le titre des étapes et des transitions et ajoutez des commentaires pour que le Grafcet soit plus facile à comprendre.**

S1 : .....

T1 : .....

S2 : .....

T2 : .....

T4 : .....

S3 : .....

T3 : .....

S5 : .....

T6 : .....

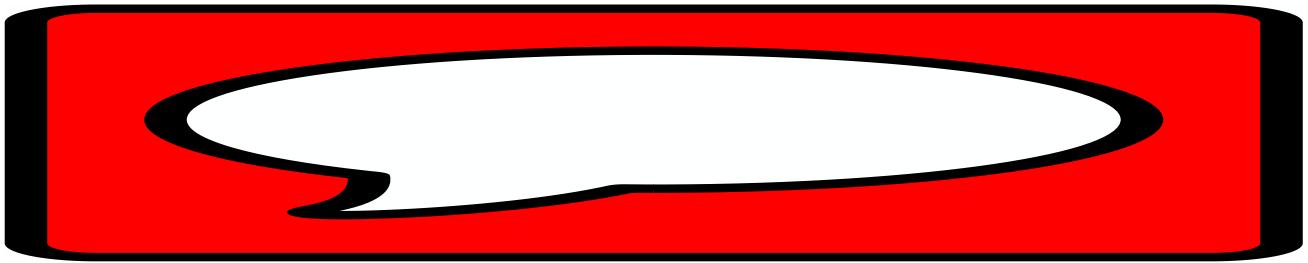
S4 : .....

T5 : .....

S6 : .....

T7 : .....





#### 4.1 Modifiez le Grafcet pour qu'il ne nécessite plus l'appui sur le bouton Start.

Que devient l'étape 2 ?

.....  
.....  
.....  
.....

Que devient la transition 2 ?

.....  
.....

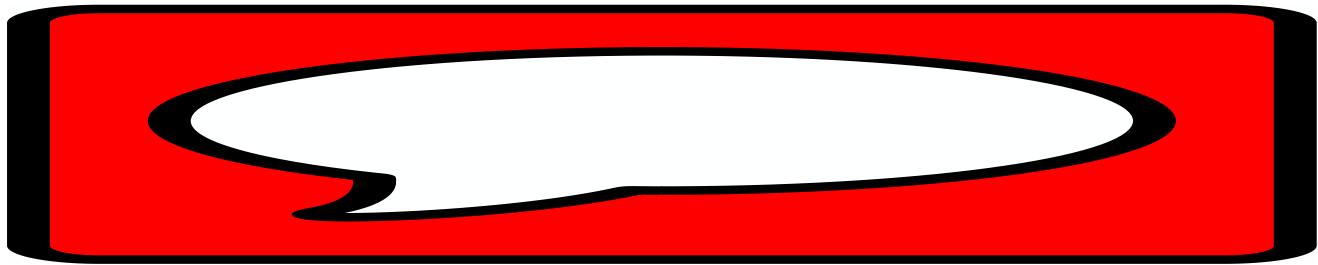
Schéma à contact :

Que devient la transition 4 ?

.....  
.....

Schéma à contact :





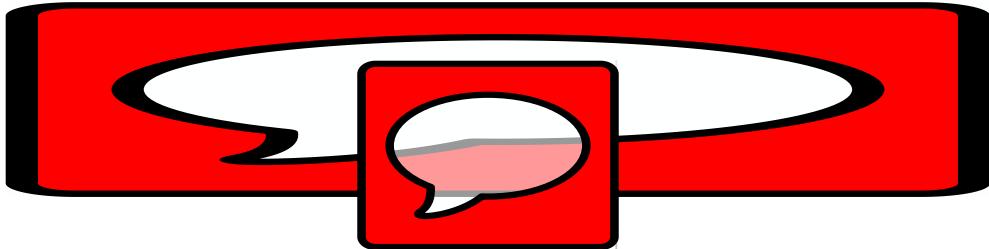
## 4.2 Modifiez le Grafcet pour prendre en compte les cylindres de couleur noire.

Quelles doivent être les valeurs des capteurs pour les pièces suivantes ?

	Rouge	Noire	Métallisée
B2			
B3			

Représentez le nouveau Grafcet :

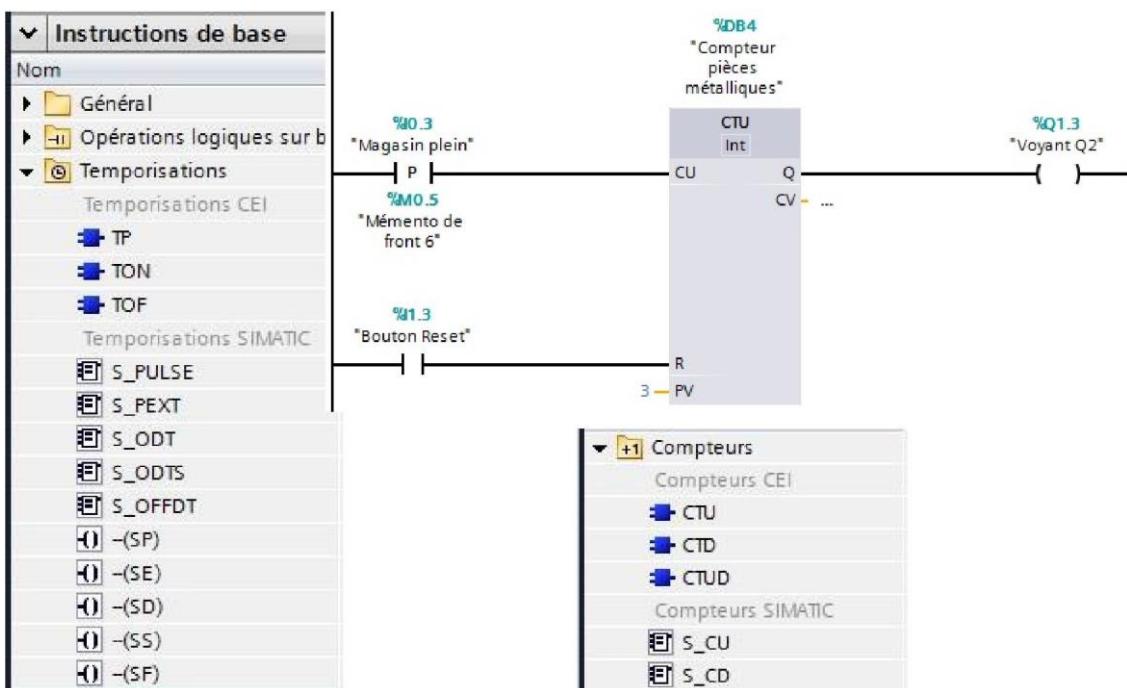




## Atelier C - TIA Portal



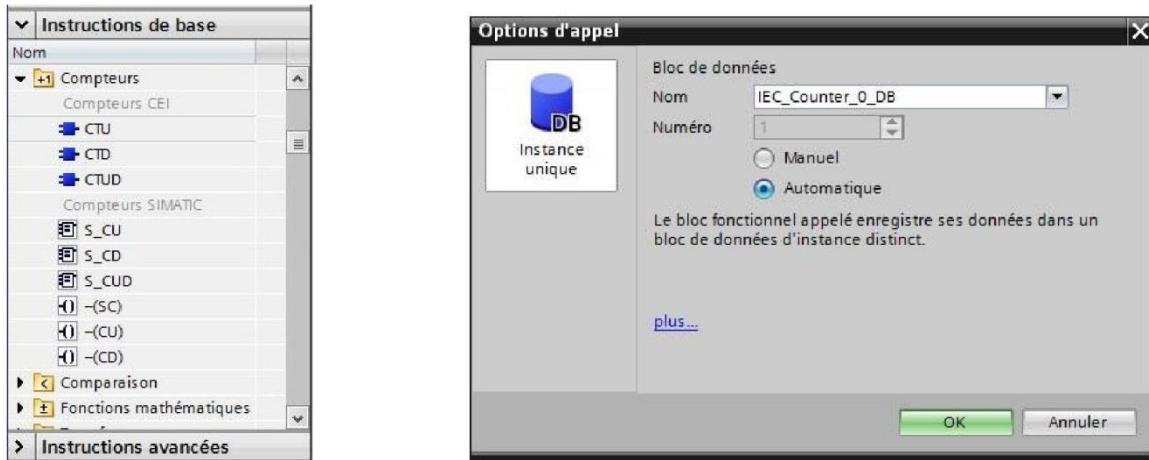
### Programmation des automates S7-300 Compteurs et Temporisations



## 1. Les compteurs

Il existe deux types de compteurs utilisables dans S7 : les compteurs CEI et les compteurs SIMATIC.

Les compteurs SIMATIC n'étant plus utilisés avec les nouvelles CPU (S7 – 1200), ils ne seront pas traités dans ce module.

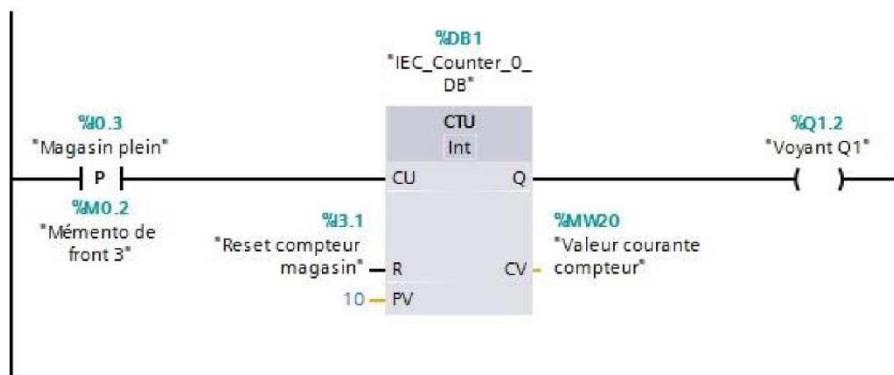


### 1.1 Les compteurs CEI

Il existe 3 types de compteurs CEI : **CTU** (comptage), **CTD** (décomptage) et **CTUD** (comptage et décomptage). A chaque compteur CEI est associée un DB.

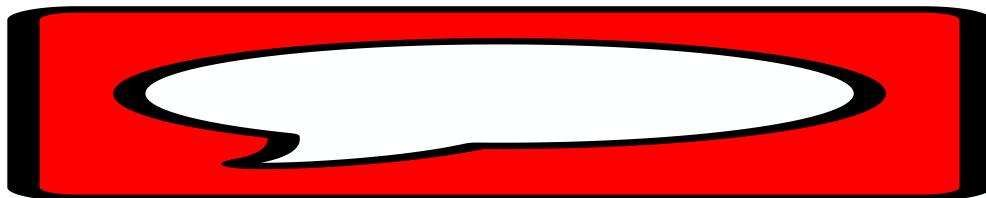
#### Compteur CTU : comptage

L'instruction « **Comptage** » permet d'incrémenter la valeur à la sortie CV.



**CU** → Entrée du compteur. La valeur du compteur est incrémentée de un lorsque l'opérande relié à cette entrée passe de « 0 » à « 1 ». Le compteur peut être incrémenté jusqu'à ce que la valeur supérieure du type de donnée (INT) soit atteinte.

**R** → Entrée de réinitialisation. La valeur de sortie CV (valeur de comptage en cours) est remise à zéro lorsque l'entrée R passe à « 1 ». Tant que R est à « 1 », l'entrée CU n'a pas d'effet sur l'instruction.



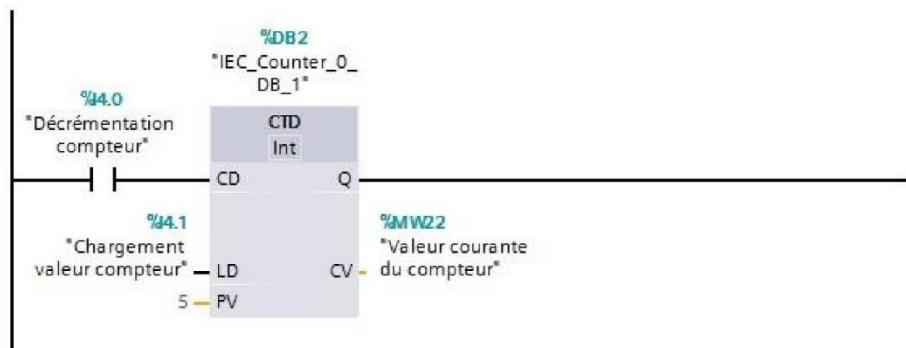
PV → Valeur à laquelle la sortie Q est mise à « 1 ». Cette entrée doit être de type INT.

Q → Etat du compteur. Cette sortie est mise à « 1 » lorsque la valeur actuelle de comptage est supérieure ou égale à la valeur PV.

CV → Valeur de comptage en cours. Cette donnée doit être de type INT.

### *Compteur CTD : décompte*

La sortie du compteur est mise à « 1 » lorsque la valeur courante du compteur est égale à « 0 ».



CD → Entrée du compteur. La valeur du compteur est décrémentée de un lorsque l'opérande relié à cette entrée passe de « 0 » à « 1 ».

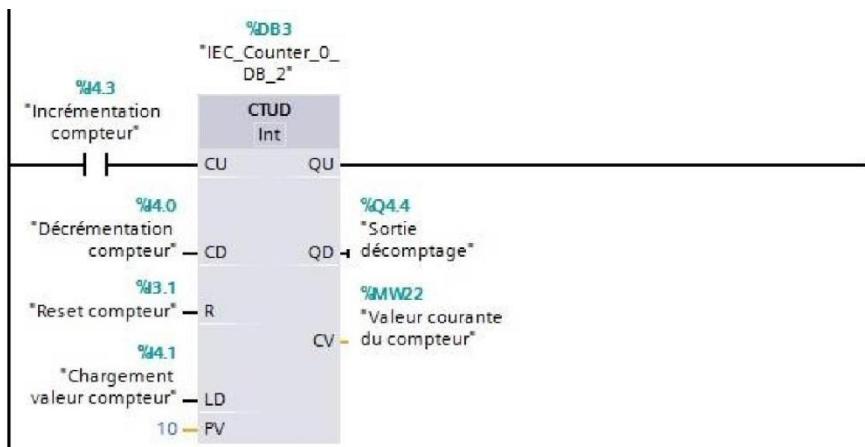
LD → Entrée de chargement. Lorsque cette entrée est à « 1 », la valeur PV est chargée dans le compteur.

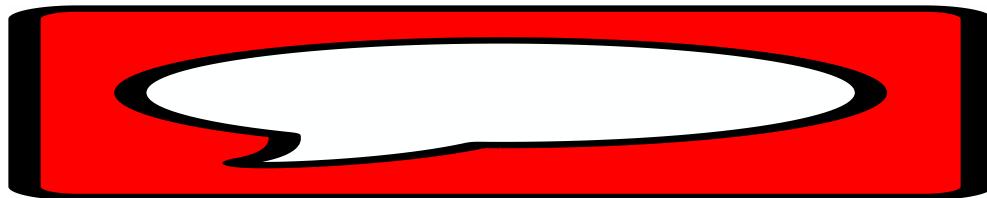
PV → Valeur chargée dans le compteur lorsque LD passe à « 1 ». Cette entrée doit être de type INT.

Q → Etat du compteur. Cette sortie est mise à « 1 » lorsque la valeur actuelle de comptage est inférieure ou égale à 0.

CV → Valeur de comptage en cours. Cette donnée doit être de type INT.

### *Compteur CTUD : comptage et décompte*





**CU** ➔ Entrée de comptage. La valeur du compteur est incrémentée de un lorsque l'opérande reliée à cette entrée passe de « 0 » à « 1 ».

**CD** ➔ Entrée de décomptage. La valeur du compteur est décrémentée de un lorsque l'opérande reliée à cette entrée passe de « 0 » à « 1 ».

**R** ➔ Entrée de réinitialisation. La valeur de sortie CV est remise à zéro lorsque l'entrée R passe à « 1 ». Tant que R est à « 1 », l'entrée CU n'a pas d'effet sur l'instruction.

**LD** ➔ Entrée de chargement. Lorsque cette entrée est à « 1 », la valeur PV est chargée dans le compteur.

**PV** ➔ Valeur chargée dans le compteur lorsque LD passe à « 1 ». Cette entrée doit être de type INT. La sortie comptage (QU) est mise à « 1 » lorsque le compteur atteint la valeur PV.

**QU** ➔ Sortie comptage. Cette sortie est mise à « 1 » lorsque la valeur actuelle de comptage est supérieure ou égale à la valeur PV.

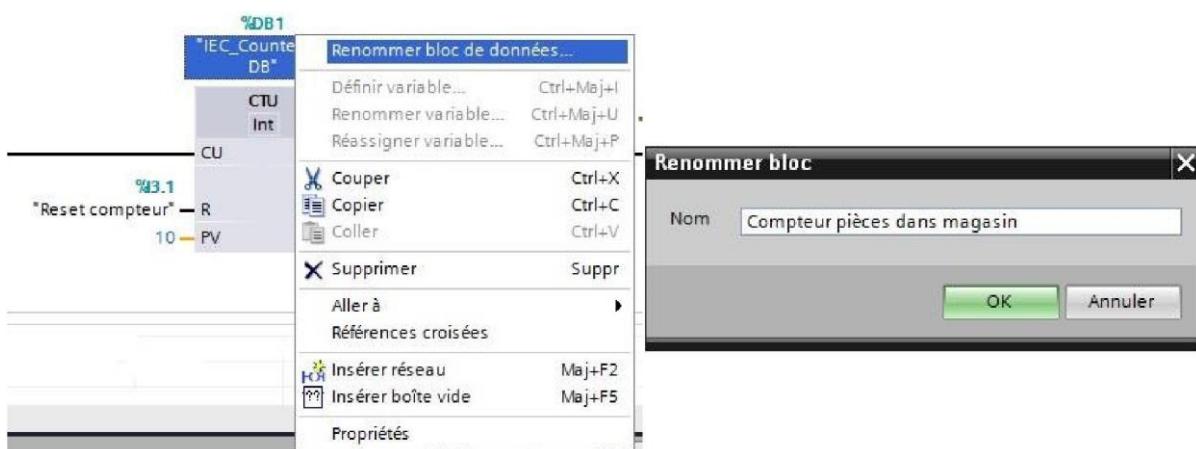
**QD** ➔ Sortie décomptage. Cette sortie est mise à « 1 » lorsque la valeur actuelle de comptage est inférieure ou égale à « 0 ».

**CV** ➔ Valeur de comptage en cours. Cette donnée doit être de type INT.

## 1.2 Renommer le bloc de donnée

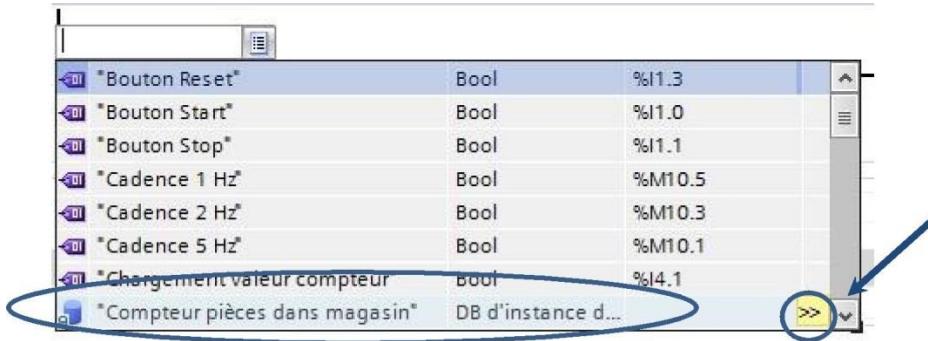
Lorsque l'on insert un compteur CEI dans un bloc de programme, le nom du bloc de donnée qui lui est associé est attribué par défaut. Il est possible de modifier ce nom :

- Lors de l'insertion du compteur, dans la fenêtre « **Option d'appel** »
- Une fois le compteur inséré, en faisant en clic droit sur son nom et en choisissant l'option « **Renommer bloc de donnée** »



### 1.3 Utilisation des données du compteur dans un programme

Il est possible d'assigner une variable liée au compteur à un symbole d'interrogation à 1 (ou à 0). Pour cela, il faut utiliser le menu permettant d'avoir accès aux variables pouvant être associées au symbole et de choisir le nom du compteur puis la variable associée (CU, CUO, Q, R).



Nom du compteur

"Compteur pièces dans magasin".		
← DI CU	Bool	
← DI CUO	Bool	
← DI Q	Bool	
← DI R	Bool	

Choix de la variable associée

Dans l'exemple ci-dessous, c'est la sortie du compteur « Compteur pièces dans magasin » qui est associée au symbole d'interrogation à 1.

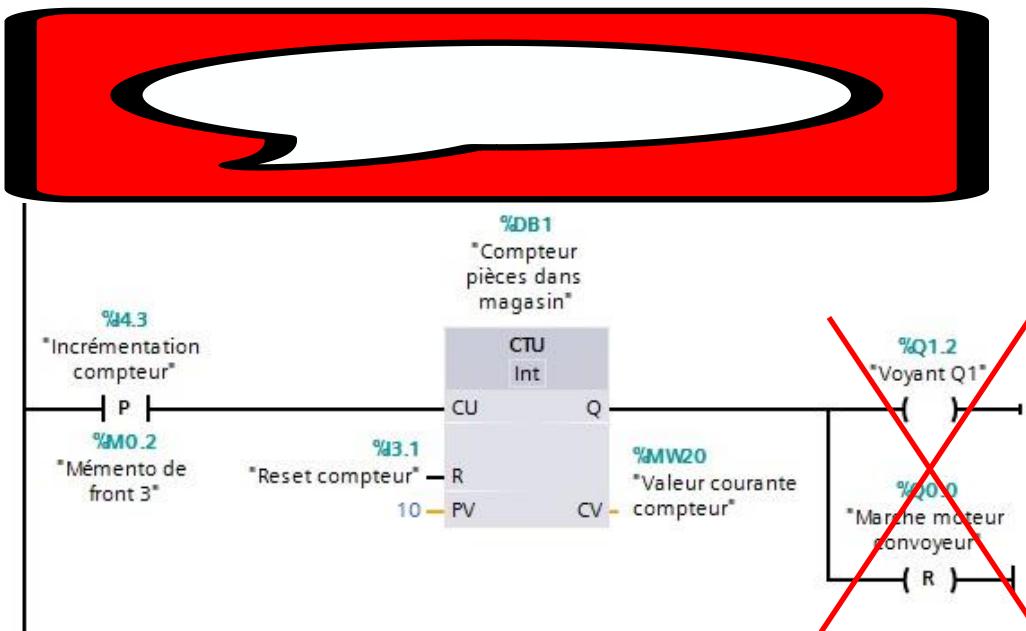
%DB1.DBX4.0  
"Compteur  
pièces dans  
magasin".Q



### 1.4 Limites de la connexion des compteurs CEI

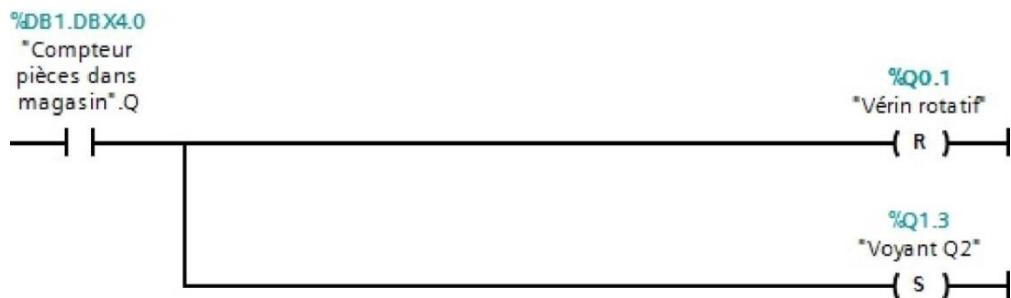
Il faut veiller à respecter la connexion des compteurs. Les règles suivantes s'appliquent à l'utilisation des temporisations CEI et des compteurs CEI sur le système cible :

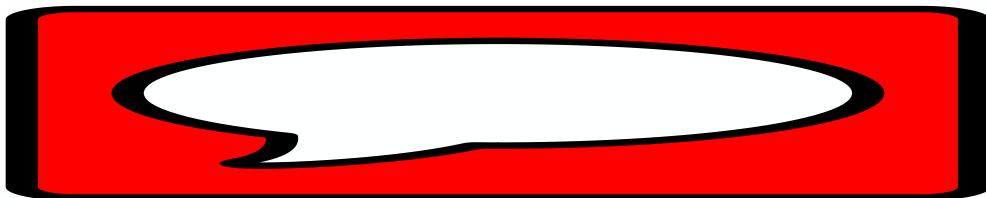
- La sortie Q ne peut être connectée qu'à une bobine. Il n'est pas possible d'utiliser une bobine SET ou RESET.
- L'entrée CU ne doit pas être connectée par branchement T à une autre entrée CU.



Pour contourner ces restrictions, on peut utiliser des mémentos ou encore des contacts NO ou NF auxquels sont associés des variables liées au compteur.

Dans l'exemple ci-dessous, on met à « 0 » la sortie « Vérin rotation » et on met à « 1 » la sortie « Voyant Q2 » lorsque la sortie du compteur est à « 1 ».





## 2. Les temporisations

Comme pour les compteurs, il existe deux types de compteur utilisable dans S7 : les temporisations CEI et les temporisations SIMATIC.

Les temporisations SIMATIC n'étant plus utilisées avec les nouvelles CPU (S7 – 1200), elles ne seront pas traitées dans ce module.

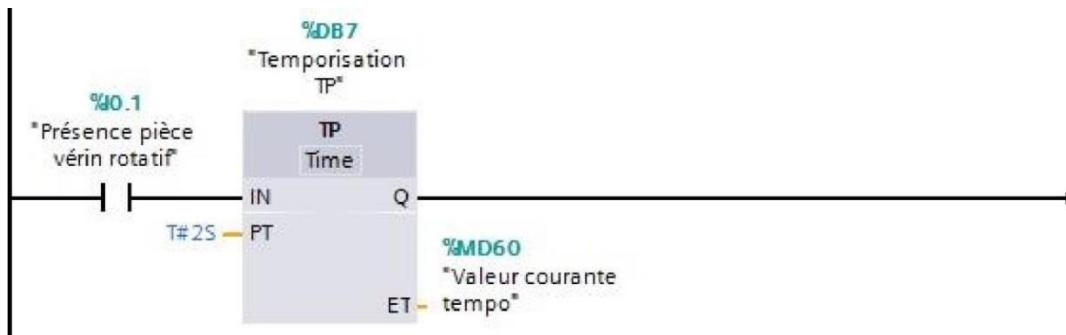
### 2.1 Les temporisations CEI

Il existe 3 types de temporisations CEI : **TP** (génération d'impulsion), **TON** (retard à la montée) et **TOF** (retard à la retombée). A chaque compteur CEI est associée un DB.

#### Temporisation TP

L'instruction « **TP** » met la sortie Q à 1 pour une durée programmée. L'instruction est démarrée lorsque le résultat logique à l'entrée IN passe de « 0 » à « 1 » (front montant du signal). La durée PT programmée commence à s'écouler au démarrage de l'instruction.

La sortie Q est mise à « 1 » pour la durée PT, indépendamment de l'évolution du signal d'entrée. Même la détection d'un nouveau front montant du signal n'a pas d'influence sur l'état logique à la sortie Q tant que la durée PT n'est pas entièrement écoulée.

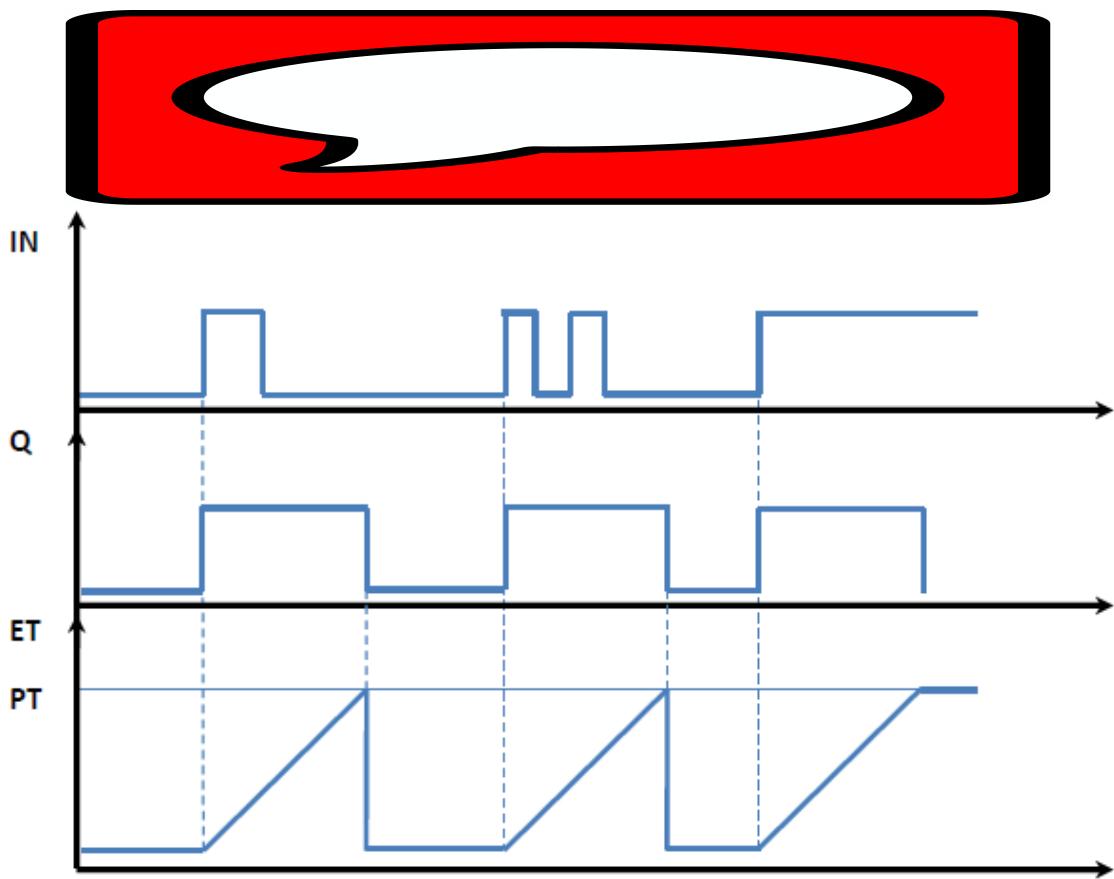


**IN** → Entrée de démarrage : la temporisation est lancée lorsque l'opérande lié à cette entrée passe de « 0 » à « 1 ».

**PT** → Durée de l'impulsion. Cette valeur doit être positive.

**Q** → Sortie de la temporisation. La sortie Q est mise à « 1 » pour la durée PT, indépendamment de l'évolution du signal d'entrée. Même la détection d'un nouveau front montant du signal n'a pas d'influence sur l'état logique à la sortie Q tant que la durée PT n'est pas entièrement écoulée.

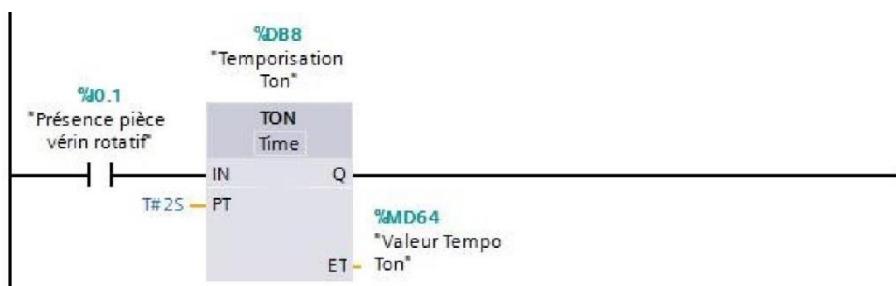
**ET** → Valeur de temps actuelle : la valeur de temps commence à T#0s et s'arrête lorsque la durée PT est atteinte. Lorsque la durée PT est atteinte et que l'état logique à l'entrée IN est « 0 », la sortie ET est remise à 0. C'est une variable de type « time ».



### Temporisation Ton

L'instruction « **Ton** » retarde la mise à « 1 » de la sortie Q de la durée programmée PT. L'instruction est démarrée lorsque le résultat logique à l'entrée IN passe de « 0 » à « 1 ». La durée PT programmée commence à s'écouler 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 de démarrage fournit "1". Lorsque l'état logique à l'entrée de démarrage passe de « 1 » à « 0 », la sortie Q est remise à 0. La fonction de temporisation est redémarrée lorsqu'un nouveau front montant est détecté à l'entrée de démarrage.



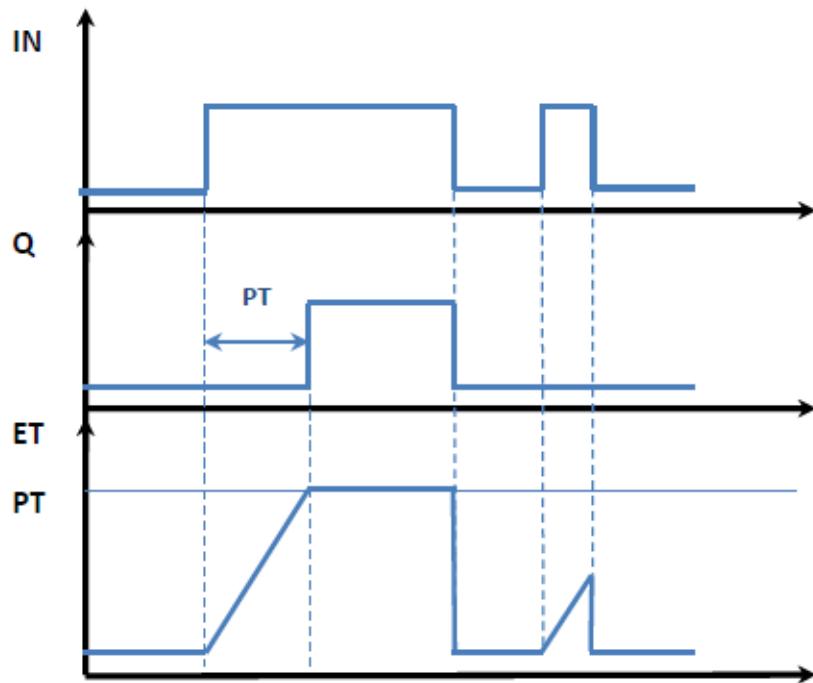
**IN** → Entrée de démarrage : la temporisation est lancée lorsque l'opérande lié à cette entrée passe de « 0 » à « 1 ».

**PT** → Durée du retard à la montée. Cette valeur dit être positive.

**Q** → Sortie de la temporisation. La sortie Q est mise à « 1 » après l'écoulement de la temporisation PT. Lorsque l'état logique à l'entrée de démarrage passe de « 1 » à « 0 », la sortie Q est remise à « 0 ». La fonction de temporisation est redémarrée lorsqu'un nouveau front montant est détecté à l'entrée de démarrage.

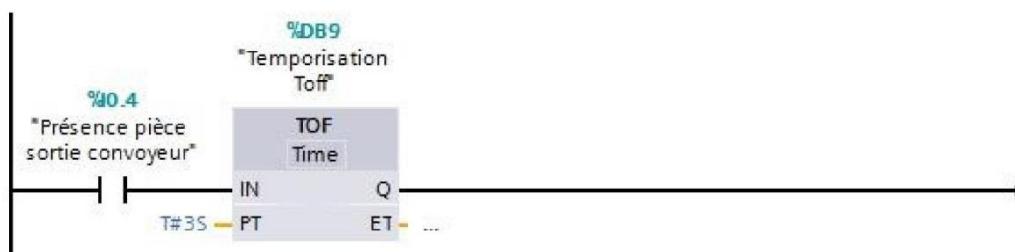


**ET** → Valeur de temps actuelle : la valeur de temps commence à T#0s et s'arrête lorsque la durée PT est atteinte. Lorsque la durée PT est atteinte et que l'état logique à l'entrée IN est « 0 », la sortie ET est remise à 0. C'est une variable de type « time ».



### Temporisation Tof

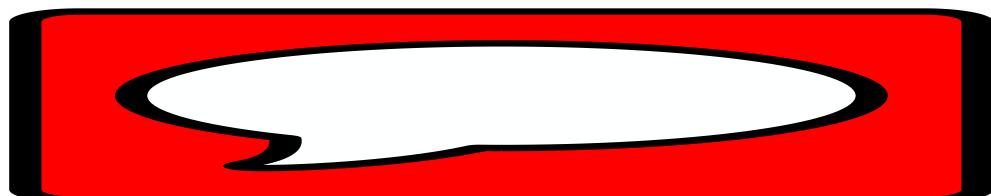
L'instruction « **Tof** » permet de retarder la mise à « 0 » de la sortie Q de la durée programmée PT. La sortie Q est mise à 1 lorsque le résultat logique à l'entrée IN passe de « 0 » à « 1 ». Lorsque l'état logique à l'entrée IN repasse à « 0 », la durée programmée PT démarre.



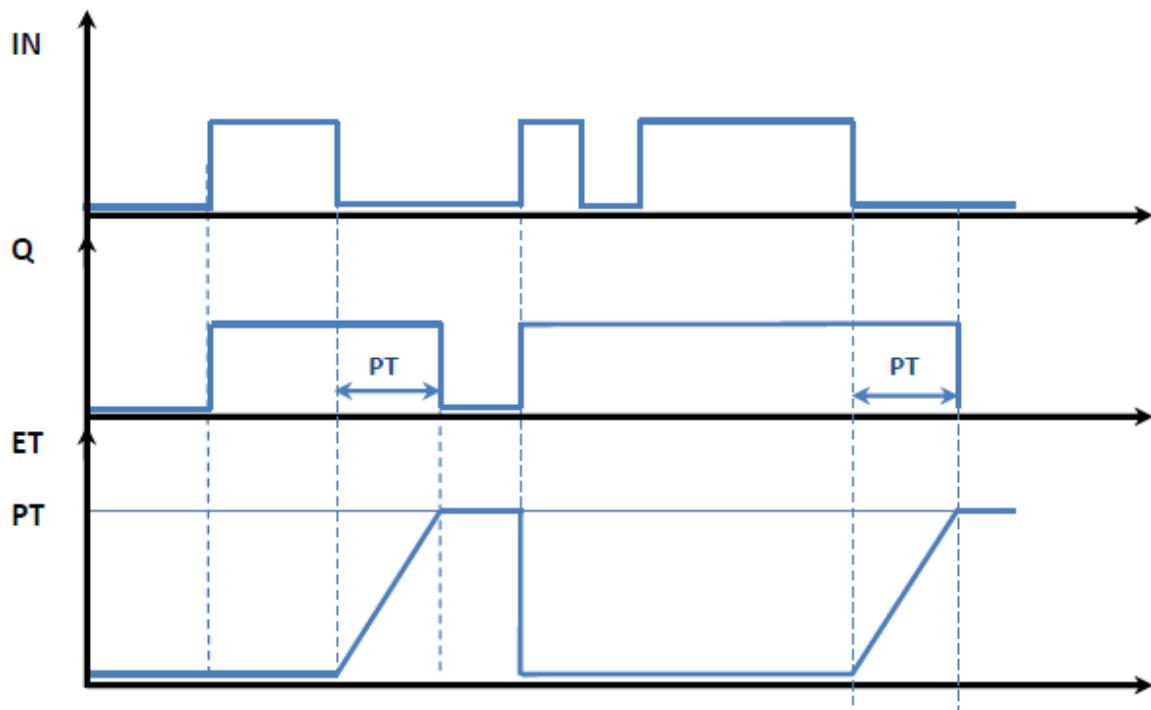
**IN** → Entrée de démarrage : la sortie Q est mise à « 1 » lorsque l'opérande lié à cette entrée passe de « 0 » à « 1 ».

**PT** → Durée du retard à la retombée. Cette valeur doit être positive.

**Q** → Sortie de la temporisation. La sortie Q est mise à « 1 » lorsque l'entrée IN passe de « 0 » à « 1 ». 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 ».

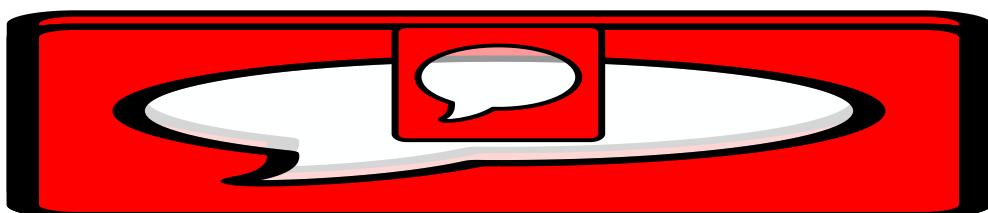
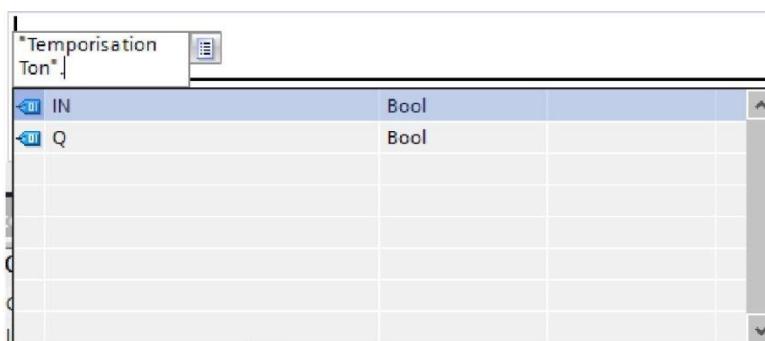


ET → Valeur de temps actuelle : la valeur de temps commence à T#0s et s'arrête lorsque la durée PT est atteinte. Lorsque la durée PT est atteinte et que l'état logique à l'entrée IN est « 0 », la sortie ET est remise à « 0 ». C'est une variable de type « time ».



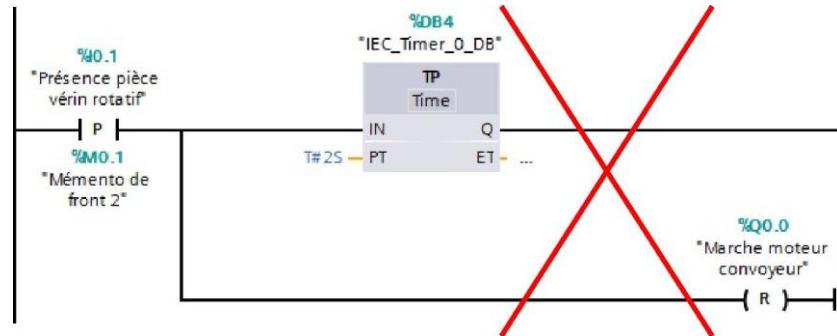
## 2.2 Utilisation des données des temporisations dans un programme

Comme pour les compteurs, il est possible d'assigner une variable liée à la temporisation à un symbole d'interrogation à 1 (ou à 0). Les variables associées peuvent être « IN » et « Q ».



## 2.3 Limites de la connexion des temporisations CEI

Les remarques faites pour les compteurs CEI restent valables pour les temporisations.



## 3. Exercices

Les exercices qui vont suivre seront réalisés à l'aide du logiciel CIROS Mechatronics « Module de projet Ligne de convoyage 4 »

1. Lorsqu'une pièce est présente à l'entrée du convoyeur et que l'on appuie sur le bouton Start, elle est acheminée vers la sortie. Si la pièce est métallique, elle est évacuée dans le magasin de stockage.

Lorsque 3 pièces métalliques se trouvent dans le magasin de stockage, le voyant Q1 s'allume. Le bouton reset permet de réinitialiser la valeur du compteur.

2. Lorsqu'une pièce est présente à l'entrée du convoyeur et que l'on appuie sur le bouton Start, le moteur se met en marche après 2 secondes (tempo 1). Lorsque la pièce se présente devant le détecteur « B2 », le moteur s'arrête durant 3 secondes puis le moteur se remet en marche (tempo 2). Le moteur s'arrête 2 secondes après que la pièce ait atteint le détecteur « B5 » (tempo 3).

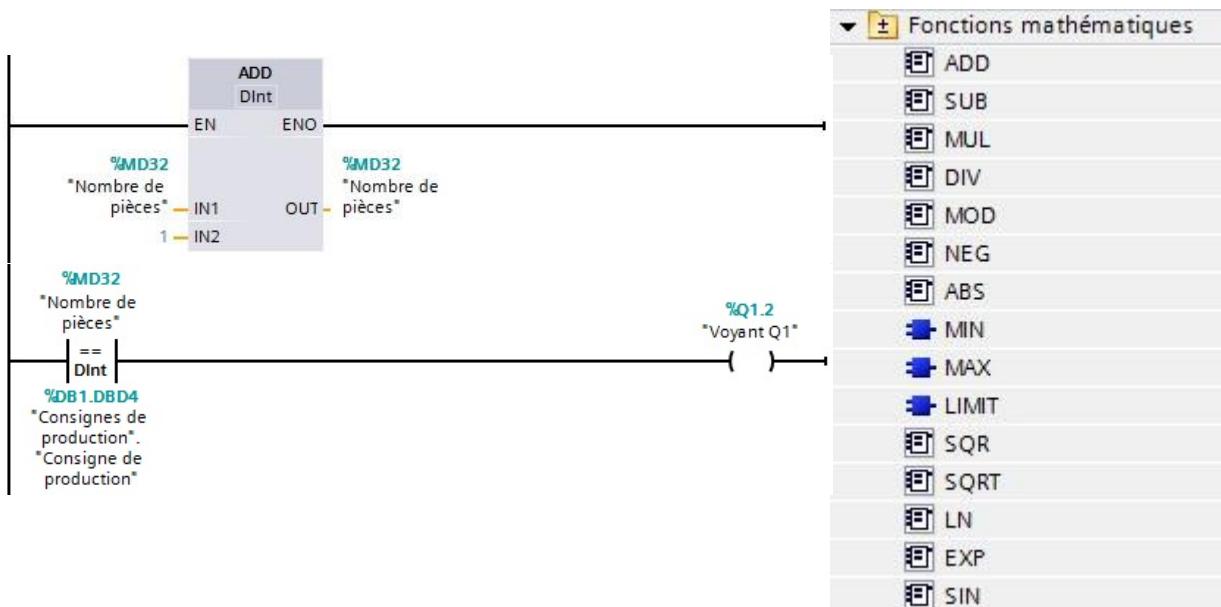
Le type de tempo est imposé :

- Tempo 1 :  $T_{on}$
- Tempo 2 : TP
- Tempo 3 :  $T_{of}$

# Atelier C – TIA Portal



## Programmation des automates S7-300 – Opérations numériques



## 1. Les types de données sous S7 - 300

Il existe plusieurs types de données utilisées pour la programmation des automates Siemens.

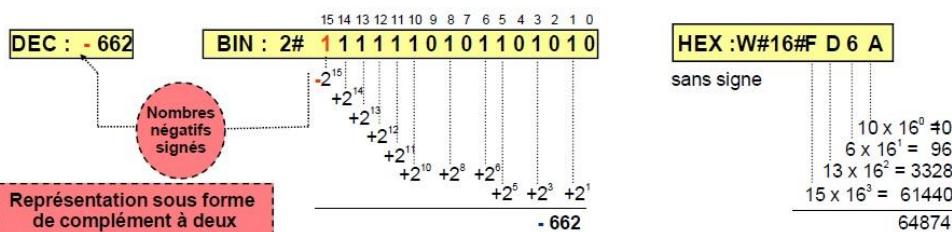
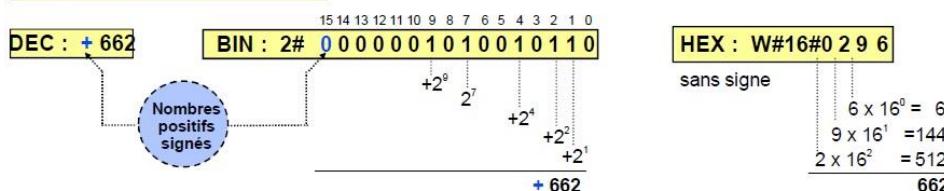
### 1.1 Les entiers INT (entiers signés sur 16 bits)

Un nombre entier est un nombre sans chiffres après la virgule. Ces nombres sont codés sous 16 bits.

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

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

#### Formats de représentation :



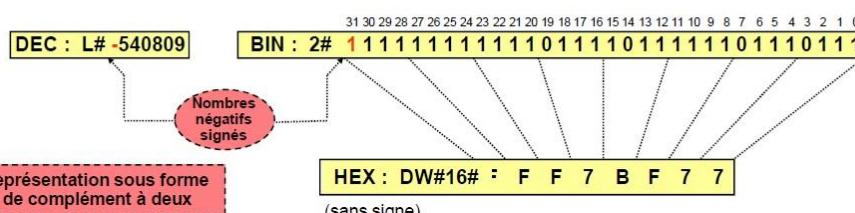
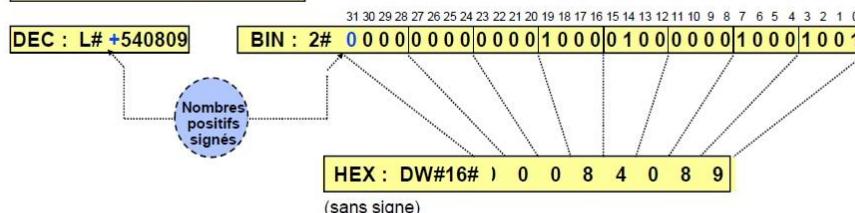
Dans STEP 7, les constantes de type INT sont entrées au format décimal avec signe et sans indication explicite du format. L'entrée d'entiers constants sous forme binaire ou hexadécimale est possible, bien qu'inappropriée.

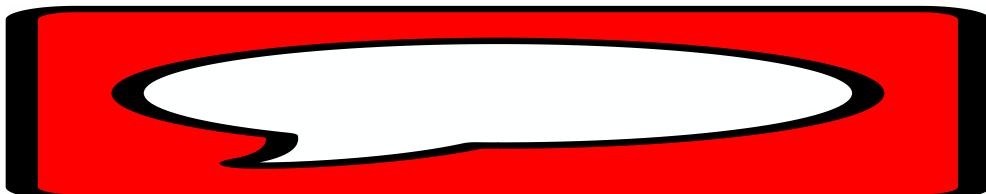
### 1.2 Les doubles entiers DINT (entiers signés sur 32 bits)

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

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

#### Formats de représentation :





Dans STEP 7, les constantes de type de données DINT sont entrées au format décimal avec signe et indication de format L#, où L signifie « long ». Si des valeurs définies sont inférieures à -32768 ou supérieures 32767, l'indication de format L# est ajoutée automatiquement. Elle doit en revanche être entrée par l'utilisateur pour des nombres négatifs supérieurs à -32768 (par exemple L#-1). Cela est indispensable lorsque la valeur est utilisée comme entier double dans une opération mathématique, sinon le traitement s'effectuerait avec une valeur erronée (valeur + signe).

### 1.3 Les réels REAL (nombre à virgule flottante sur 32 bits)

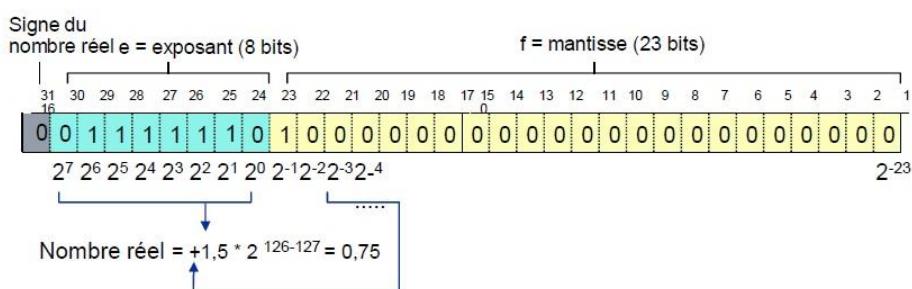
Les types de données INT et DINT s'utilisent pour enregistrer des valeurs signées. Ces valeurs ne permettent donc d'effectuer que des opérations dont le résultat est aussi un nombre entier. Cependant, lorsque les valeurs traitées sont des grandeurs analogiques de type courant, tension ou température, l'utilisation de nombre au format REAL est indispensable.

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

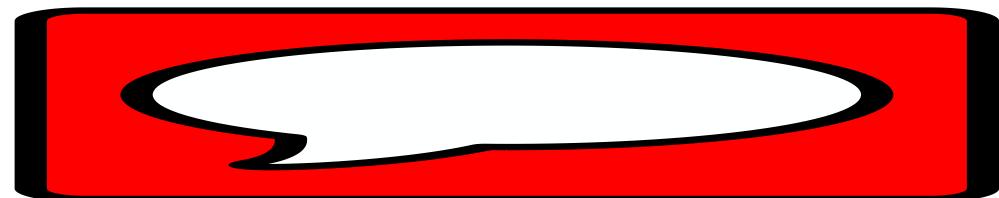
Opérations : par ex.  $+R$ ,  $*R$ ,  $<R$ ,  $==R$   
 $\sin$ ,  $\cos$ ,  $\ln$ ,  $\exp$ ,  $SQR$

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

**Exemple :** 0.75 (Lors de la saisie sur la PG, on utilise un point à la place de la virgule)



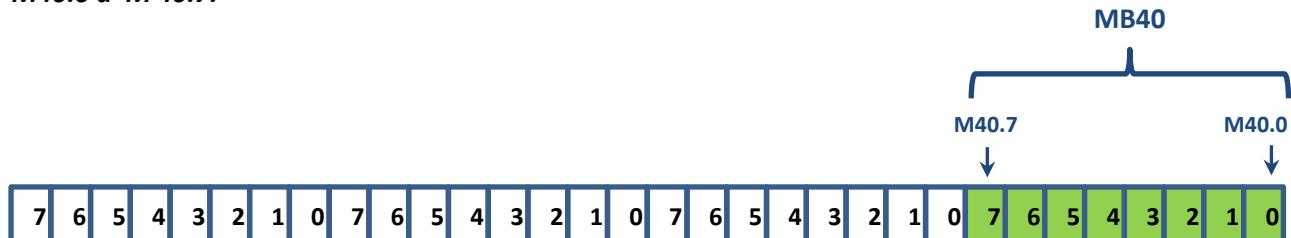
Les nombres à virgule flottante s'utilisent notamment pour le traitement des valeurs analogiques. Leur principal avantage réside dans le fait qu'ils admettent une multitude d'opérations, parmi lesquelles des opérations standard comme +, -, \*, / mais aussi des opérations comme sin, cos, exp, ln,...



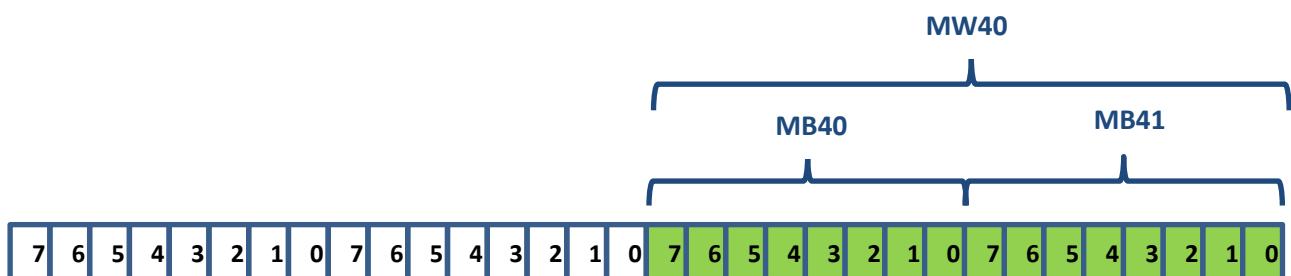
## 2. Byte, Word, Double Word

Il est important de bien savoir comment fonctionnent les bit, byte, mot et double mot.

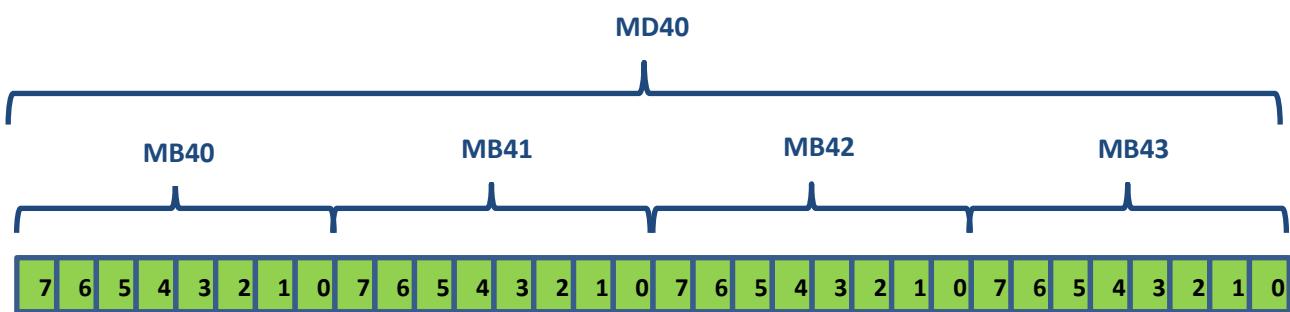
**Byte** → il se compose de 8 bits. Dans l'exemple ci-dessous, le **BYTE MB 40** se compose des bits de **M40.0 à M40.7**.



**Word = mot** → il se compose 16 bits. Dans l'exemple ci-dessous, le **MOT MW 40** se compose de 2 BYTES : **MB 40 et MB 41**. C'est le **bit 7 du BYTE MB 40 qui est le plus significatif**.



**Double Word = double mot** → il se compose de 32 bits. Dans l'exemple ci-dessous, le **DOUBLE MOT MD 40** se compose de 4 BYTES : **MB 40 à MB 43**.



Dans l'exemple ci-dessous, les mémentos 40.0, 40.1 et 40.2 sont à l'état « 1 » et on peut voir le contenu du **Byte** « MB40 », du **Mot** « MW40 » et du **Double Mot** « MW40 ».

	%MB40	Bin	2#0000_0111
	%MW40	Bin	2#0000_0111_0000_0000
	%MD40	Bin	2#0000_0111_0000_0000_0000_0000_0000_0000
"Tag_18"	%M40.0	Bool	TRUE
"Tag_20"	%M40.1	Bool	TRUE
"Tag_19"	%M40.2	Bool	TRUE



Ce tableau est le même que le précédent mais avec un format d'affichage décimal.

	%MB40	DEC	7
	%MW40	DEC	1792
	%MD40	DEC	117440512
"Tag_18"	%M40.0	Bool	<input checked="" type="checkbox"/> TRUE
"Tag_20"	%M40.1	Bool	<input checked="" type="checkbox"/> TRUE
"Tag_19"	%M40.2	Bool	<input checked="" type="checkbox"/> TRUE

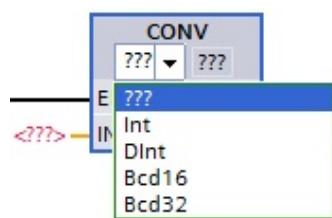
Il faut faire très attention lorsque l'on utilise les MB, MW et MD à ne pas avoir de conflit. En effet, si par exemple vous utilisez une donnée de type real dans MD40, vous ne pouvez plus utiliser les MB 40, 41, 42 et 43 ni MW41,....

Le programme ne détectera aucune erreur de compilation mais vos données seront écrasées et le programme ne fonctionnera pas.

### 3. Fonction de conversion

L'instruction « **Convert** » lit le contenu du paramètre IN et le convertit conformément aux types de données choisis dans l'instruction. La valeur convertie est fournie à la sortie OUT.

Toutes les conversions ne sont pas permises. On peut choisir le format de conversion à l'aide du menu déroulant qui est disponible en « IN » et en « OUT » dans l'instruction.

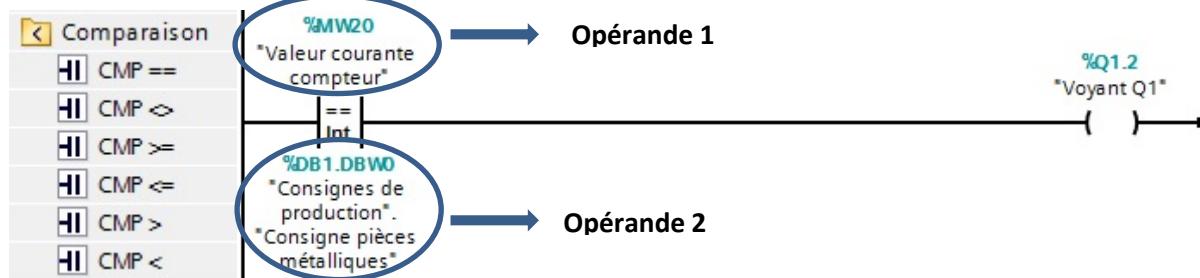


Dans l'exemple ci-dessous, la valeur de l'entrée analogique IW800, qui est de type **INT** est convertie en une valeur de type **DINT**. Ce double entier pourra par la suite être converti en une variable de type **REAL** afin de pouvoir effectuer des opérations mathématiques de type « **DIV** ».



## 4. Les comparaisons

Il existe 6 fonctions de comparaisons possibles : égal, différent, supérieur ou égal, inférieur ou égal, supérieur, inférieur.



Les instructions de comparaison nécessitent 2 opérandes. Le format des opérandes peut être varié : INT, DINT, REAL, BYTE, WORD, DWORD, TIME,....

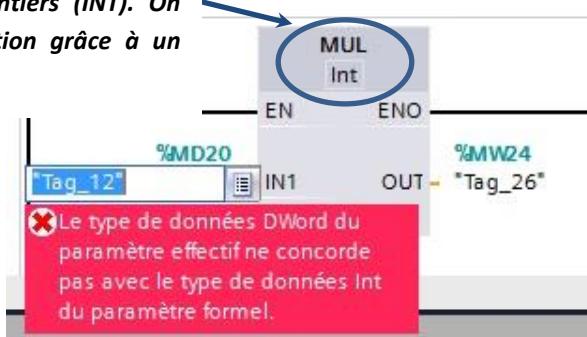
Dans l'exemple ci-dessus, le voyant Q1 s'allume lorsque la « Valeur courante compteur » (opérande 1) est égale à la valeur « Consigne pièces métalliques » (opérande 2) contenue dans un le bloc de donnée « Consignes de production ».

## 5. Les fonctions mathématiques

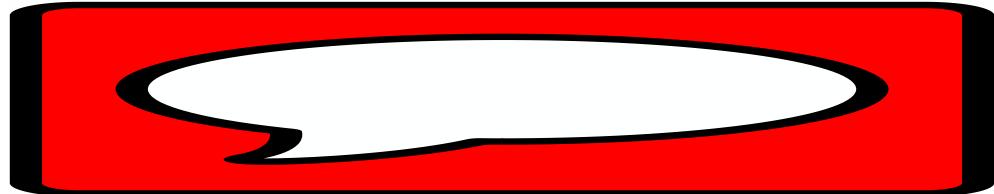
Il existe un grand nombre de fonctions mathématiques utilisable dans STEP7. Citons par exemple les additions, soustractions, multiplication, divisions mais aussi les racines carrées, les fonctions trigonométriques,...

Lorsque l'on utilise ces fonctions, il faut faire attention à bien respecter le format des données.

*L'opération réalisée est une multiplication (MUL) entre des nombres entiers (INT). On choisit le format de l'opération grâce à un menu déroulant.*



Dans l'exemple ci-dessus, on effectue une multiplication de nombres entiers (INT). On ne peut donc pas utiliser comme type de donnée un mot double (ici %MD20). Le type de donnée doit être un Mot (par exemple %MW24).



## 6. Fonctions de transfert : MOVE

L'instruction « Move », permet de transférer le contenu de l'opérande à l'entrée IN dans l'opérande à la sortie OUT1. Le transfert s'effectue toujours dans le sens croissant des adresses.

Dans l'exemple ci-dessous, on transfert le contenu de %ID280 qui provient d'une périphérie (module As-i par exemple) dans ID72.



Lorsque le type de données à l'entrée IN a une longueur de bits supérieure à celle du type de données à la sortie OUT1, les bits de poids fort de la valeur source se perdent. Quand le type de données à l'entrée IN a une longueur de bits inférieure à celle du type de données à la sortie OUT1, les bits de poids fort de la valeur cible sont remplis de zéros.

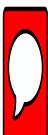
## 7. Exercice

L'exercice qui va suivre sera réalisé à l'aide du logiciel CIROS Mechatronics « Module de projet Ligne de convoyage 4 ».

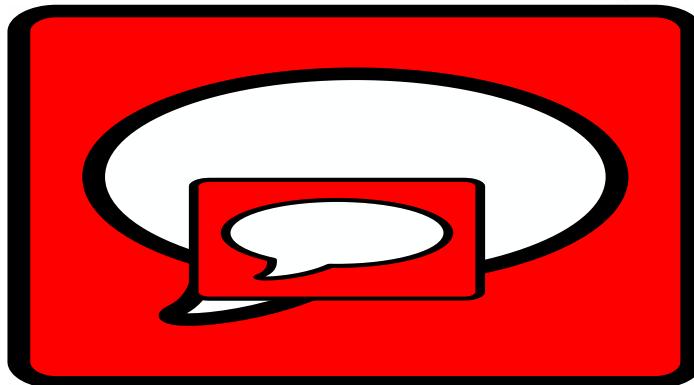
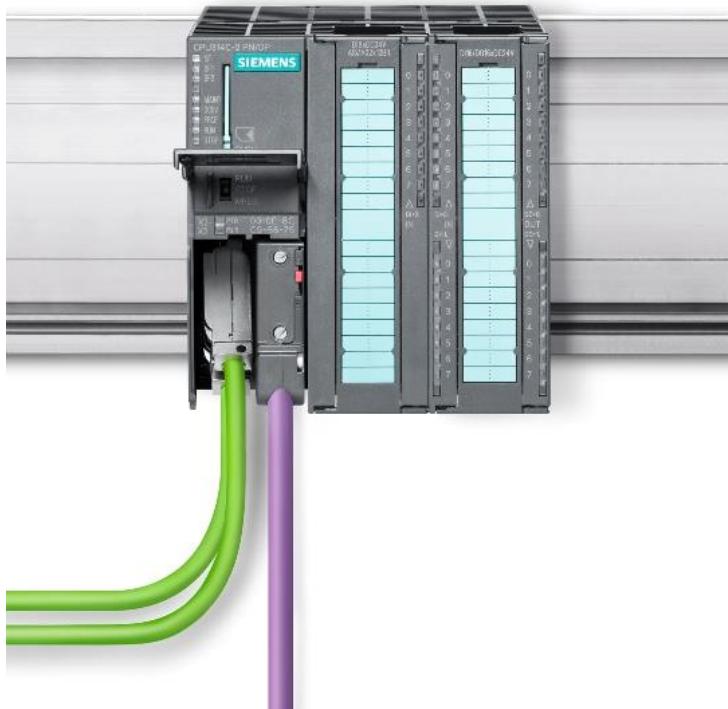
Réalisation d'un compteur par addition. Lorsqu'une pièce est présente à l'entrée du convoyeur et que l'on appuie sur le bouton Start, elle est acheminée vers la sortie. Les pièces métalliques sont comptées (comptage par addition). Lorsque 3 pièces métalliques ont été détectées, le voyant Q1 s'allume. Si le nombre de pièces métalliques dépasse 3, le voyant Q2 s'allume.

La remise à zéro du compteur se fait à l'aide du bouton Reset.

# Atelier C : TIA Portal



**Paramétrage de la connexion  
Ethernet de la CPU S7-300**



## 1. But de l'exercice

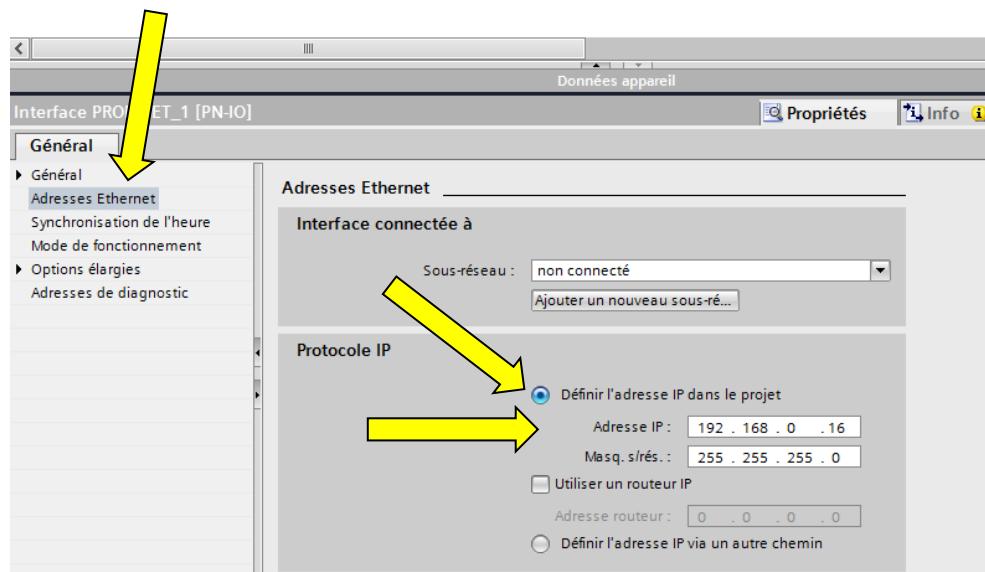
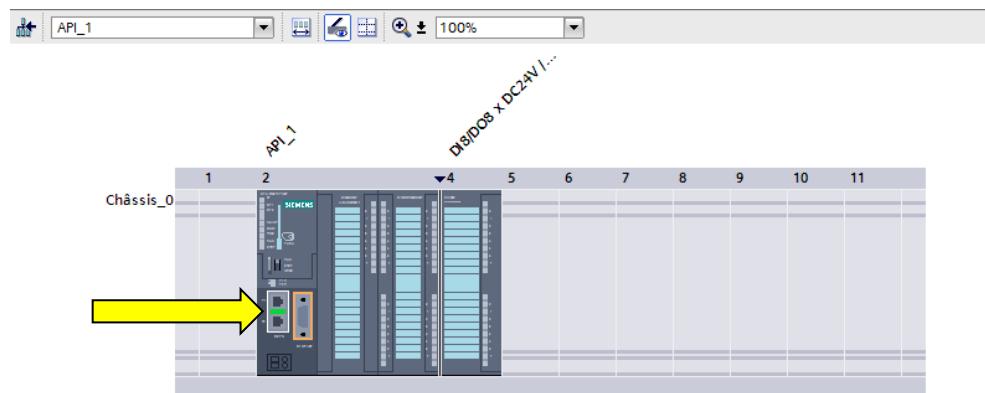
Le but de ce module est de paramétrer une CPU S7-314C2 PN/DP avec TIA Portal pour exploiter la connexion Ethernet de la CPU. La connexion Ethernet est plus rapide et permet de communiquer avec l'automate depuis n'importe quel emplacement disposant d'un accès au réseau, y compris sans fil, auquel est relié l'automate.

Les prérequis pour cette manipulation sont les suivants :

- Savoir établir une liaison avec l'automate en utilisant le module de communication Simatic S7 PC USB adapter.
- Disposer d'un projet dans TIA portal où l'automate est déjà configuré.

## 2. Travail à réaliser

- Ouvrez votre projet.
- En utilisant le volet Navigateur du projet, cliquez sur configuration des appareils pour accéder au menu de configuration de l'automate.
- Cliquez sur le module Ethernet pour avoir accès à la configuration de celui-ci.
- Cliquez sur l'onglet « Propriétés » puis sur « Adresses Ethernet ».



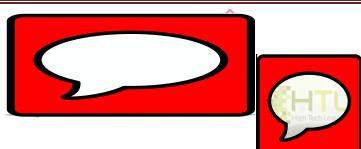
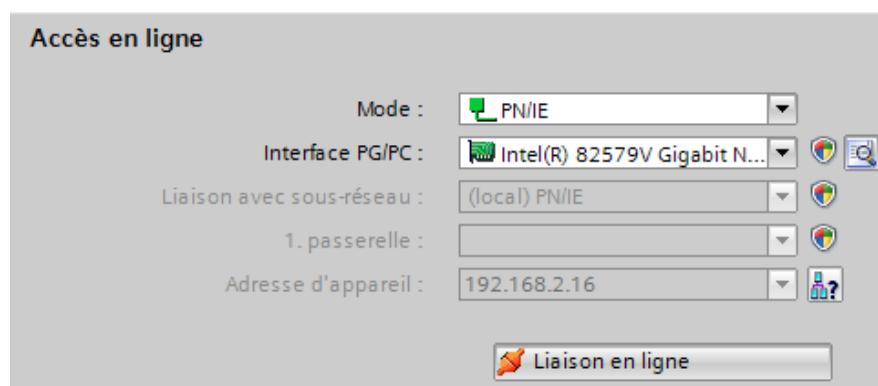
- Dans la partie « Protocol IP » indiquez l'adresse IP de l'automate.

192.168.2.\$\$

\$\$ = numéro de la CPU qui est indiqué sur celle-ci.



- Chargez la configuration dans l'automate en utilisant le module Simatic S7 PC USB adapter.
- Une fois que la nouvelle configuration est chargée dans l'automate, il faut interrompre la liaison en ligne et enlever la connexion USB.
- Il suffit maintenant de se reconnecter à l'automate en utilisant les paramètres suivants :
  - Mode : PN/IE
  - Interface PG/PC : Intel® 82579V Gigabit Network Connection



## Annexe 1

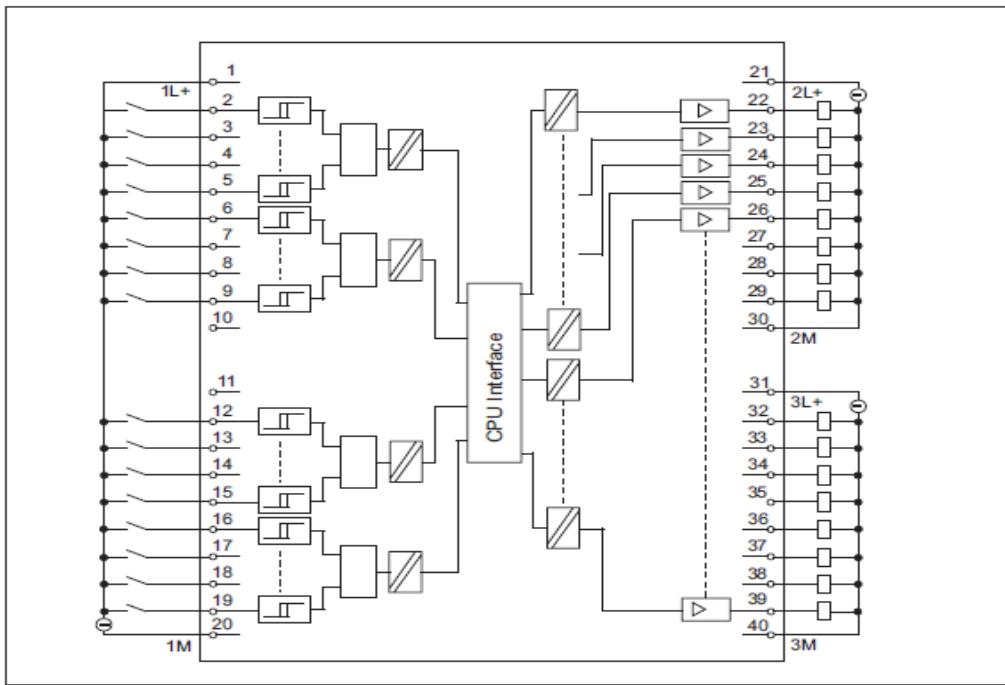


Figure 7-4 Block diagram of integrated digital I/O of CPUs 313C/313C-2/314C-2

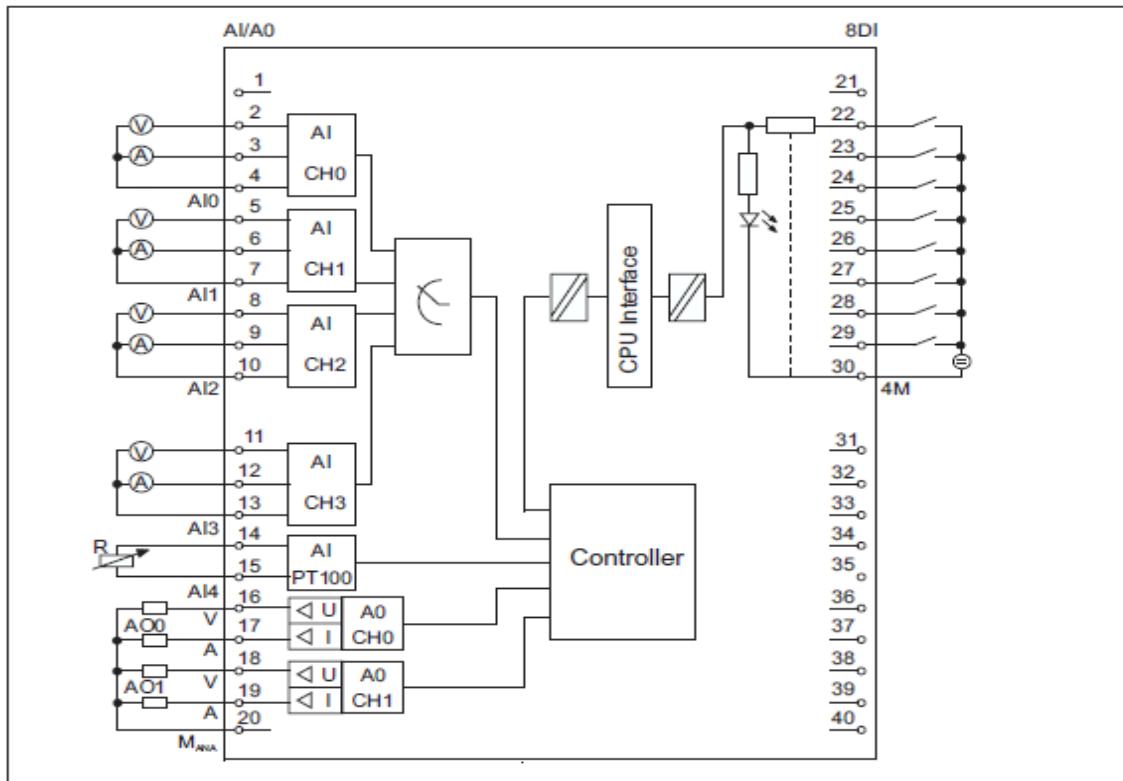
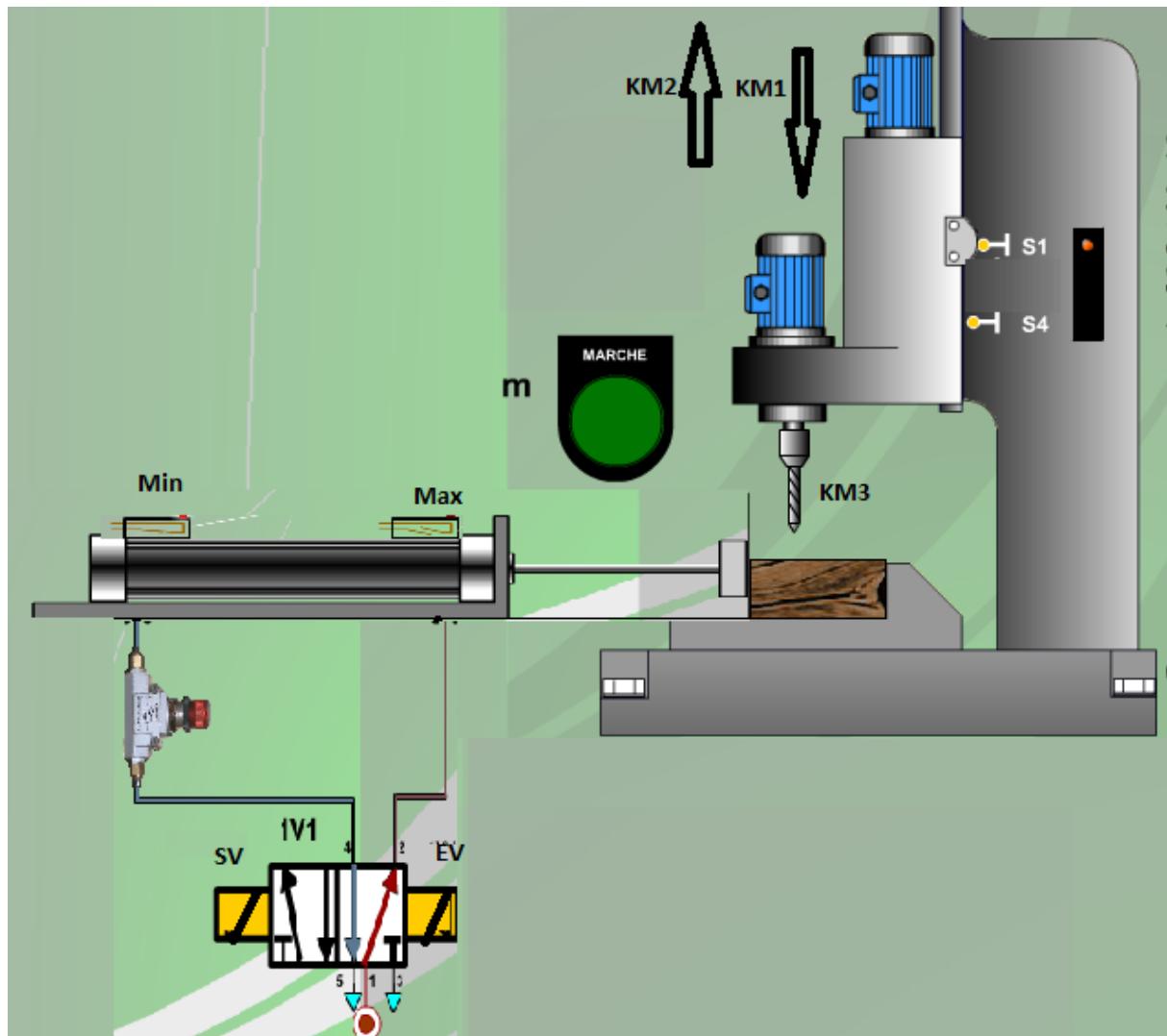
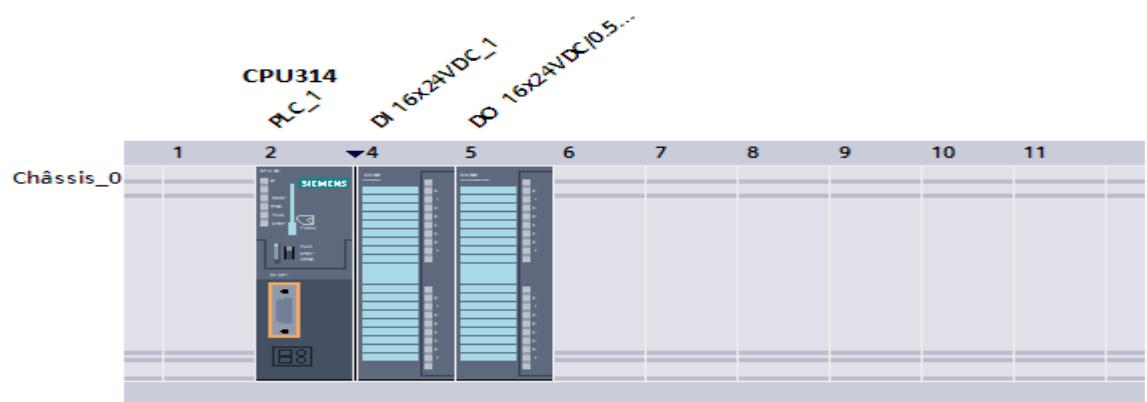


Figure 7-6 Block diagram of integrated digital/analog I/O of CPUs 313C/314C-2

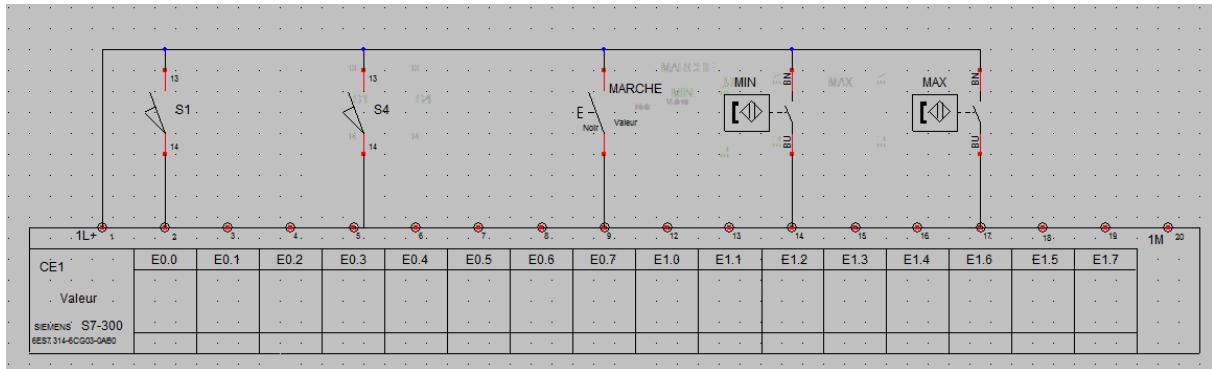
## Annexe 2



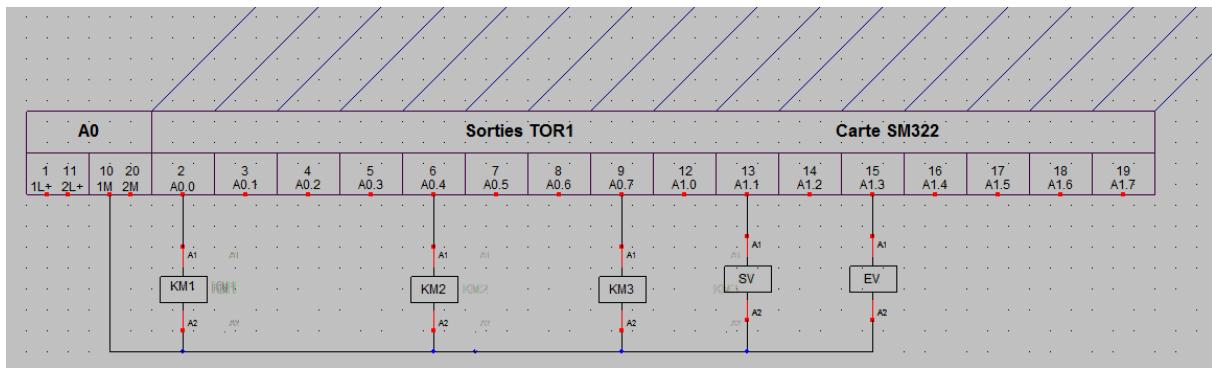
- Configuration matérielle



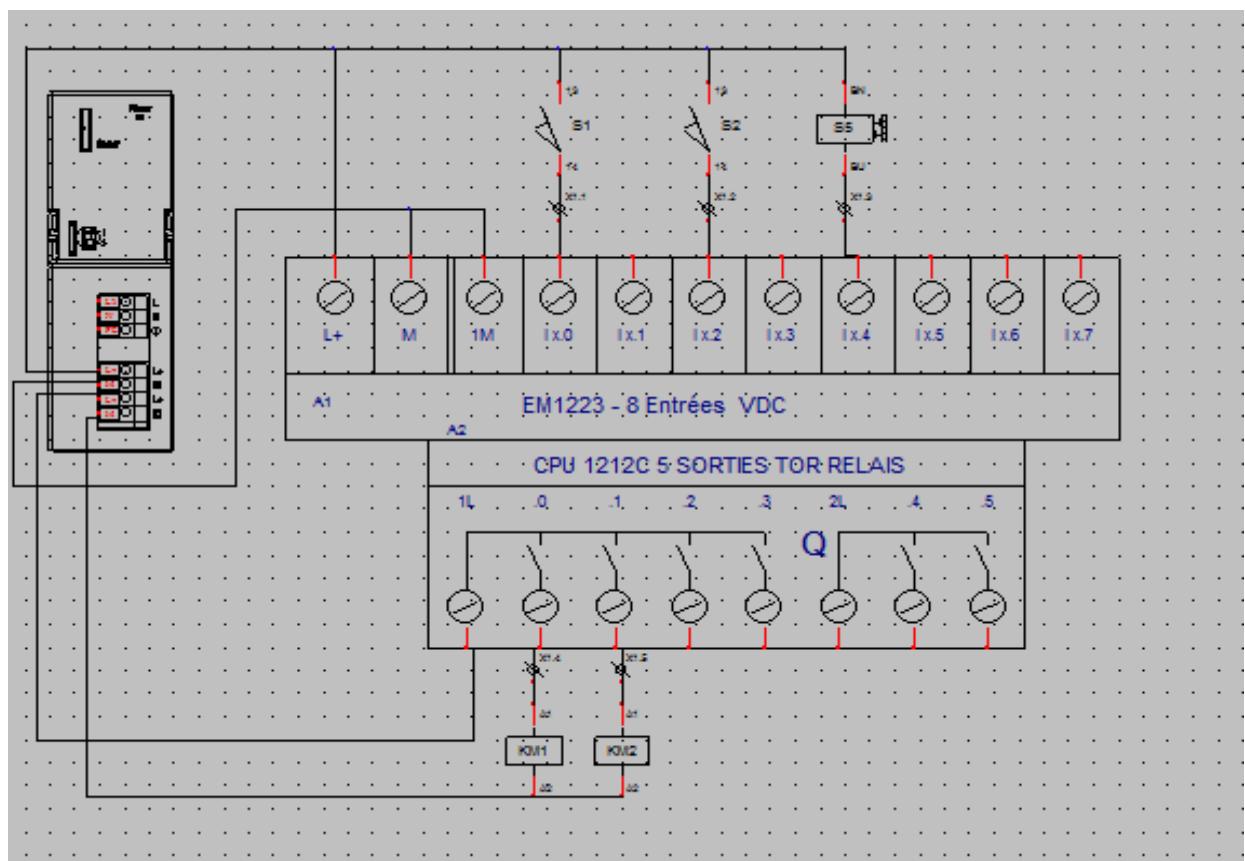
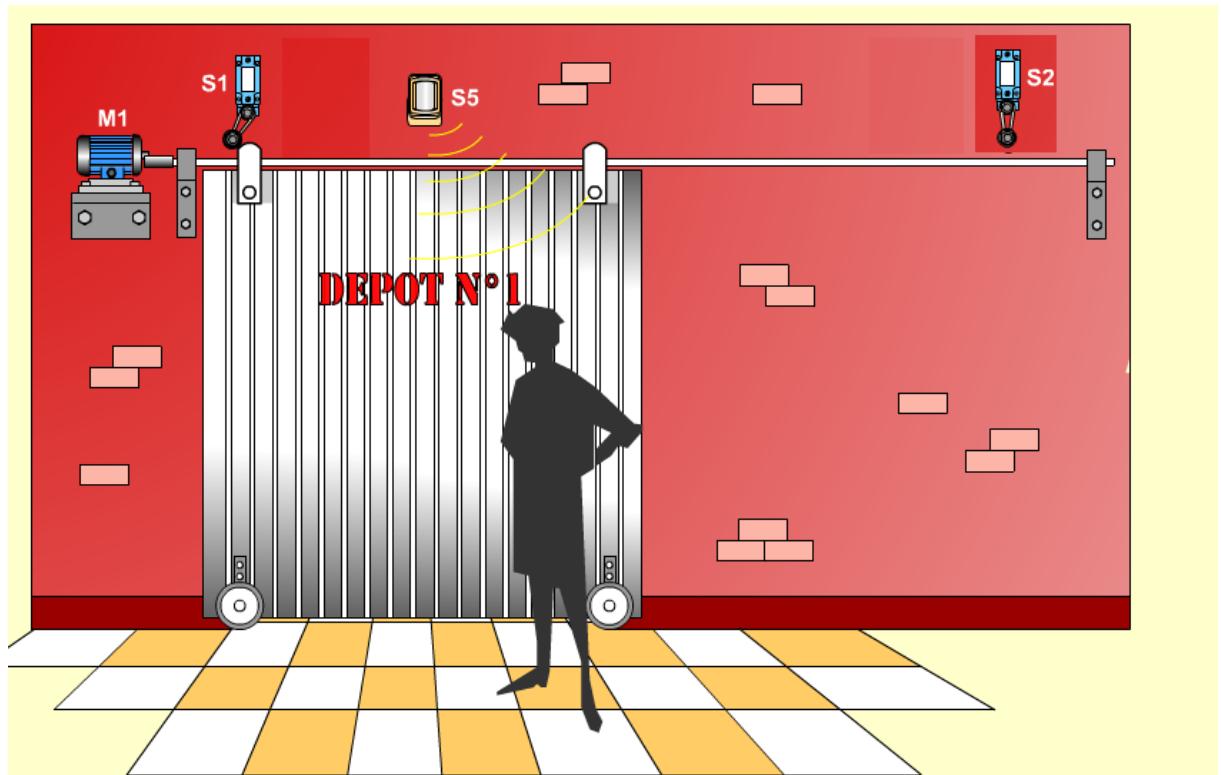
- Câblage des entrées



- Câblage des sorties



### Annexe 3



# Régulation en température d'un Four

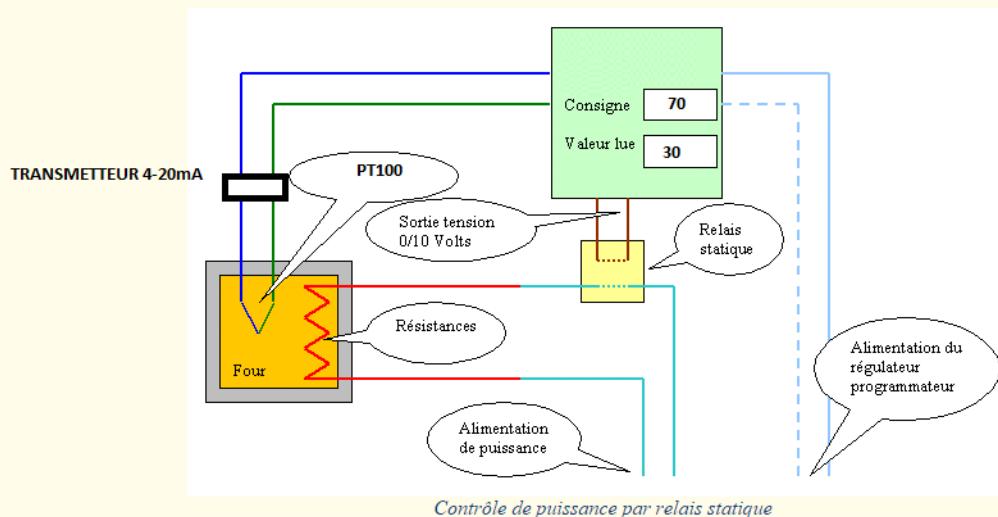
## 1. Presentation.

La partie opérative simule un four industriel destiné au séchage de pièces métalliques. La température sera comprise entre la température ambiante et 100°Celsius et sera asservie à une consigne prédefinie par un opérateur au moyen de deux interrupteurs ( 50°C, 70°C, 80°C ou 90°C).

Le contrôle du four est assuré par:

- Une résistance assurant le chauffage, commandé par relais statique
- un petit ventilateur destiné à refroidir le four en cas de dépassement de la température admissible (sécurité), commandé en TOR
- un grand ventilateur assurant le rôle de perturbation pour tester l'efficacité de la régulation en température, commandé en TOR

une sortie en tension continue 0 - 10 volts ou une sortie SSR (Solid State Relay) qui sert à commander un relais statique.



- Entrées / Sorties vue de l'API :

P_Ventilo	%Q0.0	Commande TOR (0-12V) du petit ventilateur
G_Ventilo	%Q0.1	Commande TOR (0-12V) du grand ventilateur
I1, I2	%I0.0 -%I0.1	Choix de la consigne de température
I4, I5	%I0.3 -%I0.4	Commandes manuelles des ventilateurs
Mes_Temp	%IW64	Entrée analogique (0-10V) de mesure de température. La sensibilité du capteur est de 100mV/°C (le capteur délivre 0V lorsque T=0°C)



1. Creez un projet Step7/TIA correctement configuré, et déclarez les entrées/sorties ci-dessus.

2. Calcul de la température dans l'enceinte du four.

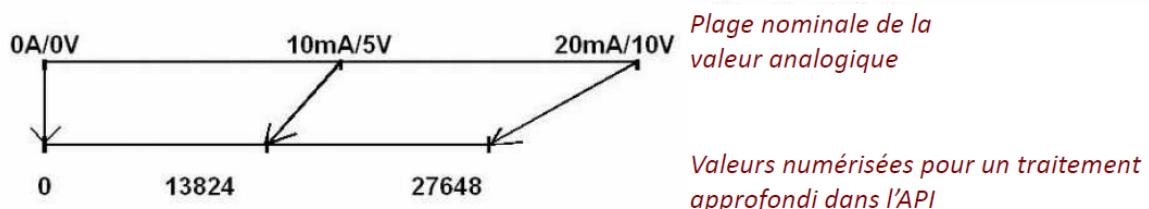
➤ Utilisation des grandeurs analogiques :

Les valeurs analogiques sont stockées dans des mots. La configuration utilisée dans ce TP permet d'accéder à 2 entrées analogiques (configurables en 0-20mA).

L'adresse de la première entrée analogique serait dans ce cas %IW 64, celle de la seconde entrée analogique %IW66, et celle de la sortie analogique %QW 80.

La transformation d'une valeur analogique en vue d'un traitement dans un API est la même pour les entrées et les sorties analogiques.

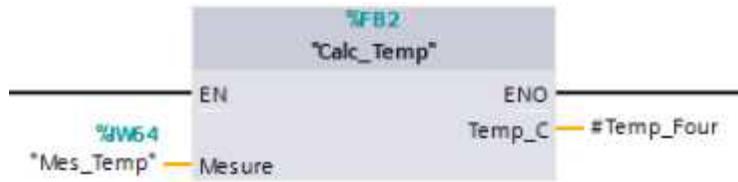
Les plages de valeurs numérisées sont les suivantes :



1. Donnez la règle de conversion  $T = f(N)$  ; où  $T$  est la température en °C et  $N$  le nombre présent sur l'entrée %IW64.

➤ Fonction "Calc\_Temp" :

L'objectif de ce programme est de calculer la température en °C, au format "Float", à partir de la valeur relevée sur l'entrée analogique.



### 3. Pilotage des Ventilateurs.

Les deux ventilateurs seront pilotés de la façon suivante :

- Le petit ventilateur est mis en service si la température dépasse 90°C, et désactivé lorsque celle-ci redescend en dessous de 87°C. Il peut également être enclenché manuellement avec l'interrupteur I4.
  - Le gros ventilateur est piloté manuellement avec l'interrupteur I5.
1. Créez un bloc fonctionnel "Cde\_Ventil" en langage CONT, définissez les entrées/sorties de cette fonction et programmez le fonctionnement décrit ci-dessus.

### 4. Fonction "Choix Consigne".

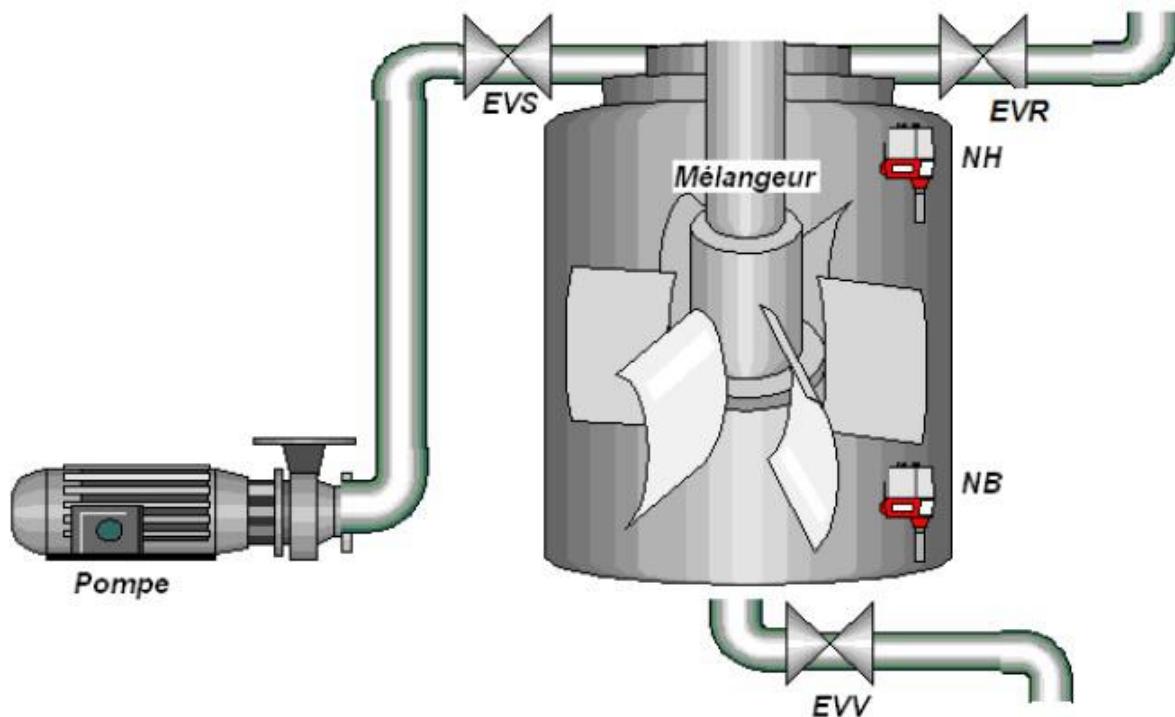
Nous allons créer une fonction FC2 nommée "Choix\_Cons" qui va définir la température à obtenir dans l'enceinte du four.

Le fonctionnement attendu est le suivant :

C2	C1	CONS
0	0	50.0
0	1	70.0
1	0	80.0
1	1	90.0

## - Mélangeur industriel -

On considère un mélangeur chargé du mélange d'un solvant acheminé par une pompe P1 (en série avec une électrovanne EVS), et d'un réactif, versé par une électrovanne EVR :



Le procédé obéit au fonctionnement suivant :

- Un appui sur le bouton poussoir Dcy lance un cycle de mélange;
- Le solvant est alors acheminé (ouverture de la vanne et mise en route de la pompe). La quantité de solvant versée est contrôlée grâce au temps de versement qui est de 12s.
- Le mélangeur, piloté par un variateur de vitesse, est mis en rotation à la vitesse de 350 tr/mn pendant durée du versement de solvant.
- Le réactif est ensuite versé dans la cuve de mélange au moyen pendant 8s;
- le mélangeur est alors à la vitesse de 945 tr/mn;
- Le produit final obtenu est vidangé de la cuve de mélange grâce à l'électrovanne de vidange, jusqu'à ce que la cuve soit vide.

Le tableau de la page suivante liste les entrées / sorties utilisées.

Description	Mnémonique	Adresse
Départ Cycle déclenché par l'opérateur	<b>Dcy</b>	%I0.0
Électrovanne de versement du solvant (monostable / NF)	<b>EVS</b>	%Q0.0
Électrovanne de versement du réactif (monostable / NF)	<b>EVR</b>	%Q0.1
Électrovanne de vidange de la cuve	<b>EVV</b>	%Q0.2
Sortie analogique 0-10V de commande de vitesse du variateur pilotant le moteur du mélangeur. (100mV/(tr/mn) )	<b>VIT</b>	%QW80
Commande du contacteur de mise sous tension de la pompe	<b>KMP</b>	%Q0.3
Détecteur de niveau bas dans la cuve (N.O)	<b>NB</b>	%I0.1

Afin d'obtenir un meilleur ré-amorçage de la pompe à chaque démarrage, on décide de retarder la mise hors tension de celle-ci 2,5s après la fermeture de la vanne EVS.

## **2.Utilisation des blocs de comptage.**

Afin de prévoir les opérations de maintenance préventive sur le système (nettoyage & changement des filtres, vérification des roulements etc...) on souhaite connaître le nombre de cycle réalisés depuis la dernière opération de maintenance.

Pour cela, on va activer un voyant d'alarme lorsque 10 cycles ont été réalisés. L'opérateur de maintenance devra alors intervenir, et acquittera de son intervention lorsqu'elle sera terminée.

Description	Mnémonique	Adresse
acquittement de l'opérateur	<b>AQ</b>	%I0.4
voyant d'alarme	<b>AL</b>	%Q0.4

## **3.Utilisation d'un memento de cadence.**

Nous allons compléter le fonctionnement précédent en modifiant le comportement du voyant d'alarme :

- Clignotement toutes les 2s si nombre de cycle >5
- Clignotement toutes les 1s si nombre de cycle >10
- Clignotement toutes les 0,5s si nombre de cycle >15

*12. Reportez-vous à l'aide en ligne pour définir un octet de "memento" sur l'API. Allez dans la configuration matérielle de la CPU → propriétés → General → Memento système & cadence puis cochez « activer l'utilisation de l'octet de memento de cadence ».*

*13. Quelles sont les adresses des bits à utiliser pour répondre au fonctionnement demandé.*

*14. Proposez un sous-programme « cligno ».*