

Chapitre I **AUTOMATES PROGRAMMABLES INDUSTRIELS (API)**

I.1 Généralités sur les Automates Programmables Industriels (API)

I.1.1 Introduction

Les Automates Programmables Industriels (API) ou encore PLC (programmable logic controller) 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 (en abrégé : API) est une ensemble électronique programmable par un personnel non informaticien et destiné commander un processus industriel (agro-alimentaire, fonderie, centre de tri, etc...). L'API (logique programmée) s'est substitué aux armoires à relais (logique câblée) 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.

I.1.2 Fonctionnement

L'automate programmable reçoit les informations relatives au système, il traite ces informations en fonction du jeu d'instruction et modifie l'état de ses sorties qui commandent les pré-actionneurs. Les fonctions que l'API doit remplir sont, Figure (I. 1) :

- ✓ **Recevoir** : nécessité d'informations d'entrées.
- ✓ **Traiter** : notion de programme et de microprocesseur.
- ✓ **Jeu d'instructions** : notion de stockage donc de mémoire.
- ✓ **Commander** : notion de sortie afin de donner des ordres.

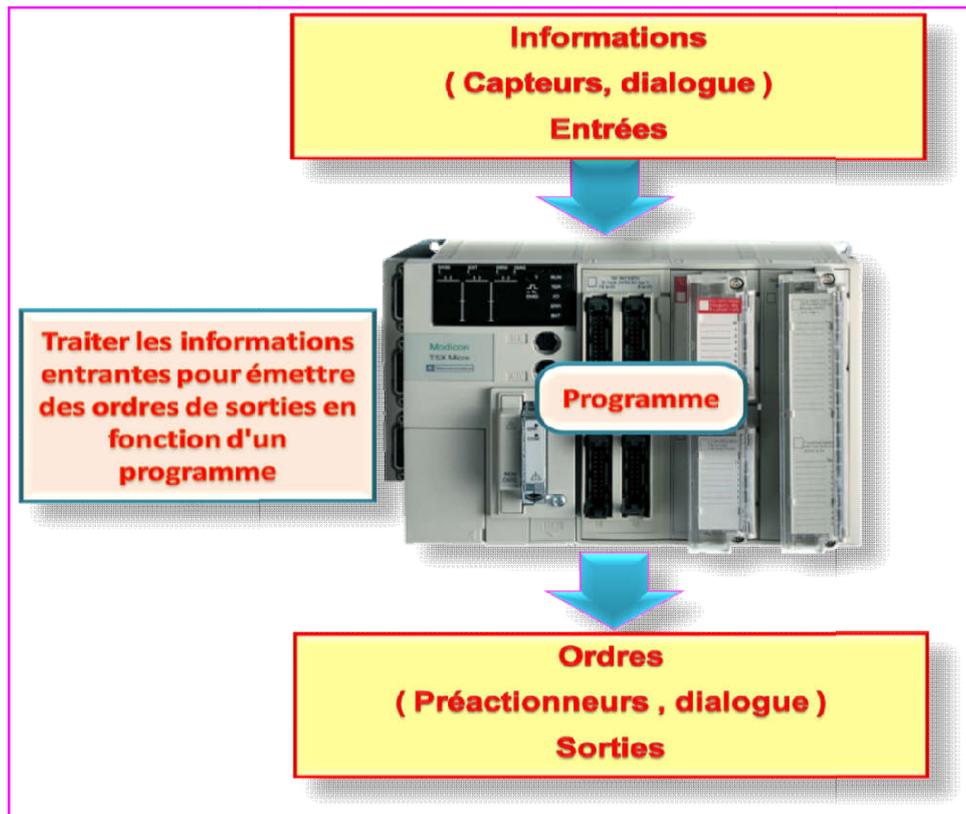


Fig I. 1 : Principe d'Automate.

I.2 Aspect extérieur des API

Les automates programmables industriels peuvent être de type compact ou modulaire.

I.2.1 API type compact

Le processeur, l'alimentation, les cartes d'entrées et sorties sont intégrés dans un seul boîtier (rack) et peuvent recevoir des extensions en nombre limité. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces API sont caractérisés par un fonctionnement simple, et leurs utilisations est réservée à la commande de petits automatismes.



a. Siemens LOGO

b. Crouzet MILLENIUM

c. Schneider ZELIO

d. Moeller PS

Fig I. 2 : API compacts.

I.2.2 API type modulaire

Ce type d'API est formé un ensemble de modules fonctionnels. Généralement, chacun des éléments : processeur, l'alimentation et les interfaces d'entrées / sorties est intégré dans des modules séparés et sont fixés sur un ou plusieurs coffret appelé racks .La communication entre ces différents modules est assurée un bus interne.

Ce type d'automate est intégré dans les automatismes complexes qui sont caractérisés par une grande : puissance, capacité de traitement et flexibilité, Figure (I. 3).



a. Siemens S7-300

b. Schneider TSX 37

c. Moeller

d. Modicon

Fig I. 3 : API modulaires.

I.3 Insertion de l'API dans un système automatisé

L'insertion de l'API dans milieu automatisé est donnée par la figure (I. 4), 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, Figure (I. 4).

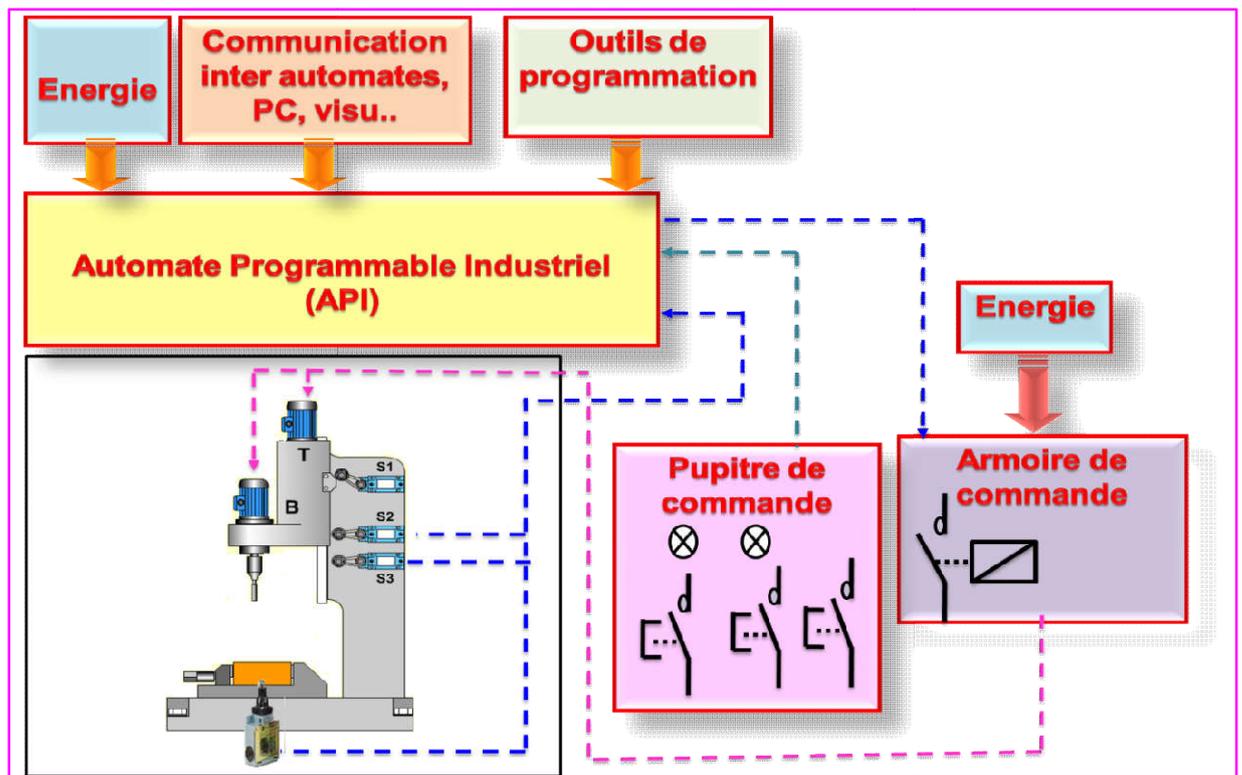


Fig I. 4 : API dans un milieu industriel.

I.4 Structure d'un A.P.I

I.4.1 Structure externe

L'aspect externe d'un API est composé, titre d'exemple, pour un API modulaire Siemens :

- | | |
|---|------------------------------|
| 1 Module d'alimentation | 6 Carte mémoire |
| 2 Pile de sauvegarde | 7 Interface multipoint (MPI) |
| 3 Connexion au 24V cc | 8 Connecteur frontal |
| 4 Commutateur de mode (à clé) | 9 Volet en face avant |
| 5 LED de signalisation d'état et de défauts | |

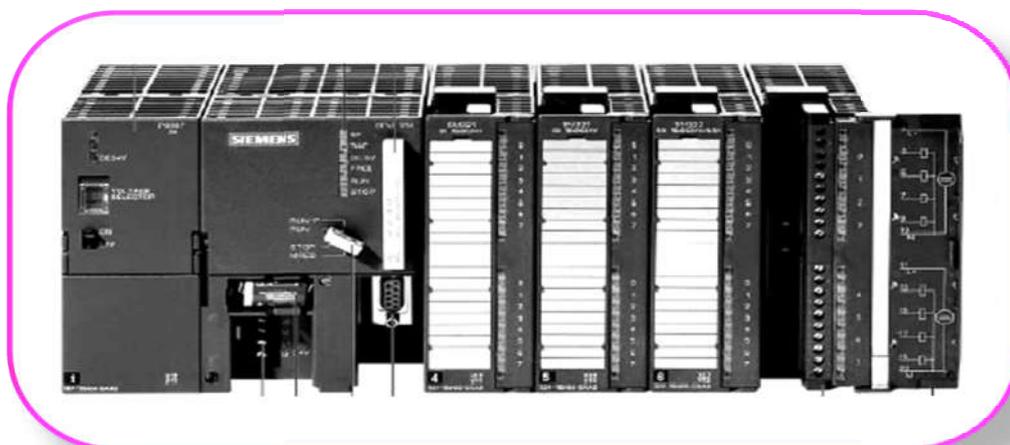


Fig I. 5 : Structure externe d'API lieu industriel.

I.4.2 Structure interne

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple, Fig (I. 6).

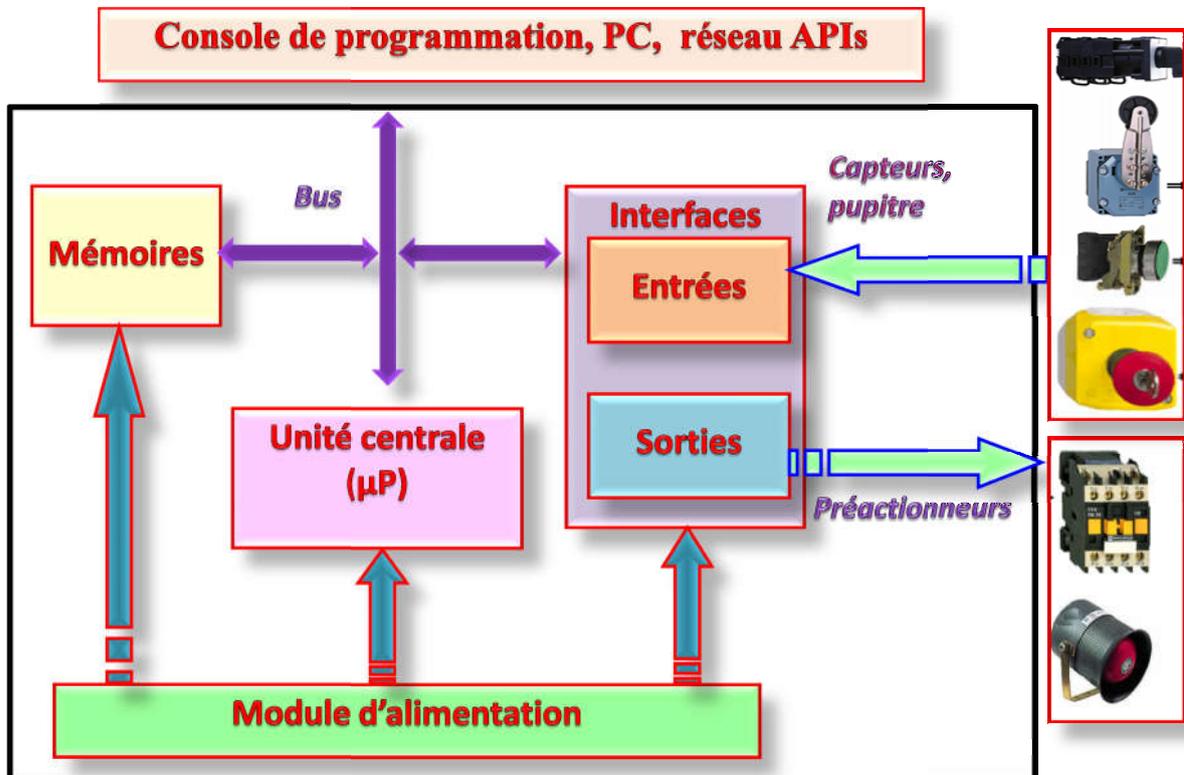


Fig I. 6 : Structure interne d'un API.

Quatre parties principales composent l'architecture interne d'API :

- ✓ Une mémoire
- ✓ Un processeur
- ✓ Des interfaces d'Entrées/Sorties
- ✓ Une alimentation

Ces quatre parties, réunies forment un ensemble compact, sont reliées entre elles par des bus (ensemble câblé autorisant le passage de l'information entre ces 4 secteurs de l'API),

I.4.3 Description des éléments d'un A.P.I

I.4.3.a. Interfaces d'Entrées/Sorties

C'est la partie se trouvant entre les deux faces partie opérative (PO) et partie commande (PC) Traduisant les ordres et les informations.

➤ Interface d'entrée:

Partie de l'API sur laquelle sont raccordés, les éléments de détection et les organes de commande installés sur le pupitre opérateur, qui donnent les informations à l'unité de traitement (capteurs, détecteurs, bouton poussoirs : départ cycle, arrêt d'urgence, réglage....)

➤ Interface de sortie:

Partie de l'API sur laquelle sont raccordés les préactionneurs ou les éléments de signalisation du pupitre qui reçoivent des ordres de l'unité de traitement.

➤ Affectation & écriture des entrées / sorties

L'affectation des entrées et des sorties permet de faire l'adressage entre le matériel et l'API en fonction de son câblage. D'un point de vue de la programmation on travaillera sur l'adressage des E/S.

Les entrées et les sorties analogiques sont notées par un mot de la façon suivante : % xy.z

x : Les sorties sont notées par la lettre **Q** (Output) pour le PL7 et **S** pour le Step7

Les entrées analogiques seront notées par la lettre **I** (Input) pour le PL7 et **A** pour le Step7

y : c'est l'emplacement physique du module analogique, numéro du rack (boitier)

z : c'est le numéro de la voie utilisée dans le même rack

▪ Exemple :

On a câblé un bouton poussoir nommé DCY sur l'entrée n°5 de la carte d'entrée du module 3 et un voyant sur la sortie n°0 de la carte de sortie du module 2.

L'écriture de l'entrée et de la sortie sont comme suit.

- ✓ Écriture de l'entrée: % I3.5
- ✓ Écriture de la sortie: % Q2.0

➤ Mémoire

La mémoire centrale est l'élément fonctionnel qui peut recevoir, conserver et restituer. Son rôle est de:

- ✓ De recevoir les informations issues des capteurs d'entrées via l'interface d'entrée ;
- ✓ De recevoir les informations générées par le processeur et destinées à la commande des sorties (valeur des compteurs, des temporisations, ...) ;
- ✓ De recevoir et conserver le programme du processus qui peut être introduit :
 - soit de la console (clavier) de programmation
 - soit à partir d'un micro-ordinateur (PC)
 - évidemment du processeur de l'API, qui lui gère et exécute le programme.

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

- ✓ La mémoire morte en lecture seulement (ROM), appelée mémoire Langage où est stocké le langage de programmation.
- ✓ La mémoire vive utilisable en lecture-écriture (RAM) appelée mémoire Travail .Elle s'efface automatiquement à l'arrêt de l'automate (nécessite une batterie de sauvegarde).

➤ Processeur

Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons parallèles appelées ' BUS ' qui véhiculent les informations sous forme binaire, son rôle est de :

- ✓ Gérer les instructions du programme ;
- ✓ Organiser les différentes relations entre la zone mémoire et les interfaces d'E/S.

➤ Module d'alimentation

Il assure la distribution d'énergie aux différents modules Composé de blocs qui permettent de fournir à l'automate l'énergie nécessaire à son fonctionnement. A partir d'une alimentation en 220 volts alternatifs, ces blocs délivrent des sources de tension dont l'automate a besoin : 24V, 12V ou 5V en continu. En règle générale, un voyant positionné sur la façade indique la mise sous tension de l'automate

I.4.4 Interfaces et cartes d'Entrées / Sorties

Les interfaces d'Entrée/Sortie (E/S), circuits électroniques, se présentent généralement sous forme d'interfaces modulaires qu'on ajoute selon le besoin, ils ont une modularité de 8, 16 ou 32 voies. Les tensions disponibles sont normalisées (24, 48, 110 ou 230V continu ou alternatif ...).

Ils doivent non seulement transmettre les ordres aux pré-actionneurs ou de recevoir les informations des capteurs, mais aussi garantir un isolement galvanique ou un découplage optoélectronique assurant ainsi la protection de l'automate contre les signaux parasites, d'où l'utilisation d'opto-coupleurs ou phototransistors à chaque entrée des cartes d'entrée et sorties.

L'interface d'entrée a pour fonction de :

- ✓ Recevoir les signaux logiques en provenance des capteurs et du pupitre opérateur ;
- ✓ Traiter ces signaux en les mettant en forme, en éliminant les parasites d'origine industrielle et en isolant électriquement l'unité de commande de la partie opérative (isolation galvanique) pour la protection.

L'interface de sortie a pour fonction de :

- ✓ Commander les pré-actionneurs et éléments de signalisation du système ;
- ✓ Adapter les niveaux de tension de l'unité de commande à celle de la partie opérative du système en garantissant une isolation galvanique entre ces dernières.

Les automates permettent de commander des sorties en T.O.R et gèrent parfois des traitements analogique ou fonctions spéciales de comptage rapide , de pesage

I.4.4.a. Cartes d'entrées TOR

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, Fig (I. 7).

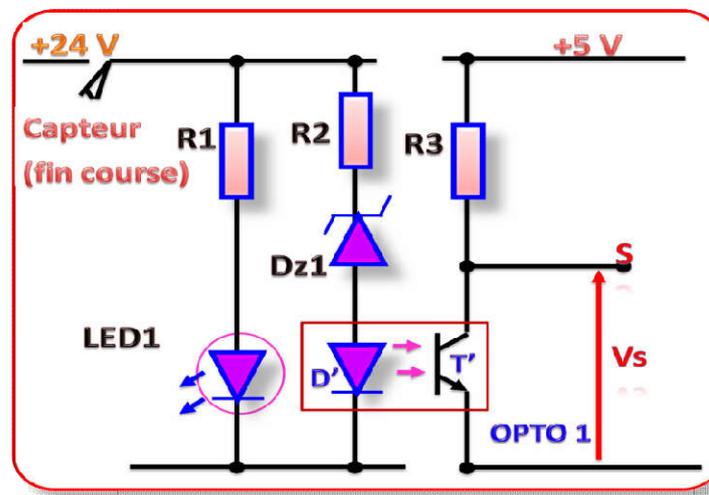


Fig I. 7 : Exemple d'une carte d'entrées TOR d'un API.

Pour le schéma ci-dessus, représentant une carte d'entrée TOR, Lors de la fermeture du capteur ;

- LED1 signal que l'entrée automate est actionnée
- La led D' de l'optocoupleur s'éclaire
- Le photo transistor T' de l'optocoupleur devient passant
- La tension $V_s=0V$

Donc lors de l'activation d'une entrée automate, l'interface d'entrée envoie un **0 logique** à l'unité de traitement et un 1 logique lors de l'ouverture du contact du capteur (entrée non actionnée).

1.4.4.b. Cartes de sorties TOR

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, Fig (I. 8).

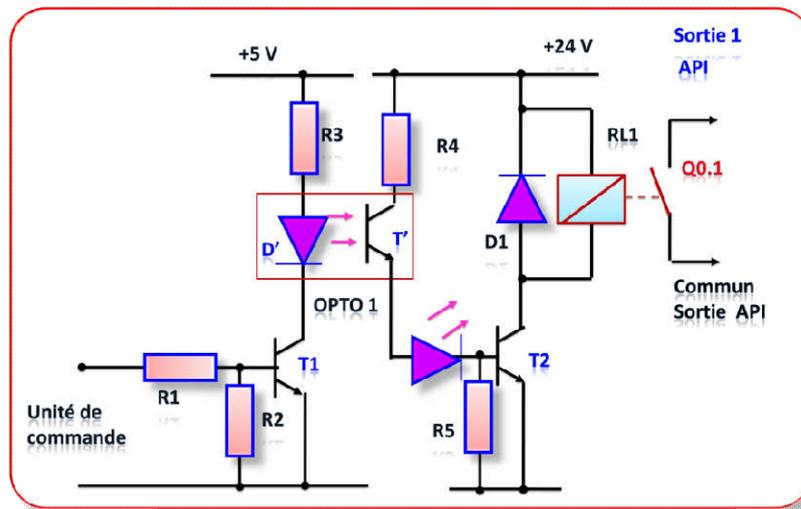


Fig I. 8 : Exemple d'une carte de sortie TOR d'un API.

Lors de la commande d'une sortie TOR d'un API ;

- ✓ L'unité de commande envoie un 1 logique (5V)
- ✓ T1 devient passant, donc D' s'éclaire
- ✓ Le photo transistor T' de l'optocoupleur devient passant
- ✓ LED 1 s'éclaire et nous informe de la commande de la sortie Q0,1
- ✓ T2 devient passant
- ✓ La bobine RL1 devient sous tension et commande la fermeture du contact de la sortie O0,1

Donc pour commander une sortie automate l'unité de commande doit envoyer :

- ✓ Un 1 logique pour actionner une sortie API,
- ✓ Un 0 logique pour stopper la commande d'une sortie API.

1.4.4.c. 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.

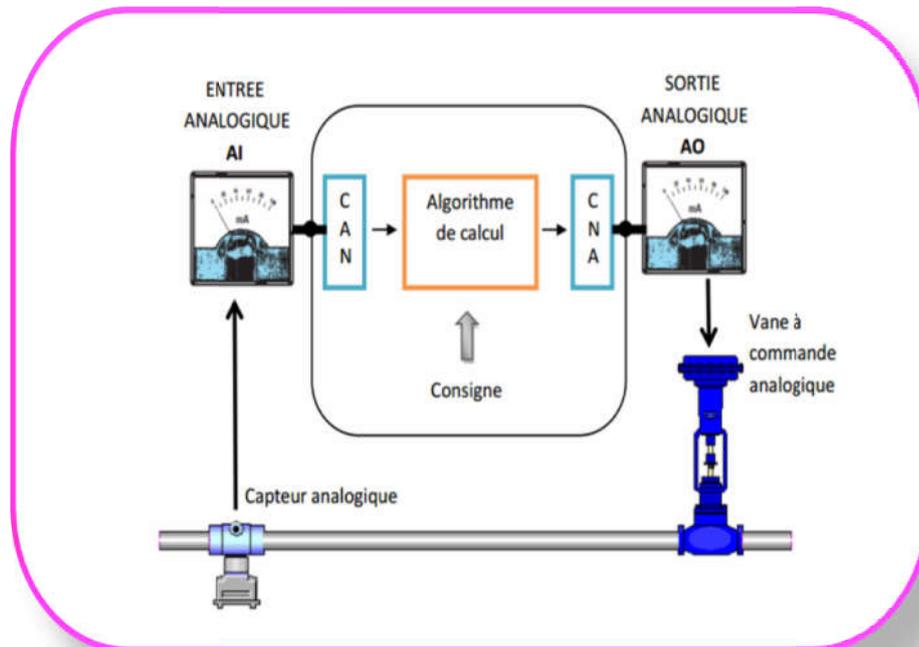


Fig I. 9 : Principe de commande des sorties état actionnée.

Exemples de grandeurs analogiques:

- Température -30 ... +120°C
- Débit 0 ... 300 l/min
- Vitesse 0 ... 3000 tr/min
- etc.

Ces variables, température, débit ou vitesse..., sont converties, en tensions électriques, en courants ou en résistances, par l'utilisation d'un transducteur de mesure. A titre d'exemple pour une vitesse mesurée de 975 tr/min, le transducteur de mesure afficherait une tension de 4,26 V. Cette tension électrique (elle peut être courant ou résistance) doit transiter par trois étapes :

1. Etape d'acquisition et de conversion analogique –numérique :

Après l'acquisition de la mesure via la carte d'entrée analogique, elle est convertie en une valeur numérique codée sur 12 à 16 Bits, grâce à un CAN (convertisseur analogique numérique).

2. Etape de traitement :

Le programme de traitement préétabli détermine sous forme numérique la valeur à fournir en sortie, en tenant compte des différents paramètres :

- ✓ La valeur de la mesure
- ✓ La valeur de la consigne
- ✓ Le type d'algorithme (PID,...)

3. Etape de Commande

Le CNA (convertisseur numérique analogique) convertit la valeur numérique de sortie, qui correspond au signal de commande après traitement, sous forme analogique (tension, courant).

I.4.1 Exemple des cartes E/S spéciales ou complexes

Elles sont connues aussi sous le nom de carte dites « intelligentes » et disposent en plus des interfaces d'entrée et de sortie, d'un petit micro-ordinateur assurant un traitement. Ceci permet, d'une part d'éviter un développement souvent épuisant de l'application, d'autre part de réduire parfois considérablement la place mémoire et le temps d'exécution au niveau de l'UC de l'automate programmable.

I.4.1.a. Cartes de commande d'axe

➤ Carte d'axes pour moteurs pas à pas

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 ou moteur pas à pas elle envoie sur sa sortie TOR une série d'impulsion avec une fréquence proportionnelle à l'accélération et à la vitesse souhaitée. La quantité totale d'impulsions envoyées correspond quand à elle au nombre de pas que doit parcourir le moteur, Figure (I. 10).

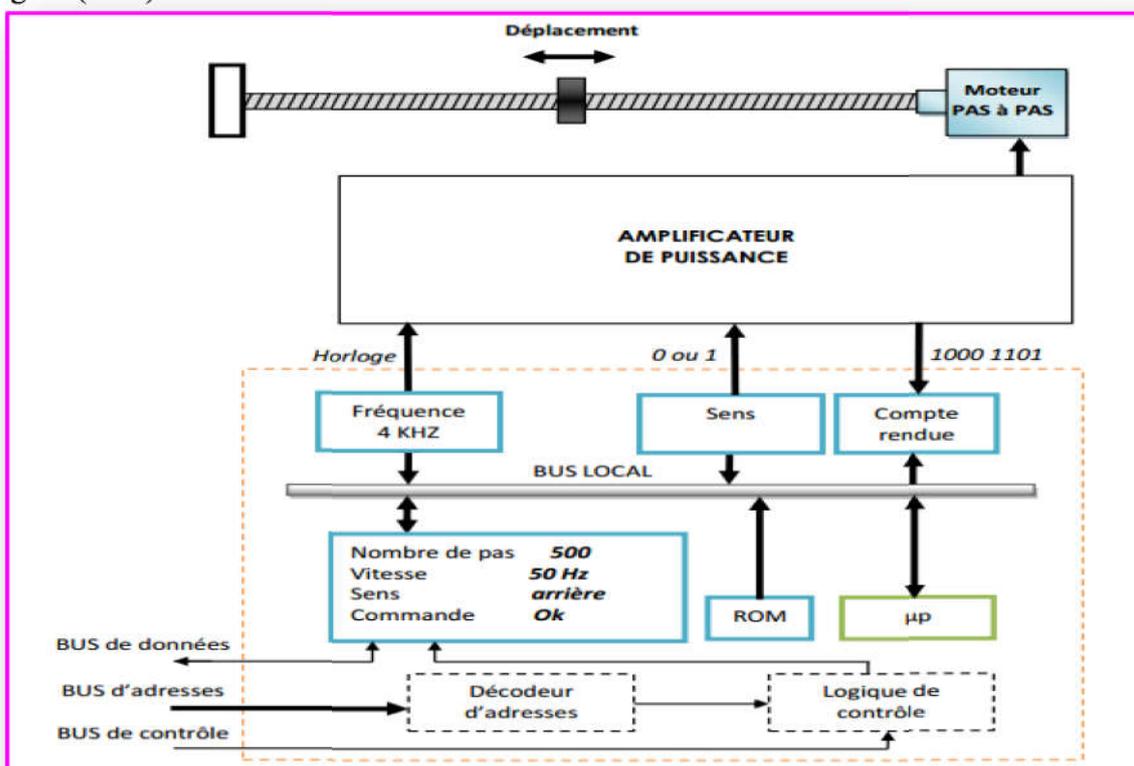


Fig I. 10 : Principe de la commande d'un moteur pas à pas avec API.

➤ Carte d'axes pour moteur à courant continu ou alternatif

La carte d'axe envoie sur sa sortie analogique $\pm 10V$ une tension proportionnelle à l'accélération et la vitesse souhaitée. Une dynamo-tachymétrique fournit une tension de retour, image de la vitesse réelle du moteur, afin que la carte puisse corriger tout écart éventuel. Un codeur de position incrémental ou absolu, permet à la carte de connaître la position de l'axe, et donc de gérer la vitesse en conséquence, Figure (I. 11).

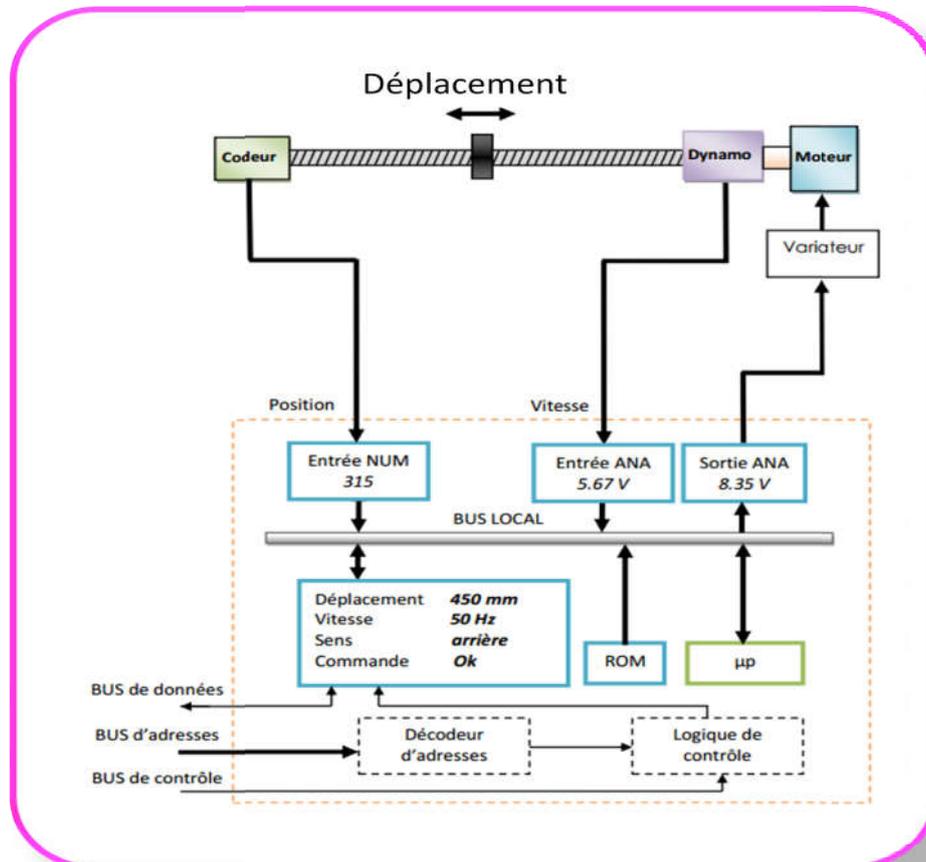


Fig I. 11 : Principe de la commande d'un axe avec API.

I.4.1.b. Cartes de régulation PID

Cette carte comporte souvent plusieurs entrées analogiques permettant de recevoir le signal de mesure issu des transmetteurs 4-20mA, ainsi que plusieurs sorties analogiques permettant de piloter les vannes de régulation.

Le microprocesseur local traite le programme élaboré à partir des différents algorithmes de régulation implantés sur la carte même (PID, Sommeation, Racine carrée,..), Figure (I. 12)

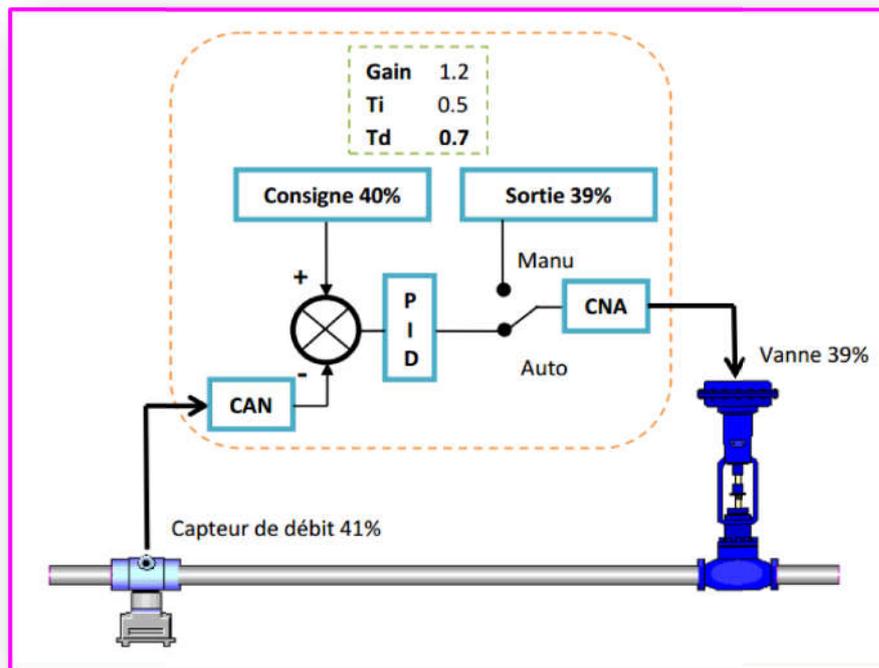


Fig I. 12 : Principe de régulation par PID par API.

I.4.1.c. 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).

I.4.1.d. Cartes de communication (RS485, Ethernet ...)

Ils permettent d'établir des communications à distance avec d'autres systèmes de traitement par lignes séries: paires téléphoniques, fibres optiques, ...

I.4.1.e. Cartes d'entrées / sorties déportées

ils permettant de décentraliser des châssis entrées / sorties industrielles sur des distances importantes (ordre du km). Cette possibilité de décentralisation permet, dans de nombreux cas, de réduire substantiellement le volume de câblage entre le processus et l'automate.

I.4.2 Traitement du programme automate

Un automate exécute son programme de manière cyclique comme suit, Figure (I. 13) :

- **Traitement interne** : L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).

- **Lecture des entrées** : L'automate reçoit des données par ses entrées et les recopies dans la mémoire image des entrées.
- **Traitement du programme** : L'automate traite les données entrantes par un programme défini dans sa mémoire et écrit les sorties dans la mémoire image des sorties.
- **Ecriture des sorties** : L'automate bascule les différentes sorties aux positions définies dans la mémoire image des sorties pour commander les actionneurs et dialoguer avec l'opérateur et les autres systèmes automatisés.

On appelle scrutation l'ensemble des quatre opérations réalisées par l'automate et le temps de scrutation est le temps mis par l'automate pour traiter la même partie de programme. Ce temps varie selon la taille du programme et la puissance de l'automate et il est de l'ordre de la dizaine de millisecondes pour les applications standards.

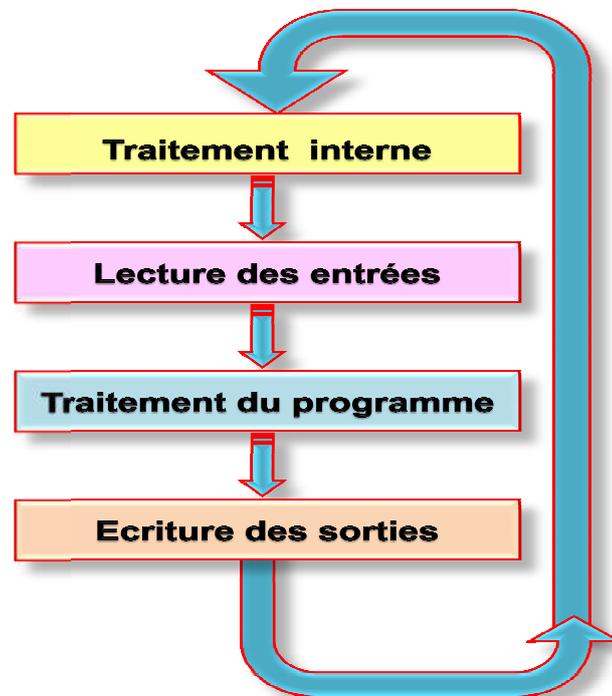


Fig I. 13 : Cycle de travail API.

I.4.3 Mise en œuvre d'un automate programmable industriel

I.4.3.a. Alimentation de l'automate

Généralement l'API est alimenté par le réseau monophasé 220V ; 50 Hz via un contacteur (contacteur KM1) et avec une protection du type magnéto-thermique.

De même, les sorties seront asservies au circuit de commande et alimentées après validation du chien de garde dont le rôle est l'enclenchement d'une procédure d'alarme en cas de dépassement du temps de scrutation à chaque cycle, Figure (I. 14).

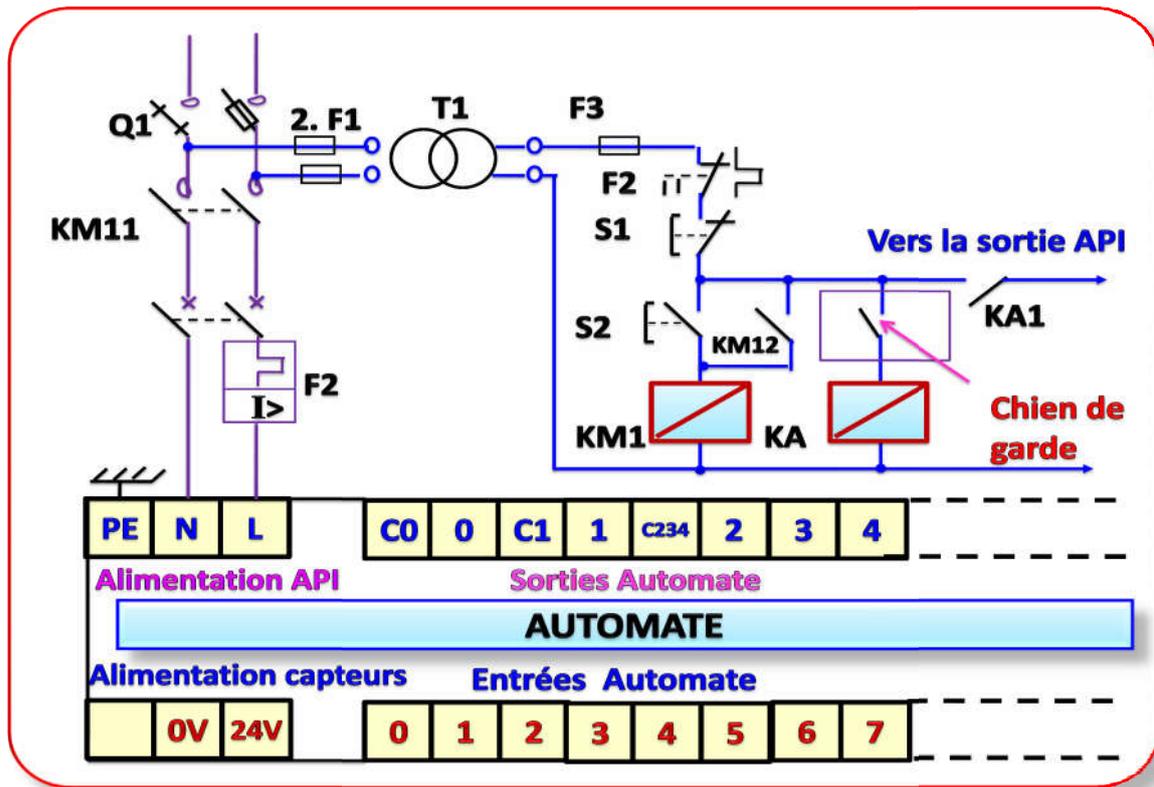


Fig I. 14 : Alimentation de l'API.

I.4.3.b. Alimentation des entrées de l'automate

L'automate est pourvu généralement d'une alimentation pour les capteurs/détecteurs. L'alimentation électrique peut être fournie par l'automate (en général 24V continu) ou par une source extérieure, Figure(I. 15).

L'alimentation des entrées et sorties d'un automate programmable peut être à :

- ✓ **Logique positive** où le commun interne des entrées est relié au 0V
- ✓ **Logique négative** où le commun interne des entrées est relié au 24V

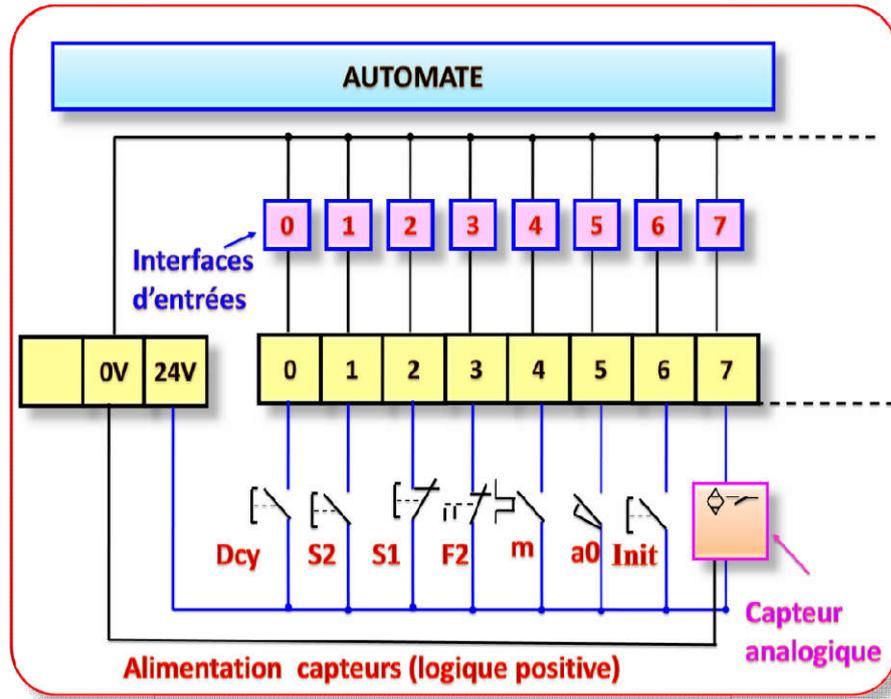


Fig I. 15 : Alimentation des entrées de l'API.

I.4.3.c. Alimentation des sorties de l'automate

Les interfaces de sorties permettent d'alimenter les divers préactionneurs, les communs des sorties sont séparés ou groupé en petit nombre afin de séparé les alimentations (alimentation des différentes valeurs des préactionneurs) ou utilisé uniquement celles dont a besoin, Figure (I. 16).

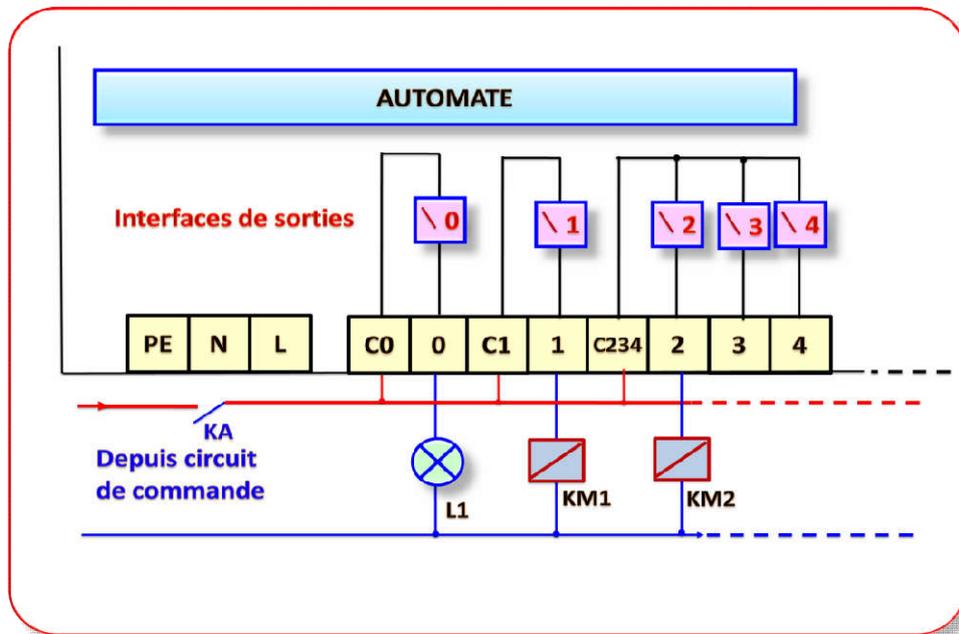


Fig I. 16 : Alimentation des sorties de l'API.

I.5 Critères de choix d'un API

Les critères techniques de choix essentiels d'un API sont :

- ✓ Le nombre et la nature des E/S ;
- ✓ Les capacités de traitement du processeur (vitesse, données, opérations, temps réel, capacité de la mémoire...).
- ✓ Fonctions ou modules spéciaux
- ✓ Les moyens de dialogue et le langage de programmation ;
- ✓ La communication avec les autres systèmes ;
- ✓ Les moyens de sauvegarde du programme ;
- ✓ La fiabilité, robustesse, immunité aux parasites ;
- ✓ La documentation, le service après vente, durée de la garantie, la formation.