

## Fiche de programmation TIA PORTAL SIEMENS

Cliquez sur Démarrer > Programmes > Siemens Automation > TIA-Portal V11 et passez au paragraphe 1 ou 2.

### 1.Ouverture d'un projet existant :

Sélectionnez votre projet dans son répertoire et cliquez sur « Ouvrir ». Passez au paragraphe 3.

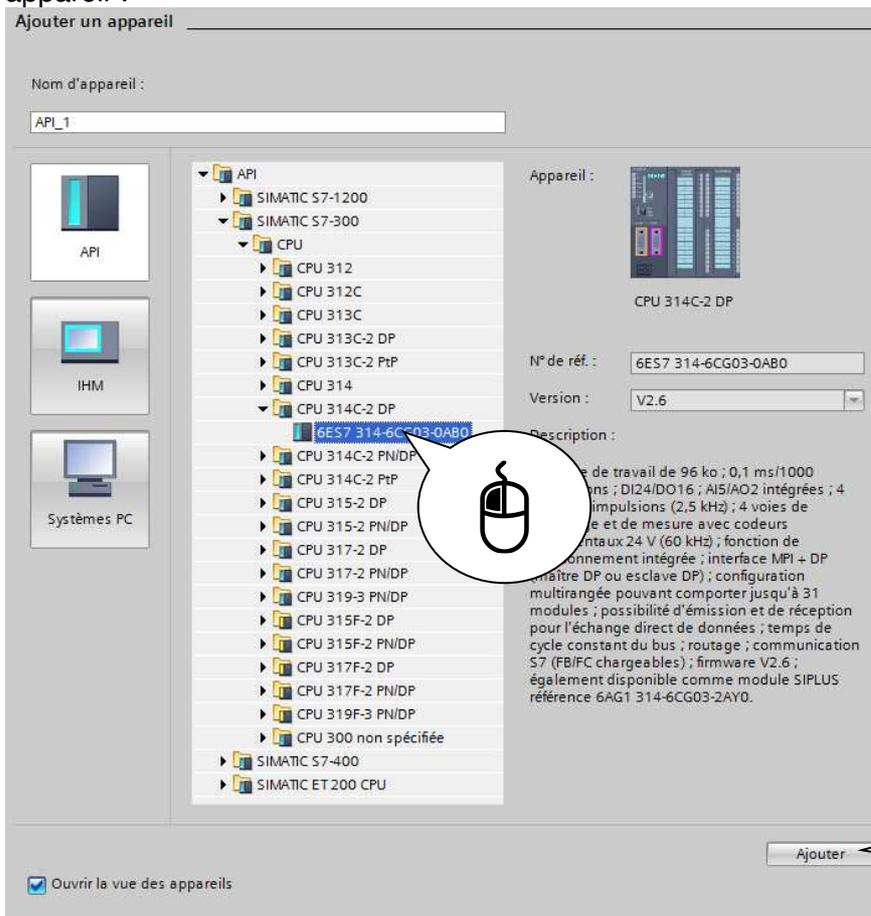
### 2.Création d'un projet :

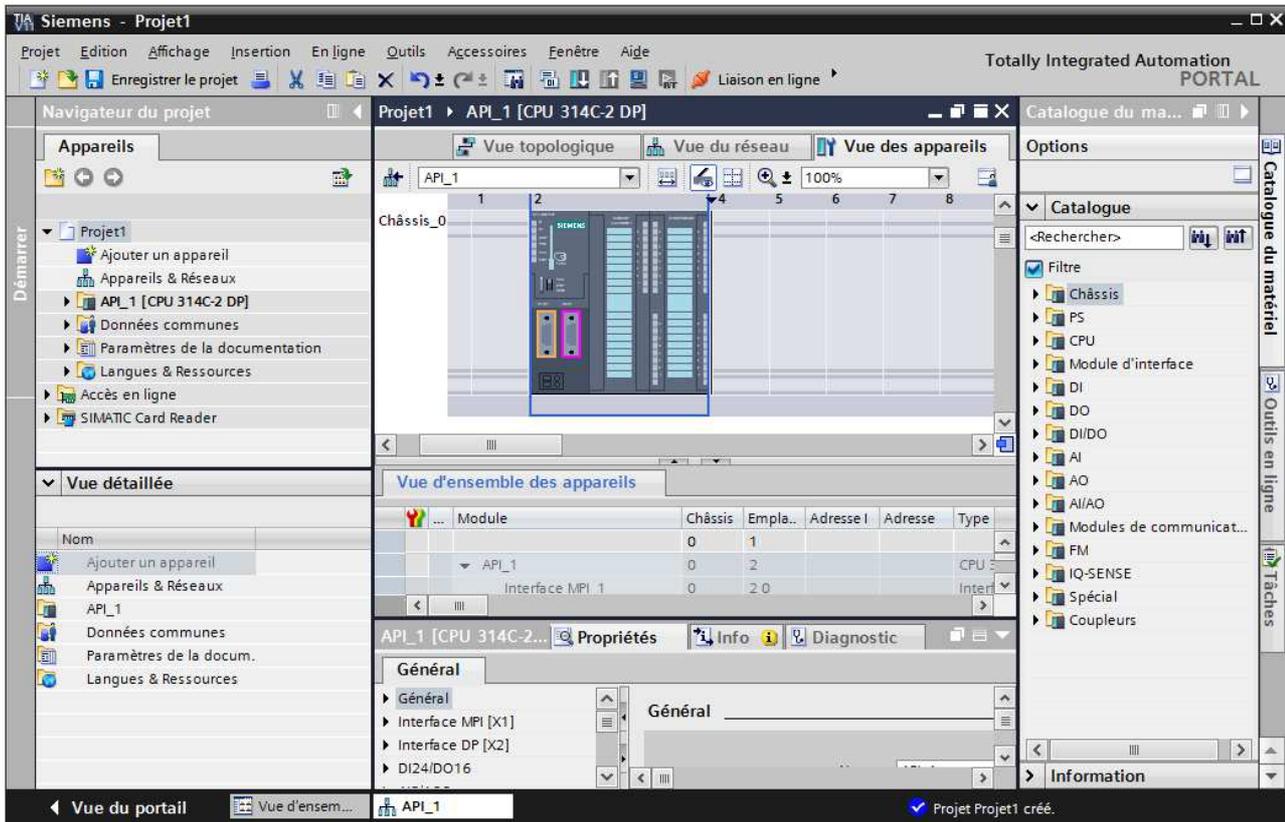


Cliquez sur « Créer un projet », entrez le nom du projet et cliquez sur « créer ».

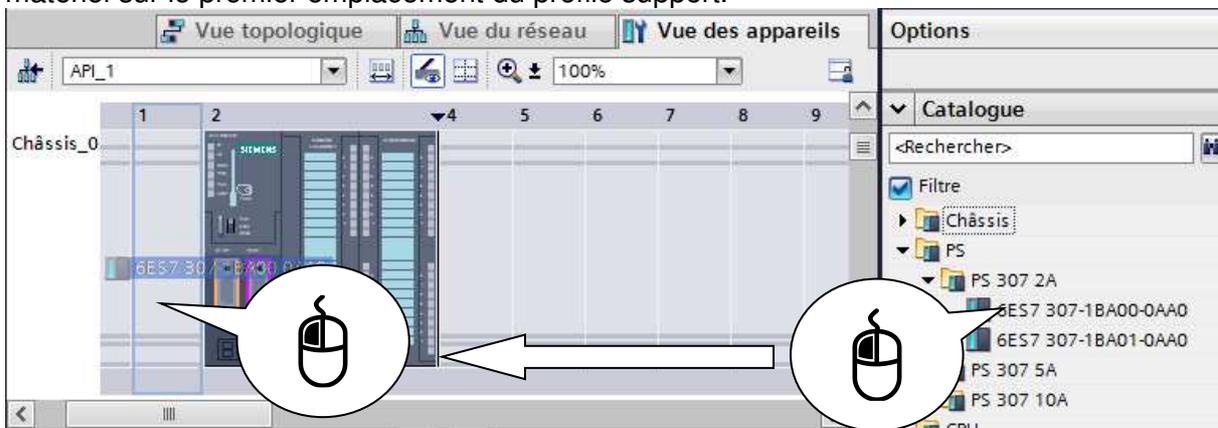
### 3.Configuration de l'automate :

Pour insérer la CPU, cliquez sur "Configurer un appareil" et cliquez sur la commande "Ajouter un appareil".

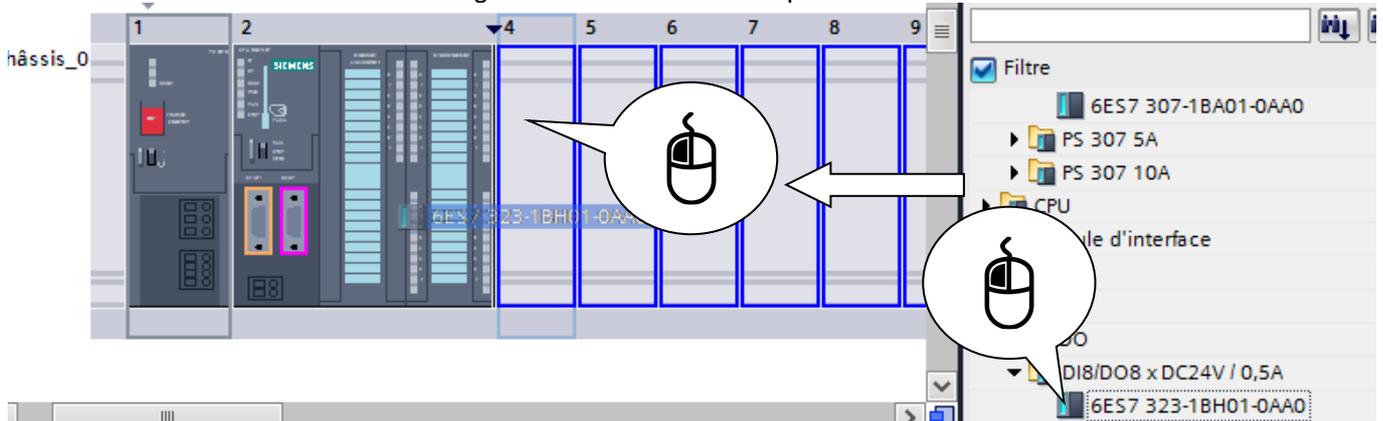




Faites glisser l'alimentation "PS 307 2A" avec le n° "6ES7 307-1BA00-0AA0" du catalogue du matériel sur le premier emplacement du profilé support.



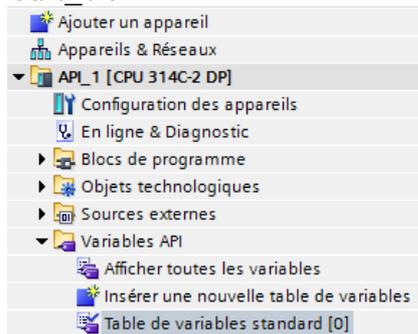
Enfichez le module d'entrée/de sortie numérique "DI8/DO8 x DC24V / 0,5A" avec le n° "6ES7 323-1BH01-0AA0" du catalogue du matériel sur l'emplacement 4.



Enregistrer le projet.

#### 4. Ecriture des mnémoniques :

Dans la navigation du projet, ouvrez le dossier "Variables API" qui se trouve sous la CPU "API\_1".



Double cliquez sur la table « Table des variables standard », vous pouvez entrer des mnémoniques maintenant ou en cours de programmation.

Table de variables standard			
	Nom	Type de données	Adresse
1	m	Bool	%I124.0
2	e1s0	Bool	%I124.1
3	e1s1	Bool	%I124.2
4	e2s0	Bool	%I124.3
5	e2s1	Bool	%I124.4
6	etape0	Bool	%M0.0
7	etape1	Bool	%M0.1
8	etape2	Bool	%M0.2
9	etape3	Bool	%M0.3
10	etape4	Bool	%M0.4
11	s1yv1_12	Bool	%Q124.0
12	s1yv1_14	Bool	%Q124.1
13	s2yv1_12	Bool	%Q124.2
14	s2yv1_14	Bool	%Q124.3

Cliquez sur Enregistrer et fermez l'éditeur de mnémoniques.

#### 5. Ecriture du programme :

On écrit le programme en utilisant différents blocs :

**FC10** : étapes,

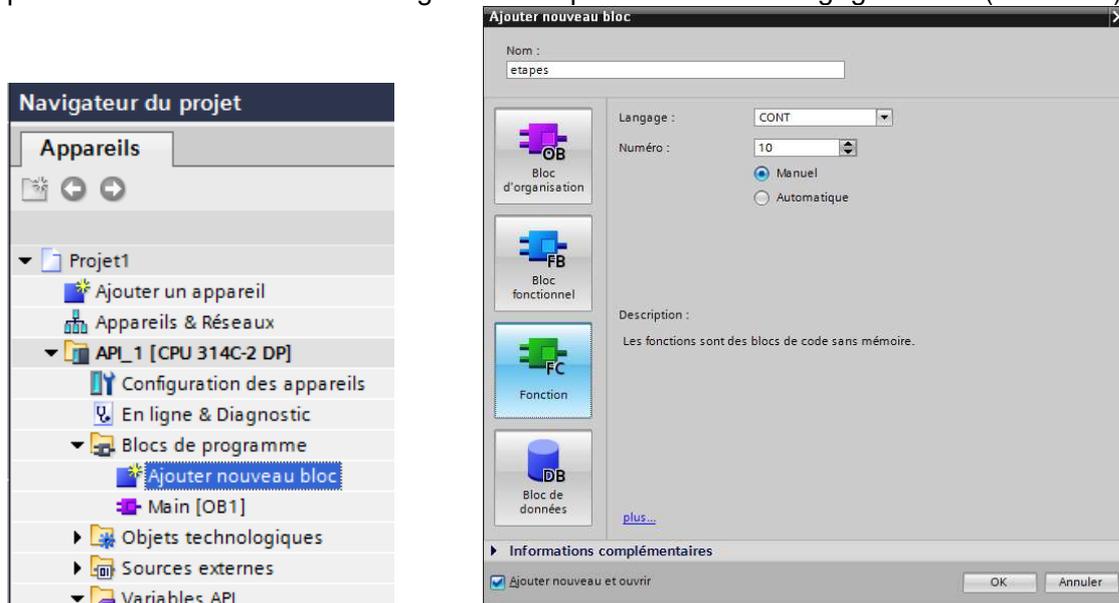
**FC30** : actions internes,

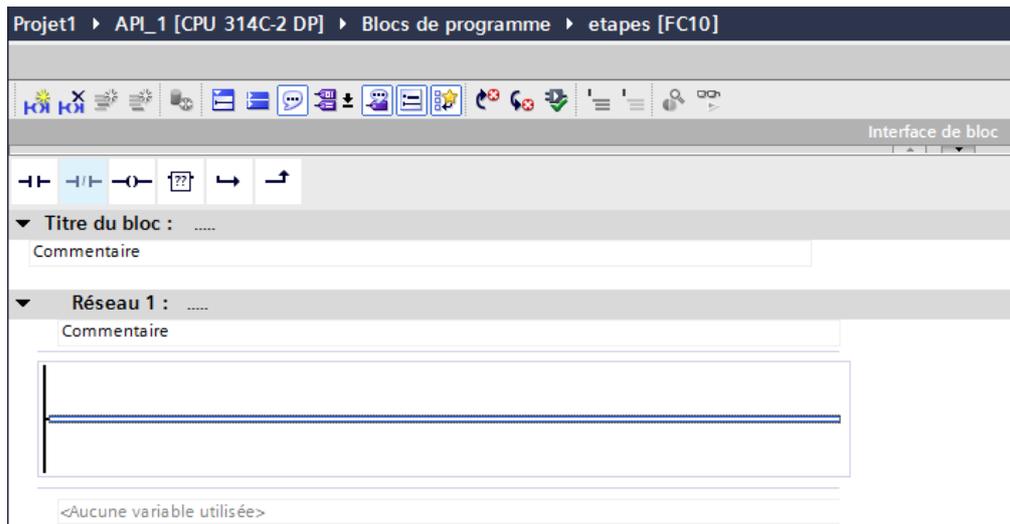
**FC40** : actions externes,

**OB1** : appel des fonctions.

**OB100** : initialisation.

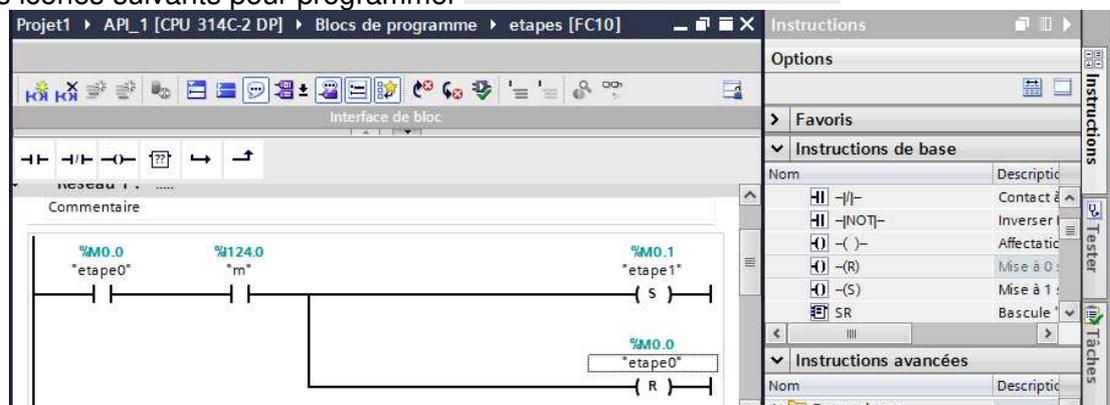
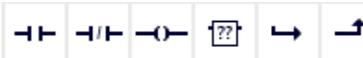
Ouvrez « Blocs de programme » et double cliquez sur "Ajouter nouveau bloc », puis "Fonction" pour FC10 à FC40 et "Bloc d'organisation" pour OB100 en langage CONT (LADDER).





**Attention : un réseau et un seul par sortie.**

Utilisez les icônes suivants pour programmer



Exemple :

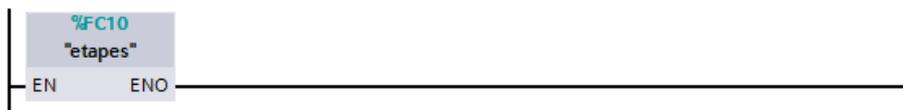
Cliquez sur « nouveau réseau »  pour passer au réseau suivant.

Une fois le programme entré, cliquez sur Enregistrer et fermer l'éditeur CONT.

Recommencez pour FC30, FC 40, OB1 et OB100.

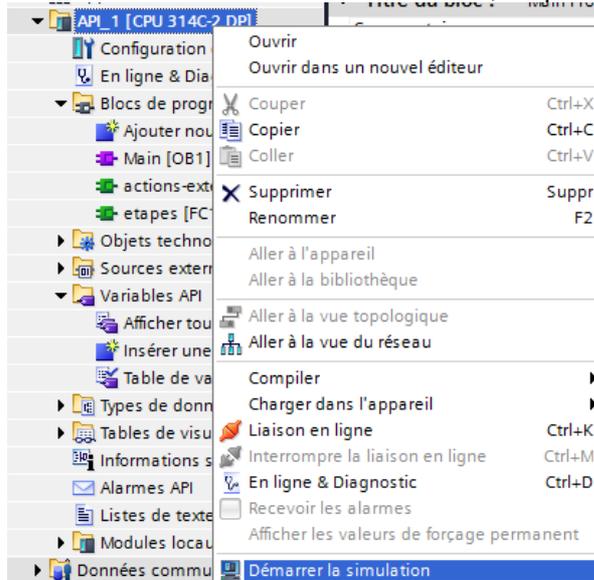
Le programme de OB1 sera un appel aux différentes fonctions (FC10, FC30 et FC40).

Exemple de l'appel de FC10 dans OB1 :

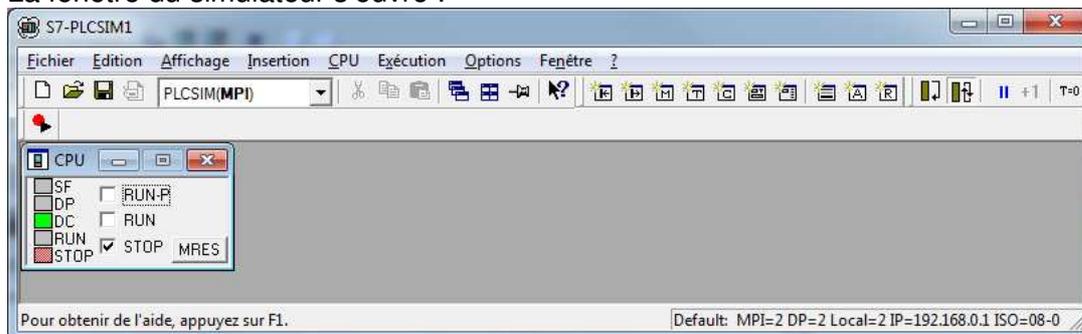


## 6. Test du programme avec l'automate de simulation :

Pour utiliser l'automate de simulation, cliquez sur « Démarrer la simulation »



La fenêtre du simulateur s'ouvre :

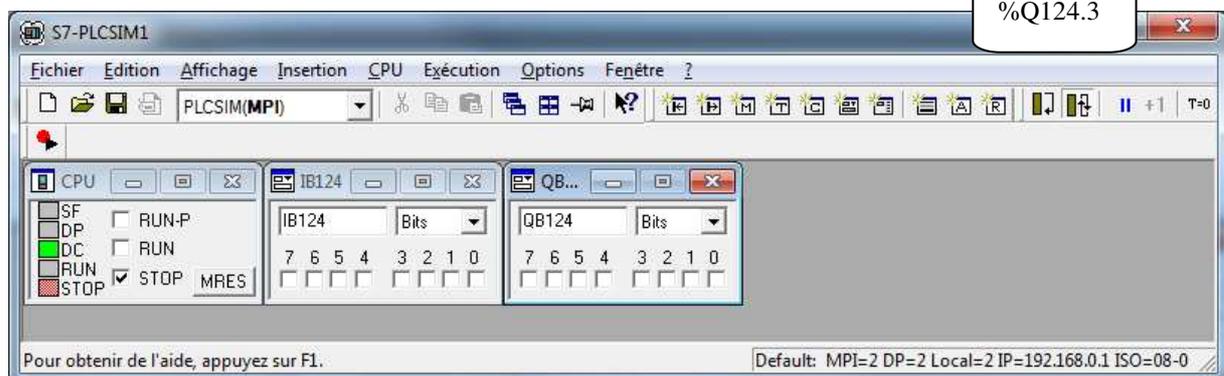
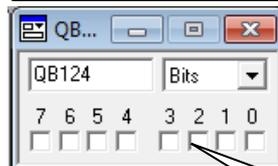


Vous devez configurer votre API avec les cartes et éventuellement des zones mémoires.

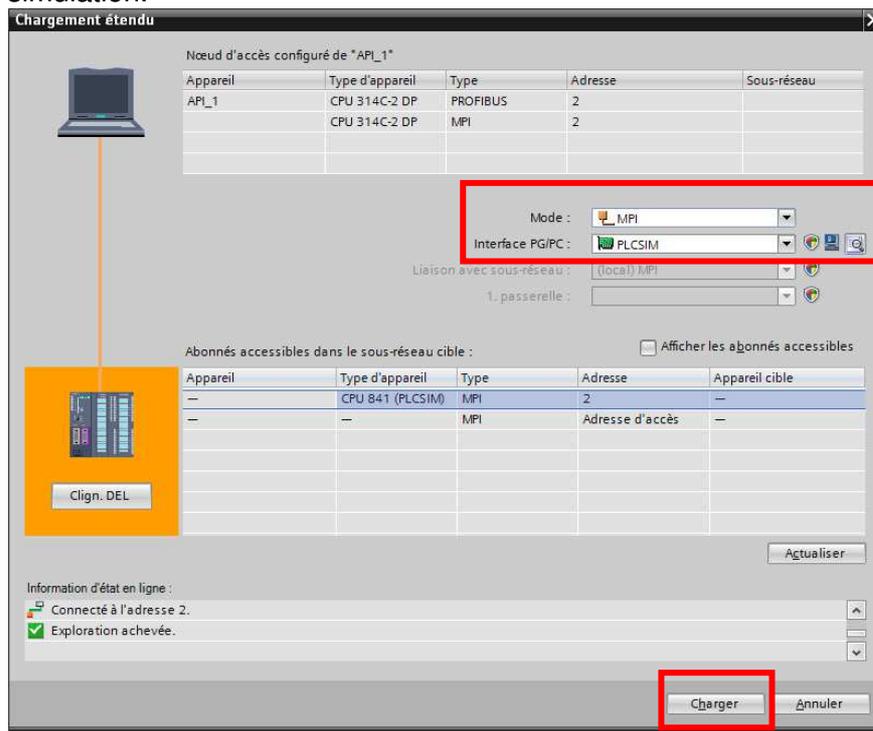
Pour ajouter une carte d'entrées, cliquez sur l'icône .



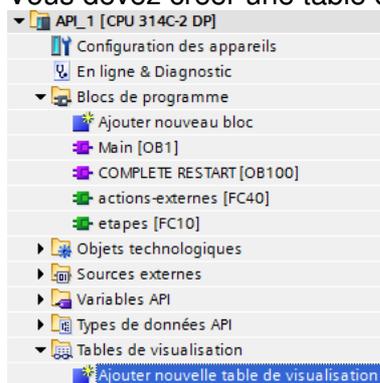
Changez l'adresse pour faire correspondre à votre projet.



Cliquez sur Charger, puis sur Charger pour transférer votre programme dans l'automate de simulation.



Vous devez créer une table d'animation :

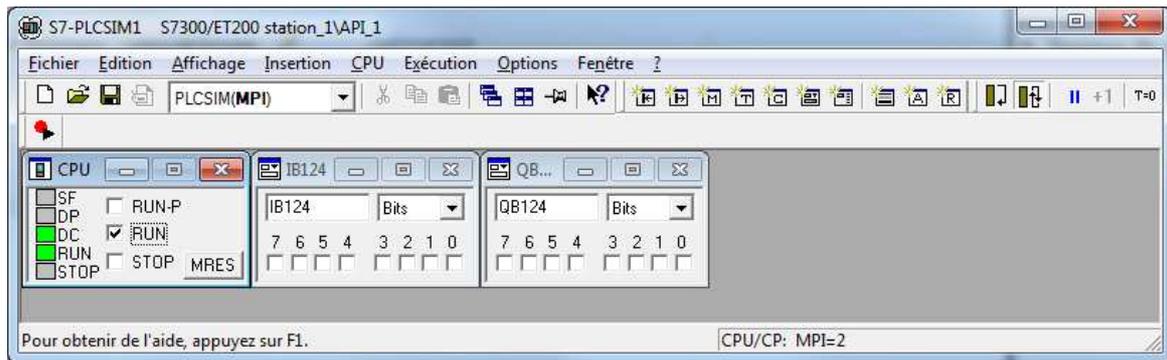


Projet1 ▶ API\_1 [CPU 314C-2 DP] ▶ Tables de visualisation ▶ Table de visualisation\_1

	i	Nom	Adresse	Format d'affichage	Valeur visualisatio	Valeur de forçage	⚡
1		"etape0"	%M0.0	Bool			<input type="checkbox"/>
2		"etape1"	%M0.1	Bool			<input type="checkbox"/>
3		"etape2"	%M0.2	Bool			<input type="checkbox"/>
4		"etape3"	%M0.3	Bool			<input type="checkbox"/>
5		"etape4"	%M0.4	Bool			<input type="checkbox"/>

Insérer les variables à visualiser (toutes les étapes). Cliquez sur  pour voir l'état de ces variables en lignes.

Placez l'automate en RUN.



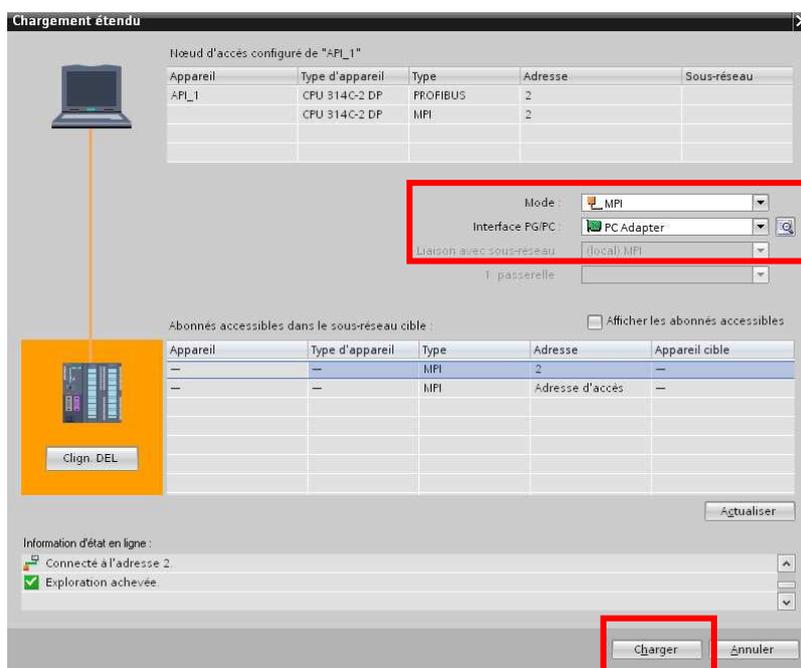
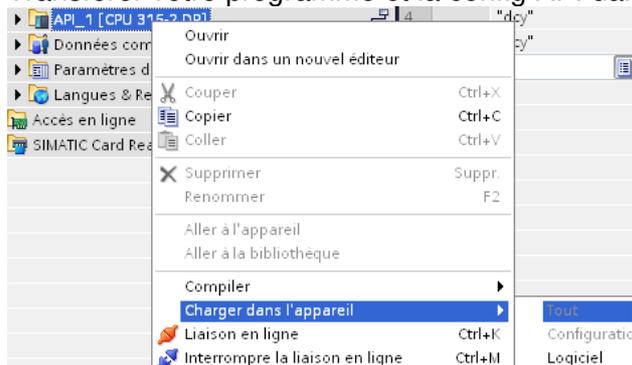
Projet1 ▶ API\_1 [CPU 314C-2 DP] ▶ Tables de visualisation ▶ Table de visualisation\_1

	i	Nom	Adresse	Format d'affichage	Valeur visualisatio	Valeur de forçage
1		"etape0"	%M0.0	Bool	<input checked="" type="checkbox"/> TRUE	<input type="checkbox"/>
2		"etape1"	%M0.1	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>
3		"etape2"	%M0.2	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>
4		"etape3"	%M0.3	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>
5		"etape4"	%M0.4	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>

Testez votre programmation en modifiant les valeurs des variables d'entrées. Corrigez là, si besoin.

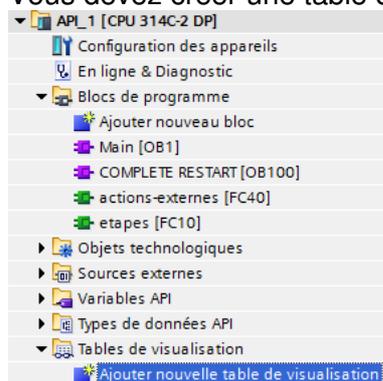
## 7. Test réel

Transférer votre programme et la config API dans la mémoire de l'API.



Cliquez sur Charger, puis sur Charger pour transférer votre programme dans l'automate de simulation.

Vous devez créer une table d'animation :



Projet1 ▶ API\_1 [CPU 314C-2 DP] ▶ Tables de visualisation ▶ Table de visualisation\_1

	i	Nom	Adresse	Format d'affichage	Valeur visualisatio	Valeur de forçage	⚡
1		"etape0"	%M0.0	Bool			<input type="checkbox"/>
2		"etape1"	%M0.1	Bool			<input type="checkbox"/>
3		"etape2"	%M0.2	Bool			<input type="checkbox"/>
4		"etape3"	%M0.3	Bool			<input type="checkbox"/>
5		"etape4"	%M0.4	Bool			<input type="checkbox"/>

Insérer les variables à visualiser (toutes les étapes). Cliquez sur  pour voir l'état de ces variables en lignes.

Placez l'automate en RUN, avec le bouton situé sur l'unité centrale.

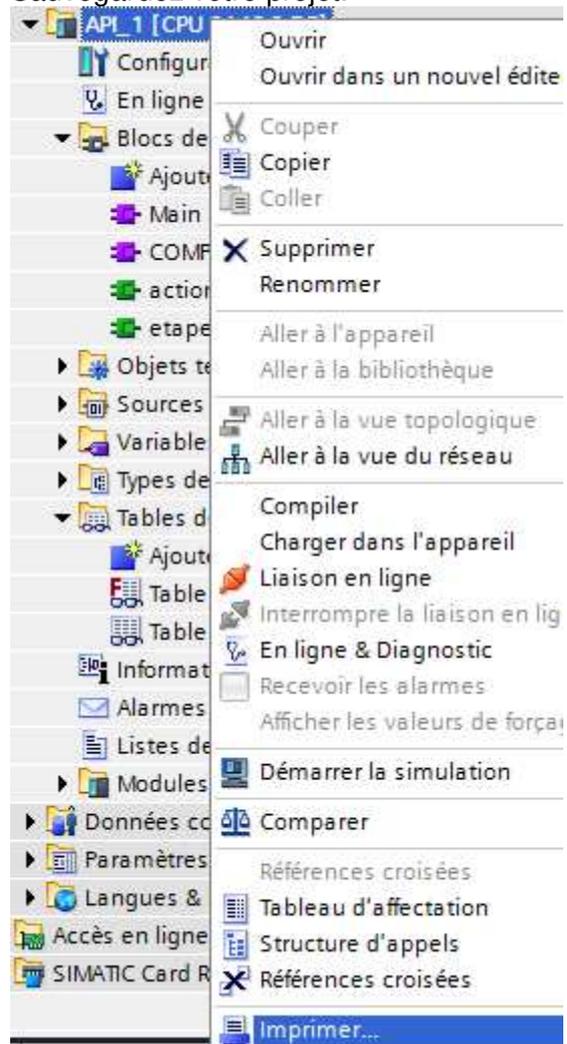
Projet1 ▶ API\_1 [CPU 314C-2 DP] ▶ Tables de visualisation ▶ Table de visualisation\_1

	i	Nom	Adresse	Format d'affichage	Valeur visualisatio	Valeur de forçage	⚡
1		"etape0"	%M0.0	Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
2		"etape1"	%M0.1	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
3		"etape2"	%M0.2	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
4		"etape3"	%M0.3	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
5		"etape4"	%M0.4	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>

Testez votre programmation.

## 8. Impression

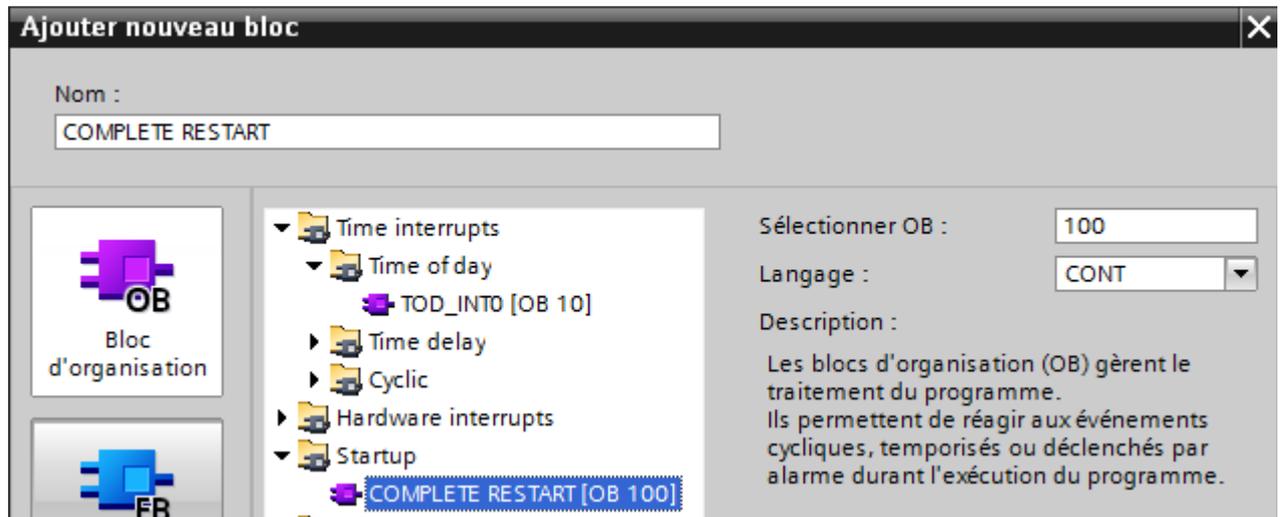
Sauvegardez votre projet.



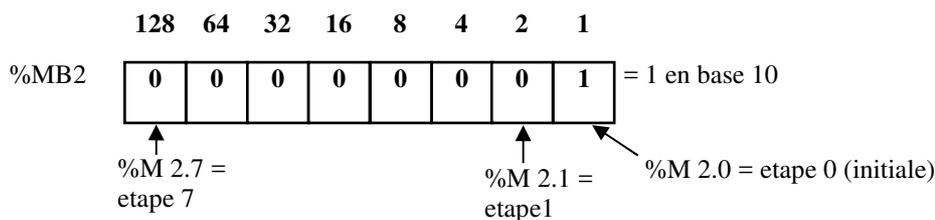
**Annexes :**

Programmation de OB100 :

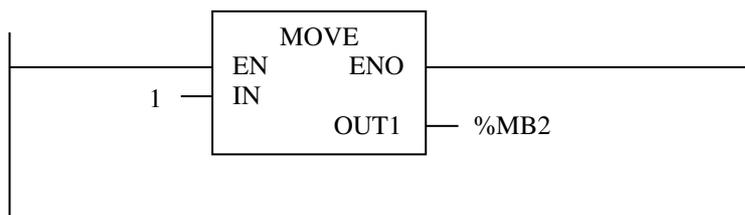
OB100 : OB de démarrage à chaud. Il ne sera exécuté qu'une seule fois à la mise en RUN de l'API. On l'utilise pour initialiser le ou les grafjets.



Exemple : Soit un grafjet comportant 8 étapes de 0 à 7 avec l'étape 0 comme étape initiale, ce grafjet est mémorisé dans l'octet 2 (%MB2).



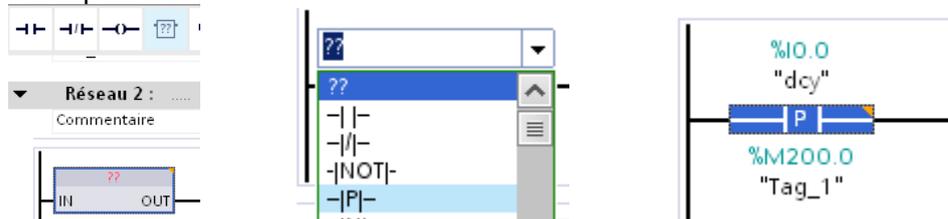
Initialiser le grafjet consiste à donner la valeur décimale « 1 » à l'octet %MB2 à la mise en RUN de l'automate.  
Programme de OB100 :



Utilisation des fronts montants :

Il faut utiliser le bloc POS, ce bloc utilise un memento de front et l'entrée sur laquelle on veut détecter le front.

Exemple : Front montant de DCY



%M200.0 : memento de front (on peut choisir n'importe quel memento).

Le memento de cadence (clignotement) :

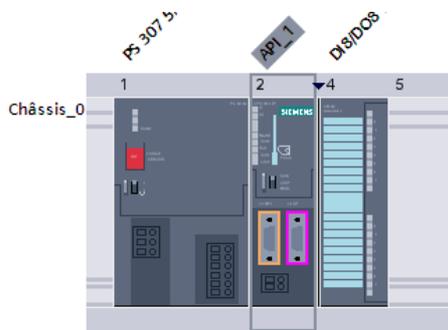
Le memento de cadence est un octet. Chacun des bits de cet octet change d'état suivant une horloge interne.

Une durée de période et la fréquence correspondante sont affectées à chaque bit de l'octet de memento de cadence :

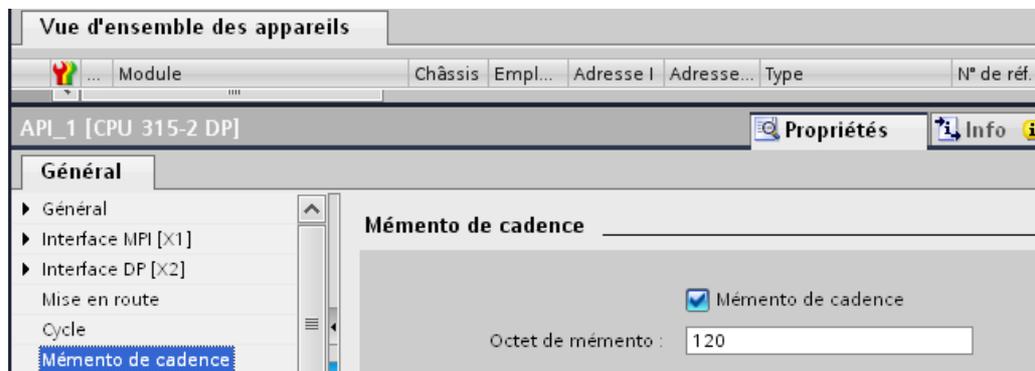
Bit	7	6	5	4	3	2	1	0
Durée de période (s)	2	1,6	1	0,8	0,5	0,4	0,2	0,1
Fréquence (Hz) :	0,5	0,625	1	1,25	2	2,5	5	10

Exemple : On veut faire clignoter un voyant à la fréquence de 1Hz.

On choisit un octet de cadence en double cliquant sur Configuration des Appareils.

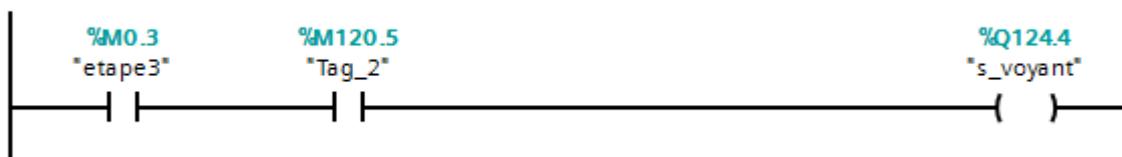


Sélectionnez l'unité centrale et en bas, allez dans l'onglet « Propriétés »



Cochez « Memento de cadence » et choisissez un octet (120 par exemple).

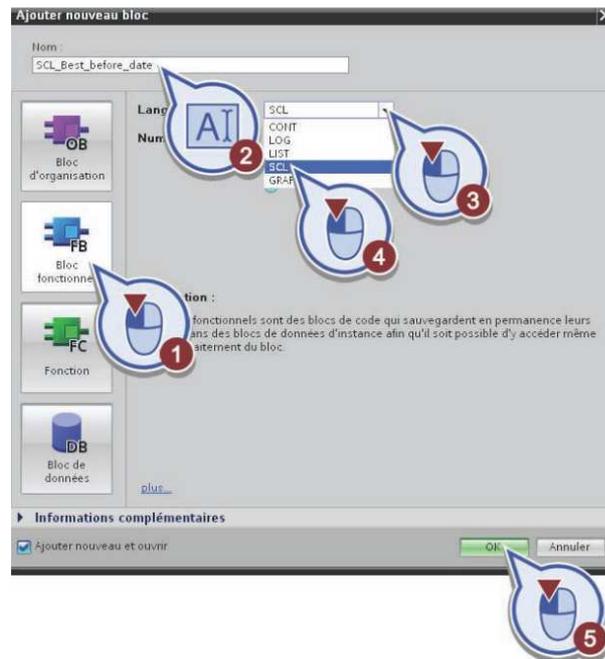
Le bit 5 de %MB120 change d'état toutes les secondes



Programmation d'un bloc fonctionnel en langage structuré (SCL) :

Créez un projet classique.

Dans « Ajouter nouveau bloc », insérez un bloc fonctionnel en langage SCL



Exemple pour le bloc fonctionnel FB1 :

	Nom	Type de données
1	Input	
2	manu	Bool
3	bp_plus	Bool
4	bp_moins	Bool
5	<ajouter nouvelle>	
6	Output	
7	cod_manu	Int

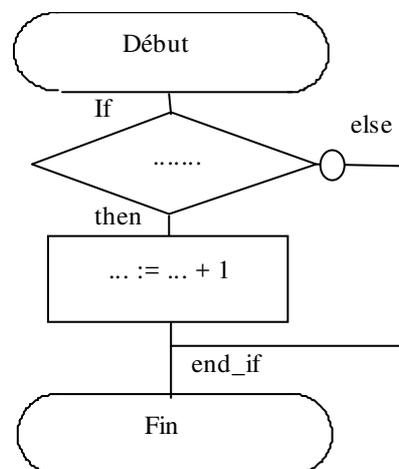
  

IF...	CASE... OF...	FOR...TO DO...	WHILE... DO...
1	2	3	4

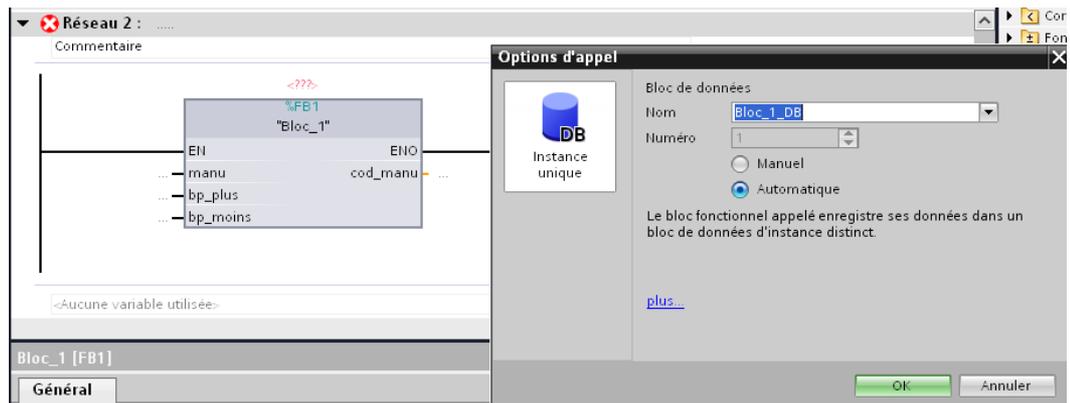
```

1
2 IF #manu AND #bp_plus THEN
3   #cod_manu:=#cod_manu+1;
4 END_IF;

```



Enregistrez et compilez (clic droit, compiler). Le bloc FB1 est généré dans les blocs et peut être utilisé dans OB1.



Il faut lui adjoindre un bloc de données (DB1 par exemple).