CHAPITRE II : GENERALITE SUR L'AUTOMATISME (SAP, GRAFCET, API)

L'automatisme est devenu une technologie incontournable aujourd'hui de par son utilisation dans tous les domaines de fabrication. Il est donc important d'en connaître les bases et d'en suivre l'évolution.

Ce chapitre, dans sa structure, suit le cheminement de la conception d'un système automatisé.

II.1. SYSTEME AUTOMATISE DE PRODUCTION (SAP)

Un système de production est dit automatisé lorsqu'il peut gérer de manière autonome un cycle de travail préétabli qui se décompose en séquences et/ou en étapes.

Le système automatisé est un ensemble d'éléments en interaction, et organisés dans un but précis : agir sur une matière d'œuvre afin de lui donner une valeur ajoutée, en fonction des contraintes : énergétiques, de configuration, de réglage et d'exploitation qui interviennent dans tous les modes de marche et d'arrêt du système.

Dans le secteur industriel, il possède une structure de base identique, constitué de deux parties reliées entre elles :

- La partie opérative (PO)
- La partie commande (PC) ou système de contrôle/ commande (SCC)

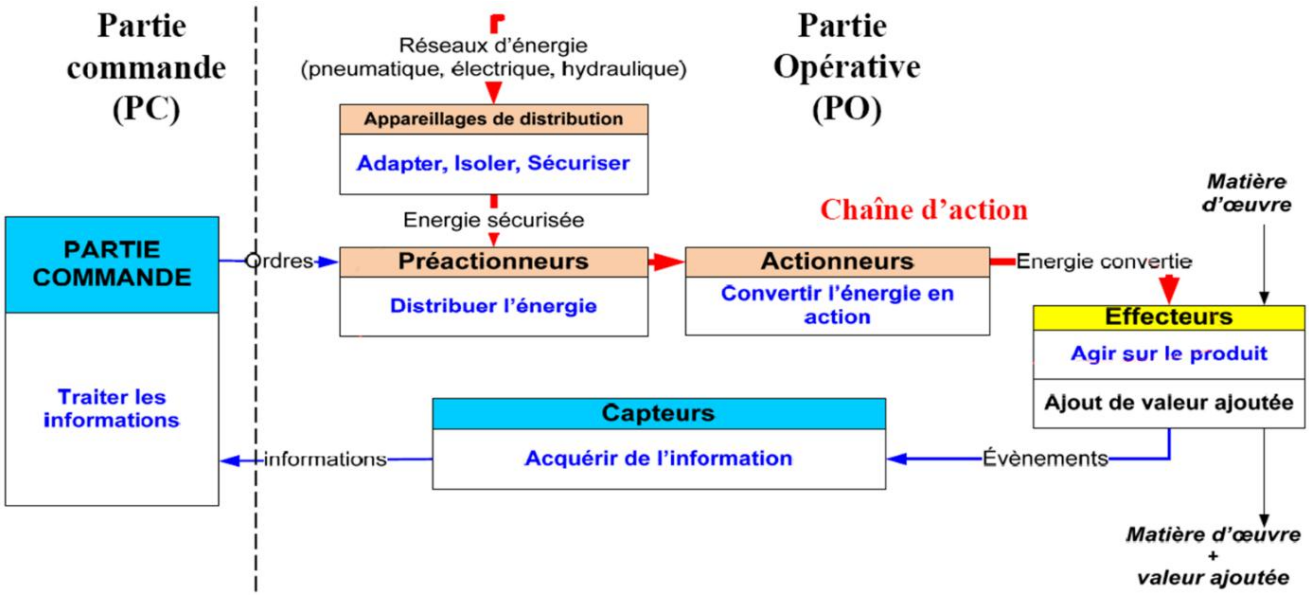


Figure 13: Structure d'un SAP

II.1.1. Description des différentes parties

II.1.1.1. La partie opérative

C'est la partie visible du système et elle agit sur la matière d'œuvre afin de lui donner sa valeur ajoutée :

- Les pré-actionneurs (distributeurs, contacteurs) qui reçoivent des ordres de la partie commande ;
- Les actionneurs (vérins, moteurs, vannes) qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique ;
- Les capteurs qui informent la partie commande de l'exécution du travail (contrôle, mesure, surveille et informe).

II.1.1.2. La partie commande

Ce secteur de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la partie opérative, et les restitue vers cette même partie opérative en directions des pré-actionneurs. L'outil de description de la partie commande s'appelle le GRAphe Fonctionnel de Commande Etape/ Transition (GRAFCET).

II.1.2. Les différents types de commande

Le traitement des données est géré par une logique câblée ou programmée.

II.1.2.1. Logique câblée

Le fonctionnement de l'installation de l'automatisme est défini par câblage (schéma électrique, tableau de connexion…)

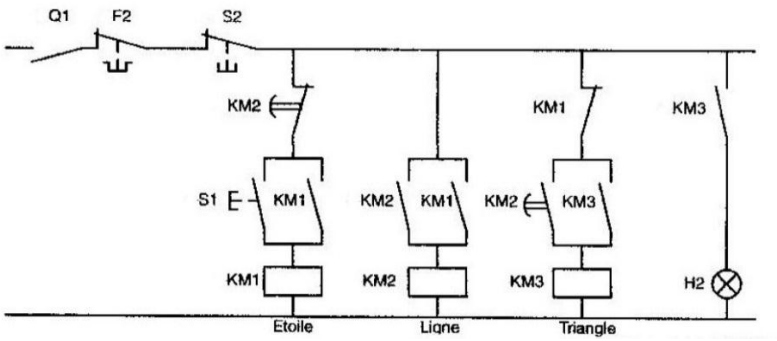


Figure 14: Présentation d'un logique câblée

Avantages :

- Automatisme simple et rapide à mettre en œuvre.
- Obligatoire pour le traitement d'arrêt d'urgence et de sécurité.

Inconvénients :

- Encombrement du câblage.
- Modifications du fonctionnement impose une modification de câblage.

II.1.2.2. Logique programmée

Le fonctionnement de l'installation de l'automatisme est défini par un programme (instructions).

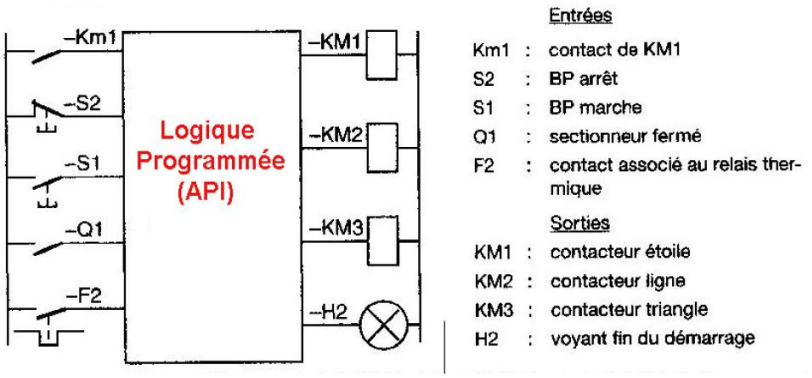


Figure 15: Présentation d'une logique programmée

Avantages :

- Câblage et volume réduits.
- Souplesse et adaptabilité de l'installation (erreurs, modifications, extensions : facile à réaliser).

Inconvénients :

- Plus cher.

II.1.3. Domaines d'application de la SAP

De notre temps, il serait difficile de concevoir un système de production sans avoir recours aux différentes technologies et composants qui forment les systèmes automatisés.

- Exemples de domaines d'application :
  - Le conditionnement, par exemple le déplacement d'objet suivant un angle quelconque ou le conditionnement sur palette après emballage.
  - L'industrie automobile avec l'utilisation des robots industriels pour effectuer l'assemblage et la peinture des carrosseries.
  - L'industrie du bois avec les opérations de débit, de sciage et d'usinage du bois.
  - Machine-outil : l'automatisation est ici assez importante. L'une des principales applications est dans les unités de perçage.
  - Contrôle de produits : Détection de défauts en bout de chaine de production.
  - Automatisation de services : ouvertures programmées de portes et fenêtres, gestion centralisée de bâtiment.
  - Mise en forme de produits finis : poinçonnage de tôles et leur mise en forme, travail dans des conditions extrêmes et environnement difficile.
- Les avantages :
  - La capacité de production accélérée ;
  - L'aptitude à convenir à tous les milieux de production ;
  - La souplesse d'utilisation ;
  - La création des postes d'automaticiens.
- Les inconvénients :
  - Le coût élevé du matériel, principalement avec les systèmes hydrauliques ;
  - La maintenance doit être structurée.

II.2. LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API)

L'automate programmable est un système de traitement logique d'informations dont le programme de fonctionnement est effectué à partir d'instructions établies en fonction du processus à réaliser.

Fonction de l'automate :

- A partir des informations que lui fournissent les capteurs et, suivant un algorithme déterminé par programmation, élabore les commandes transmises aux actionneurs.
- Assure la communication avec l'opérateur (interface avec l'usager) et les autres processeurs qui gèrent la production ou qui interviennent dans le même procédé.

II.2.1. Présentation d'un API

II.2.1.1. Architecture externe d'un API

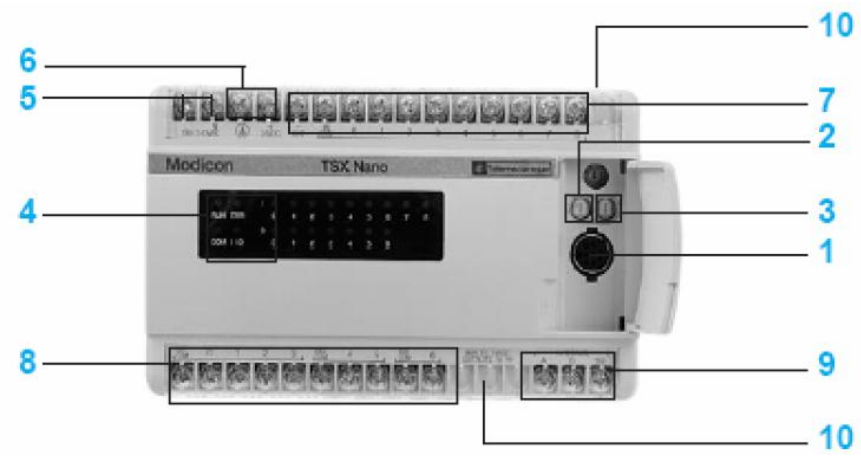


Figure 16: Architecture externe d'un API

Les automates comprennent en général :

- ***1*** : Prise pour raccordement du terminal permettant la programmation à partir d'une console ou d'un ordinateur.
- ***2 et 3*** : Prises pour des fonctions de communication spécialisées avec d'autre équipement.
- ***4*** : Ecran pour visualiser l'état des entrées/sorties et celle de l'automate.
- ***5*** : Raccordements de l'alimentation secteur.
- ***6*** : Alimentation des capteurs (généralement $24VDC/150mA$).
- ***7*** : Raccordement des capteurs d'entrées.
- ***8*** : Raccordement des pré-actionneurs de sorties.
- ***9*** : Raccordement pour entrées spécifiques.
- ***10*** : Cache amovible pour protection des borniers à vis.

II.2.1.2. Architecture interne d'un API

L'automate programmable reçoit les informations relatives à l'état du système et commande les pré-actionneurs suivant le programme inscrit dans sa mémoire. Un API se compose donc de cinq grandes parties : l'alimentation, le processeur, la zone mémoire, les interfaces Entrées/Sorties et le bus interne.

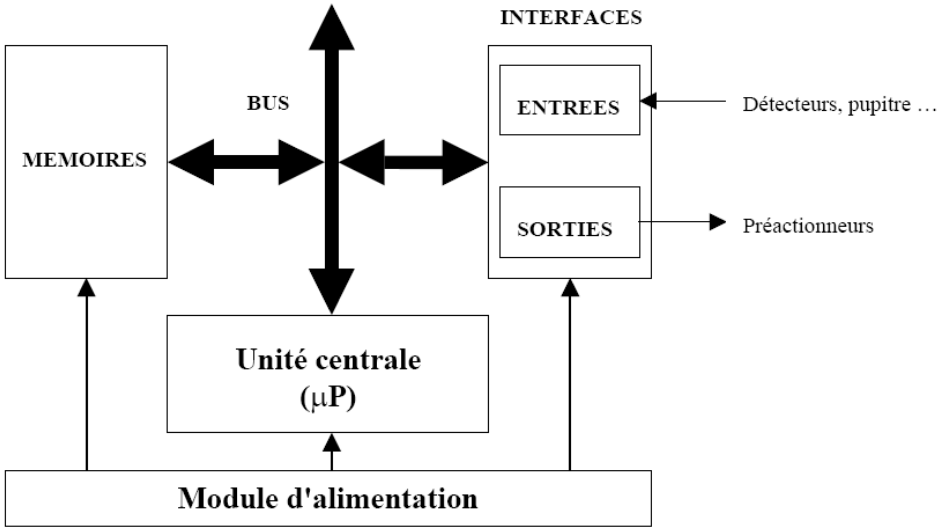


Figure 17: Structure interne d'un API

- **L'alimentation** : Il assure la distribution d'énergie aux différents modules (alimenté en $240V_{ac}$ et délivre une tension de $24V_{cc}$).
- **L'unité centrale à base de microprocesseur** : il réalise toutes les fonctions logiques, arithmétiques, les fonctions de temporisation, de comptage, de calcul…à partir d'un programme contenu dans sa mémoire.
- **La mémoire** : 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.
    - De recevoir et de conserver le programme du processus (cette conception et élaboration du programme font appel à la RAM et l'EEPROM).

Les actions possibles sur la mémoire sont : **ECRIRE** (pour modifier le contenu d'un programme), **EFFACER** (pour supprimer les informations qui ne sont plus nécessaires), **LIRE** (pour lire le contenu d'un programme sans le modifier).

- **Le bus interne :** Il permet la communication de l'ensemble des blocs de l'automate et des éventuelles extensions.
- **Les interfaces d'entrées/sorties** :

Les nombres d'entrée et sortie varient suivant le type d'automate. Ils possèdent une modularité de 8,16 ou 32 voies et délivrent des tension continues $0 - 24V_{cc}$.

Interface d'entrée	*Interface de sortie*
Les entrées reçoivent des informations en provenance des éléments de détection (capteurs) et des pupitres de commande (Bouton poussoir).	Les sorties transmettent des informations aux pré-actionneurs (relais, électrovanne…) et aux éléments de signalisation (voyants) du pupitre.

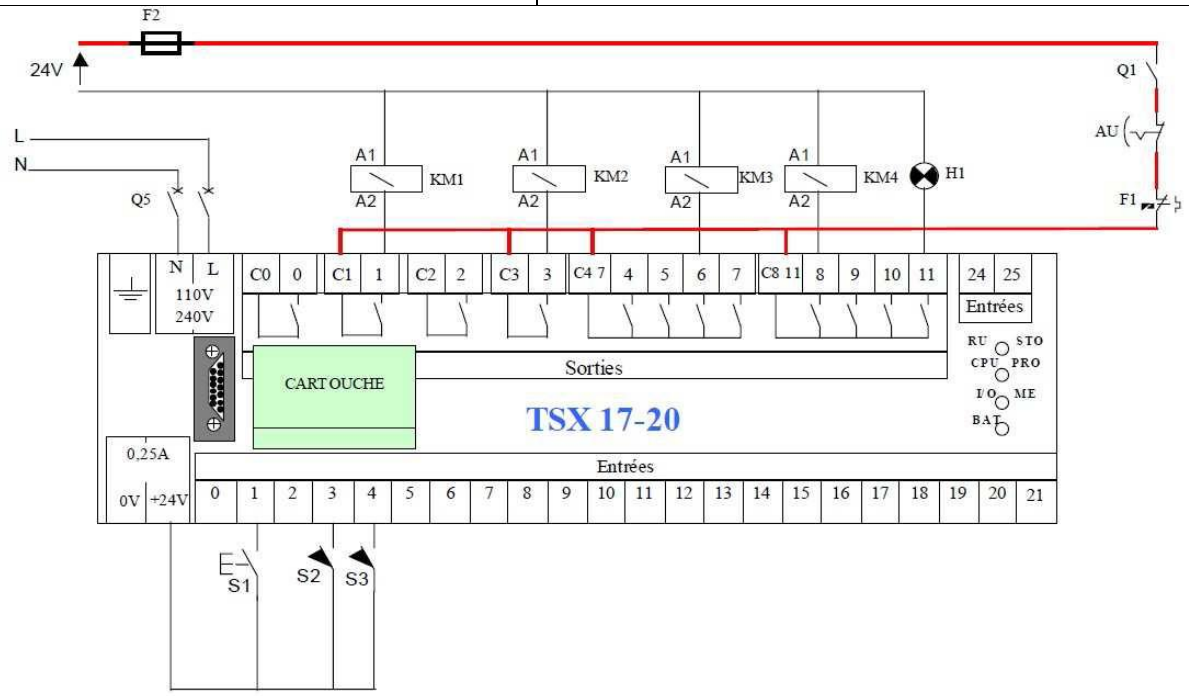


Figure 18: Les interfaces d'entrées et sorties

II.2.2. Programmation d'un API

Elle peut s'effectuer de trois manières différentes :

- Sur l'API lui-même à l'aide de touches.
- Avec une console de programmation reliée par un câble spécifique à l'API.
- Avec un PC et un logiciel approprié.

II.2.2.1. Langages de programmation pour API

Chaque automate possède son propre langage. Mais les constructeurs proposent tous une interface logicielle répondant à la norme CEI 1131-3 qui définit les cinq langages de programmation utilisables, tels que :

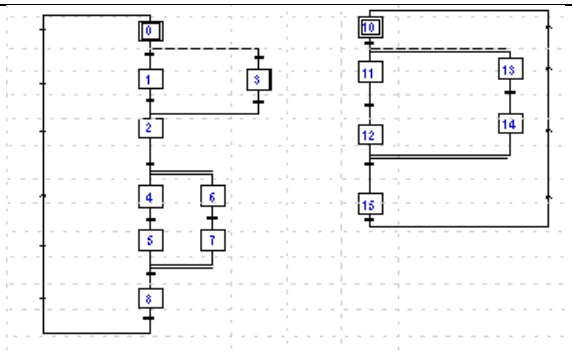
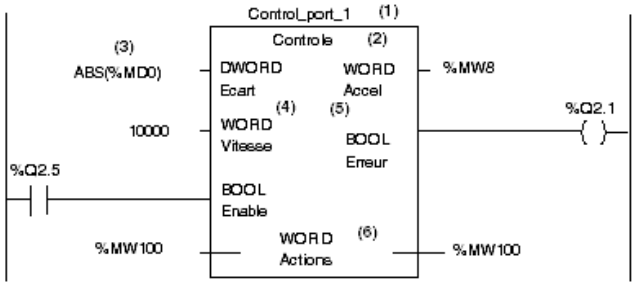
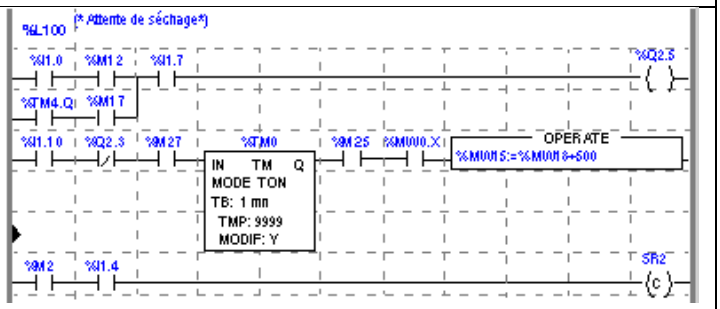
GRAFCET ou SFC (Sequentiel Function Chart) : ce langage de programmation de haut niveau permet la programmation aisée de tous les procédés séquentiels.	
FBD (Function Block Diagram) ou Schéma par Bloc : ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Manipulation de tous les variables.	
LD (Ladder Diagram) ou Schéma à contacts : ce langage graphique se compose d'une suite de réseaux de contacts composés d'un ensemble d'élément graphique disposé sur grille organisée en lignes et colonnes.	
ST (Structured Text) ou texte structuré : ce langage permet de programmer tout type d'algorithme plus ou moins complexe.	**(* Mise à jour du voyant cycle en cours *)** **IF %M0 THEN** **SET %M18;** **ELSE RESET %M18;** **END_IF;** **(* RAZ application *)**
IL (Instruction List) ou liste d'instructions : ce langage de textuel est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.	! %L0 : LD %I1.0 ANDN %M12 OR (%TM4.Q AND %M17) AND %I1.7 ST %Q2.5 ! %L5 : LD %I1.10 ANDN %Q2.3 ANDN %M27 IN %TM0 LD %TM0.Q AND %M25 AND %MW0:X5 [%MW15 := %MW18+500]

Tableau 6: Les différents langages de programmation pour API

II.2.2.2. Eléments graphiques du langage LADDER

Nous utiliserons le programme LADDER afin de programmer notre système de production dans la suite de notre étude. C'est le langage le plus utilisé par les automaticiens dans le domaine de l'automatisation vue que c'est un langage standard au API.

- Les éléments graphiques de base du langage LADDER

Le langage à contact est adapté à la programmation de traitements logiques, il utilise le schéma développé. Nous retrouvons :

- La fonction ET en utilisant des contacts en série.
- La fonction OU en utilisant des contacts en parallèle.

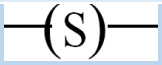
Désignation		Fonction	Symboles
Eléments de test	Contact à fermeture	Contact passant quand il est actionné.	
	Contact à ouverture	Contact passant quand il n'est pas actionné.	
Eléments de liaison	Connexion horizontale	Permet de relier les éléments action série.	
	Connexion verticale	Permet de relier les éléments action en parallèle.	
Eléments d'action	Bobine directe	La sortie prend la valeur du résultat logique.	
	Bobine inverse	La sortie prend la valeur inverse du résultat logique.	
	Bobine d'enclenchement	Le bit interne est mis à 1 et garde cet état.	
	Bobine de déclenchement	Le bit interne est mis à 0 et garde cet état.	
	Arrêt programme	Terminaison du programme	<END>

Tableau 7: Les éléments graphiques de base

- Structure d'un réseau de contacts :

Un réseau de contacts se compose de la manière suivante : étiquette (ou titre) et zone graphique (zone test + zone action).

La zone de test accueille :

- Les contacts
- Les blocs fonction (temporisateurs, compteur…)
- Les blocs comparaison.

La zone action accueille :

- Les bobines
- Les blocs opérations.

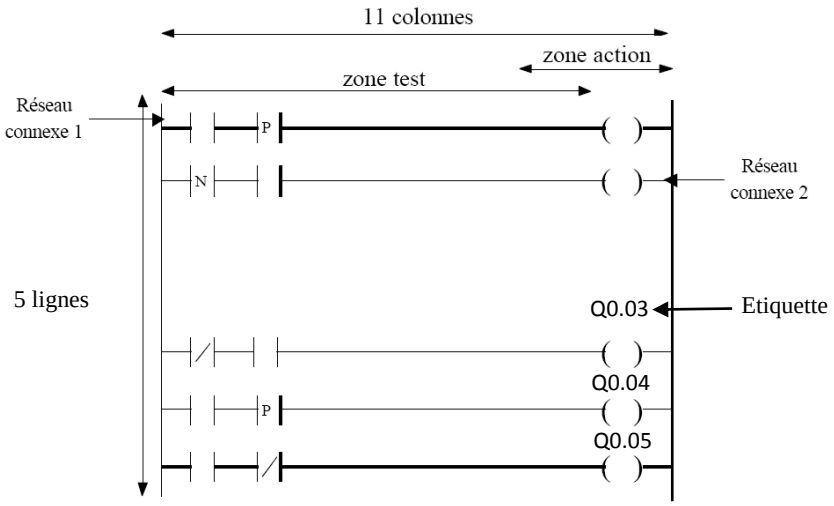


Figure 19: Structure d'un réseau de contacts

➢ <u>Règles d'évolution d'un réseau de contacts</u> :

La lecture d'un réseau se fait réseau connexe par réseau connexe (de haut en bas), puis de gauche à droite à l'intérieur d'un réseau connexe qui est constitué d'éléments graphiques tous reliés entre eux, mais indépendants des autres éléments graphiques du réseau.

A la rencontre d'une liaison verticale de convergence, on évalue d'abord le sous-réseau qui lui est associé avant de continuer l'évaluation du sous-réseau qui l'englobe.

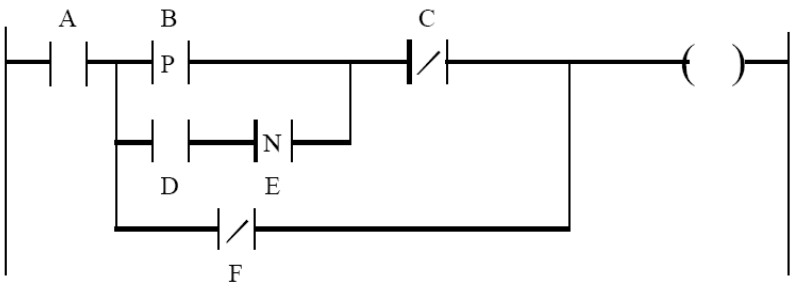


Figure 20 : Exemple d'un réseau connexe

L'ordre d'exécution des éléments de ce réseau est le suivant :

- 1ère phase : lecture des contacts A et B jusqu'à la rencontre de la 1ère liaison verticale de convergence entre les contacts B et C.
- 2ème phase : lecture du premier sous réseau, contact D et E.
- 3ème phase : reprise de lecture de la première ligne du réseau connexe, contact E, jusqu'à la rencontre de la deuxième liaison verticale de convergence.
- 4ème phase : lecture du deuxième sous-réseau, contact F.
- 5ème phase : lecture de la bobine.

II.3. NOTION DE GRAFCET

II.3.1. Définition

Le GRAFCET est un outil graphique permettant de décrire le fonctionnement et le comportement des systèmes de commandes en établissant une représentation graphique indépendante de la réalisation technologique et qui est utilisé pour programmer l'API. Il comprend :

- Des étapes associées à des actions ;
- Des transitions associées à des réceptivités ;
- Des liaisons orientées reliant étapes et transitions.

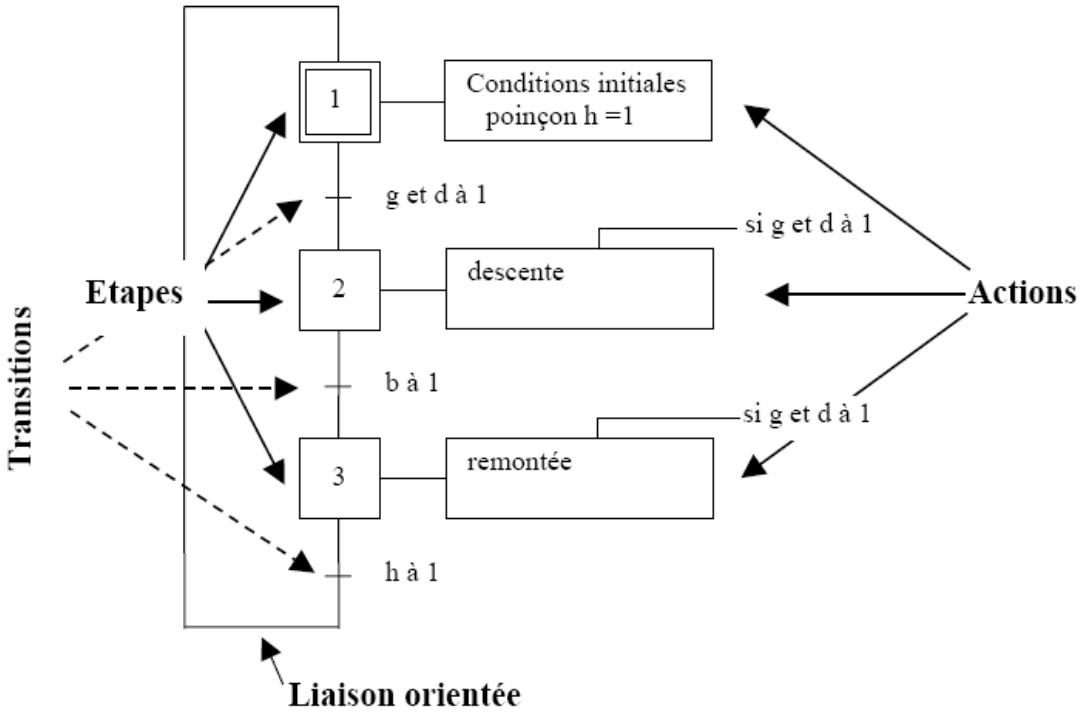
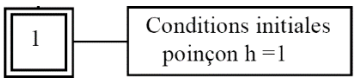


Figure 21: Les éléments constitutifs d'un GRAFCET

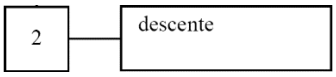
II.3.2. Description du GRAFCET

- Les étapes :

Il symbolise un état ou une partie de l'état du système. L'étape possède deux états possibles : active ou inactive.

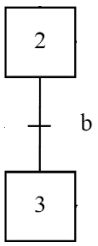


Etape initiale qui représente le système à l'état de repos initial. Elle est activée au début du cycle.



Etape : A chaque étape est associée une action ou plusieurs actions, c'est-à-dire un ordre vers la partie opérative ou vers d'autres grafcets.

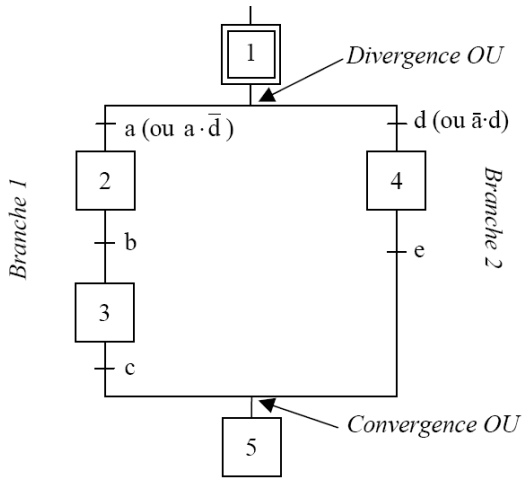
- Les transitions :



Transition : Elle indique la possibilité d'évolution qui existe entre deux étapes, donc la succession de deux actions dans une séquence. A chaque transition est associée une réceptivité qui exprime la condition nécessaire pour le passage d'une étape à une autre.

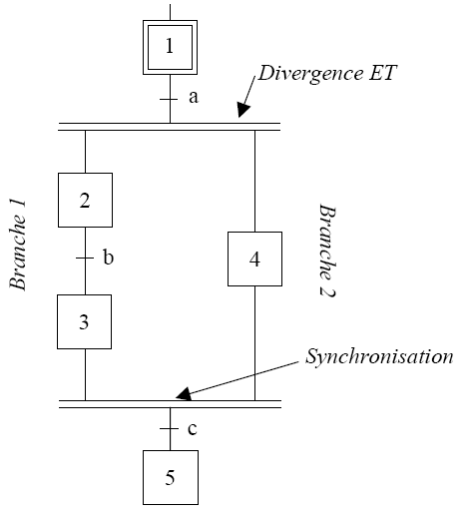
- Les aiguillages :

Un GRAFCET peut être constitué de plusieurs séquences, il est souvent nécessaire d'effectuer une sélection exclusive d'une parmi ces séquences afin de n'exécuter qu'une seule à la fois. Pour cela, les réceptivités sur les deux branches doivent être des conditions exclusives.



➢ Les parallélismes :

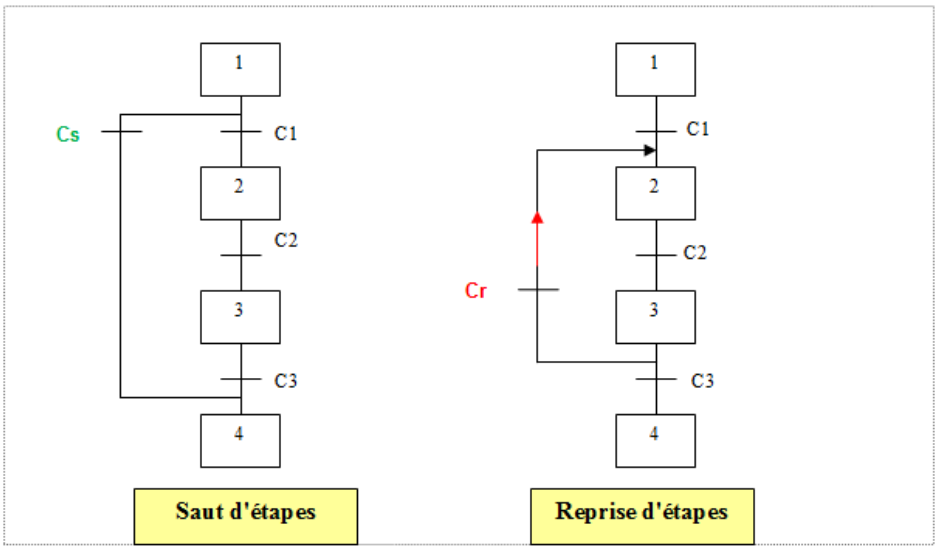
Au contraire de l'aiguillage où ne peut se dérouler qu'une seule séquence à la fois, il peut y avoir plusieurs séquences d'activités pouvant se dérouler en parallèle ou simultané.



Après le franchissement de « a », nous obtenons deux activités simultanées ou parallèles. La synchronisation permet d'attendre la fin de plusieurs activités se déroulant en parallèle, pour continuer par une seule.

➢ Le saut d'étape et le reprise de séquence :

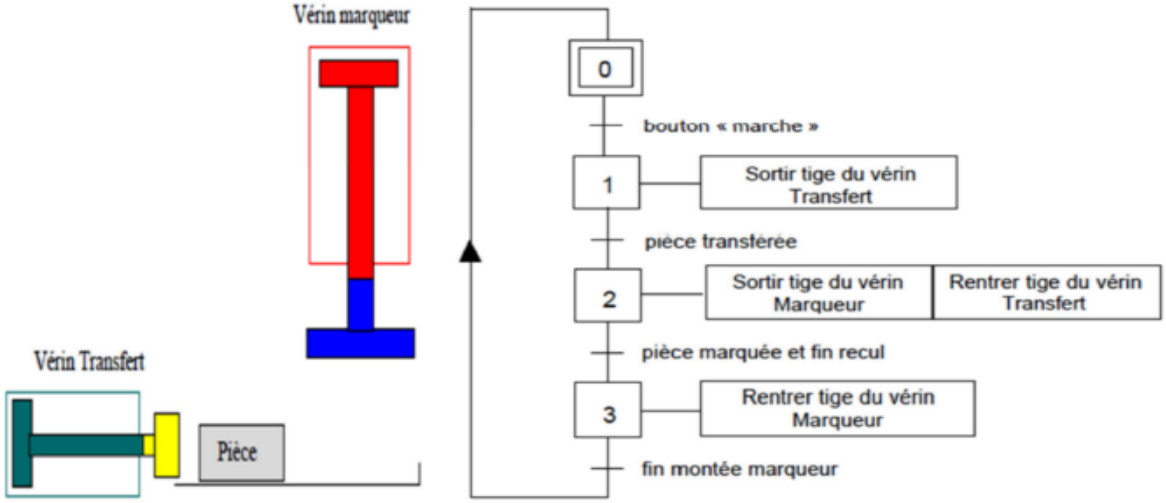
Le saut d'étape est un aiguillage particulier permettant de sauter une ou plusieurs étapes lorsque les actions à réaliser deviennent inutiles, tandis que la reprise de séquence permet au contraire de reprendre une ou plusieurs fois la même séquence tant qu'une condition fixée n'est pas obtenue.



II.3.3. Les niveaux de GRAFCET

- GRAFCET de niveau 1 : Spécifications fonctionnelles

C'est la description du comportement de la partie commande en fonction de l'évolution de la partie opérative, les contraintes technologiques (des actionneurs, des capteurs) n'interviennent pas, seules comptent les fonctionnalités.

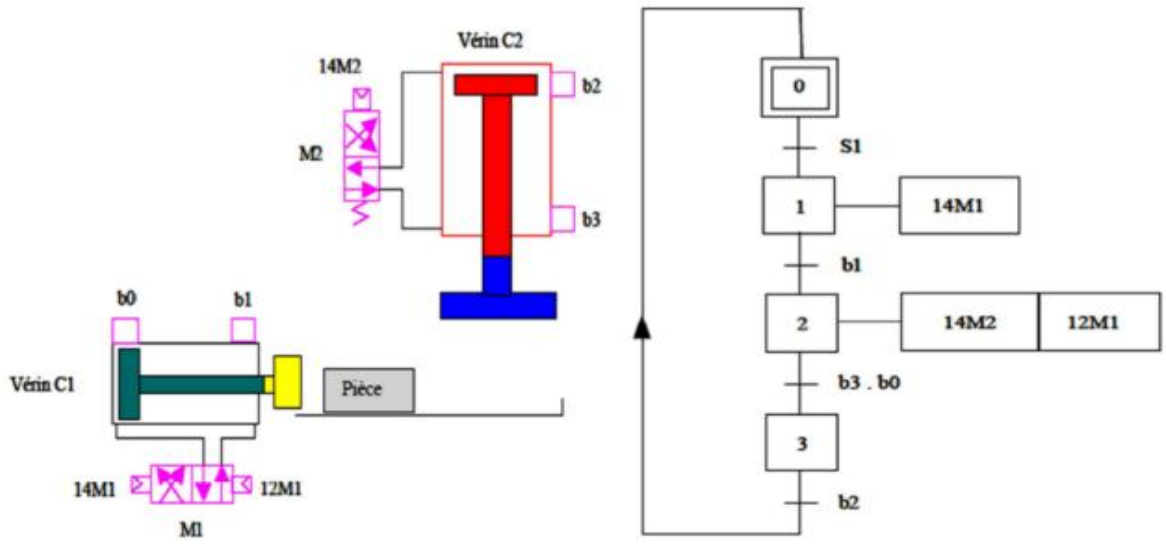


Exemple de GRAFCET niveau 1

- GRAFCET de niveau 2 : Spécifications technologiques et opérationnelles

Il précise l'ensemble des échanges de la partie commande avec la partie opérative et le dialogue avec l'opérateur en prenant en compte les choix technologiques.

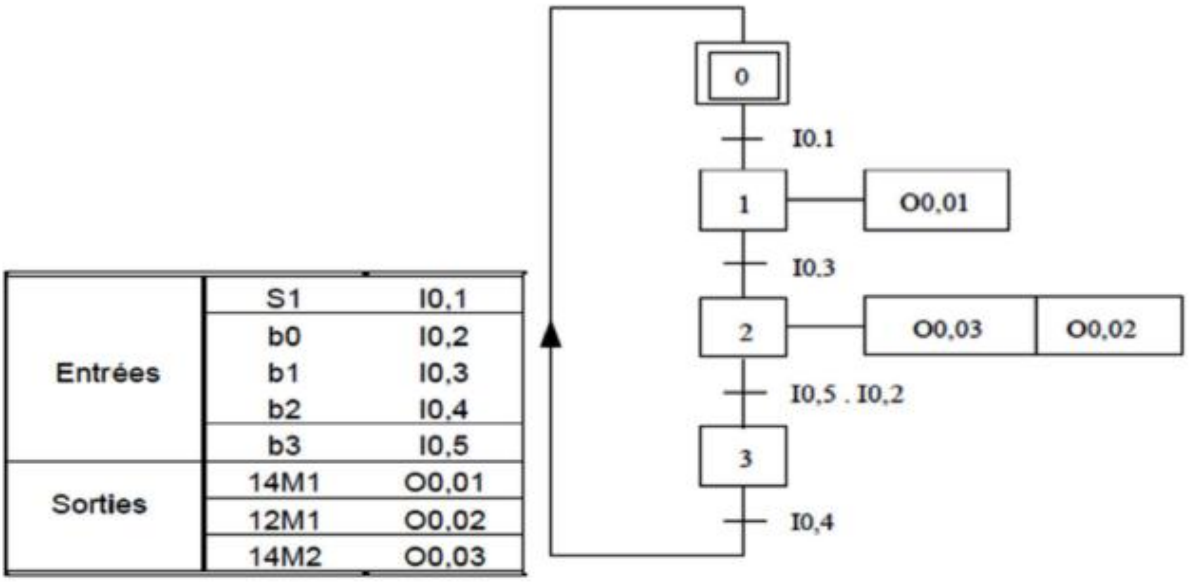
Point de vue partie commande :



Exemple de GRAFCET niveau 2 : point de vue commande

Point de vue automate :

Il possède la même structure que le GRAFCET point de vue commande mais il précise les entrées/sorties de l'automate programmable en liaison avec système.



Entrées	S1	I0,1
	b0	I0,2
	b1	I0,3
	b2	I0,4
	b3	I0,5
Sorties	14M1	O0,01
	12M1	O0,02
	14M2	O0,03

Exemple de GRAFCET niveau 2 : point de vue automate