

## Chapitre 2

# LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API)

## I. Introduction

### 1) Historique

Les automatismes séquentiels ont été réalisés, depuis longtemps, à base de relais électromagnétiques. L'inconvénient c'est qu'il s'agit d'une système câblée ce qui impose la refonte complète du câblage et ceci pour la moindre modification dans l'ordonnement des séquences. En 1966, l'apparition des relais statiques a permis de réaliser des divers modules supplémentaires tel que le comptage, la temporisation, le pas à pas ... Cependant cette technologie avait le même problème : technologie câblée.

En 1968 et à la demande de l'industrie automobile nord-américaine, sont apparus les premiers dispositifs de commande logique aisément modifiable : Les PLC (Programmable Logic Controller) par *Allen Bradley, Modicom et Digital Equipement*. Le premier dispositif français était le PB6 de Merlin Gerin en 1973

### 2) API - Définition

Norme NFC 63-850 : « Appareil électronique qui comporte une mémoire programmable par un utilisateur automatique (et non informaticien) à l'aide d'un langage adapté, pour le stockage interne des instructions composant les fonctions d'automatisme comme par exemple :

logique séquentiel et combinatoire ;

temporisation, comptage, décomptage, comparaison ;

calcul arithmétique ;

réglage, asservissement, régulation, etc.

pour commander, mesurer et contrôler au moyen de modules d'entrées et de sorties (logiques, numériques ou analogiques) différentes sortes de machines ou de processus, en environnement industriel. »

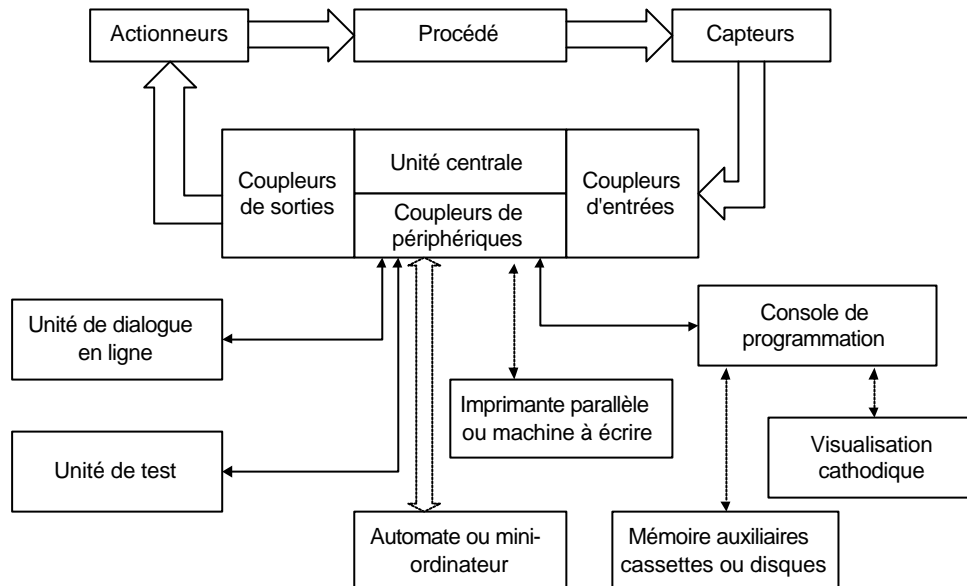
C'est donc une machine électronique qui se place entre deux grands courants : la logique câblée et le calculateur universel. Elle se distingue par plusieurs caractéristiques : conçu pour fonctionner dans des ambiances industrielles qui peuvent être sévères ; gérer un grand nombre de signaux d'E/S en temps réel ; dispose de langages adaptés aux fonctions d'automatismes et qui ne réclament pas de connaissances particulières en informatique (programmation simple); flexibles et montage rapide (structure modulaire).

Les domaines d'utilisation sont très divers : métallurgie et sidérurgie (sécurité), mécanique et automobile (montage, banc d'essais, ...), chimique, pétrolière, alimentaires, ...

## II. Structure d'un API :

### 1) Structure d'un API

Elle ressemble à celle d'un micro-ordinateur, constitué d'une unité centrale (unité de traitement), des coupleurs, des modules d'entrées (interface d'E), des modules de sortie (interfaces de S), d'une console de programmation,...



*Remarque* : En général, les automates sont conçus pour être modulaires, notamment pour pouvoir augmenter le nombre d'E/S. D'où l'utilisation d'une structure d'un rack dans lequel s'encastrent les différentes cartes (UC, alim, E/S, ...).

L'automate peut être :

soit de conception monobloc (nano-automates ou micro-automates). Les capacités d'association et de communication des matériels récents, même parmi les gammes les plus basses, offrent de nombreuses possibilités de mise en réseau de ces constituants monoblocs ; soit de conception modulaire (automates multifonctions). Cette organisation réunit un module d'alimentation et une unité centrale dans un bac pouvant accueillir divers types de modules allant des modules d'E/S TOR aux plus complexes. Elle permet ainsi des configurations évolutives et personnalisées.

#### a- L'unité centrale (UC)

C'est le coeur de la machine, comporte le(s) processeur(s) et la mémoire(s).

**Processeur** : appelé unité de traitement, il assure le contrôle de l'ensemble de la machine et effectue les traitements demandés par les instructions du programme. Il réalise les fonctions logiques, temporisation, comptage, calcul. Il comporte un certain nombre de registres (compteur ordinal, registre d'instructions, registre d'adresse, registres de données, accumulateurs, ...). Il est connecté aux autres éléments (mémoires, interfaces d'E/S, ...) par l'intermédiaire des bus.

**Mémoire** : La mémoire centrale est découpée en plusieurs zones :

zone mémoire programme ;

zone mémoire des données (états des E/S, valeurs des compteurs, temporisations, ...)

zone où sont stockés des résultats de calcul utilisés ultérieurement dans le programme ;

zone pour les variables internes.

Il existe différents types de mémoires :

mémoires vives : RAM ce sont des mémoires volatiles : elles perdent l'information en cas de coupure de l'alimentation. Certaines d'elles sont équipées de batteries de sauvegarde (autonomie réduite). Elles sont accessible en lecture et en écriture.

Mémoires mortes : les contenus sont figés. Ce sont des mémoires à lecture seule. Les informations sont conservées en permanence sans source externe.

ROM : mémoire programmé par le fabricant et ineffaçables.

PROM : vendues vierges et programmables une seule fois par l'utilisateur ;

REPROM ou EPROM : utilisables plusieurs fois (écriture / effacement). Effacement à l'UV (Ultra violet) pendant 10 à 30 minutes. Elles ne peuvent être reprogrammées qu'après un effacement total ;

EEPROM : effacement électrique et reprogrammation rapide sur place !

### **b- Bloc d'alimentation**

Permet de fournir à l'automate l'énergie nécessaire à son fonctionnement. Ils délivrent, à partir du 220 V alternatif, des sources de tension nécessaires à l'automate tels que : +5V, 12V et 24V en continu.

### **c- Coupleurs**

Ce sont des cartes électroniques qui assurent la communication entre les périphériques (modules d'E/S ou autres) et l'unité centrale.

En général, les échanges entre l'UC et les modules d'E/S s'effectuent par l'intermédiaire d'un bus interne (liaison parallèle codée) alors que ceux avec les périphériques de l'automate (console, lecteur de cassette, ...) s'effectuent par un bus externe (liaison parallèle ou série).

*Remarques :*

- Liaison série asynchrone : assurent la communication avec des périphériques plus ou moins éloignés (~1500 m). Elle transmet des mots sous format d'une succession de bits commençant par un/deux bit de start, la donnée sur 5 à 8 bits, bit de parité, ½ bits de stop.

La vitesse de la communication varie entre 75 et 19200 bauds (bit/s).

Deux types : half duplex : entre l'automate et un terminal : fonctionne tantôt en émission, tantôt en réception ou full duplex : communication entre l'automate et le calculateur : échanges simultanés en émission et réception.

- Liaisons entre API : les API peuvent être connectés entre eux par des coupleurs spécialisés. Les informations sont alors transmises de coupleur à coupleur par l'intermédiaire d'un bus.

### **d- Les cartes d'E/S**

Les E/S des automates programmables revêtent une importance évidente au plan technique. Leur coût dépasse fréquemment la moitié de l'investissement total d'une configuration. Ces facteurs justifient une étude détaillée de leur architecture générale, suivi de celle des E/S industrielles typiques.

Les caractéristiques des entrées sont :

nombre et nature (TOR, numérique, analogique, etc.) ;

spécifications électriques de raccordement (tension, courant, alimentation) ;

filtrage, c'est-à-dire. capacité à ne pas laisser passer les parasites ou signaux d'une durée inférieure à une valeur définie. Les valeurs standard vont jusqu'à quelques dizaines de millisecondes.

Les caractéristiques des sorties sont :

nombre et nature ;

technologie : à contact mécanique (relais) ou statique (composant électronique), et les temps de commutation associés (de la milliseconde pour les contacts à quelques dizaines de microsecondes pour les transistors) ;

spécifications électriques de raccordement (tension, courant, puissance, etc.).

Les entrées et les sorties sont proposées :

soit par quantité imposées (10, 16, 24 E/S pour les nano-automates par exemple). Ces configurations de base peuvent être étendues en connectant des blocs d'extensions ou d'autres appareils ;

soit sous forme de cartes ou modules à 4, 8, 16 ou 32 entrées ou sorties embrochables dans des racks.

#### **e- Les consoles :**

- Console d'exploitation : permet le paramétrage et les relevés d'informations (modification des valeurs et visualisation) sur site ;
- Console de programmation, réglage et exploitation.

Cette dernière effectuée dans la phase de programmation l'écriture, la modification, l'effacement et le transfert d'un programme dans la mémoire de l'automate ou dans une mémoire REPRM.

Dans la phase de réglage et d'exploitation, elle permet d'exécuter le programme pas à pas, de le visualiser, de forcer ou de modifier des données telles que les entrées, les sorties, les bits internes, les registres de temporisation, les compteurs, ..., la sortie sur imprimante du programme en cas de présence d'un port de sortie.

La console, généralement équipée d'un écran à cristaux liquides, peut afficher le résultat de l'auto-test comprenant l'état des modules d'E/S, l'état de la mémoire, de la batterie, ...

*Remarque :* certaines console ne peuvent être utilisées que sur l'automate (alimentation fournie par ce dernier), d'autres sont autonomes grâce à leur mémoire interne et à leur alimentation.

#### **f- les boîtiers de test :**

C'est un dispositif rudimentaire et donc moins onéreux que la console. Il permet aux personnels d'entretien d'accéder à la mémoire de l'API pour détecter d'éventuelles erreurs et de visualiser le programme ou les valeurs des paramètres. Exemple :

- affichage de la ligne (ou de son numéro) de programme à contrôler ;
- visualisation de l'instruction (code opératoire et adresse de l'opérande) ;
- La visualisation du contenu de l'accumulateur logique, après exécution de l'instruction ; ce qui permet en pas à pas de localiser de l'erreur ;
- visualisation de l'état des entrées, sortie, mémoire interne ;

#### **g- Les unités de dialogue en ligne :**

C'est un outil qui complète les boîtiers de test. Il a pour rôle de permettre certaines interventions sur l'API qui auraient nécessité autrement la connexion d'une console ou

l'utilisation des dispositifs ponctuels tel que la roue codeuse... Mais la connexion de la console posait un problème de coût et surtout de sécurité puisqu'elle accède à l'ensemble des fonctions de l'automate.

L'UDEL est ainsi considéré comme outil privilégié de l'exploitant auquel il permet :

- modification des constantes, compteurs, temporisations ;
- forçage des entrées / sorties ;
- exécution de parties de programme ;
- chargement de programmes en mémoire à partir de cassettes, ...

Ces boîtiers se présentent sous forme enfichable dans l'unité centrale ou séparés de celle-ci. Ils comportent des touches de fonctions, des touches numériques, une visualisation, un dispositif de sécurité, ... Certains ont un aspect très semblable à celui d'une console de programmation, d'autres ont une physionomie plus robuste.

### III. Les cartes entrées / sorties :

#### 1) Architecture :

parallèle (faible et moyenne distance), série asynchrone (grandes distances), coupleur avec microprocesseur pour traitement des signaux et décharger processeur principal, modules intelligents et réseaux locaux, fibre optique (vitesse et immunité aux parasites électromagnétiques), réseaux locaux ...

Les interfaces entre le procédé et la logique interne d'un automate sont assurés par des cartes électroniques appelées coupleurs. Ces coupleurs accèdent d'une part au bus, d'autre part au bornier. Celui-ci se trouve généralement sur la face avant de l'automate, il doit être à la fois protégé et facilement accessible.

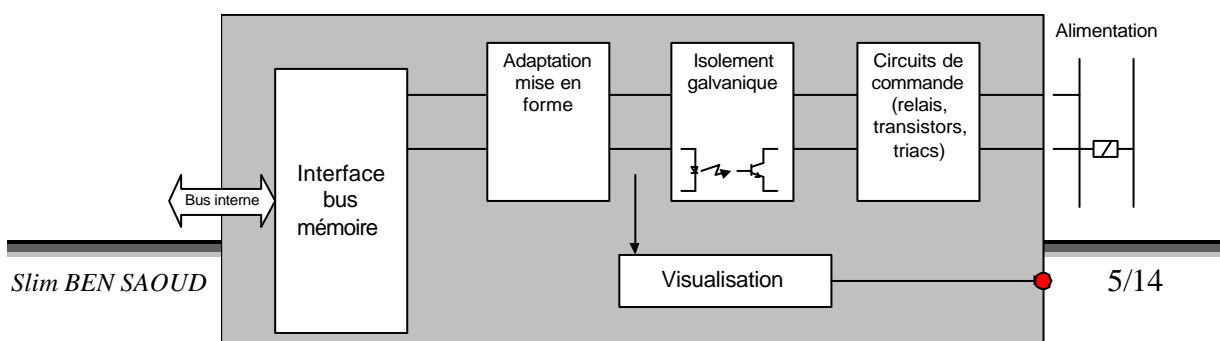
#### a- Les entrées / sorties TOR

Les API offrent une grande variété d'E/S TOR adaptées au milieu industriel et qui peuvent accepter suivant les cartes, des informations en courant ou en tension, alternatifs ou continus.

- Les entrées TOR : Elles permettent de raccorder à l'automate les différents capteurs logiques tels que : boutons poussoirs, thermostats, fins de course, capteur de proximité, photo-électriques, roues codeuses, ... Outre l'acquisition de l'information, les modules d'E binaires réalisent un prétraitement du signal : mise en forme, élimination des parasites (filtrage), découplage des niveaux de puissance.
- les sorties TOR : Elles permettent de raccorder à l'automate les différents pré-actionneurs tels que : vannes, contacteurs, voyants, électrovannes, relais de puissance, afficheurs,...Le même souci d'isolement électrique se retrouve au niveau des sorties. De plus, il convient de rendre disponible sur celles-ci une certaine puissance utilisable à la commande du procédé.

Tensions de sortie : 5, 24, 48, 125 VCC ou 24, 48, 120, 220 VCA, les courants variant de quelques mA à quelques A.

La sortie s'effectue sur relais ou sur triacs conducteurs au zéro de tension en alternatif, ou sur transistor de puissance en continu.



Le découplage se fait par transformateur d'isolement en alternatif ou par opto-électronique en continu.

La puissance de sortie est souvent suffisante pour commander directement des vannes ou des petits moteurs. Toutefois, l'automate ne supprime pas tout le relayage classique en raison des puissances importantes mises en jeu fréquemment dans la commande des procédés industriels.

**Remarques :** Pour assurer la sécurité du procédé, il est parfois indispensable que certaines sorties soient protégées contre les incidents pouvant survenir sur l'API tels que les microcoupures et les coupures de l'alimentation. Les sorties à accrochage (opposé à volatile) remplissent ce rôle : différentes solutions technologiques : relais à automaintien type magnétique ou mécanique, tores ferrites (Crouzet), relais bistables au mercure (Macq Electronique).

La modularité, la visualisation de chaque sortie au niveau du bornier, la possibilité de (dé)brancher les actionneurs de la machine sous tension sont comparables à ce qui a été vu pour les entrées.

### **b- Les entrées sorties numériques :**

Utilisés pour les API haute ou moyenne gamme effectuant des traitement numériques. La longueur définit par la taille du mot mémoire de l'API (ex : 16 bits).

Les cartes d'E/S numériques se présente donc comme 16 E ou S binaires rassemblées, pour lesquels on utilise les mêmes précautions d'isolement. Souvent, on utilise une seule carte d'E/S numérique et on multiplexe les diverses entrées numériques souhaitées à l'aide de sorties binaires. Cette idée sera utilisée de la même façon pour le multiplexage d'E binaires ou le démultiplexage des sorties. Elle permet artificiellement de multiplier le nombre d'E/S physiques sans augmenter le nombre d'E/S machine.

**Remarque :** Sur de nombreux automates les E/S binaires ou numériques sont banalisées : pour certains, les E/S numériques sont obtenues par regroupement du nombre de bits nécessaires parmi les E/S binaires. Pour d'autres, les E/S binaires sont en réalité des éléments des E/S numériques.

L'interprétation est effectuée par le moniteur selon les directives de l'utilisateur, exemple : U21 repère l'entrée binaire n°21 alors que UN21 identifie l'entrée numérique n°21.

### **c- Les E/S analogiques :**

Les E/S analogiques transforment une grandeur analogique en une valeur numérique et vice versa. La précision dépend du nombre de bits utilisés.

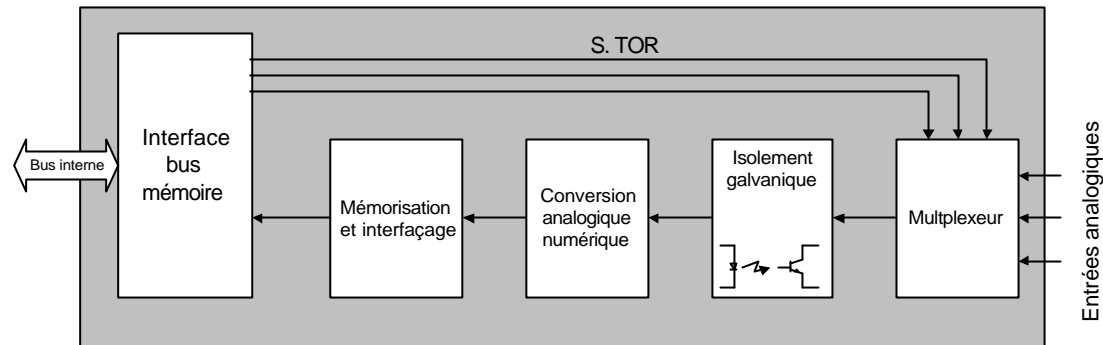
Technologiquement, les EA/SA sont caractérisées par l'amplitude du signal analogique (typiquement 0/10V ou -10/+10V) et par le courant correspondant.

- **Les EA :** En générale on opère sur 12 bits (ou 11 bits + 1 bit de signe si nécessaire) = 4096 niveaux quantifiés, ainsi pour un signal admissible de 12V, le niveau correspond à 2,44mV. Si l'on souhaite bénéficier d'une précision de 0,1 %, il ne sera pas possible de mesurer des signaux d'amplitude inférieure à 2,44V. C'est pourquoi, les EA performants comprennent un amplificateur à gain programmable destiné à permettre une mesure à pleine échelle, avec une excellente précision.

Le convertisseur analogique numérique opère par approximations successives. Il réclame autour de 2 $\mu$ s pour effectuer une résolution de 12 bits. Ce délai introduit un retard pur dans l'acquisition et nécessite le maintien à un niveau constant du

signal à convertir pendant la conversion. D'où l'utilisation d'un bloc échantillonneur bloqueur.

Ce dernier scrutant le signal de la voie sélectionnée pendant un court instant à une cadence déterminée (période d'échantillonnage) et conservant cette valeur analogique instantanée jusqu'à la fin de la conversion.



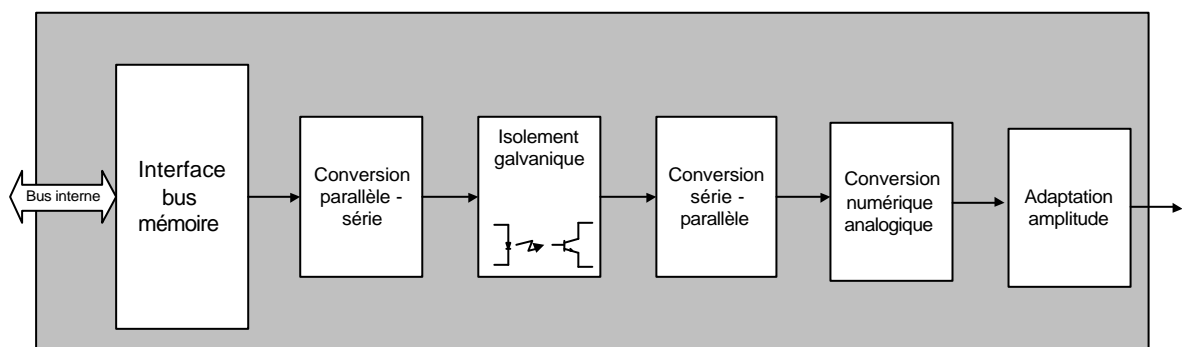
L'échantillonneur est le dispositif qui prélève à une fréquence  $f_e$ , fréquence d'échantillonnage, les valeurs prises par le signal analogique. Le théorème de Shannon indique que  $f_e$  doit être au moins égale au double de la fréquence du signal échantillonné pour que la mesure soit significative.

La fréquence d'échantillonnage, directement liée à la vitesse du convertisseur, doit être plus élevée que la fréquence maximale du signal à convertir, de manière à fournir une représentation numérique fidèle de la forme analogique du signal.

*Remarque :* Le coût du dispositif de conversion impose le multiplexage des voies analogiques : cette contrainte économique et technologique induit un retard de l'ordre de  $1\mu s$  (temps de multiplexage pour multiplexeur CMOS !!!). Des solutions technologiques existent pour accroître les performances des EA : Utiliser des EA intelligentes autonomes et effectuant des prétraitements localement.

- Les SA : Ces modules assurent la conversion numérique analogique. L'intensité ou la tension de sortie est proportionnelle à la valeur numérique.

Les sorties analogiques peuvent posséder un convertisseur par voie. Le nombre des voies sur ces cartes est 2 ou 4.



#### d- Les E/S spécialisées :

- Compteurs et temporisateurs : permettent de gagner du temps (comptage électronique) ou encore d'économiser de l'espace mémoire (temporisation).
- Entrées logiques à seuil ajustable : Détectent le franchissement d'un seuil par une grandeur physique continue.

- Modules intelligents : conçus souvent à base de microprocesseur, ils assurent de façon autonome certaines fonctions d'automatismes : modules de positionnement / commande d'axes, de régulation numérique, de communication, ...

#### IV. Cycle de l'API:

Une caractéristique originale et unique des API est le fonctionnement cyclique de l'UC. Le programme est composé d'une suite d'instructions placés dans la mémoire (RAM, EPROM, EEPROM,...). Il peut être par exemple, constitué des phases :

- Acquisition des entrées ;
- Traitement ;
- Activation des sorties.

Ce cycle se reproduit ainsi indéfiniment : à chaque cycle, tout le programme est exécuté. La durée d'un cycle est de l'ordre de 20 ms.

Suivant la conception de l'API ou de la programmation, on distingue 3 autres possibilités :

- Toutes les entrées sont acquises au début du cycle et les sorties sont commandées en fin de cycle, c'est-à-dire après que toutes les équations aient été résolues.
- Toutes les entrées sont acquises au début du cycle et les sorties sont activées après chaque résolution d'équation intégrée dans le programme. On parle d'un cycle synchrone vis à vis des entrées et asynchrone vis à vis des sorties ;
- Les équations sont traitées une par une en prenant uniquement la valeur des entrées concernées. Ce cycle peut présenter un inconvénient, en effet, une entrée peut être appelée plusieurs fois avec des valeurs différentes pour plusieurs équations au cours du même cycle. Risque de provocation d'aléas.
- Les entrées sont scrutées toutes les n ms, quelle que soit la durée du cycle ; les sorties sont activées à la demande.

*Remarques :*

- L'organisation du cycle devient plus complexe lorsque l'on se trouve en présence d'instructions de sauts particulières.
- Si le cycle n'est pas effectué dans une période définie par l'automate, exemple d'un programme ou l'on engendre une boucle infinie involontaire, une alarme est déclenchée par l'intermédiaire du « chien de garde » qui est une temporisation exécutée au début de chaque cycle.
- Le temps effectué par l'UC pour exécuter un programme de 1K mots ( $2^{10} = 1024$  mots), est appelé période d'un automate programmable. Ce temps varie d'un modèle à un autre et ne se différencie qu'à la milliseconde.
- A la notion du temps de cycle est liée la notion du temps de réponse c'est-à-dire. l'intervalle de temps séparant la variation d'une entrée et la variation de la sortie conséquente. La pire situation se rencontre dans le cycle synchrone vis à vis des entrées et des sorties :  $T_R < 2.T_C$ .

#### V. Le TSX 17-20:

##### 1) Structure matérielle :

Cet automate monobloc se place dans un armoire et se fixe sur un rail DIN normalisé.

Il comporte :

- Une alimentation 110 à 220 VCA fournissant une alimentation 24VCC, 250 mA destinée aux capteurs 24VCC raccordés au TSX17 ;
- Une UC avec prise RS485 pour raccordement au terminal de programmation, au bus multipoint UNI-TELWAY ou à un périphérique de dialogue ;



- Une RAM interne de 24Ko programme sauvegardable par une pile lithium (1 an) ;
- Un emplacement pour cartouche mémoire EEPROM ou EPROM (sauvegarde programme utilisateur) ;
- Un emplacement pour cartouche micro-logicielle PL7-2 ;
- 20, 34 ou 40 E/S TOR avec raccordement sur borniers à vis déconnectables (entrées 24VCC ou 110VCA isolées, sorties relais ou transistors 0,35 A protégées) ;
- une alimentation capteurs 24VCC (avec automates de base 110/220VCA – 50/60 Hz) ;
- Un ensemble de visualisation de l'état automate (RUN, CPU, I/O, BATT) et des entrées sorties ;
- Deux entrées événementielles 24 VCC ;
- Un compteur rapide 2 kHz, 5/24VCA.
- Un connecteur pour une autre extension au bus d'E/S.

*Remarque :* Lors d'une coupure secteur, les informations en mémoire automate (programme, configuration, données, horodateur) sont sauvegardées 1 heure minimum. Trois dispositifs permettent d'augmenter la durée de cette sauvegarde : cartouche EEPROM (sur automate), EPROM (sur automate ou sur adaptateur TSX17ACC7), pile lithium de durée 1 an (sur automate).

## 2) Position, adressage et raccordement des E/S:

Les entrées sorties sont désignées de la façon suivante : I/O x,i, avec :

- x le numéro du module (0 à 2 pour 17-20/PL7-1, 0 à 3 pour 17-20/PL7-2) : x vaut 0 pour l'automate de base, 1 pour la 1<sup>ère</sup> extension connectée, ...
- i le numéro de la voie d'E/S : OO à 23 pour les E et 00 à 15 pour les sorties.

## VI. Mise en œuvre et programmation d'un API

Les automates doivent fonctionner sans danger pour les utilisateurs et sans risque d'interrompre la production et ceci malgré les contraintes très sévères qu'ils subissent en milieu industriel. Les défaillances peuvent provenir de la conception, de l'installation, de l'exploitation, d'un constituant ou d'un composant, de l'environnement, de la maintenance, ... Il faut donc veiller à la sécurité en analysant les risques et les normes en vigueur.

Les automates sont soumis à différents types d'environnement :

- Environnement physique et mécanique : vibrations, chocs, humidité, température. Pour les températures élevées, il faut prévoir une ventilation. Une humidité relative de plus de 85% provoque des condensations et accélère la corrosion alors qu'une humidité inférieure à 30% provoque des potentiels électrostatiques. Les vibrations et les chocs provoquent sur l'appareil des dégâts non négligeables en agissant, soit sur des contacts, ou sur des soudures provoquant une rupture du circuit.
- Environnement chimique : gaz corrosifs (Cl<sub>2</sub>, H<sub>2</sub>S, SO<sub>2</sub>), vapeurs d'hydrocarbures, poussières métalliques (fonderies, aciéries, ...), poussières minérales (cimenteries, ...) : Ils entraînent une corrosion qui endommage les contacts et provoque des courts-circuits. Pour éviter ces problèmes les industriels utilisent des filtres pour éliminer les poussières ou gaz et recouvrent d'un enduit les circuits imprimés. Certains proposent des automates totalement étanches pour les environnements très pollués.
- Environnement électrique : les éléments perturbateurs sont :

- les forces électromotrices (fem), thermoélectriques (effet Peltier, quelques mV) ;
- les parasites d'origine électrostatiques ;
- les interférences électromagnétiques (transformateurs, postes de soudure, ...)
- Certains émetteurs récepteurs à des fréquences correspondant à l'API peuvent détériorer le processeur de celui-ci.

### 1) Choix de l'API :

Après l'établissement du cahier des charges, il revient à l'utilisateur de regarder sur le marché l'automate le mieux adapté aux besoins, en considérant un certain nombre de critères importants :

- Le nombre et la nature des E/S ;
- 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, immunité aux parasites ;
- La documentation, le service après vente, durée de la garantie, la formation

### 2) Mise en oeuvre :

La réalisation de tout ou partie d'une partie commande en logique programmée nécessite la traduction du modèle concerné (GRAFSET, 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 constructeurs. 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 TSX17-20, 0 à 11 pour les entrées et 0 à 7 pour les sorties. Exemple : machine de conditionnement commandée par le TSX17-20 :

OBJET	ADRESSE
Entrées	
présence palette au poste 1 (S4)	I0,5 (entrée 5)
présence flacon au poste 1 (S5)	I0,6 (entrée 6)
Sorties	
commande vibreur (M2)	O0,0 (sortie 0)
commande sole tournante (M3)	O0,1 (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.

## 3) Programmation :

Suivant l'automate et le type de console, on distingue deux catégories de langages : les langages littéraux et les langages graphiques.

### a- les langages littéraux :

- *Le langage booléen* : Ils sont caractérisés par la notion de ligne dans laquelle les constituants sont les opérateurs booléens (et les variables associées), le symbole d'affectation étant le signe d'égalité. ex :  $(A.B+C).D=Y$

La valeur calculée du premier membre est affectée au second membre, contrairement à l'usage en informatique et en mathématique.

- *Le langage mnémorique : par liste d'instructions* : Le langage mnémorique s'apparente aux « langages informatiques » offrant des possibilités de traitement des plus élémentaires (traitements sur bits) aux plus complexes (traitements numériques, sur texte, graphique, ...). Leurs utilisation varie peu d'un constructeur à l'autre. Néanmoins quelques nuances existent dans l'écriture du programme. Exemple : la complémentation d'une variable logique s'écrit en fonction du matériel : N nom de la variable ou / nom de la variable, ...

Les lignes de programmation sont constituées de caractères alphanumériques écrit selon des codes et des règles définissant le langage.

*Remarque* : Si la programmation ne permet qu'un traitement logique, le langage est dit booléen.

Le programme est une suite de ligne. Chaque ligne, introduite par une adresse, comporte une instruction qui peut être suivie de commentaire. Chaque instruction se compose d'un code opération suivi d'un opérande :

Adresse Adresse mémoire	OP code Code opératoire	OPERANDE Données
Exemple 0	LD	700
1	OUT	500
2	LD	500
3	OUT	501

### b- Les langages graphiques :

Ils permettent une transcription aussi directe que possible des modèles afin de faciliter les tâches de programmation et de réduire les sources d'incertitudes. Les caractéristiques principales d'un tel langage sont montrées à travers la présentation du logiciel PL7-2 de Télémécanique :

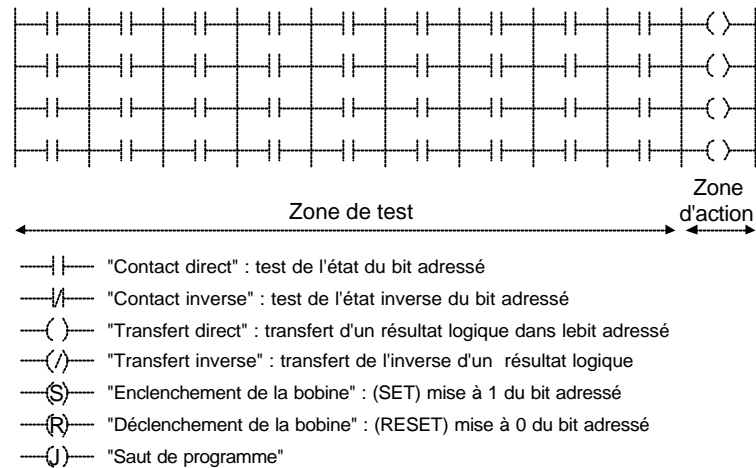
- **Le langage PL7-2**

Il combine deux langages de programmation complémentaires à savoir :

- le langage à contacts adapté au traitement logique (ladder diagram) ;
- le langage adapté au traitement séquentiel et inspiré du GRAFCET.

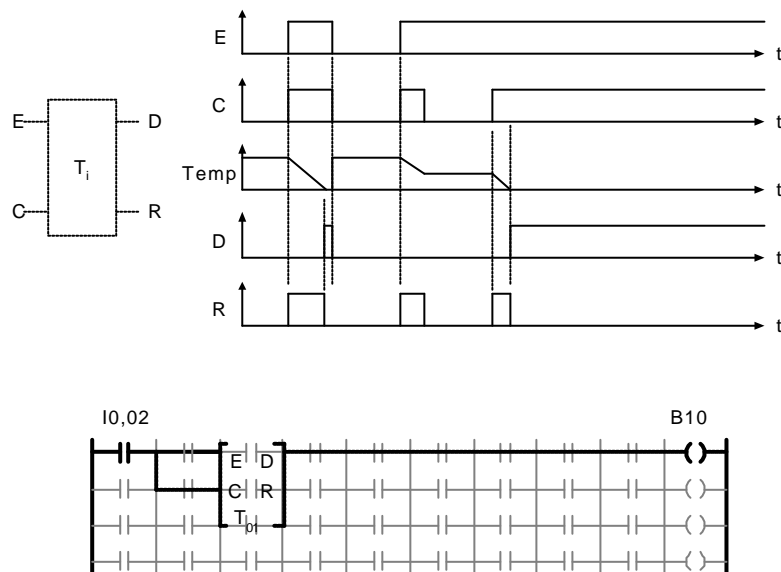
- Programmation en langage à contacts :

Constitué de plusieurs lignes horizontales contenant des symboles graphiques de tests répartis en zone de test (contacts, blocs fonction) et d'action répartis en zone d'action (bobines). Ces symboles sont assemblés selon le principe des schémas électriques, à savoir mise en série et mise en parallèle des contacts. Pour le langage considéré, le constructeur propose 999 réseaux de 4 lignes permettant chacun la saisie de 9 contacts et une bobine.



Chaque réseau est repéré par une étiquette (LABEL) portant un numéro de 1 à 999.

Ce langage autorise également l'intégration, au sein du réseau à contacts, de blocs préprogrammés (blocs fonctionnels) permettant la réalisation de



fonctions d'automatismes (temporisation, comptage, communication, ...) par l'intermédiaire de blocs de texte.

Exemple : Bloc temporisateur (décompte) à deux entrées d'activation à 1 (E : armement et C : lancement) et deux sorties (D : fin de temporisation et R : temporisation en cours). D et R peuvent être testés par programme comme n'importe quel autre bit. Il est également possible de tester la valeur courante de la temporisation V ainsi que la valeur de présélection P. Le réglage de la durée de temporisation est obtenu par une combinaison de la base de temps choisie

(10 ms, 100 ms, 1s, 1 mn) et de la valeur de la présélection P : valeur comprise entre 0 et 9999.

- Programmation en langage inspiré GRAFCET :

La programmation d'un GRAFCET fait apparaître trois zones scrutées consécutivement :

- une zone de traitement préliminaire (PRL) correspondant à une logique d'entrée décrite en logique à contacts (prépositionnement de variables, élaboration de réceptivités complexes, ...);
- une zone de traitement séquentiel (SEQ) utilisant le langage « GRAFCET » et comprenant des étapes, des transitions auxquelles sont associées des réceptivités, programmées en langage à contacts et des liaisons orientées;
- une zone de traitement postérieur (POST) correspondant à la logique de sortie et décrite également en langage à contacts (prise en compte des modes de marches, des sécurités programmées, des expressions des sorties, ...).

L'écriture du programme séquentiel s'effectue à l'aide de symboles graphiques voisins des symboles de l'outil GRAFCET avec des terminaux doués de capacités graphiques pour la saisie et l'affichage.

LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API) .....	1
I. Introduction .....	1
1) Historique .....	1
2) API - Définition.....	1
II. Structure d'un API :.....	2
1) Structure d'un API.....	2
a- L'unité centrale (UC) .....	2
b- Bloc d'alimentation.....	3
c- Coupleurs.....	3
d- Les cartes d'E/S .....	3
e- Les consoles : .....	4
f- les boîtiers de test : .....	4
g- Les unités de dialogue en ligne : .....	4
III. Les cartes entrées / sorties :.....	5
1) Architecture :.....	5
a- Les entrées / sorties TOR .....	5
b- Les entrées sorties numériques :.....	6
c- Les E/S analogiques : .....	6
d- Les E/S spécialisées :.....	7
IV. Cycle de l'API :.....	8
V. Le TSX 17-20:.....	8
1) Structure matérielle :.....	8
2) Position, adressage et raccordement des E/S:.....	9
VI. Mise en œuvre et programmation d'un API.....	9
1) Choix de l'API :.....	10
2) Mise en oeuvre : .....	10
a- Affectations :.....	10
b- Adressage des E/S : .....	10
c- Variables internes :.....	11

3) Programmation : .....	11
a- les langages littéraux : .....	11
b- Les langages graphiques : .....	11