

# Les Automates Programmables Industriels (API)

## 1- Introduction

Les **Automates Programmables Industriels (API)** sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

Un Automate Programmable Industriel (**API**) est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels. Un **automate programmable** est adaptable à un maximum d'application, d'un point de vue traitement, composants, langage. C'est pour cela qu'il est de construction modulaire.

Il est en général manipulé par un personnel électromécanicien. Le développement de l'industrie à entraîner une augmentation constante des fonctions électroniques présentes dans un automatisme c'est pour ça que l'API s'est substitué aux armoires à relais en raison de sa souplesse dans la mise en œuvre, mais aussi parce que dans les coûts de câblage et de maintenance devenaient trop élevés.

## 2- Pourquoi l'automatisation ?

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée par le système. Ces éléments sont exprimables en termes d'objectifs par :

- Accroître la productivité (rentabilité, compétitivité) du système
- Améliorer la flexibilité de production ;
- Améliorer la qualité du produit
- Adaptation à des contextes particuliers tel que les environnements hostiles pour l'homme (milieu toxique, dangereux.. nucléaire...), adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées...),
- Augmenter la sécurité, etc...



• Figure1 : Automate programmable Module STEP7-300

## 3– Structure générale des API :

Les caractéristiques principales d'un automate programmable industriel (**API**) sont :  
coffret, rack, baie ou cartes

- Compact ou modulaire
- Tension d'alimentation
- Taille mémoire
- Sauvegarde (EPROM, EEPROM, pile, ...)
- Nombre d'entrées / sorties
- Modules complémentaires (analogique, communication,..)
- Langage de programmation

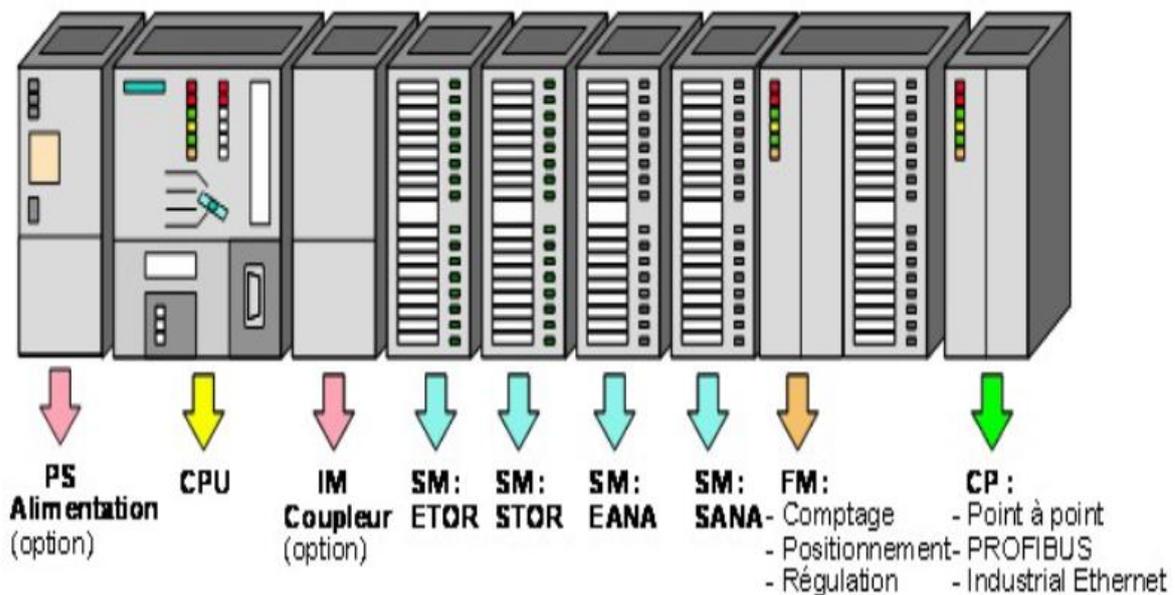


Figure 2 : Disposition des modules de l'APIS7-300

Des API en boîtier étanche sont utilisés pour les ambiances difficiles (température, poussière, risque de projection ...) supportant ainsi une large gamme de température, humidité ... L'environnement industriel se présente sous trois formes :

- environnement physique et mécanique (poussières, température, humidité, vibrations);
- pollution chimique ;
- perturbation électrique. (parasites électromagnétiques)

les deux types des automates compact et modulaire :



➤ PLC-200 Programmable Logic Controller (S.



Figure3 : Les automates compact et modulaire

## 4- Structure interne d'un automate programmable industriel (API) :

Les API comportent quatre parties principales (Figure 4.4) :

- Une unité de traitement (un processeur CPU);
- Une mémoire ;
- Des modules d'entrées-sorties ;
- Des interfaces d'entrées-sorties ;
- Une alimentation 230 V, 50/60 Hz (AC) - 24 V (DC).

La structure interne d'un **automate programmable industriel (API)** est assez voisine de celle d'un système informatique simple, L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

Deux types de mémoire cohabitent :

- **La mémoire Programme** où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte)

- **La mémoire de données** utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle fait partie du système entrées-sorties. Elle fige les valeurs (0 ou 1) présentes sur les lignes d'entrées, à chaque prise en compte cyclique de celle-ci, elle mémorise les valeurs calculées à placer sur les sorties.

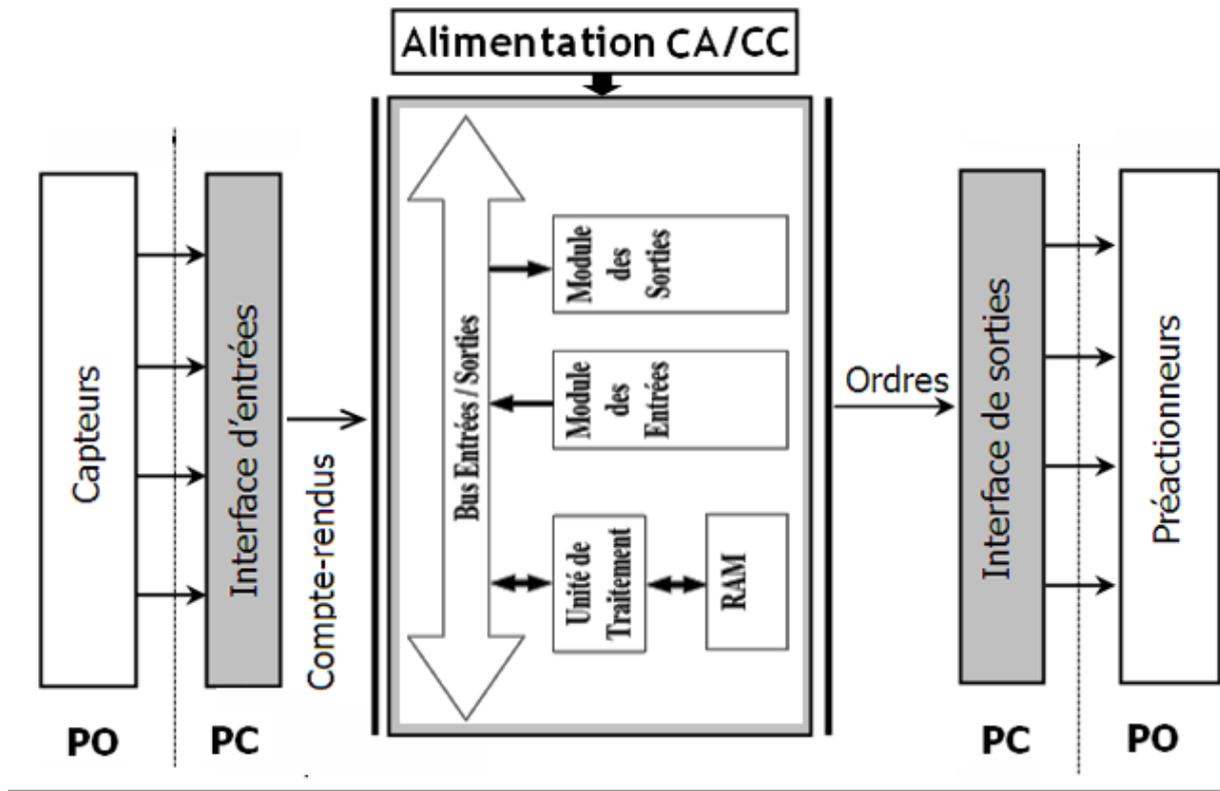
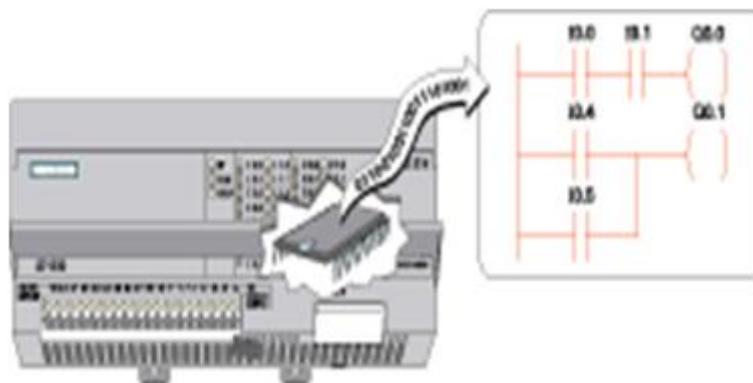


Figure4 : Structure interne d'un automate programmable industriel (API)

## 5- Description des éléments d'un API :

### 5.1- La mémoire :

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système que sont le terminal de programmation (PC ou console) et le processeur, qui lui gère et exécute le programme. Elle reçoit également des informations en provenance des capteurs.



existe dans les automates deux types de mémoires qui remplissent des fonctions différentes :

- La mémoire Langage où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte)
- La mémoire Travail utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle s'efface automatiquement à l'arrêt de l'automate (nécessite une batterie de sauvegarde).

Répartition des zones mémoires :

- Table image des entrées
- Table image des sorties
- Mémoire des bits internes
- Mémoire programme d'application

## **5.2- Le processeur :**

Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'entrées et de sorties et d'autre part à exécuter les instructions du programme.

## **5.3- Les interfaces et les cartes d'Entrées / Sorties:**

### **5.3.1. Les interfaces d'entrées-sorties :**

Les entrées/sorties TOR (Tout ou Rien) assurent l'intégration directe de l'automate dans son environnement industriel en réalisant la liaison entre le processeur et le processus. Elles ont une double fonction :

Une fonction d'interface pour la réception et la mise en forme de signaux provenant de l'extérieur (capteurs, boutons poussoirs, etc.) et pour l'émission de signaux vers l'extérieur (commande de pré-actionneurs, de voyants de signalisation, etc.).

La conception de ces interfaces avec un isolement galvanique ou un découplage opto-électronique assure la protection de l'automate contre les signaux parasites.

Une fonction de communication pour l'échange des signaux avec l'unité centrale par l'intermédiaire du bus d'entrées/sorties. L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses. L'interface de sortie comporte de la même façon des adresses de sortie. Chaque préactionneur est relié à une de ces adresses. Le nombre de ces entrées est sorties varie suivant le type d'automate. Les cartes d'E/S ont une modularité de 8, 16 ou 32 voies. Les tensions disponibles sont normalisées (24, 48, 110 ou 230 V continu ou alternatif ...).

### **5.3.2- Cartes d'entrées :**

Elles sont destinées à recevoir l'information en provenance des capteurs et adapter le signal en le mettant en forme, en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative.

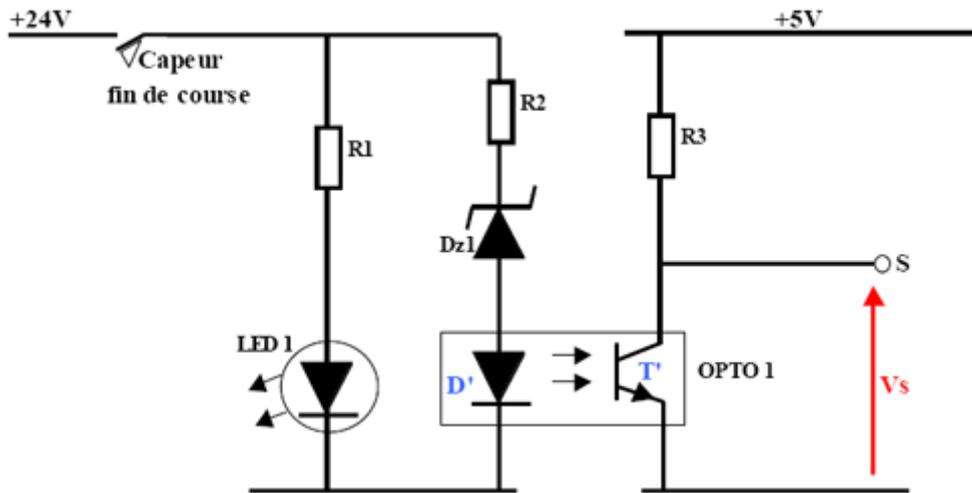


Figure 5: Exemple d'une carte d'entrées typique d'un API

### 5.3.1- Cartes de sorties:

Elles sont destinées à commander les pré-actionneurs et éléments des signalisations du système et adapter les niveaux de tensions de l'unité de commande à celle de la partie opérative du système en garantissant une isolation galvanique entre ces dernières

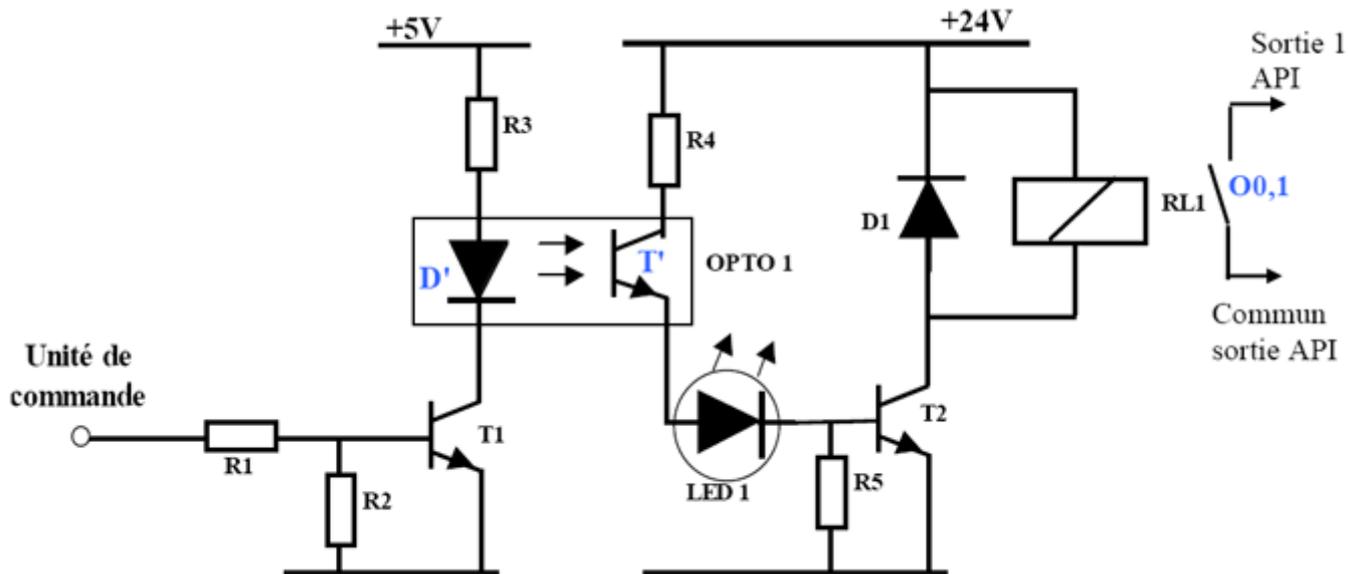


Figure 6: Exemple d'une carte de sortie typique d'un API

### 5.4- Exemple de cartes :

- **Cartes de comptage rapide** : elles permettent d'acquérir des informations de fréquences élevées incompatibles avec le temps de traitement de l'automate. (signal issu d'un codeur de position)
- **Cartes de commande d'axe** : Elles permettent d'assurer le positionnement avec précision d'élément mécanique selon un ou plusieurs axes. La carte permet par exemple de piloter un

servomoteur et de recevoir les informations de positionnement par un codeur. L'asservissement de position pouvant être réalisé en boucle fermée.

- **Cartes d'entrées / sorties analogiques** : Elles permettent de réaliser l'acquisition d'un signal analogique et sa conversion numérique (CAN) indispensable pour assurer un traitement par le microprocesseur. La fonction inverse (sortie analogique) est également réalisée. Les grandeurs analogiques sont normalisées : 0-10V ou 4-20mA.

- **Cartes de régulation PID**

- **Cartes de pesage**

- **Cartes de communication** (RS485, Ethernet ...)

- **Cartes d'entrées / sorties déportées**

**5.5. Le Bus** C'est un ensemble de conducteurs qui réalisent la liaison entre les différents éléments de l'automate. Dans un automate modulaire, il se présente sous forme d'un circuit imprimé situé au fond du bac et supporte des connecteurs sur lesquels viennent s'enficher les différents modules : Processeur, extension mémoire, interfaces et coupleurs.

Le bus est organisé en plusieurs sous ensemble destinés chacun à véhiculer un type bien défini d'informations :

-Bus de données.

-Bus d'adresses.

-Bus de contrôle pour les signaux de service tels que tops de synchronisation, sens des échanges, contrôle de validité des échanges, etc...

-Bus de distribution des tensions issues du bloc d'alimentation.

**5.6- L'alimentation électrique :**

Tous les automates actuels sont équipés d'une alimentation 240 V 50/60 Hz, 24 V DC. Les entrées sont en 24 V DC et une mise à la terre doit également être prévue.

## **6- Fonctionnement :**

L'automate programmable **reçoit** les informations relatives à l'état du système et puis **commande** les pré-actionneurs suivant le programme inscrit dans sa mémoire. Généralement les automates programmables industriels ont un fonctionnement cyclique (Figure).

Le **microprocesseur** réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul... Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons **parallèles** appelées '**BUS**' qui véhiculent les informations sous forme binaire.. Lorsque le fonctionnement est dit synchrone par rapport aux entrées et aux sorties, le cycle de traitement commence par la prise en compte des entrées qui sont figées en mémoire pour tout le cycle.

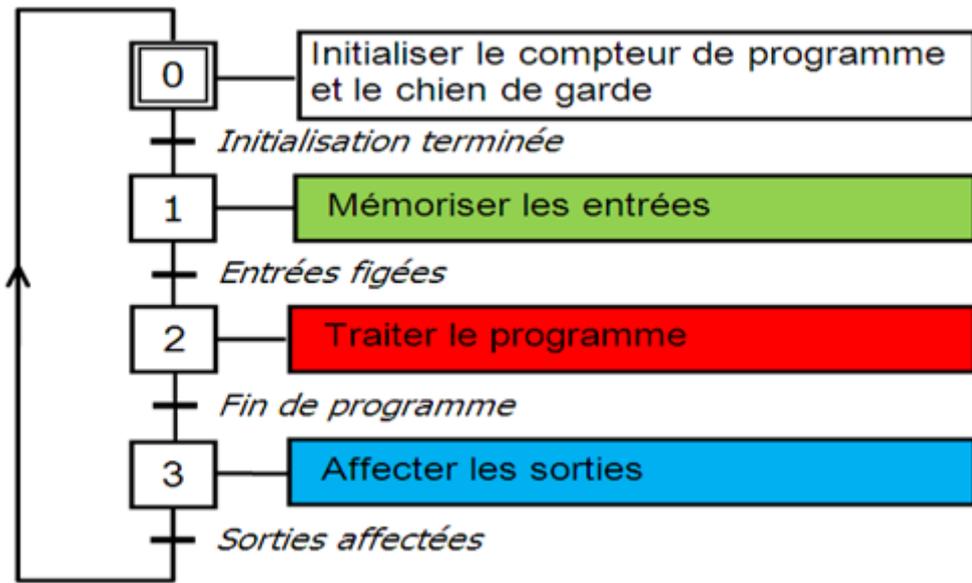


Figure 7 : Fonctionnement cyclique d'un API

Le processeur exécute alors le programme instruction par instruction en rangeant à chaque fois les résultats en mémoire. En fin de cycle les sorties sont affectées d'un état binaire, par mise en communication avec les mémoires correspondantes. Dans ce cas, le temps de réponse à une variation d'état d'une entrée peut être compris entre un ou deux temps de cycle (durée moyenne d'un temps de cycle est de 5 à 15 ms Figure 8).

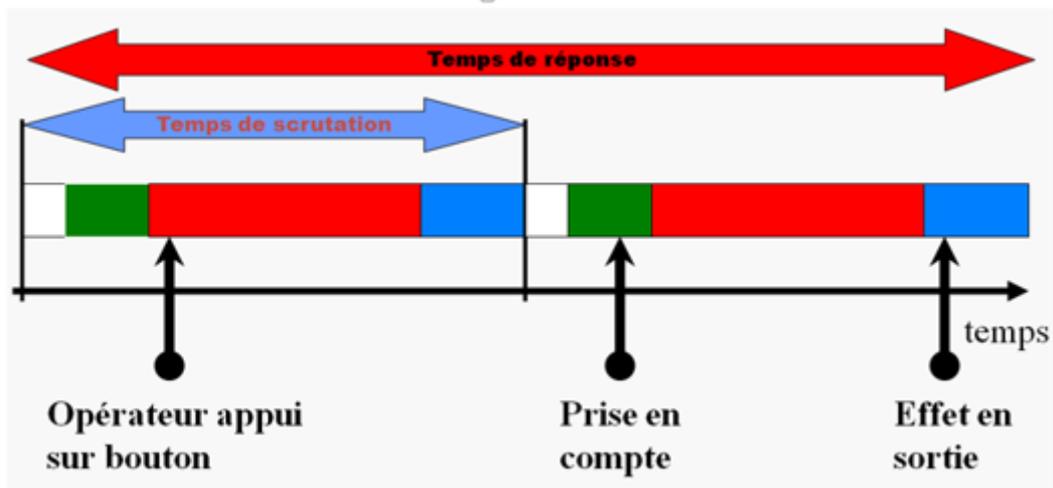


Figure 8 : Temps de scrutation vs temps de réponse

## 7- Jeu d'instructions :

Le processeur peut exécuter un certain nombre d'opérations logiques ; l'ensemble des instructions booléennes des instructions complémentaires de gestion de programme (saut, mémorisation, adressage ...) constitue un jeu d'instructions.

Chaque automate possède ses propres jeux d'instructions. Mais par contre, les constructeurs proposent tous une interface logicielle de programmation répondant à la norme CEI1131-3. Cette norme définit cinq langages de programmation utilisables, qui sont :

- Les langages graphiques :
  - **LD** : **Ladder Diagram** (Diagrammes échelle)
  - **FBD** : **Function Block Diagram** (Logigrammes)
  - **SFC** : **Sequential Function Chart** ( Grafcet)
- Les langages textuels :
  - **IL** : **Instruction List** (Liste d'instructions).
  - **ST** : **Structured Text** (Texte structuré).

-LD (« Ladder Diagram », ou schéma à relais): Ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (vraie/faux).

-IL (« Instruction List », ou liste d'instructions): ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.

-FBD (« Function Block Diagram », ou schéma par blocs): ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables.

-SFC (« Sequential Function Char »): issu du langage GRAFCET, ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels.

-ST («Structured Text » ou texte structuré): ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

## 8- Critères de choix d'un automate

Le choix d'un automate programmable industriel :

Le choix d'un API est adapté aux besoins après l'établissement du cahier des charges. On doit tenir compte de plusieurs critères à savoir :

- Le nombre et la nature d'entrées/sorties intégrés ;
- La nature du traitement (temporisation, comptage, ...) ;
- Les moyens de dialogue et le langage de programmation ;
- La communication avec les autres systèmes ;
- Les moyens de sauvegarde du programme ;
- La fiabilité, robustesse et immunité aux parasites ;
- Capacité de la mémoire ;
- La documentation, le service après-vente, durée de la garantie et la formation ;

## 9- Principaux automates programmables industriels :

La programmation de ces automates se fait soit à partir de leur propre console, soit à partir du logiciel de programmation propre à la marque.

## OMRON :

- **CQM1 – CPU 11/21/41**
  - E - 192 Entrées/Sorties (à relais, à triac, à transistors ou TTL) ;
  - 32 K RAM data on Board ;
  - structure multifonction ;
  - structuration multitâche ;
  - SYSWIN 3.1, 3.2 ... 3.4 et CX\_Programmer (Littéral, Ladder) ;
  - communication sur RS 232 – C ;
  - programmation sur IBM PC/PS.

## TELEMECANIQUE :

- **TSX 17/20 :**
  - Nombre d'entrées et de sorties variable : 20 à 160 E/S.
  - microprocesseur 8031.
  - langage de programmation PL7.2.
- **TSX 67.20 :** La compacité d'un automate haut de gamme, à E/S déportables par fibre optique:
  - 1024 E/S en six bacs de huit modules;
  - extension de bacs à distance par fibre optique à 2000 m;
  - 16 coupleurs intelligents;
  - 24 K RAM data on Board;
  - 32 K RAM / EPROM cartouche utilisateur;
  - structure multifonctions;
  - structuration multitâche;
  - langage PL7.3 (Grafcet, Littéral, Ladder);
  - programmation sur IBM PC/PS.
- **FESTO :** Architecture modulaire : carte de base; carte processeur; carte de mémorisation; carte E/S.
  - FPC 202 :
  - 16 entrées 24 V DC;
  - 16 sorties 24 V DC - 1 A;
  - 8 K RAM, 8 K EPROM;
  - interface série, 20 mA boucle de courant pour imprimante;
  - console de programmation externe : console ou IBM PC;
  - programmation : grafcet, langage Festo, schéma à relais.

## SIEMENS:

- **S7 – 300.** 0 à 7 pour les entrées et 0 à 7 pour les sorties. Exemple : machine de conditionnement commandée par le SIMATIC S7-300

OBJET	ADRESSE
<b>Entrées</b>	
présence palette au poste 1 (S4)	E0,5 (entrée 5)
présence flacon au poste 1 (S5)	E0,6 (entrée 6)
<b>Sorties</b>	
commande vibreur (M2)	A4,0 (sortie 0)
commande sole tournante (M3)	A4,1 (sortie 1)

- 24 V DC;
- console de programmation externe : PG 702;
- programmation STEP7: schéma à relais , Ladder.
- PS 307

## 10. Mise en œuvre d'un automate programmable industriel :

La réalisation de la partie commande ou d'une partie de celle-ci en logique programmée nécessite la traduction du modèle concerné (GRAFCET, schémas, équations, ...) en programme exécutable par la machine. L'élaboration d'un tel programme vise donc à écrire les équations d'activation de sorties de l'API et les conditions associées. Elle constitue la phase logicielle de l'application.

**a. Affectations :** En l'absence de normes universelles, l'automaticien doit adapter les modèles aux langages des matériels retenus. Cela nécessite au préalable l'affectation des E/S et des variables auxiliaires (variables internes), c'est-à-dire l'identification avec le repérage à caractère informatique (adressage). Ce repérage est fonction du matériel et du type de programmation retenus. Les raccordements et leurs caractéristiques essentielles (tension, courant,...) sont souvent précisés dans la documentation constructrice. En général, les objets techniques raccordés en entrée sont symbolisés par un contact et ceux raccordés en sortie par un rectangle. Des liaisons peuvent être précâblées à la construction (communs des 0V) imposant les modes de branchements extérieurs.

**b- Adressage des E/S :** C'est l'association aux différentes affectations les repérages adéquats : par exemple pour le SIMATIC S7-300, 0 à 7 pour les entrées et 0 à 7 pour les sorties. Exemple : machine de conditionnement commandée par le SIMATIC S7-300 :

OBJET	ADRESSE
<b>Entrées</b>	
présence palette au poste 1 (S4)	E0,5 (entrée 5)
présence flacon au poste 1 (S5)	E0,6 (entrée 6)
<b>Sorties</b>	
commande vibreur (M2)	A4,0 (sortie 0)
commande sole tournante (M3)	A4,1 (sortie 1)

**c- Variables internes :** L'affectation consiste également à identifier ces variables destinées à mémoriser les états et valeurs intermédiaires durant l'exécution du programme.