

# AUTOMATISME

## Module AUTOM- TP1

24/09/2018

### PRISE EN MAIN TIA PORTAL

*S71200**Démonstrateur**(8 entrées TOR/8 sorties TOR/2 entrées ANA/1 sortie ANA)*

### Présentation :

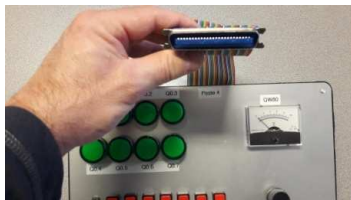
Pour ce premier TP, l'objectif est de se familiariser avec l'outil d'ingénierie TIA PORTAL. Les postes informatiques de la salle sont équipés de TIA PORTAL en version V13 SP2. L'API sera un S71200, modèle compact disposant de : 14 entrées TOR - 10 sorties TOR - 2 entrées ANA (0-10V) - 1 sortie ANA (0-10V).

Cet automate propose 3 langages CEI : LADDER - LOGIGRAMME - SCL

Après une présentation de TIA PORTAL, cette première séance permettra d'effectuer les opérations suivantes :

- Créer un projet
- Configurer l'architecture matérielle
- Configurer le réseau industriel
- Déclarer les variables API
- Tester les entrées sorties (table de visualisation et de forçage)
- Programmer des opérations binaires
- Programmer des opérations combinatoires
- Programmer un traitement séquentiel
- Programmer une temporisation
- Programmer un compteur

### Adressage des entrées-sorties API :

*Nappe et connecteur 50 points*

L'automate est relié électriquement au démonstrateur via **une nappe et un connecteur 50 points**.

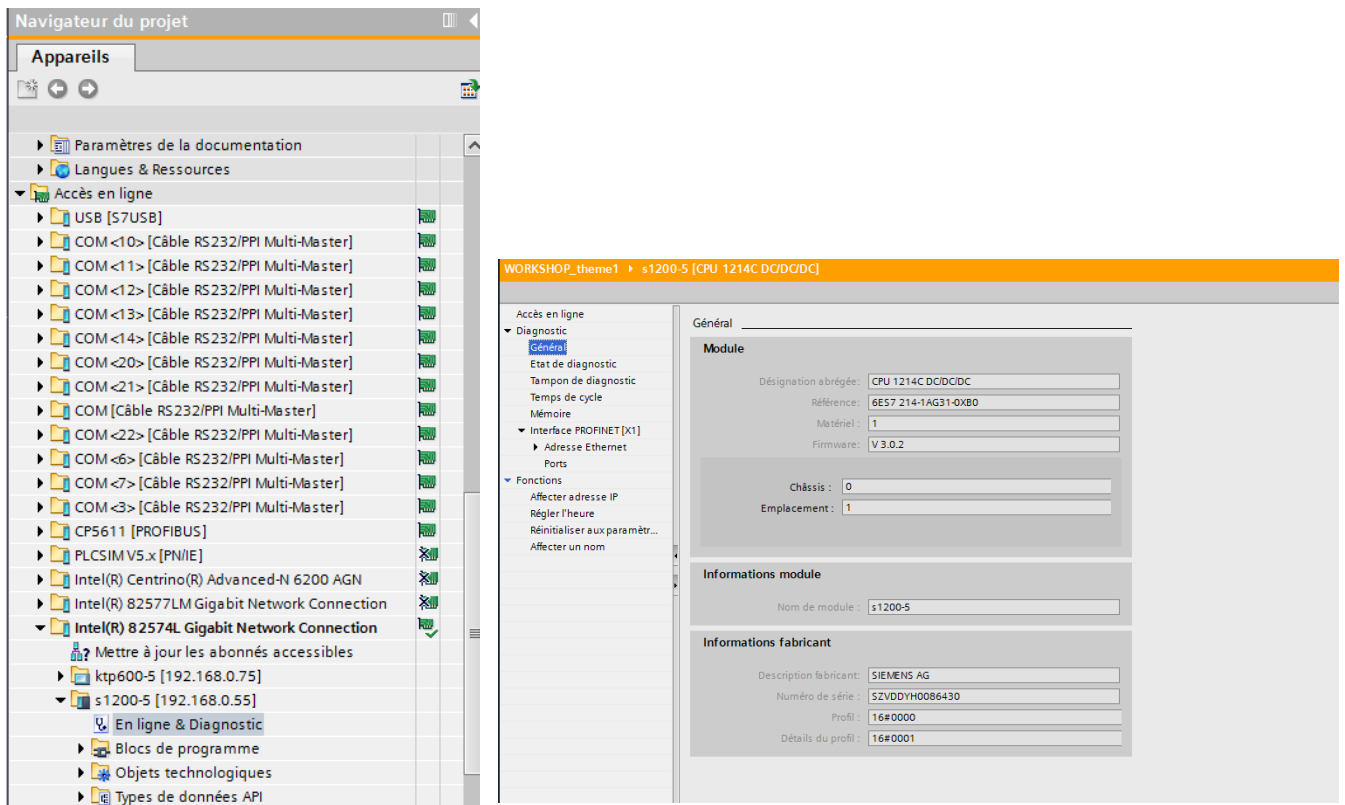
Pour ce démonstrateur, nous configurerons l'adressage des entrées/sorties de l'API suivant ce plan d'adressage :

Module	Empla..	Adresse I	Adresse Q	Type
	103			
	102			
	101			
▼ PLC_1	1			CPU 1214C DC/DC/DC
DI 14/DQ 10_1	1 1	0...1	0...1	DI 14/DQ 10
AI 2_1	1 2	64...67		AI 2
AQ 1x12BIT_1	1 3		80...81	Signal Board AQ1

### Configuration matérielle :

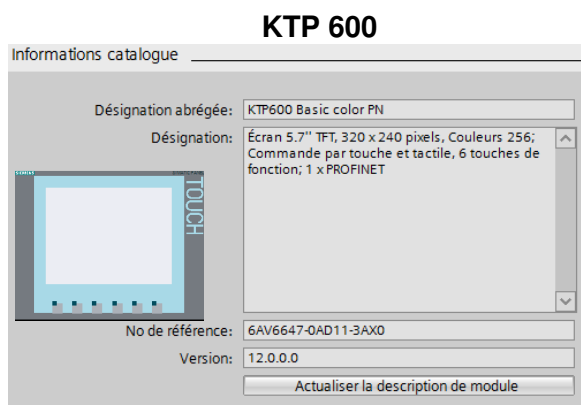
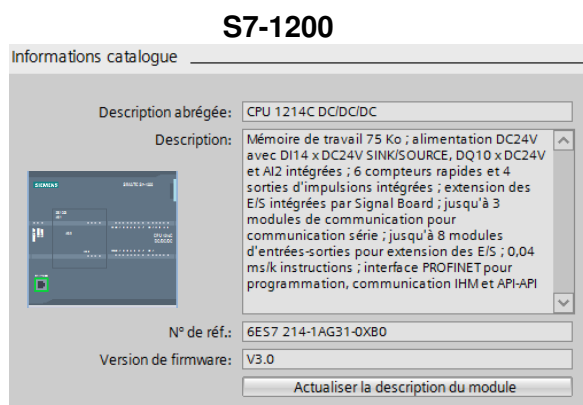
Sur un réseau existant, avec TIA PORTAL, nous sommes en mesure de scanner le réseau et de vérifier le matériel en ligne (adresse et firmware).

D'autres fonctionnalités sont proposées depuis cette interface comme : retour aux paramètres usine, affectation d'une adresse IP, diagnostic, temps de cycle, etc.

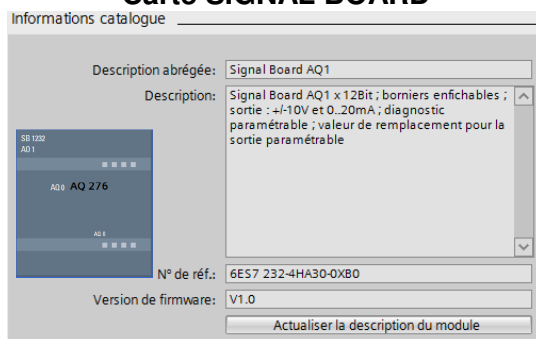


1. Scanner le matériel présent sur le réseau et vérifier les firmwares du S71200 et du KTP600 de votre poste de travail.

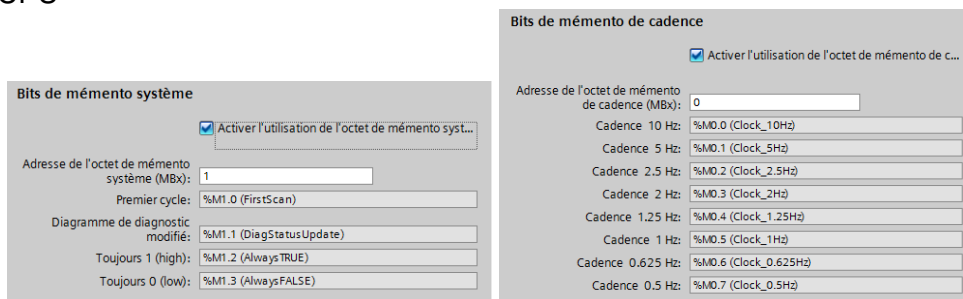
Voici la configuration matérielle que l'on souhaite obtenir :



### Carte SIGNAL BOARD

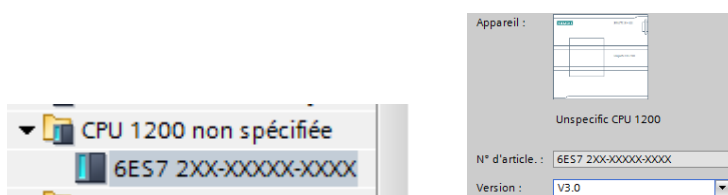


2. Effectuer la configuration matérielle complète API, HMI et SIGNAL BOARD
3. Déclarer le memento de cadence (byte 0) ainsi que les bits système (byte 1) de votre CPU

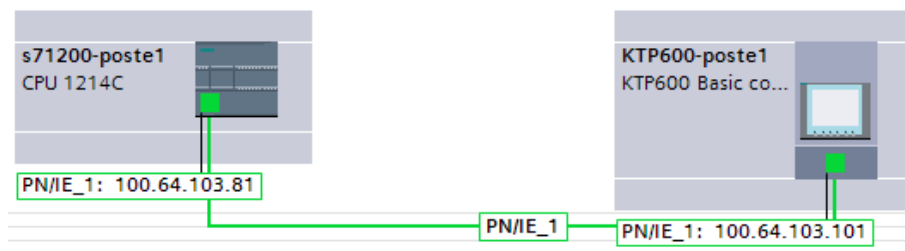


4. Observer les autres propriétés associées à votre CPU

**Remarque :** La configuration peut également se faire en détectant directement la CPU en ligne. Dans ce cas, il faut positionner dans la fenêtre de travail une CPU non spécifiée, attention le firmware doit tout de même correspondre.



## Adresse réseau :



### Plan d'adressage S71200 :

Adresse IP S7-1200 poste 1: 100.64.103.81  
Adresse IP S7-1200 poste 2 :100.64.103.82  
Adresse IP S7-1200 poste 3 :100.64.103.83  
Adresse IP S7-1200 poste 4: 100.64.103.84  
Adresse IP S7-1200 poste 5 :100.64.103.85  
Adresse IP S7-1200 poste 6 :100.64.103.86

### Plan d'adressage KTP600 :

Adresse IP KTP600 poste 1: 100.64.103.101  
Adresse IP KTP600 poste 2 :100.64.103.102  
Adresse IP KTP600 poste 3 :100.64.103.103  
Adresse IP KTP600 poste 4: 100.64.103.104  
Adresse IP KTP600 poste 5 :100.64.103.105  
Adresse IP KTP600 poste 6 :100.64.103.106

1. Configurer les adresses IP conformément au plan d'adressage de votre poste de travail.
2. Compiler et charger votre configuration matérielle API.

## Déclaration des variables :

1. Créer une table de variables : « **DEMONSTRATEUR** ».
2. Saisir l'ensemble de vos variables nécessaires à votre programmation : variables %I, %Q et %M.
3. Enregistrer vos variables.
4. Compiler et charger vos variables dans l'API.

DEMONSTRATEUR			
	Nom	Type de données	Adresse
<input type="checkbox"/>	S0	Bool	%I0.0
<input type="checkbox"/>	S1	Bool	%I0.1
<input type="checkbox"/>	S2	Bool	%I0.2
<input type="checkbox"/>	S3	Bool	%I0.3
<input type="checkbox"/>	S4	Bool	%I0.4
<input type="checkbox"/>	S5	Bool	%I0.5
<input type="checkbox"/>	S6	Bool	%I0.6
<input type="checkbox"/>	S7	Bool	%I0.7
<input type="checkbox"/>	octet_entrées	Byte	%IB0
<input type="checkbox"/>	H0	Bool	%Q0.0
<input type="checkbox"/>	H1	Bool	%Q0.1
<input type="checkbox"/>	H2	Bool	%Q0.2
<input type="checkbox"/>	H3	Bool	%Q0.3
<input type="checkbox"/>	H4	Bool	%Q0.4
<input type="checkbox"/>	H5	Bool	%Q0.5
<input type="checkbox"/>	H6	Bool	%Q0.6
<input type="checkbox"/>	H7	Bool	%Q0.7
<input type="checkbox"/>	octet_sorties	Byte	%QB0
<input type="checkbox"/>	etape_x0	Bool	%M100.0
<input type="checkbox"/>	etape_x1	Bool	%M100.1
<input type="checkbox"/>	etape_x2	Bool	%M100.2
<input type="checkbox"/>	etape_x3	Bool	%M100.3
<input type="checkbox"/>	etape_x4	Bool	%M100.4
<input type="checkbox"/>	octet_étapes	Byte	%MB100
<input type="checkbox"/>	pass	Bool	%M200.0
<input type="checkbox"/>	potentiomètre	Int	%IW64
<input type="checkbox"/>	voltmètre	Int	%QW80

**Remarque :** D'autres variables peuvent être déclarées ultérieurement. Pour que l'API les prenne en compte, il faudra les recharger.


### Test des entrées sorties :

1. Créer une table de visualisation et de forçage : «**TEST\_ES\_DEMONSTRATEUR** ».
2. Saisir l'ensemble des variables à tester : variables %I et %Q.
3. Mettre l'API en ligne
4. Tester vos variables d'E/S

Nom	Type de données	Adresse	Réma...	Visibl...	Acces...	Valeur visualisatio
S0	Bool	%I126.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
S1	Bool	%I126.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
S2	Bool	%I126.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> TRUE
S3	Bool	%I126.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
S4	Bool	%I126.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE
S5	Bool	%I126.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> FALSE

### Programmation d'opérations binaires :

1. Créer une fonction FC30 en LOGIGRAMME

 Opérations\_binaires [FC30]

2. En vous servant des instructions sur bit, programmer chaque équation dans des réseaux distincts




- $H1 = S0 \cdot S1$
- $H2 = S2 + S3$
- $H6 = S5 \oplus S6$
- $H3 = H4 = S4 \cdot (S0 + S5)$
- $H7 = (S1 + S3) \cdot (S6 + S7)$

3. Appeler la fonction FC30 dans l'OB1.
4. Compiler, charger et tester.

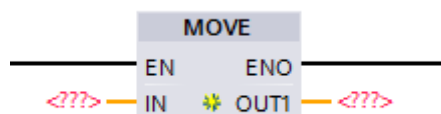
### Programmation d'opérations numériques :

#### **Instruction de transfert MOVE.**

1. Créer une fonction FC40 en LADDER

 Opérations\_numériques [FC40]

2. Placer deux blocs MOVE (réseau 1 et réseau 2 de FC40)



Premier bloc MOVE :

- Si S0=1 alors octet\_sortie := clock\_byte

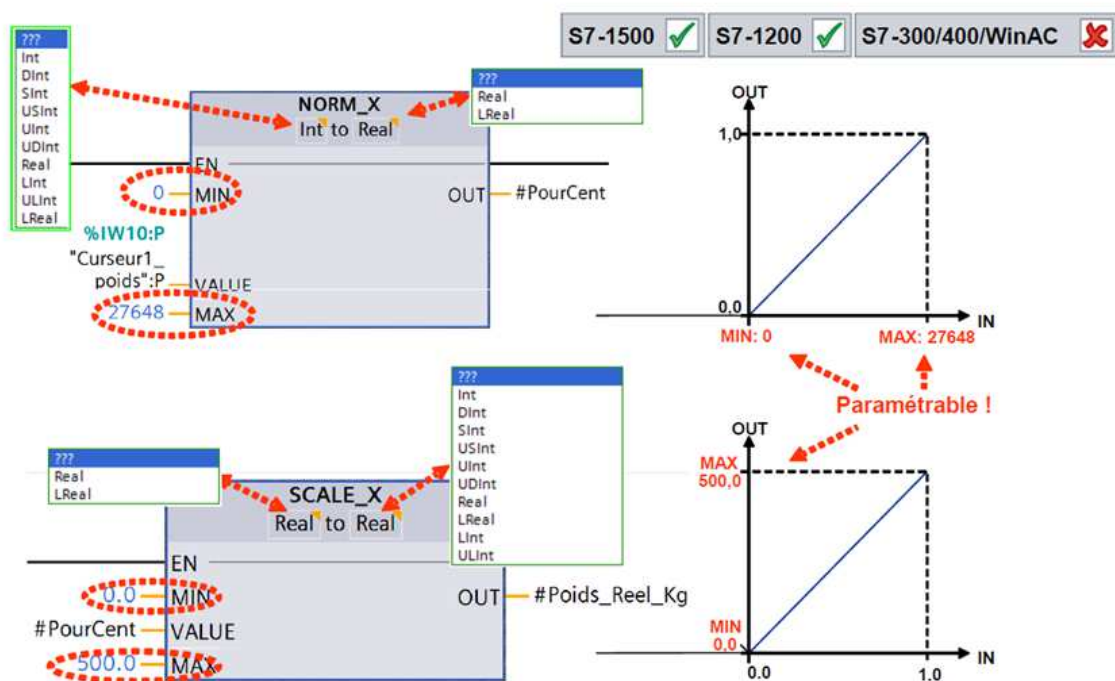
Deuxième bloc MOVE :

- Si S0=0 alors octet\_sortie := 0

3. Appeler la fonction FC40 dans l'OB1, la fonction précédente FC30 doit être désactivée (always\_false).
4. Compiler, charger et tester.

### Instruction de mise à l'échelle

Rappel des fonctions NORM\_X et SCALE\_X pour la mise à l'échelle de capteurs analogiques :



1. Créer une fonction FC50 en LADDER

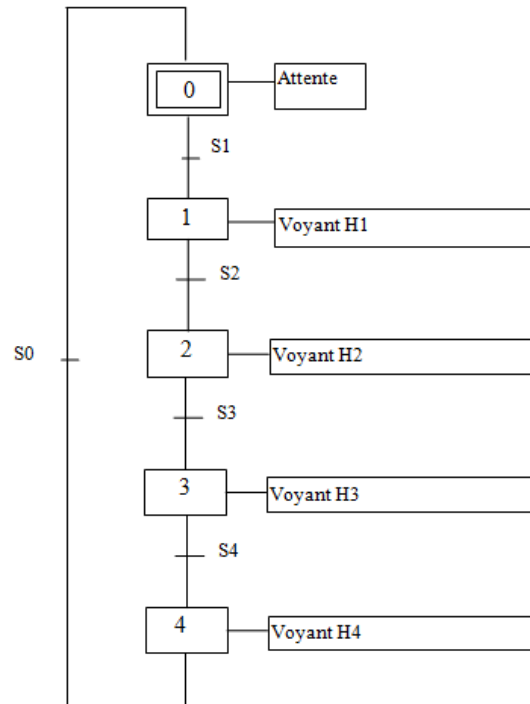
Mise\_à\_échelle [FC50]

2. Placer les deux instructions NORM\_X et SCALE\_X dans le même réseau.
3. Programmer ces instructions pour mettre à l'échelle la variable « potentiomètre » : %IW64 à l'échelle en 0-100%
4. Appeler la fonction FC50 dans l'OB1
5. Compiler, charger et tester.

## Programmation d'un traitement séquentiel :

**Rappel :** Règles de GRAFCET, une étape doit avoir deux équations, une équation d'activation et une de désactivation. Votre programme doit permettre l'activation de l'étape initiale lors du chargement du programme (bit First scan)

Graphe de fonctionnement



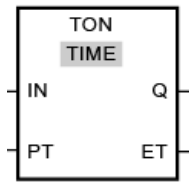
1. Créer deux fonctions fonction FC60 et FC70 en LADDER

➤ Actions\_GRAFCET [FC70]

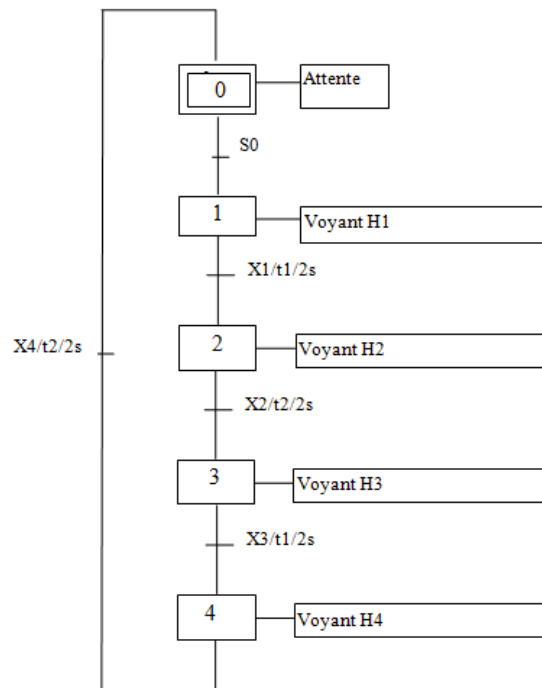
➤ GRAFCET [FC60]

2. Programmer le traitement séquentiel (FC60) correspondant au GRAFCET ci-dessus.
3. Programmer les actions associées (FC70) correspondant au GRAFCET ci-dessus.
4. Appeler les fonctions FC60 et FC70 dans l'OB1 (Attention aux conflits éventuels avec les autres fonctions).
5. Compiler, charger et tester.

## Programmation d'une temporisation :



Paramètres	Déclaration	Type de données	
		S7-1200	S7-1500
IN	Input	BOOL	BOOL
PT	Input	TIME	TIME, LTIME
Q	Output	BOOL	BOOL
ET	Output	TIME	TIME, LTIME



1. Reprendre les 2 fonctions FC60 et FC70 en LADDER

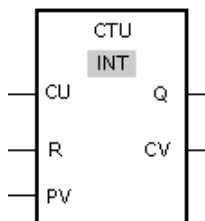
Actions\_GRAFCET [FC70]

GRAFCET [FC60]

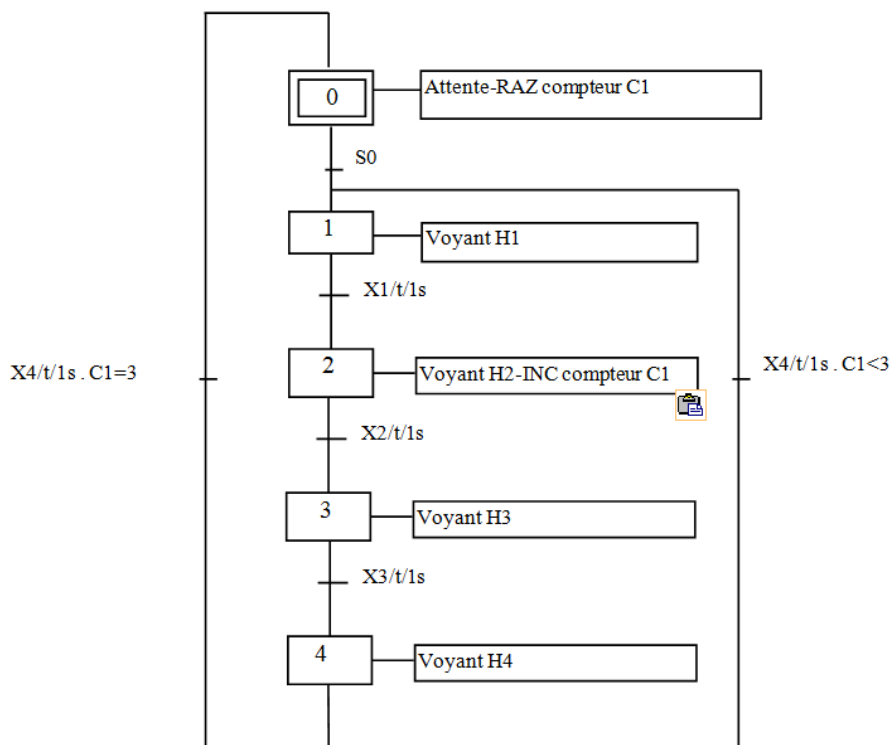
2. Modifier le traitement séquentiel (FC60) correspondant au GRAFCET ci-dessus en intégrant des temporisations de type TON : T1 et T2.
3. Modifier le programme les actions associées (FC70) correspondant au GRAFCET ci-dessus.
4. Compiler, charger et tester.



## Programmation d'un compteur :



Paramètre	Déclaration	Type de données	Zone de mémoire	
			S7-1200	S7-1500
CU	Input	BOOL	I, Q, M, D, L ou constante	I, Q, M, D, L ou constante
R	Input	BOOL	I, Q, M, D, L, P ou constante	I, Q, M, D, L, T, C, P ou constante
PV	Input	Nombres entiers	I, Q, M, D, L, P ou constante	I, Q, M, D, L, P ou constante
Q	Output	BOOL	I, Q, M, D, L	I, Q, M, D, L
CV	Output	Nombres entiers, CHAR, WCHAR, DATE	I, Q, M, D, L, P	I, Q, M, D, L, P



1. Reprendre les 2 fonctions FC60 et FC70 en LADDER

Actions\_GRAFSET [FC70]


GRAFCET [FC60]


2. Modifier le traitement séquentiel (FC60) correspondant au GRAFCET ci-dessus en intégrant un compteur CTU : C1
3. Modifier le programme les actions associées (FC70) correspondant au GRAFCET ci-dessus.
4. Compiler, charger et tester.


### Temporisation réglable :

On souhaite pouvoir régler le temps de la temporisation T1 avec le potentiomètre.  
Le temps de temporisation (Input : ET) pourra être réglée entre 1 et 15s.

1. Reprendre les 3 fonctions FC50, FC60 et FC70 en LADDER

 Actions\_GRAFCET [FC70]

 GRAFCET [FC60]

 Mise\_à\_échelle [FC50]

2. Modifier la fonction (FC50) : mise à l'échelle.
3. Modifier le programme les actions associées (FC70) : action du GRACET.
4. Compiler, charger et tester.