



Niveau d'étude:

3^{ème} année Licence en
(Génie électriques)

Support de cours

Automatismes industriels



A.Abderrahim

2020

Chapitre 2:

Le Grafcet

* Objectif : l'étudiant sera capable de traduire les cahiers des charges d'automatisation industrielle en langage Grafcet.

* Moyen utilisé: logiciel Automgen..

2.1. Historique

Un groupe d'universitaires et industriels de la section "*Systèmes Logiques*" de l'**AFCET** (**A**ssociation **F**rançaise de **C**ybernétique **E**conomique et **T**echnique) s'est penché , en 1975, pour inventer une représentation des évolutions séquentielles d'un système automatisé quelconque d'avoir les caractéristiques suivantes :

- Simple;
- Accepté par tous;
- Compréhensible à la fois par les concepteurs et les exploitants;
- Avoir la facilité de passage à une réalisation, à base matérielle et/ou logicielle de l'automatisme.

Au début de leur travail et de leur recherche , les universitaires et les industriels de la section ont recensé trois grandes classes d'outils de modélisation qu'il a fallu mettre en évidence telles que :

- les organigrammes ;
- Les Réseaux de Pétri;
- Les graphes d'état.

Leur analyse profonde les mena, en 1977, à la définition du **GRAFCET** .

2.2. Domaine d'application du GRAFCET

Le GRAFCET (**G**raphe **F**onctionnel de **C**ommande **E**tape **T**ransition) est un outil méthode, descriptif du cahier des charges de tout système séquentiel. Il permet de faire la description des comportements de l'automatisme au niveau du traitement des informations de la partie opérative et les ordres transmis à cette même partie.

2.3. Différents points de vue d'un GRAFCET

Selon l'observateur s'implique au fonctionnement de ce système, la représentation d'un système automatisé par un grafcet prend en compte **trois "points de vue"** différents:

- GRAFCET du point de **vue système** ;
- GRAFCET du point de **vue partie opérative** ;
- GRAFCET du point de **vue partie commande**.

a. GRAFCET du point de vue système :

C'est un graphe qui décrit le **fonctionnement global du système**. Il est fidèle au **cahier des charges** sans avoir des connaissances de la technologie appliquée . Il offre un dialogue avec des personnes non qualifiées ou spécialisées ; sa langage ou son écriture est compréhensible par tout le monde.

b . GRAFCET du point de vue partie opérative :

Dans ce cas, on va **spécifier la technologie de la partie opérative** aussi le type de ses informations envoyées (comptes-rendus) et reçues (ordres). L'observateur ici est considéré comme un spécialiste seulement de la partie opérative, sans avoir besoin de connaissances de la partie commande.

c. GRAFCET du point de vue partie commande :

Dans ce cas, on va **spécifier la technologie de la partie commande** ainsi la technologie des éléments de dialogue :

- entre PC et PO ;
- entre PC et opérateur ;
- entre PC et autre système.

2.4 Eléments de Base

- La description du fonctionnement d'un automate peut alors être représentée graphiquement par un ensemble :
- - d'**ETAPES** auxquelles sont associées des **ACTIONS**,
- - de **TRANSITIONS** auxquelles sont associées des **RECEPTIVITES**,
- - de **LIAISONS** (ou **ARCS**) **ORIENTEES**,

2.4.1.Étapes et actions associées

a. Étape

Une étape est symbolisée par un carré repéré par un numéro pour la distinguer aux autres étapes . Une étape correspond à un état du système automatisé dans lequel le comportement de la partie commande est stable vis à vis de ses entrées et ses sorties. Une étape peut être active ou inactive à un instant "t" donné. On repère l'activation de cette étape par un point plein placé sous le chiffre.



L'étape initiale est désignée par un carré double ainsi que toute étape initialement activée.

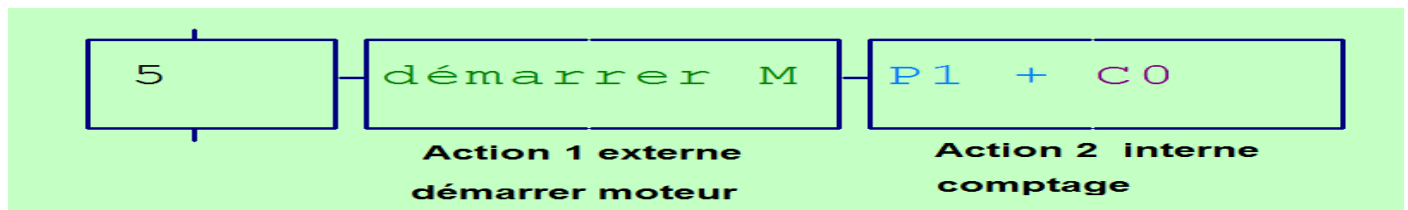
b. Action

Une ou plusieurs actions peuvent être raccordées à une étape. Les actions jointes à une étape quelconque traduisent ce qui doit être effectué si l'étape est active.

Les actions qui sont les résultats aboutis du traitement logique des informations grâce à la partie commande qui peuvent être :

- **externes** : ordres émis vers la partie opérative ou vers les éléments externes.
- **internes** : fonctions spécifiques telles que : temporisation, comptage, etc ..

Les actions sont repérées par un ou plusieurs rectangles :



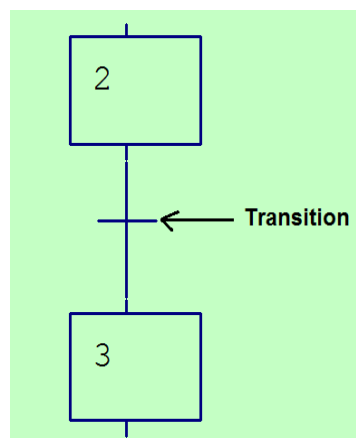
2.4.2. Transitions et réceptivités associées

a. transition

Une transition indique la **possibilité du passage** d'une étape à l'étape suivante. La transition d'une à l'autre se fait par le franchissement d'une transition.

Une transition est dite validée lorsque toutes les étapes immédiatement précédentes raccordées à cette transition sont achevées.

Une transition entre deux étapes est repérée par une barre perpendiculaire.



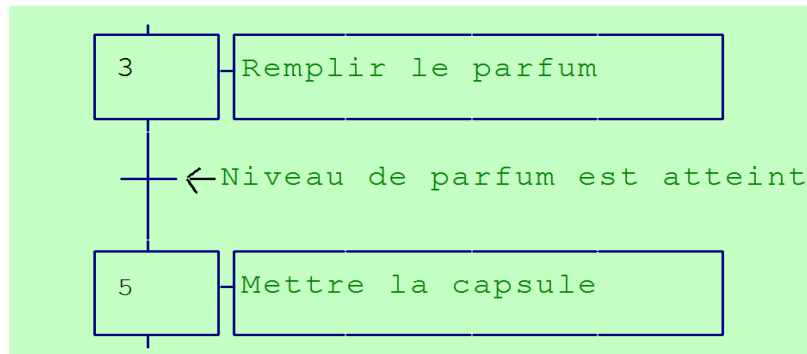
b. réceptivité

A chaque transition est associée une **proposition logique** appelée réceptivité qui peut être soit vraie soit fausse.

La réceptivité écrite sous forme de proposition logique est une fonction combinatoire:

- d'**informations externes** qui proviennent de la partie opérative ou du poste opérateur,
- d'**informations internes** qui ont relations avec les fonctions spécifiques (temporisation, comptage), ou qui dépendent des états actifs ou inactifs des autres étapes.

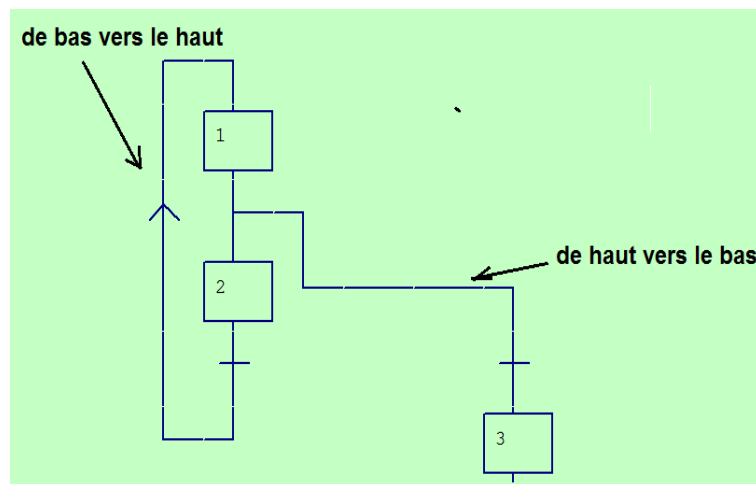
Exemple :



2.4.3. Liaisons orientées

Les liaisons orientées permettent de relier les étapes aux transitions et inversement. Le sens général du parcours est du haut vers le bas. Lorsque le parcours est différent des flèches sont nécessaires.

Exemple :



2.5. Règles d'évolution du GRAFCET

Un grafcet possède un comportement dynamique dirigé par cinq règles, elles précisent les causes et les effets du franchissement des transitions.

- **Règle N°1**

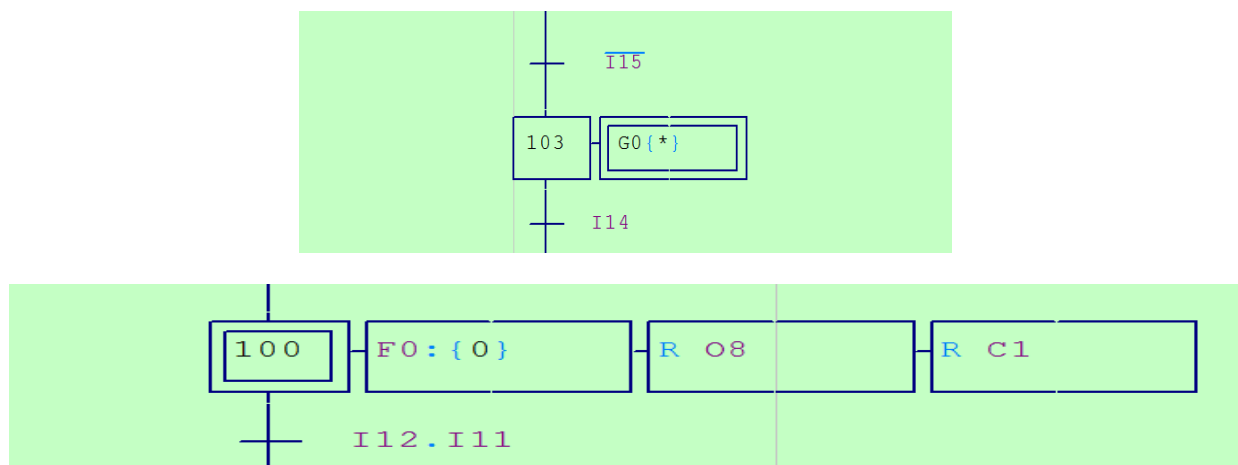
La situation initiale d'un grafcet caractérise le comportement initial de la partie commande vis-à-vis de la partie opérative, de l'opérateur et/ou des éléments extérieurs. Elle correspond aux étapes actives au début du fonctionnement : ces étapes sont les étapes initiales. Un ordre de forçage d'un grafcet partiel est présenté dans un rectangle associé à une étape.

Quelques exemples de situations du grafcet partiel G0 provoqué par le figeage, le forçage et la réinitialisation du comptage:

G0 {*} situation ou le grafcet G0 est " figé" (maintien dans l'état)

F0 {0} situation ou le grafcet G0 est forcé à l'état initial 0

R C1 réinitialisation du compteur C1



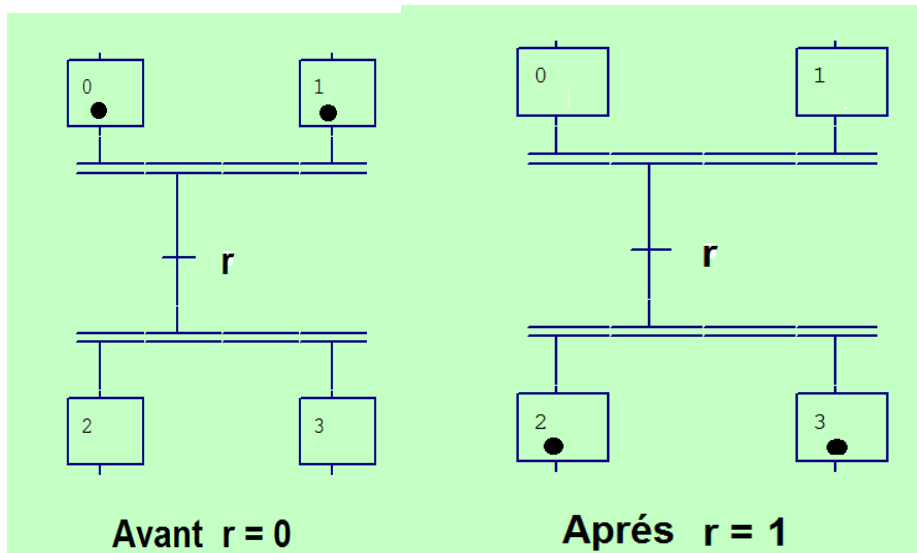
- **Règle N°2**

Une transition est validée lorsque toutes les étapes immédiatement précédentes sont actives.

Elle ne peut être franchie que :Lorsqu'elle est validée, et que la réceptivité associée à la transition est vraie

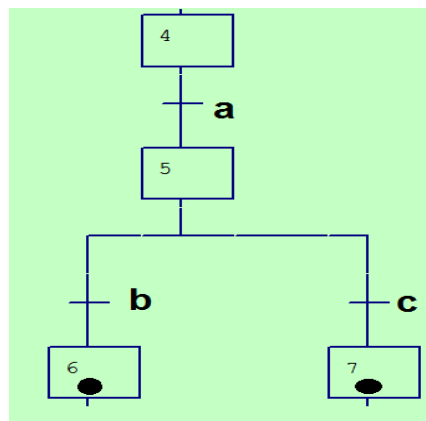
- **Règle N°3**

Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.



- Règle N°4

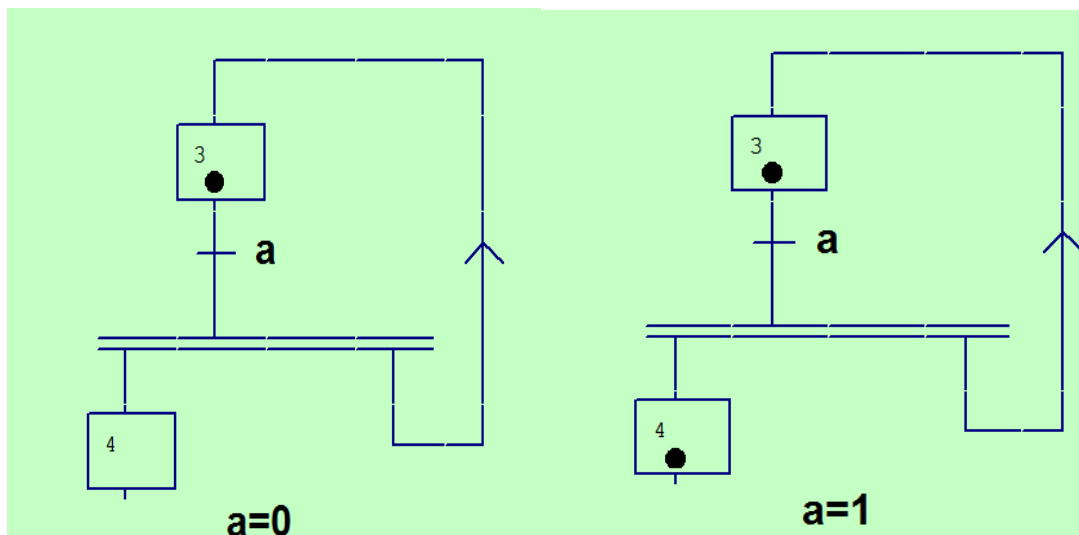
Evolution simultanée Plusieurs transitions simultanément franchissables sont simultanément franchies



Règle N°5

Activation et désactivation simultanée d'une étape

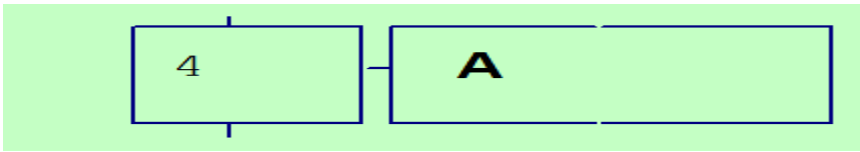
Si au cours du fonctionnement la même étape est simultanément activée et désactivée elle reste active .



2.6. Les actions associées aux étapes

2.6.1. Actions continues

L'action dure tant que l'étape est active:

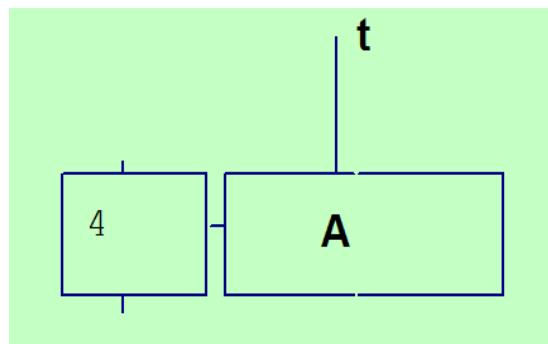


2.6.2. Action conditionnelles

L'action sera effective si l'étape est active et si la condition associée à l'étape est vérifiée.

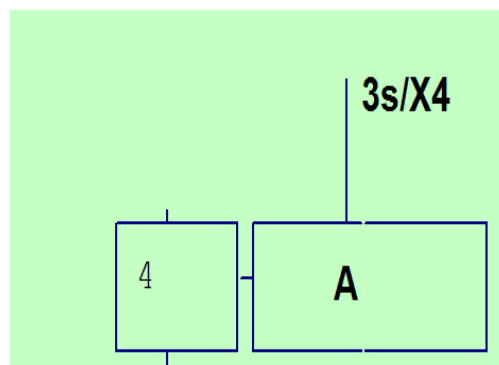
Le chauffage sera activé pendant la durée de l'étape 4 tant que t sera égale à 1:

$$X4=A.t$$



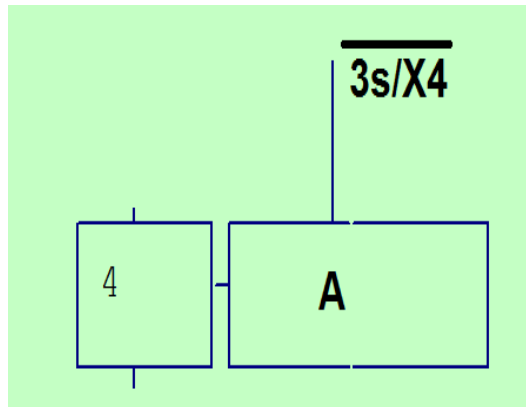
2.6.3. Actions retardées

Après l'activation de X4 l'action A sera retardée de 3 s:



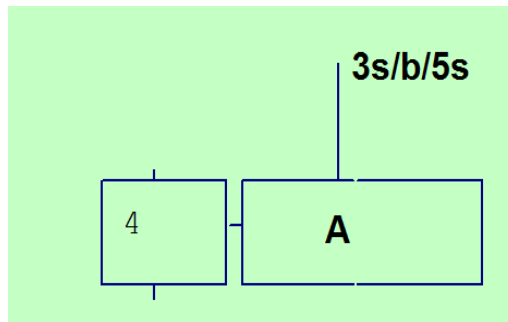
2.6.4. Actions limitées dans le temps

Après l'activation de X4 l'action A sera limitée seulement de 3 s:



2.6.5. Action retardées et limitées dans le temps

Après l'activation de X4 l'action A sera retardée de 3 s et sera limitée seulement de 5 s :



2.6.6. Actions temporisées

Enclenche un temporisateur t (ici t_3).

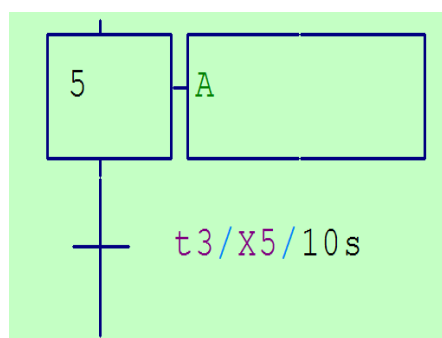
Exemple :

Syntaxe d'une réceptivité : $t_3 / X5 / 10s$;

T3 : repère du temporisateur, ici n° 4;

X5 : numéro de l'étape qui lance la temporisation;

10s : durée de la temporisation.

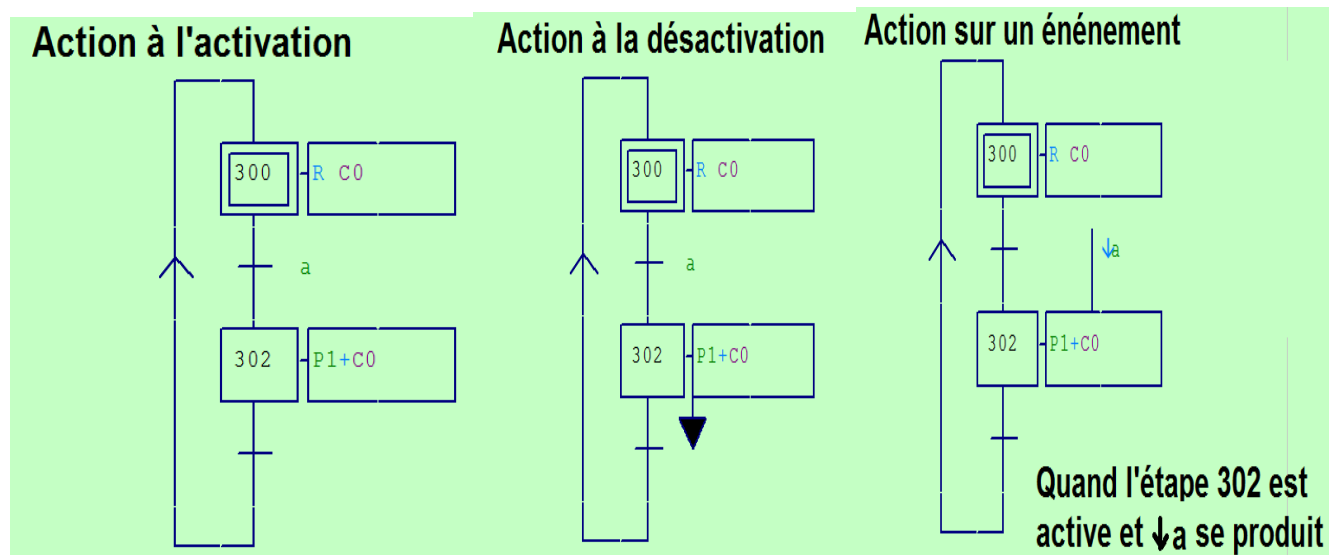


2.6.7. Actions mémorisées

Une étape à action mémorisée permet de mettre la sortie correspondante dans un état spécifié lors de son activation ou de sa désactivation et nous pouvons désactiver cette mémorisation dans une autre étape en associant la lettre R (reset) dans le cadre de l'action attachée à cette étape.

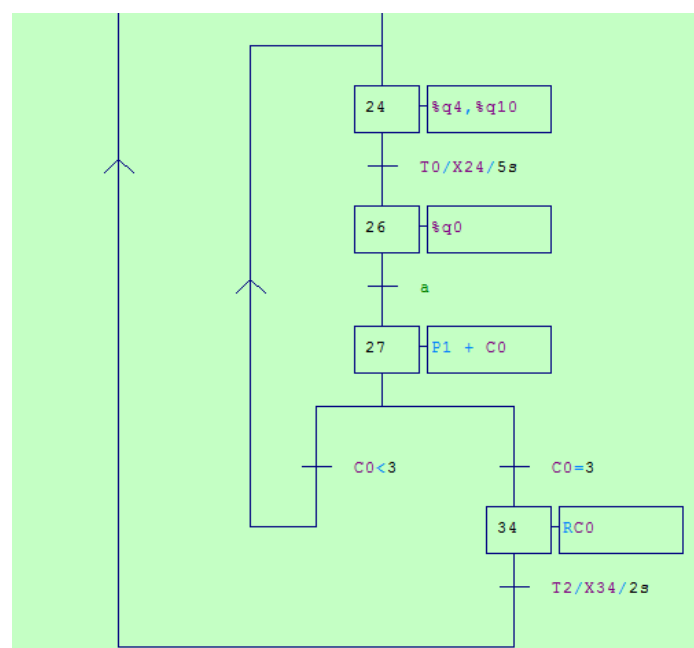
Exemple:

Compteur C0 est incrémenté chaque fois à l'étape 302 puis est réinitialisé à l'étape 300 "remettre à zéro":



Remarque sur les compteurs

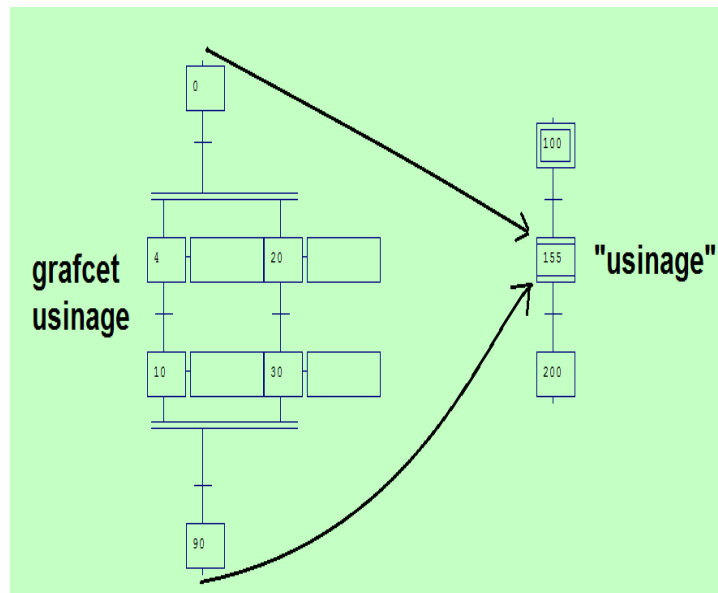
Un réservoir peut permettre d'exécuter une actions ou des actions plusieurs fois mais cette répétition peut être faite par un compteur: ici les actions des étapes 24 , 26 et 27 se répètent 3 fois avant de se poursuivre.



2.7. Les Macro étapes

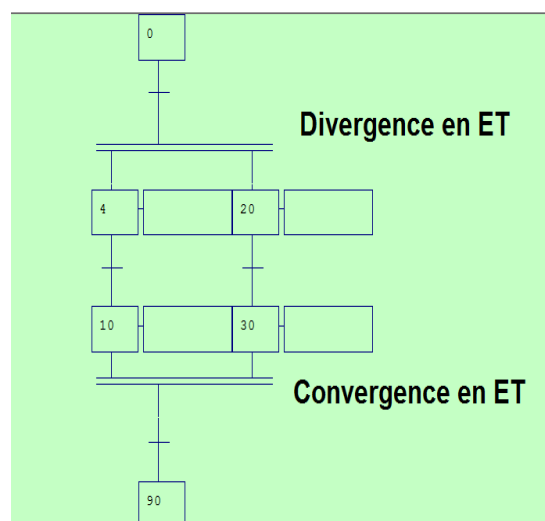
Une macro-étape est un ensemble d'étapes et de transitions qui commence par une seule étape d'entrée et se termine par une seule étape de sortie qui a pour objectif de simplifier un grafcet général en réduisant les grafkets des tâches secondaires d'un processus technologique.

On note qu'il n'y a pas d'action associée à une macro-étape.



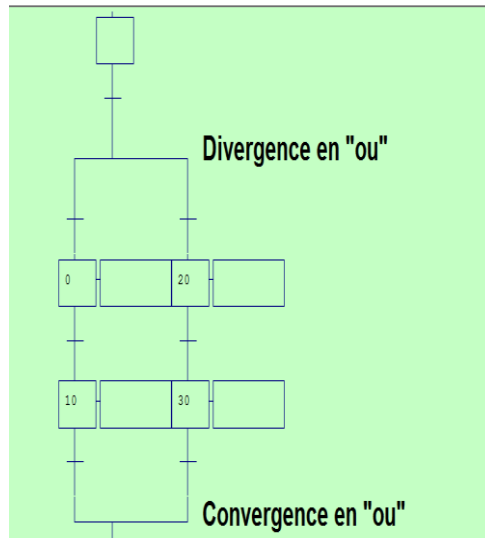
2.8. Convergence et divergence "en et"

Convergence "en et" de plusieurs étapes vers une même transition, donc on rassemble les arcs issues de ces étapes avec d'une double barre horizontale. Par contre la **divergence "en et"** de plusieurs étapes vers une même transition, alors on rassemble les arcs allant vers ces étapes à l'aide d'une double barre horizontale.



2.9. Convergence et divergence "en ou"

La convergence "en ou" de plusieurs transitions sont connectées à une même étape et on rassemble les arcs par un simple trait horizontal. Par contre la divergence "en ou" de plusieurs transitions sont jointes à une même étape, on rassemble les arcs par un simple trait horizontal.



Exemple d'application 1 :

Cahier des charges du WAGONNET:

- Ce wagonnet (chariot) doit aller chercher des produits aux endroits matérialisés par les capteur **S4**, **S3** et **S5** puis revenir à son point de départ .
- chaque nouvel appui sur **S1** relance un nouveau cycle .

Question: Réaliser 1- *Grafcet fonctionnel* 2- *Grafcet technologique* 3- *Grafcet d'automate*

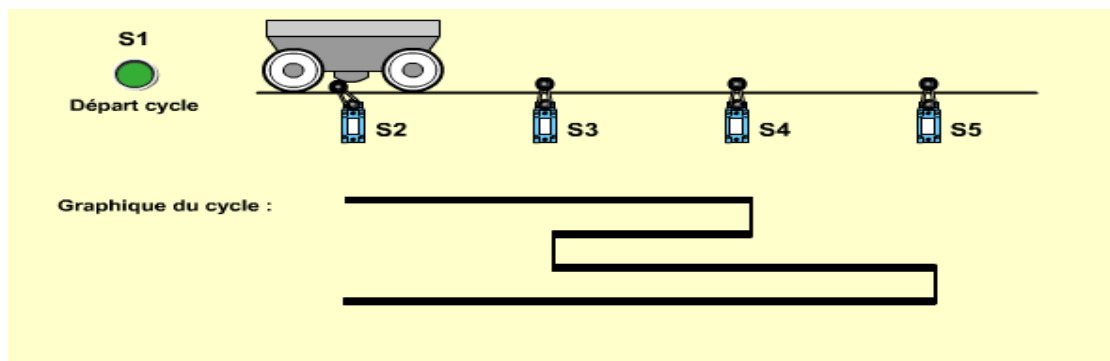
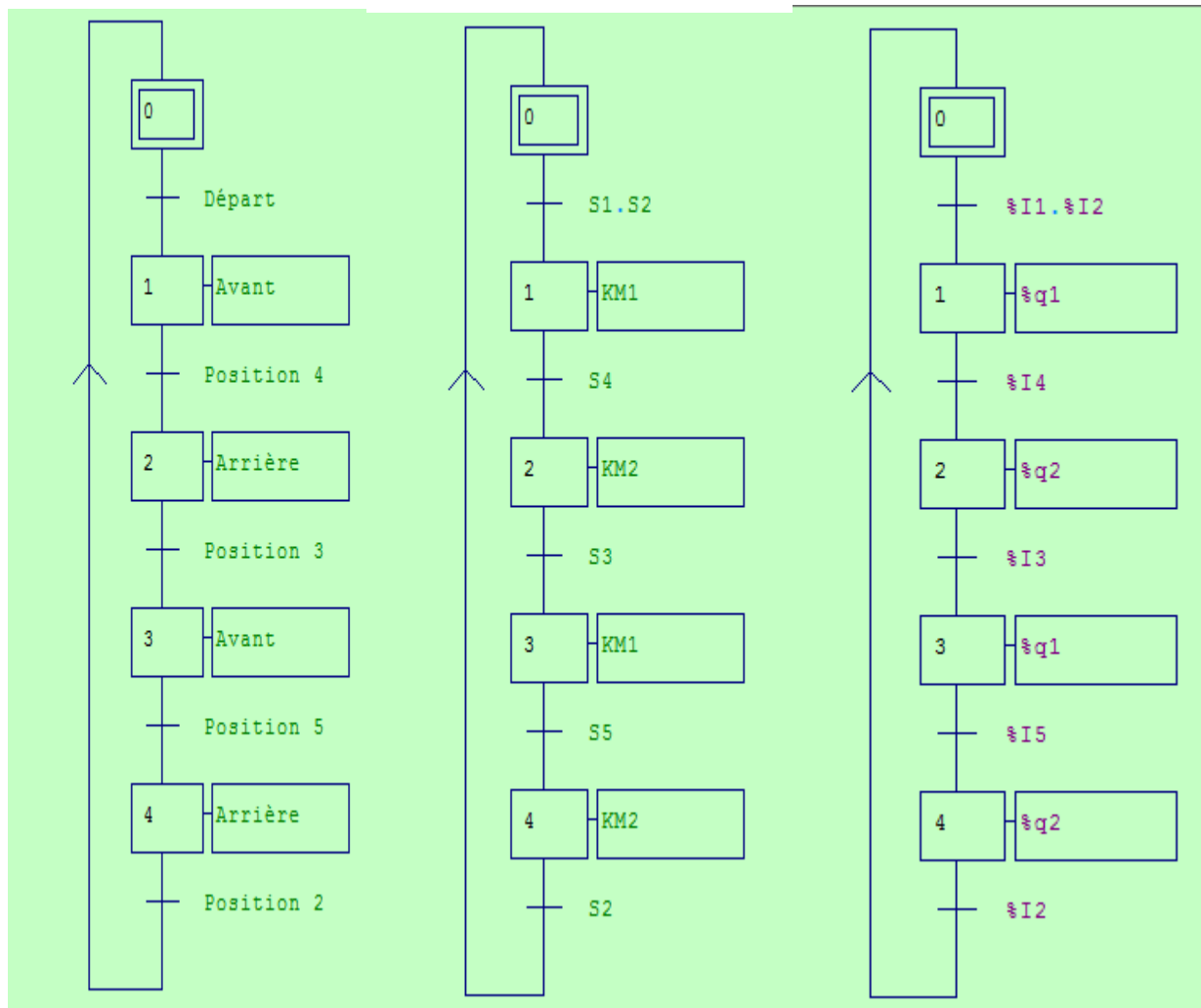


photo capturée à partir de l'animation de: **Ph.JUGUET**



Grafcet du point de vue système

Grafcet du point de vue partie opérative

Grafcet du point de vue partie commande

Grafcet Fonctionnel

Grafcet technologique

Grafcet d'automate

Tableau des variables

Variables d'entrées (capteurs)		Variables de sorties (actionneurs)	
S ₁	%I ₁	KM1	%q ₁
S ₂	%I ₂	KM2	%q ₂
S ₃	%I ₃		
S ₄	%I ₄		
S ₅	%I ₅		

Chapitre 3

L' Automate programmable industriel (API)

- * Objectif : l'étudiant sera capable de connaître
- * sa structure interne et la fonction de tous ses éléments ainsi que les méthodes et les langages pour sa programmation.
- * les différents types des APIs et les critères de choix d'un APIs.

3.1. Introduction

Les **Automates Programmables Industriels (API)**, ou en anglais, **Programmable Logic Controller (PLC)**, sont apparus aux Etats-Unis vers 1969 où ils ont remplacés la logique à relais câblée, à la demande des industries de l'automobile (**GM**), qui réclamait plus de développement des chaînes de fabrication automatisées pour poursuivre l'évolution des techniques et des modèles fabriqués.

3.2. Définition

Un **automate programmable industriel** est un appareil électronique, programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande des préactionneurs d'actionneurs au moyen de signaux d'entrées et de sorties à partir d'informations logiques, analogiques ou numériques.

3.3. Avantages et inconvénients des API

Les automates programmables industriels ont de nombreux avantages par rapport à la logique câblée. on peut citer parmi ces avantages :

- la simplicité : avec le même API, on peut avoir différentes applications par contre la logique câblée, chaque fois, il lui faut des matériels différents.
- la flexibilité : le changement du fonctionnement du processus commandé se fait uniquement par un simple changement du programme.
- la diminution des coûts de câblage et le nombre de maintenance.
- la réduction de l'espace des installations.
- les éléments des APIs sont particulièrement robustes qui peuvent supporter des environnements hostiles et les conditions extrêmes.
- Ils offrent un temps d'exécution minimal en respectant un délai fixé.

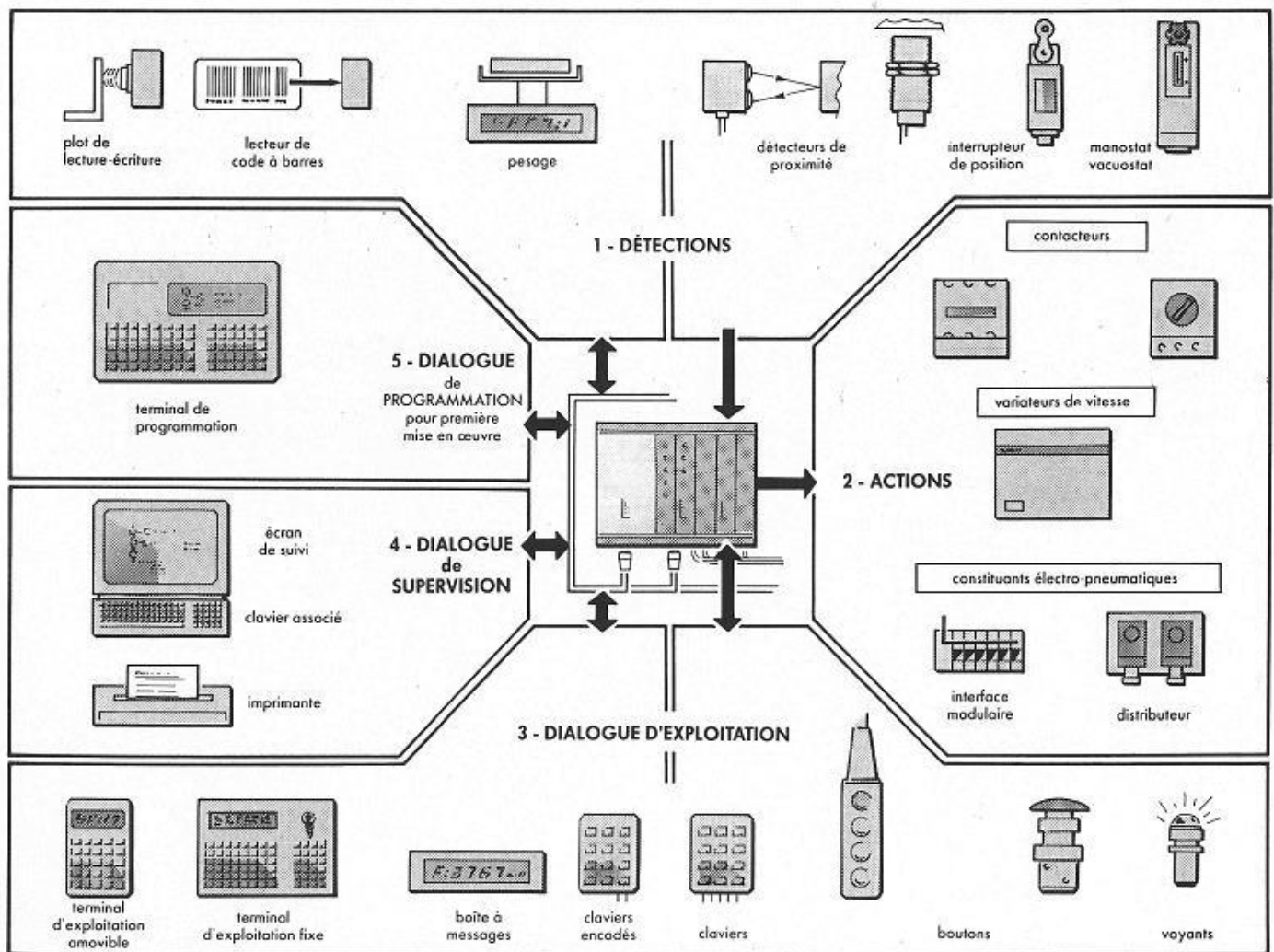
Par contre, ils ont des inconvénients suivants :

- un prix cher qui dépend du nombre d'entrées/sorties nécessaires, de la capacité de mémoire pour le programme à réaliser, ...etc.
- Avoir une connaissance des langages de programmation des APIs.

3.4. Fonctions principales réalisées par API

L'automate programmable Industriel effectue les fonctions suivantes :

1. Détecter les différentes informations par tous types de capteurs implantés sur la machine.
2. Commander des actions par les actionneurs et les pré-actionneurs
3. Avoir un dialogue d'exploitation : hommes-machine pour la bonne conduite des machines automatiques , pour leurs réglages et pour leurs dépannages.
4. Avoir un dialogue de supervision de production: Les APIs ont un écran de visualisation pour voir l'évolution des entrées / sorties
5. Avoir un dialogue de programmation pour la première mise en oeuvre.



Fonctions principales réalisées par API

3.5. Aspect extérieur des API

Il y a deux types des automates programmables industriels : type compact ou type modulaire.

3.5.1. Type compact ou aussi appelé centralisé

Il contient dans un seul boîtier (appelé **rack**): le processeur, l'alimentation, les entrées et les sorties et peut avoir des extensions en nombre limité. Ces automates sont généralement utilisés pour la commande de petits automatismes.



API compact

3.5.2. Type modulaire

L'automate programmable est réalisé par un ensemble de blocs qui est formé d'un module (rack, coffret, cartes). Ces différents modules sont connectés ensemble autour du bus interne pour la communication. Ce type d'automate contenant un rack, un processeur, des modules d'E/S, un module d'alimentation, des modules de communication et de comptage. Ce type d'automate offre une grande souplesse de configuration, de diagnostic et la facilité de la maintenance et elle utilisé dans les automatismes complexes.

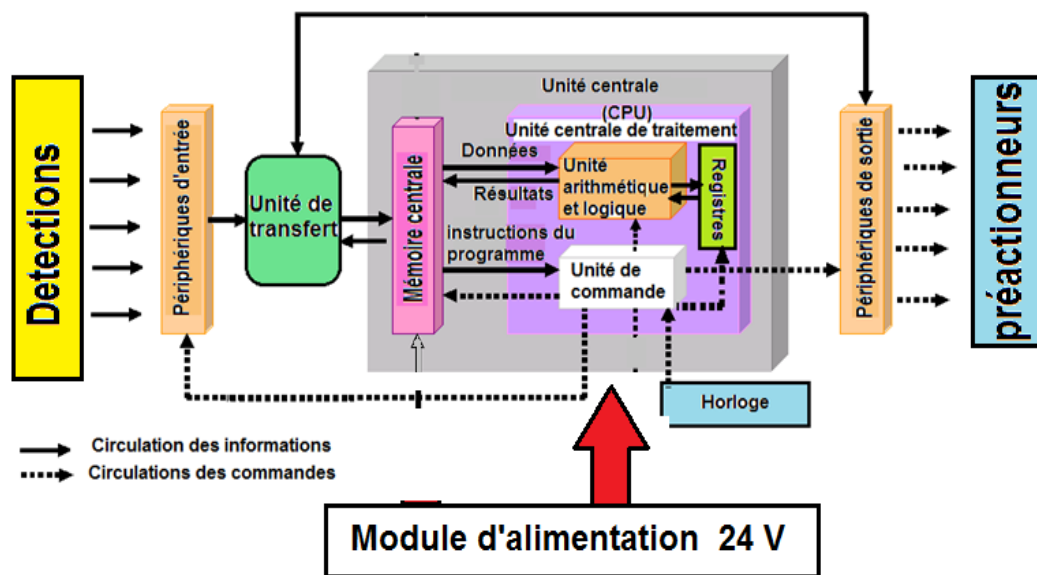


API modulaire.

3.6. Les grandes marques

-  GENERAL ELECTRIC;
-  SIEMENS;
-  TELEMECANIQUE;
-  TEXAS INSTRUMENT.

3.7. Structure interne des APIs



Structure interne des API

La structure interne d'un automate programmable industriel (API) comporte : Une unité de traitement (un processeur CPU); Une mémoire centrale ; des Interfaces d'entrées destinées pour la lecture des informations des détecteurs ou capteurs ; des Interfaces sorties pour la commande des préactionneurs ou les actionneurs

Une alimentation souvent de 24 V . des bus internes pour l'échanger les informations entre les différents composants internes de l'automate (processeur , interfaces entrées-sorties, mémoires).

3.7.1. Le processeur (microprocesseur)

Il est pris comme une boîte noire dans ce cours de l'automatisme industriel puisque nous nous intéressons uniquement à l'automatisation des machines industrielles les systèmes micro-programmés. **Le processeur** fait la manipulation des données sous formes de transfert, opérations arithmétiques, opérations logiques.... En un mot, il est considéré comme l'unité intelligente de traitement des informations. Son rôle, autrement dit, est de lire des programmes (suites d'instructions), de les décoder et finalement de les exécuter en suivant le rythme et les séquences de l'horloge .

Le cerveau d'un automate est **le microprocesseur** qui est composé de **l'unité centrale de traitement (UCT)** ou (en anglais CPU : Central Processing Unit) [13].

3.7.2. Mémoire

Elle est conçue pour :

- recevoir les informations de l'interface d'entrées issues des capteurs
- recevoir les informations traitées par le processeur et envoyées à travers l'interface des sorties pour la commande des préactionneurs .
- recevoir et conserver le programme de l'automatisation du processus industriels.

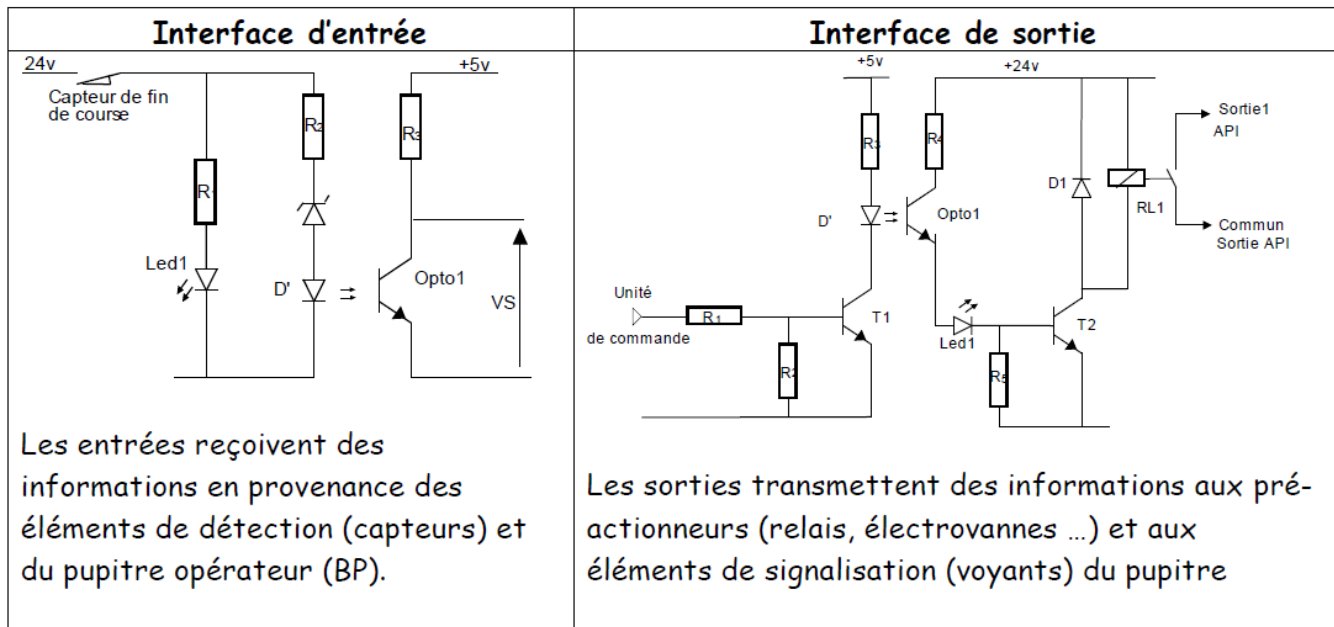
Il y a dans les APIs trois types de mémoires:

- ✓ **Mémoire de programme** : Cette mémoire est pour stockage du programme . Elle est en général de type EEPROM (mémoires mortes peut être reprogrammables par effacement électrique).
- ✓ **Mémoire système** : elle présente, dans le cas d'automates à microprocesseurs, est employée pour stocker le programme dans l'industrie par le constructeur. Elle est réalisée en technologie PROM (programmable une seule fois,) ou ROM (mémoire morte accessible uniquement en lecture sans possibilité d'effacement).
- ✓ **Mémoire de données** : Elle offre la possibilité de la lecture ou l'écriture des données. Elle est réalisée en technologie spéciale (CMOS) dite RAM (mémoire vive) à l'état de repos pour sauvegarder les données, il lui faut une batterie de sauvegarde.

3.7.3. Interfaces et cartes d'Entrées / Sorties

Les interfaces d'entrées reçoivent des informations de la détection des capteurs et du pupitre de l'opérateur (BP). Par contre les interfaces de sorties transmettent des commandes aux pré-actionneurs et aux différents éléments de signalisation du pupitre.

Le nombre de voies de ces entrées ou des ces sorties ou leur cartes d'E/S varient selon le type d'automate (8, 16 ou 32 voies). Les tensions disponibles sont normalisées (24, 48, 110 ou 230 V) peuvent être continu ou alternatif.



3.7.4. Alimentation électrique

Tous les APIs actuels sont dotés d'une alimentation 240 V 50/60 Hz, 24 V continu. Les entrées sont en 24 V continu avec une mise à la terre qui est indispensable .

3.7.5. Modules complémentaires (spéciaux)

Les automates compacts offre la possibilité de commander les sorties en T.O.R et réalisent le comptage et le traitement analogique. Les automates modulaires permettent de gérer autres fonctions grâce à des modules intelligents que l'on ajoute sur un ou plusieurs racks ainsi on évite de surcharger le CPU .

➤ Principales fonctions

- **Cartes de comptage rapide:** elles offrent des informations de fréquences élevées incompatibles avec la fréquence basse de traitement de l'automate.
- **Cartes d'entrées / sorties analogiques:** Elles permettent de recevoir un signal analogique et le convertir en numérique (CAN) nécessaire pour traitement par le CPU. Ainsi que la fonction inverse est possible (sortie analogique). Les grandeurs analogiques sont standardisées : 0-10V ou 4-20mA.
- **Cartes de communication (RS485, Ethernet ...)** : Ils offrent la possibilité des communications à distance avec d'autres systèmes d'exploitation et de traitement par: fibres optiques, paires téléphoniques, coaxiales,...
- **Cartes d'entrées / sorties déportées:** pour réduire le volume de câblage entre l'automate et le processus , ces **cartes d'entrées / sorties déportées** permettent de déconcentrer des châssis entrées / sorties sur des longues distances (des kms).

3.7. Critères de choix des APIs

Les critères de choix essentiels d'un automate programmable industriel sont :

- Le nombre et la nature de l'interface E/S ;
- La puissance de traitement du CPU (vitesse, largeur du mot de données, largeur du bus d'adresse, temps de réponse ,...).
- Fonctions disponibles ou modules spéciaux
- Les langages de programmation possibles et les moyens de dialogue
- Les différentes communications possibles avec les autres systèmes ;
- Les méthodes de la conservation du programme ;
- La robustesse, fiabilité, immunité aux parasites de différents types;
- La disponibilité et la richesse de la documentation ,la durée de la garantie, le service après vente, la formation.

3.8. Programmation d'un API :

Elle peut être faite de trois façons différentes :

- ✓ à l'aide de touches qui se trouvent sur l'A.P.I. lui-même.
- ✓ Grâce à une console spécifique de programmation attachée par un câble spécifique à l'A.P.I.
- ✓ Grâce à un PC dans lequel on a installé un logiciel approprié à cet A.P.I.

3.8.1. Langages de programmation :

On cite les langages de programmation les plus connus dans l'industrie :

- **Les langages graphiques :**
 - ✓ **LD (Leader Diagram → schéma à contacts)** : Un programme écrit en langage à contacts se compose d'une suite de réseaux de contacts composés d'un ensemble d'éléments graphiques disposés sur grille organisée en lignes et colonnes.
 - ✓ **SFC (Séquentiel Function Chart (Grafcet) → langage G7)** : permet la programmation de tous les procédés séquentiels .
 - ✓ **FBD (Function Block Diagram → Schéma par Bloc)** : permet de programmer graphiquement à l'aide des blocs, représentant des variables , des opérateurs ou des fonctions .

- **Les langages textuels :**

- ✓ **IL**(Instruction List → liste d'instructions) : Un programme écrit en langage liste d'instructions se compose d'une suite d'instructions exécutées séquentiellement par l'automate. Chaque instruction est composée d'un code instruction et d'un opérande .
- ✓ **ST**(Structured Text →Texte structure) : permet la programmation de tout type d'algorithme plus ou moins complexe.

Références

- 1-<https://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm>
- 2-https://www.est-usmba.ac.ma/GRAFCET/co/module_cours_grafcet_23.html
- 3-<https://www.conrad.fr/p/module-de-commande-eaton-256269-easy-819-dc-rc-24-vdc-1-pcs-198372>
- 4- <http://4electromecanique.blogspot.com/2016/10/les-types-dautomate-programmable.html>
- 5- <https://sites.google.com/a/usmba.ac.ma/mostafa-elmallahi/home/cours-n-2-architectureordinateur?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>
- 6- Jlassi Khaled, Microprocesseurs et Microcontrôleurs, Université Virtuelle de Tunis,2008.
- 7- J.Y. Haggège, Microprocesseur, Support de cours, Institut Supérieur des Etudes Technologiques de Radés ; 2003.
- 8-Cours les Automates programables industriels "KAAOUANA ISMAIL, Lycée Hannibal ARIANA".
- 9- Cours Automates Programmables Industriels ,Dr. Sofiane Doudou;Université Mohammed Seddik Benyahia -Jijel.
- 10-Automates Programmables Industriels, professeur Mahbab Lycée technique Mohammedia Maroc
- 11-Notes Cours Automatismes Industriels réalisés par Tahar Askri
- 12- Module d'automatisme ; Université Kasdi Merbah d'Ouargla Université Kasdi Merbah d'Ouargla; 2008/2009.
- 13- <https://www.technologuepro.com/cours-automate-programmable-industriel/Cours-Grafcet-notions-de-base.htm>
- 14- <http://www.coursmaroc-ayochti.com/2015/03/formation-grafcet.html>
- 15- Livre "Automatismes et automatique" cours et exercices corrigés ; Jean-Yves FABERT, chez ellipses;édition 2005.
- 16- Les photos sont capturées à partir de l'animation de: **Ph.JUGUET** / BEP Électrotechnique / Lycée A.KASTLER / 91410.DOORDAN_ et à partir du site web de FESTO