

# R119 – Algèbre de Boole

Licence Pro Rob&IA

*Laurent ROY*

- **PARTIE 1 (sept-octobre) – 3h de cours ; 6h de TP**
  - Appréhender la structure et le fonctionnement d'un système informatique, Initiation à l'administration d'un OS Linux,
  - Comprendre les mécanismes permettant de communiquer avec l'extérieur, liens avec les réseaux informatiques.
- **PARTIE 2 (nov-déc-janv) – 10h de cours ; 14h de TP**
  - Apprendre les règles d'adressage IPv4 et comprendre le rôle d'une passerelle ,
  - Connaître la structure d'un réseau d'entreprise, Modèle OSI,
  - Protocoles ARP / IP / ICMP / TCP / UDP

- **Première partie : Initiation aux systèmes informatiques Linux**
  - 1) Introduction à l'informatique. Principe d'un OS.
  - 2) Linux : Un bref historique
  - 3) Notion des principales commandes `mkdir` ; `cd` ; `cp` ; `mv` ; `ps` ; `kill` ...
  - 4) Gestion des utilisateurs et des droits
  - 5) Lien avec les réseaux informatiques

- **Deuxième partie : Initiation aux réseaux d'entreprise**

- 1) IPv4 ; configuration cartes réseaux ; rôle d'une passerelle,

- 2) Modèles OSI

- 3) Analyse de protocoles

- a) Couche 2 : Adresses MAC ; protocole ARP

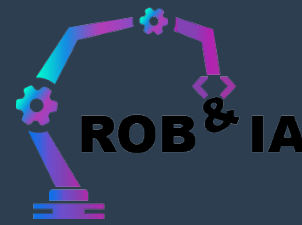
- b) Couche 3 : protocoles IP, ICMP

- c) Couche 4 : protocoles TCP et UDP

- 4) Topologie des réseaux

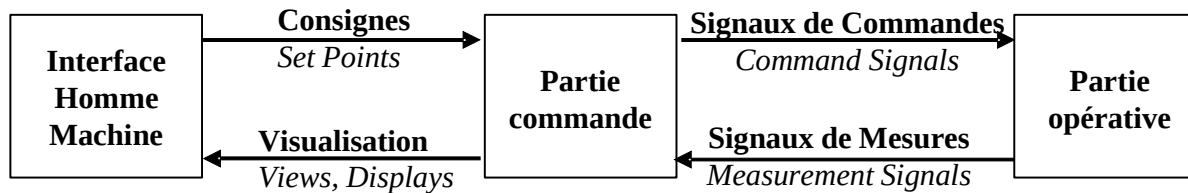
Cours  
Inversés

# I-1 Structure d'un Système Automatisé de Production (SAP)

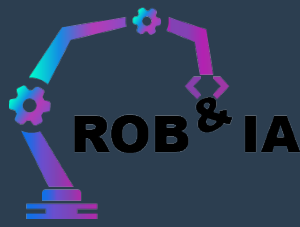


Une installation industrielle automatisée comporte :

- une partie opérative ,
- une partie commande,
- une partie Interface Homme-Machine (IHM)



# Historique



- **Avant : automatismes réalisés à l'aide d'une logique câblée**
  - Modifications = recâblage
  - Extensions = armoire + grande
  - Difficile à dépanner



- **Après : Automates Programmables Industriels (API)**
  - Modification = changement du programme
  - Extensions = carte additionnelle
  - facile à dépanner

PLC en anglais

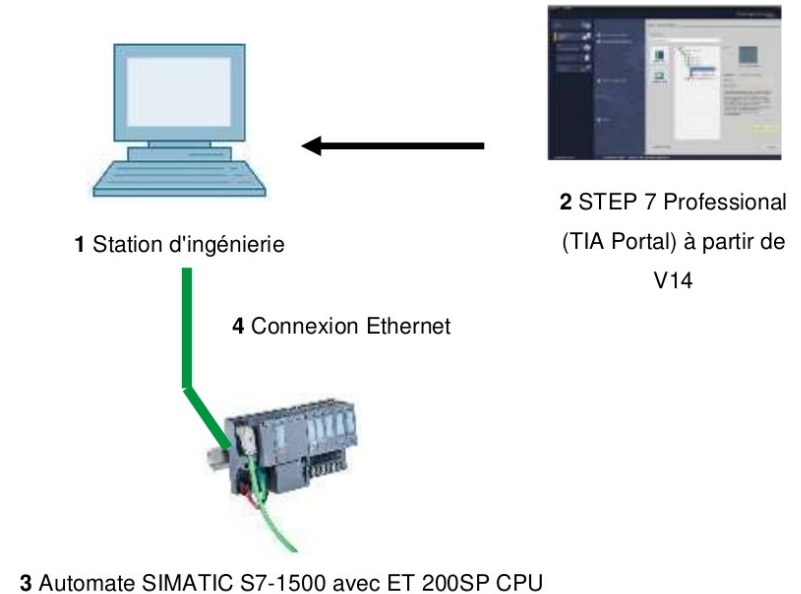


*PLC : Programmable Logic Controller*

- **Programmation**

- Depuis un poste d'ingénierie (*Engineering*)
- Transfert du programme
- Vue en ligne pour debug

**/!\ L'automate fonctionne en production sans le poste d'ingénierie**





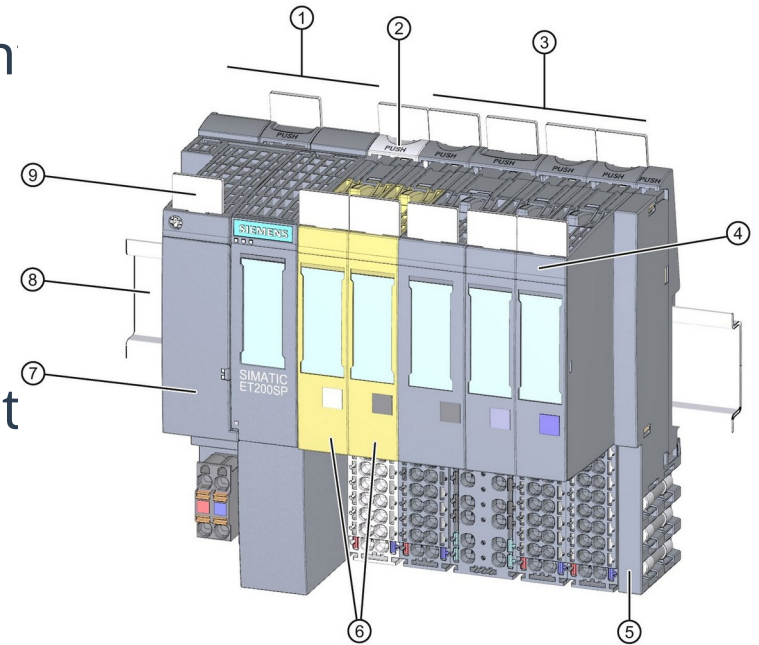
- **Compacts**

- regroupent les principaux composants dans un espace réduit
- Difficilement extensibles



- **Modulaires**

- peuvent être configurés individuellement selon les besoins
- facilement extensibles
- différents modules encliquetés sur un profilé support. L'unité centrale (CPU) est reliée aux autres modules par un bus interne.



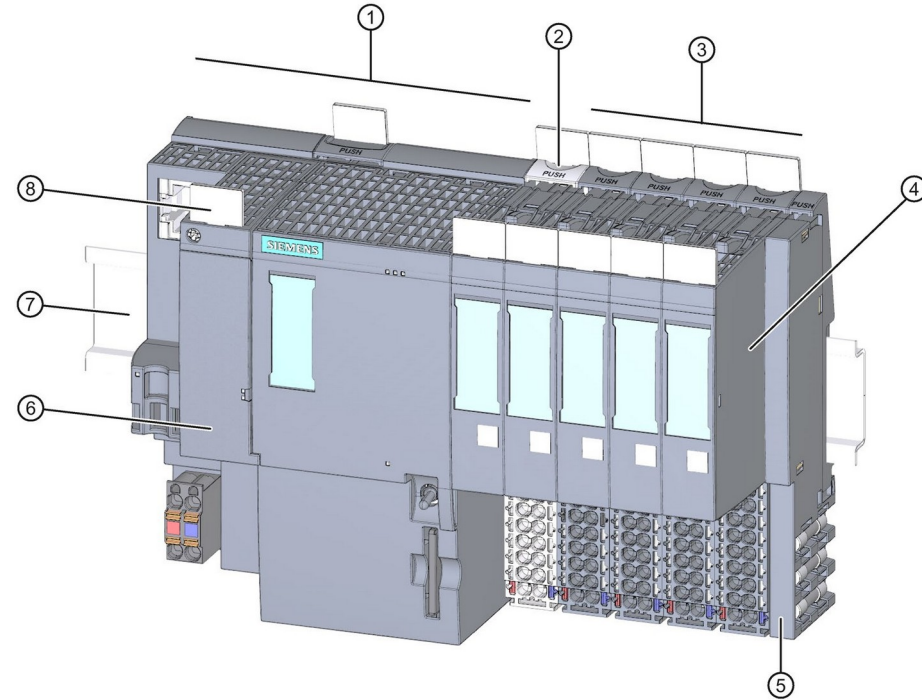
# Différents types d'API



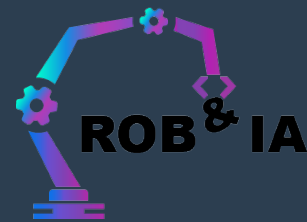
- **Modulaires**

- Exemple : ET200sp

1. CPU/module d'interface
2. BaseUnit BU..D claire avec arrivée de l'alimentation
3. BaseUnits BU..B foncées pour le prolongement du groupe de potentiel
4. Module de périphérie
5. Module serveur (fourni avec la CPU/le module d'interface)
6. BusAdapter
7. Profilé support
8. Etiquette de repérage



# Composition d'un API



## • Alimentation :

- convertit la tension alternative du réseau 230V en 24V continu,
- fournit l'énergie non seulement pour l'automate mais aussi pour les capteurs et actionneurs raccordés

## • Unité centrale (CPU) :

- mémorise le programme (carte SD + RAM),
- exécute le programme
- un commutateur de mode permet (marche(ou *RUN*) / arrêt (ou *STOP*))

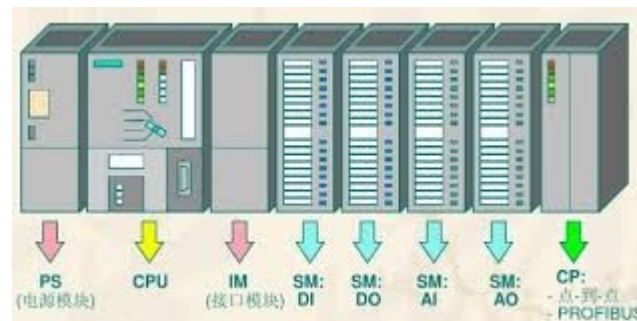
*Central  
Processing Unit*

## • Cartes spécialisées

- Communication
- Pesage
- ...

## • Cartes d'Entrées / Sorties

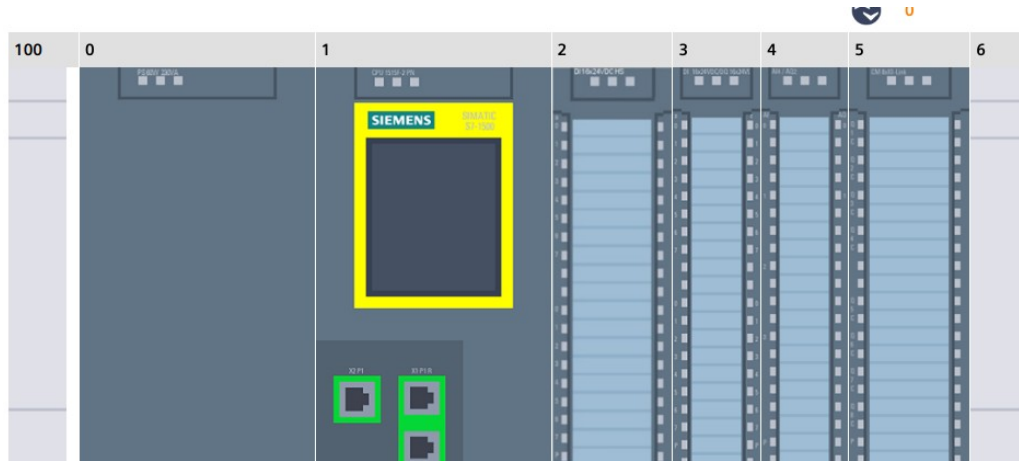
- Tout Ou Rien
- Analogiques



# Composition d'un API

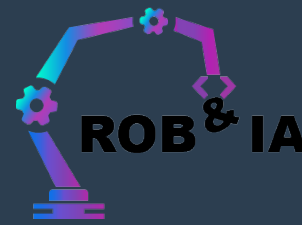


- **S7-1500**



<b>Alimentation</b>	0	S7-1500, PS 60W 120/230V AC/DC	6ES7507-0RA00-0AB0
<b>CPU</b>	1	CPU 1515F-2 PN, 1,5MB Prog., 4,5MB Data	6ES7515-2FN03-0AB0
<b>Carte entrées logiques</b>	2	S7-1500, DI 16X24VDC HS	6ES7521-7BH00-0AB0
<b>Carte entrées/sorties logiques</b>	3	S7-1500, DI 16x24VDC/DQ 16x24VDC/0.5A BA	6ES7523-1BL00-0AA0
<b>Carte entrées/sorties analogiques</b>	4	S7-1500, AI 4X U/I/RTD/TC/AQ 2X U/I ST	6ES7534-7QE00-0AB0
<b>Carte de communication</b>	5	S7-1500, CM 8xIO-Link	6ES7547-1JF00-0AB0

# Types de Signaux



- **Signal analogique (*Analog Signal*)** : une infinité de valeurs.

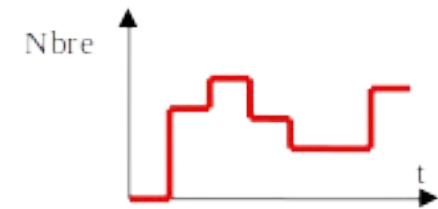
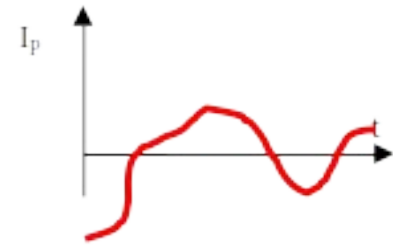
Exemples ?

- **Signal numérique (*Digital Signal*)** : un nombre fini de valeurs.

Exemples

- **Signal logique Tout Ou Rien (T.O.R.) (*Digital Signal / Bool*)**
  - Un cas particulier du signal numérique
  - ne peut prendre que deux valeurs : 0 ou 1, Vrai ou Faux, Ouvert ou Fermé.

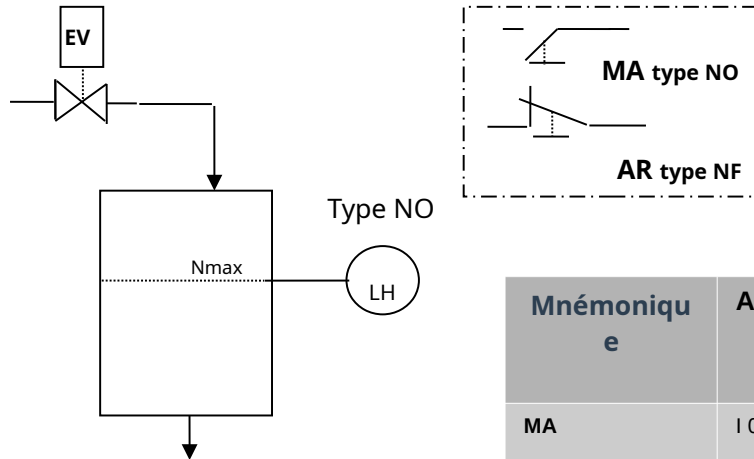
Exemples ?



**La norme CEI 61131-3 classe les langages de programmation des API en 3 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)

## Exemple



Mnémonique	Adresse API	Commentaire
MA	I 0.0	BP marche type NO
AR	I 0.1	BP arrêt type NF
LH	I 0.2	Détecteur niveau haut type NO
EV	Q 4.0	Electrovanne type NF



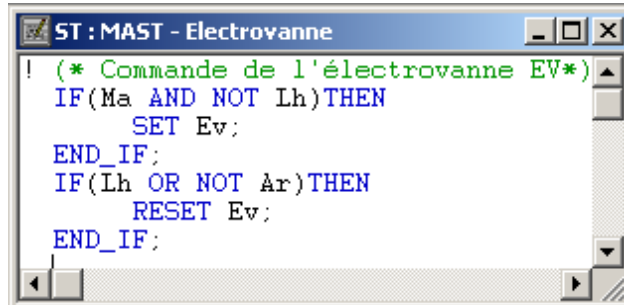
- Langage IL  $\approx$  assembleur

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

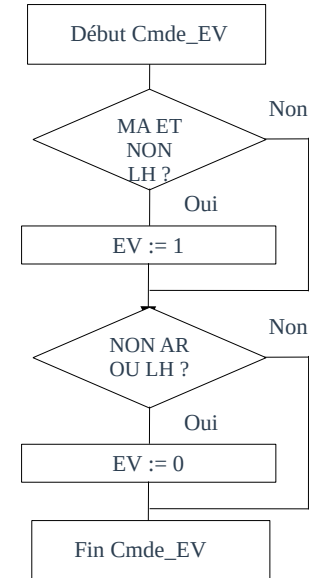
```
A(  
O      "MA"          IO.0      -- BP marche type NO  
O      "EV"          Q4.0      -- Electrovanne type NF  
)  
A      "AR"          IO.1      -- BP arrêt type NF  
AN     "LH"          IO.2      -- Niveau haut type NO  
=      "EV"          Q4.0      -- Electrovanne type NF
```

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

≈ Basic



```
! (* Commande de l'électrovanne EV*)  
IF(Ma AND NOT Lh)THEN  
    SET Ev;  
END_IF;  
IF(Lh OR NOT Ar)THEN  
    RESET Ev;  
END_IF;
```

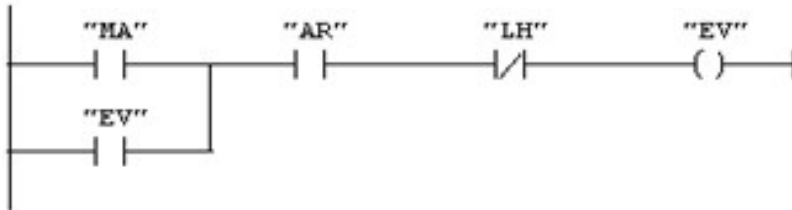


- Langage Ladder (LD)

Inspiré des schémas électriques

**Réseau 1:** Commande electrovanne

Automatisation de l'électrovanne de remplissage



Informations mnémotechnique:

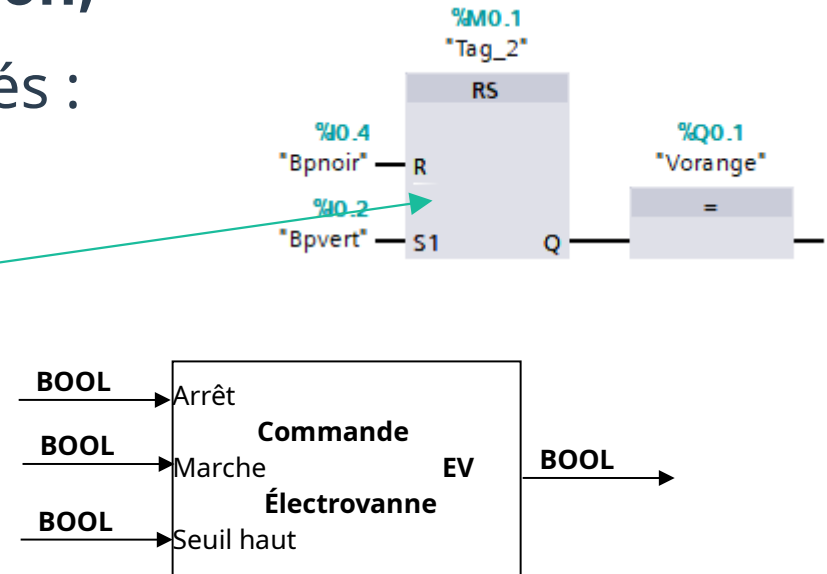
I0.0	MA	BP marche type NO
Q4.0	EV	Electrovanne type NF
I0.1	AR	BP arrêt type NF
I0.2	LH	Niveau haut type NO

- Langage Function Block Diagram (FBD)

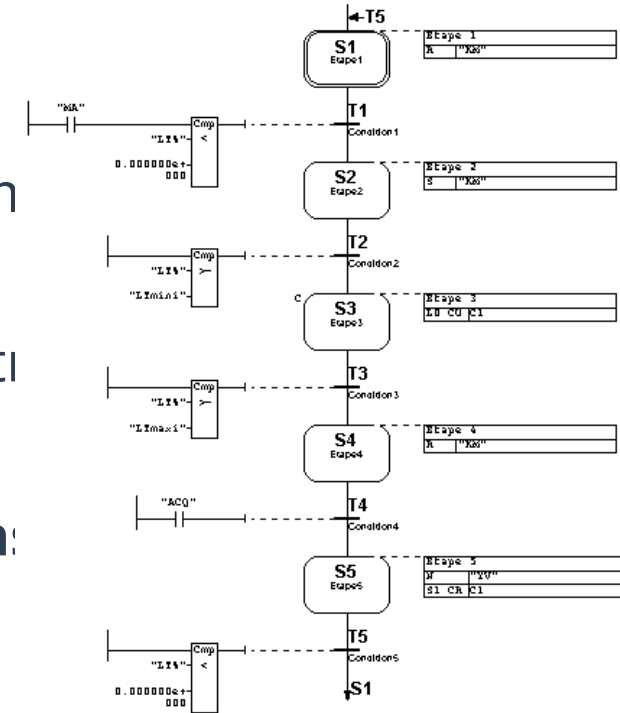
- fonctions **préprogrammées** ou non,
- Exemples de blocs préprogrammés :

- Temporisations
- Compteurs
- Bascules
- ....

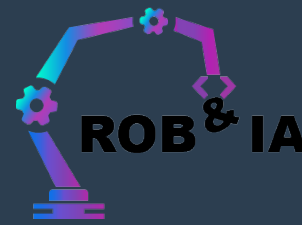
- Exemple de bloc utilisateur :



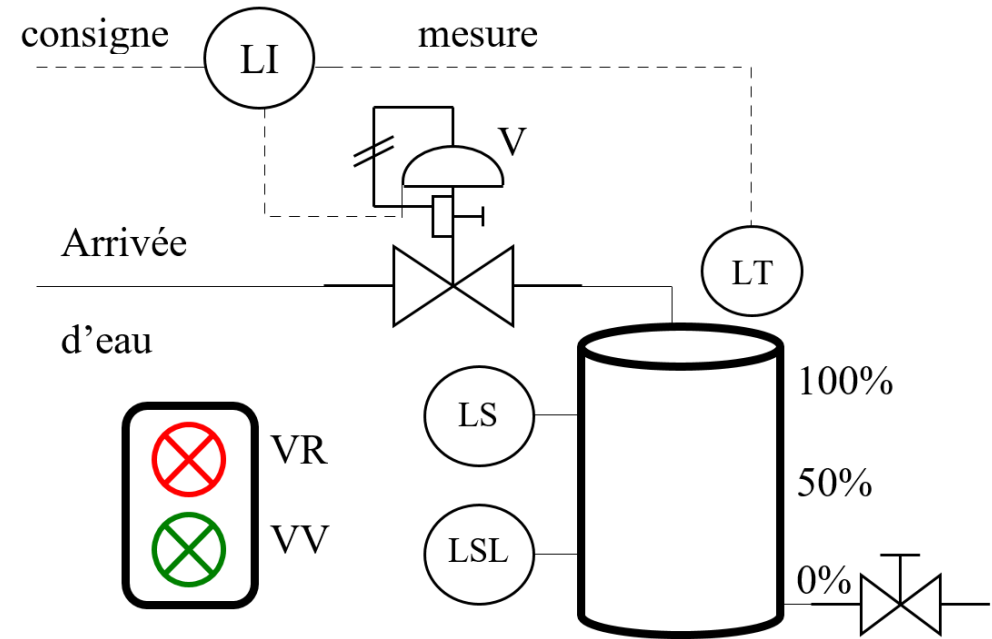
- **Langage Sequential Function Chart (SFC)**
  - ne pas confondre avec la description du comportement d'un système (connu sous le nom de GRAFCET en France, voir norme CEI 60848)
- un ensemble d'**étapes** et de **transitions** reliés entre elles par des **liaisons dirigées**.
- Chaque étape est associée à un ensemble d'**actions**.
- Chaque transition est associée à une **condition de transition**.



# Exemple d'application



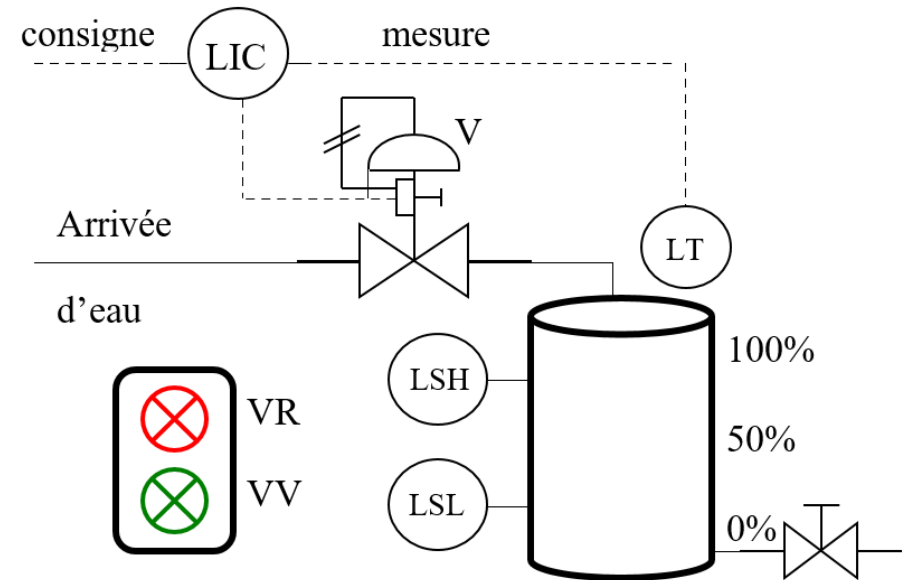
- On souhaite réaliser :
  - Une régulation de niveau.
  - Un affichage de sécurité sur les niveaux
- Entourer sur le schéma les éléments intervenant dans la **régulation analogique de niveau**
- Entourer sur le schéma les éléments intervenant dans l'affichage de sécurité : c'est l'**automatisme logique T.O.R**



# Exemple d'application



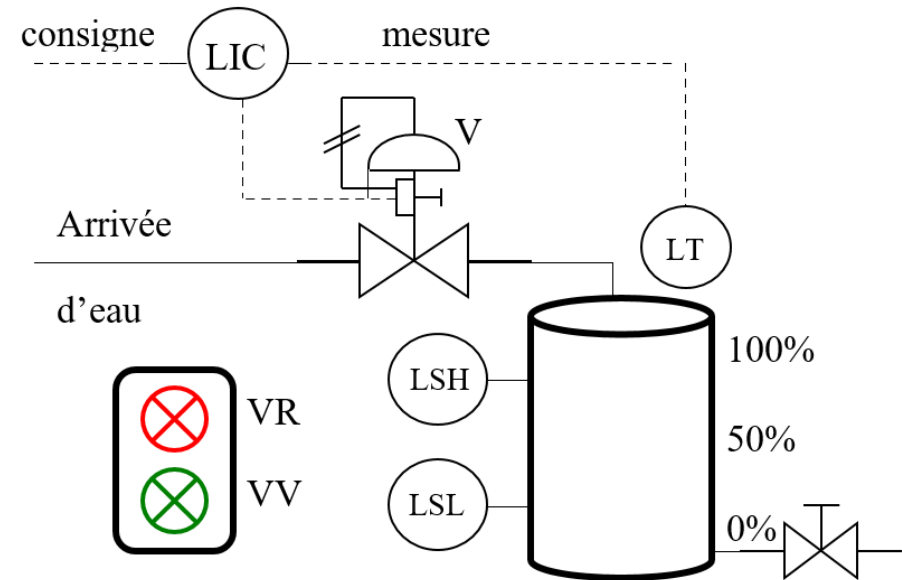
- Le détecteur de niveau haut **LSH** est de technologie **NF** et le détecteur de niveau bas **LSL** est **NO**.
- Que veut dire NO (NO) et NF (NC) ?
- Représenter le schéma de principe de câblage de ces entrées automate (24V).



# Exemple d'application



Niveau	LSL (NO)		
	Etat physique capteur	Tension entrée API	Etat logique variable API
0%	Repos	0 Volt	$Niv\_LSL = 0$
50%	Actionné	24 Volt	$Niv\_LSL = 1$
100%	Actionné	24 Volt	$Niv\_LSL = 1$

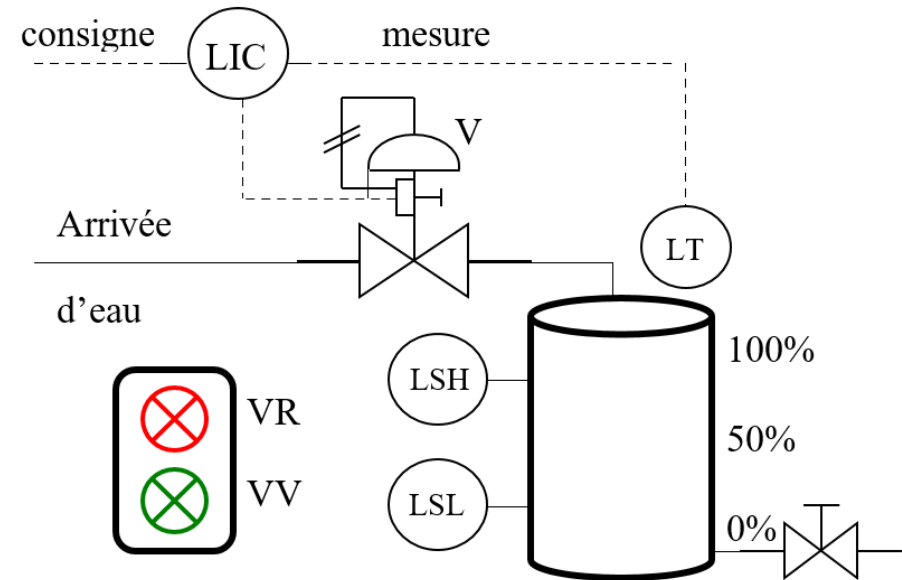




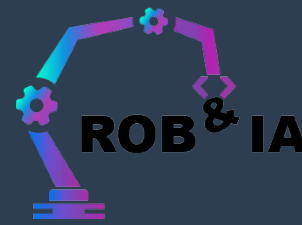
# Exemple d'application



Niveau	LSH (NF)		
	Etat physique capteur	Tension entrée API	Etat logique variable API
0%	Repos	24 Volt	$Niv\_LSL = 1$
50%	Repos	24 Volt	$Niv\_LSL = 1$
100%	Actionné	0 Volt	$Niv\_LSL = 0$

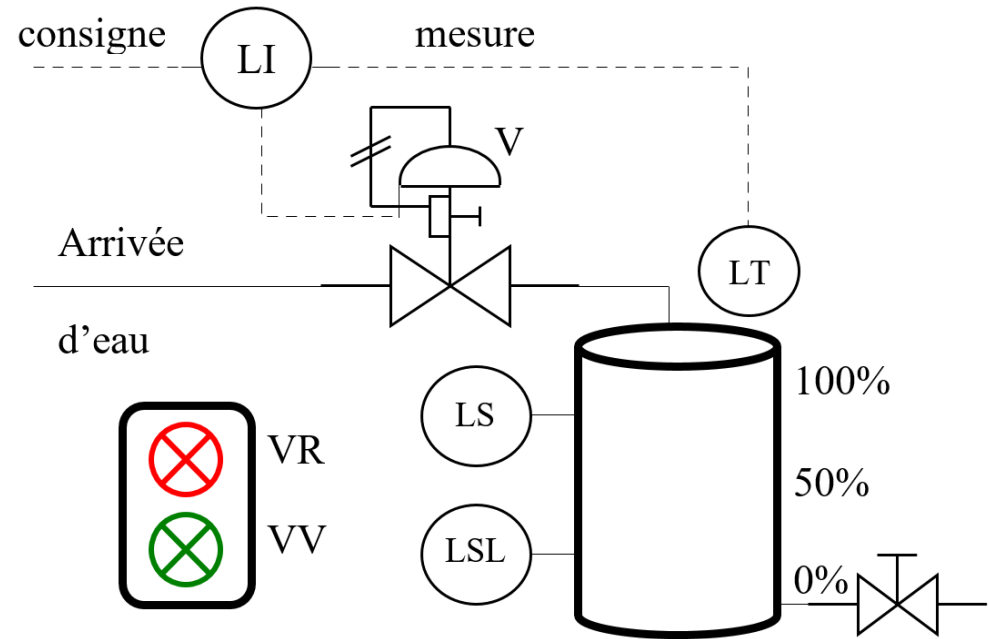


# Exemple d'application



## • Programme ladder

- Allumer le voyant rouge si niveau  $>$  au seuil de LSH.
- Allumer le voyant vert si niveau  $<$  au seuil de LSL.

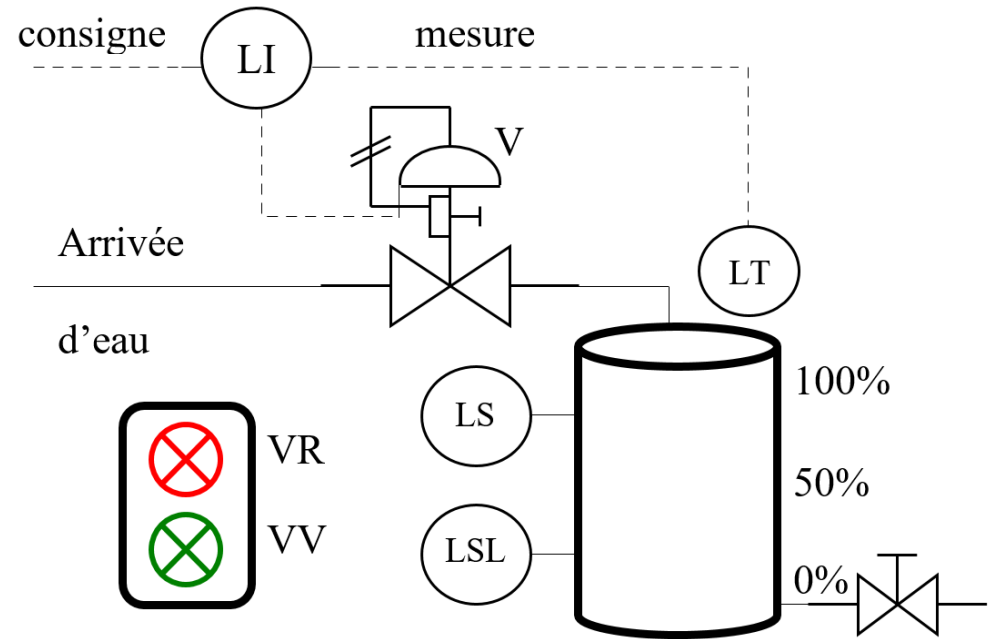
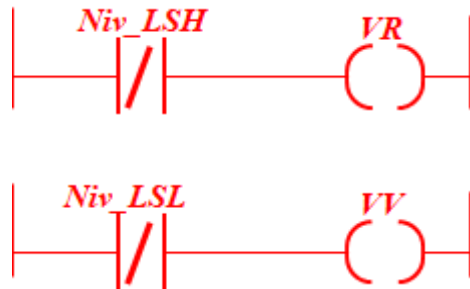


# Exemple d'application



## • Programme ladder

- Allumer le voyant rouge si niveau  $>$  au seuil de LSH.
- Allumer le voyant vert si niveau  $<$  au seuil de LSL.





**Fin**

Bibliographie :

Boujat G., Anaya P. : Automatique industrielle en 20 fiches, Dunod

Documentation constructeur Siemens :

<https://mall.industry.siemens.com>