Résumé : *Les problèmes d'automatisme rencontrés dans l'hydraulique ne font qu'appel à une solution technologique programmable, ils nécessitent l'imbrication du dispositif automate programmable. Cependant cette automatisation doit être maîtrisée par connaissances des notions de bases des automates programmables d'une part, et par le savoir faire programmer.*

Le GRAFCET (GRAphe de Commande Etape-Transition) est un outil de représentation du cahier des charges d'un système séquentiel automatisé. La représentation graphique par GRAFCET est très utile, pour programmer l'automate et représenter le fonctionnement du système hydraulique en vue de l'automatisé.

Chapitre 2 : Système automatisé et logique Séquentiel à base de l'automate programmable

1. Introduction

D'une façon générale, tout système automatisé peut se décomposer en deux parties qui coopèrent : la partie opérative (P.O) et la partie commande (P.C).

La partie opérative est constituée du procédé physique à commander, la partie commande l'automatisme proprement dit, qui élabore en sortie des ordres destinés au procédé en fonction des comptes-rendus venant du procédé et des consignes qu'il reçoit en entrée.

L'automatisation des systèmes hydrauliques basés sur la logique câblée sont devenues trop complexe à réaliser. L'introduction d'une nouvelle technique comme la logique programmable est devenue nécessaire. Un automate programmable industriel (API) est un dispositif électronique qui assure l'automatisation (la partie commande) dans une logique programmée. La programmation est un des atouts majeurs des API puisqu'elle permet une multitude de traitements des informations reçues sans toucher à la configuration matérielle.
La représentation graphique ou la programmation par GRAFCET est intéressante, pour programmer et représenter le fonctionnement d'un système séquentiel automatisé.

Le GRAFCET (GRAphe de Commande Etape-Transition) est un outil de représentation du cahier des charges d'un automatisme séquentiel complexe permettent de pallier aux différents inconvénients liés aux méthodes classiques de description et de synthèse des automatismes logiques.

2. Composition d'un système automatisé

Dans le schéma suivant, on donne la structure fonctionnelle d'un système automatisé, on distingue une partie opérative, où les actionneurs agissent sur le processus, et une partie commande où les automates récupèrent les informations sur l'état de ce processus à l'aide des capteurs et coordonnent les actions.

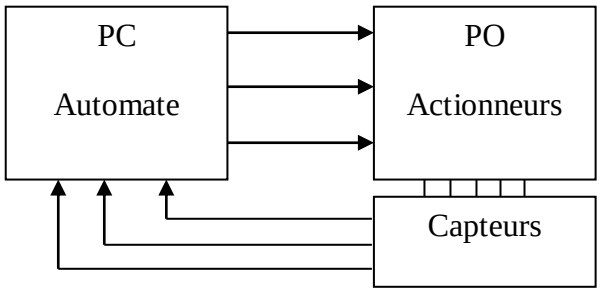

Figure 2.1 : Schéma fonctionnel d'un système automatisé

3. Partie commande (Automate)

La structure matérielle interne d'un automate programmable est donnée par la figure suivante :

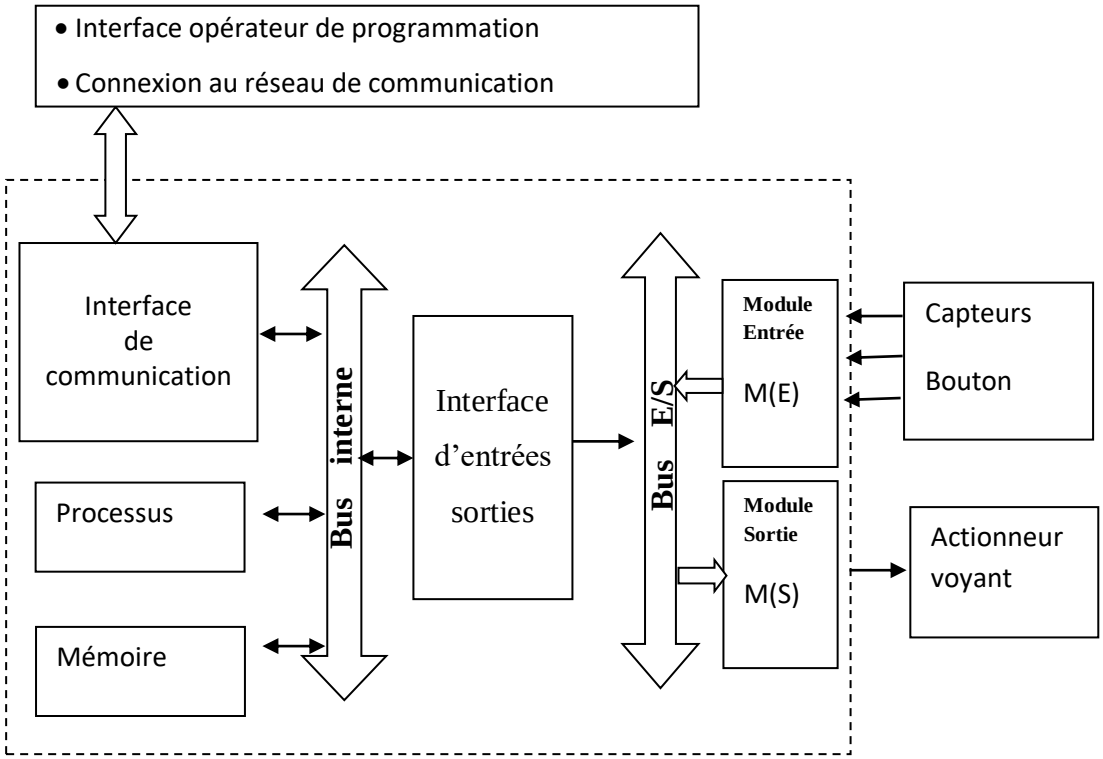

Figure 2.2 : Structure interne de l'automate

3.1 Processeur

Cœur de l'appareil, dans l'unité centrale UC est une carte électronique bâtie autour de la (ou des) « puce(s) » processeur(s), qui assure au moins les fonctions suivantes :

- Opérations logiques sur bits (le bit, contraction de « binary digit », étant l'information élémentaire à deux états) ou sur mots (ensemble de bits, le plus souvent 16 bits pour les API) ;
- Temporisation et comptage.
- Opération arithmétique.

3.2 Interface d'entrée

Module d'entrée analogique

• Mesure une tension ou un courant électrique et le convertit en une valeur numérique (ADC). Adapté à des capteurs de température, de pression ou autres variables continues.

Module d'entrée logique

• Mesure un signal binaire (deux états). Adapté à des interrupteurs de fin de course, des capteurs de proximité, des détecteurs photoélectriques, capteur de niveau à contact ou d'autres interrupteurs (manuels ou non).

3.3 Interface de Sortie

Module de sortie analogique

• Génère une tension ou un courant électrique proportionnel à une valeur numérique (DAC). Adapté aux moteurs (AC et DC), aux électrovannes proportionnelles, …

Module de sortie logique

• Génère un signal de contrôle binaire (deux états). Adapté aux lampes témoin, un relais électromagnétique, aux systèmes de verrouillage de porte, …

3.4 Mémoire (Éléments de stockage)

Le stockage des données et des programmes s'effectue dans des **mémoires**. La mémoire vive (RAM) est volatile mais secourue par batterie. La mémoire morte (ROM) dont l'utilisateur ne peut que lire le contenu, contient le système d'exploitation, tandis que les programmes installés (utilisables) peuvent se stocker dans des mémoires reprogrammables (EEPROM) ou mémoire flash.

La capacité de stockage d'une mémoire s'exprime en kilooctets (kO) : 1 kO = 1024x8 bits). Il faut connaître la capacité minimale utile de l'API, mais aussi la capacité maximale que l'on peut obtenir par diverses extensions. La mémoire des automates est très inférieure à celle des microordinateurs.

3.5 Interface de communication et de liaison

Les **liaisons** s'effectuent :

- avec l'extérieur par des **borniers** (à visser, à clipser, etc.) sur lesquels arrivent des câbles transportant le signal électrique ;
- avec l'intérieur par des **bus**, liaisons parallèles entre les divers éléments, il peut y avoir plusieurs bus, car on doit transmettre des données, des états, des adresses.

4. Branchement de la partie Commande avec la partie Opérative

4.1 Branchement des modules d'entrées/sorties de l'automate (E/S)

Les modules d'entrées/sorties assurent le rôle d'interface de la partie commande (PC) et la partie opérative (PO) (schéma de la figure **2.3)**, où l'automate récupérant les informations sur l'état de ce processus et coordonnant en conséquence les actions, les actionneurs agissent physiquement sur le système pour atteindre les objectifs prescrits.

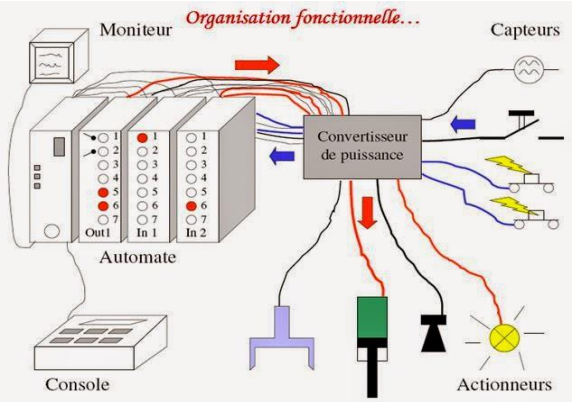

Figure 2.3 : câblage de la partie opérative et la partie commande

• Les modules d'entrées traduisent les signaux industriels (tension, courant, résistance, pulsation, …) en information logique ou numérique interprétable par le processeur (CAN).

Inversement, les modules de sorties traduisent les commandes du processeur en des signaux industriels (CNA).

• Ces modules comportent 1, 4, 8, 16 ou 32 voies (ports) d'entrée et/ou de sortie (figure 2.4).

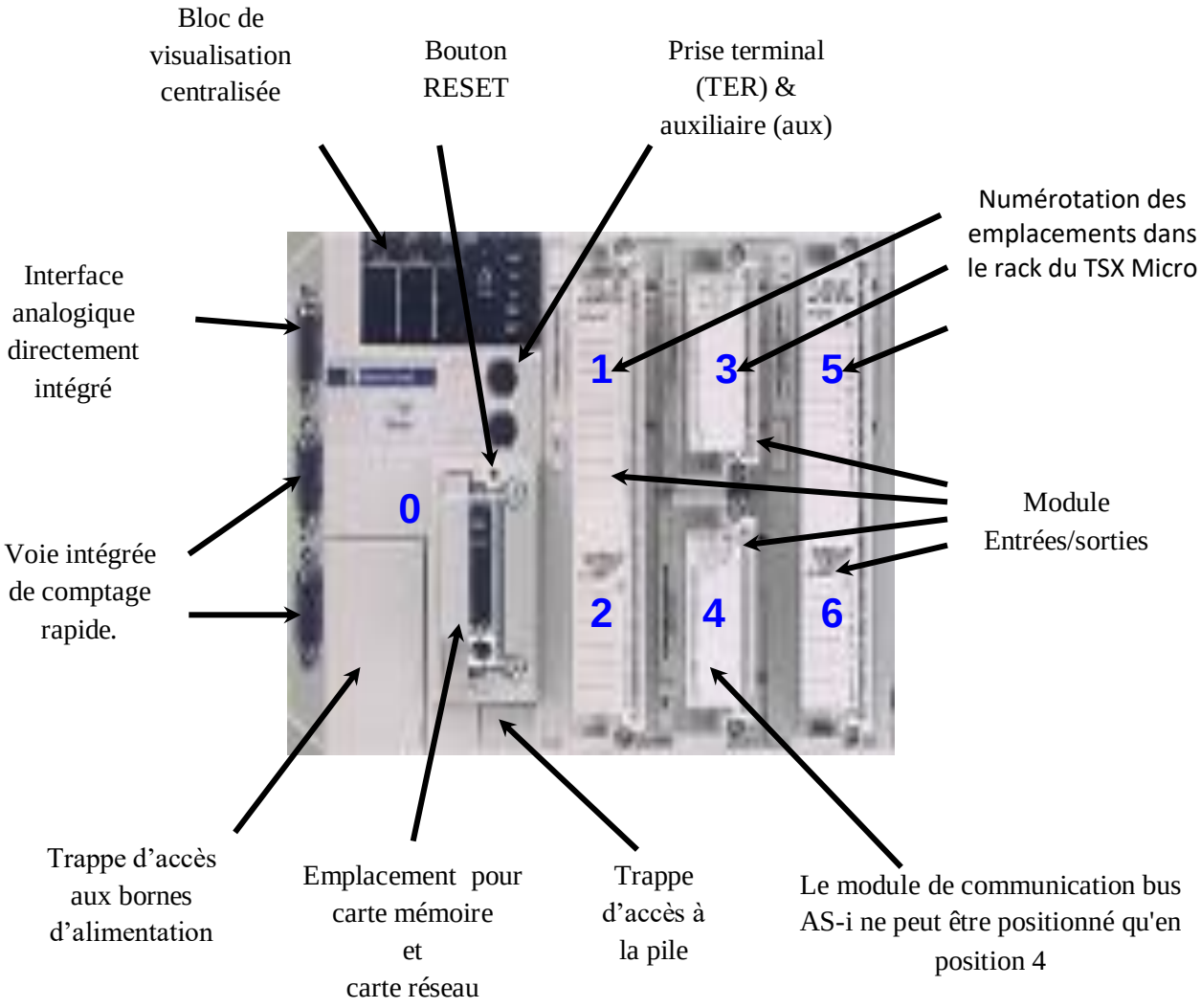

Figure 2.4 : Descriptif physique des modules d'entrées/sorties de l'automate TSX 37

4.2 Affectation et écriture des entrées/sorties

L'affectation des entées et des sorties permet de faire **l'adressage** entre le matériel (capteurs, actionneurs) et l'automate (API) en fonction de sont 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 analogiques seront notées par la lettre Q (Output)

les entrées analogiques seront notées par la lettre I (Input)

y : c'est l'emplacement physique du module analogique

z : c'est le numéro de la voie utilisée

Ainsi l'adressage est donné par le tableau suivant :

Variable	Repère	Désignation
Entrée (I input)	**%Ix.i**	**X : N° module** **i : N° de voie**
Sortie (Q output)	**%Qx.i**	**X : N° module** **i : N° de voie**
Variable d'étape	**%Xi**	**X : étape,** **i : N° étape**
Mémoire mot	%MWi	i : N° du mot interne
Temporisateur	%TMi	i : N° du temporisateur
Compteur	%Ci	i : N° du compteur
Mémoire bit	%M.i	i : N° du bit interne

4.3 Programmation de l'automate

La programmation est un des atouts majeurs des API puisqu'elle permet une multitude de traitements des informations reçues sans toucher à la configuration matérielle.

La norme CEI (Commission Electrotechnique Internationale) définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces cinq langages sont :

1. Le langage à contacts LD (Ladder Diagram)
2. Liste d'instruction IL (Instruction List)
3. Blocs Fonctionnels FBD (Function Block Diagram)
4. Langage littérale structuré ST (Structured Text)
5. Le langage SFC (Sequential Function Chart), ou **GRAFCET**.

Le GRAFCET, langage de spécification, est utilisé par certains constructeurs d'automate (Schneider, Siemens) pour la programmation. Parfois associé à un langage de programmation, il permet une programmation aisée des systèmes séquentiels tout en facilitant la mise au point des programmes ainsi que le dépannage des systèmes.

La programmation d'un API s'effectue à l'aide de langages spécialisés, fournis par le constructeur (exemple : Step7 pour Siemens). Chaque automate se programme via une console de programmation propriétaire ou par un ordinateur équipé du logiciel spécifique (step7, Automgen) et connecté à l'automate.

Une fois l'automate est programmé on doit le branche avec la partie opérative (voir les étapes décrit au paragraphe 4), ainsi le système automatisé étudié est fonctionnel.

5. Rappel des principes du GRAFCET

Le GRAFCET (Graphe Fonctionnel de Commande Etapes- Transitions) est un outil graphique de représentation du cahier des charges d'un automatisme séquentiel. Il est basé sur les notions d'**étapes** auxquelles sont associées des **actions** et de **transitions** auxquelles sont associées des **réceptivités**. Il décrit les ordres émis par la partie commande vers la partie opérative en mettant en évidence les actions engendrées.

Le GRAFCET se compose des éléments suivants :

- Étapes auxquelles sont associées des actions;
- Transition auxquelles sont associées des réceptivités;
- Liaisons orientées reliant les étapes aux transitions et les transitions aux étapes.

5.1 Etape

Une étape est symbolisée par un carré repéré numériquement, (Figure 2.5).

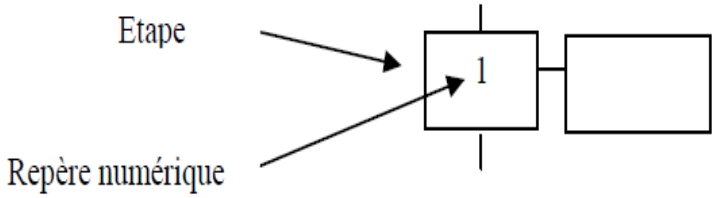

Figure 2.5: Représentation d'une Etape

L'étape initiale est représentée par un double carré, (Figure 2.6).

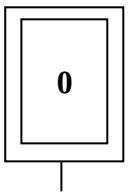

Figure 2.6 : Représentation d'une Etape Initiale

5.2 Action associée à une étape

Une action est symbolisée par un rectangle relié au symbole de l'étape associé, **(Figure 2.7).**

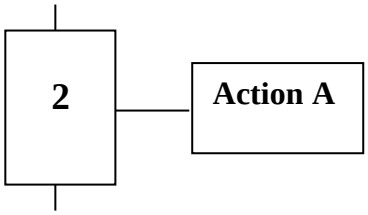

Figure 2.7: Représentation d'une action

5.3 Transition

Une transition est représentée par une barre perpendiculaire à la liaison orientée. Une transition indique la possibilité d'évolution entre deux étapes successives, (**Figure 2.8**).

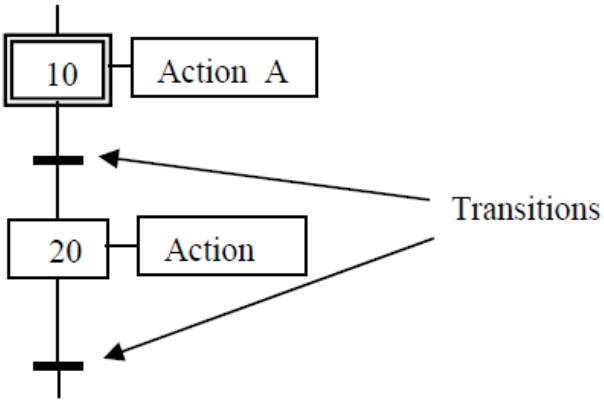

Figure 2.8: Représentation d'une Transitions

Une transition est soit validée soit non validée. Elle est dite validée lorsque toutes les étapes immédiatement précédentes reliées à cette transition sont actives.

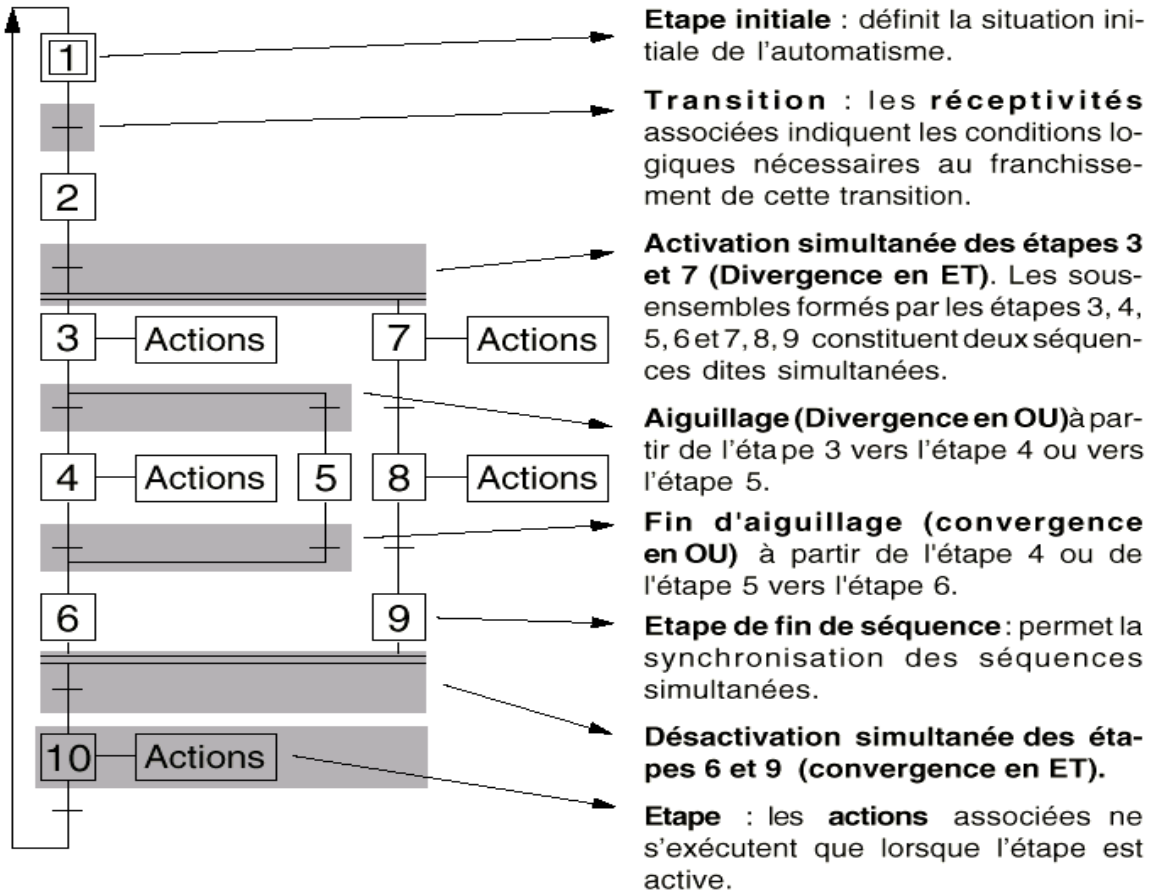

Figure 2.9 : Principe synthétique du GRAFCET

Exemple

On souhaite programmer un automate en langage GRAFCET qui commande la montée et la descente d'une presse à l'aide d'un vérin simple effet (figure 2.10). Sur l'ordre (m) de l'opérateur la presse descend, puis une fois arrivée en (b), elle remonte automatiquement jusqu'à la position haute (h)

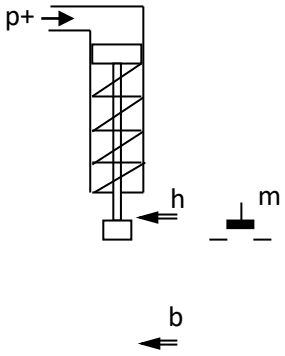

Figure 2.10 : Presse à vérin simple effet

Le GRAFCET est donné par la figure 2.11

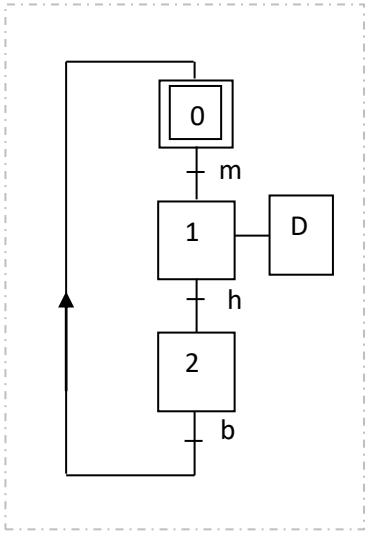

Figure 2.11 : GRAFCET de commande de la presse