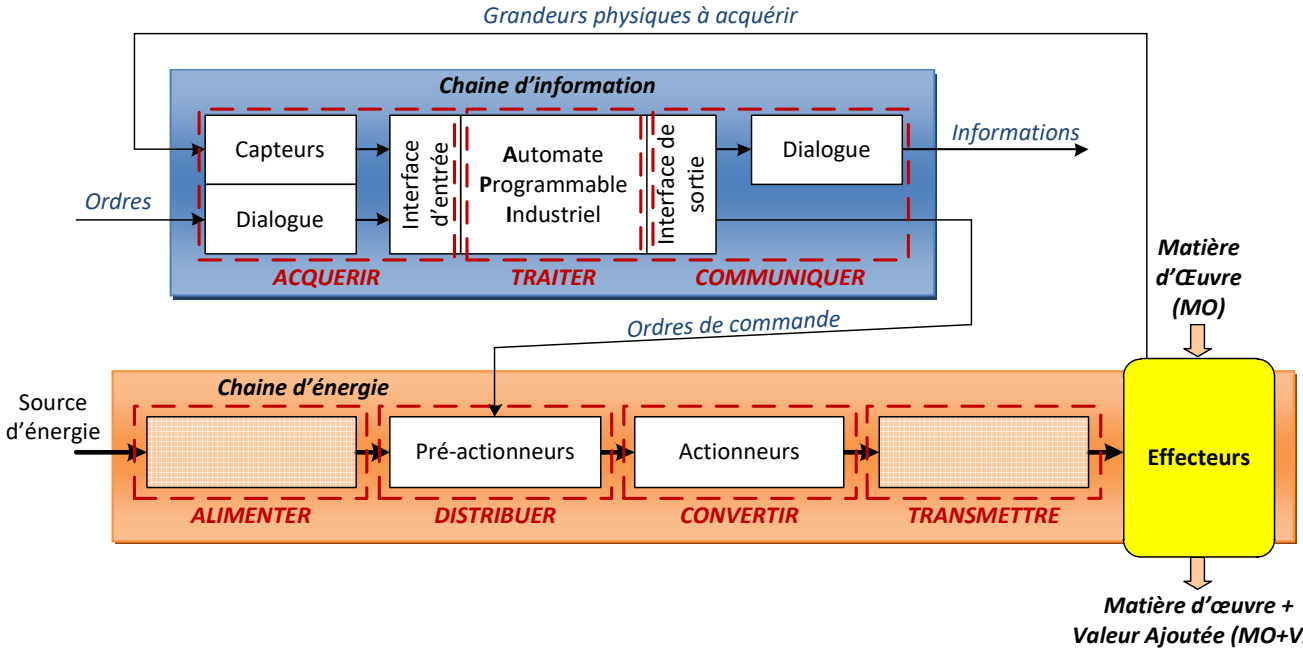
L'AUTOMATE PROGRAMMABLE - PLC

1. Rappels : structure fonctionnelle des systèmes ou produits automatisés

Les principales catégories de constituants assurant les différentes fonctions d'un système automatisé sont les suivantes :



Fonction acquérir :

- Commandes et consignes : .....
  .....
- Informations : .....
  .....

Fonction traiter :

- Logique câblée : .....
  .....
- Logique programmée : .....
  .....
  .....

Fonction communiquer :

- Messages et visualisation : .....
- Ordres de commande : .....
  .....
  .....

2. L'automate programmable ou PLC « Programmable Logic Controller »

2.1. Définition d'un automate programmable

Selon la norme NFC 63-850 : un automate est un appareil électronique qui comporte une mémoire, programmable par un utilisateur automaticien à l'aide d'un langage adapté, pour le stockage interne des instructions composant les fonctions d'automatisme comme par exemple :

- Logique séquentielle et combinatoire ;
- Temporisation, comptage, décomptage, comparaison ;
- Calcul arithmétique ;
- Réglage, asservissement, régulation, etc,

.....

.....

.....

ZELIO de chez Schneider

EM4 de chez Crouzet

S71200 de chez Siemens

2.2. Avantages et inconvénients

Les automates présentent de nombreux intérêts :

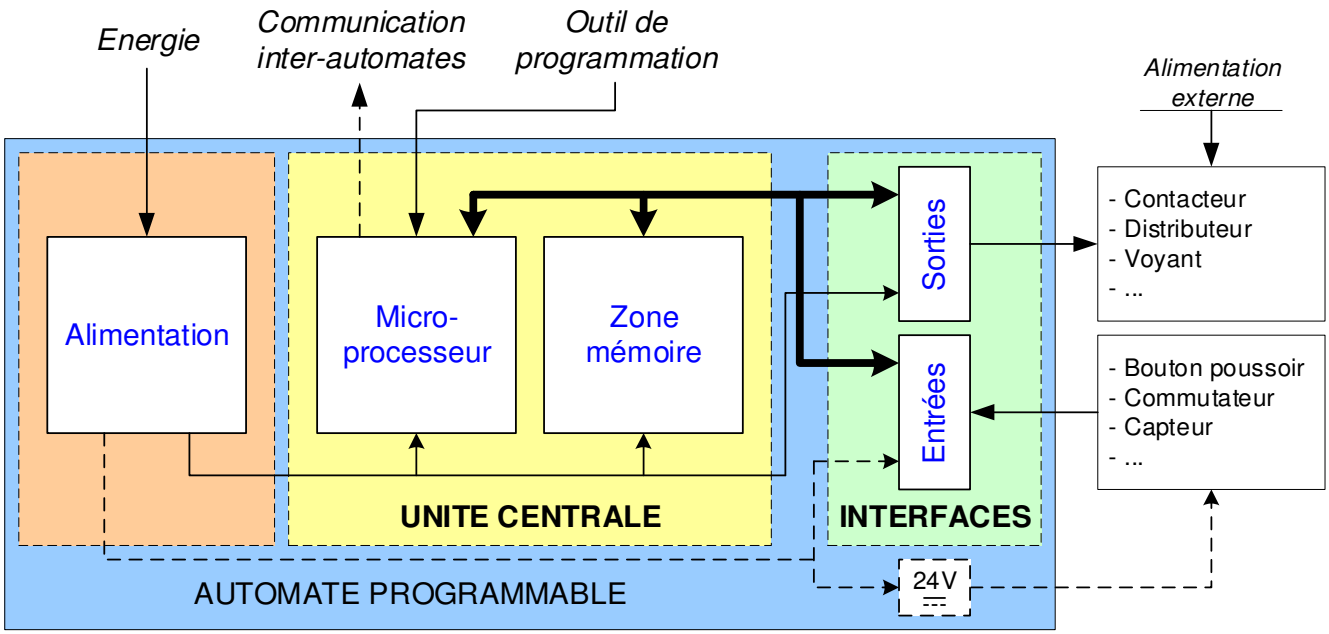
- ..... (poussière environnante, perturbations électromagnétiques, vibrations des supports, variations de température...)
- .....

  Les envois et réceptions de signaux se font très rapidement avec l'environnement... Avec de plus une exécution séquentielle cyclique sans modification de mémoire, ils permettent d'assurer un temps d'exécution minimal, respectant un déterminisme temporel et logique, garantissant un temps réel.

Ils restent à l'heure actuelle les seules plateformes d'exécution considérées comme fiables en milieu industriel (avec les ordinateurs industriels). De plus ils nécessitent la maîtrise de langages spécifiques conformes à la norme CEI 61131-3 qui reprennent dans leur forme la logique d'exécution interne de l'automate. Ces langages apparaissent toutefois à beaucoup d'utilisateurs plus accessibles et plus visuels que les langages informatiques classiques.

2.3. Structure des automates programmables :

Un automate programmable reçoit les informations relatives à l'état du système et commande les pré-actionneurs suivant un ordre de séquencement défini dans une liste d'instructions qui lui sera communiquée.



.....

- Elle fournit à partir des tensions usuelles (230 V, 24 V = ou ~) les tensions continues nécessaires au fonctionnements des circuits électroniques.
- Elle fournit parfois aussi l'alimentation nécessaire au fonctionnement des entrées.

.....

- Le **microprocesseur** (unité de traitement) réalise toutes les fonctions logiques, les fonctions de temporisation, de comptage, de calcul … à partir d'un programme contenu dans sa mémoire.

  Il possède des voies de communication avec l'extérieur :
  - dialogue avec l'outil de programmation,
  - raccordement sur un réseau de communication inter-automates.
- La **zone mémoire** contient, la liste d'instructions qui constituent le programme de fonctionnement du système, ainsi que toutes les données utiles au fonctionnement interne de l'automate.

.....

- Les **entrées** reçoivent des informations en provenance des éléments de détection et du pupitre opérateur.
- Les **sorties** transmettent des informations aux pré-actionneurs et aux éléments de signalisation du pupitre.

3. Langages de programmation pour automate : Norme IEC 1131-3

La norme définit cinq langages qui peuvent être utilisés pour la programmation des automates. Ces cinq langages sont :

- ..... ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (vraie/faux).
- ..... ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.
- ..... ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions logiques. Il permet de manipuler tous les types de variables.
- ..... ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels. Il se rapproche du GRAFCET (Graphe Fonctionnel de Commande des Étapes et Transitions) , mode de représentation et d'analyse d'un automatisme.
- ..... ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

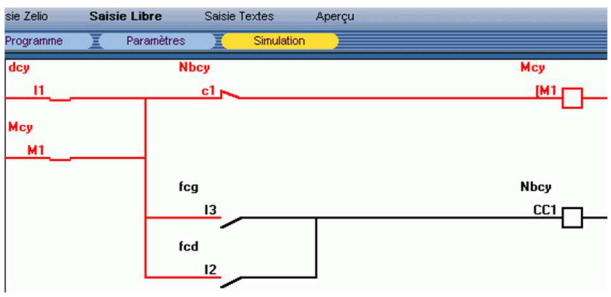
3.1. Objets communs

Toute expression, constante ou variable, utilisée dans un programme doit être caractérisée par un type :

- ..... ce type donne la valeur TOR qui sont équivalent à « 1 » ou « 0 ».
- ..... c'est un nombre entre -2147483647 et +2147483647. Il est exprimé dans l'une des bases suivantes : décimale, hexadécimale, octale ou binaire.
- ..... en base 10. Il peut prendre 1 bit de signe.
- ..... , souvent codé en hexadécimal
- ..... c'est une valeur strictement positive et commence par T#
- ..... elle doit être précédée et suivie par une apostrophe, et ne doit jamais excéder 255 caractères).

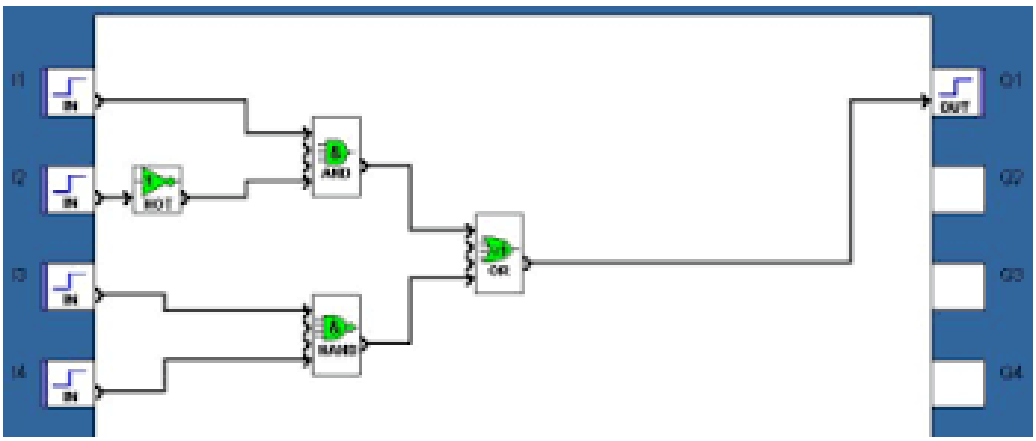
3.2. Langage LD

Le langage LD (ladder diagram) est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation.



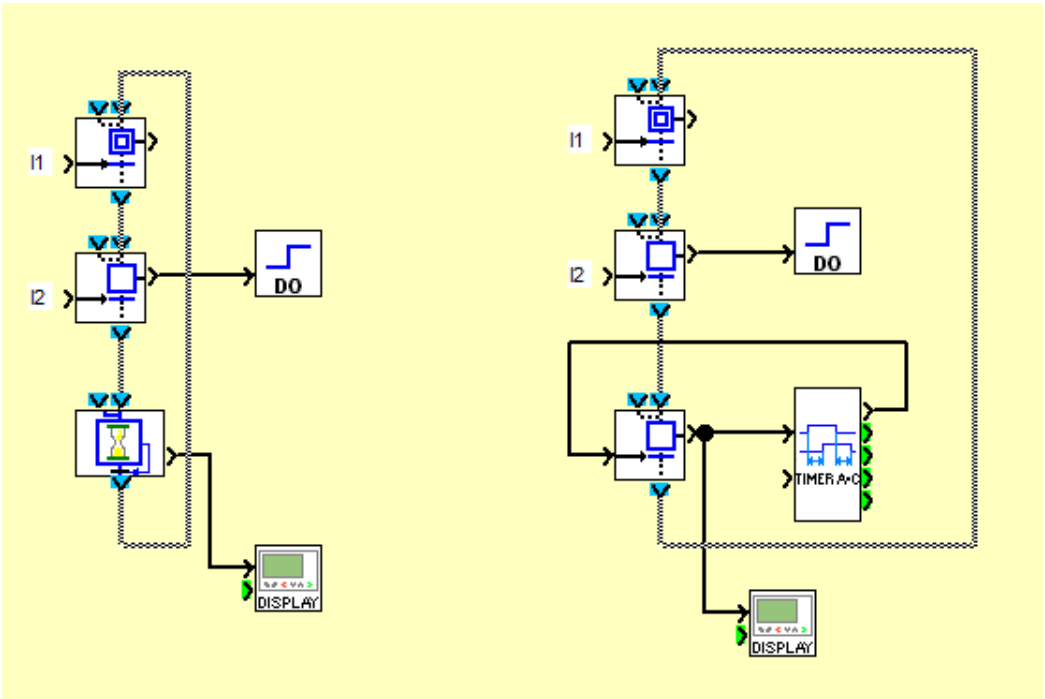
3.3. Langage FBD

Le langage FBD (function block diagram) est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standard, de fonctions ou de blocs fonctionnels issus de la logique combinatoire.



3.4. Langage SFC

Le langage SFC (Sequential Function Chart), ou GRAFCET, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD.



3.5. Table d'adressage.

Les variables sont déclarées en utilisant un nom symbolique et une adresse logique :

%	Attribut	Type
	I: Entrée Q : Sortie M : mémoire K : constante	x : Boolean, 1 bit B : Byte, Octet, 8 bits W : Word, Mot,16 bits D : Double Word, 32 bits F : Floating, Flottant

Ex : %I1.5 Entrée TOR n°5 de la carte n°1 du rack par défaut n°0
%Q2.4.F Sortie TOR voie F du module 4 du rack 2
%M128 bit interne n°128
%MW4 entier 16 bits n°4

4. Raccordement d'un automate

4.1. Généralités

Conformément à la structure de la chaine d'information, on raccorde à l'automate :

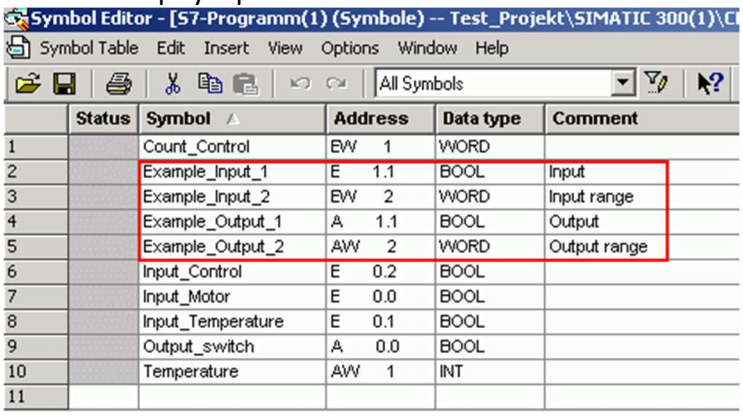
Les entrées (input) : Tout élément transmettant :

- des ordres en provenance de l'opérateur (boutons poussoirs, commutateurs, coups de poings d'arrêt d'urgence…),
- des informations en provenance du système lui-même (capteurs, relais thermiques…).

Les sorties (output) : Tout élément transmettant :

- soit des informations en direction de l'opérateur (voyants, afficheurs…),
- soit des ordres en direction du système lui-même (contacteurs, distributeurs…).

On créer souvent alors la table d'adressage, appelée aussi table des mnémoniques utilisés qui est la correspondance entre les éléments physique du schéma et les adresses de l'automate :



	Status	Symbol	Address	Data type	Comment
1		Count_Control	EW 1	WORD	
2		Example_Input_1	E 1.1	BOOL	Input
3		Example_Input_2	EW 2	WORD	Input range
4		Example_Output_1	A 1.1	BOOL	Output
5		Example_Output_2	AW 2	WORD	Output range
6		Input_Control	E 0.2	BOOL	
7		Input_Motor	E 0.0	BOOL	
8		Input_Temperature	E 0.1	BOOL	
9		Output_switch	A 0.0	BOOL	
10		Temperature	AW 1	INT	
11					

4.2. Raccordement des entrées/sorties TOR

Exemple de raccordement d'un bouton poussoir S1 sur l'entrée 1 (I1) en 24 VDC et d'un relais KM1 en 24VDC sur la sortie 2 (O2), d'un voyant HCY sur la sortie (O3), en utilisant un automate en 24 VDC.

Table d'adressage (tableau mnémonique) :

ENTREES		
Mnémonique	Commentaire / Description	Adresse
S1	Bouton poussoir Marche	I1
SORTIES		
Mnémonique	Commentaire / Description	Adresse
KM1	Relais de commande moteur	O2
HCYC	Voyant cycle en cours	O3

Schéma