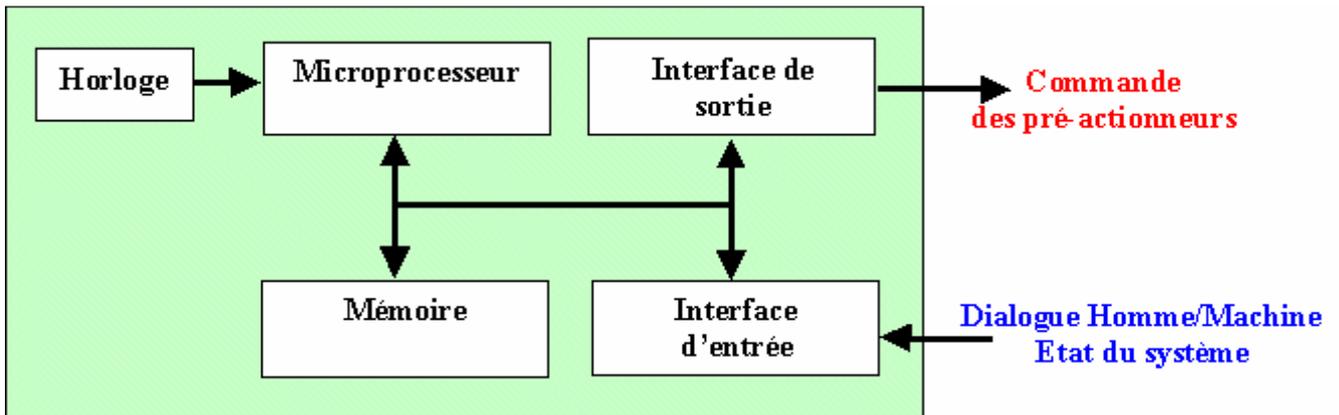


## Les automates industriels programmables

### D)- Architecture d'un API :

La structure interne d'un API peut se représenter comme suit :



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 se compose donc de trois grandes parties :

- Le processeur ;
- La zone mémoire ;
- Les interfaces Entrées/Sorties

#### 1)- Le microprocesseur :

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..

#### 2)- La zone mémoires :

##### a)- La Zone mémoire va permettre :

- De recevoir les informations issues des capteurs d'entrées
- 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 système.

##### b)- Action possible sur une mémoire :

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

NOM : .....	<b>Electrotechnique</b>	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....
		CHAPITRE :

**c)- 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 dans laquelle on ne peut que lire.
- **EPROM** mémoires mortes reprogrammables effaçables aux rayons ultra-violets.
- **EEPROM** mémoires mortes reprogrammables effaçables électriquement.

**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.

**3) Les interfaces d'entrées/sorties :**

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

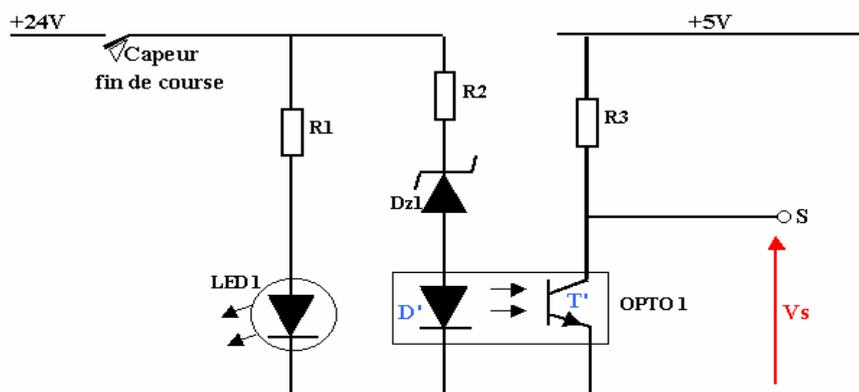
Les **sorties** transmettent des informations aux **pré-actionneurs** et aux **éléments de signalisation** du pupitre.

**a)- Interfaces d'entrées :**

Elles sont destinées à :

- Recevoir l'information en provenance du capteur
- Traiter le signal en le mettant en forme, en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative.

**Fonctionnement de l'interface :**



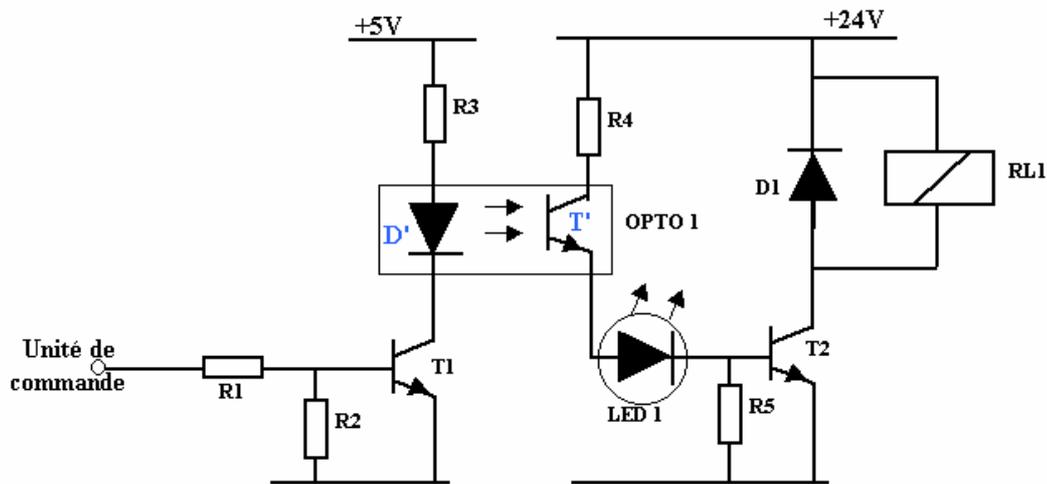
NOM : .....	<b>Electrotechnique</b>	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....
		CHAPITRE :

***b)- Interfaces de sorties :***

Elles sont destinées à :

- Commander les pré-actionneurs et éléments des signalisation 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 :***



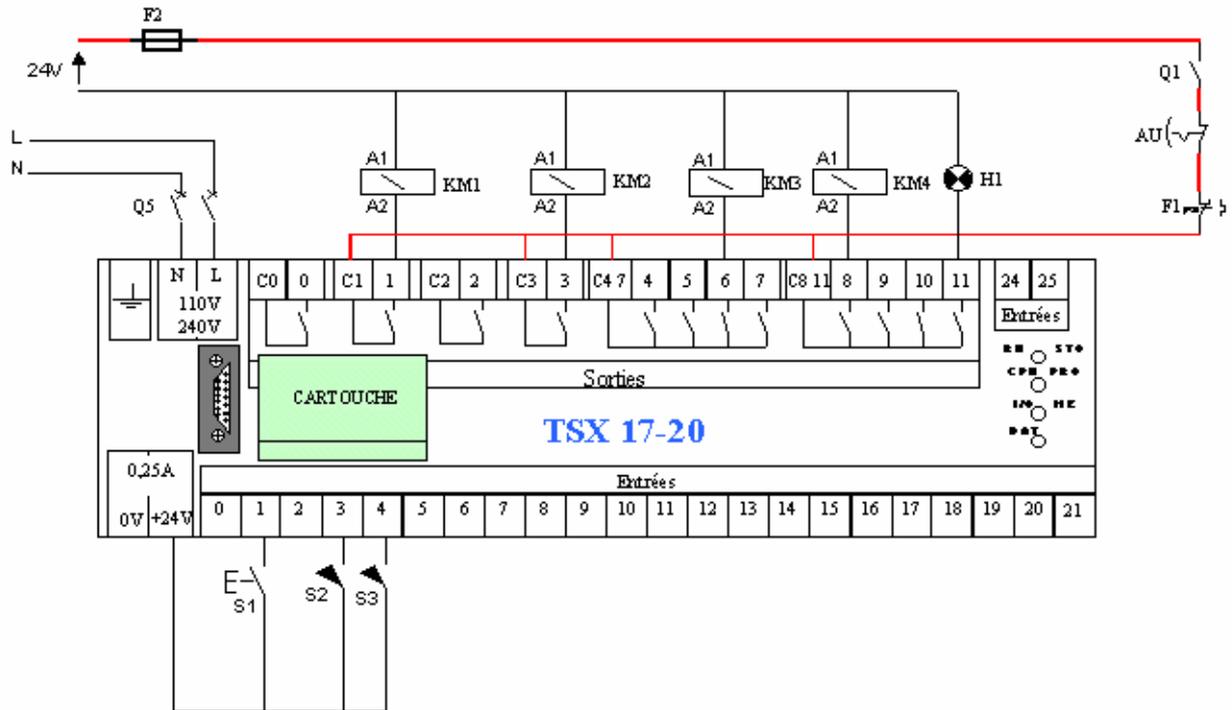
**4)- Alimentation de l'automate programmable industriel :**

L'alimentation intégrée dans l'API, fournit à partir des tensions usuelles des réseaux ( **230 V, 24 V=** ) les tensions continues nécessaire au fonctionnement des circuits électroniques.

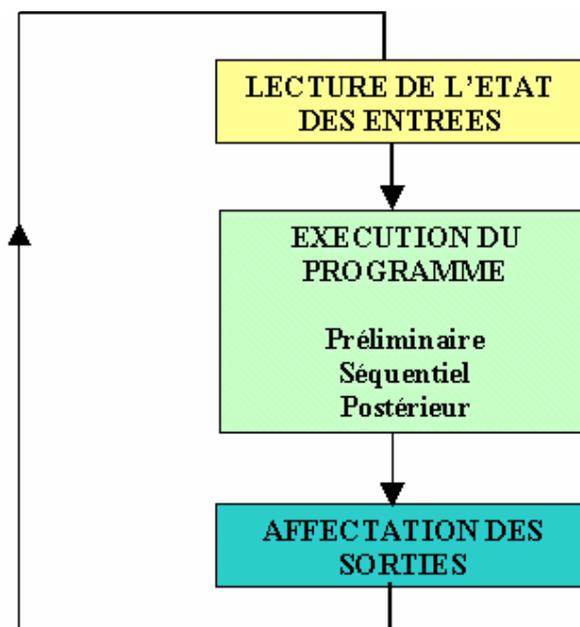
NOM : .....	Electrotechnique	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....
		CHAPITRE :

## II)- API TSX 17 de chez télémeccanique

### 1) Câblage des entrées/sorties



### 2)- Fonctionnement du TSX



NOM : .....	Electrotechnique	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....  CHAPITRE :

### 3)- Programmation du TSX en langage PL7-2

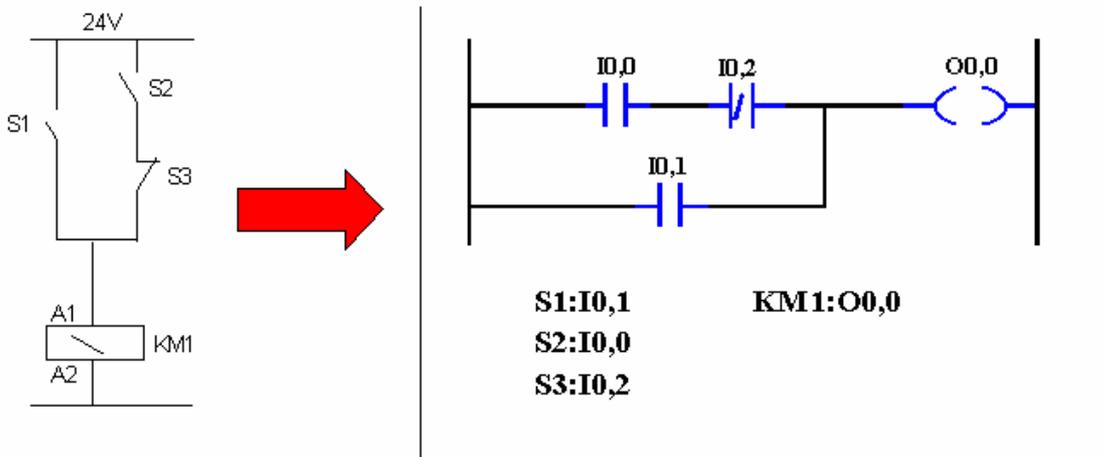
Le PL7-2 associe deux langages :

- Le Ladder " schéma à contacts "
- Le Grafcet " Chart "

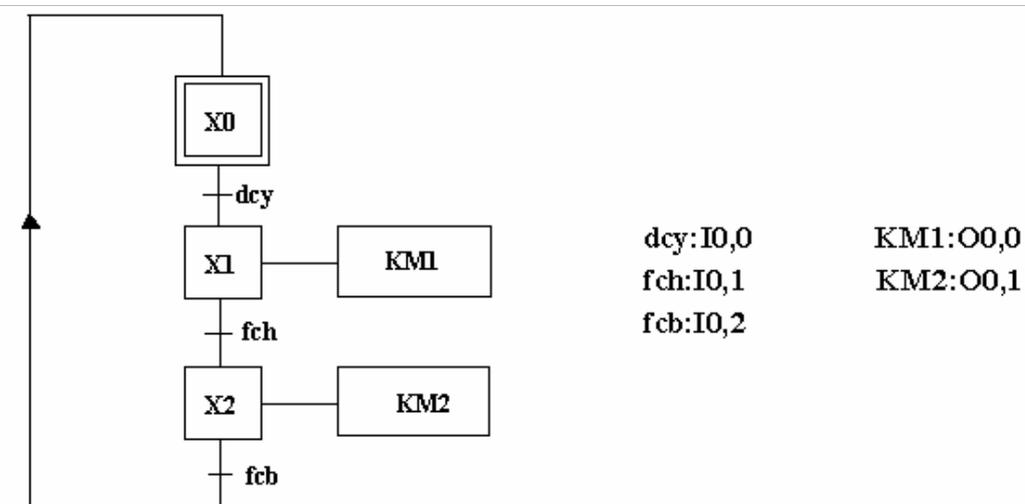
#### a)- Le ladder

Le langage Ladder est une succession " de réseaux de contacts " véhiculant des informations logiques depuis les entrées vers les sorties. Le résultat dépend des fonctions programmées.

Exemple :

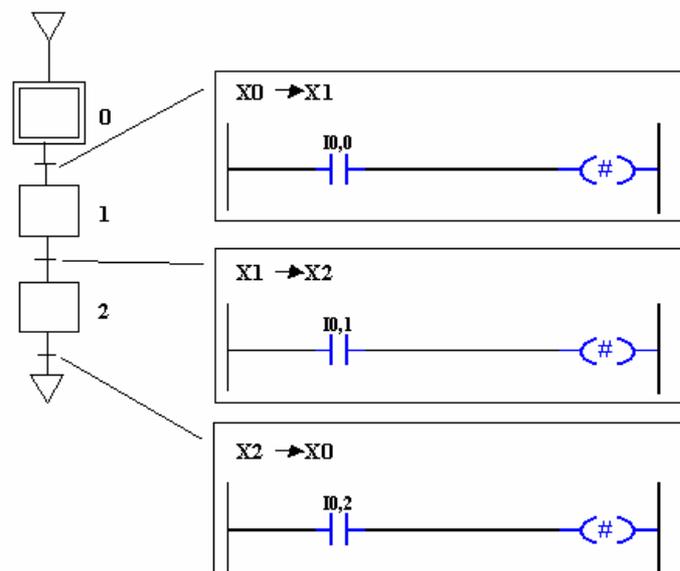


#### b)- Le Grafcet " Chart "



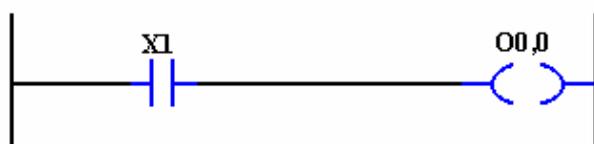
NOM : .....	<b>Electrotechnique</b>	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....
		CHAPITRE :

La construction du Grafcet se fait en **CHART** comme ci-dessous :

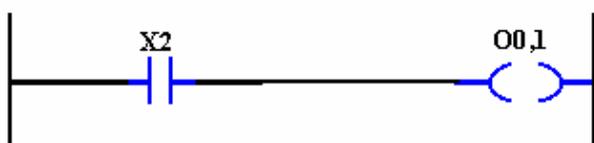


L'activation des sorties associées aux étapes du Grafcet s'effectue dans le **POSTERIEUR**

Label 1



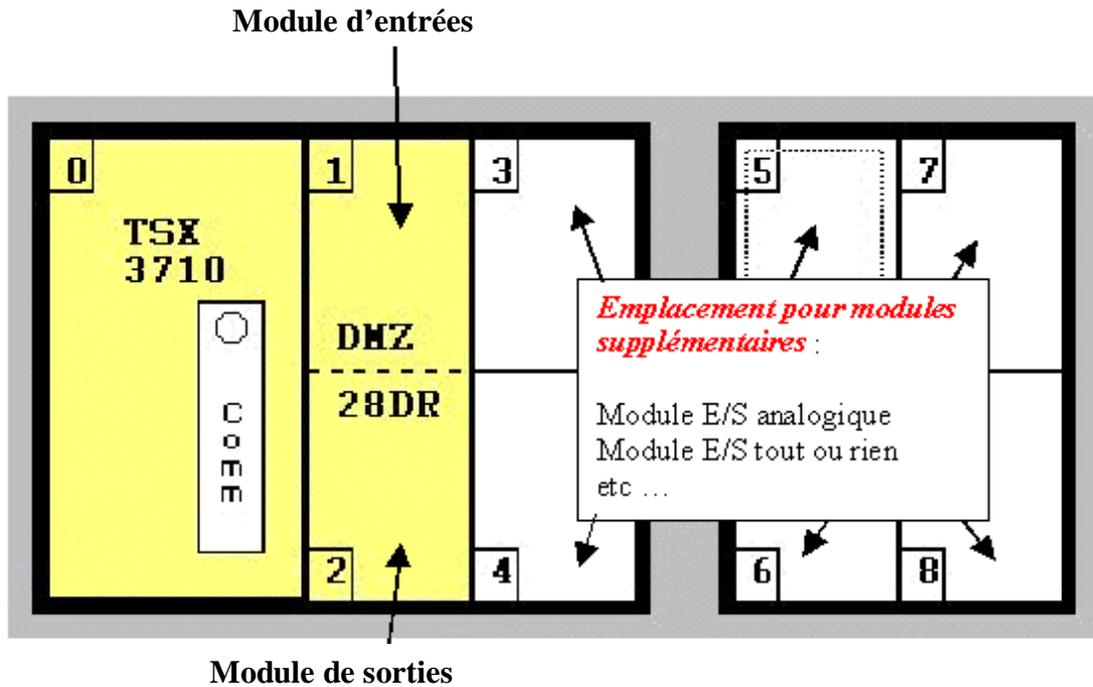
Label 2



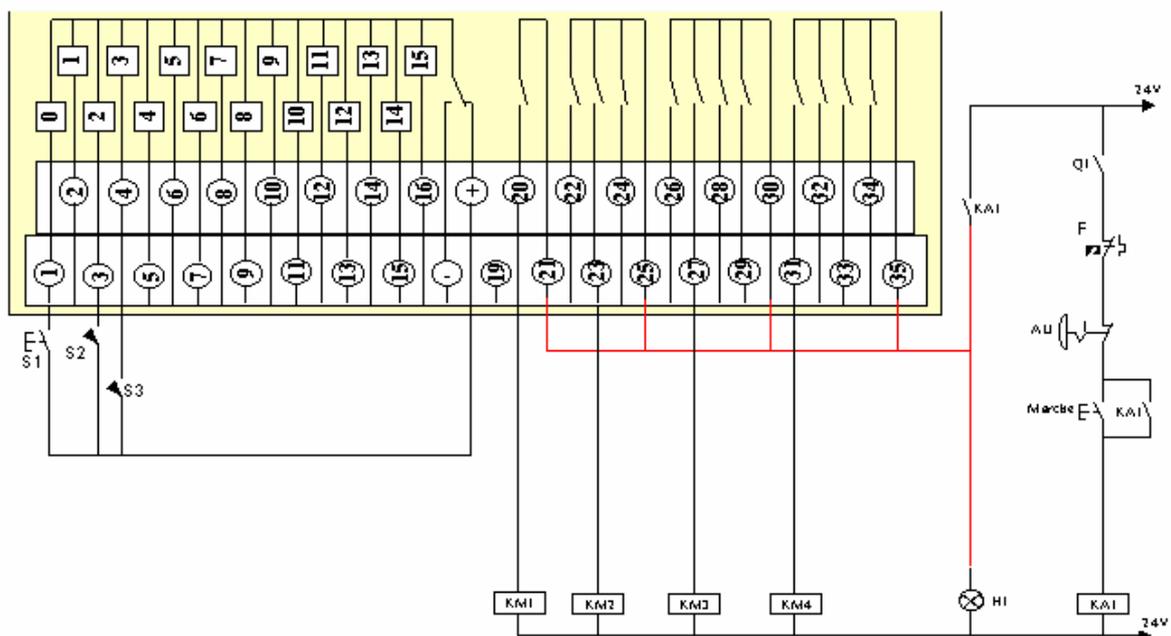
NOM : .....	<b>Electrotechnique</b>	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....
		CHAPITRE :

### III)- API TSX micro de chez télémécanique

#### 1)- Configuration matériel du TSX micro



#### 2) Câblage des entrées/sorties



NOM : .....	Electrotechnique	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....  CHAPITRE :

### 3)- Programmation du TSX micro en langage PL7-micro

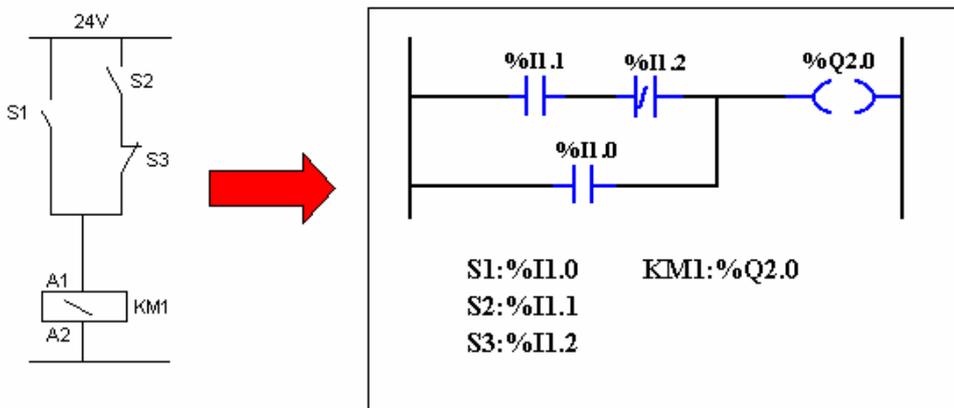
Le PL7-micro associe deux langages :

- Le Ladder " schéma à contacts "
- Le Grafcet " Chart "

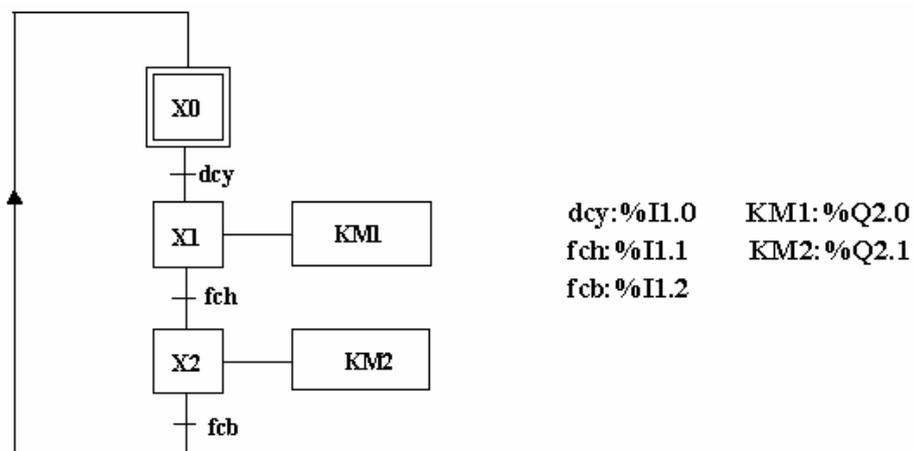
#### a)- Le ladder

Le langage Ladder est une succession " de réseaux de contacts " véhiculant des informations logiques depuis les entrées vers les sorties. Le résultats dépend des fonctions programmées.

Exemple :

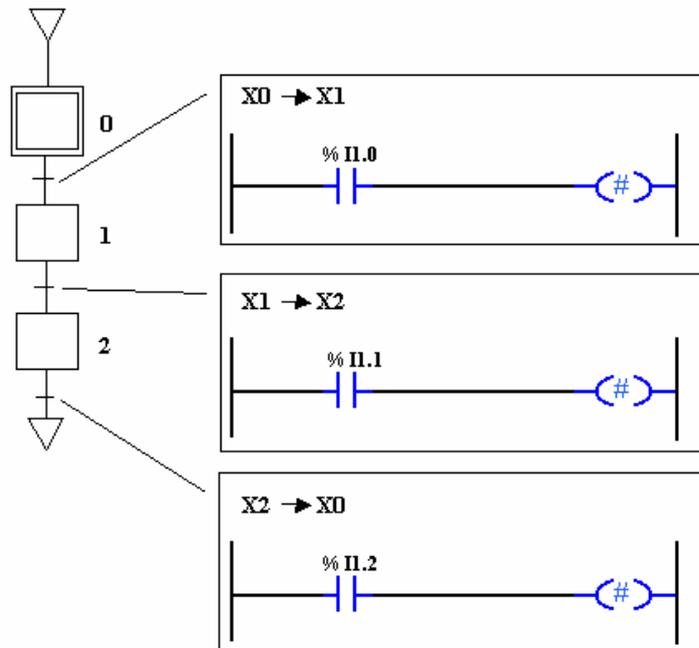


#### b)- Le Grafcet " Chart "



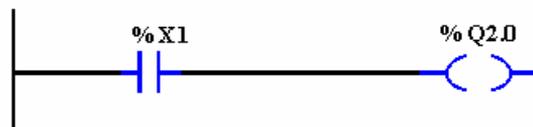
NOM : .....	<b>Electrotechnique</b>	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....
		CHAPITRE :

La construction du Grafcet se fait en **CHART** comme ci-dessous :

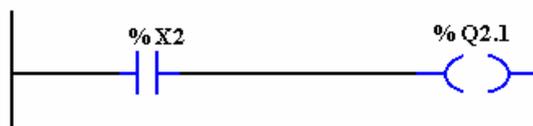


L'activation des sorties associées aux étapes du Grafcet s'effectue dans le **POSTERIEUR**.

**Label 1**



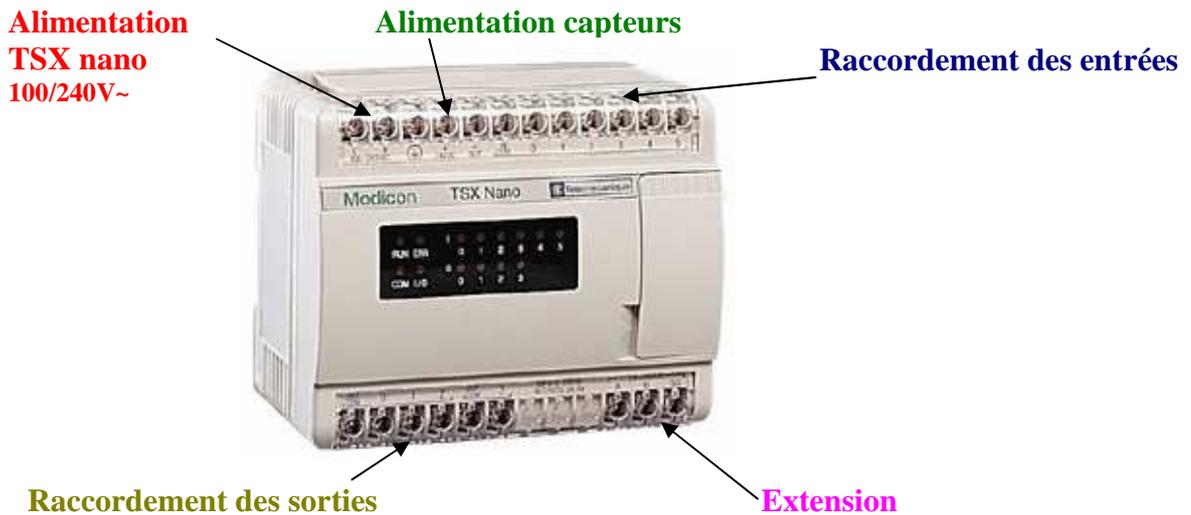
**Label 2**



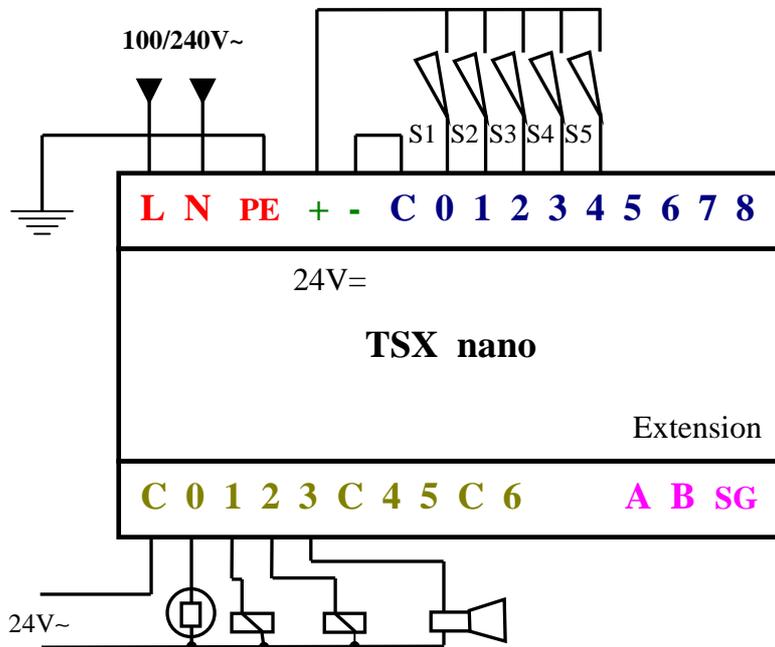
NOM : .....	<b>Electrotechnique</b>	CLASSE : .....
PRENOM : .....	<b>SUJET : L'AUTOMATE PROGRAMMABLE</b>	DATE : .....
		CHAPITRE : .....

### IV)- API TSX nano de chez télémécanique

#### 1)- Configuration matériel du TSX nano



#### 2) Câblage des entrées/sorties



#### 3)- Programmation du TSX nano en langage PL7-nano

La programmation du TSX nano se fait en langage PL7-nano. C'est un langage à contact du même type que le langage « ladder » du PL7-2 sur TSX 17.