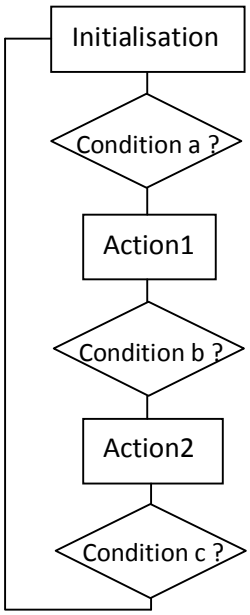
Chapitre 3: La Partie Commande

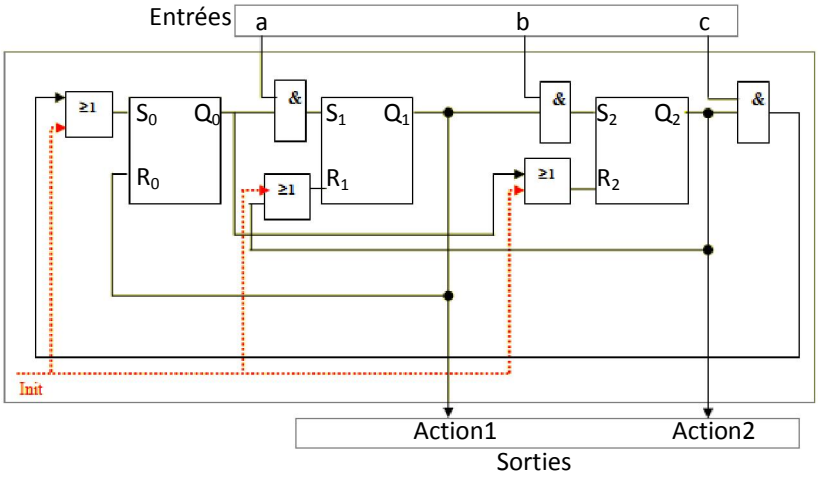
3.1. Introduction :

La partie de commande a pour tâche de donner les ordres de fonctionnement à la partie opérative. Elle reçoit les consignes de l'opérateur et les informations transmises par les capteurs. En fonction de ces consignes et de son programme. Elle va commander les préactionneurs et renvoyer des informations aux systèmes de supervision. Deux solutions sont empruntées pour la réalisation de la partie commande :

3.2. Logique câblée :

L'automatisme est obtenu en reliant entre eux les différents constituants de base ou fonctions logiques (combinatoire et séquentielle) par câblage.





Exemple de réalisation de la partie commande d'un système automatisé linéaire à l'aide de câblage par des bascules RS.

3.3. Logique programmée :

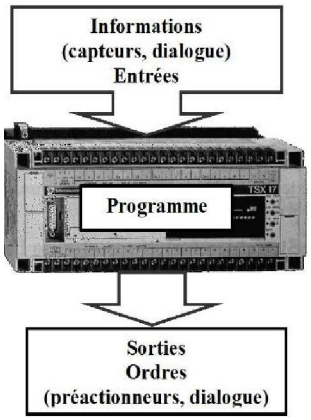
Le schéma du système est transcrit en une suite d'instructions constituant le programme, qui s'exécute par un équipement spécial (automate programmable API, microprocesseur,…). En cas de modification, l'installation ne comporte aucune modification de câblage seul le jeu d'instructions est modifié.

Le tableau suivant présente les principales différences entre les deux types : câblée et programmée

	Câblée	**Programmée**
Usage	S'utilise pour des systèmes simples	S'utilise pour des systèmes complexes.
Complexité	la taille des circuits croit avec la complexité du problème.	La taille de circuits n'augmente plus avec la complexité du problème
Evolutivité	La moindre modification du problème entraîne le renouvellement du montage.	Nécessite seulement une modification du programme
Coût	Faible (Pour un système simple)	Plus élevé

3.3.1. Les automates programmables industriels API :

Les API (en anglais : PLC programmable logic controller) sont apparues à la fin des années soixante, à la demande de l'industrie automobile américaine (GM). Il existe sur le marché de nombreuses marques d'automates : Siemens, Omron, Allen Bradley, Cegetel, Jetter, Shneider, etc.



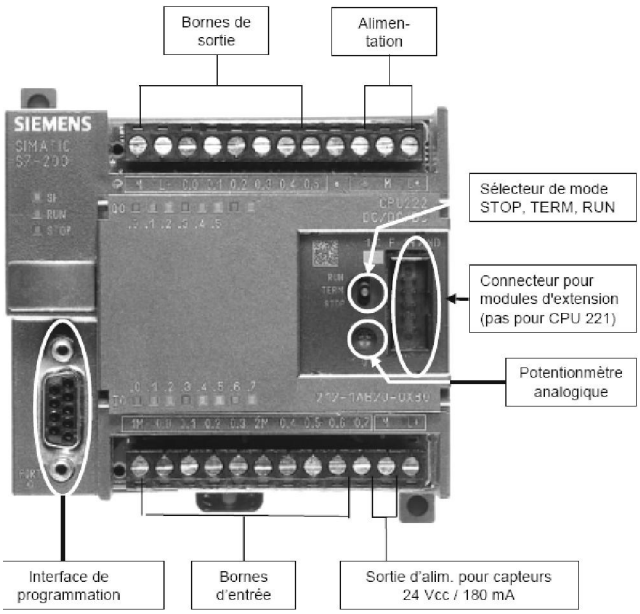
L'API est un appareil électronique programmable (par un automaticien, non informaticien) similaire à un ordinateur servant à commander des procédés industriels en élaborant des actions (pour les préactionneurs) selon un programme, à partir des informations fournies par les capteurs

Un API est constitué essentiellement des parties suivantes :

a- Le microprocesseur :: c'est le cerveau de l'automate, il réalise toutes les fonctions logiques, 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.

b- La mémoire : est conçue pour recevoir, gérer, stocker des informations issues des différents éléments du système : l'utilisateur (PC ou console), le microprocesseur, les capteurs. Deux types de mémoire cohabitent :

- La mémoire Programme (ROM : mémoire morte) : où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement.
- La mémoire de travail (RAM : mémoire vive) : utilisable en lecture-écriture pendant le fonctionnement Elle s'efface à l'arrêt de l'automate. Elle est répartie en différentes zones mémoires : Table image des entrées, Table image des sorties, Mémoire des bits internes, Mémoire programme d'application.



c- Les interfaces d'Entrées / Sorties:

- L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses.

- L'interface de sortie comporte de la même façon des adresses de sortie. Chaque préactionneur est relié à une de ces adresses. Le nombre de ces entrées est sorties varie suivant le type d'automate.

Les API peuvent être compacts ou modulaires :

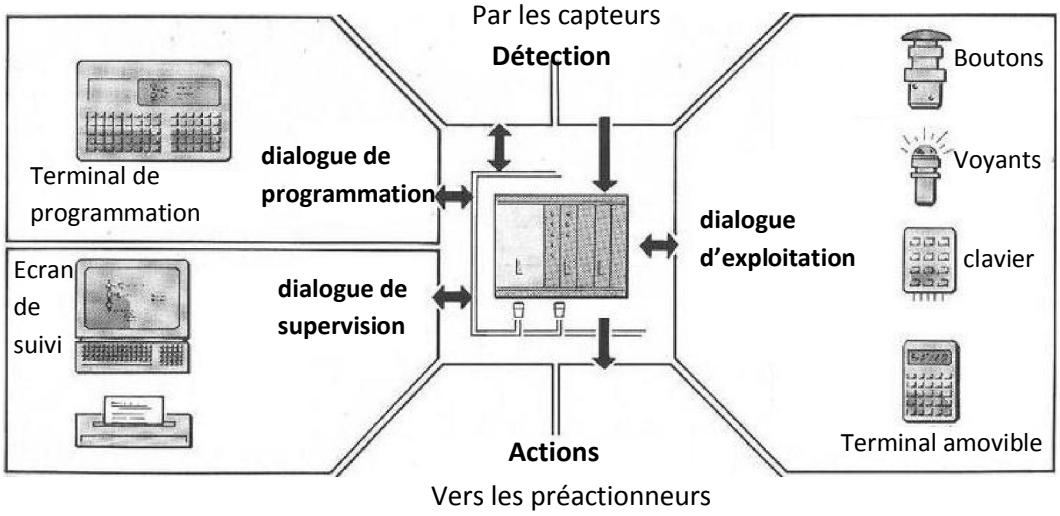
a- Le type compact (monobloc) possède un nombre d'entrées et de sorties restreint et son jeu d'instructions ne peut être augmenté. Ce type a pour fonction de résoudre des automatismes simples avec la logique séquentielle et utilisant des informations TOR.

b- Le type modulaire : est adaptable à toutes situations. Selon le besoin, des modules d'E/S analogiques sont disponibles en plus de modules spécialisés tels : PID, BASIC, Langage C,... La modularité permet un dépannage rapide et une plus grande flexibilité.

3.4. La partie dialogue :

Assure l'échange d'informations entre l'opérateur et le système (dialogue homme-machine). On distingue :

- ***Le dialogue de programmation :*** lors de la phase de développement et de mise au point du système ; Il consiste à : - Ecrire et interpréter l'ensemble des instructions du programme ;
  - Implanter le programme en mémoire.
- ***Le dialogue d'exploitation :*** A partir d'un terminal d'exploitation (clavier et écran) l'opérateur peut :
  -Lire sur un écran un message relatif à : l'état du système, à la nature du produit traité, à des mesures, à des défauts de fonctionnement.
  -Commander par l'intermédiaire d'un clavier l'évolution du système (sélection des modes de fonctionnement ; saisie de consignes ; émission d'ordres...)
- ***Le dialogue de supervision :*** assure la coordination avec les autres systèmes concernés.



3.5. Conception à l'aide de Grafcet :

Le GRAFCET (GRAphe Fonctionnel de Commande par Etapes et Transitions) est un outil graphique qui décrit l'évolution d'un automatisme et établit une correspondance à caractère séquentiel et combinatoire entre les entrées et les sorties. C'est un outil graphique puissant, directement exploitable en programmation.

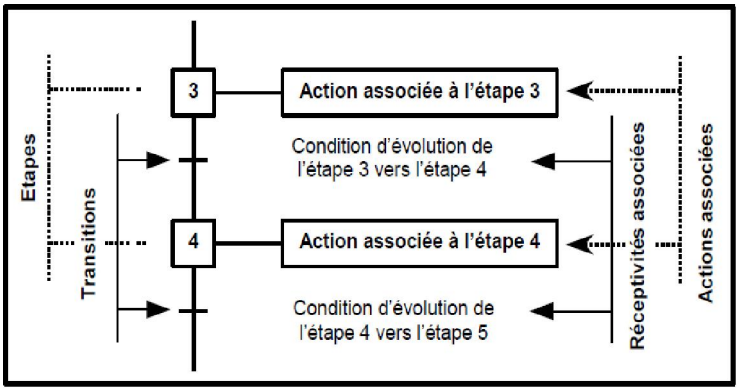
Un Grafcet est constitué de trois éléments :

-Etapes associées à des actions : Une étape caractérise une situation donnée. Elle peut être active ou inactive. Elle est représentée par un carré. L'étape initiale est représentée par un double carré.

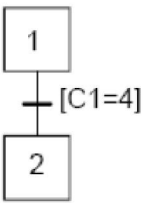
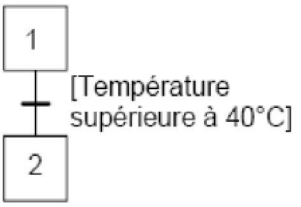
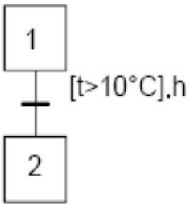
-Transitions associées à des réceptivités :

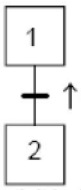
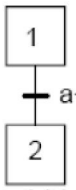
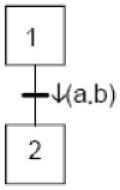
Une transition indique la condition d'évolution qui existe entre deux étapes. La réceptivité est une information d'entrée fournie par : l'opérateur, les capteurs, ou tout opération logique, arithmétique...

-Liaisons : traits verticaux orientées de haut vers le bas qui relient une étape à l'étape suivante.



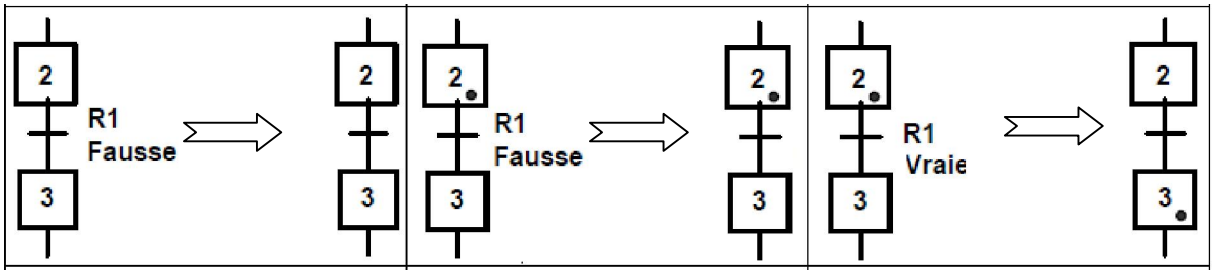
- **La réceptivité (conditions de transitions) :** porte une valeur logique (vrai ou faux). Par exemple :

		
La réceptivité est vraie lorsque la valeur courante du compteur est égale à 4.	Le langage littéral peut être utilisé.	La réceptivité est vraie lorsque la température est supérieure à 10°C et le niveau haut h est atteint.

 La réceptivité n'est vraie que lorsque a passe de l'état 0 à l'état 1	 La réceptivité n'est vraie que lorsque a est vraie ou que b passe de l'état 0 à l'état 1	 La réceptivité n'est vraie que lorsque le produit logique « a.b » passe l'état 1 à l'état 0

- **Franchissement d'une transition :**

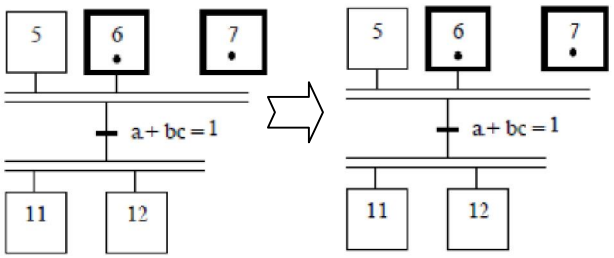
- A l'instant initial, seules les étapes initiales sont actives. (on indique l'activation par un jeton)
- Pour franchir une étape, il faut que toutes ses étapes amont (immédiatement précédentes) soient actives et que la réceptivité associée soit vraie.
- Le franchissement d'une transition entraîne obligatoirement l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.



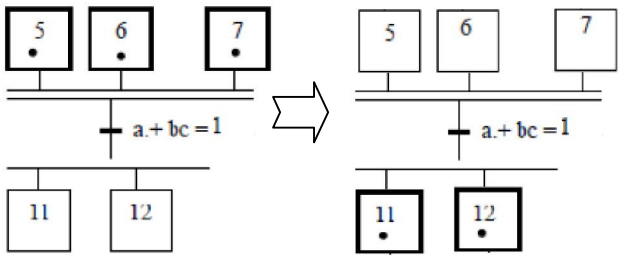
L'étape 2 n'est pas active et la réceptivité est fausse : pas de franchissement

L'étape 2 est active mais la réceptivité est fausse : pas de franchissement

L'étape 2 est active et la réceptivité est vraie : franchissement : 3 sera active et 2 désactivée

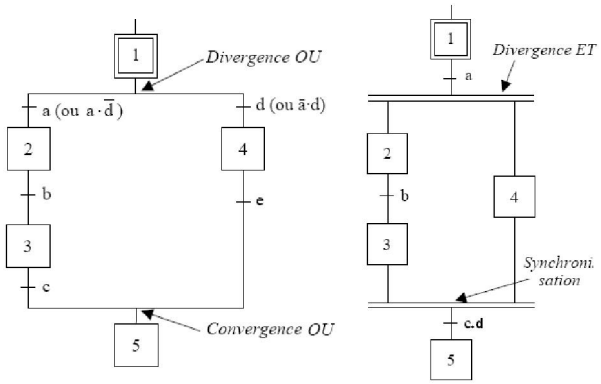


pas de franchissement



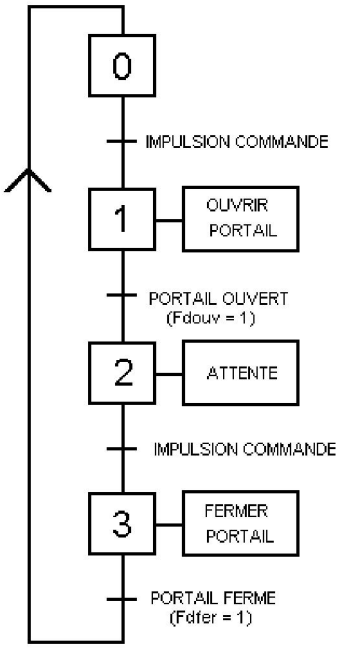
franchissement

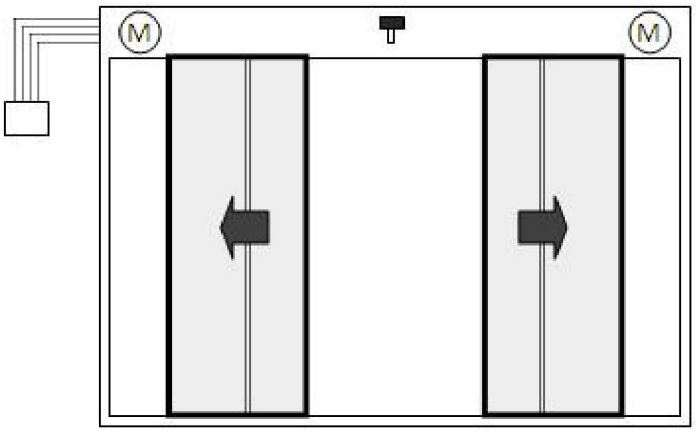
- On dit qu'il y a Aiguillage en OU lorsque le grafcet se décompose en deux ou plusieurs séquences selon un choix conditionnel.
- On dit qu'il y a un parallélisme ET, si plusieurs activités indépendantes pouvant se dérouler en parallèle.



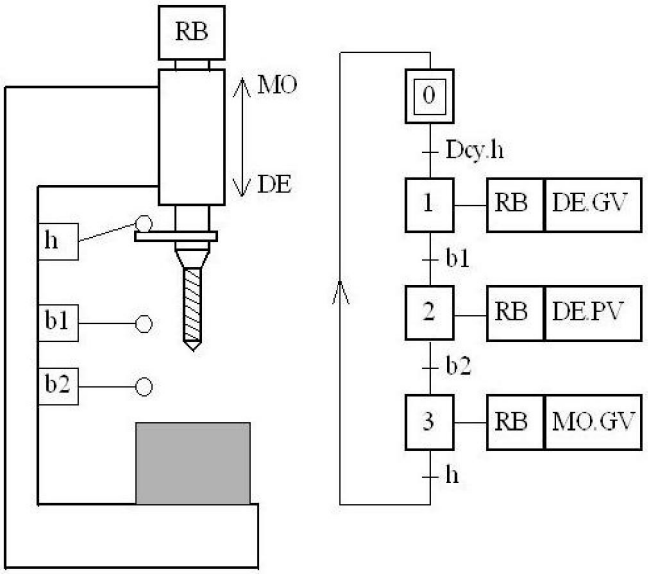
Exemples de Grafcet :

a. Ouverture d'une porte automatique :





b. Grafcet d'une fraise automatisée :



h, b1, b2 : capteurs de position.
Dcy.h : ordre de démarrage.
RB : actionneur.
DE GV : descente en grande vitesse
DE GV : descente en petite vitesse
MO GV : montée en grande vitesse

3.6. Langages de programmation des API :

Il existe 4 langages de programmation des automates qui sont normalisés par la norme CEI 61131-3. L'automate se programme via une console de programmation ou par un PC équipé d'un logiciel spécifique.

- ***Langage à contacts (LD : Ladder diagram) :*** Langage graphique utilise les symboles tels que : contacts, relais et blocs fonctionnels et s'organise en réseaux (labels). C'est le plus utilisé.
- ***Blocs Fonctionnels (FBD : Function Bloc Diagram) :*** Langage graphique où des fonctions sont représentées par des rectangles avec les entrées à gauche et les sorties à droites. Les blocs sont programmés (bibliothèque) ou programmables. Utilisé par les automaticiens.
- ***Liste d'instructions (IL : Instruction list) :*** Langage textuel de même nature que l'assembleur (programmation des microcontrôleurs). Très peu utilisé par les automaticiens.
- ***Langage littéral structuré (ST : Structured Text) :*** Langage informatique de même nature que le Pascal, il utilise les fonctions comme if ... then ...else ... Peu utilisé par les automaticiens.