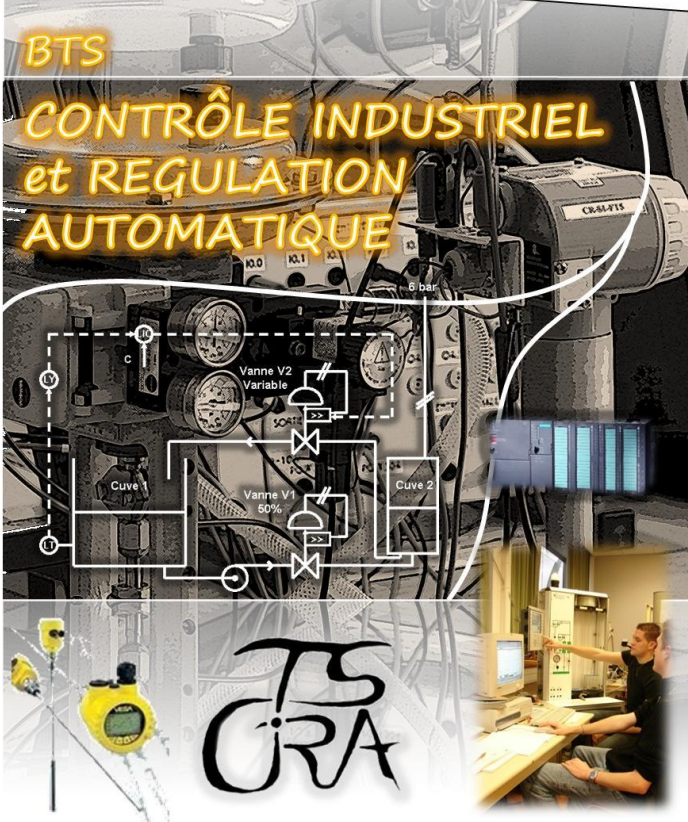


2012-2013

# TP D'AUTOMATISME TS2 CIRA (première partie)

Lycée technologique  
LOUIS COUFFIGNAL  
Strasbourg

Lycée  
COUFFIGNAL



YVES DARCO

LYCEE COUFFIGNAL

2012-2013

## TABLE DES MATIERES

<b>I] LANGAGES DE PROGRAMMATION DES API .....</b>	<b>2</b>
1] ELEMENTS COMMUNS AUX DIFFERENTS LANGAGES.....	2
2] LANGAGES LITTERAUX .....	4
3] LANGAGES GRAPHIQUES .....	6
4] STRUCTURATION D'UN PROGRAMME: .....	9
<b>II] PREMIERS PAS DANS LA PROGRAMMATION D'UN AUTOMATE SIEMENS EN S7 .....</b>	<b>12</b>
1-ADRESSAGE DES VARIABLES SIEMENS: .....	12
2- OUVERTURE ET ENREGISTREMENT D'UN PROJET EXISTANT : .....	13
3- VERIFICATION DE LA CONFIGURATION MATERIELLE DE L'API : .....	13
4- CREER UNE TABLE DE MNEMONIQUES : .....	15
5- PROGRAMMER UN RESEAU DANS UNE FONCTION FC1 ET L'APPELER DANS OB1 : .....	16
6- CREER UNE TABLE DE VARIABLES VAT1 : .....	17
7- TRANSFERER ET TESTER LE FONCTIONNEMENT DU PROGRAMME : .....	18
8- FORÇAGE DES VARIABLES : .....	19
9- REPRESENTATION D'UN GRAFCET: .....	20
10- VARIABLES ASSOCIEES AUX ETAPES DU GRAFCET: .....	23
11- MISE A L'ECHELLE D'UNE ENTREE ANALOGIQUE: .....	24
12- MISE A L'ECHELLE D'UNE SORTIE ANALOGIQUE: .....	25
<b>III] UTILISATION DES ENTREES ET SORTIES ANALOGIQUES API .....</b>	<b>27</b>
<b>IV] PLANNING EXERCICES DE REVISION DE TP D'AUTOMATISME TS2CIRA .....</b>	<b>32</b>
<b>V] EXERCICES DE REVISION TP D'AUTOMATISME TS2CIRA .....</b>	<b>33</b>
<b>VI] TP TEST AUTO N°1.....</b>	<b>34</b>
<b>VII] TP TEST AUTO N°2.....</b>	<b>35</b>
<b>TEST D'EVALUATION TP N°2 AUTOMATISME.....</b>	<b>37</b>
<b>VIII] TP TEST TP AUTO N°3 .....</b>	<b>38</b>
<b>TEST D'EVALUATION TP N°3 AUTOMATISME.....</b>	<b>40</b>
<b>IX] EXEMPLE DE PROGRAMMATION D'UN PROFIL SUR SORTIE ANALOGIQUE (STEP7) : .....</b>	<b>41</b>
<b>X] DIAGNOSTIC VANNE SAMSON AVEC POSITIONNEUR PA.....</b>	<b>42</b>
<b>XI] PROCEDURE ENREGISTREMENT ET TRANSFERT DES MESURES LOGGER GL220→PC (EXCEL) .....</b>	<b>48</b>
<b>XII] ADRESSAGE DES MOTS INTERNES D'UN API.....</b>	<b>54</b>

## II LANGAGES DE PROGRAMMATION DES API

### NORME CEI 61131-partie 3

Cette norme des langages de programmation des automates programmables permet de les classer suivant trois catégories :

Langages littéraux	Langages graphiques	Structure de programme
<b>Langage IL</b> (liste d'instructions) <b>Langage ST</b> (littéral structuré)	<b>Langage LD</b> (à contacts) <b>Langage FBD</b> (à blocs fonctionnels)	<b>Diagramme SFC</b> (suite de séquences)

La structure d'un programme séquentiel utilise les éléments **du diagramme fonctionnel de séquence SFC à ne pas confondre avec la description du comportement d'un système** (connu sous le nom de GRAFCET en France, voir norme **CEI 60848**)

## I] ELEMENTS COMMUNS AUX DIFFERENTS LANGAGES

### 1-1) Principaux types de données élémentaires :

Le type de données, la description et le nombre de bits sont précisés dans le tableau suivant :

Type de données	Description	Taille ou nombre de bits	Etendue
<b>BOOL</b>	Booléen	1	0 ou 1
<b>BYTE</b>	Cordon de 8 bits	8	Pas d'étendue numérique pour ce type de données
<b>WORD</b>	Cordon de caractères de longueur 16	16	
<b>DWORD</b>	Cordon de caractères de longueur 32	32	
<b>LWORD</b>	Cordon de caractères de longueur 64	64	
<b>INT</b>	Entier signé	n=16	$-2^{n-1}$ à $(2^{n-1} - 1)$
<b>UINT</b>	Entier non signé	n=16	0 à $(2^n - 1)$
<b>REAL</b>	Nombre réel	32	Virgule flottante
<b>STRING</b>	Cordon de caractères	Encadré par deux apostrophes	
<b>TIME</b>	Durée	Dépend de l'application concernée	

### 1-2) Représentation des variables :

Une variable permet d'identifier des objets de données dont le contenu peut varier (données associées aux entrées, aux sorties ou aux emplacements mémoire de l'API)

Le tableau suivant permet de représenter symboliquement une variable. La variable débute par le symbole %

Préfixe	Signification
I	Input : Emplacement d'une entrée automate
Q	Output : Emplacement d'une sortie automate
M	Emplacement de memento ou mémoire interne
X	Taille d'un seul bit
B	Taille d'un byte ou octet
W	Taille d'un word : mot de 16 bits
D	Taille d'un double Word : mot double de 32 bits
L	Taille d'un mot long : mot de 64 bits

Exemples :

%W125 : Emplacement du mot d'entrée 125

%QB17 : Emplacement de l'octet de sortie 17

%MD48 : Emplacement du mot mémoire double 48

### 1-3) Unités d'organisation de programmes : (structuration d'un programme grâce à 3 types de modules)

- Fonction : Module ayant plusieurs entrées possibles, **une seule variable de sortie** et **pas de mémoire interne**

- Bloc fonctionnel: Module **ayant plusieurs variables d'entrée et de sortie possibles** et **une mémoire interne**

- Programme : Module construit à l'aide de fonctions et de blocs fonctionnels

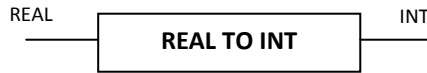
L'organisation interne d'un programme peut faire intervenir un diagramme fonctionnel de séquence SFC.

## Inventaire des fonctions standards:

Description des principales fonctions communes à tous les langages de programmation d'API :

### - Fonctions de conversion de types

**Exemple** cette fonction permet la conversion d'une variable d'entrée du type réel en une variable de sortie du type entier



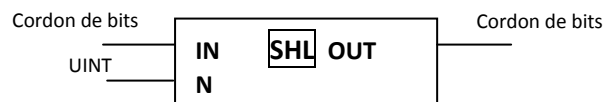
### - Fonctions numériques

**Exemple** : cette fonction permet de calculer la racine carrée d'une variable d'entrée

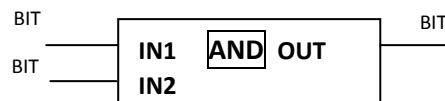


### - Fonctions cordons de bits

**Exemple** : cette fonction permet le décalage à gauche de N bits d'une variable d'entrée et le remplissage de zéros à droite

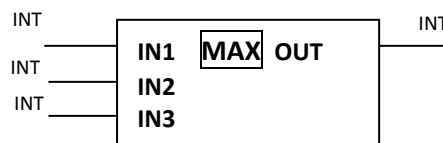


**Exemple** : cette fonction réalise le ET Booléen

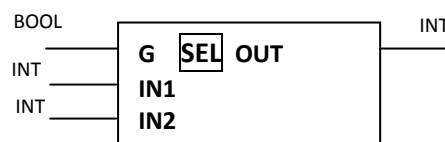


### -Fonctions de sélection et de comparaison

**Exemple** : cette fonction permet de déterminer la valeur maximale entre trois variables d'entrée



**Exemple** : cette fonction permet de sélectionner une des deux variables d'entrée suivant la variable G



**Exemple** : cette fonction réalise la comparaison de supériorité entre deux variables d'entrée (IN1>IN2)



## 2] LANGAGES LITTERAUX

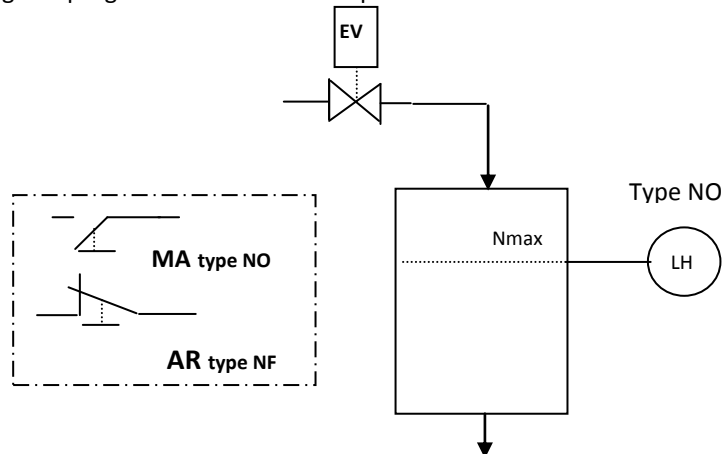
### 2-1) Langage IL (Instruction List ou langage à liste d'instructions)

Comme son nom l'indique, le programme est constitué d'une suite d'instructions respectant le format suivant :

Etiquette (non obligatoire)	Opérateur	Opérande(s)	Commentaire (non obligatoire)
--------------------------------	-----------	-------------	----------------------------------

Ce langage est proche du langage de programmation d'un microprocesseur : l'assembleur

**Exemple :** Soit à commander une électrovanne EV du schéma TI suivant :



Etiquette (non obligatoire)	Opérateur	Opérande(s)	Commentaire (non obligatoire)
Début :	AND (		
	OR	%I0.0	BP Marche NO
	OR	%Q4.0	Electrovanne
	)		
	AND	%I0.1	BP Arrêt NF
	AND N	%I0.2	Niveau haut NO
	ST	%Q4.0	Affectation électrovanne

Exemple de programmation du logiciel STEP7 de Siemens :

Electrovanne\SIMATIC 300(1)\CP...\Mnémoniques				
	Mnémonique	Opérande	e de donn	Commentaire
1				
2	MA	I	0.0	BOOL BP marche type NO
3	AR	I	0.1	BOOL BP arrêt type NF
4	LH	I	0.2	BOOL Niveau haut type NO
5	EV	Q	4.0	BOOL Electrovanne type NF

Programme correspondant en IL ou LIST :

Réseau 1: Commande électrovanne				
Automatisation de l'électrovanne de remplissage				
A(				
O	"MA"	I0.0		-- BP marche type NO
O	"EV"	Q4.0		-- Electrovanne type NF
)				
A	"AR"	I0.1		-- BP arrêt type NF
AN	"LH"	I0.2		-- Niveau haut type NO
=	"EV"	Q4.0		-- Electrovanne type NF

### 2-2) Langage ST ( Structured Text) ou langage littéral structuré)

Ce langage est composé d'expressions littérales constituées d'opérateurs et d'opérandes et d'énoncés.

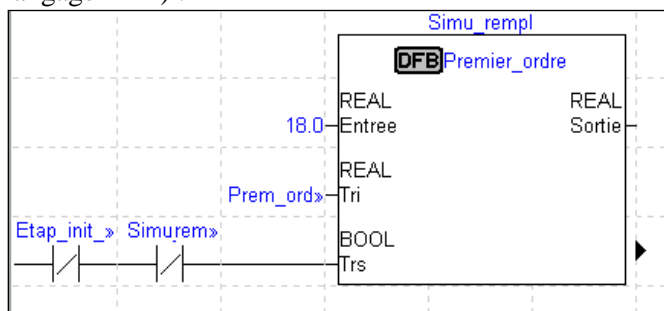
Ce langage est proche d'un langage informatique comme le PASCAL.

**Exemple 1** de la commande de l'électrovanne :

```
%L1 (« commande électrovanne »)
IF (%I0.0 OR %Q4.0) AND %I0.1 AND NOT %I0.2 THEN SET %Q4.0
END IF;
```

**-Exemple 2** issu d'un programme de démonstration « Machine de dosage et mélange produits » du logiciel **PL7 de Schneider**.

Soit la simulation d'un premier ordre sous forme d'un bloc fonctionnel DFB « Premier ordre » (voir langage FBD) :



Le contenu de ce bloc écrit en **langage ST** est le suivant :

```

(** Routine de Simulation d'un procédé de 1er ordre pour Sortie PID **)
(* Récupération de la Période de la tâche MAST nécessaire à la formule *)

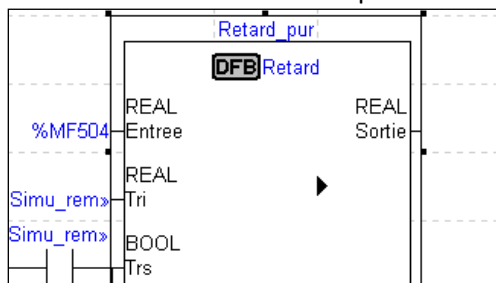
IF Trs THEN
    Sortie := Tri;
ELSE
    Te:=INT_TO_REAL(%SW0)/1000.0;
    Sortie:=(Filtrage/(Filtrage+Te))*Sortie+(Gain*Te)/(Filtrage+Te)*Entree;
END_IF;

```

Avec :

**%SW0** : Période de scrutation de la tâche maître. Permet de modifier la période de la tâche maître définie en configuration, par le programme utilisateur ou par le terminal. La période est exprimée en ms (1..255ms). %SW0=0 en fonctionnement cyclique.

Soit la simulation d'un retard pur sous forme d'un bloc fonctionnel DFB « Retard » :



Le contenu de ce bloc écrit en **langage ST** est le suivant :

```

Nb := REAL_TO_INT(1000.0*Retard/INT_TO_REAL(%SW0));
IF (Nb>=999) THEN
    Nb := 999;
END_IF;
IF Trs THEN
    Memoire := Tri;
    Sortie := Tri;
    Pos := 0;
ELSE
    Sortie := Memoire[Pos];
    Memoire[Pos]:= Entree;
    Pos := Pos +1;
    IF (Pos>= Nb) THEN
        Pos := 0;
    END_IF;
END_IF;

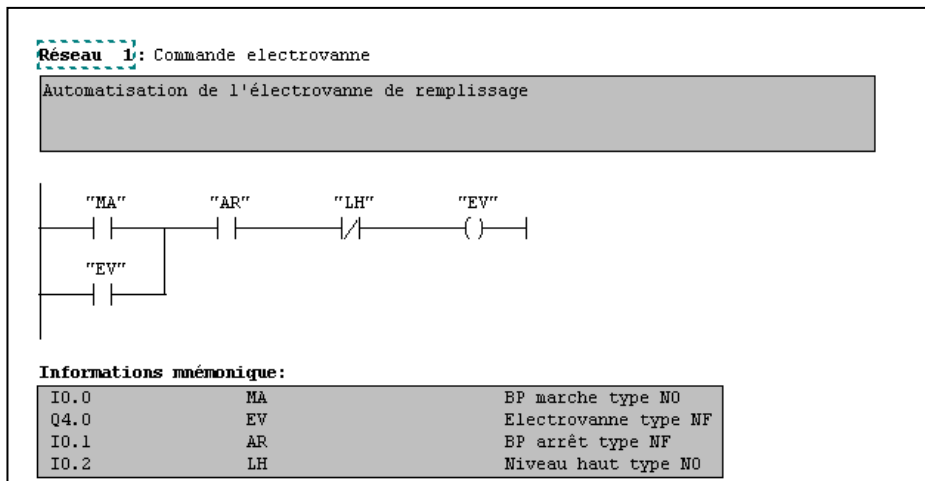
```

### 3] LANGAGES GRAPHIQUES

#### 3-1) Langage LD (Ladder Diagram) ou langage à contacts)

Ce langage est constitué de réseaux de contacts et de bobines entre deux barres d'alimentation. Ce langage est proche des schémas électriques.

En reprenant l'exemple de la page précédente et sur API Siemens en langage à contacts STEP 7 :

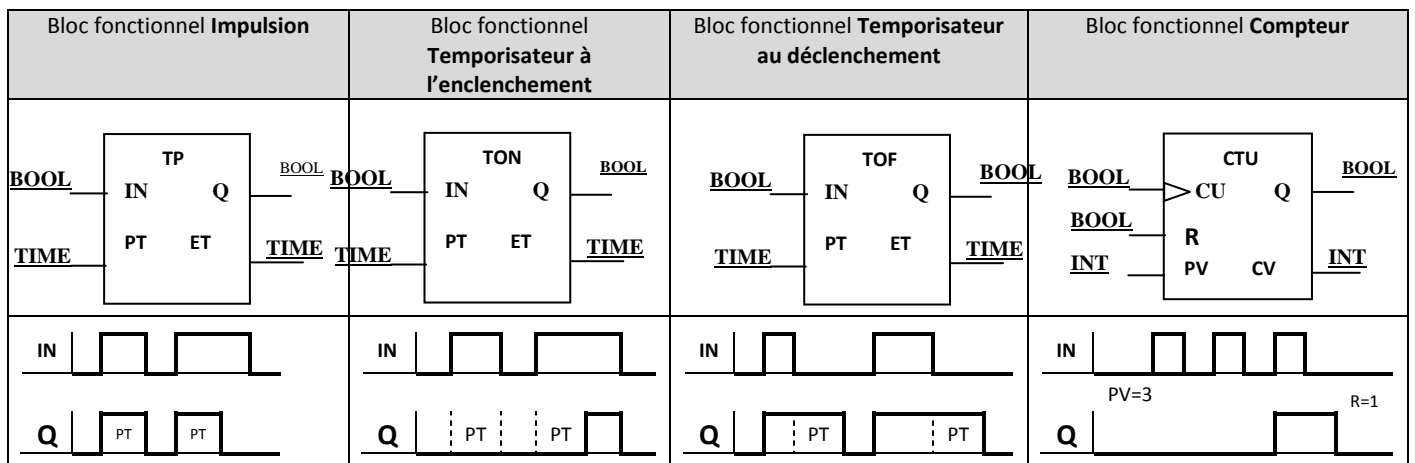


#### 3-2) Langage FBD (Function Block Diagram) ou langage en blocs fonctionnels

Ce langage se compose de réseaux de fonctions **préprogrammées ou non**, représentées par des rectangles. Ces blocs fonctionnels sont connectés entre eux par des lignes, le flux des signaux se faisant de la sortie (à droite) d'une fonction vers l'entrée à gauche de la fonction raccordée.

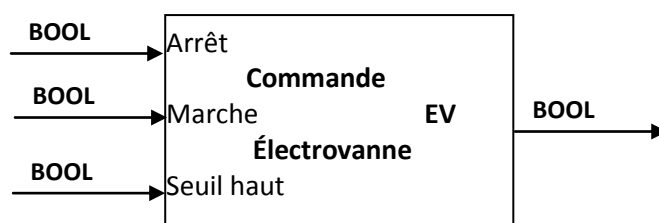
#### Exemples de blocs fonctionnels standards (fourni par le constructeur de logiciel):

- Bloc fonctionnel compteurs
- Bloc fonctionnel temporisateurs
- Bloc fonctionnel PID...



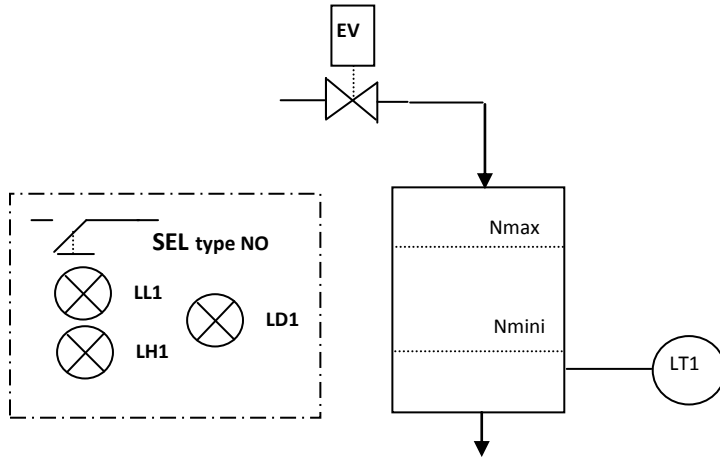
#### Exemples de blocs fonctionnels utilisateurs (développés par le programmeur et réutilisables)

- Exemple 1 de la page précédente :



- Exemple 2 : Alarmes

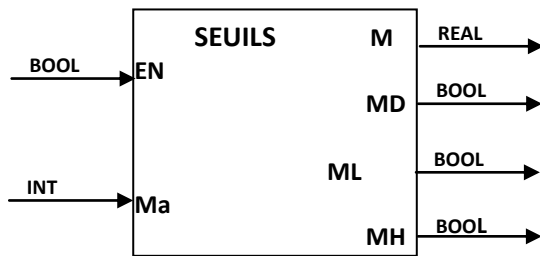
**d'une cuve**



**Cahier des charges :**

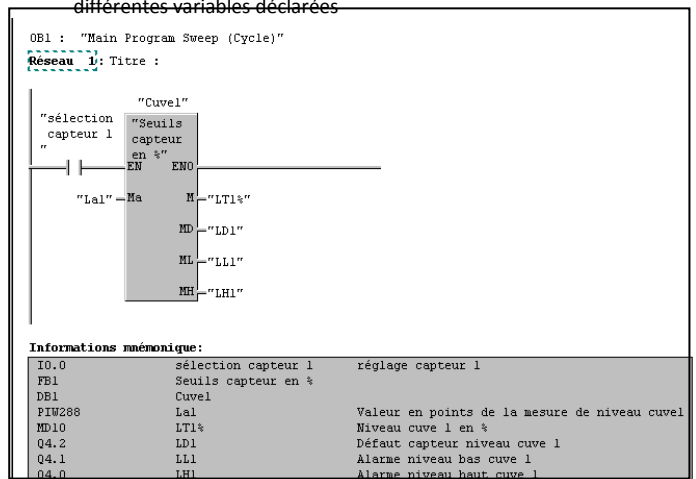
- Surveiller les seuils haut et bas d'une cuve
- Convertir le niveau en %
- Signaler un défaut du capteur

Soit à créer le bloc fonctionnel suivant :



**programmer en STEP 7 :** À l'appel du bloc

FB on indique le N° du bloc de données d'instance DB associé (mémoire interne). Ce bloc sert à mémoriser l'état des différentes variables déclarées



Liste des variables formels associées au bloc fonctionnel « SEUILS » :

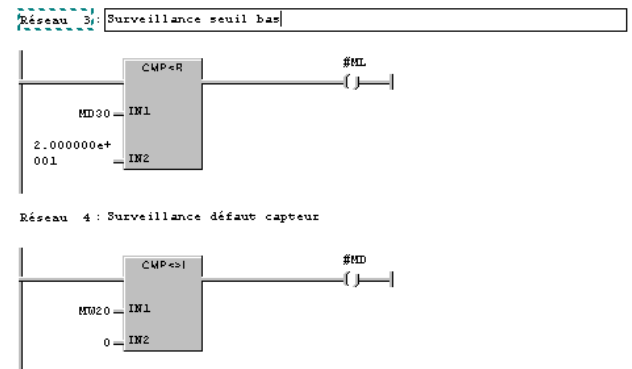
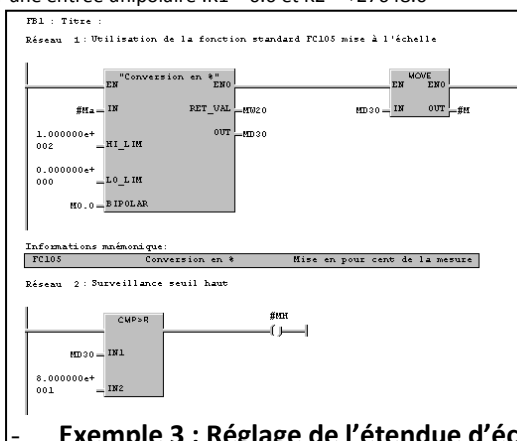
Adresse	Décl.	Nom	Type	Valeur initiale	Commentaire
0.0	in	Ma	INT	0	Valeur en "points" du capteur
2.0	out	M	REAL	0.000000e+000	Valeur en % du capteur
6.0	out	MD	BOOL	FALSE	Signalisation défaut capteur
6.1	out	ML	BOOL	FALSE	Seuil haut capteur de 80%
6.2	out	MH	BOOL	FALSE	Seuil bas capteur de 20%

**Le contenu du bloc fonctionnel est le suivant :**

Il fait appel à une fonction standard : « Mise à l'échelle (SCALE) » qui prend une valeur entière (IN) et la convertit selon l'équation ci-après en une valeur réelle exprimée en unités physiques, comprises entre une limite inférieure (LO\_LIM) et une limite supérieure (HI\_LIM), dans notre cas entre 0 et 100%.

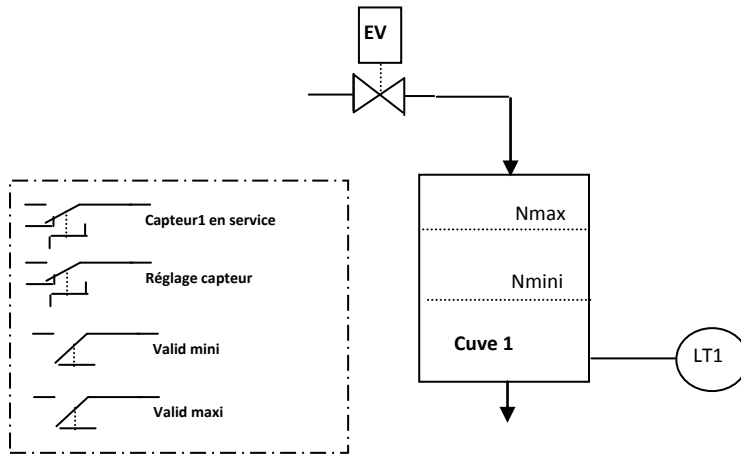
$$OUT = [ ((FLOAT (IN) - K1)/(K2-K1)) * (HI\_LIM-LO\_LIM)] + LO\_LIM$$

Pour une entrée unipolaire :K1 = 0.0 et K2 = +27648.0



**Exemple 3 : Réglage de l'étendue d'échelle d'un capteur**





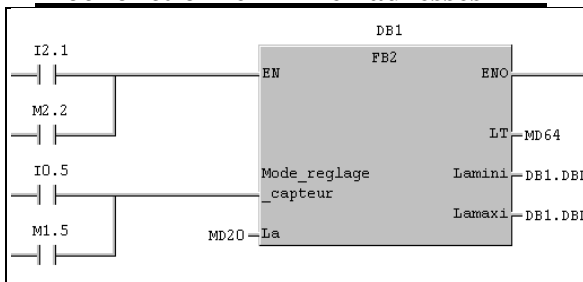
**Cahier des charges :**

- Régler l'étendue d'échelle du capteur
- Procédure réalisée par l'opérateur à l'aide du produit
- Stocker les valeurs maxi et mini de cette étendue
- Disposer de la valeur de la mesure en %

**Procédure avec produit dans la cuve :**

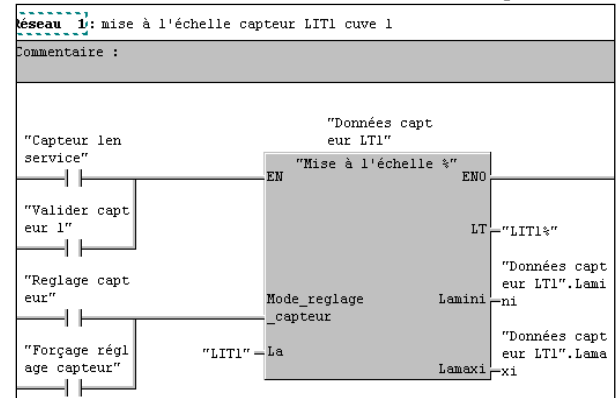
1. Sélectionner le capteur à mettre en service « Capteur1 en service »
2. Se positionner en mode réglage « Réglage capteur »
3. Amener le liquide au niveau mini de l'étendue d'échelle désirée
4. Valider ce niveau « Validation réglage mini »
5. Amener le liquide au niveau maxi de l'étendue d'échelle désirée
6. Valider ce niveau « Validation réglage maxi »
7. Quitter le mode réglage, la mesure en % est alors disponible.

**Bloc fonctionnel FB2 en adresses API**



Les entrées « Valider capteur 1 » et « Forçage réglage capteur » sont des variables associées à un 2<sup>e</sup> pupitre

**Bloc fonctionnel FB2 « Mise à l'échelle » en mnémoniques**

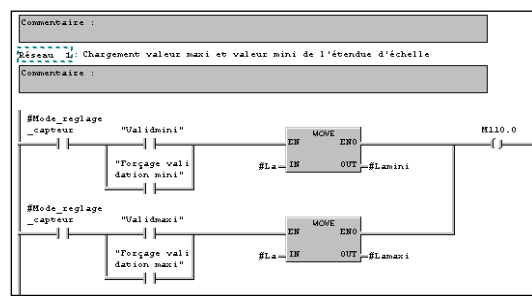


**Liste des variables formels associées au bloc fonctionnel « Mise à l'échelle » :**

Adresse	Décl.	Nom	Type	Valeur initiale	Commentaire
0.0	in	Mode_reglage_capteur	BOOL	FALSE	Mode correspondant à la saisie des valeurs extrêmes
2.0	in	La	REAL	0.000000e+000	Valeur numérique issue du capteur
6.0	out	LT	REAL	0.000000e+000	Valeur de la mesure en %
10.0	out	Lamini	REAL	0.000000e+000	Valeur maximale de la mesure
14.0	out	Lamaxi	REAL	0.000000e+000	Valeur minimale de la mesure

**Le contenu du bloc fonctionnel FB2 est le suivant :**

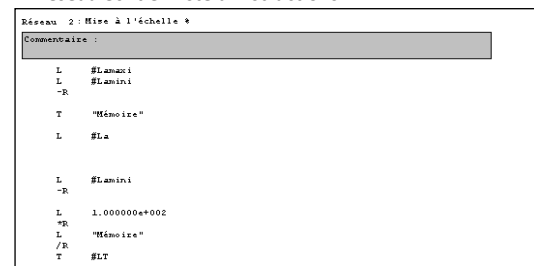
**1° Réseau écrit en ladder**



(Les entrées « Forçage validation mini » et « Forçage validation maxi » sont des

variables associées à un 2<sup>e</sup> pupitre opérateur)

**2° Réseau écrit en liste d'instructions**



## 4| STRUCTURATION D'UN PROGRAMME:

**Le diagramme SFC** (Sequentiel Function Chart) ou langage fonctionnel de séquences

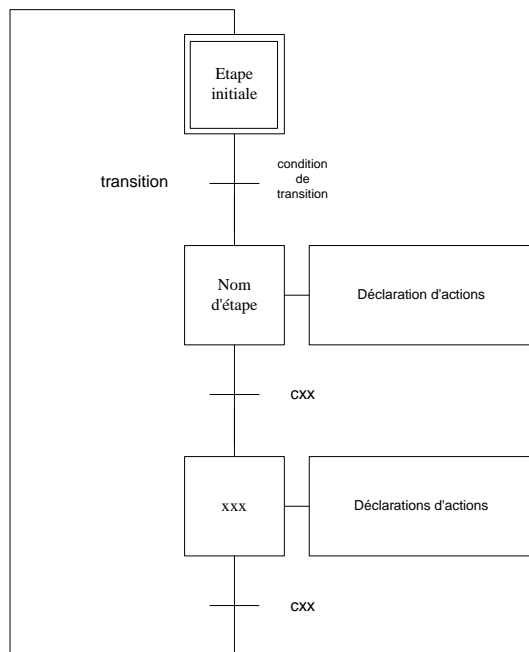
**Le diagramme fonctionnel de séquence SFC à ne pas confondre avec la description du comportement d'un système** (connu sous le nom de GRAFCET en France, voir norme **CEI 60848**).

Ce langage est destiné à la description de fonctions de commande séquentielles.

Le programme correspondant est constitué d'un ensemble d'**étapes** et de **transitions** reliés entre elles par des **liaisons dirigées**.

Chaque étape est associée à un ensemble d'**actions**.

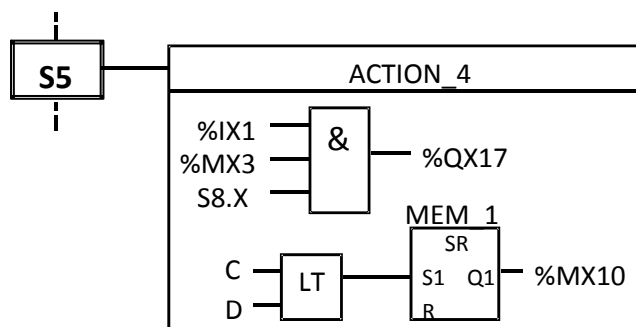
Chaque transition est associée à une **condition de transition**.



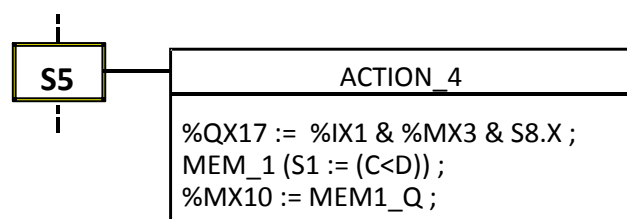
Le programme API traduisant cette représentation peut utiliser l'un des quatre langages de programmation.

### Exemple de la même déclaration d'actions sous différents langages:

Déclaration d'actions en langage FBD :

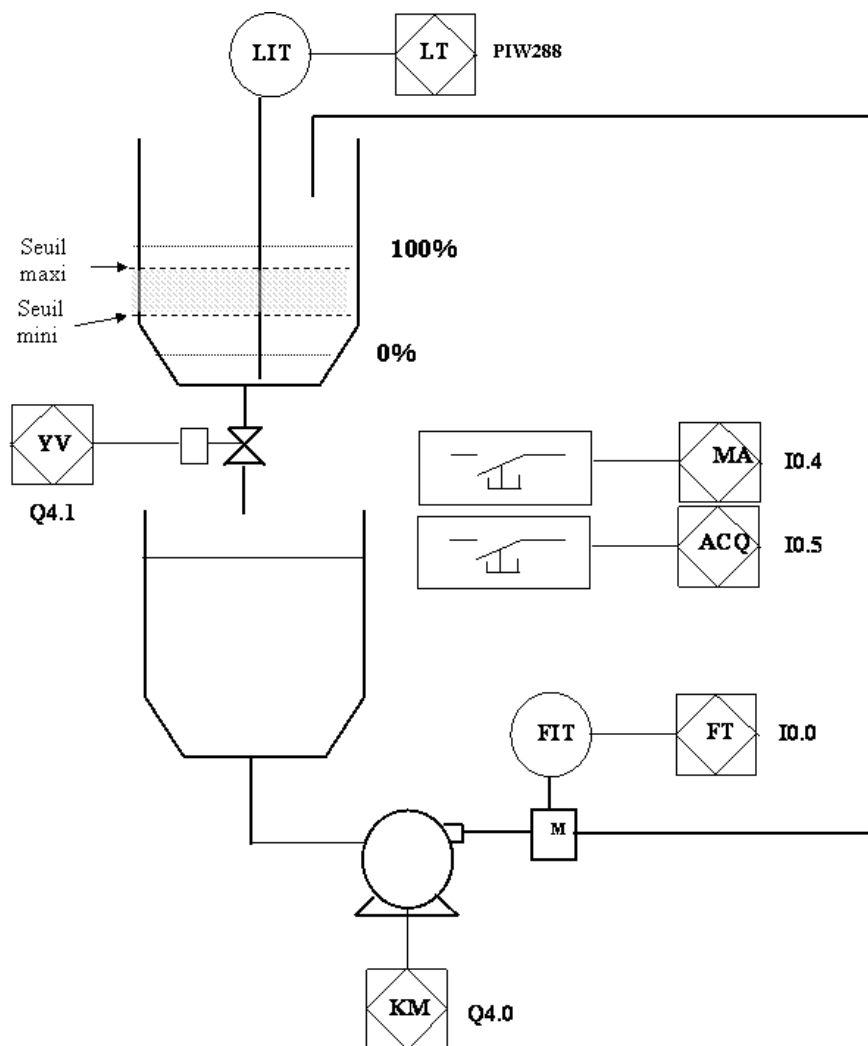


Déclaration d'actions en langage ST :



## Exemple de programme écrit en SFC : ici le logiciel STEP7-GGRAPH de Siemens

### Schéma TI de l'installation :



### Cahier des charges :

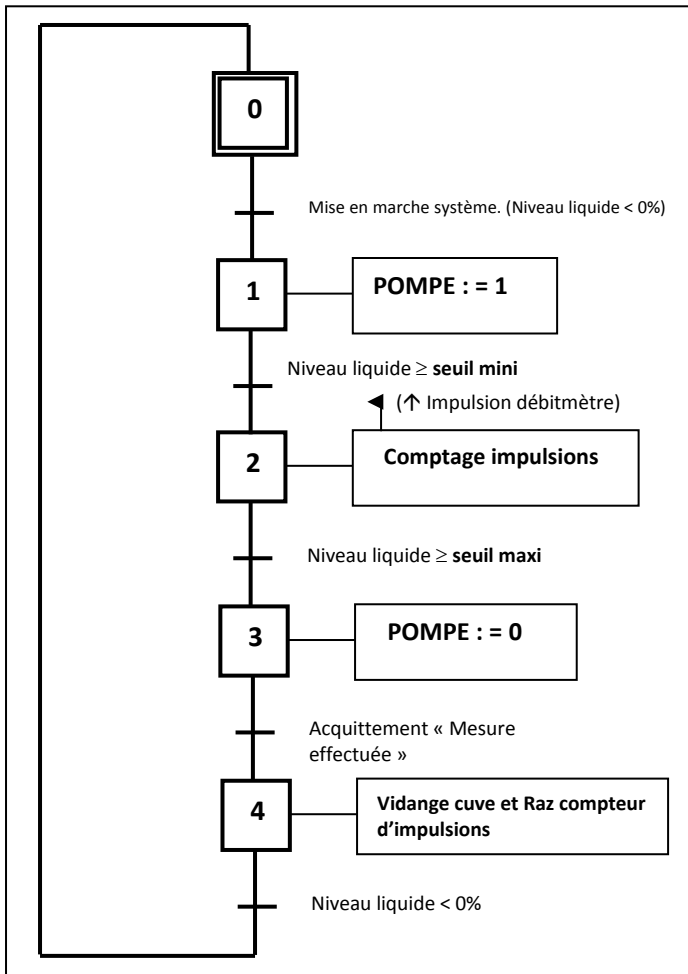
Vous disposez d'une cuve munie d'un capteur de niveau et d'un débitmètre à sortie à impulsions.

On vous demande de déterminer, de façon automatique et à l'aide d'un API, le volume d'eau compris entre les repères seuil mini et seuil maxi de la cuve

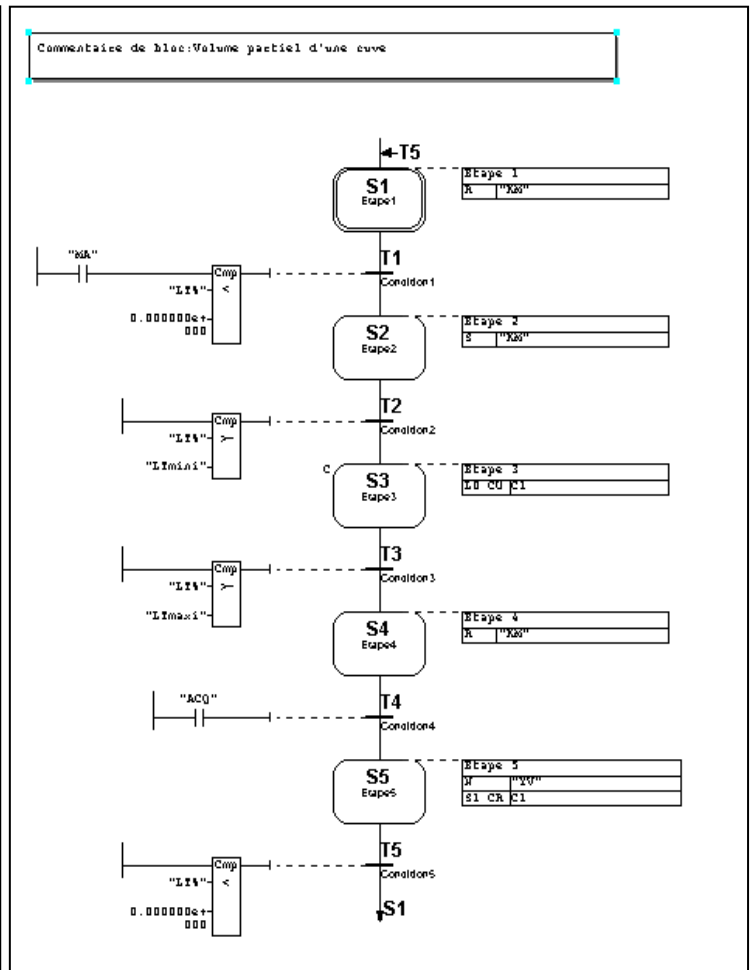
### Table de mnémoniques :

Mnémonique	Opérande	de donn	Commentaire		
ACQ	I	0.5	BOOL	BP acquitement	
FT	I	0.0	BOOL	Mesure de débit	
KM	Q	4.0	BOOL	Commande moteur pompe	
LT	PIW	288	WORD	Mesure de niveau 4-20 mA	
LT%	MD	3	REAL	Mesure de niveau en %	
MA	I	0.4	BOOL	BP marche	
SCALE	FC	105	FC	105	Scaling Values
YV	Q	4.1	BOOL	Commande vanne	
LTmini	MD	20	REAL	Niveau mini du volume partiel	
LTmaxi	MD	30	REAL	Niveau maxi du volume partiel	

## Grafcet de l'application :



## Grafcet en langage SFC de STEP7 :



## III] Premiers pas dans la programmation d'un automate Siemens en S7

# Sommaire

1-ADRESSAGE DES VARIABLES SIEMENS:.....	12
2- OUVERTURE ET ENREGISTREMENT D'UN PROJET EXISTANT : .....	13
3- VERIFICATION DE LA CONFIGURATION MATERIELLE DE L'API : .....	13
4- CREER UNE TABLE DE MNEMONIQUES : .....	15
5- PROGRAMMER UN RESEAU DANS UNE FONCTION FC1 ET L'APPELER DANS OB1 : .....	16
6- CREER UNE TABLE DE VARIABLES VAT1 : .....	17
7- TRANSFERER ET TESTER LE FONCTIONNEMENT DU PROGRAMME : .....	18
8- FORÇAGE DES VARIABLES : .....	19
9- REPRESENTATION D'UN GRAFCET: .....	20
10- MISE A L'ECHELLE D'UNE ENTREE ANALOGIQUE:.....	23
11- MISE A L'ECHELLE D'UNE SORTIE ANALOGIQUE:.....	25

### 1-Adressage des variables Siemens:

- Bits internes M0.0 à M255.7 **dépendants** des mots suivants
- Octets internes : ensemble de 8 bits MBi
- Mots internes : ensemble de 16 bits MWi
- Mots doubles : ensemble de 32 bits MDi

Principe de l'adressage des mots internes : (**attention aux chevauchements des mots**)

MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
MW6		MW4		MW2		MW0	
MD4				MD0			
MD2							

Poids fort et poids faible :

Mi.7	Mi.6	Mi.5	Mi.4	Mi.3	Mi.2	Mi.1	Mi.0
poids fort							poids faible
<b>MBi</b>							

MBi+1	MBi
poids faible	poids fort
<b>MWi</b>	

poids faible	poids fort
<b>MDi</b>	

MWi+1	MWi
-------	-----

## 2- Ouverture et enregistrement d'un projet existant :

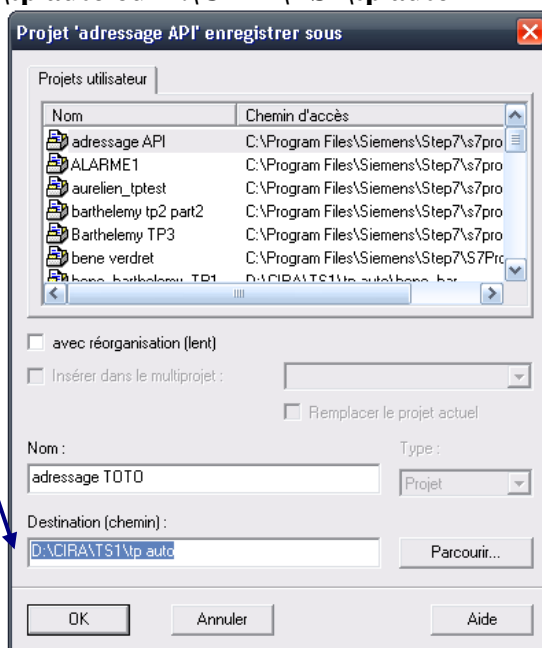
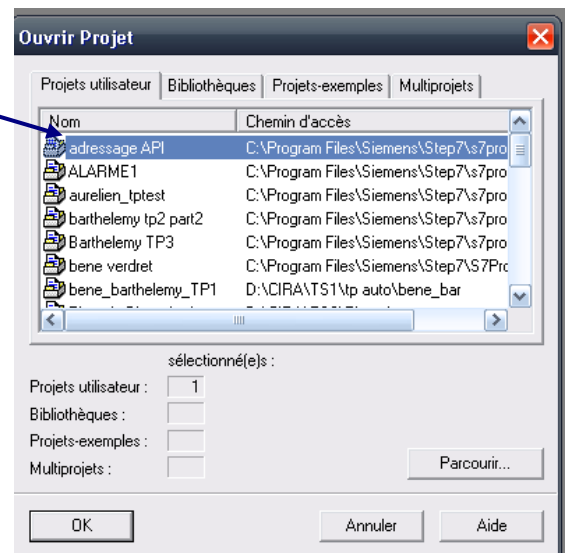
Dans le bureau de Windows, double-cliquer sur l'icône « SIMATIC Manager » :



Cliquer ensuite sur l'icône « ouvrir » :

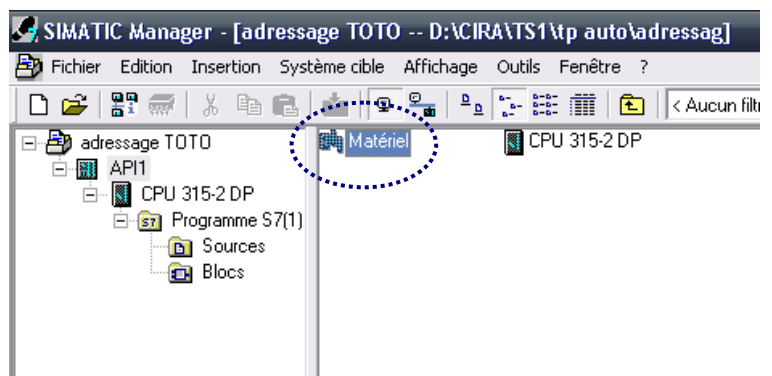


Choisir le nom du projet à ouvrir, par exemple :  
« **adressage API** » et enregistrer le sous un autre nom  
« **adressage TOTO** » dans le répertoire (avec **TOTO** votre nom):  
**D:\CIRA\TS1\tp auto** ou **D:\CIRA\TS2\tp auto**

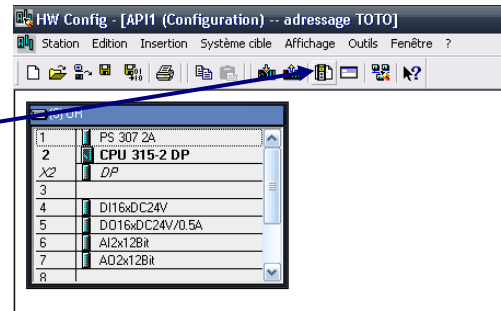


## 3- Vérification de la configuration matérielle de l'API :

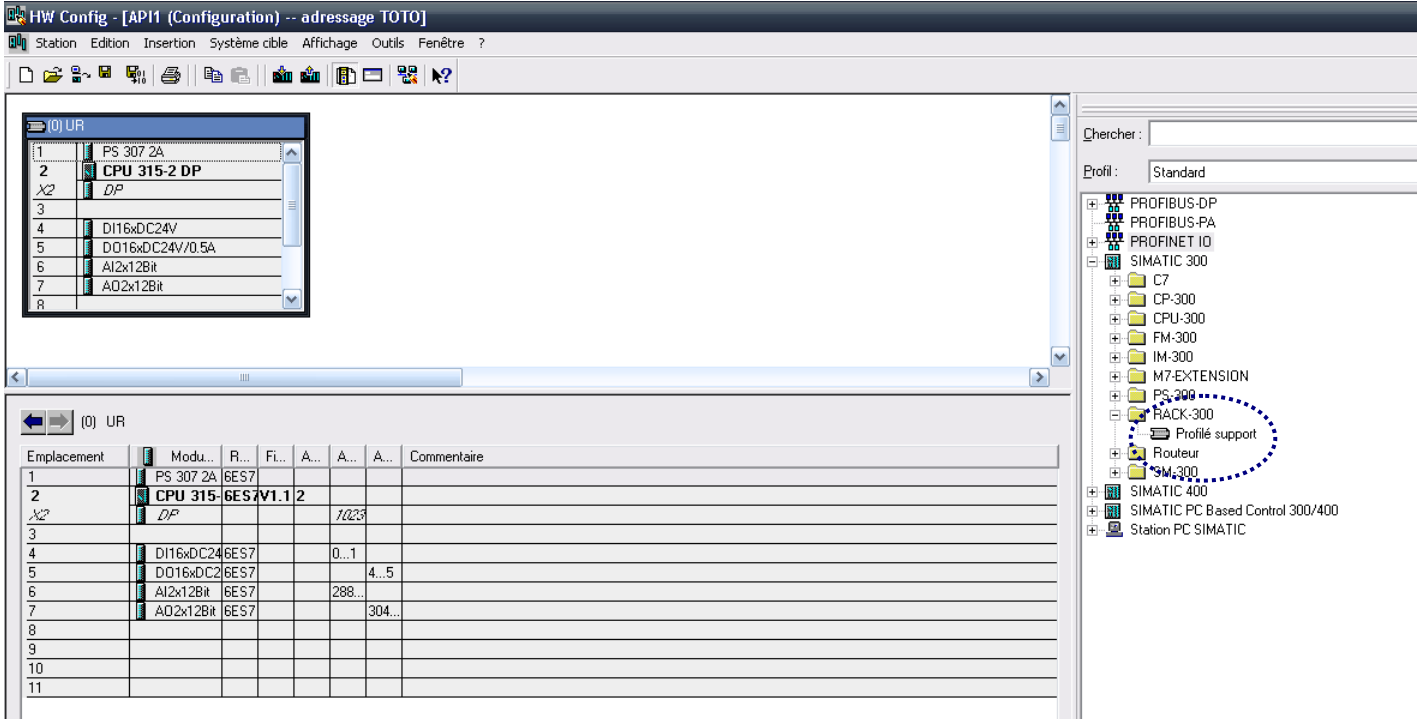
Afin de vérifier la configuration matérielle, double-cliquer sur l'icône « matériel » dans la partie droite de l'écran :



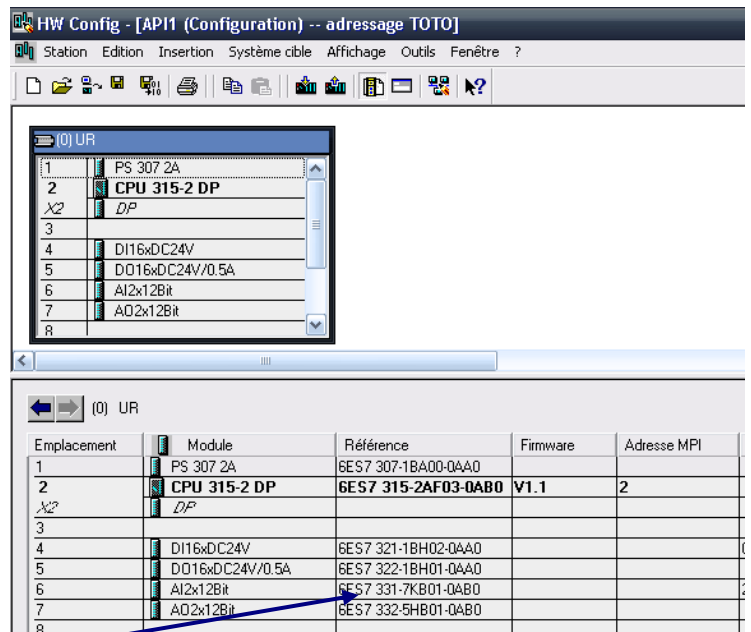
La nouvelle fenêtre qui s'ouvre va nous permettre de configurer l'automate. Si la liste des composants (sur la droite) n'apparaît pas, cliquer sur l'icône « catalogue » (ou menu « Affichage », puis cocher « catalogue ») :



La configuration matérielle est établie sur un « Profil support » issu d'un « Rack 300 » :

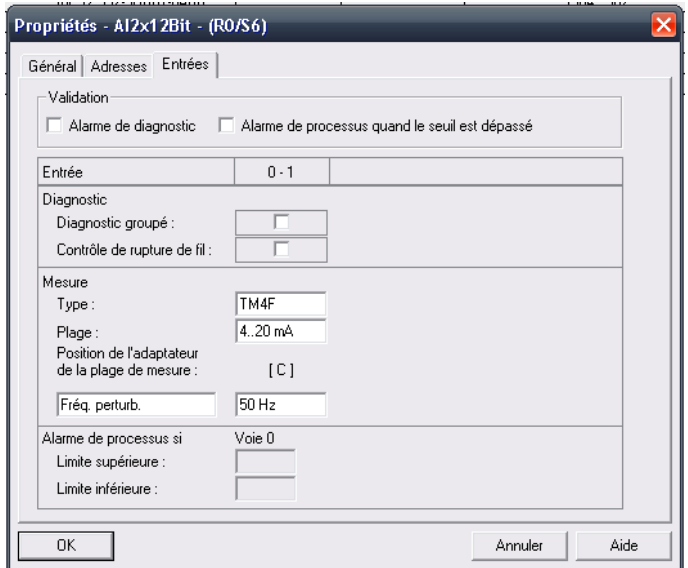


- Le module alimentation **PS 307 2A** (power supply) est placé sur le premier emplacement du rack .
- Le module Unité de Traitement **CPU 315 2 DP** occupe deux emplacements 2 et 3
- Le module d'entrée TOR ou DI (digital input) occupe l'emplacement 4. Ce module comporte **16 entrées TOR à 24VDC** (voir le schéma de câblage en ouvrant le volet du module sur l'API) Les entrées dont les adresses vont de **I0.0 à I0.7** sont câblées. Les entrées dont les adresses vont de **I1.0 à I1.7** sont en réserve.
- Le module de sorties TOR ou DO (digital output) occupe l'emplacement 5. Ce module comporte **16 sorties TOR à relais 24VDC -0,5A** (voir le schéma de câblage en ouvrant le volet du module sur l'API) Les sorties dont les adresses vont de **Q4.0 à Q4.7** sont câblées et relayées dans le coffret de connexion. Les sorties dont les adresses vont de **Q5.0 à Q5.7** sont en réserve.

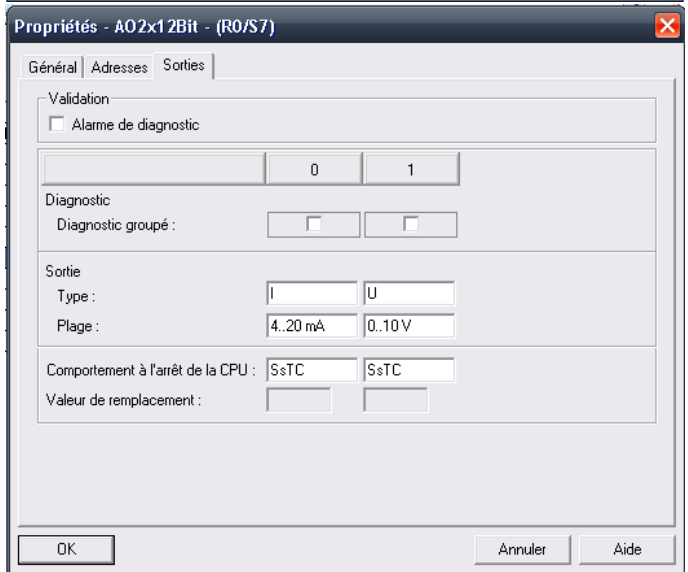


**Remarque importante** : bien vérifier que les références indiquées correspondent bien à celles en bas de chacun des modules.

- Le module d'entrées analogiques ou AI (analog input) occupe l'emplacement 6. Ce module comporte 2 entrées de 12 bits configurables, ici en 4-20 mA. Les adresses sont **PIW 288 et PIW290**. Un double clic sur l'emplacement 6 permet de vérifier leur configuration :



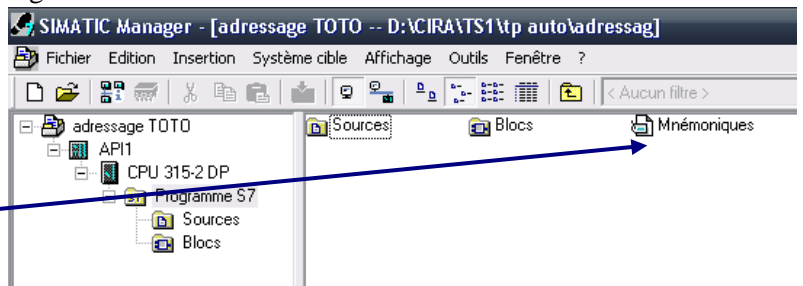
- Le module de sorties analogiques ou AO (analog output) occupe l'emplacement 7. Ce module comporte 2 sorties de 12 bits configurables, ici en 4-20 mA et 0-10 V. Les adresses sont **PQW 304 et PQW306**. Un double clic sur l'emplacement 7 permet de vérifier leur configuration :



Fermer la fenêtre en cliquant en haut à droite et enregistrer les modifications.

#### **4- Créer une table de mnémoniques :**

Développer « API1 », puis « CPU 315-2DP » et « Programme S7 ». Double-cliquer sur l'icône « mnémoniques » qui apparaît dans la partie droite de l'écran:



#### **Remplir la table après un double clic:**

La colonne « mnémonique » doit permettre d'identifier facilement la fonction de cette variable. Indiquer dans la colonne « opérande » l'adresse de la variable (I pour entrée, Q pour sortie...). Le logiciel donne le type de données dans la colonne suivante (ici « BOOL » pour booléen, c'est-à-dire binaire), mais celui-ci peut être modifié en cas de besoin.

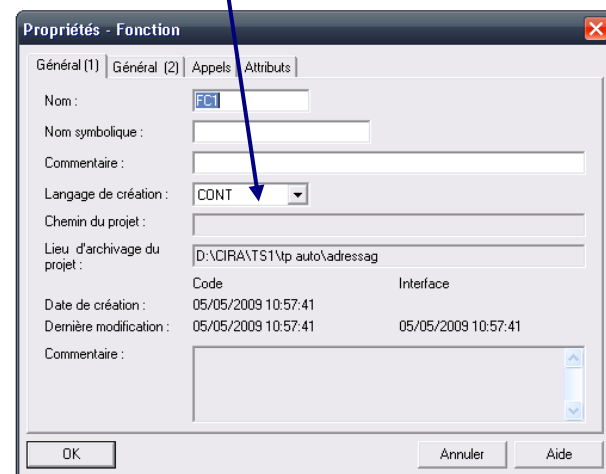
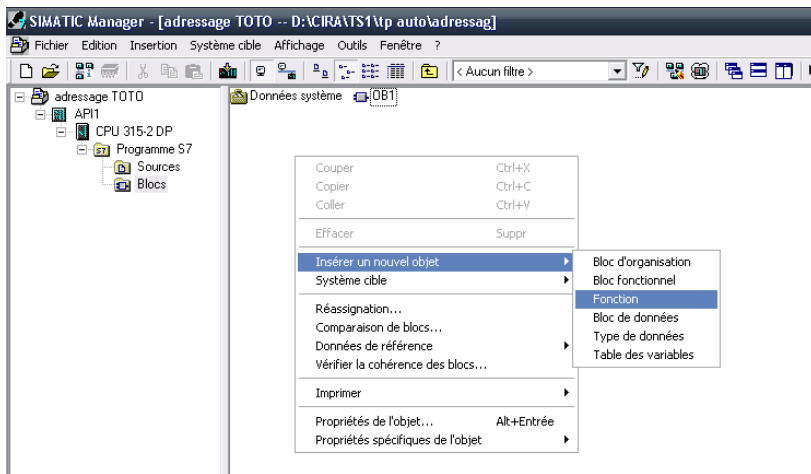
Programme S7 (Mnémoniques) -- adressage TOTO\API1\CPU 315-2 DP					
	Etat	Mnémonique	Opérande	Type de données	Commentaire
1		LL1	I 0.0	BOOL	Capteur à lames vibrantes de niveau bas de la cuve1
2		KM1	Q 4.0	BOOL	Contacteur moteur malaxeur
3		EV1	Q 4.1	BOOL	Electrovanne de remplissage de type NF
4					

Enregistrer la table et revenir à l'écran initial de « SIMATIC Manager ».

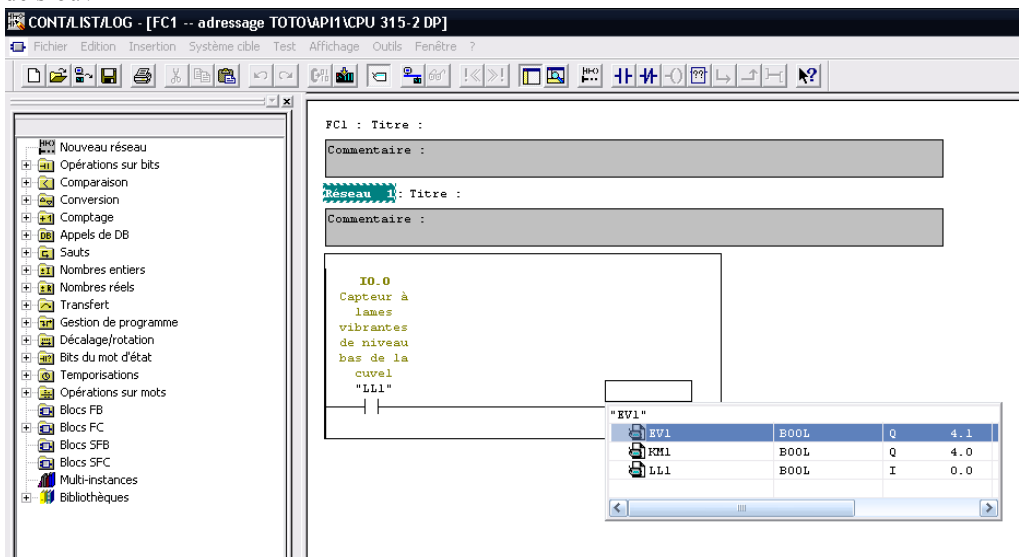


## 5- Programmer un réseau dans une fonction FC1 et l'appeler dans OB1 :

Cliquer dans la fenêtre de droite de « Programme S7 », puis cliquer sur « Insérer un nouvel objet », choisir « Fonction » en ladder ou langage contact.



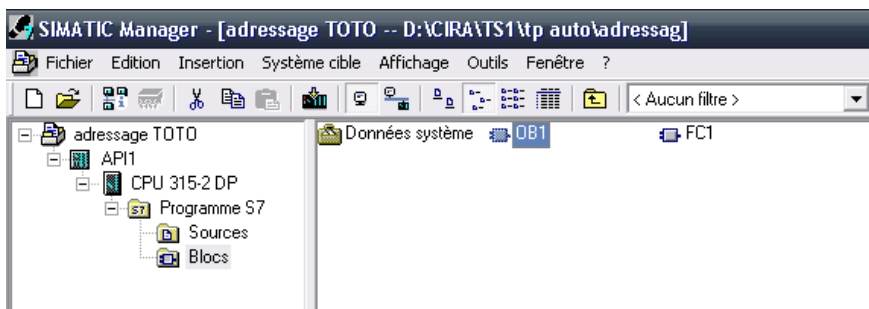
Double-cliquer sur « FC1 » : créer le réseau 1 correspondant à l'équation logique demandée dans la fenêtre qui vient de s'ouvrir



Sur les contacts, on peut indiquer l'adresse ou insérer un mnémorique et compléter la zone commentaire de ce réseau, afin d'en faciliter la compréhension.

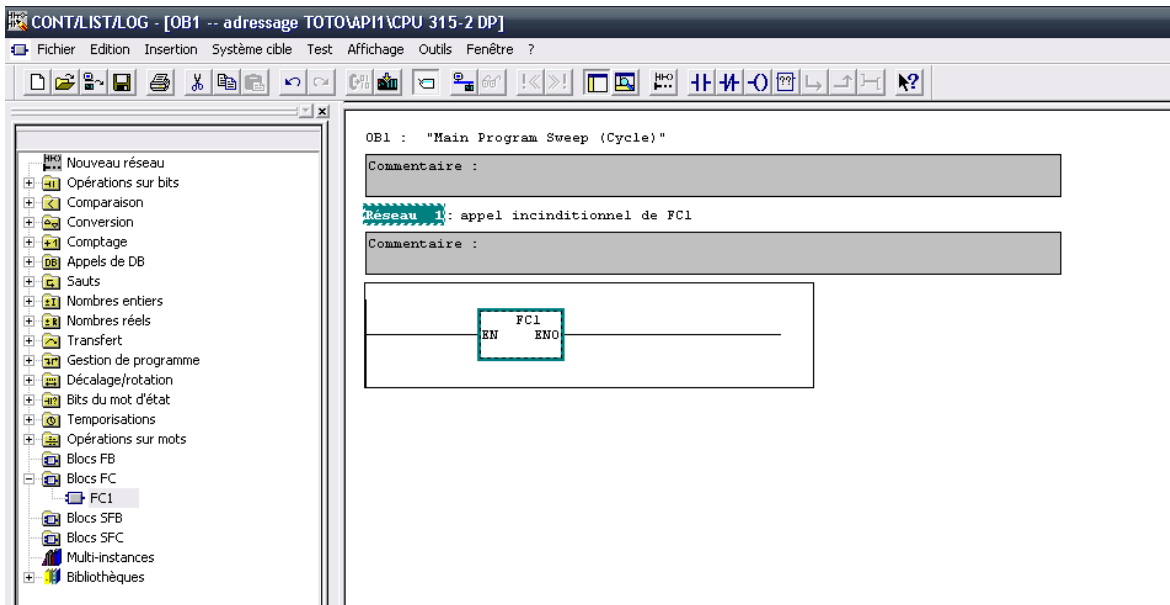
**Fermer et enregistrer « FC1 ».**

**Double-cliquer sur OB1 :**



OB1 est un bloc d'organisation, il permet d'autoriser le fonctionnement des autres blocs.

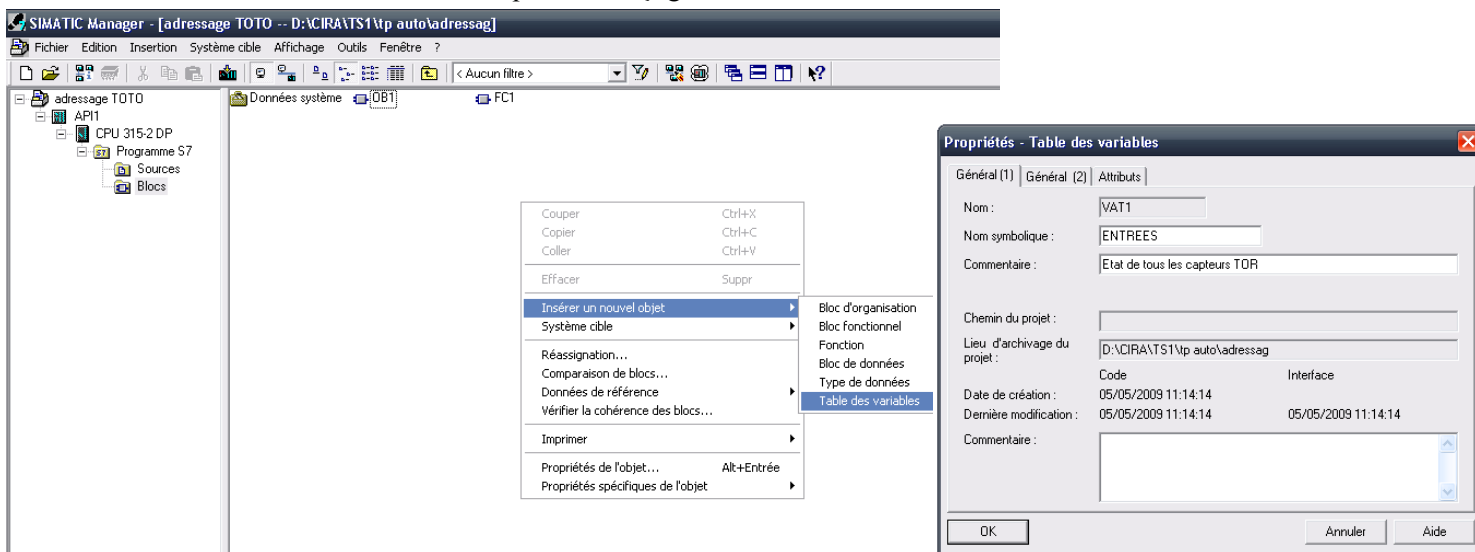
Dans la fenêtre qui s'ouvre, faire apparaître le catalogue, développer « Blocs FC » et double-cliquer sur « FC1 » afin de l'appeler inconditionnellement.



Fermer et enregistrer « OB1 ».

## 6- Créer une table de variables VAT1 :

Il est possible de regrouper par thème une partie des variables d'un programme, dans une table de variables VAT1 nommée ici « ENTREES ». Lors de la visualisation du programme cette table permettra d'afficher la valeur des variables ainsi que leur forçage.



Var - ENTREES

Table Edition Insertion Système cible Variable Affichage Outils Fenêtre ?

ENTREES -- adressage TOTO\API1\CPU 315-2 DP\Programme S7

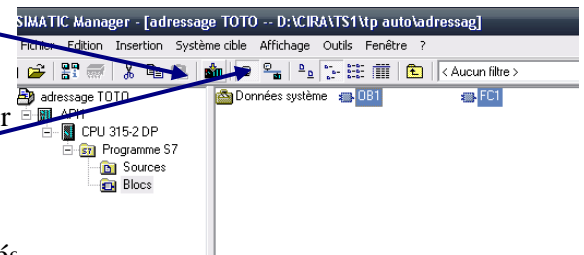
	Opérande	Mnémonique	Format d'affichage	Valeur d'état	Valeur de forçage
1	I	O.0	"LL1"	BOOLEEN	
2					

Fermer et enregistrer « VAT1 ».

## 7- Transférer et tester le fonctionnement du programme :

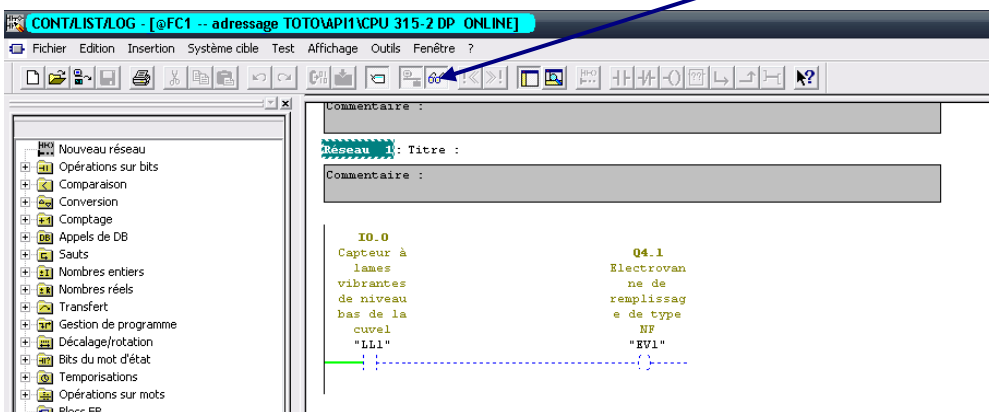
Une fois le programme sauvegardé, il est nécessaire de le **transférer** à la CPU par l'intermédiaire du câble PG/PC. Sélectionner l'ensemble des blocs à transférer, ici FC1 et OB1, **par la suite ne transférer que les blocs modifiés**. Puis confirmer le transfert de tous les blocs sur la fenêtre suivante.

**Vérifier la position de la clef de l'API**

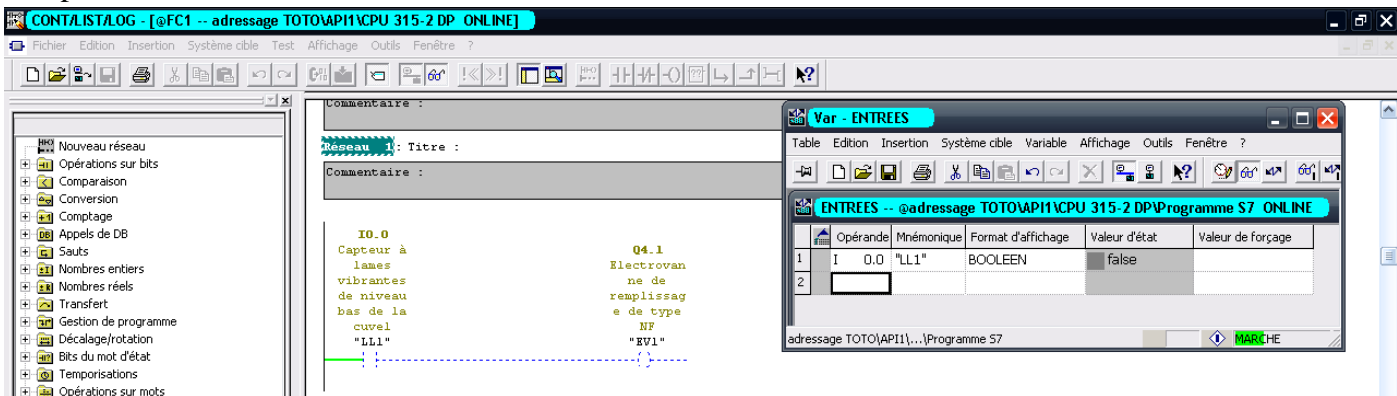


Se **mettre en ligne** et vérifier la présence des blocs transférés en plus des blocs systèmes existants SF..

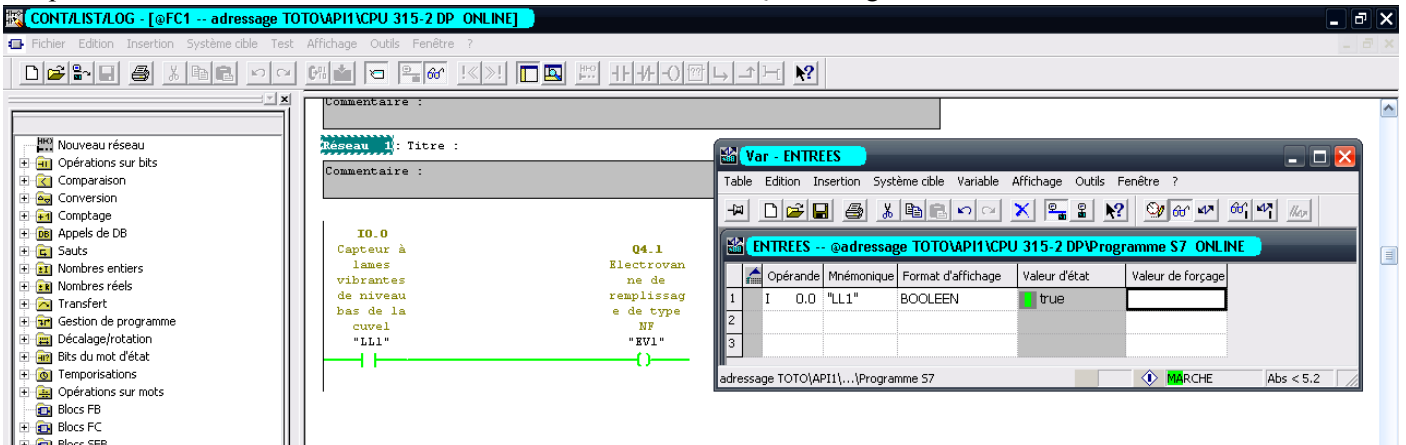
Double cliquer sur le bloc FC1 puis sélectionner l'icône des « **lunettes** » afin de visualiser l'état dynamique du bloc.



Il est possible de visualiser simultanément la table de variables VAT1 :



Après activation de l'entrée I0.0 à l'aide d'un câble, la sortie Q4.0 est également activée :



## 8- Forçage des variables :

Afin de tester un réseau, il est possible de **forcer l'état de l'entrée I0.0**, indiquer 1 dans la valeur de forçage, celle ci sera traduite par « true ». **Confirmer le mode de déclenchement du forçage à chaque cycle**

The screenshot shows the SIMATIC Manager interface. The main window displays a ladder logic diagram with a variable declaration for 'I0.0' and 'Q4.1'. The 'Var - ENTREES' window is open, showing a table of variables:

Opérande	Mnémonique	Format d'affichage	Valeur d'état	Valeur de forçage
1	I 0.0	"LL1"	BOOLEEN	false
2				
3				

The 'Déclenchement' dialog box is open, showing options for 'Point de déclenchement' and 'Condition de déclenchement'. The 'Condition de déclenchement' is set to 'Cyclique'.

Activer l'icône « forcer la variable », un message d'alerte, confirmer le pour poursuivre

The screenshot shows the SIMATIC Manager interface. The main window displays a ladder logic diagram with a variable declaration for 'I0.0' and 'Q4.1'. The 'Var - ENTREES' window is open, showing a table of variables:

Opérande	Mnémonique	Format d'affichage	Valeur d'état	Valeur de forçage
1	I 0.0	"LL1"	BOOLEEN	false
2				
3				

The 'Forçage (1491:5021)' dialog box is open, showing a warning message: 'Un fois la fonction de forçage lancée, les valeurs seront forcées à chaque cycle. Voulez-vous poursuivre le forçage?'.

Vous obtenez la fenêtre ci contre, la sortie Q4.0 est activée, conformément au réseau.

The screenshot shows the SIMATIC Manager interface. The main window displays a ladder logic diagram with a variable declaration for 'I0.0' and 'Q4.1'. The 'Var - ENTREES' window is open, showing a table of variables:

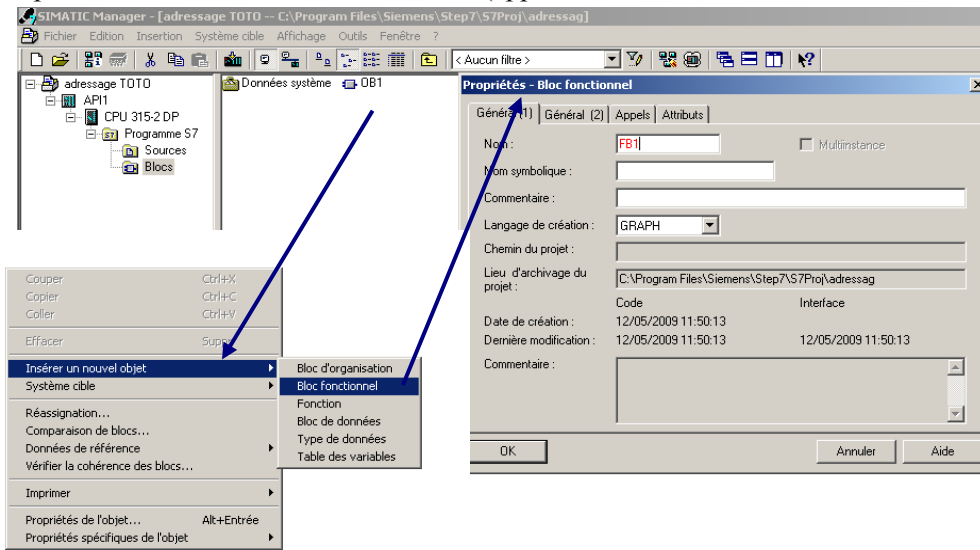
Opérande	Mnémonique	Format d'affichage	Valeur d'état	Valeur de forçage
1	I 0.0	"LL1"	BOOLEEN	true
2				
3				

The 'Var - ENTREES' window shows the 'Valeur de forçage' for 'I 0.0' set to 'true'.

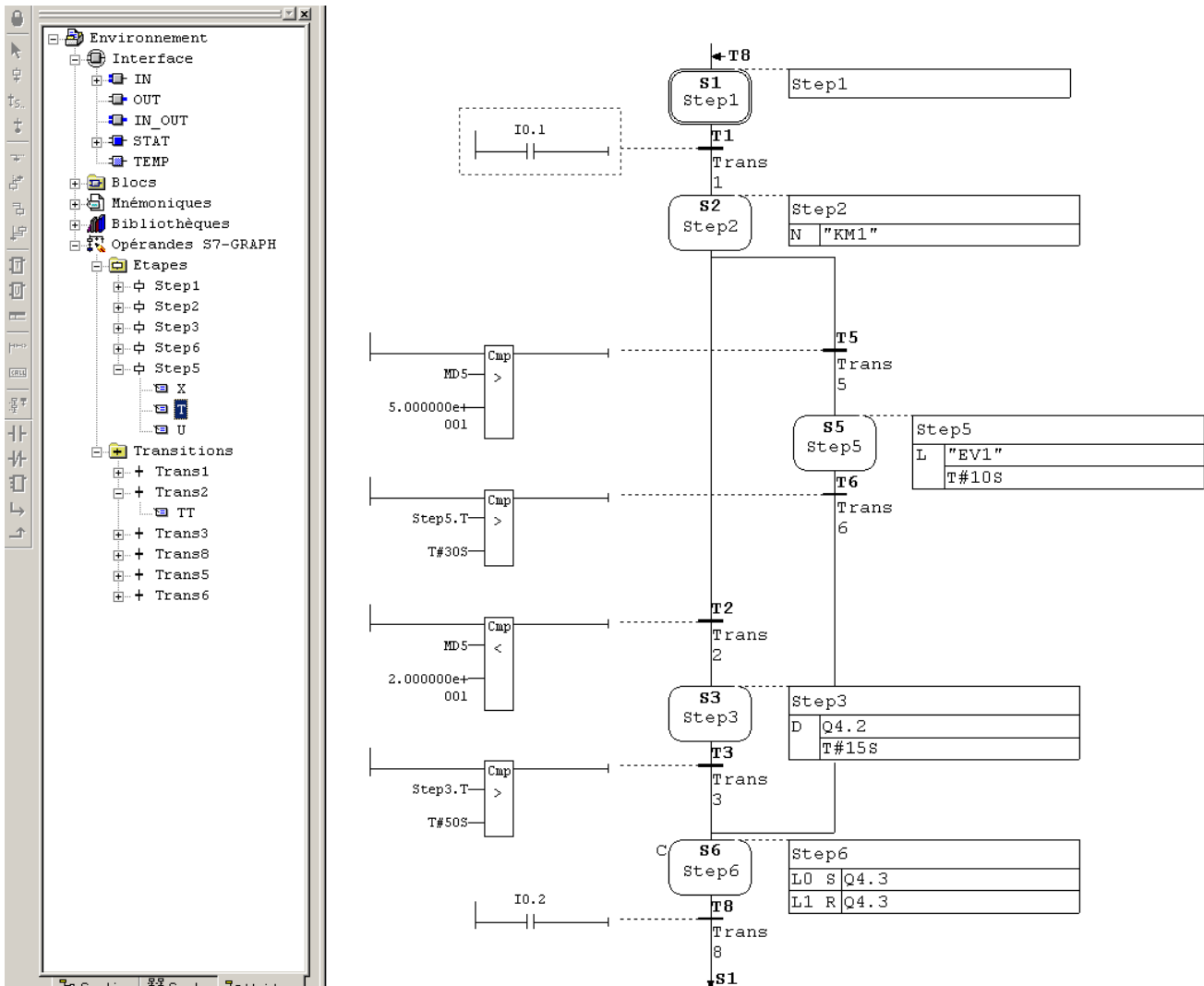
## 9- Représentation d'un grafcet:

La programmation d'une séquence séquentiel à partir d'un grafcet de commande est possible en faisant appel à un bloc fonctionnel FB associé à un bloc d'instance DB.

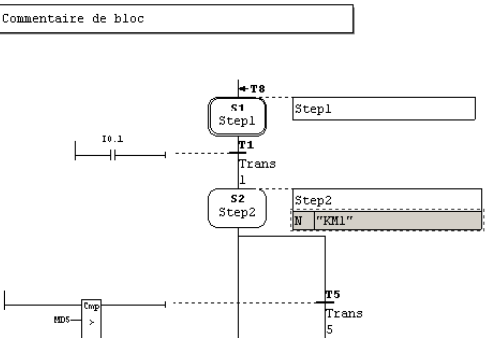
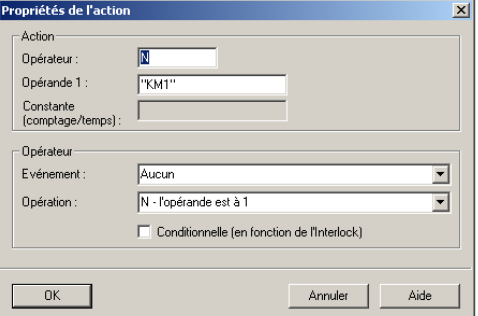
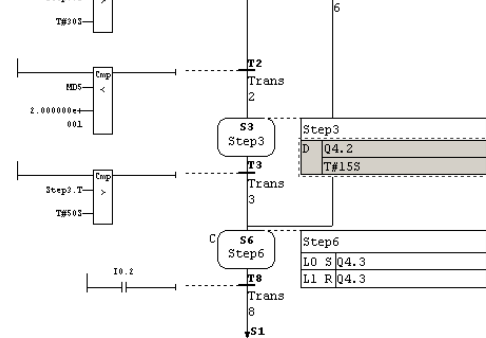
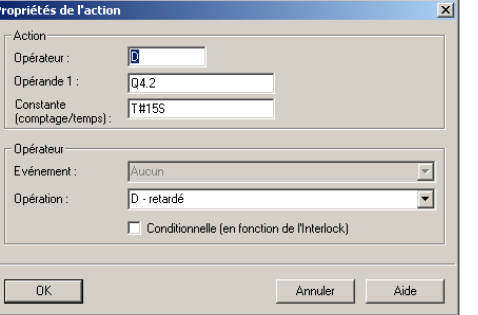
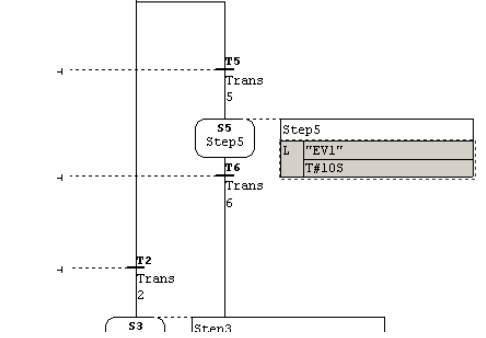
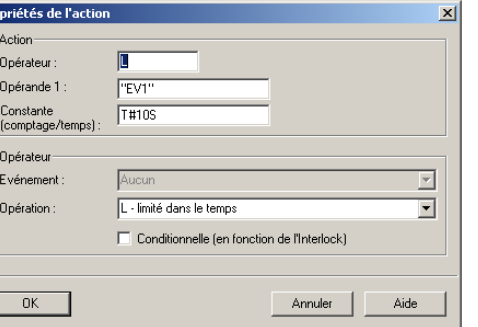
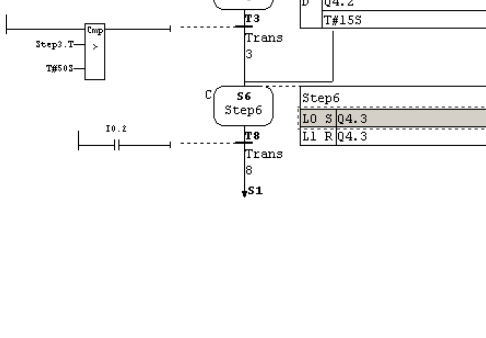
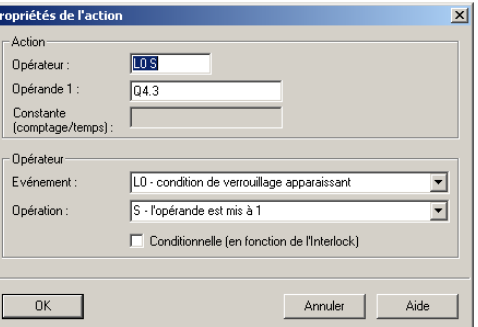
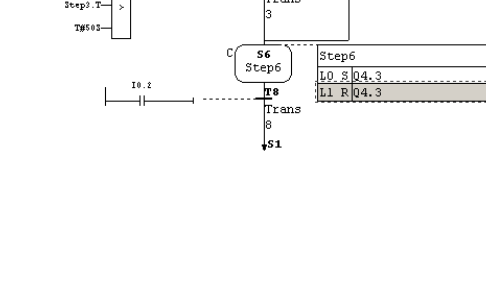
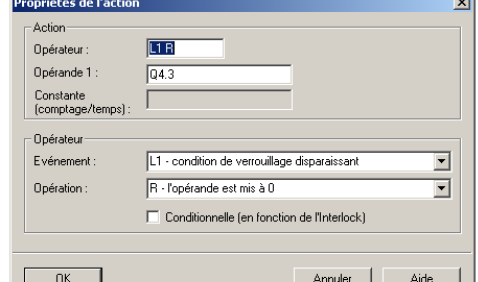
Dans l'exemple suivant nous utilisons le bloc FB1 (appelé « alarme ») associé à DB1.



On réalise d'un double clic sur le bloc FB1, afin de programmer le grafcet :



Exemples de programmation détaillée des actions :

Type d'action	Représentation	
<p><b>Action « normale »</b></p>		
<p><b>Action « retardée de 15s »</b></p>		
<p><b>Action « limitée de 10s »</b></p>		
<p><b>Action « conditionnelle sur front montant »</b></p>		
<p><b>Action « conditionnelle sur front descendant »</b></p>		

## Représentation détaillée d'une étape :

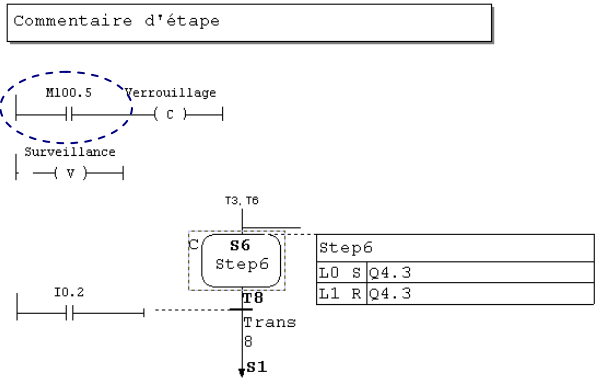
La condition étant le memento de cadence M100.5

Celui-ci doit être configuré au préalable dans la configuration matérielle (HW Config) de la CPU :

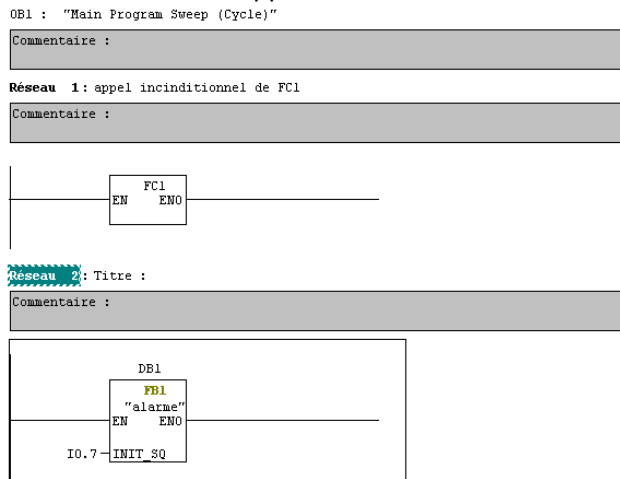
The screenshot shows the HW Config interface for a CPU 315-2 DP. The 'Cycle/Mémoire de cadence' tab is active, showing the 'Mémoire de cadence' section with the 'Memento de cadence' checkbox checked and the 'Octet de memento' set to '100'. A blue dashed circle highlights the '100' value. To the right, a help window titled 'Aide sur les unités centrales et FM spécifiques' displays a table for 'Durée de période' with bit 5 circled in blue.

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

La représentation détaillée de l'étape sera la suivante ( la condition de verrouillage correspond à la condition associée à l'action):



Finalement le bloc FB1 associé à DB1 sera appelé inconditionnellement dans le réseau 2 de OB1



## 10- Variables associées aux étapes du grafcet:

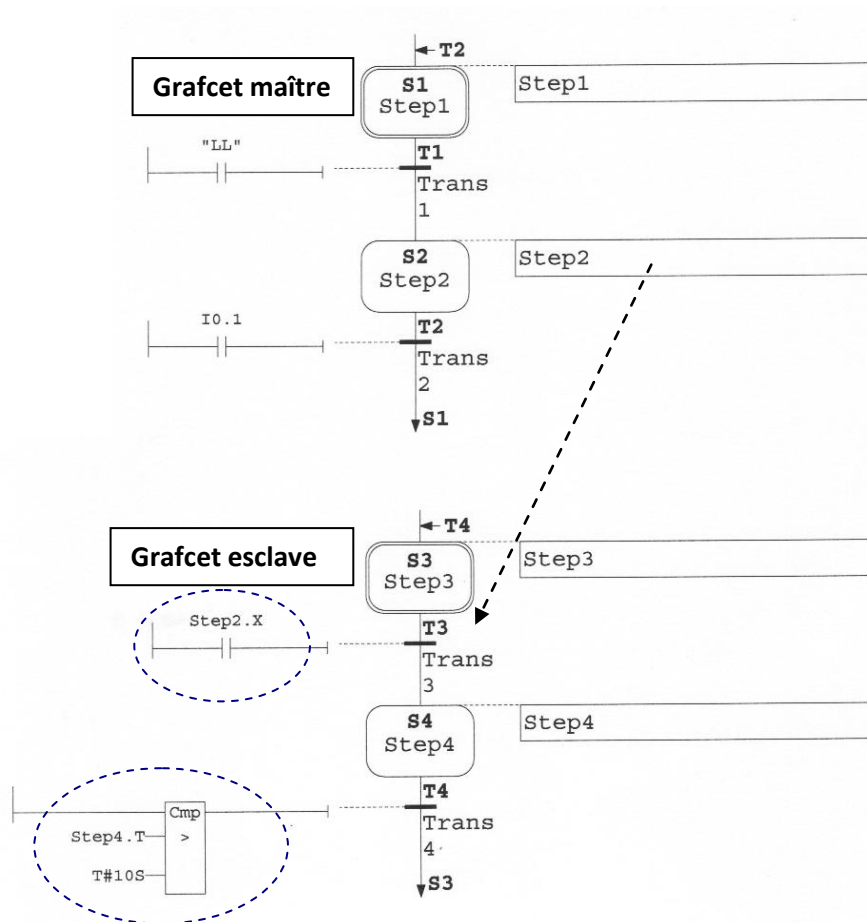
### Opérandes spécifiques à S7-GRAPH dans les conditions

Certaines informations système relatives aux étapes peuvent être utilisées comme opérandes dans les transitions, les surveillances (Supervision), les verrouillages (Interlock), les actions et les conditions permanentes.

Opérande	Signification	Emploi dans
Si.T	Temps d'activation en cours ou dernier temps d'activation de l'étape i	Comparateur, affectation
Si.U	Temps d'activation de l'étape i sans compter le temps de défaut	Comparateur, affectation
Si.X	Indique si l'étape i est active	Contact à fermeture/à ouverture
Transi.TT	Transition i est vraie Vérification si toutes les réceptivités de la transition sont vraies	Contact à fermeture/à ouverture

Vous pouvez utiliser les opérandes spécifiques à S7-GRAPH comme tous les autres opérandes dans CONT, LOG et dans les actions.

Exemple d'utilisation de ces opérandes :





## 11- Mise à l'échelle d'une entrée analogique:

### Exercice d'application : CAHIER DES CHARGES D'UNE ALARME

Surveillance d'un niveau haut sur une cuve de dimension 0 à 10,0 m.

Alarme niveau HH enclenchée si niveau de 850 cm dépassé et effacée si niveau haut inférieur à 830 cm.

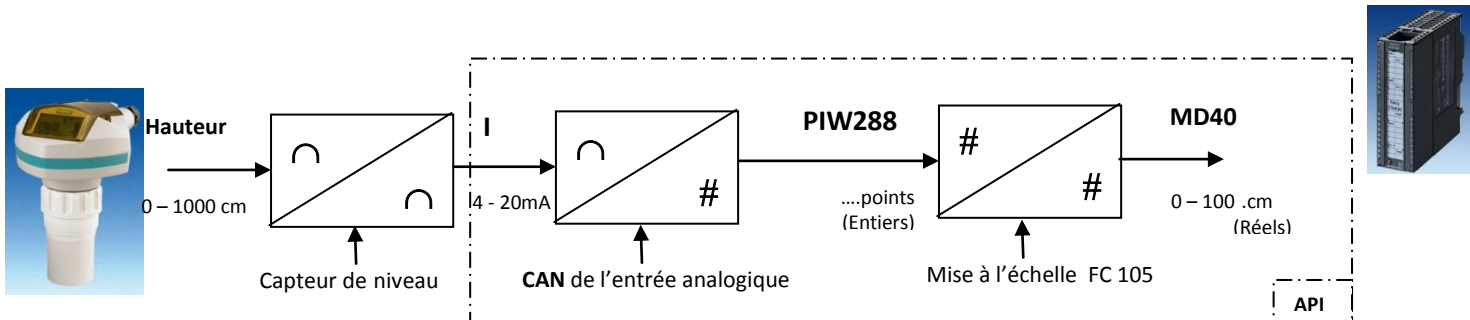
Caractéristiques du capteur de niveau : sortie 4-20 mA, étendue d'échelle de 0 à 1000 cm.

Caractéristiques de l'entrée analogique API : entrée 4-20 mA, conversion numérique sur 12 bits.

### Configuration de l'entrée analogique d'adresse PIW288:

Emplacement	Module	Référence	Fi...	A...	Adresse d'entrée	A...	Commentaire
1	PS 307 2A	6ES7 307-1BA00-0AA0					
2	CPU 315-2 DP	6ES7 315-2AF03-0AB0	V1.1	2			
3	DP				1/23°		
4	DI16xDC24V	6ES7 321-1BH02-0AA0				0...1	
5	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0					4...5
6	AI2x12Bit	6ES7 331-7K01-0AB0			288...291		
7	AO2x12Bit	6ES7 332-5HB01-0AB0				304...	
8							
9							
10							
11							

### SCHEMA DE LA CHAINE DE MESURE



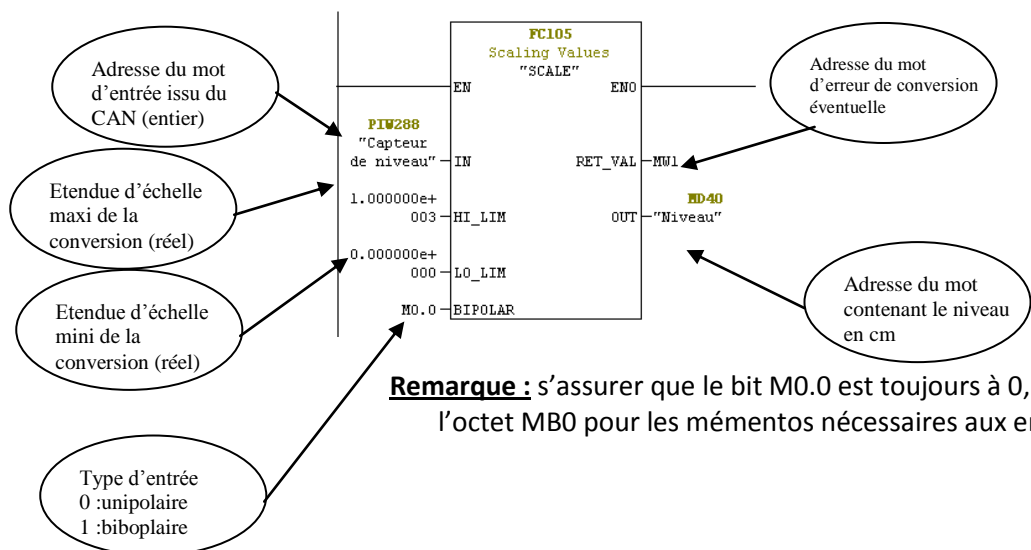
### Programme de mise à l'échelle en cm :

(Utiliser le bloc fonctionnel FC105 dans la bibliothèque Standard librairie-TI/S7)

```

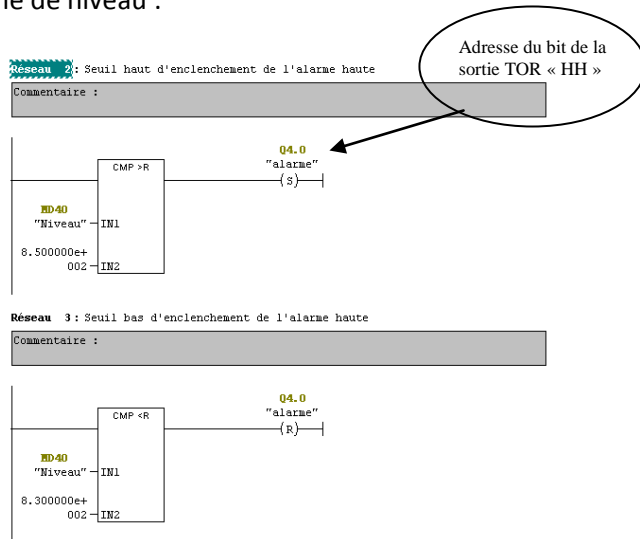
OB1 : Alarme sur niveau
Commentaire :

Réseau 1: Mise à l'échelle du capteur de niveau en cm.
Commentaire :
    
```



**Remarque :** s'assurer que le bit M0.0 est toujours à 0, pour cela réserver l'octet MBO pour les mémotos nécessaires aux entrées analogiques

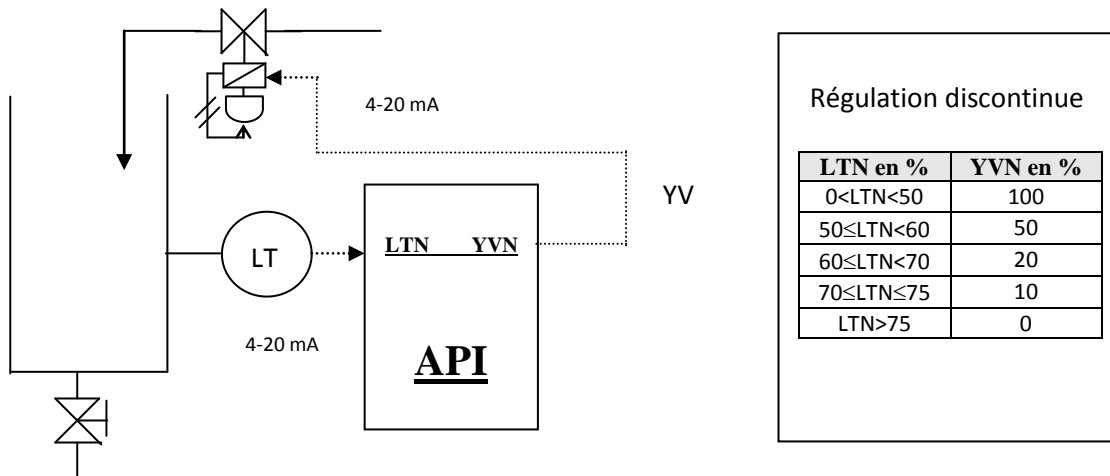
Programme de l'alarme de niveau :



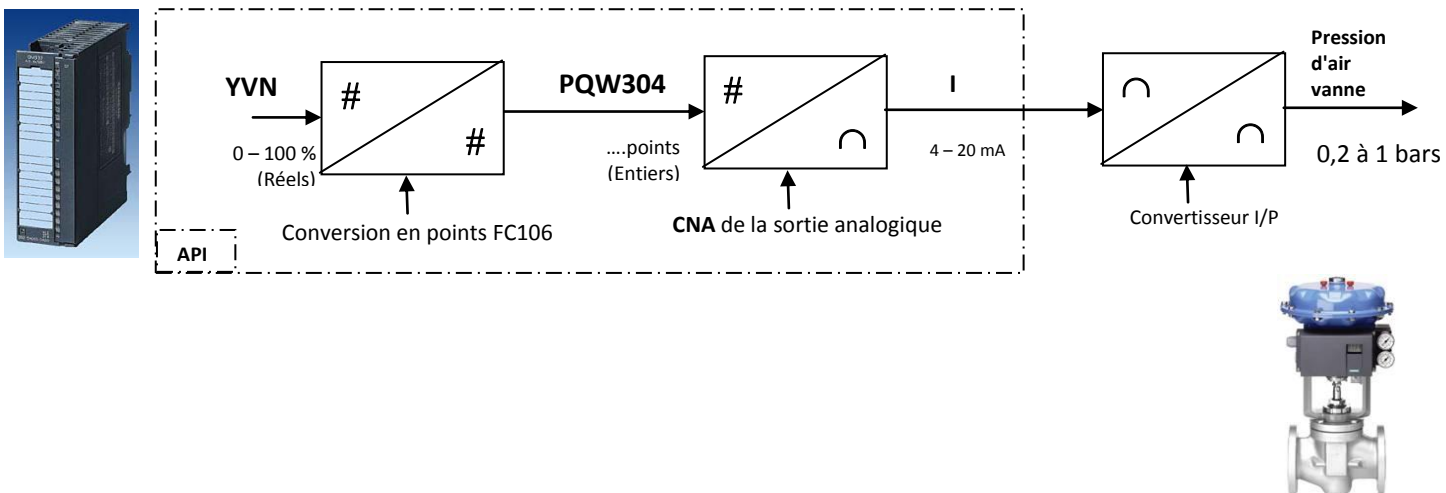
**12- Mise à l'échelle d'une sortie analogique:**

**Exercice d'application : CAHIER DES CHARGES D'UNE REGULATION DISCONTINUE**

On désire réaliser une régulation discontinue de niveau autour d'un point de consigne de 75% et suivant le tableau de commande de la vanne de type NF fourni ci-dessous.  
 La sortie analogique commandant le convertisseur I/P de la vanne est du type 4-20 mA avec une conversion numérique sur 10 bits.



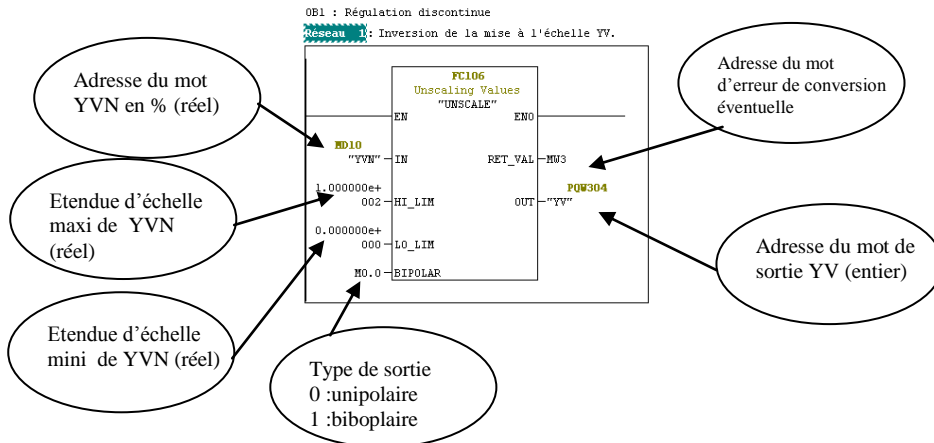
**SCHEMA DE LA CHAINE D'ACTION :**



## Configuration de la sortie analogique d'adresse PQW304:

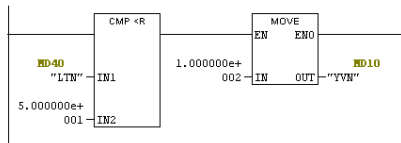
Emplacement	Module	Référence	Firmware	Adresse MPI	Adresse d'entrée	Adresse de s...	Commentaire
1	FS 307 2A	6ES7 307-1BA00-0AA0					
2	CPU 315-2 DP	6ES7 315-2AF03-0AB0	V1.1	2			
3	DP				1023"		
4	DI16xDC24V	6ES7 321-1BH02-0AA0			0...1		
5	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0				4...5	
6	AI2x12Bit	6ES7 331-7K01-0AB0			288...291		
7	AO2x12Bit	6ES7 332-5H01-0AB0				304...307	
8							
9							
10							
11							

## Programme d'inversion de la mise à l'échelle (utiliser le bloc fonctionnel FC106)

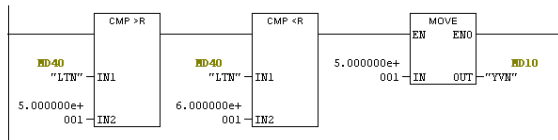


## Programme de commande du convertisseur de la vanne YV:

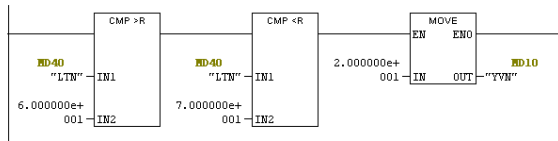
Réseau 2: Premier seuil de régulation



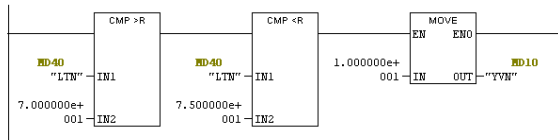
Réseau 3: Deuxième seuil de régulation



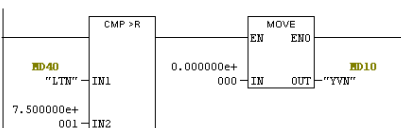
Réseau 4: Troisième seuil de régulation



Réseau 5: Quatrième seuil de régulation

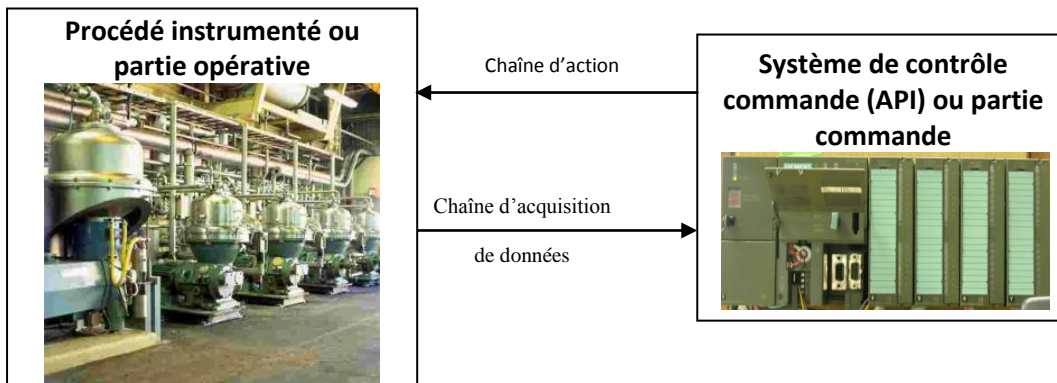


Réseau 6: Cinquième seuil de régulation

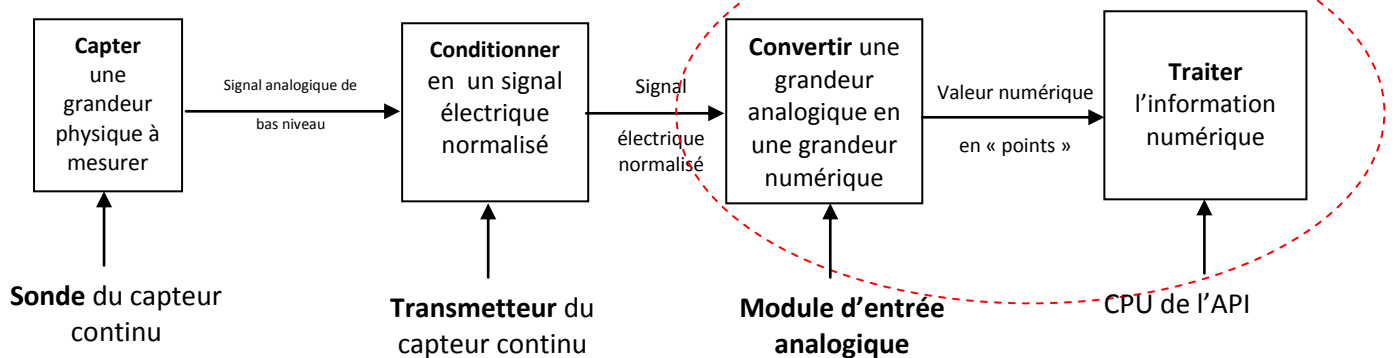


### III] Utilisation des entrées et sorties analogiques API

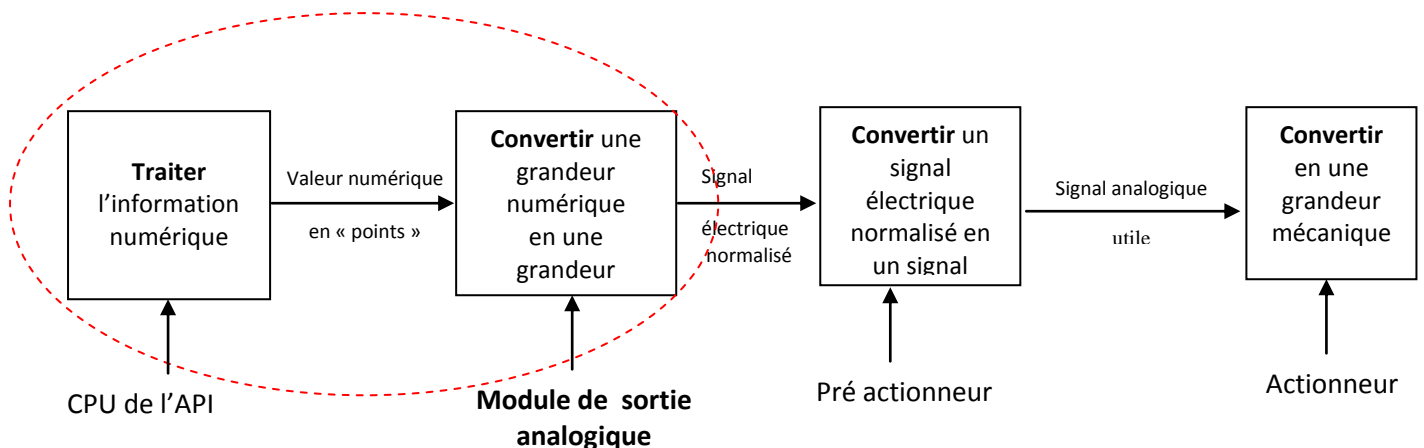
Ce document précise la mise en œuvre des entrées et sorties analogiques d'un API entrant dans la composition des chaînes d'acquisition de données et des chaînes d'action. Ces chaînes forment « les liens » entre un procédé instrumenté et son système de contrôle commande.



#### Chaîne d'acquisition des données d'un API



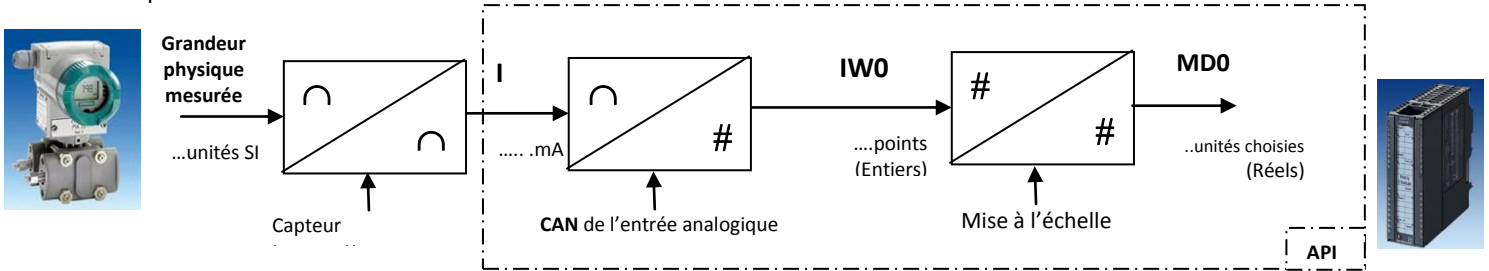
#### Chaîne d'action d'un API



## II Entrées analogiques API

**Schéma bloc type d'une chaîne d'acquisition** pour une entrée courant unipolaire 4 à 20 mA et une conversion numérique sur 12 bits (sans bit de signe).

**Exemple** : compléter ce schéma pour un capteur de pression d'étendue 0-1bar, sortie 4-20 mA et représentation en mbar sur API.



$IWO_{maxi} = 2^n - 1$  avec n bits pour la conversion numérique sans bit de signe dans notre exemple à 20 mA correspond :  $2^{12} - 1 = 4095$

Tableau des variables associées:

Grandeur physique en bar	0	0,25	0,50	0,75	1,00
I mesuré (mA)					
IWO (points)					
MD0 (réels) unités ou représentation choisies					

**Equation** caractéristique de la fonction de mise à l'échelle  $MD0 = f(IWO)$  :

$MD0 = IWO \times \dots\dots\dots$

**Caractéristiques** associées (à compléter)

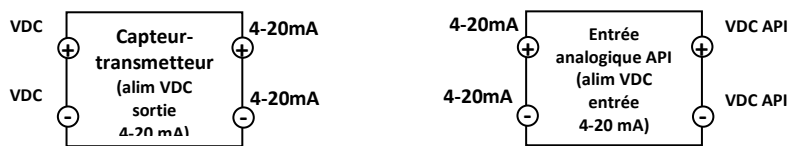
Du convertisseur analogique- numérique  $IWO = f(I)$

Fonction de mise à l'échelle (scale)  $MD0 = f(IWO)$

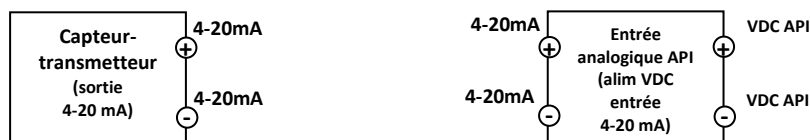


**Câblages électriques** à compléter suivant la technologie du capteur et de l'entrée API :

Capteur transmetteur « 4 fils » et entrée analogique « 4 fils » API



Capteur transmetteur « 2 fils » et entrée analogique « 4 fils » API



### Exercice d'application : CAHIER DES CHARGES D'UNE ALARME

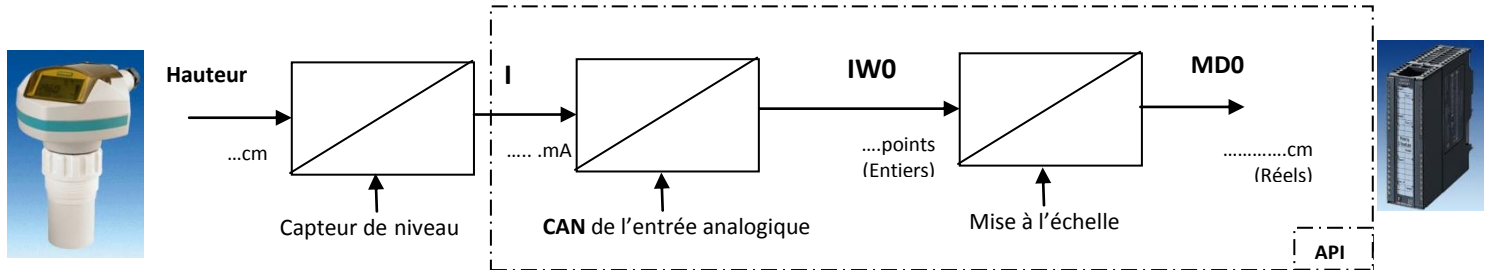
Surveillance d'un niveau haut sur une cuve de dimension 0 à 10,0 m.

Alarme niveau HH enclenchée si niveau de 850 cm dépassé et effacée si niveau haut inférieur à 830 cm.

Caractéristiques du capteur de niveau : sortie 4-20 mA, étendue d'échelle de 0 à 1000 cm.

Caractéristiques de l'entrée analogique API : entrée 4-20 mA, conversion numérique sur 10 bits.

COMPLETER LE SCHEMA DE LA CHAINE DE MESURE



COMPLETER LE TABLEAU SUIVANT

Hauteur (cm)	0			500			800			850			1000		
I (mA)															
IWO (points) en base 10															
IWO (points) en binaire	00	0000	0000												
IWO (points) en hexadécimal															
MDO (réels)															

Equation de mise à l'échelle  $MDO = f(IWO)$  ?

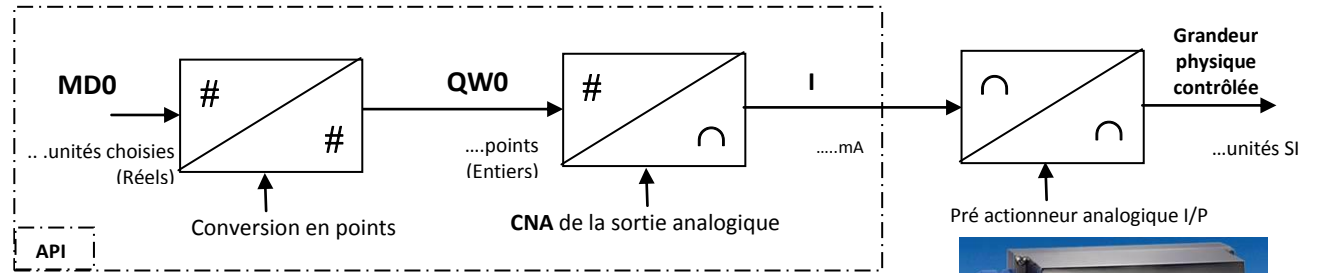
Equation du convertisseur analogique-numérique  $IWO = f(I)$

Résolution du convertisseur analogique :

### III Sorties analogiques API

**Schéma bloc type d'une chaîne d'action** pour une sortie courant unipolaire 4 à 20 mA et une conversion numérique sur 12 bits (sans bit de signe).

**Exemple :** compléter ce schéma pour une sortie 4-20 mA utilisée pour un convertisseur I/P d'une vanne réglante d'étendue 0,2 – 1 bar, représentation de la commande en % sur l'API.



$QW0_{maxi} = 2^n - 1$  avec n bits pour la conversion numérique sans bit de signe



Tableau des variables associées:

MD0 (réels) unités ou représentation choisie en %	00	25	50	75	100
QW0 (points)					
I fourni (mA)					
Grandeur physique contrôlée en bar					

**Equation** caractéristique de la fonction de conversion en points  $QW0 = f(MD0)$  :

$QW0 = MD0 \times \dots\dots\dots$

**Caractéristiques** associées (à compléter)

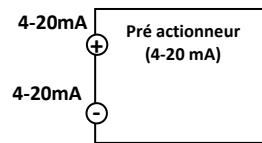
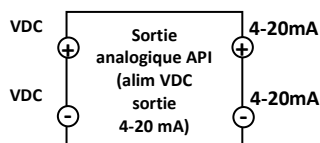
De la conversion en points (unscale)  $QW0 = f(MD0)$

Convertisseur numérique-analogique  $I = f(QW0)$



**Câblage électrique type:**

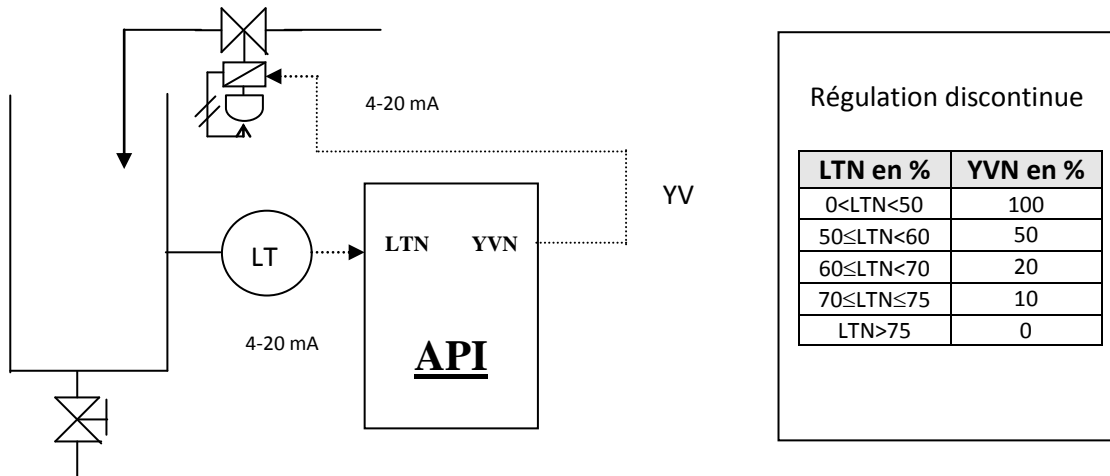
Sortie analogique « 4 fils » API et pré actionneur « 2 fils »



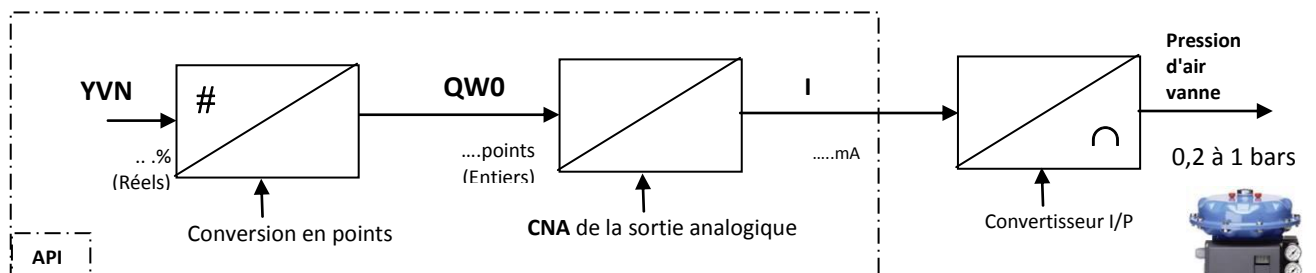
**Exercice d'application : CAHIER DES CHARGES D'UNE REGULATION DISCONTINUE**

On désire réaliser une régulation discontinue de niveau autour d'un point de consigne de 75% et suivant le tableau de commande de la vanne de type NF fourni ci-dessous.

La sortie analogique commandant le convertisseur I/P de la vanne est du type 4-20 mA avec une conversion numérique sur 10 bits.



COMPLETER LE SCHEMA DE LA CHAINE D'ACTION :

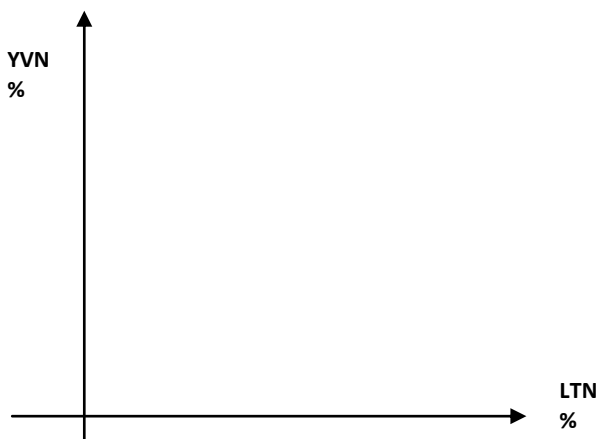


COMPLETER LE TABLEAU SUIVANT :

YVN en %	0	10	20	50	100
QW0 (points) en base 10					
QW0 (points) en hexadécimal					
I (mA)					
Pression de commande (bars)					



REPRESENTER L'EVOLUTION DE YVN = f (LTN)





**IV] PLANNING EXERCICES DE REVISION DE TP D'AUTOMATISME TS2CIRA**

ELEVE	Ex n°1 :		Ex n°2 :		Ex n°3 :		Ex n°4	
	SCHNEIDER	SIEMENS	SCHNEIDER	SIEMENS	SCHNEIDER	SIEMENS	SCHNEIDER	SIEMENS

## VI Exercices de révision TP d'automatisme TS2CIRA

**Réaliser les exercices suivants par roulement sur les six postes de programmation disponible (3 API Schneider et 3 API Siemens)**

**1- Alarme sur grandeur analogique avec API Schneider et Siemens:  
(entrée analogique avec un calibrateur 4-20mA)**

- Déclencher un voyant clignotant lors du dépassement de la pression d'une cuve à 0,8 bars, cette alarme disparaît lorsque la pression redescend à 0,7bar.
- La mesure de pression se fait à l'aide d'un capteur d'étendue d'échelle 0 à 1 bar muni d'une sortie courant 4 – 20 mA.
- Réaliser la configuration matérielle API nécessaire
- Prévoir une table d'animation avec affichage de la pression en mbar et visualisation de l'alarme.
- Programme ladder avec mnémoniques et commentaires
- Câbler un générateur de courant pour simuler le capteur et tester cette application.
- Imprimer le programme commenté.

**2- Alarme avec comptage sur grandeur TOR avec API Schneider :  
(entrée et sortie TOR avec un voyant)**

- L'alarme clignotante doit se déclencher lorsque l'on a comptabilisé 5 apparitions d'un défaut TOR de durée supérieure à 5 secondes.
- Réaliser la configuration matérielle API nécessaire
- Prévoir une table d'animation avec les variables TOR .
- Programme ladder avec mnémoniques et commentaires
- Câbler un voyant pour l'alarme et tester cette application
- Imprimer le programme commenté.

**3- Rampe sur sortie analogique avec API Schneider et Siemens:  
(sortie analogique avec un milliampèremètre 4-20mA)**

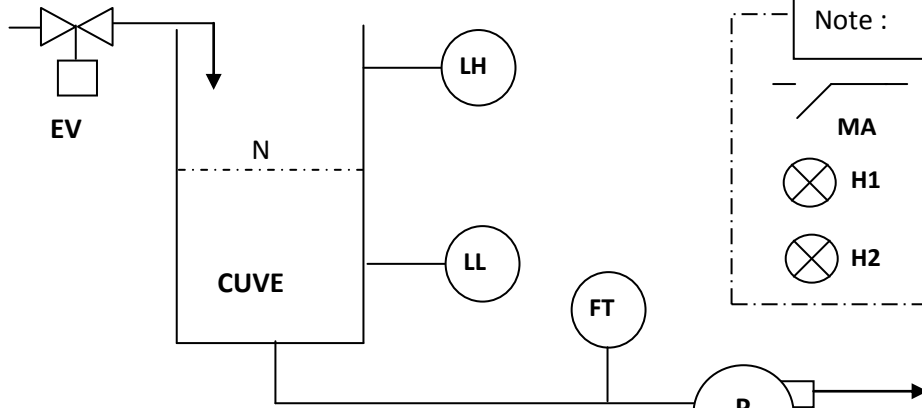
- Pente imposée, limite supérieure et durée du palier : sortie 4- 20mA, montée de 1% toutes les secondes sur déclenchement d'une entrée TOR, palier de 20 secondes à 50%, descente de 2% toutes les secondes jusqu'à 0%.
- Réaliser la configuration matérielle API nécessaire
- Prévoir une table d'animation avec la sortie en %
- Programme ladder avec mnémoniques et commentaires
- Câbler une vanne réglante sur la sortie analogique et tester cette application
- Imprimer le programme commenté.

**4- Régulation de niveau TOR sur API Siemens ou API Schneider :  
(entrée analogique avec un calibrateur 4-20mA)**

- Consigne de 40% et hystérésis de 5% de la consigne
- Entrée analogique 4-20mA
- Réaliser la configuration matérielle API nécessaire
- Prévoir une table d'animation avec l'entrée en %
- Programme ladder avec mnémoniques et commentaires
- Câbler un générateur de courant pour simuler le capteur et tester cette application
- Imprimer le programme commenté.

## VII TP TEST AUTO N°1

Soit l'installation suivante à automatiser :



### Caractéristiques de l'instrumentation :

- Les capteurs de niveaux sont actifs (=1) si le niveau N du liquide est supérieur au seuil du capteur.
- Le capteur de débit fournit une impulsion pour chaque litre de liquide qui le traverse sur sa sortie TOR.
- L'électrovanne EV contrôlant le débit d'entrée est du type FMS.

### Fonctionnement attendu :

- Ouverture et **mémorisation** de EV si le niveau bas est découvert.
- Fermeture de EV si niveau haut est recouvert.
- La pompe fonctionne pour deux conditions possibles:
  - Soit appui permanent sur le BP MA à la condition que le niveau bas soit recouvert
  - Soit niveau haut recouvert
- Les voyants d'alarme :
  - H1 est allumé si le niveau haut est recouvert pendant plus de 10s
  - H2 est allumé si le niveau bas est découvert pendant plus de 15 s
- Comptage du volume de liquide pompé (en litres) dans un mot interne de l'API

### Travail demandé :

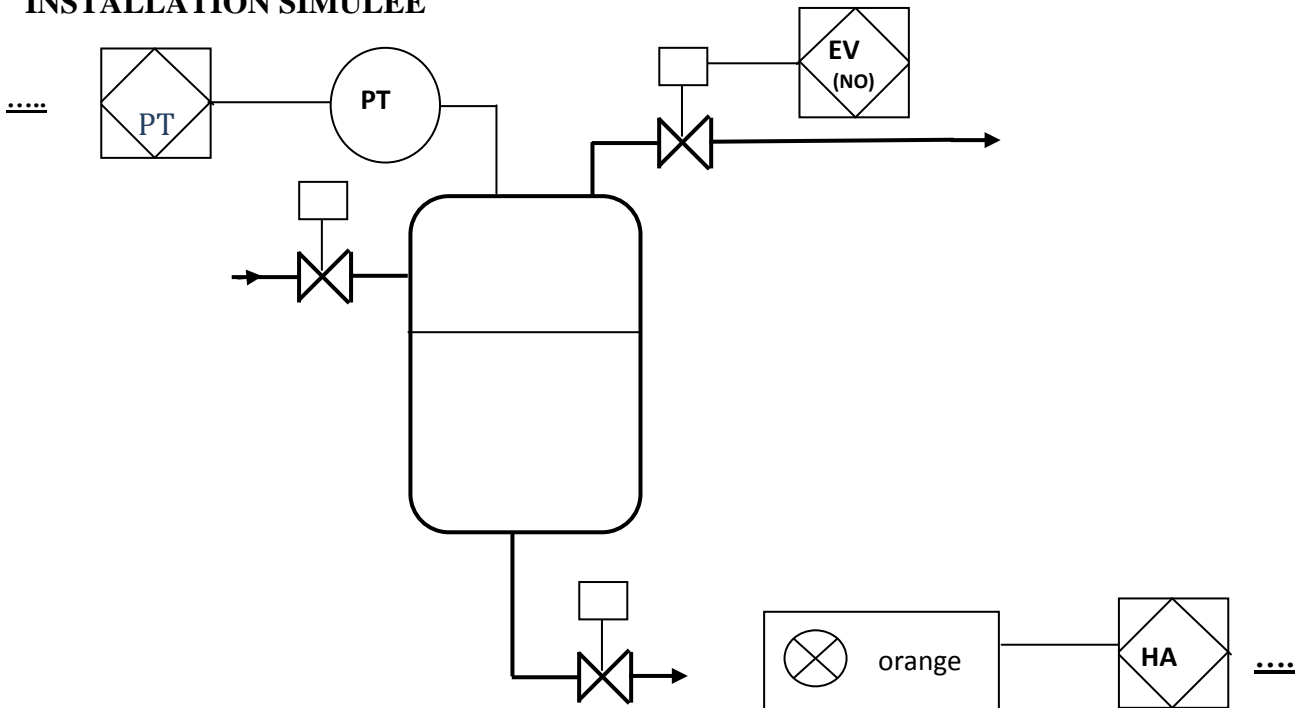
Questions	Note / Barème
1) Ouvrir le fichier TP TEST LADDER et enregistrer le sous votre nom dans le répertoire de votre groupe	/1
2) Préciser le nombre d'entrées TOR et de sorties TOR sur cet API :	/1
3) Identifier les variables d'entrée et de sortie du point de vue de l'API. Faire un tableau à deux colonnes:	/2
4) Programmer les mnémoniques de ces variables avec leurs adresses API, et rajouter un commentaire pour chacune des variables.	/3
5) Créer deux tables de variables (Siemens) ou d'animation (Schneider), l'une pour les entrées appelée : ENTREES et l'autre pour les sorties appelée : SORTIES	/3
6) Programmer en langage à contacts les équations des variables de ce système, en donnant une légende pour chaque réseau de programmation	/6
7) Tester votre programme en simulant les variables d'entrée et en vérifiant les états possibles des sorties (à expliquer au professeur)	/3
6) Visualiser le volume de liquide pompé	/1
<b>NOTE</b>	<b>/20</b>

## VII TP TEST AUTO N°2

### MATERIEL :

- ☞ API SIEMENS S7-300 disposant d'un coupleur d'entrée analogique SM 331
- ☞ Programme partiel appelé : « **alarme1** » comportant la structure matérielle de l'API , à **enregistrer sous votre nom** et dans le répertoire TS1CIRA groupe A ou B
- ☞ Calibrateur de courant simulant le capteur de pression d'étendue - 1 bar à + 1 bar et de sortie 4-20mA
- ☞ Ordinateur avec les logiciels STEP7 V5.1

### INSTALLATION SIMULEE

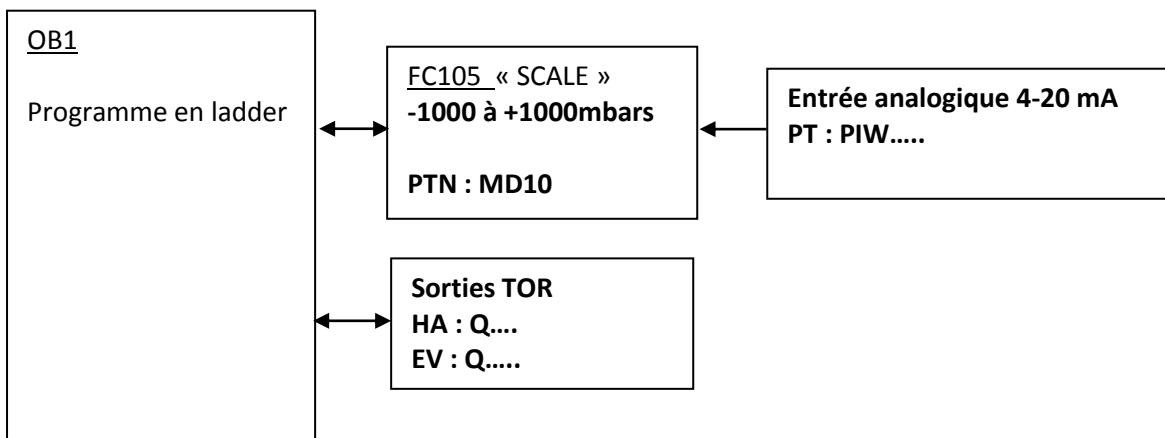


### CAHIER DES CHARGES

Programmer :

- Une alarme HA surveillant le seuil haut de pression de la cuve, celle-ci doit s'enclencher si la pression de 0,8 bar apparaît et se déclencher pour une pression de 0,7 bar.
- Une électrovanne de mise à l'air EV qui doit s'ouvrir uniquement si la pression devient supérieure à 0,9 bar.

### STRUCTURE DU PROGRAMME « ALARME 1 »



**① Mise en œuvre de l'installation :**

- A partir du schéma de l'installation et de la structure du programme, identifier les différentes variables de l'automatisme en complétant le tableau suivant :

Symbole TI	Type	Adresse API
PT		
EV		
HA		

- Compléter le programme en introduisant un bloc FC105 réalisant la mise à l'échelle ( en mbar) de l'entrée analogique. Ce bloc fonctionnel standard FC105 SCALE fait partie de la bibliothèque « STANDARD LIBRARY » dans le programme « TI-S7 Converting Blocks ».
- Proposer un schéma de câblage de l'entrée analogique et vérifier la correspondance entre les grandeurs analogique et numérique et le courant 4-20 mA (compléter le tableau de mesures). **Créer une table de variables « PRESSION »**

	Valeur analogique (en mA)	Valeur numérique API (en points)	Valeur numérique API (mbar)
Adresse API			
0 %			
25 %			
50%			
75%			
100%			

**② Mise en œuvre du dispositif de contrôle commande :**

- Réaliser le programme avec des commentaires et en ladder correspondant au cahier des charges ci-dessus.
- Représenter l'évolution de la variable HA en fonction du courant d'entrée analogique (sens croissant et sens décroissant) graduer les axes.

**NOM :**  
 Prénom :  
 Date :  
 Durée :  
 Note :

**TEST D'EVALUATION TP N°2 AUTOMATISME**

**SEQUENCE D'ALARME**

**① Mise en œuvre de l'installation :**

<b>Thèmes</b>	<b>Correct</b>	<b>Incorrect</b>	<b>Aide</b>	<b>Points</b>
- Identification variables				/1
- Enregistrer le programme sous un autre nom et dans le bon répertoire				/1
- Programme mise à l'échelle, paramétrages du bloc FC105				/3
- Schéma de câblage entrée ana.				/1
- Programmation de la table de variable « PRESSION »				/2
- Tableau de mesures				/3
- Evolution de HA en fonction du courant				/2

**② Mise en œuvre du dispositif de contrôle commande :**

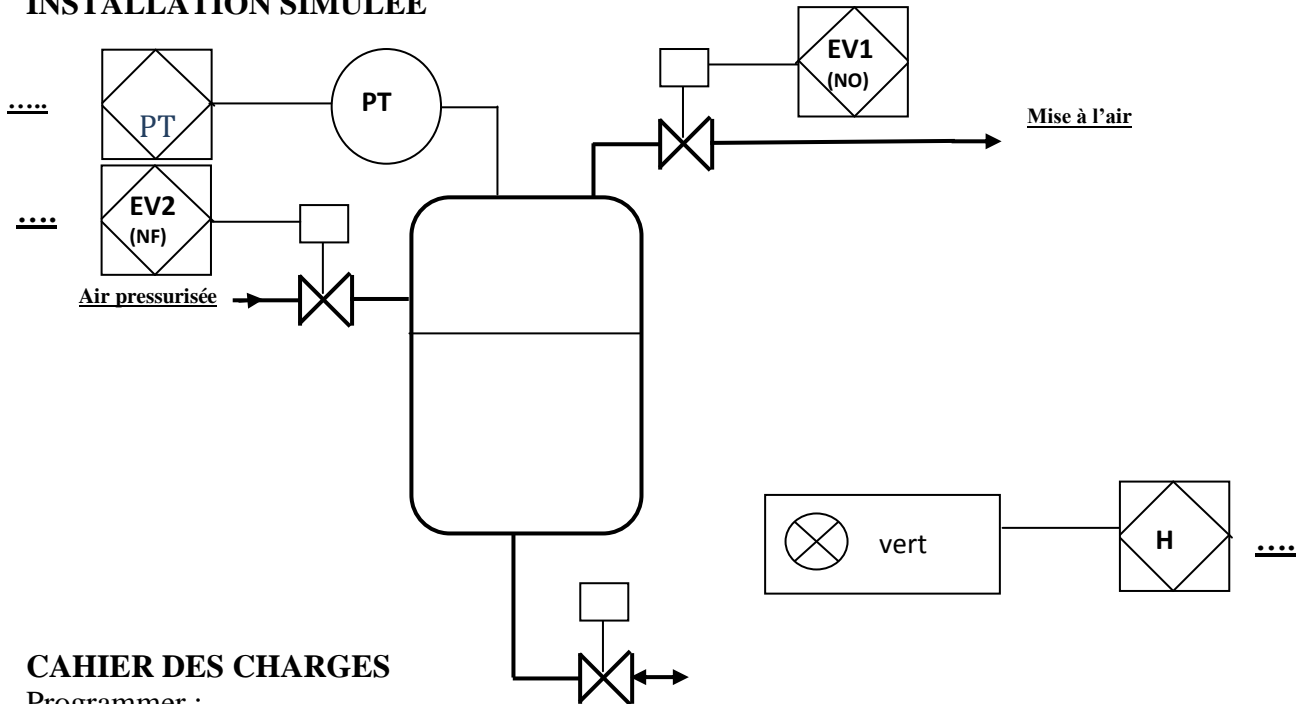
<b>Thèmes</b>	<b>Correct</b>	<b>Incorrect</b>	<b>Aide</b>	<b>Points</b>
- Programme HA				/2
- Programme EV				/2
- Mise en service et simulation avec explications				/3

## VIII] TP TEST TP AUTO N°3

### **MATERIEL :**

- ☞ **API SIEMENS S7-300** disposant d'un coupleur d'entrée analogique SM 331
- ☞ Programme partiel appelé : « regulator » comportant la structure matérielle de l'API , à **enregistrer sous votre nom** et dans le répertoire TS1CIRA groupe A ou B
- ☞ **Calibrateur de courant simulant le capteur de pression d'étendue - 0,5 bar à + 0,5 bar et de sortie 4-20mA**
- ☞ Ordinateur avec les logiciels STEP7 V5.1

### **INSTALLATION SIMULEE**

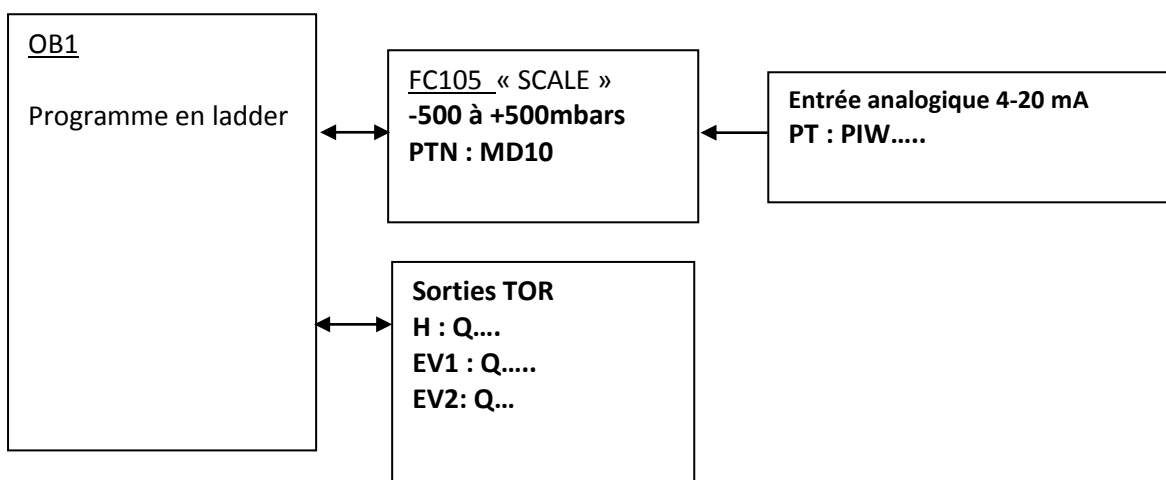


### **CAHIER DES CHARGES**

Programmer :

- Une régulation maintenant une pression comprise entre 0,3 et 0,4 bar dans la partie supérieure de la cuve.
- Un voyant H signalant que la pression est bien comprise entre ces deux valeurs.

### **STRUCTURE DU PROGRAMME « REGULTOR »**



**① Mise en œuvre de l'installation :**

- A partir du schéma de l'installation et de la structure du programme, identifier les différentes variables de l'automatisme en complétant le tableau suivant :

Symbole TI	Type	Adresse API
PT		
EV1		
EV2		
H		

- Compléter le programme en introduisant un bloc FC105 réalisant la mise à l'échelle ( en mbar) de l'entrée analogique. Ce bloc fonctionnel standard FC105 SCALE fait partie de la bibliothèque « STANDARD LIBRARY » dans le programme « TI-S7 Converting Blocks ».
- Proposer un schéma de câblage de l'entrée analogique et vérifier la correspondance entre les grandeurs analogique et numérique et le courant 4-20 mA (compléter le tableau de mesures).  
**Créer une table de variables « PRESSION »**

	Valeur analogique (en mA)	Valeur numérique API (en points)	Valeur numérique API (mbar)
Adresse API			
0 %			
25 %			
50%			
75%			
100%			

**② Mise en œuvre du dispositif de contrôle commande :**

- Réaliser le programme avec commentaires et en ladder correspondant au cahier des charges ci-dessus.
- Représenter l'évolution de la variable H en fonction du courant d'entrée analogique (sens croissant et sens décroissant) graduer les axes.



**NOM :**  
 Prénom :  
 Date :  
 Durée :  
 Note :

**TEST D'EVALUATION TP N°3 AUTOMATISME**

**REGULATION TOR**

**① Mise en œuvre de l'installation :**

Thèmes	Correct	Incorrect	Aide	Points
- Identification variables				/1
- Enregistrer le programme sous un autre nom et dans le bon répertoire				/1
- Programme mise à l'échelle, paramétrages du bloc FC105				/3
- Schéma de câblage entrée ana.				/1
- Programmation de la table de variable « PRESSION »				/2
- Tableau de mesures				/3

**② Mise en œuvre du dispositif de contrôle commande :**

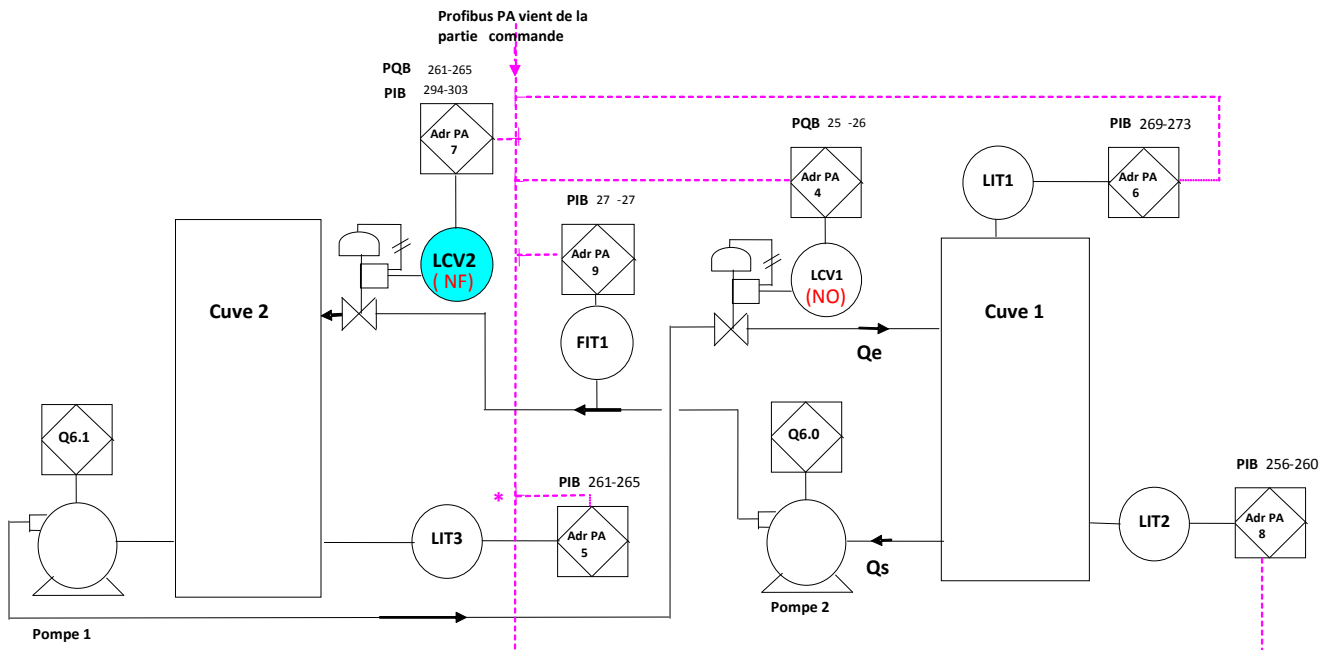
Thèmes	Correct	Incorrect	Aide	Points
- Programme H				/1
- Programme EV1 et EV2				/3
- Evolution de H en fonction du courant				/2
- Mise en service et simulation avec explications				/3



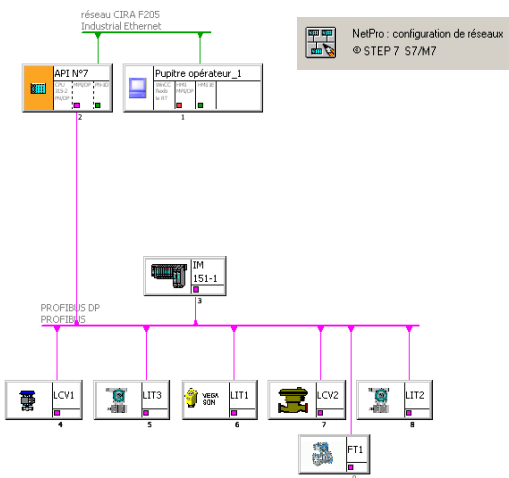
## X] Diagnostic vanne Samson avec positionneur PA

**Objectif :** Utiliser les possibilités de diagnostic sur une vanne implantée sur un réseau de terrain Profibus PA (comparaison entre la consigne et la position réelle de la vanne) et activer une alarme dans le cas où l'écart entre la consigne et la position de la vanne est supérieur à 5%. dans vue d'un IHM .

**Contexte :** schéma TI, l'étude portera sur la vanne LCV2



**Configuration matériel sur Simatic manager (ouvrir le projet « diagnostic vanne LCV2 »)**



ADRESSES DES CAPTEURS ET POSITIONNEURS SUR LE BUS DE TERRAIN PA		
Adr PA 4	LCV1	Samson avec Sipart PS2-PA
Adr PA 5	LIT3	Siemens Sitrans P DS III PA
Adr PA 6	LIT1	Vega Vegason 51
Adr PA 7	LCV2	Samson positionneur 3785 profil 3
Adr PA 8	LIT2	Siemens Sitrans P DS III PA
Adr PA 9	FIT1	Krohne Optiflux 4300C

Emplac.	Identification...	N° de référence / Description	Longueur entrée	Longueur sortie	Commentaire
1	199	SP+REARBACK+POS_D+CHECKBACK	294...303	261...265	

10 octets en entrée pour la position et l'état de la vanne

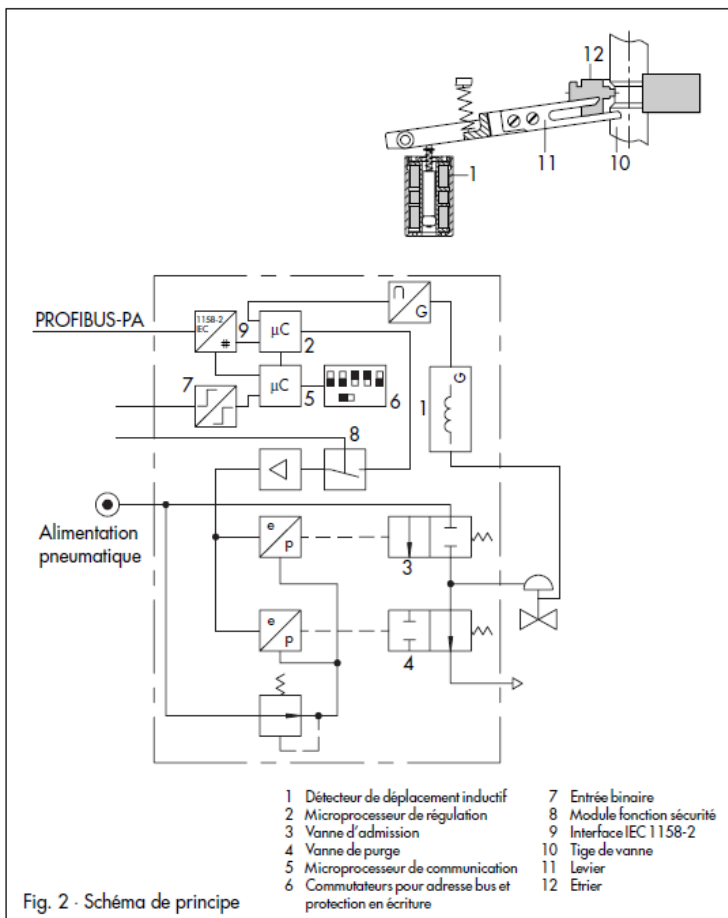
5 octets en sortie pour la commande de la vanne

**Référence vanne à décoder :**

**3785-00011310.04**

Positionner		Type 3785	X	X	X	X	X	3	X
Not explosion protected			0						
Ⓔ II 2 G EEx ia IIC T6 / II 2 D IP 65 T 80 °C acc. to ATEX			1						
Ex ia FM/CSA			3						
Additional accessories									
Inductive limit switch	without		0						
	with Type SJ 2-SN		2						2
with forced venting	without			0					
	24 V DC			1					2
PA device profile	Version 2.0				0				
	Version 3.0				1				
Pneumatic connections	NPT ¼ - 18						1		
	ISO 228 / - G ¼						2		
Electrical connections	Cable gland M20 x 1.5 with Shielding, nickel-plated brass								1
	Quantity 1								2
	Quantity 2								

**Principe du positionneur**



**A comprendre et être capable de l'expliquer !**

## Commande vanne et retour lecture position

### Variant 5:

Module = "READBACK + POS\_D + CHECKBACK, SP" 0x99, 0xA4, or 0xC2, 0x84, 0x89, 0x82, 0xB3

Input value (Input)

Byte 0	1	2	3	4	5	6	7	8	9
Octet 1 Sign, Exponent	Octet 2 Exponent, Fraction	Octet 3 Fraction	Octet 4 Fraction	Octet 5	Octet 1	Octet 2	Octet 1	Octet 2	Octet 3
READBACK, Value (Floating Point, IEEE)				Status	POS_D Value	POS_D Status	CHECK_ BACK[0]	CHECK_ BACK[1]	CHECK_ BACK[2]

Output value (Output)

Byte 0	1	2	3	4
Octet 1 Sign, Exponent	Octet 2 Exponent, Fraction	Octet 3 Fraction	Octet 4 Fraction	Octet 5
SP, Value (Floating Point, IEEE)				Status

**Rappel du format dit « à virgule flottante sur 2 bits »** et ce sur 5 octets (exemple pour la consigne de la vanne LCV2 de MB261 à MB265):

MB265	MB264	MB263	MB262	MB261
Octet 4	Octet 3	Octet 2	Octet 1	Octet 0
Etat	Signe	Exposant	Mantisse	
	Valeur de la mesure en virgule flottante			

L'octet d'état permet de connaître un éventuel défaut du capteur (si la mesure est « OK », sa valeur est de 80<sub>(16)</sub>)

**Formule = (-1)<sup>signe</sup> x 2<sup>(exposant - 127)</sup> x (1 + mantisse)**

### Explications des variables :

**SP : consigne vanne LCV2**

**SP – Set point with status: Reference variable w in "Auto" operating mode**

In automatic mode ("Auto"), the reference variable w of the positioner is preset via SP. SP consists of a floating point value (4 byte) and the associated status (1 byte). Value and status must be transmitted together (data consistency = 5 byte). If the status of the reference variable is "bad" (value < 64 decimal), the positioner remains in the fail-safe position determined by the actuator.

**READBACK : position réelle de la vanne en % de la plage de réglage**

**POS\_D : copie position binaire**

**READBACK – Current position with status: Controlled variable x**

The position feedback is transmitted via the READBACK parameter and consists of a floating point value (4 byte) and the associated status (1 byte).

**POS\_D – Discrete valve position feedback with status: Final position indication**

The final valve position is indicated via the POS\_D parameter and consists of one message value (1 byte) and the associated status (1 byte).

The message value is encoded as follows:

0 = not initialized

1 = closed (x < 0.5 %)

2 = open (x > 99.5 %)

3 = intermediate position

## CHECKBACK : état de l'appareil

### CHECKBACK – Device status: Detailed device information, bit-wise encoded

Each bit can be hidden by the Class 2 master for cyclic communication. This allows a specific selection to be made from the existing messages.

Bit no.	Name	Description	Byte
0	CB_FAIL_SAFE	Fail safe position (MODE = out of service) set when FSAFE is active	0
1	CB_REQ_LOC_OP	Request for local operation	
2	CB_LOCAL_OP	Device in local mode, initialization or zero calibration running	
3	CB_OVERRIDE	Emergency operation / forced venting active	
4, 5, 6	Not assigned		
7	CB_TRAV_TIME	Status of movement monitoring (is reset automatically)	
8, 9	Not assigned		
10	CB_UPDATE_EVT	Set when static data are changed	1
11	CB_SIMULATE	Simulation mode, i.e. values are not derived from the process	
12	CB_DISTURBANCE	Error, see DIAGNOSIS parameter for cause	
13	CB_CONTR_ERR	Internal control loop error (must be confirmed via Class 2 Master). Indicated by LED, reset automatically as soon as control loop monitoring can no longer detect any error.	
14	CB_CONTR_INACT	Positioner inactive (MODE = out of service)	
15	CB_SELFTEST	Device in self-test mode (MODE = out of service)	2
16	CB_TOT_VALVE_TRAV	Limit value for total valve travel exceeded	
17	CB_BINARY_INPUT	Status of binary input	
18...23	Not assigned		

### Raccordement bornier :

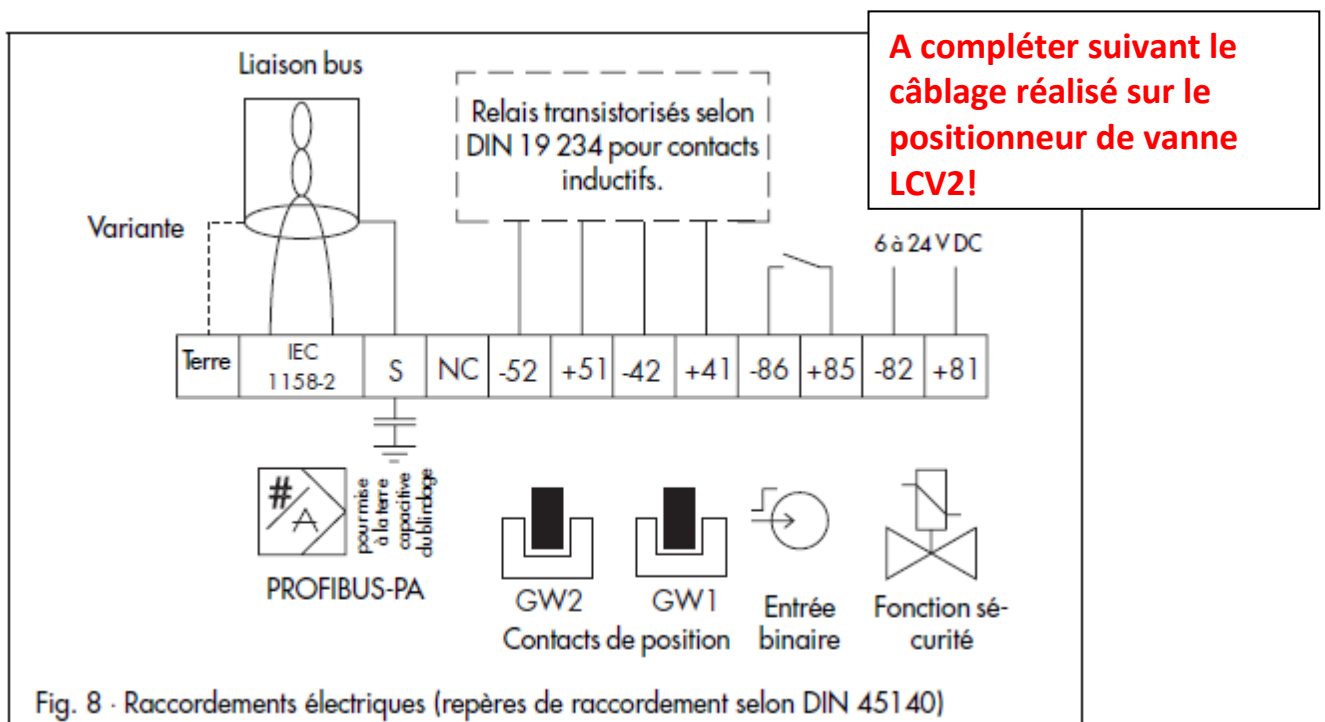
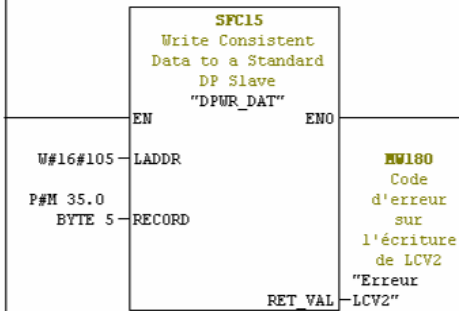


Fig. 8 - Raccordements électriques (repères de raccordement selon DIN 451 40)

**Programme API Simatic manager STEP7 pour le contrôle de la vanne LCV2 extrait de la fonction FC1**

Réseau 2 : Ecriture vanne LCV2 (NF)

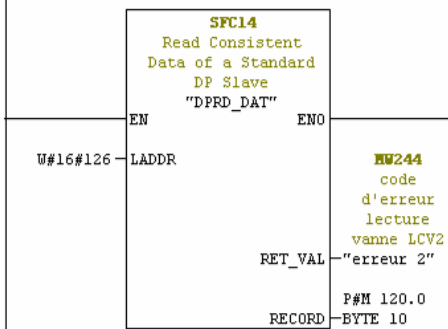
LCV2 commande positionneur vanne Samson PA n°7 avec positionneur type 3785 profil 3 de Samson



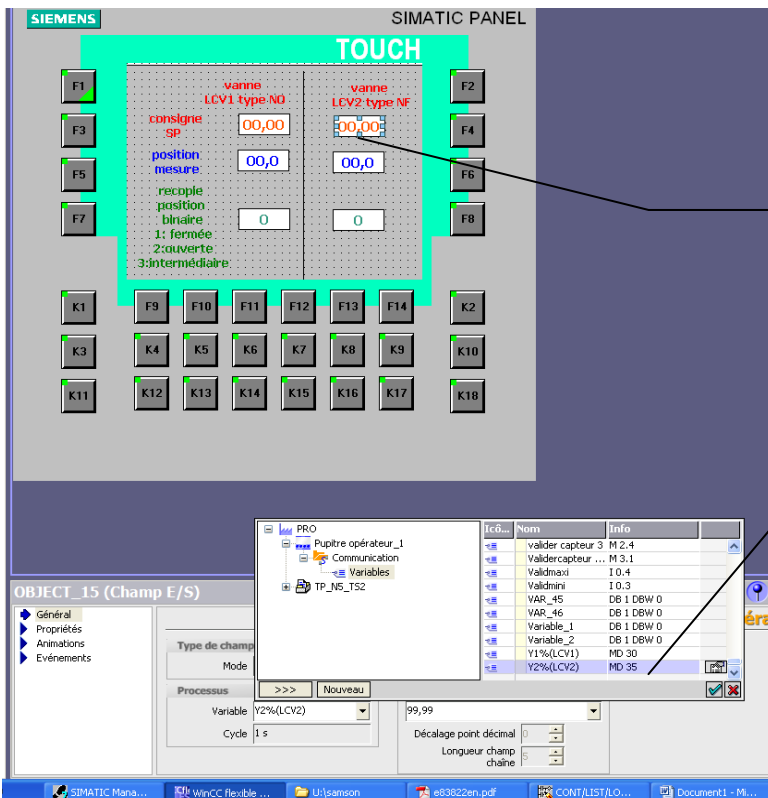
**A comprendre et être capable de l'expliquer !**

Réseau 9 : Lecture retour positionneur LCV2 (NF)

Retour vanne LCV2 sur 10 octets

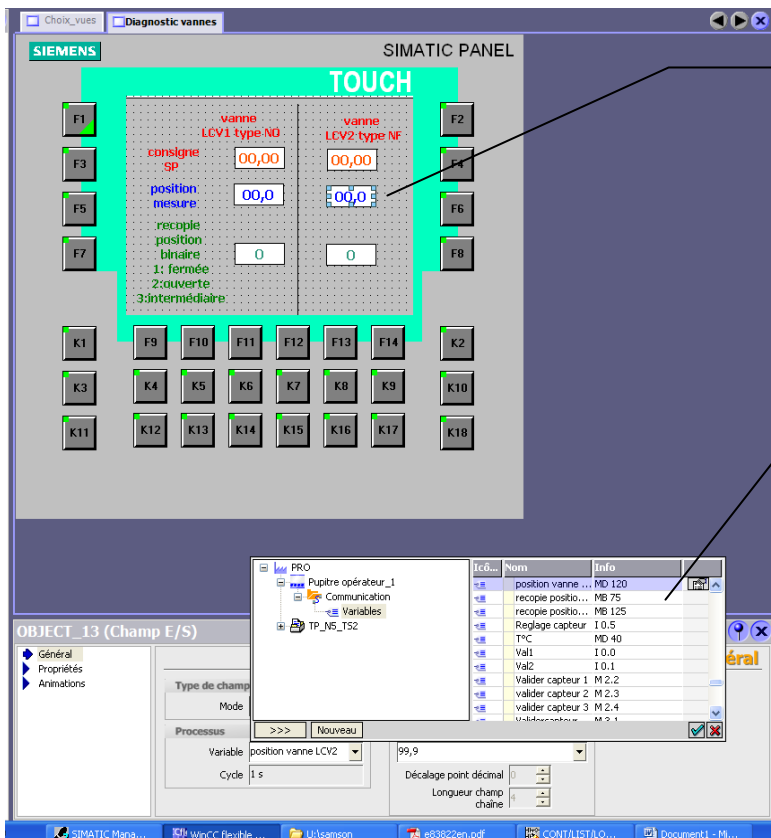


**Ecran WinCC de la vue « diagnostic vannes » :**



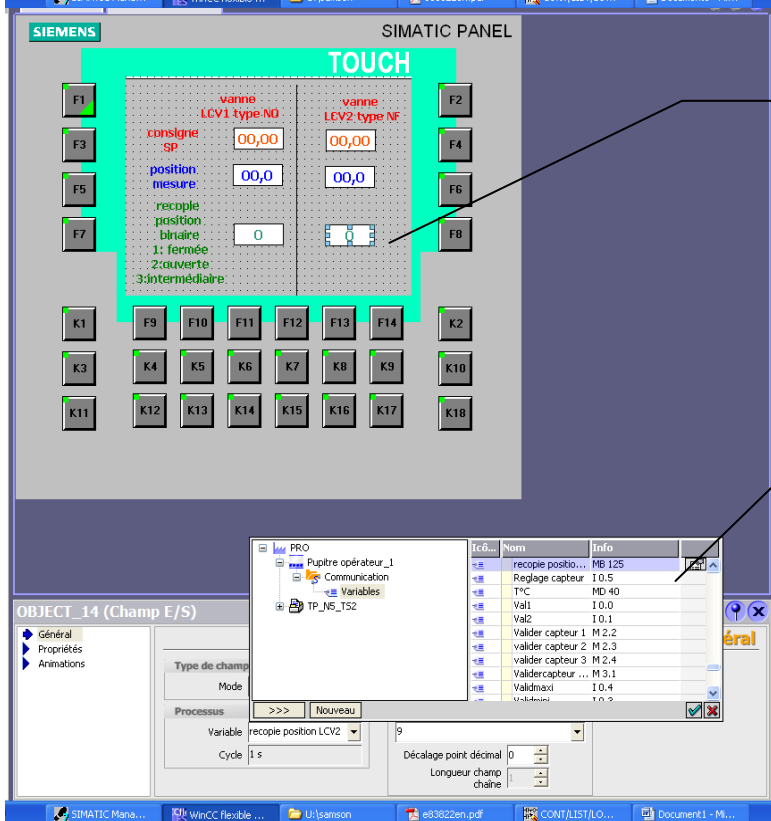
Variable consigne ouverture de la vanne LCV2

Adresse API consigne ouverture de la vanne LCV2



Variable retour position vanne LCV2

Adresse API retour position vanne LCV2



Adresse API POS\_D de la vanne LCV2

Adresse API POS\_D de la vanne LCV2

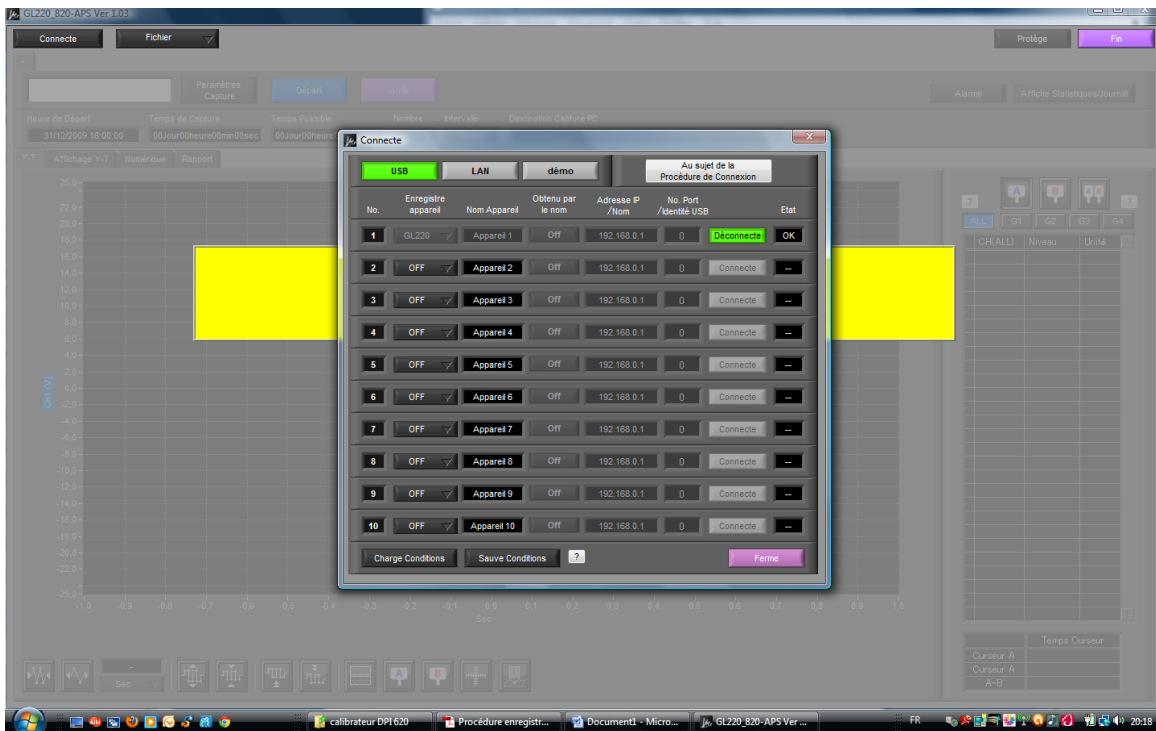
**Travail demandé après avoir renommé votre projet et enregistré dans D:\CIRA\TS2)**

- Expliquer pour une valeur de consigne pour LCV2 =50% le résultat en virgule flottante, à vérifier sur la table de variables « **capteurs actionneurs** ».
- Vérifier les différentes valeurs de la variable « **POS\_D** » pour les différentes positions de la vanne LCV2
- Ajouter un voyant d'alarme sur la vue « **diagnostic vannes** » si l'écart entre la consigne et la position de la vanne LCV2 excède 5% de la plage de réglage.



## XI) Procédure enregistrement et transfert des mesures LOGGER GL220→PC (Excel)

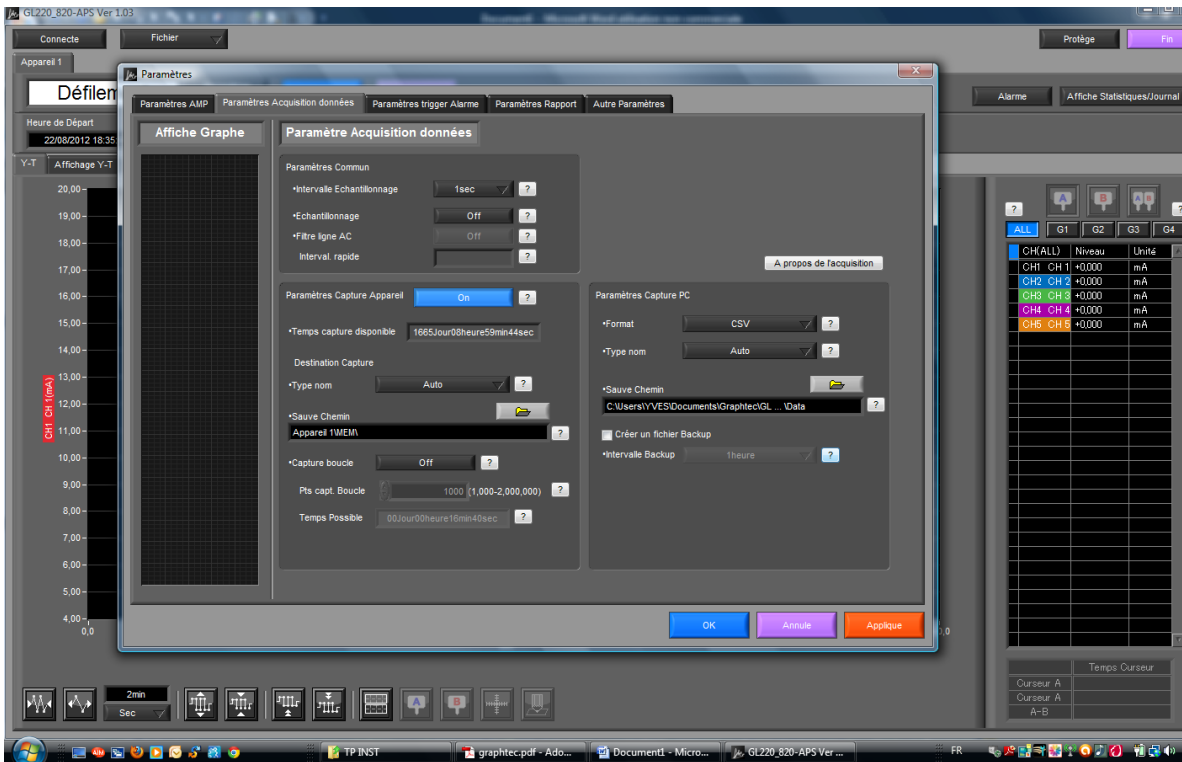
Connecter le logger au PC via la câble USB et vérifier sa connexion via le logiciel Graftec : GL220\_820APS



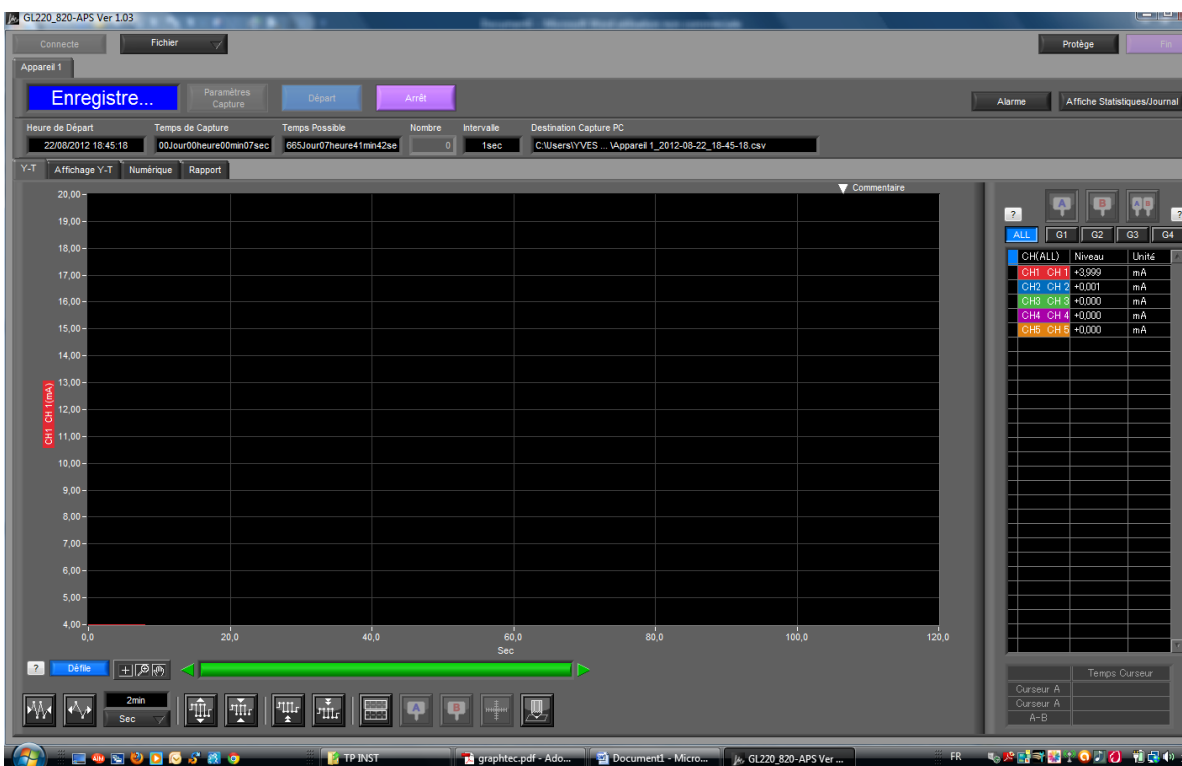
Vérifier la configuration des 5 voies de mesure suivant l'écran ci-dessous dans l'onglet paramètres acquisition des données: (shunt de 250 ohms sur chacune des voies configurées en 1-5V avec une étendue d'échelle de 4-20 mA)



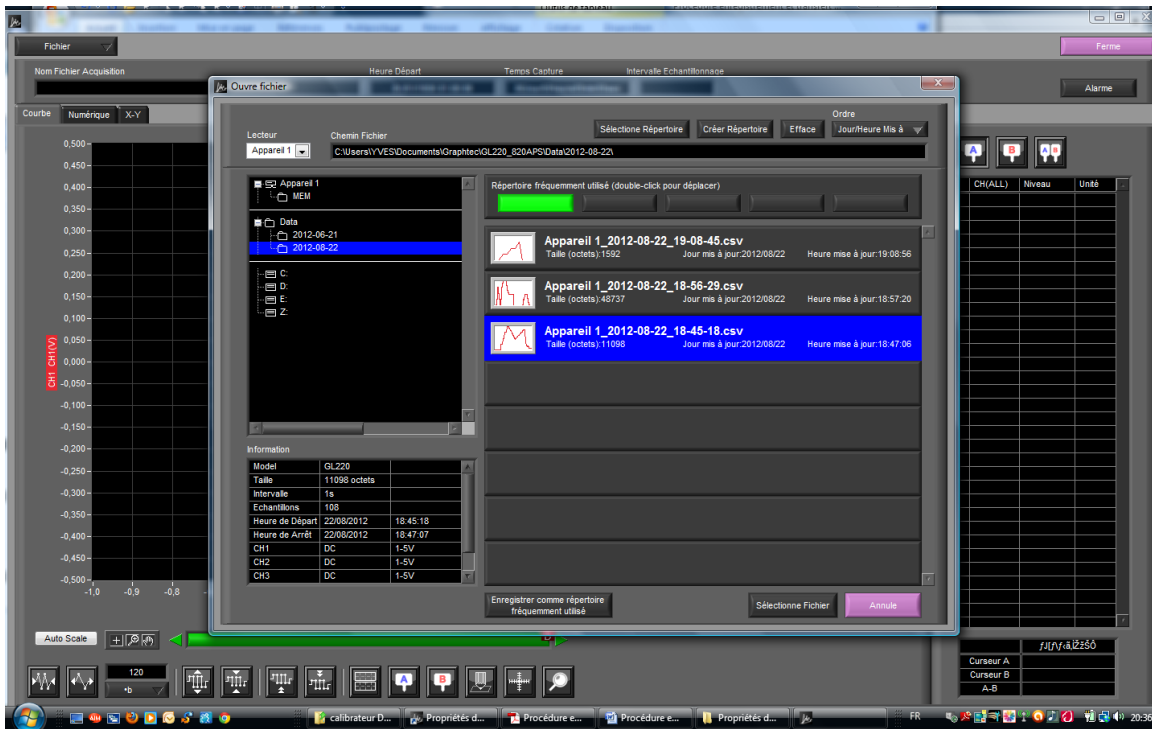
Modifier éventuellement l'intervalle d'échantillonnage, ici 1 seconde :



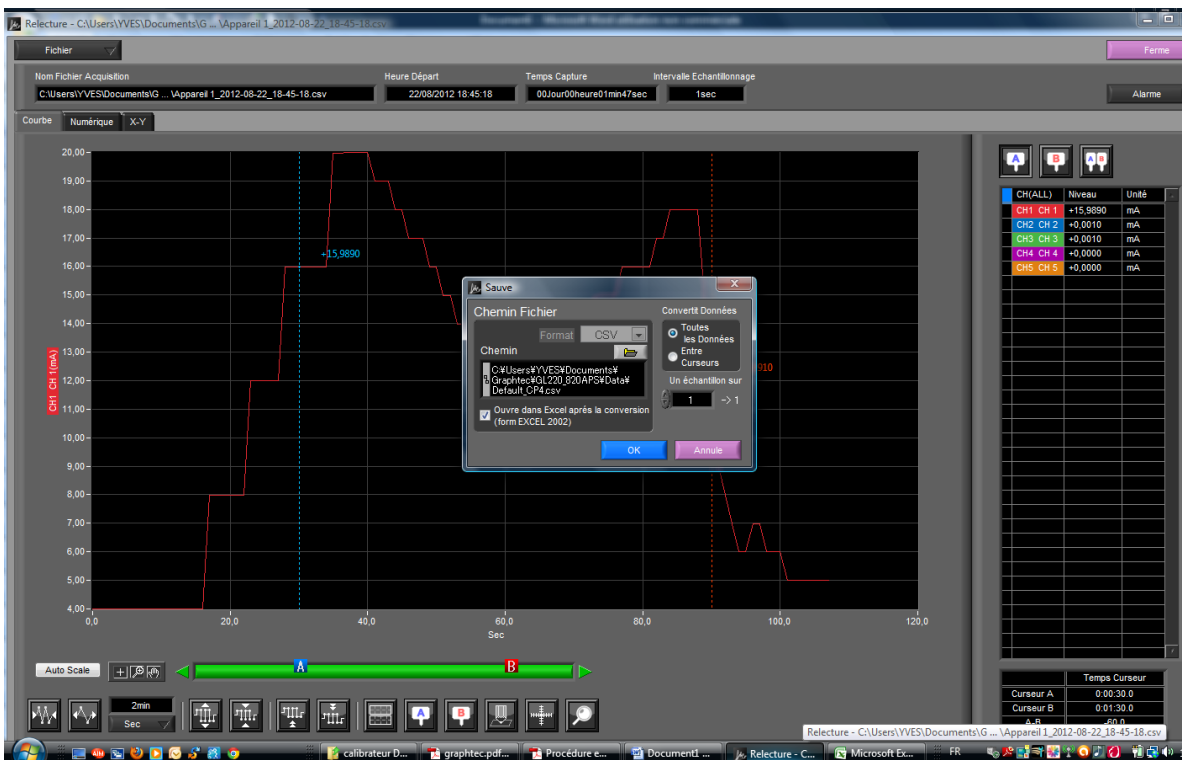
Câbler votre capteur 4-20mA sur la voie 1 et démarrer l'enregistrement :



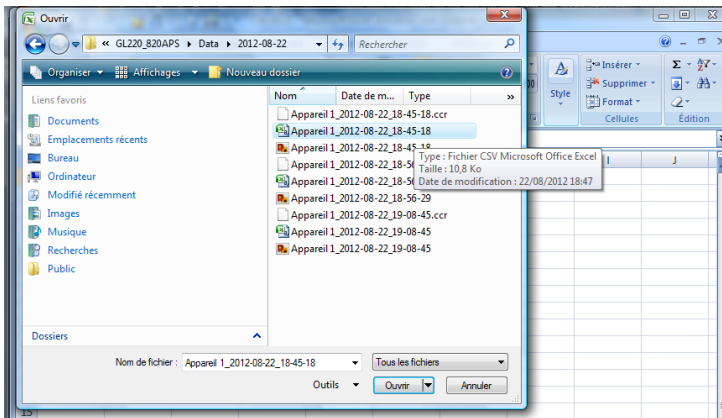
Après la fin de votre enregistrement récupérer votre fichier de données au format CSV :



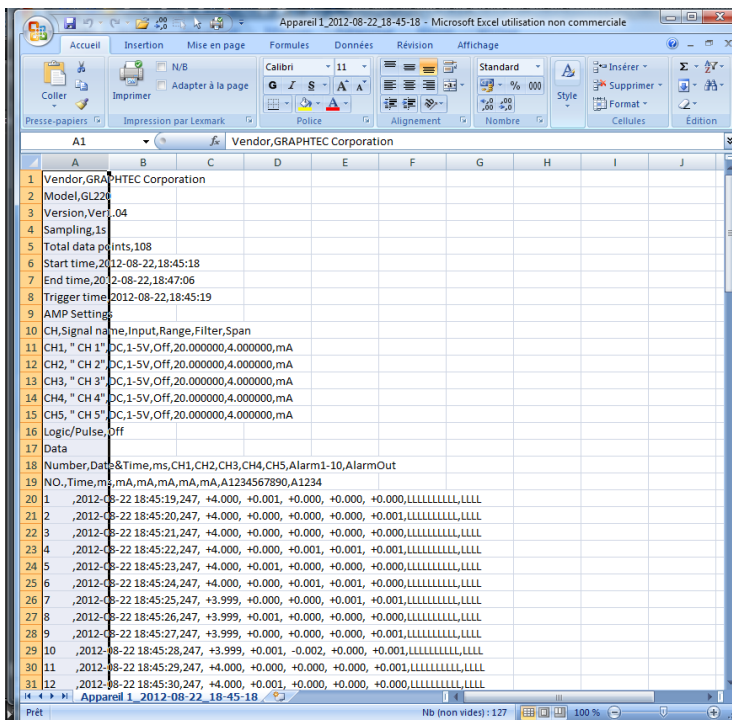
Convertir et sauve ce fichier afin de l'ouvrir avec Excel :



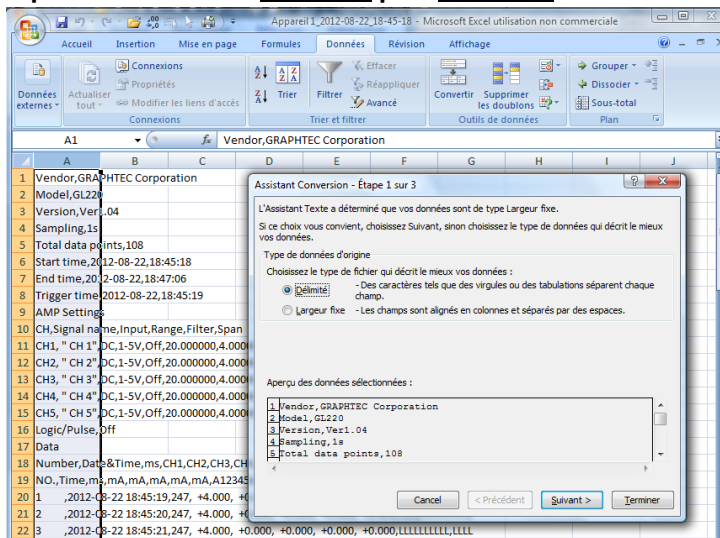
Ouvrir Excel et rechercher le fichier .csv des données dans le répertoire correspondant:



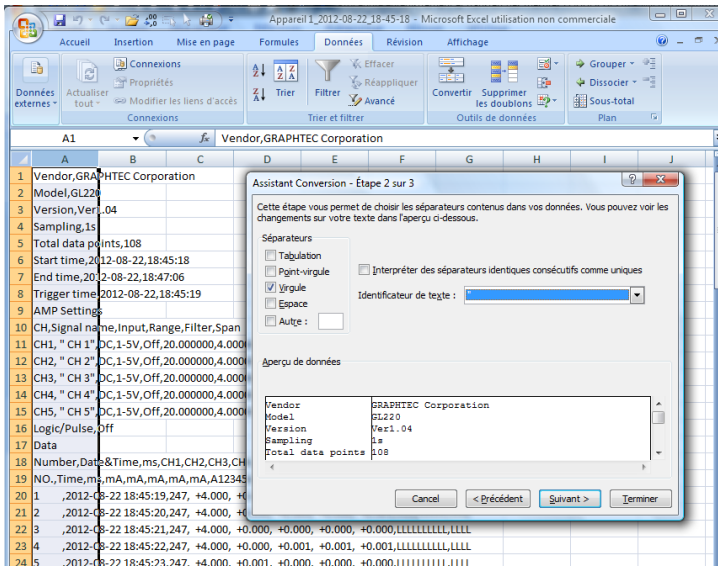
Ouvrir le fichier et sélectionner la colonne A



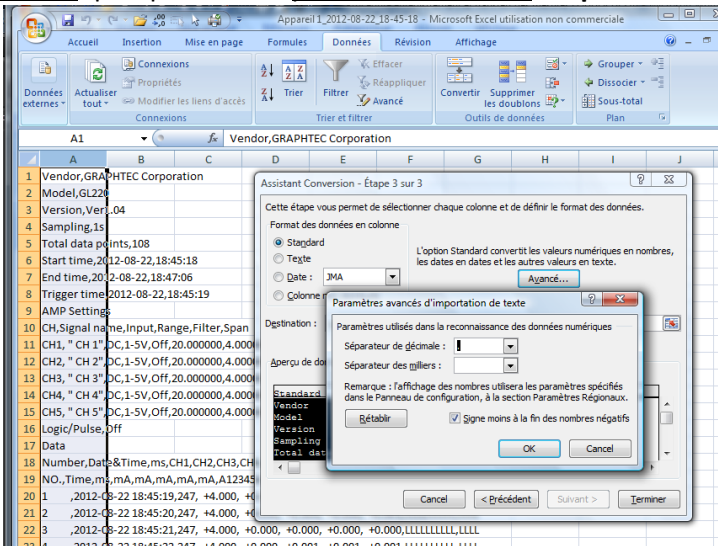
Cliquer sur le menu : **Données** puis **convertir** et cocher la case **Délimité**



**suivant** puis cocher le séparateur : **virgule** « , »



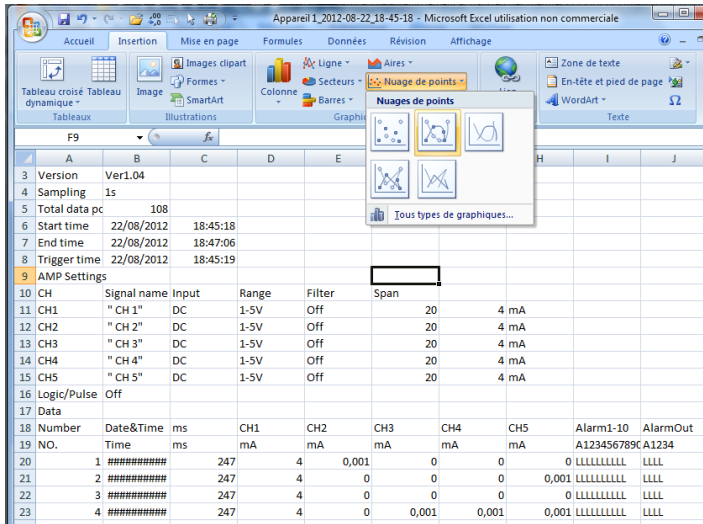
**Suivant** puis préciser les **paramètres avancés** : **séparateur de décimale** utilisé, ici le « . »



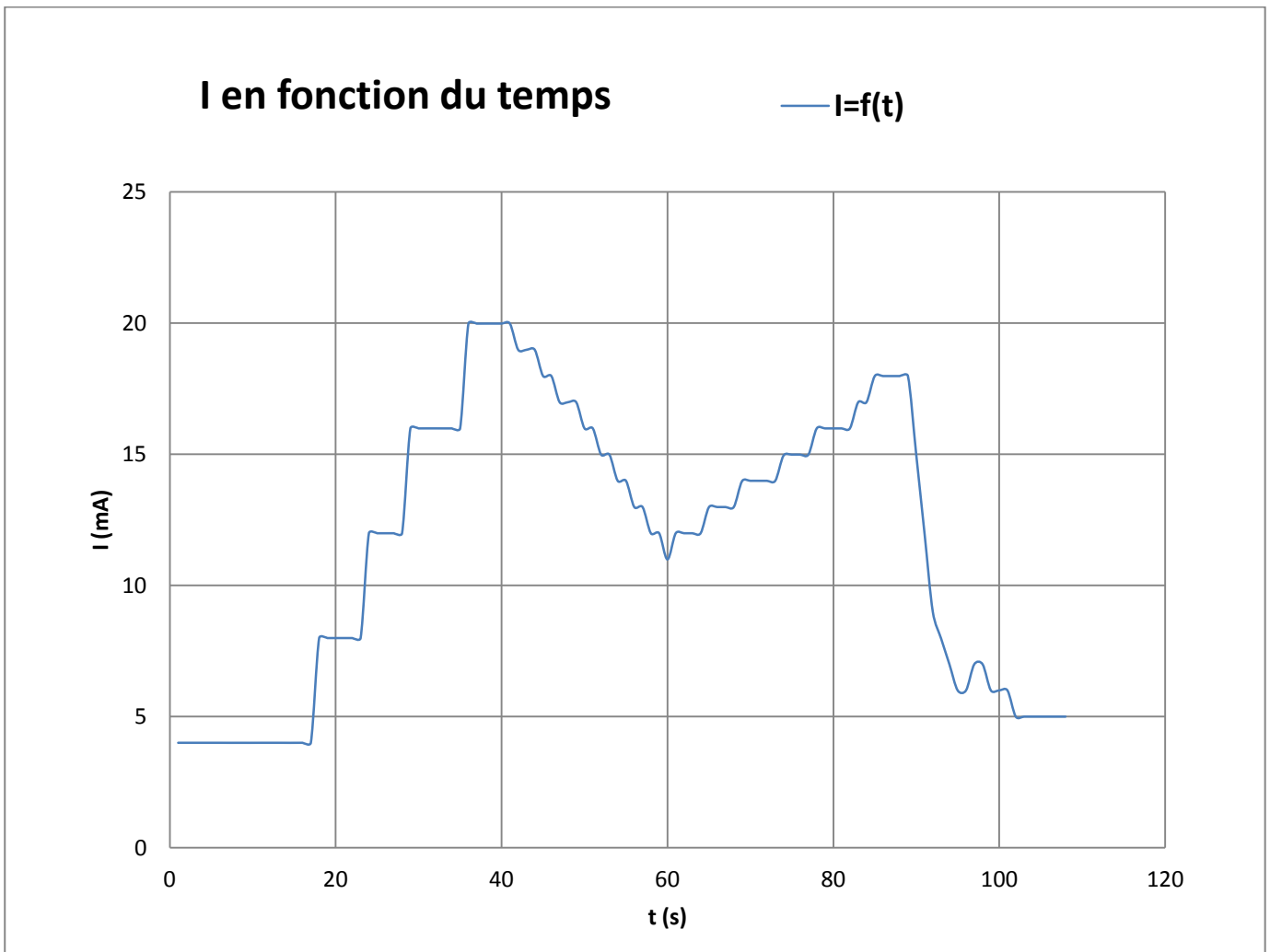
Puis **terminer** afin d'obtenir le tableau sous la forme suivante :

AMP Settings		Signal name	Input	Range	Filter	Span					
CH1	" CH 1"	DC	1-5V	Off	20	4 mA					
CH2	" CH 2"	DC	1-5V	Off	20	4 mA					
CH3	" CH 3"	DC	1-5V	Off	20	4 mA					
CH4	" CH 4"	DC	1-5V	Off	20	4 mA					
CH5	" CH 5"	DC	1-5V	Off	20	4 mA					
Logic/Pulse		Off									
Data		Number	Date&Time	ms	CH1	CH2	CH3	CH4	CH5	Alarm1-10	AlarmOut
		NO.	Time	ms	mA	mA	mA	mA	mA	A123456789CA1234	
20	1	#####	247	4	0,001	0	0	0	0	0	LLLLLLLLLLLL LLLL
21	2	#####	247	4	0	0	0	0	0,001	0	LLLLLLLLLLLL LLLL
22	3	#####	247	4	0	0	0	0	0	0	LLLLLLLLLLLL LLLL
23	4	#####	247	4	0	0,001	0,001	0	0,001	0	LLLLLLLLLLLL LLLL
24	5	#####	247	4	0,001	0	0	0	0	0	LLLLLLLLLLLL LLLL
25	6	#####	247	4	0	0,001	0,001	0	0	0	LLLLLLLLLLLL LLLL
26	7	#####	247	3,999	0	0	0,001	0,001	0,001	0	LLLLLLLLLLLL LLLL
27	8	#####	247	3,999	0,001	0	0	0	0	0	LLLLLLLLLLLL LLLL
28	9	#####	247	3,999	0	0	0	0	0,001	0	LLLLLLLLLLLL LLLL
29	10	#####	247	3,999	0,001	-0,002	0	0	0,001	0	LLLLLLLLLLLL LLLL
30	11	#####	247	4	0	0	0	0	0,001	0	LLLLLLLLLLLL LLLL
31	12	#####	247	4	0,001	0	0	0	0	0	LLLLLLLLLLLL LLLL
32	13	#####	247	4	0	-0,002	0	0	0,001	0	LLLLLLLLLLLL LLLL
33	14	#####	247	3,999	0	0,001	0	0	0	0	LLLLLLLLLLLL LLLL

Tracer le graphique correspondant en format Nuages de points en courbe lissée :



Mettre en forme le graphique afin d'obtenir un format précisant les grandeurs et les unités utilisées ainsi que le titre du graphique :



## XIII ADRESSAGE DES MOTS INTERNES D'UN API

### SCHNEIDER

- Bits internes %M0 à %M255 **indépendants** de la zone des mots internes définis ci-dessous
- Octets internes : ensemble de 8 bits %MBj
- Mots internes : ensemble de 16 bits %MWi
- Mots doubles : ensemble de 32 bits %MDi
- Mots flottants : ensemble de 32 bits %MFk

Principe de l'adressage des mots internes : (**attention aux chevauchements des mots**)

%MB7	%MB6	%MB5	%MB4	%MB3	%MB2	%MB1	%MB0
%MW3		%MW2		%MW1		%MW0	
%MD2				%MD0			
%MD1							

Poids fort et poids faible :

%MBj+1 poids fort	%MBj poids faible
%MWi (avec j=2xi)	

%MWi+1 poids fort	%MWi poids faible
%MDi	

### SIEMENS

- Bits internes M0.0 à M255.7 **dépendants** des mots suivants
- Octets internes : ensemble de 8 bits MBi
- Mots internes : ensemble de 16 bits MWi
- Mots doubles : ensemble de 32 bits MDi

Principe de l'adressage des mots internes : (**attention aux chevauchements des mots**)

MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
MW6		MW4		MW2		MW0	
MD4				MD0			
MD2							

Poids fort et poids faible :

Mi.7 poids fort	Mi.6	Mi.5	Mi.4	Mi.3	Mi.2	Mi.1	Mi.0 poids faible
MBi							

MBi+1 poids faible	MBi poids fort
MWi	

MWi+1 poids faible	MWi poids fort
MDi	