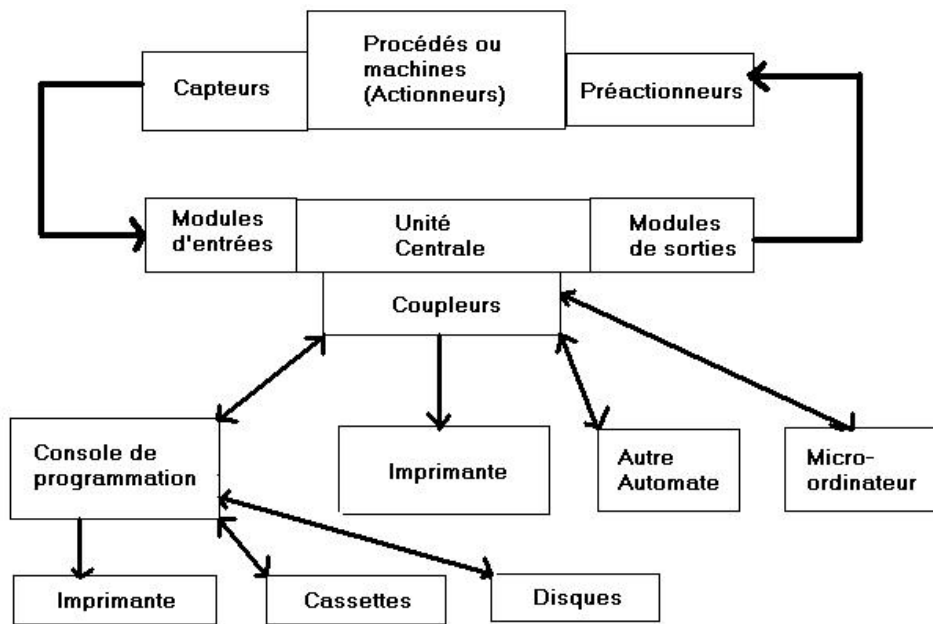

API

Sommaire

- 1- Introduction
- 2- Architecture d'un API
 - 2.1. Structure interne
 - 2.2. Structure externe
 - 2.3. Fonctionnement
 - 2.4. Mise en situation
- 3- Raccordement automate
- 4- Jeu d'instructions
 - 4.1. Opérations logiques de base
 - 4.2. Instructions complémentaires
 - 4.3. Langages d'automates
 - 4.4. Logiciels de programmation
- 5- Réseaux d'automates
- 6- Principaux automates programmables industriels
- 7- L'essai d'un automatisme simple commandé par un automate
- 8- Les applications de l'automate



L'automate programmable et ses auxiliaires

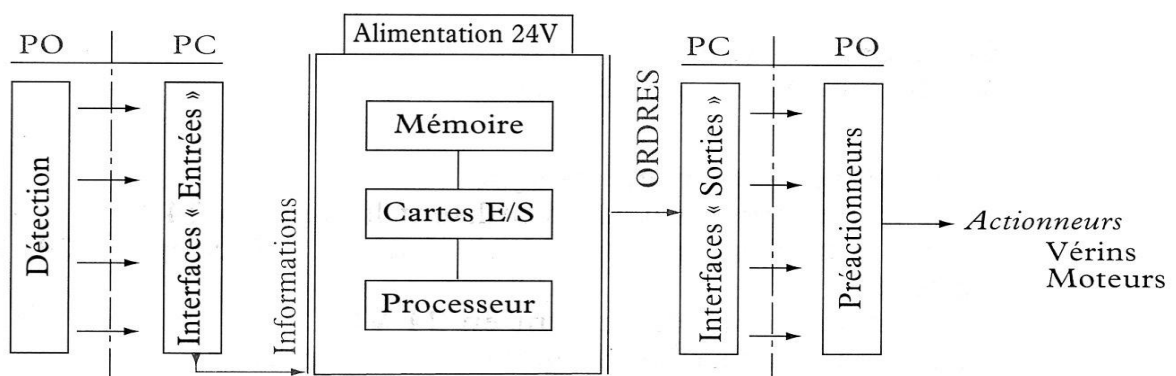
1. Introduction

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

Un **API** se distingue d'un ordinateur par le fait qu'il s'agit d'un système électronique programmable spécialement adapté pour les non-informaticiens. Il est en général destiné à être mis entre les mains d'un personnel dont la formation a été surtout orientée vers l'électromécanique. L'**API** s'est substitué aux armoires à relais en raison de sa souplesse (mise en œuvre, évolution ...), mais aussi parce que dans les automatisations de commande complexe, les coûts de câblage et de mise au point devenaient trop élevés.

2. Architecture d'un API

2.1. Structure interne



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.

Un API est constitué essentiellement de 5 modules :

- 1- L'unité centrale
- 2- Le module d'entrées
- 3- Le module de sorties
- 4- Le module d'alimentation
- 5- Le module de communication

2.1.1. L'unité centrale (CPU)

Toute "l'intelligence" de l'automate réside dans son unité centrale ; c'est le "cerveau" qui gère l'automate .Elle représente le cœur de la machine et comprend le/les processeur(s) et la mémoire.

2.1.1.1. Le microprocesseur

Fonction

Le **microprocesseur** réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul... à partir d'un programme contenu dans sa **mémoire**.

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

Composition

- une Unité Logique (UL) qui traite les opérations logiques ET, OU et Négation.
- une Unité Arithmétique et Logique (UAL) qui traite les opérations de temporisation, de comptage et de calcul.
- un Accumulateur qui est un registre de travail dans lequel se range une donnée ou un résultat.
- un Registre d'Instruction qui contient, durant le temps de traitement, l'instruction à exécuter.
- un Décodeur d'Instruction qui décode l'instruction à exécuter en y associant les microprogrammes de traitement.

- un Compteur Programme ou Compteur Ordinal qui contient l'adresse de la prochaine instruction à exécuter et gère ainsi la chronologie de l'exécution des instructions du programme.

2.1.1.2. La zone mémoire

a)- La Zone mémoire va permettre :

Fonction

- Recevoir les informations issues des capteurs d'entrées
- Recevoir les informations générées par le processeur et destinées à la commande des sorties (valeur des compteurs, des temporisations, ...) (**EPROM**)
- Recevoir et conserver le programme du processus (**RAM et EEPROM**)

Action possible sur une mémoire

- **ECRIRE** pour modifier le contenu d'un programme
- **EFFACER** pour faire disparaître les informations qui ne sont plus nécessaires
- **LIRE** pour en lire le contenu d'un programme sans le modifier

Technologie des mémoires

- **RAM** (Random Acces Memory): mémoire vive dans laquelle on peut lire, écrire et effacer (contient le programme)
- **ROM** (Read Only Memory): mémoire morte accessible uniquement en lecture.
- **EPROM** "Erasable Programmable Read Only Memory", en Français "Mémoire à lecture seule programmable et effaçable" mémoires mortes reprogrammables effacement aux rayons ultra-violets.
- **EEPROM** signifie "Electrically Erasable Programmable Read Only Memory", en Français "Mémoire à lecture seule effaçable électriquement". mémoires mortes reprogrammables effacement électrique.

Remarque :

La capacité mémoire se donne en mots de 8 BITS (Binary Digits) ou octets.

Exemple:

Soit une mémoire de 8 Koctets = $8 \times 1024 \times 8 = 65\,536$ BITS. Cette mémoire peut contenir 65 536 informations binaires.

Cette CPU est programmable directement par console ou par le biais d'une liaison série et d'un logiciel adapté. Elle peut être en RUN (elle exécute le programme) ou en STOP (toutes les sorties sont au repos, contacts ouverts). Son rôle consiste à recevoir, mémoriser et traiter

les informations reçues aux entrées et détermine l'état des sorties en fonction du programme établi.

2.1.2. Les modules d'entrées/sorties

Les **entrées** reçoivent des informations en provenance des **éléments de détection (capteurs)** et du **pupitre opérateur (BP)**.

Les **sorties** transmettent des informations aux **pré-actionneurs (relais, électrovannes ...)** et aux **éléments de signalisation (voyants)** du pupitre.

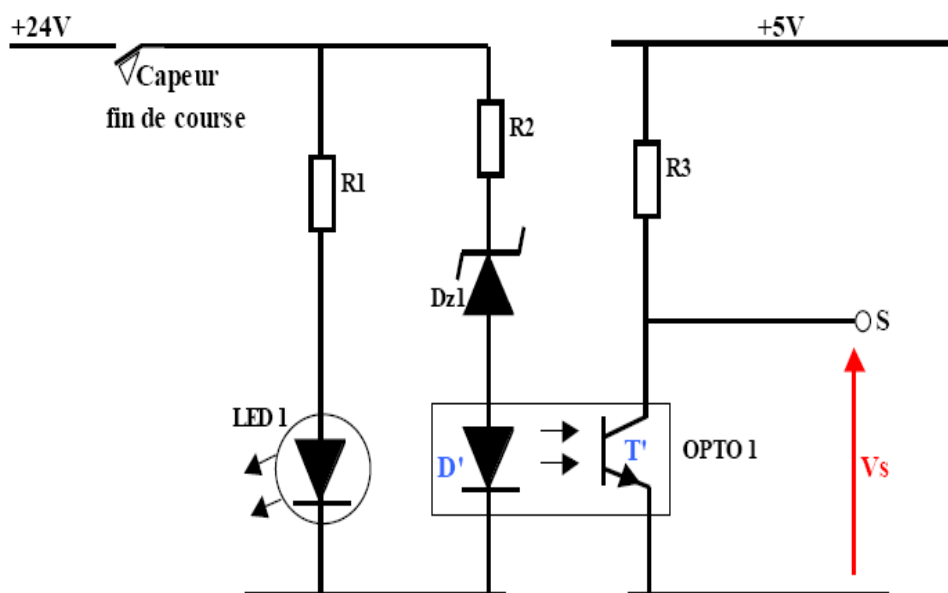
2.1.2.1. Les interfaces d'entrées

a) Interfaces d'entrées TOR

Elles sont destinées à :

- Recevoir l'information en provenance des capteurs logiques tels que les boutons poussoirs, les pressostats, thermostats, fins de course, capteurs de proximité inductifs ou capacitifs, capteurs photo-électriques, roues codeuses etc.

- Assurer l'adaptation, l'isolement, le filtrage et la mise en forme des signaux électriques. Une diode électroluminescente située sur la carte donne l'état de chaque entrée. Le nombre d'entrées sur une carte est de : 4, 8, 16, 32.



Fonctionnement de l'interface d'entrée TOR:

Lors de la fermeture du capteur ;

- LED1 signale 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 $V_s = V_{ce} = 0V$ (transistor saturé)

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).

b) Transmetteurs analogiques

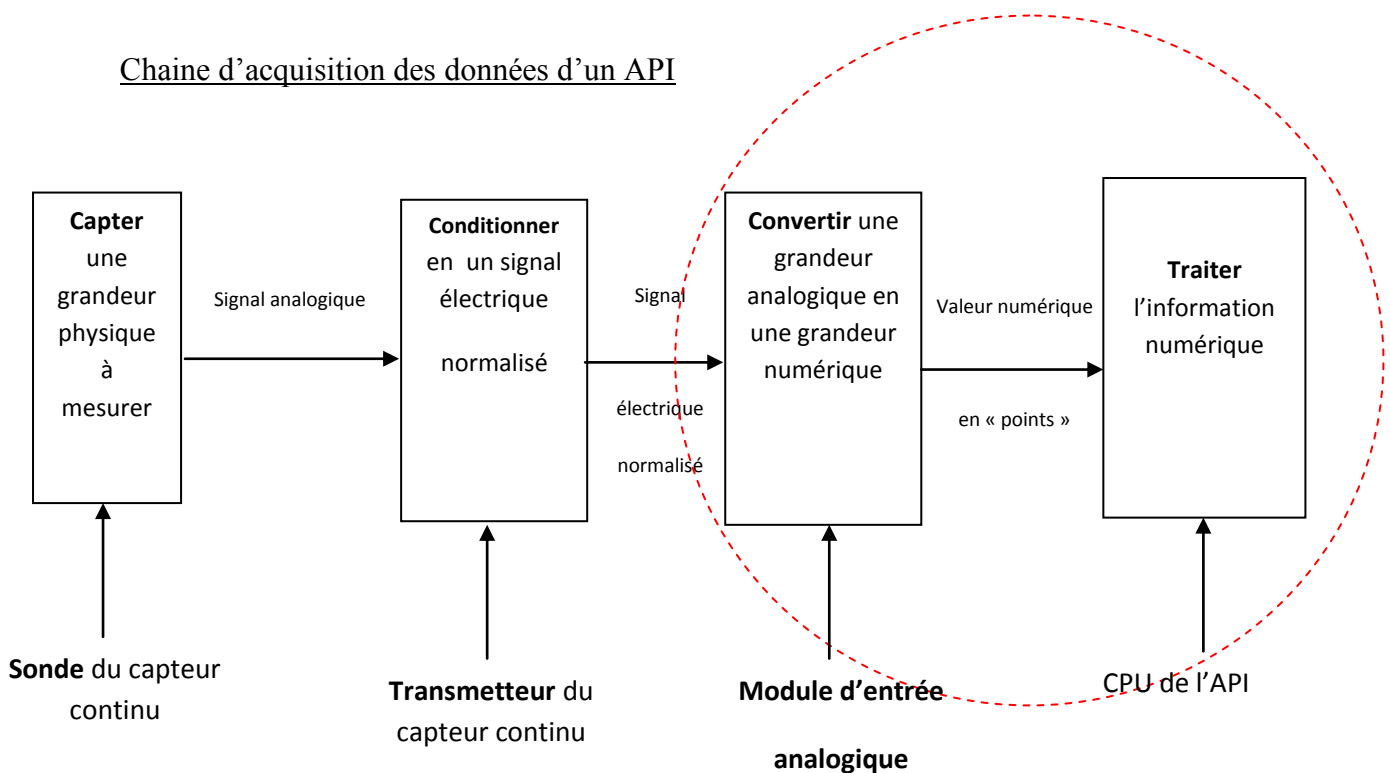
Les transmetteurs analogiques : Tension / intensité permettent d'adapter les signaux issus des capteurs pour les rendre compatibles avec l'unité de traitement. La variation de la grandeur d'entrée est convertie en une variation :

- En tension : de 0V, à 10V,
- En intensité : de 0 mA à 20 mA, ou de 4 mA à 20 mA.

Exemple : Transmission de mesure de température effectuée par une sonde PT

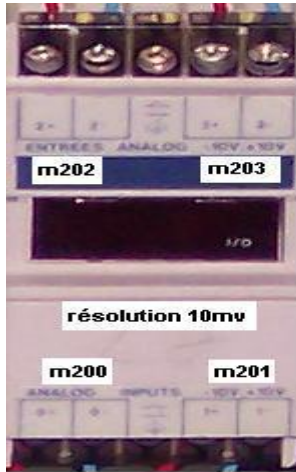
Sur le marché, il existe des modules à 2, 4, 8 voies d'entrées. Une carte standard assure jusqu'à 8 connections.

Chaîne d'acquisition des données d'un API



Exemple : Module d'entrée CAN du TSX 17

ce module (appelé aussi d'entrées analogiques) est muni de 4 entrées (0-15v) traduisant le résultat dans 4 mots respectifs (m200 à m203) la résolution est de **10mv**



c) Les cartes d'entrées numériques

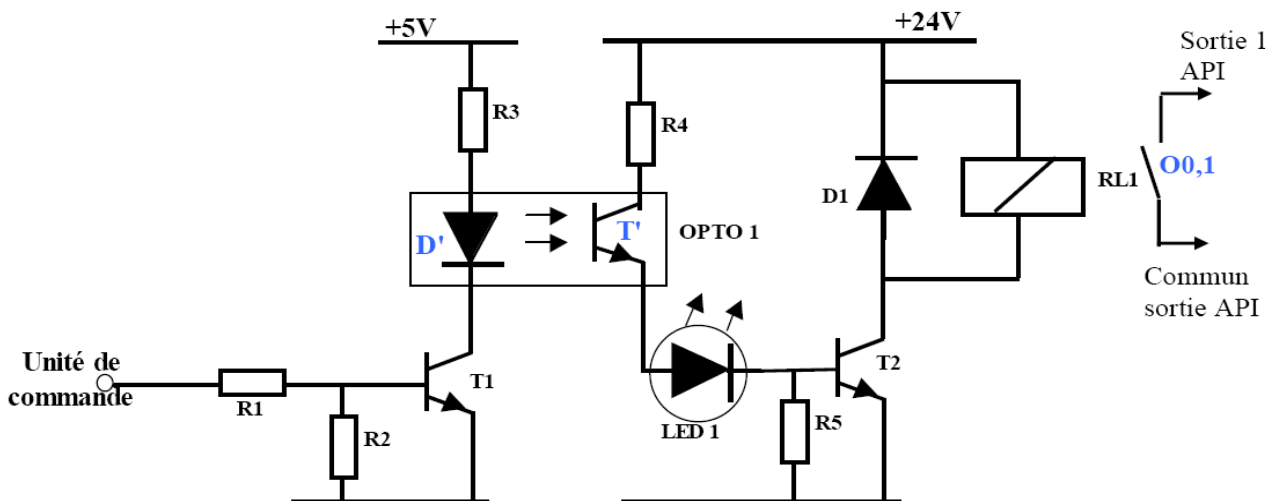
Elles permettent de dialoguer directement avec les roues codeuses. Certaines sont dotées d'un nombre fixe de voies. Chaque voie possède un nombre d'entrées qu'on organise selon ses besoins pour la constitution des mots en respectant les règles prescrites par le constructeur. Généralement ce sont des mots de 16 bits codés en BCD. (binary-coded decimal, exemple : le nombre 127, il suffit de coder chacun des chiffres 1, 2 et 7 ce qui donne 0001, 0010, 0111.)

2.1.2.2. Les interfaces de sorties

a)- Interfaces de sorties TOR

Elles sont destinées à :

- Commander les pré-actionneurs et éléments des signalisations du système
- 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



Fonctionnement de l'interface de sortie :

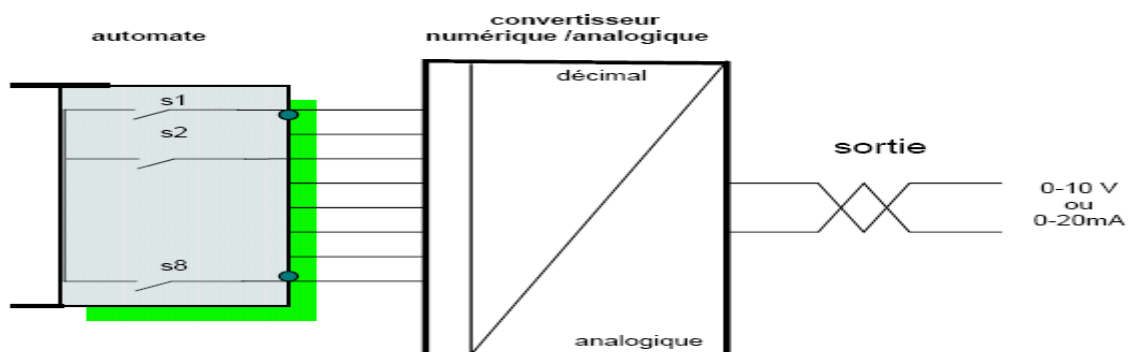
Lors de la commande d'une sortie automate ;

- 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 O0,1
 - 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

b) Interfaces de sorties analogiques

Les conventions digitales /analogiques ont pour fonction de générer un signal analogique normalisé (0-10 V ; 0-20 mA) à partir d'une information numérique, délivrée par l'unité de traitement et codée en binaire, sur des sorties digitales TOR raccordées aux entrées de l'interface(ou convertisseur).



Exemple : module de sortie analogique du TSX 17

ce module (appelé aussi "de **sorties analogiques**")est muni de 2 voies (0-10v)convertissant le résultat de 2 mots respectifs (**m204 à m205**)en une tension de 0 à 10v .

la résolution est de 10mv



c) Les cartes de sorties numériques

Elles permettent de dialoguer directement avec les afficheurs numériques. Comme pour le cas des entrées numériques, les sorties de ce type sont dotées d'un nombre fixe de voies. Chaque voie possède un nombre d'entrées qu'on organise selon ses besoins pour la constitution des mots, en respectant les règles prescrites par le constructeur. Généralement ce sont des mots de 16 bits codés en BCD.

2.1.3. Le module d'alimentation

Composé de blocs qui permettent de fournir à l'automate l'énergie nécessaire à son fonctionnement. Cette énergie sera dimensionnée en fonction des consommations des différentes parties.

En règle générale, un voyant positionné sur la façade indique la mise sous tension de l'automate.

On distingue trois sources d'alimentation : alimentation alternative, continue ou auxiliaire.

- L'alimentation alternative : délivre, à partir du secteur 110V ou 220V, les principales tensions dont l'automate a besoin : 24V, 12V ou 5V en continu. Ces alimentations comportent deux parties : le redressement et la régulation. En cas de défaillance de l'automate, certaines alimentations sont dotées d'un contact à relais et d'un voyant.
- L'alimentation continue : certains automates fonctionnent à partir de tensions continues fournies par des alimentations externes. Elle nécessite l'utilisation de transformateurs à enroulements séparés.

- Les alimentations auxiliaires : Lorsque l'alimentation principale ne peut pas assurer le fonctionnement de toutes les cartes, il faut prévoir des alimentations auxiliaires .Celles –ci sont implantées soit à l'intérieur du coffret, soit dans des racks d'extension, soit dans les colonnes d'interface, ou encore à l'extérieur de l'automate.

2.1.4. Le module de communication

Il Comprend les consoles, les boîtiers de tests et les unités de dialogue en ligne.

2.1.4.1 Les consoles



Il exis
(modi



me permet le paramétrage et les relevés d'informations (modification), l'autre permet en plus la programmation, le réglage et l'exploitation. Cette dernière dans la phase de programmation effectue :

- L'écriture
- La modification
- L'effacement
- Le transfert d'un programme dans la mémoire de l'automate ou dans une mémoire EEPROM.

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, etc.
- La sortie sur une imprimante du programme si un port de sortie existe.

La console peut également afficher le résultat de l'autotest comprenant l'état des modules d'entrées et de sorties, l'état de la mémoire, de la batterie, etc. Les consoles sont équipées (pour la plupart), d'un écran à cristaux liquides.

Certaines consoles ne peuvent être utilisées que connectées à un automate (l'automate fournit l'alimentation à la console), d'autres peuvent fonctionner de manière autonome grâce à la mémoire interne et à leur alimentation.

2.1.4.2 Les boîtiers de tests



Destinées aux personnels d'entretien, ils permettent de visualiser le programme ou les valeurs des paramètres. Par exemple :

- Affichage de la ligne de programme à contrôler
- Visualisation de l'instruction (code opératoire et adresse de l'opérande)
- Visualisation de l'état des entrées
- Visualisation de l'état des sorties.

2.1.4.3. Les unités de dialogue en ligne

Elles sont destinées aux personnels spécialistes du procédé et non de l'automate programmable, et leur permet d'agir sur certains paramètres :

- Modification des constantes, compteurs temporisations
- Forçage des entrées/sorties
- Exécution de parties de programme
- Chargement de programmes en mémoire à partir de cassettes.

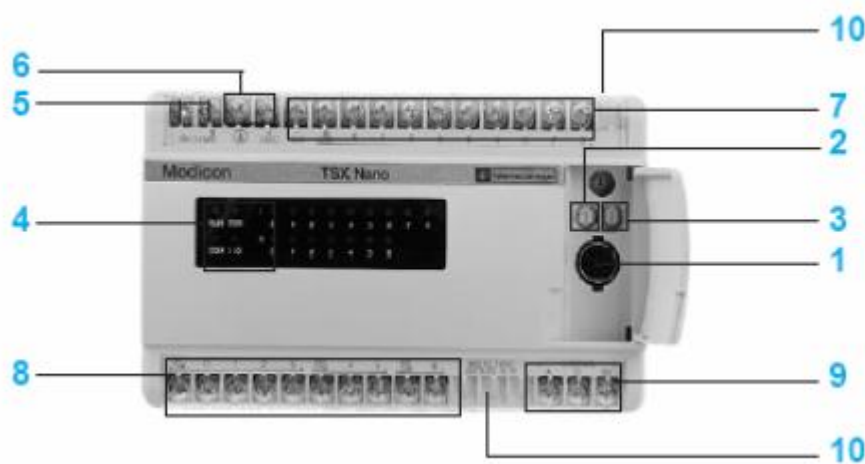
Ces boîtiers se présentent sous la forme enfichable dans l'unité centrale ou séparées de celle-ci. Il comporte des touches de fonctions, numériques, une visualisation, un dispositif de sécurité, l'ensemble est piloté par micro-processeur.

2.2. Structure externe

2.2.1. Automate monobloc

Le type **monobloc** possède généralement un nombre d'entrées et de sorties restreint et son jeu d'instructions ne peut être augmenté. Bien qu'il soit parfois possible d'ajouter des extensions

d'entrées/sorties, le type monobloc a pour fonction de résoudre des automatismes simples faisant appel à une logique séquentielle et utilisant des informations tout-ou-rien.



Automate monobloc TSX Nano

Les automates monobloc comprennent en général :

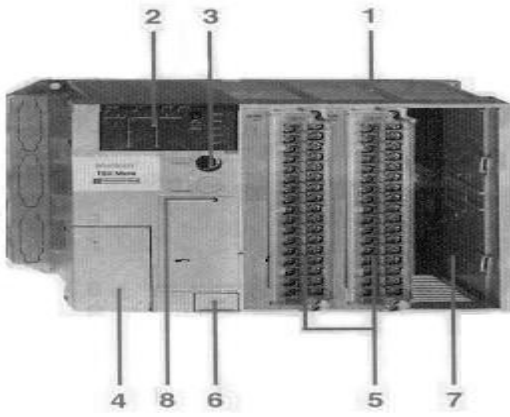
- 1 - Une prise pour raccordement du terminal permettant la programmation à partir d'une console ou d'un ordinateur.
- 2 & 3 - Des prises pour des fonctions de communication spécialisées avec d'autre équipement.
- 4- Ecran sur lequel on peut visualiser :
 - l'état des entrées et sorties relais "Tout ou Rien"
 - l'état automate
- 5 - un raccordement de l'alimentation secteur.
- 6 - une alimentation capteurs (en général 24 VDC/150mA)
- 7 - un raccordement des capteurs d'entrées
- 8 - un raccordement des pré actionneurs de sorties
- 9 - un raccordement pour des entrées spécifiques
- 10 - un cache amovible pour protection des borniers à vis

2.2.2. Automate modulaire

Par ailleurs, le type modulaire est adaptable à toutes situations. Selon le besoin, des modules d'entrées/sorties analogiques sont disponibles en plus de modules spécialisés tels: PID, BASIC et Langage C, etc. La modularité des API permet un dépannage rapide et une plus grande flexibilité.

L'automate TSX 37-08 comprend :

- 1- Un bac à 3 emplacements.



- 2- Un bloc de visualisation centralisé.
- 3- Une prise terminale repérée TER.
- 4- Une trappe d'accès aux bornes d'alimentation.
- 5- Deux modules à 16 entrées et 12 sorties « Tout ou Rien » positionnés dans le premier et le deuxième emplacement (positions 1, 2, 3 et 4).
- 6- Une trappe d'accès à la pile optionnelle.
- 7- Un emplacement disponible.
- 8- Un bouton de réinitialisation

Automate TSX 37-08

2.4. Fonctionnement

La plus part des automates programmables industriels ont un fonctionnement cyclique .Le processeur est géré en fonction d'un programme qui est une suite d'instructions placées en mémoire. 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.

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).

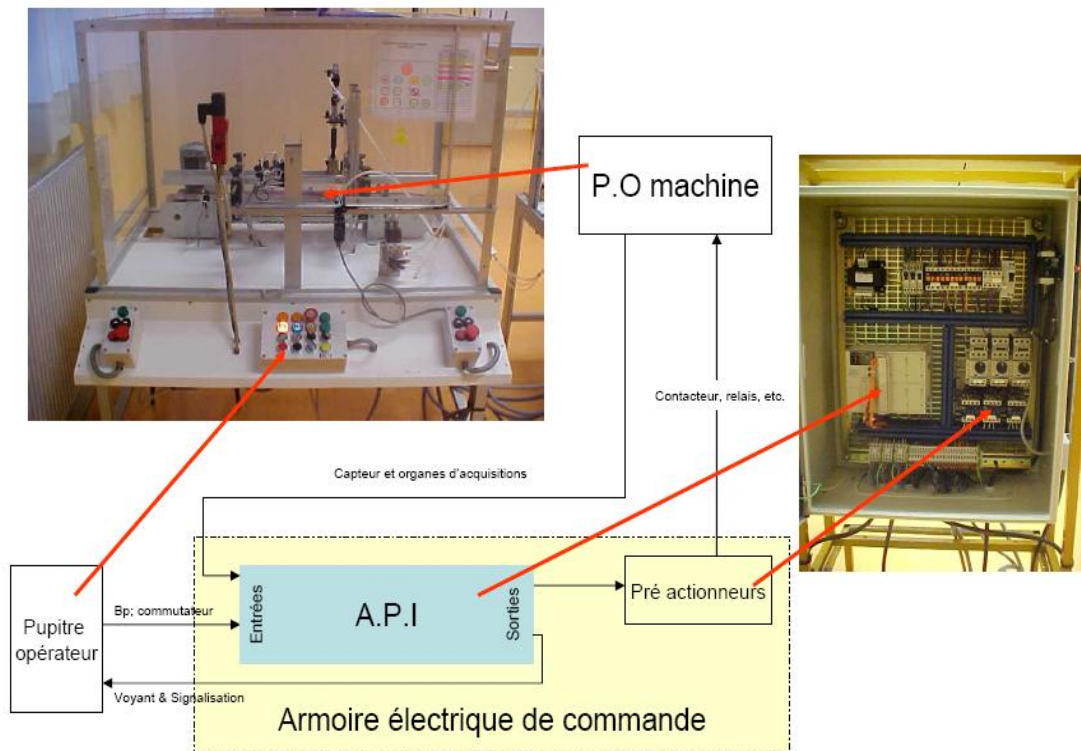
Il existe d'autres modes de fonctionnement, moins courants :

- synchrone par rapport aux entrées seulement ;
- asynchrone.



Cycle de Fonctionnement d'un API

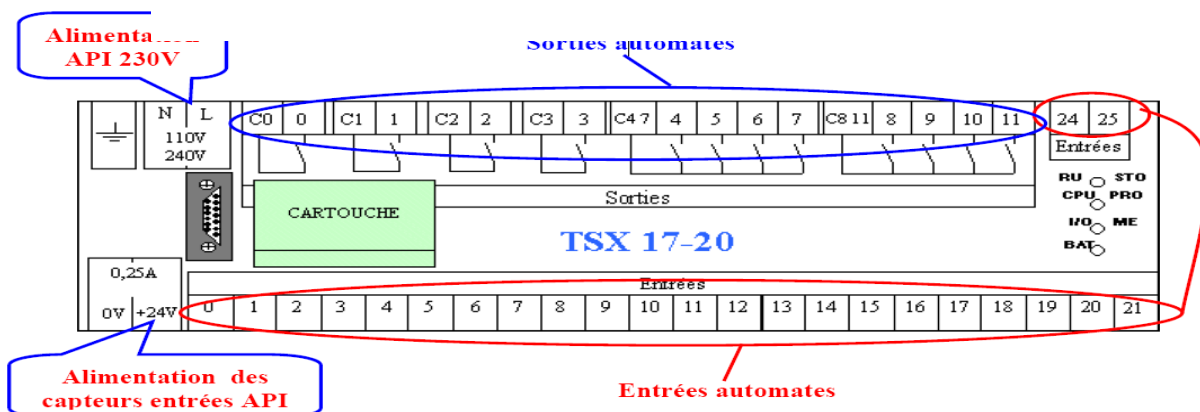
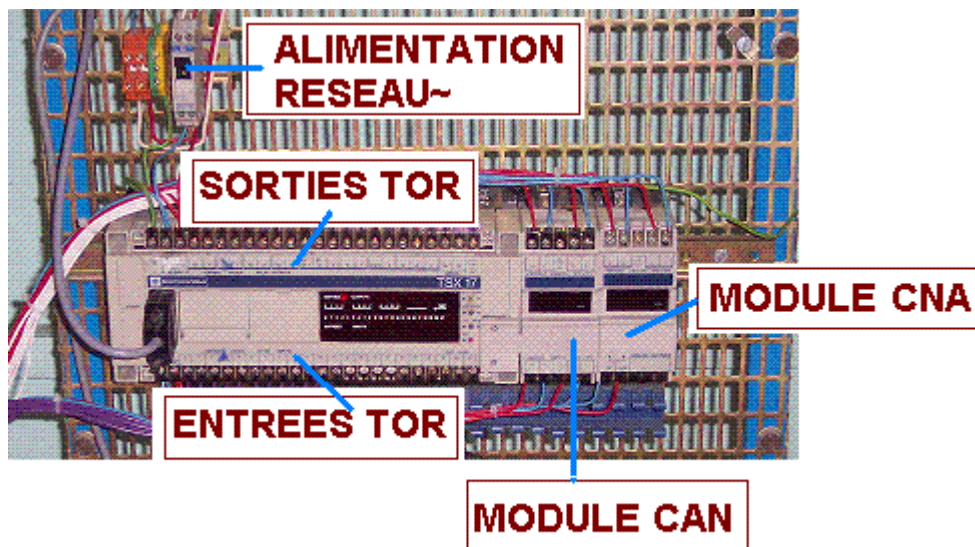
2.4. Mise en situation



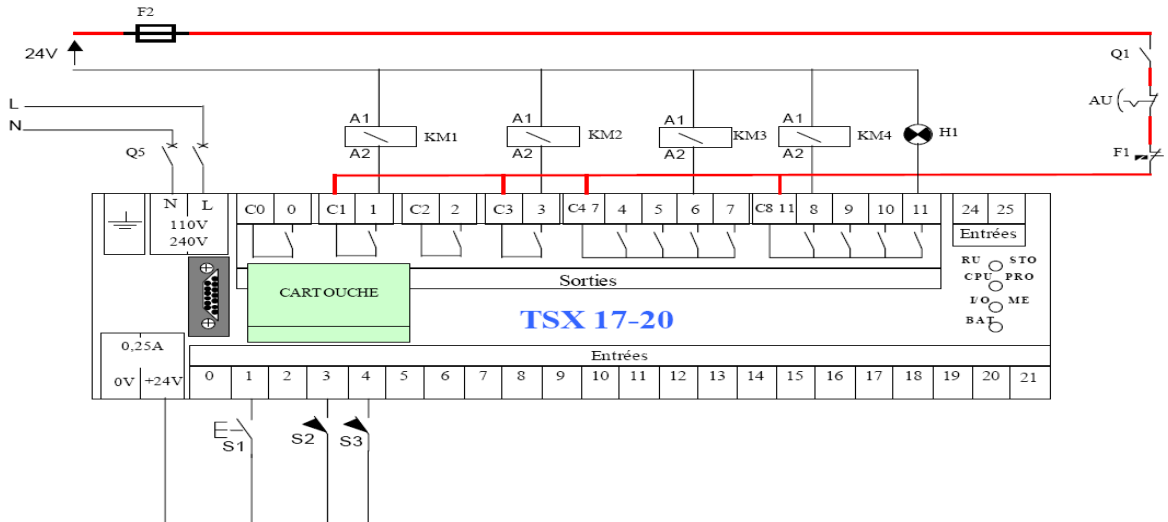
3. Raccordement automate

3.1. TSX 17-20 de chez Télémécanique

a)- Présentation



b)- Câblage des entrées/sorties



- Câblage des entrées

Elles disposent généralement d'une alimentation fournie par l'automate. De même, il y a un commun à toutes les entrées.

Il est possible de lui câbler des contacts classiques (contact sec sans potentiel) mais aussi des capteurs électroniques deux ou trois fils.

- Câblage des sorties

Certaines sorties peuvent avoir le même commun d'alimentation.

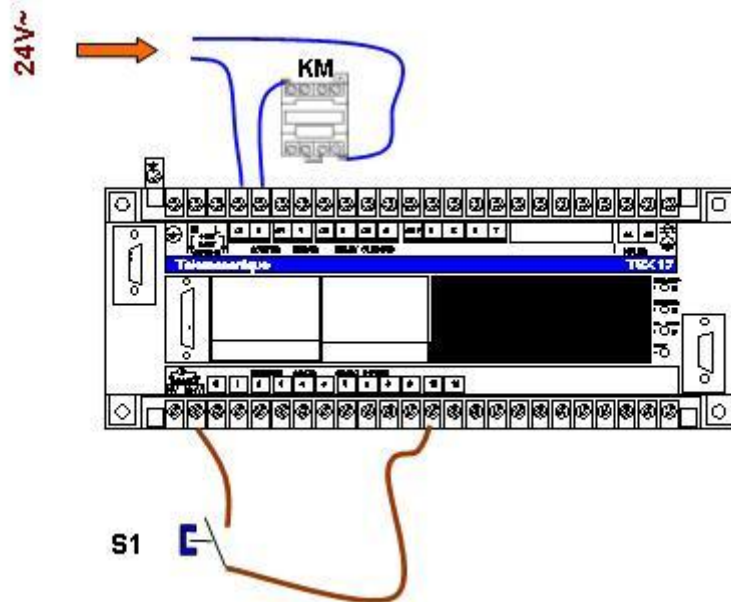
Les sorties de l'automate peuvent être sous la forme de contact ou sous la forme statique (ces dernières devront être relayées).

Rq: De plus en plus, la ou les premières sorties sont réservées pour assurer des fonctions de sécurité.

Exemple :

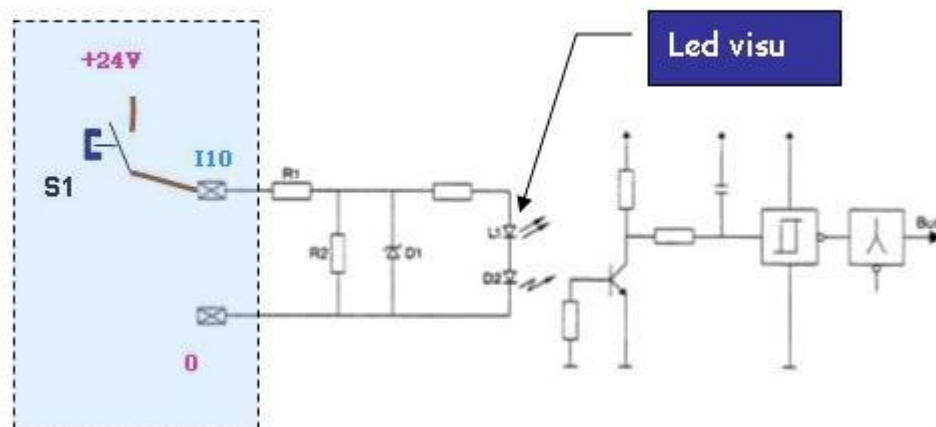
CABLAGE D'UNE ENTREE/SORTIE

un bouton poussoir sur l'adresse Ii,10 et un contacteur sur l'adresse Oj,0



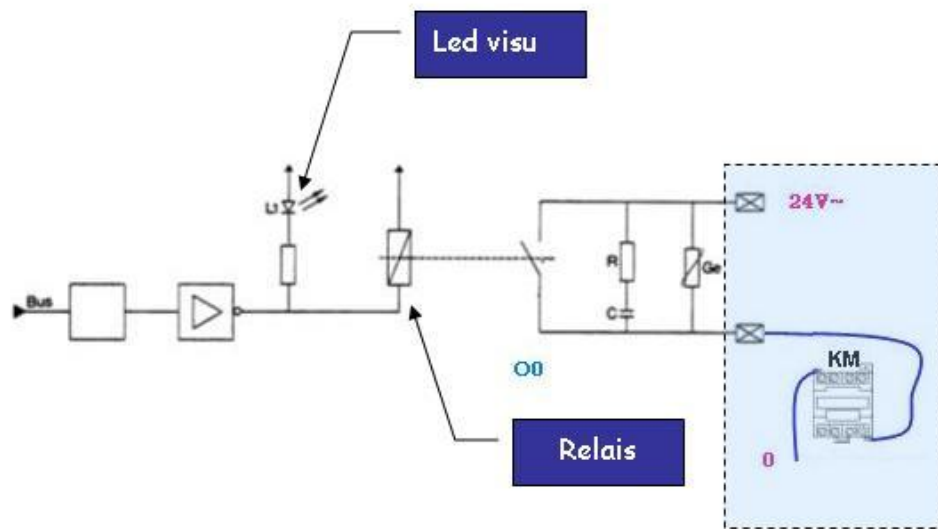
INTERFACE D'ENTREE TOR SUR L'API TSX

Le schéma ci dessous explique le fonctionnement de cette interface. L'action sur le BP informe l'opto coupleur de liaison au bus de données interne.

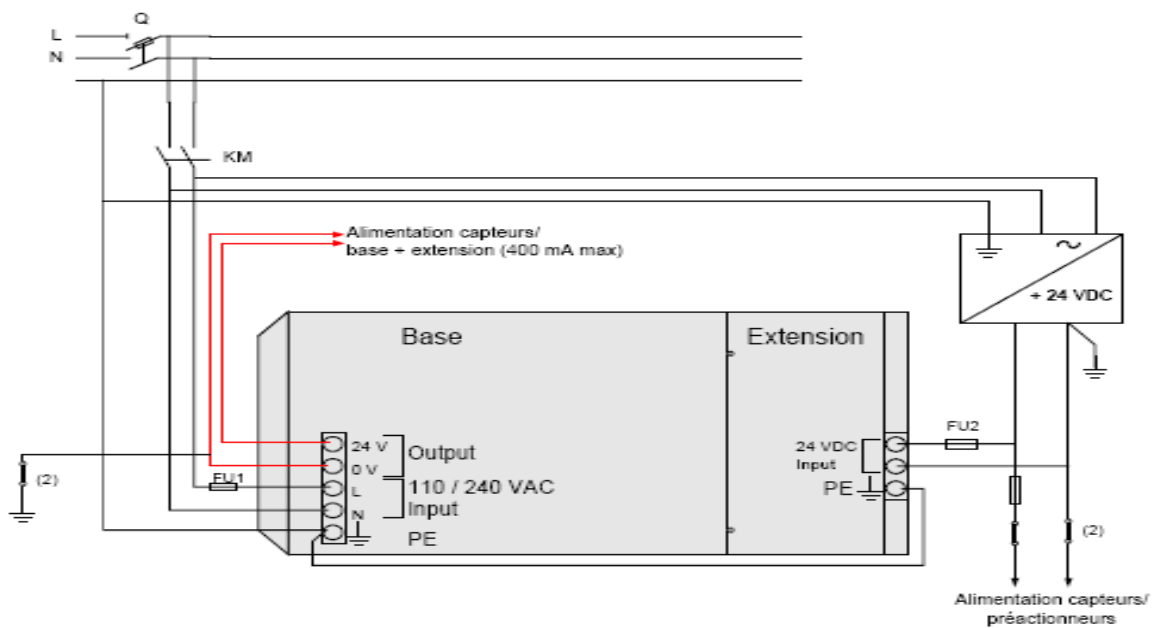


INTERFACE DE SORTIE TOR SUR L'API TSX

Le niveau de l'adresse permet au relais d'être sollicité ou non, son contact réagit et actionne ou non le contacteur.



c) câblage des alimentations



4. 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.

Sur une ligne de programme, on trouve systématiquement un code d'instruction, suivi éventuellement de l'adresse de l'opérande (variable) sur laquelle s'applique l'opération

4.1. Opérations logiques de base

Les opérations logiques de base sont :

- Lecture de l'état d'une variable (Load, If, ... etc);
- Et logique (AND);
- OU logique (OR, + ...);
- Affectation ou égal (=, SET, OUT);
- Négation ou PAS (NOT, Non, Pas).

4.2. Instructions complémentaires

Les instructions complémentaires sont la mémorisation, la temporisation, le comptage, le saut (avant ou arrière ou le deux),...

4.3. Langages d'automates

Les automates programmables industriels doivent pouvoir être utilisés facilement par du personnel habitué aux techniques classiques d'automatisation et peu à l'informatique. Ceci a conduit les constructeurs des API à concevoir des langages d'application spécialement adaptés à la réalisation d'automatisme. On distingue :

- 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).

Le langage à relais (Ladder Diagram) est basé sur un symbolisme très proche de celui utilisé pour les schémas de câblage des systèmes à relais. Les symboles les plus utilisés sont donnés au tableau suivant :

Fonction	Symbole	
	Européen	Américain
Contact normalement ouvert	---o o---	— —
Contact normalement fermé	---o—o---	— / —
Début de branchement	—┐	—┐
Fin de branchement	—┘	—┘
Affectation	---()---	---()

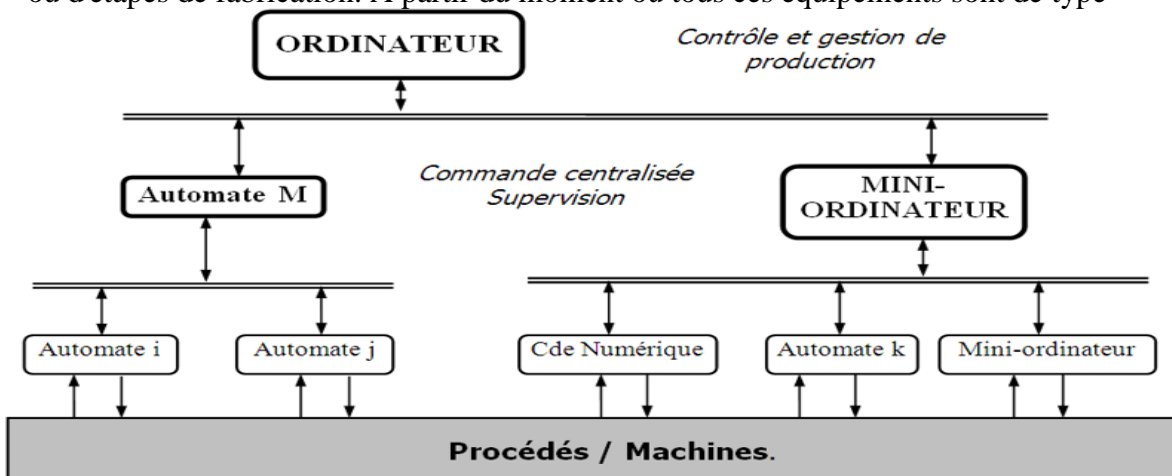
4.4. Logiciels de programmation

Marque	Automate	Logiciel
Télemécanique	TSX Nano	PI707
	TSX 3708, TSx22	PI7- micro
	TSX Premium	PI7 junior
ALENBRADLEY	SLC 500	APSF
SIEMENS	Serie 5:S5 Serie 7:S7	Step 5 Step 7

5. Réseaux d'automates

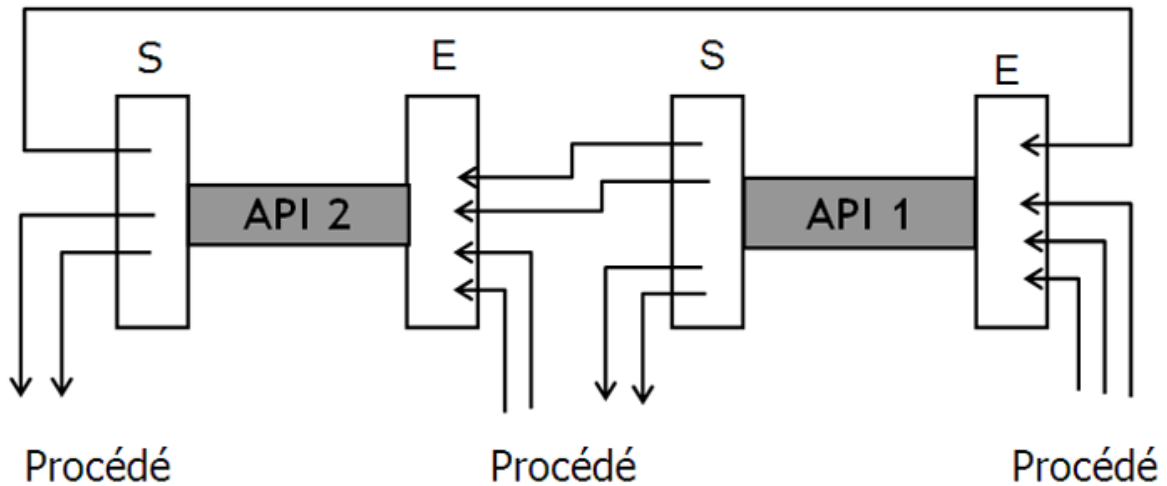
5.1. Principe

Les commandes d'automatismes câblés sont de plus en plus remplacées par de API ou de micro-ordinateur qui assurent la commande et le réglage des machines, d'éléments de machine ou d'étapes de fabrication. A partir du moment où tous ces équipements sont de type



informatique, il devient intéressant de les interconnecter à un mini-ordinateur ou à un automate de supervision

En conformité avec la philosophie des automates, les besoins de communication inter-automates sont couverts par des réseaux d'automates. L'interconnexion entre deux automates peut être réalisée très simplement en reliant certaines sorties d'un automate à des entrées de l'autre automate et vice-versa



Cette méthode ne permet pas de transférer directement des variables internes d'un automate sur l'autre, de sorte que celles-ci doivent être converties par programme en variables de sortie avant leur transfert. Elle devient coûteuse en nombre d'entrées/sorties mobilisé pour cet usage et lourde du point de vue du câblage, lorsque le nombre de variables qui doivent être échangées devient important.

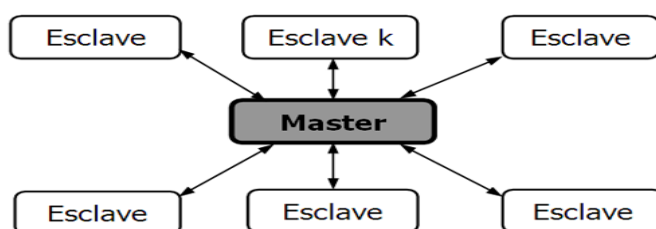
Il est préférable de transférer le contenu de la mémoire image d'un des API sur l'autre par l'intermédiaire d'une liaison série prise sur un coupleur RS 232C.

Dès que les échanges mettent en jeu plus de deux API, les méthodes précédentes ne conviennent plus, et il faut faire appel à un réseau d'automates sur lequel ne sont transmises que les données concernées par la communication, et qui met en œuvre un véritable protocole d'accès. Les constructeurs ont développés un grand nombre de types différents de réseaux d'automates.

5.2. Différents types de réseaux d'automates

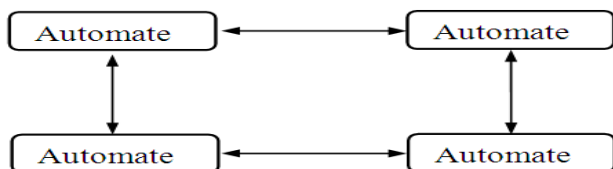
5.2.1. Réseau en étoile

Un centre de traitement commun échange avec chacune des autres stations. Deux stations ne peuvent pas échanger directement entre elles



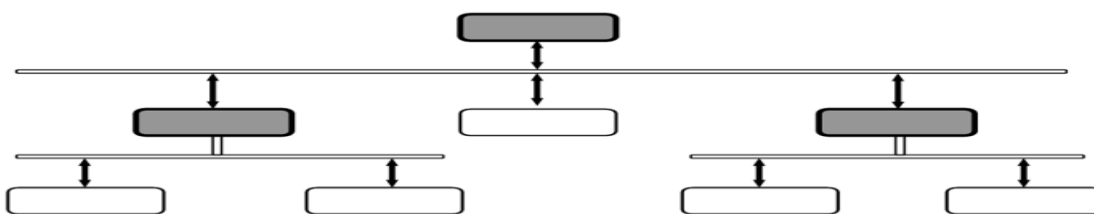
5.2.2. Réseau en anneau

Chaque station peut communiquer avec sa voisine. Cette solution est intéressante lorsqu'une station doit recevoir des informations de la station précédente ou en transmettre vers la suivante



5.2.3. Réseau hiérarchisé

C'est la forme de réseaux la plus performante. Elle offre une grande souplesse d'utilisation, les informations pouvant circuler entre-stations d'un même niveau ou circuler de la station la plus évoluée (en général un ordinateur) vers la plus simple, et réciproquement



6. Principaux automates programmables industriels

La programmation des automates se fait :

- soit à partir de leur propre console;
- soit à partir du logiciel de programmation propre à la marque.

TELEMECANIQUE

- **TSX 17/20 :**
 - Nombre d'entrées et de sorties variable : 20 à 160 E/S.
 - microprocesseur 8031.
 - langage de programmation PL7.2.
- **TSX 47.30 :** Le produit de milieu de gamme à très haute performance et très compact :
 - 512 E/S en deux bacs de 8 modules;
 - 5 coupleurs intelligents;
 - 24 K RAM data on Board;
 - 32 K RAM / EPROM cartouche utilisateur;
 - structure multifonction;

- structuration multitâche;
- langage PL7.3 (Grafcet, Litteral, Ladder);
- programmation sur IBM PC/PS.
- **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 – 200.**
 - 64 entrées 24 V DC;
 - 64 sorties 24 V DC - 1 A ;
 - 8 Entrées analogiques AEW0
 - AEW14 ; - 8 Sorties analogiques AAW0
 - AAW6 ; - interface série,
 - console de programmation externe : PG 702;
 - programmation STEP7: schéma à relais , Ladder.

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.

7. L'essai d'un automatisme simple commandé par un automate

7.1. Les dangers potentiels liés à l'utilisation d'un automate

Le forçage des entrées / sorties consiste à mettre à 1 le bit image de ces dernières grâce au logiciel de programmation.

Ou procède au forçage lorsque :

- Ou on ne dispose pas matériellement des entrées ou sorties. (absence d'un bouton poussoir pour la mise en marche d'un engin, absence d'un contacteur...)
- On veut déceler les défauts provenant des E/S.
- On veut voir l'évolution de la programmation du processus automatisme avant de passer au câblage.

Le forçage est prioritaire, il est conseillé de l'utiliser avec précaution car il présente certains dangers.

Exemple :

Lorsqu'on force à 1 une sortie automate qui commande l'ouverture d'une vanne évacuant de l'air chaud, celle-ci peut brûler les travailleurs, endommager le matériel autour.

7.2. L'essai d'un automatisme simple

On doit confirmer qu'un automatisme commandé par automate est fonctionnel, après avoir lancé l'exécution et vérifier que :

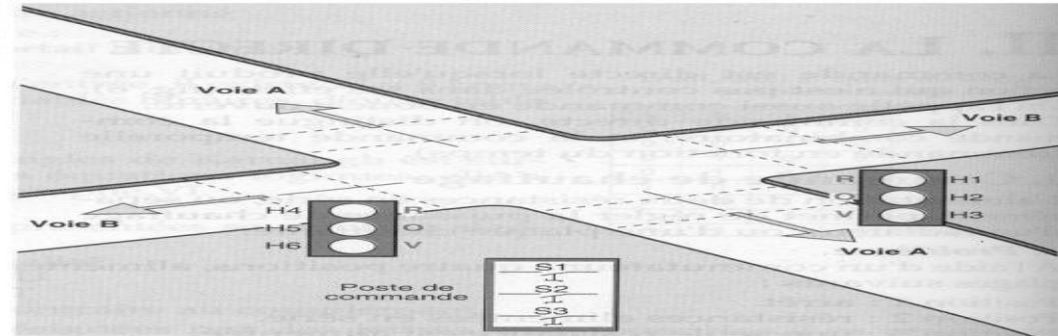
- L'ensemble fonctionne pendant une durée assez suffisante permettant d'atteindre les températures de fonctionnement afin de s'assurer que le système est fonctionnel et qu'il ne va pas tomber en panne à cause d'une dérive thermique de composants.

- Si on effectue un essai après avoir changé un ou plusieurs composants défectueux, on vérifie que tout entre en action après le changement effectué.

8. Les applications de l'automate

Les automates trouvent leur application en milieu industriel, domestiques. On cite quelques exemples courants :

Exemple n°1: Feux de carrefour



Description

On règle la circulation d'un carrefour de deux voies A et B par des feux tricolores (Rouges, orange, vert).

Exemple 2 : système de pompage

