

## Réalisation d'un bargraph par commande HMI en semi-automatique

### Démarrage d'un nouveau projet ...

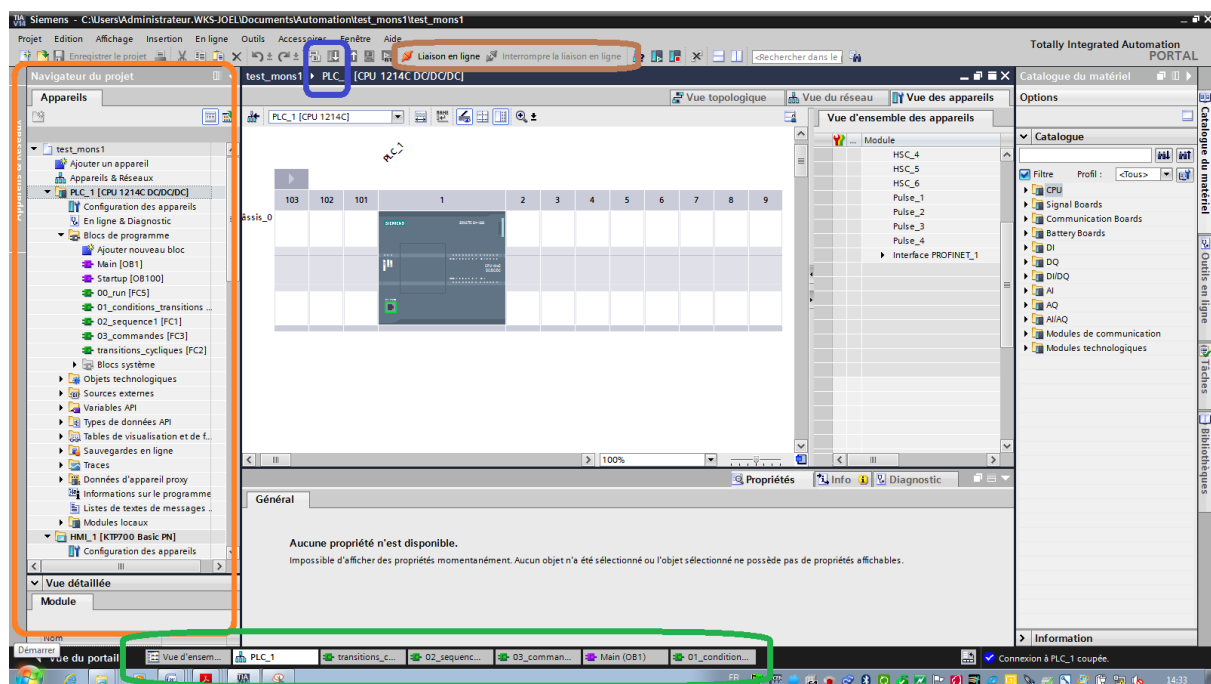
### Choix du matériel

Lorsqu'on démarre un projet (sur TIA portal V14), il faudra préciser la référence de l'automate. Deux possibilités :

- On précise la référence exacte (1214C DC/DC/DC ... )
- On choisit une référence indéterminée (XXX) et on lance une recherche pour avoir la référence exacte.

On verra aussi que, si l'HMI est interconnecté, il apparait l'HMI dans le résultat de recherche.

### Mini description



En orange à gauche, l'arborescence du projet. On y trouve les infos du PLC, les programmes, les infos HMI, ...

En bleu, la compilation et le chargement.

Remarque : parfois il faut sélectionner le PLC de l'arborescence pour le chargement total.

En brun, la connexion / déconnexion du PC à l'API.

En vert, les onglets des éléments ouverts.



## Réalisation d'un bargraph par commande HMI en semi-automatique

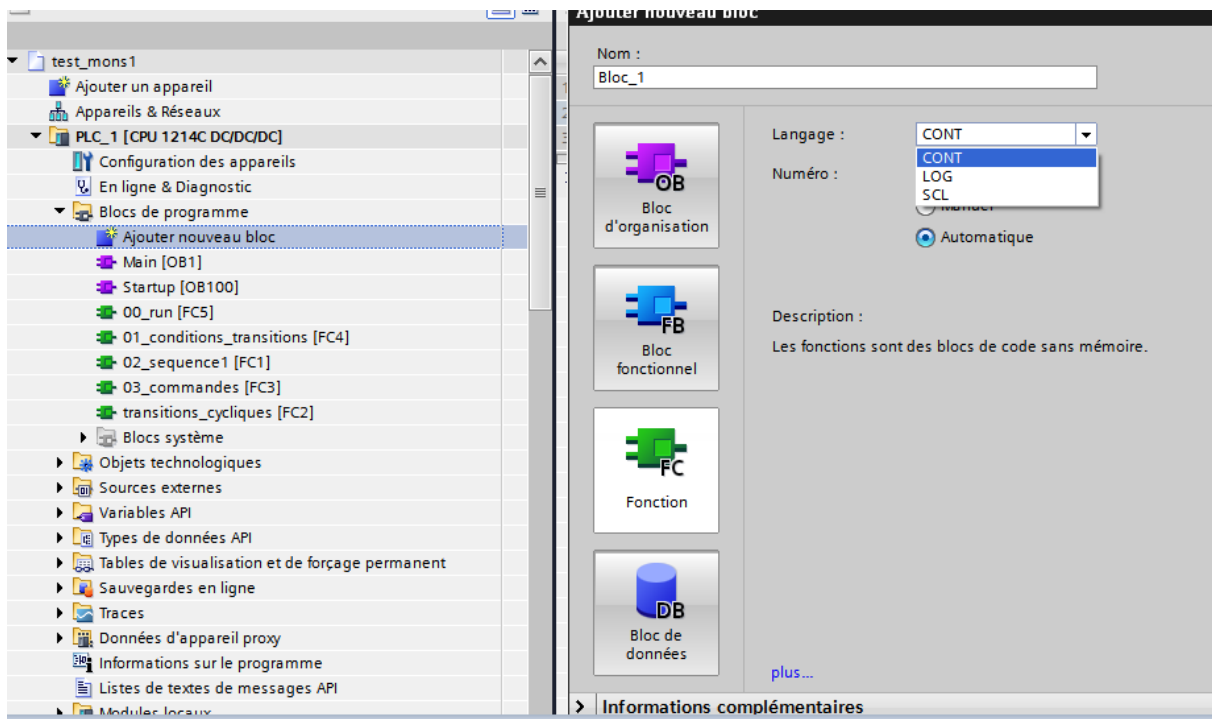
### Programme de test

Le programme de base appelé s'appelle par défaut OB1.

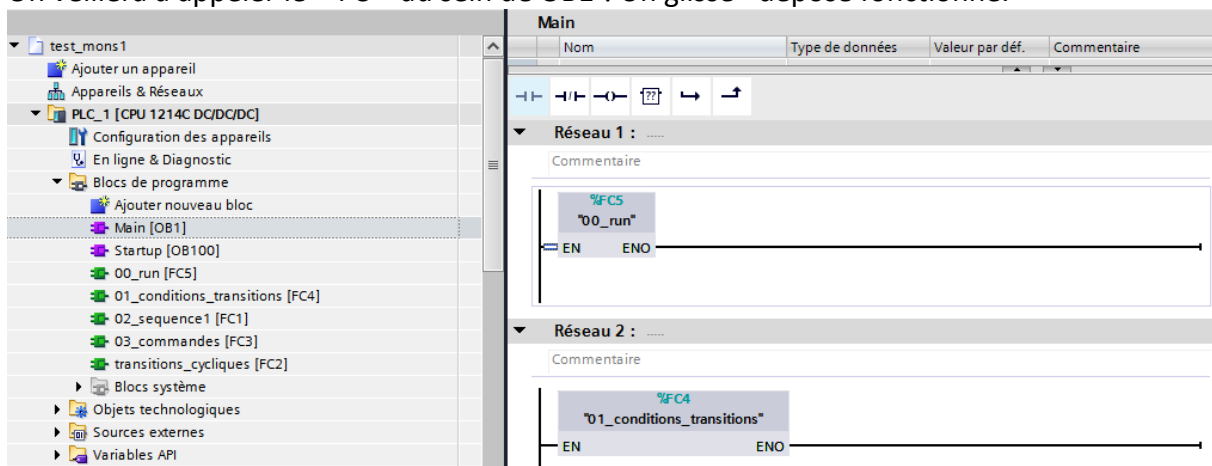
On peut créer des FC (fonctions) appelées au sein de OB1 pour la structure.

La FC sera au choix parmi 3 langages :      CONT (contact -> Ladder)  
  LOG (blocs fonctionnels)  
  SCL (langage littéral)

On peut ajouter un nouveau bloc, choisir son langage et lui donner un nom.



On veillera à appeler le « FC » au sein de OB1 ! Un glissé –déposé fonctionne.



## Réalisation d'un bargraph par commande HMI en semi-automatique

Les « FC » appelé dans OB1 seront exécuté de gauche à droite et de haut en bas.

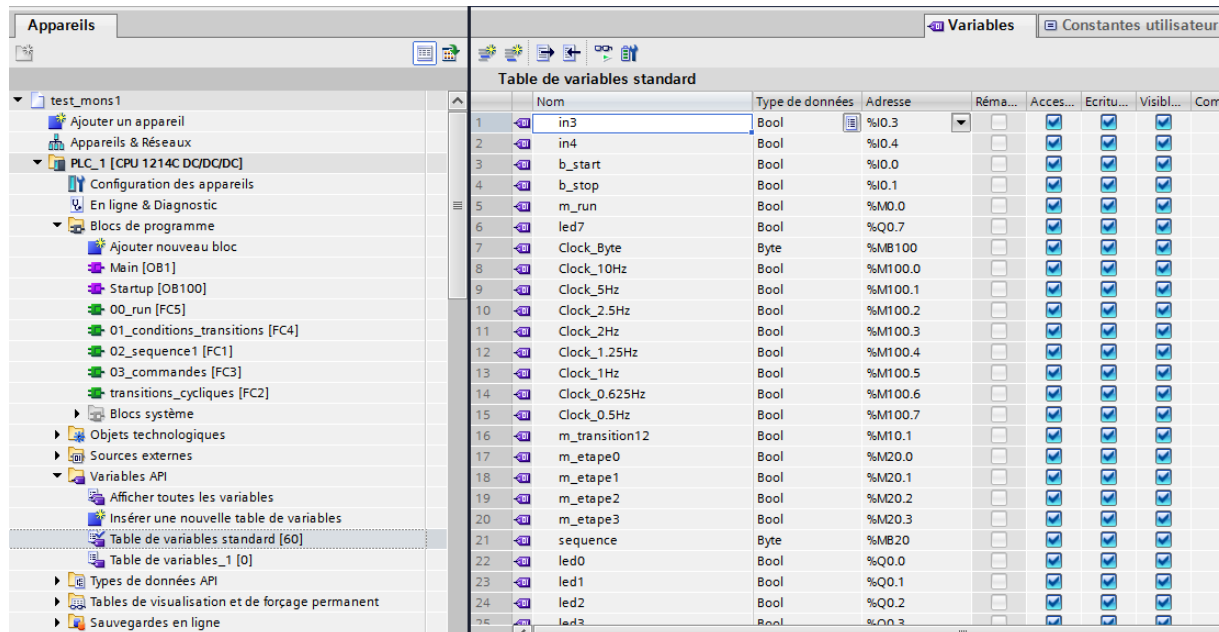
Au sein d'un « FC », en ladder par exemple, on trouvera des réseaux où chaque réseau se charge (en général) de la gestion d'une sortie.

**ATTENTION :** Chaque sortie (bobine) ne sera gérée qu'une seule fois... au sein d'un seul réseau !

Les variables associées aux sorties (%Q pour des sorties ou M pour mémontos) peut être définies au choix (+ le nom associé).

Les variables de « contacts » (% pour les entrées ou M pour les mémontos) idem.

L'ensemble des variables sont visibles via la table des variables standards.



	Nom	Type de données	Adresse	Réma...	Acces...	Ecritu...	Visibl...	Com
1	in3	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	in4	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	b_start	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	b_stop	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	m_run	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	led7	Bool	%Q0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Clock_Byte	Byte	%MB100		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Clock_10Hz	Bool	%M100.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Clock_5Hz	Bool	%M100.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Clock_2.5Hz	Bool	%M100.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Clock_2Hz	Bool	%M100.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Clock_1.25Hz	Bool	%M100.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Clock_1Hz	Bool	%M100.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Clock_0.625Hz	Bool	%M100.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Clock_0.5Hz	Bool	%M100.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	m_transition12	Bool	%M10.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	m_etape0	Bool	%M20.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	m_etape1	Bool	%M20.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	m_etape2	Bool	%M20.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	m_etape3	Bool	%M20.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	sequence	Byte	%MB20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	led0	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	led1	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	led2	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	led3	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

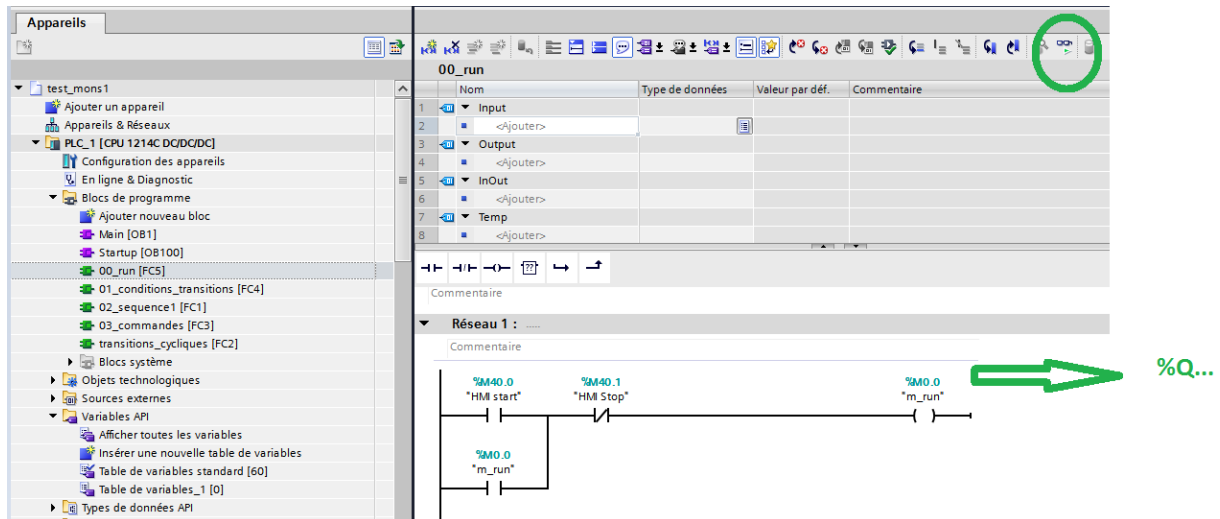
On peut choisir les noms, les zones mémoires,...

On veillera à réserver des zones mémoires de façon à s'y retrouver (avec des plages plus facilement identifiables)

## Réalisation d'un bargraph par commande HMI en semi-automatique

### Exercice : réaliser un maintient start / stop / led

Inspirez vous du « FC » ci-dessous en mettant entrées physiques (%I), et sortie led (%Q)



Nom	Type de données	Valeur par déf.	Commentaire
1	Input		
2	<Ajouter>		
3	Output		
4	<Ajouter>		
5	InOut		
6	<Ajouter>		
7	Temp		
8	<Ajouter>		

Une fois le « FC » fait et appelé au sein de OB1, on peut charger le programme dans l'API et voir en live en cliquant sur les jumelles comme ci-dessus.

On mettra le « start » sur une entrée, le « stop » sur une autre entrée pour activer une sortie (%Q). On peut éventuellement ajouter une variable memento supplémentaire (M) que l'on contrôle de cette façon (start/stop et maintien) + utiliser cette variable pour contrôler la sortie désirée (%Q) -> l'idée d'utiliser une variable intermédiaire.

## Vers un programme gérant une séquence.

### 1.) Gestion du mode semi-automatique

Comme pour l'exercice précédent, gérez une sortie nommée par exemple m\_run (le m... rappelle qu'on a un memento) qui illustre le mode run.  
Mettez ce programme en 1<sup>er</sup> au sein des appels de OB1

### 2.) Créez un programme sequence

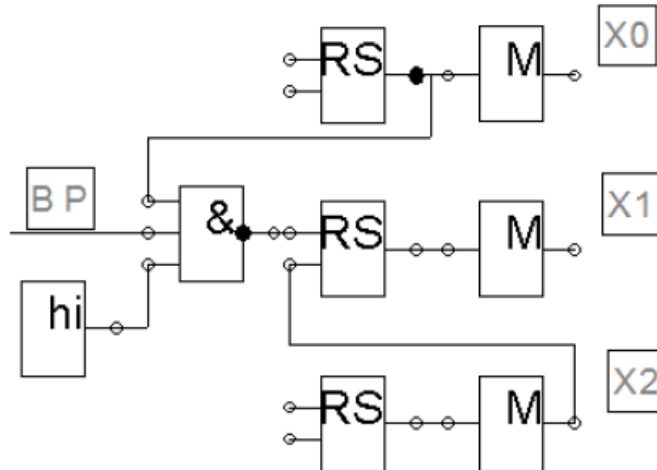
Ce programme va faire apparaitre les étapes d'une séquence bargraph.

On considèrera 4 étapes (etape0, etape1, etape2, etape3) constituant une séquence.

On créera un « faux grafcet » en gérant les activations/désactivations d'étapes nous-mêmes à l'aide de bascules SR (priorité Reset).

## Réalisation d'un bargraph par commande HMI en semi-automatique

Comme avec fluidsim :




Cependant on n'aura pas besoin d'utiliser de blocs mémoire.

Il faudra prévoir des bits mémoires pour mémoriser les états d'activation des différentes étapes de la séquence.

Remarque : Si on a plus de 8 étapes, on utilise plus un Byte mais un WORD (16 bits).  
Mais on doit se méfier lors de l'affectation du WORD (16 bits) qui n'est pas l'ordre naturel (convention LSB/MSB)  
→ Mieux d'affecter par Byte ou alors par mot binaire

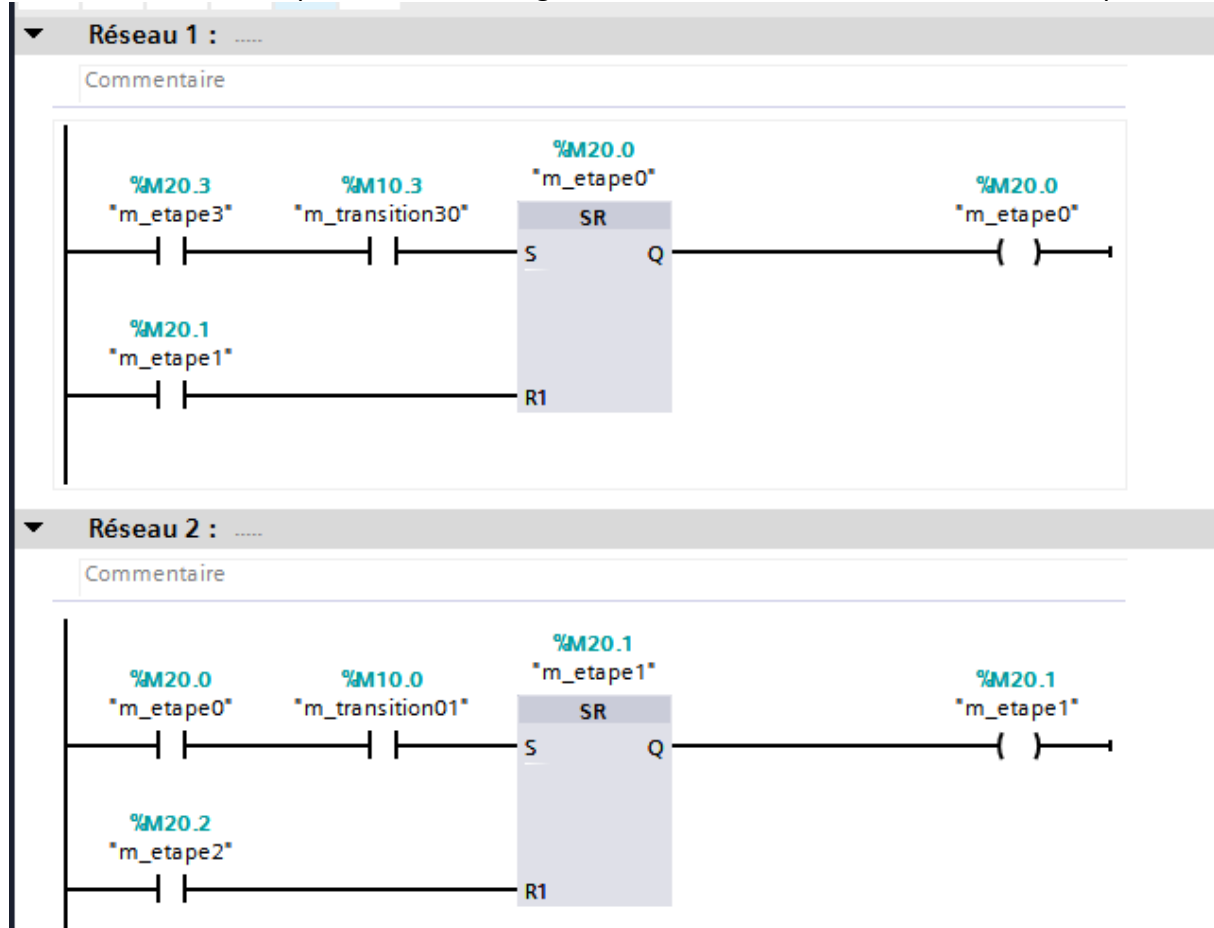
Les bits mémoires seront choisis, par facilité à des emplacements consécutifs, comme par exemple ci-dessous :

17		m_etape0	Bool	%M20.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18		m_etape1	Bool	%M20.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19		m_etape2	Bool	%M20.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20		m_etape3	Bool	%M20.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21		sequence	Byte	%MB20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

On remarque le BYTE (octet) séquence permettant d'agir directement sur le mot informant les activations des étapes de la séquence.

## Réalisation d'un bargraph par commande HMI en semi-automatique

De la même manière que ci-dessous, on gèrera l'activation / la désactivation des étapes :



On remarque qu'une étape sera désactivée dès que la suivante sera activée.

Une étape sera activée (une transition a lieu) que si on été à la fois :





- Actif à l'étape précédente
- La condition-transition est vrai

Les contacts en série traduisent la fonction ET

On gère tout ce « FC » et il sera en fait appelé en 3<sup>ème</sup> position au sein de l'OB1.

Le deuxième appel au sein de l'OB1 correspond à la gestion des transitions.

En effet, l'ensemble des transitions devront être gérés comme des informations mémentos également. Par exemple :

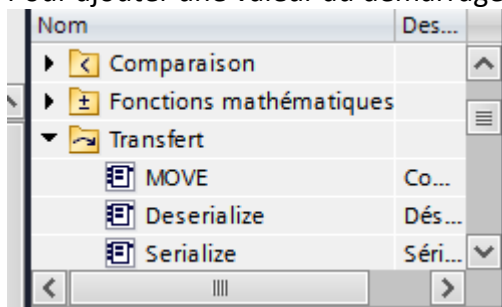
	m_transition01	Bool	%M10.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	m_transition12	Bool	%M10.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	m_transition23	Bool	%M10.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	m_transition30	Bool	%M10.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## Réalisation d'un bargraph par commande HMI en semi-automatique

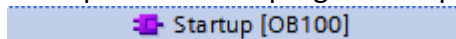
Avant de créer le programme « transition », il est nécessaire de s'assurer que l'API fixera bien l'étape initiale (étape0) comme active dès le départ.

On affectera alors, au démarrage de l'API, la valeur 1 au Byte « sequence ». Ce qui correspond au bit 0 à 1 et tous les autres à 0.

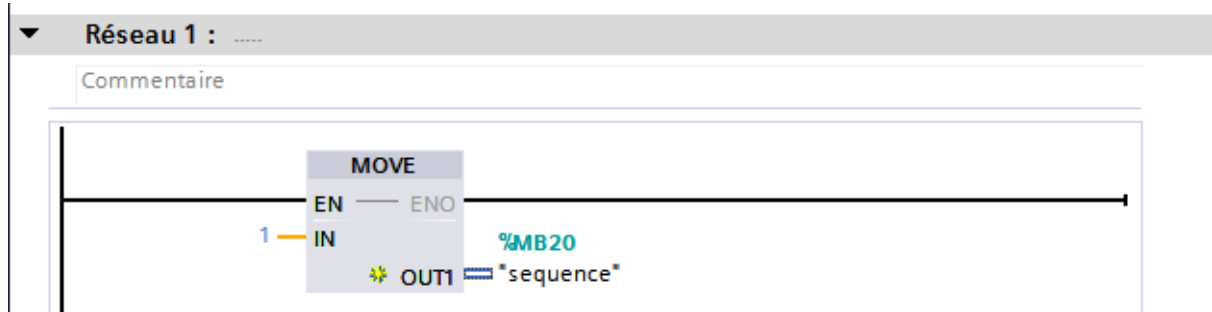
Pour ajouter une valeur au démarrage, on fait : Fonction de transfert-> move->



On le place dans un programme qui ne sera exécuté qu'une seule fois au démarrage :



Et on met la valeur 1 au sein de notre Byte « état d'activation »

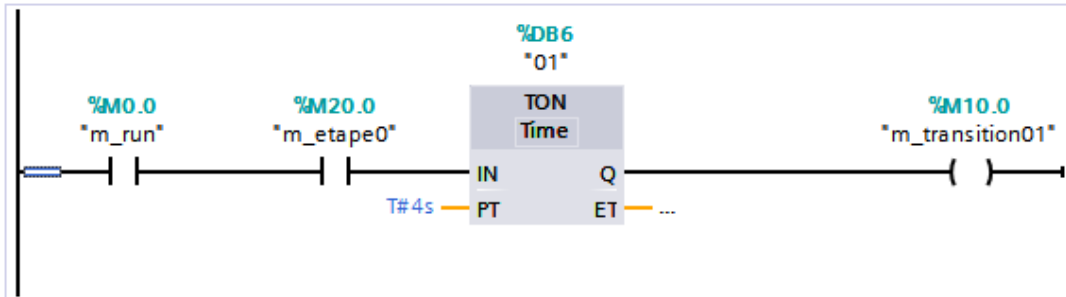


### 3.) Créer le programme transition

On crée le « FC » qui commande l'ensemble des éléments de transition.

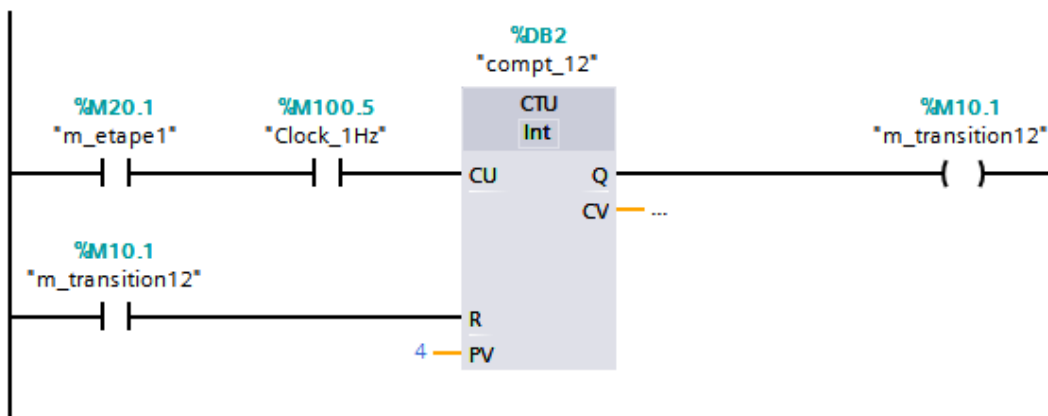
On essaiera de faire en sorte que les transitions ne soient que des délais pour le bargraph comme illustré ci-après.

## Réalisation d'un bargraph par commande HMI en semi-automatique



Réseau 2 : .....

Commentaire



Ci-dessus illustre l'utilisation d'un compteur TON (retard à l'enclenchement) ainsi que l'utilisation d'un compteur et d'une clock.

La clock est disponible grâce au Byte à l'adresse M100

	Clock_Byte	Byte	%MB100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Clock_10Hz	Bool	%M100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Clock_5Hz	Bool	%M100.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Clock_2.5Hz	Bool	%M100.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Clock_2Hz	Bool	%M100.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Clock_1.25Hz	Bool	%M100.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Clock_1Hz	Bool	%M100.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Clock_0.625Hz	Bool	%M100.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Clock_0.5Hz	Bool	%M100.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### 4.) Créez un programme commandes

Le programme de commande peut être, comme ici à droite dans un langage littéral.

Si on préfère, on peut utiliser le ladder à la place. On l'appellera en dernier au sein de OB1

```

3  "led0" := "m_etape0";
4  "led1" := "m_etape1";
5  "led2" := "m_etape2";
6  "led3" := "m_etape3";
7
8  "led7" := "m_run";

```

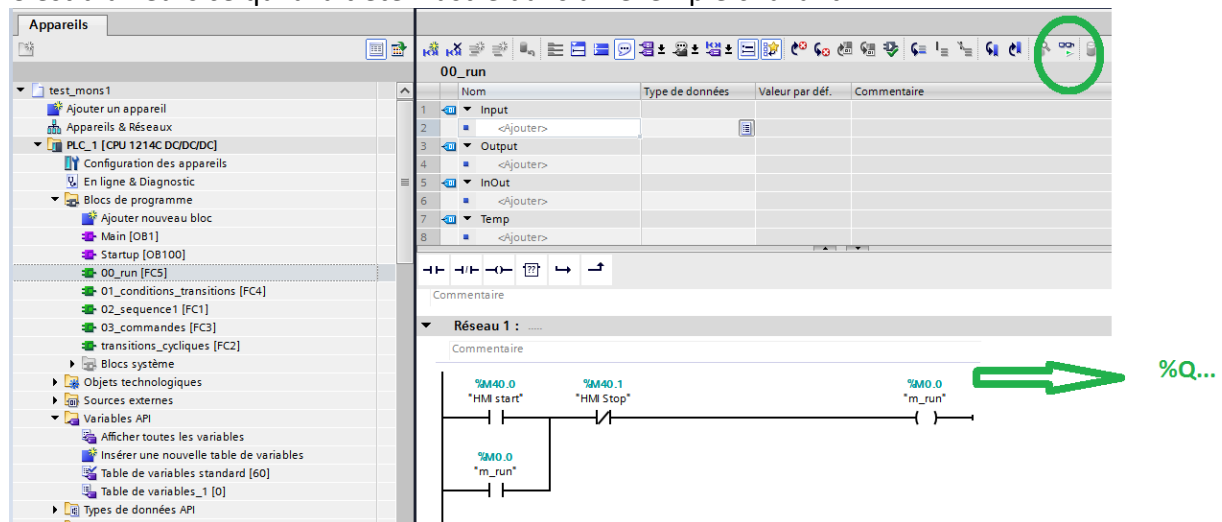
## Réalisation d'un bargraph par commande HMI en semi-automatique

### Vers une commande HMI

On fera le même programme que précédemment sauf qu'on remplacera le start/stop des entrées physiques par des boutons start/stop sur l'HMI.

On réservera des variables spécifiques aux boutons virtuels de l'HMI.

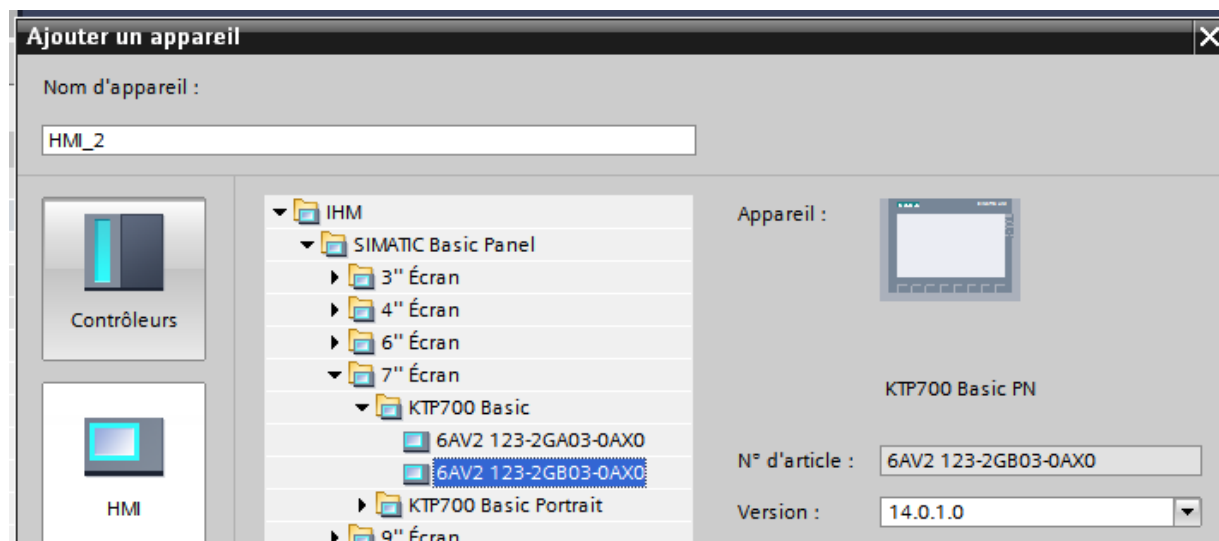
C'est d'ailleurs ce qui avait été illustré dans un exemple ci-avant :



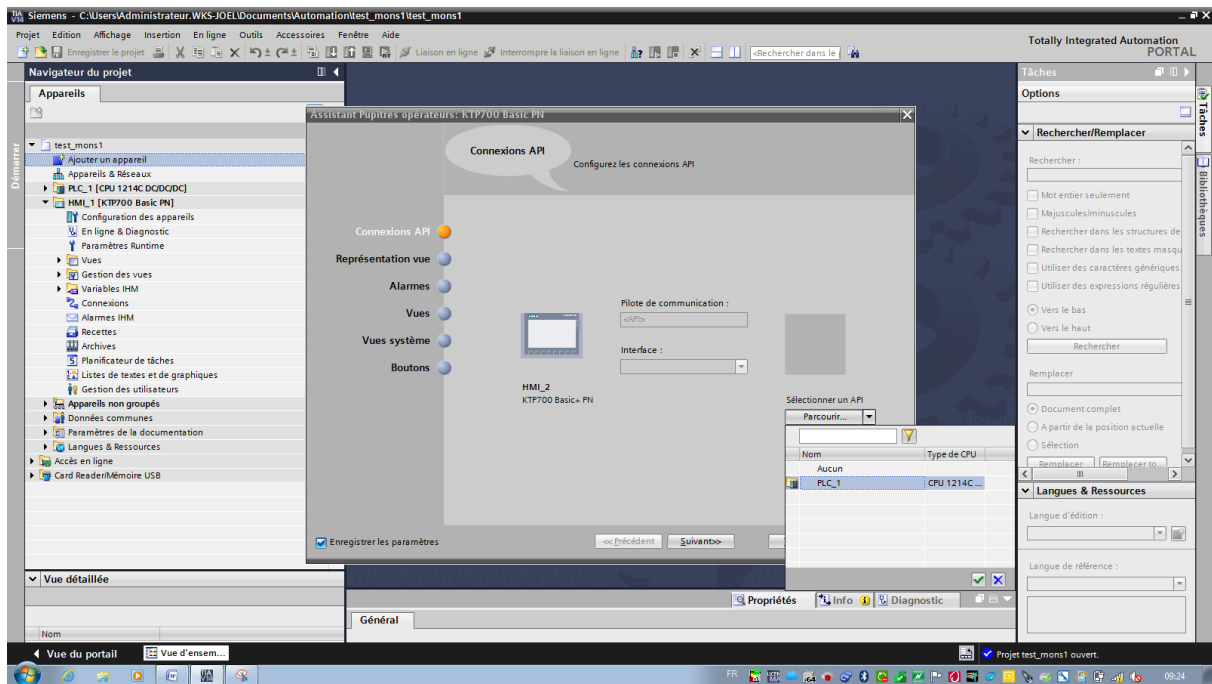
Les variables ont été nommées « HMI start » et « HMI stop » et ont été placées dans des zones d'adresses réservées.

On doit ajouter un appareil (HMI) au sein de notre arborescence de projet.

Ajout appareil → HMI. Et on choisira 7 pouces KTP700 (GB03 pour la version profinet)

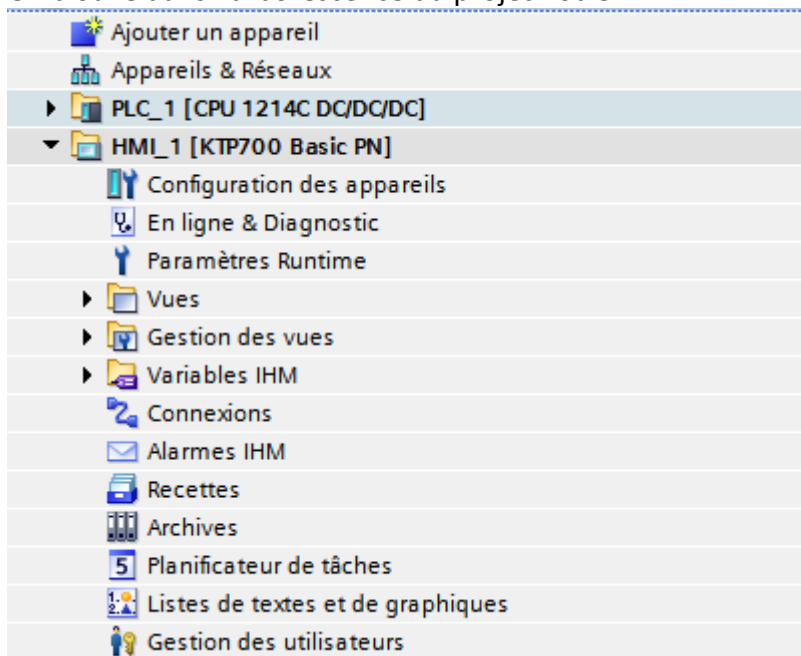


## Réalisation d'un bargraph par commande HMI en semi-automatique



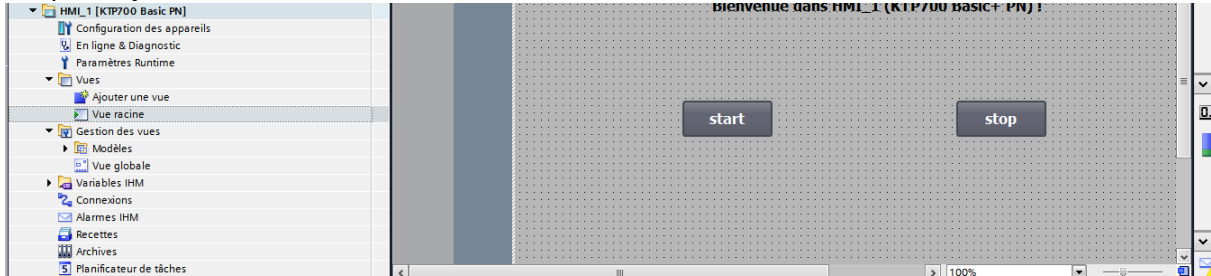
On paramètre ... suivant ... suivant et on termine.

On trouve dans l'arborescence du projet notre HMI.

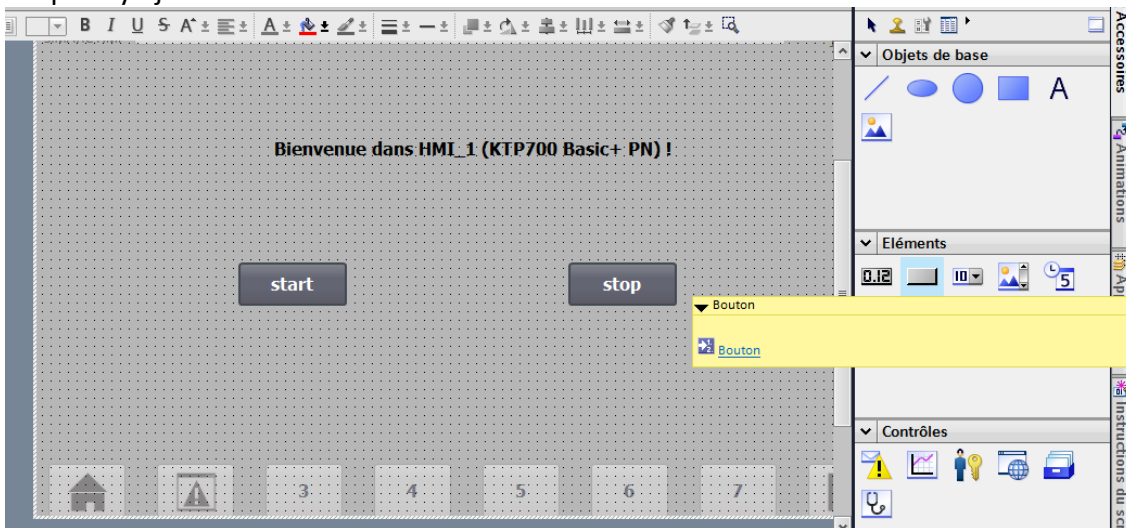


## Réalisation d'un bargraph par commande HMI en semi-automatique

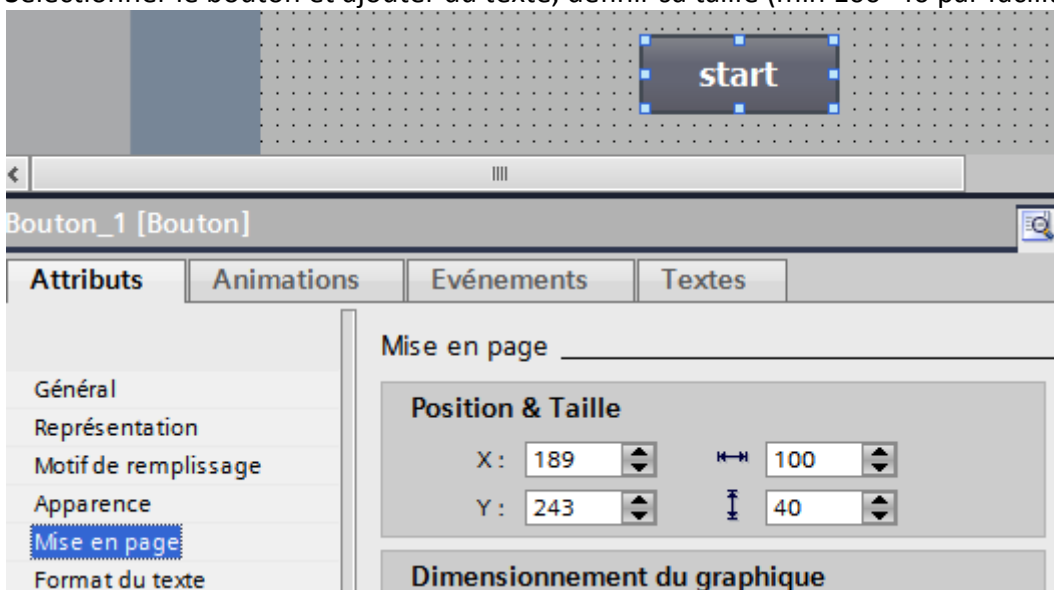
On peut ajouter/visualiser des vues :



On peut y ajouter un bouton



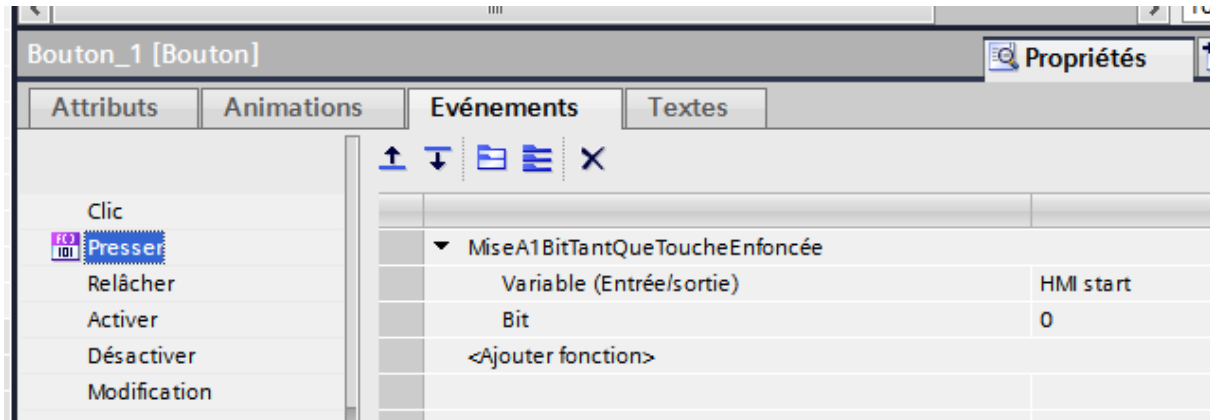
Sélectionner le bouton et ajouter du texte, définir sa taille (min 100\*40 par facilité) :



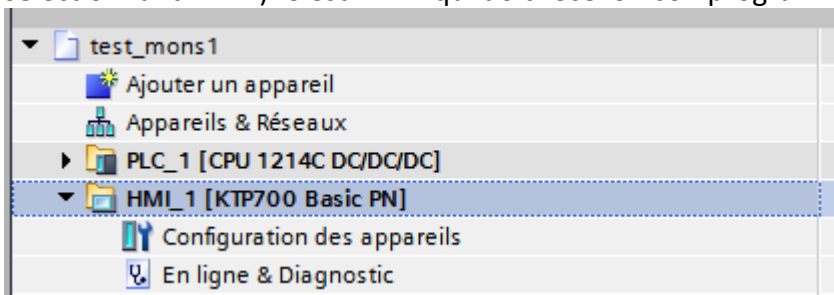
## Réalisation d'un bargraph par commande HMI en semi-automatique

---

Dans les événements on peut par exemple paramétrer comme ci-dessous et, surtout, préciser la variable associée.



Une fois terminé, on ne doit pas oublier de compiler et charger la partie liée à l'HMI (en sélectionnant l'HMI). C'est l'HMI qui doit recevoir son programme.



On teste ensuite l'application avec l'HMI.