



République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Direction Générale des Etudes Technologiques

---***---

Institut Supérieur des Etudes Technologiques de Sousse



Support du cours automatisme industrielle

Spécialité : Génie électrique

Elaboré par : TLILI KAIS

TECHNOLOGUE à L'ISET DE SOUSSE

Sommaire

Chapitre 1 : systèmes automatisés de production.....	1
1. Présentation générale des systèmes.....	1
1.1- Définition d'un système selon 2 aspects.....	1
1.2- Les systèmes automatisés.....	1
2. Classification d'un système.....	1
2.1- Selon le domaine d'application.....	1
2.2- Selon le contexte technico-économique.....	2
3. Organisation structurelle d'un système automatisé.....	2
<i>Étude sur un exemple : Fermeture et ouverture automatique d'un portail :</i>	2
3.1- Constitution d'un système automatisé : PE, PC et PO.....	3
3.2- Relations entre PE, PC et PO.....	3
(Consignes, ordres, comptes rendus, signalisations).....	3
La PC émet à la PE des signalisations pour informer de l'état dans lequel elle se trouve.	3
4. Organisation fonctionnelle d'un système automatisé : Chaîne fonctionnelle.....	4
4.1- Chaîne fonctionnelle : Obtention d'une seule fonction.....	4
4.2- Chaînes d'énergie et d'information.....	4
4.2- Flux d'énergie, flux d'information et flux de matière.....	5
5. Constituants de la chaîne d'énergie.....	5
5.1- Les préactionneurs.....	5
5.2- Les actionneurs.....	6
5.3- Les transmetteurs (ou adaptateurs).....	7
6. Les effecteurs.....	8
7. Constituants de la chaîne d'information.....	9
7.1- Les capteurs.....	9
7.2- Les interfaces homme/machine.....	11
7.3- Les parties commandes.....	12
Chapitre 2 : Synthèse des automatismes décrits par GRAFCET.....	13
1. Introduction.....	13
2. Définition.....	15
3. Règles d'évolution.....	17
4. Structures de base.....	21
4.1- Divergence et convergence en ET (séquences simultanées).....	21
4.2- Divergence et convergence en OU (aiguillage).....	22
4.3- Saut en avant (saut de phase).....	22
4.4- Saut en avant (saut de phase).....	23
5. Classification des actions.....	24
Chapitre 3 : Mise en œuvre du GRAFCET.....	26

1. Mise en équation du GRAFCET	26
1.1-Objectif	26
1.2.- Mise en équation d'une étape.....	26
1.3- Gestion des modes Marche/Arrêt et des arrêts d'urgences	28
Chapitre 4 : Matérialisation des automatismes décrits par grafcet à l'aide de séquenceurs. ..	30
1. Réalisation par câblage.....	30
1.1- Elément de la logique combinatoire (portes logiques)	30
1.2- Elément de la logique Séquentielle (bascule RS)	31
2. Séquenceur pneumatique.....	34
2.1- Introduction.....	34
2.2- Montage d'un séquenceur pneumatique à partir de modules d'étape	37
Chapitre 5 : Automates Programmables Industriels (API).....	39
1. Introduction	39
2. Structure d'un API	39
2.1- Description des éléments d'un API.....	41
2.2- Le choix de l'unité centrale.....	43
2.3- Le temps de réponse d'un API.....	44
3. Langages de programmation pour les API.....	46
4. Etude des différentes interfaces.....	47
4.1- Les cartes d'entrées TOR.....	47
4.2- Les cartes de sorties TOR	48
4.3- Les cartes d'entrées analogiques.....	49
4.3- Les cartes de sorties analogiques	50
4.3- La carte de comptage rapide	51
4.4- La commande d'axe	53

Chapitre 1 : systèmes automatisés de production

1. Présentation générale des systèmes.

1.1- Définition d'un système selon 2 aspects.

Un système est un produit (rappel de la définition d'un produit : ce qui est fourni à l'utilisateur pour répondre à un besoin). Il devra vérifier les performances attendues du cahier des charges. On peut donc regarder un système suivant deux aspects :

Exemple : Aspect structurel : Ensemble de constituants en relation (moteur, aile, gouverne...) un avion
Aspect fonctionnel : Ensemble de fonctions en relation (propulser, porter, diriger...)

1.2- Les systèmes automatisés.

Les systèmes techniques peuvent être de nature très diverse et différentes disciplines peuvent s'interpénétrer : l'automatique, l'informatique, la mécanique, l'électronique, l'électrotechnique, ...

Plus le système est complexe, plus les disciplines concernées sont nombreuses. Nous nous intéresserons principalement aux systèmes automatisés dans lesquels tout ou une partie du savoir-faire est confié à une machine.

2. Classification d'un système.

2.1- Selon le domaine d'application.

- ✓ L'aéronautique : avion, mais aussi à l'intérieur de celui-ci son système de navigation, système d'armes...
- ✓ L'automobile : voiture, système de climatisation, ABS, direction assistée, suspension active...
- ✓ L'électroménager : machine à laver, cafetière, four micro-onde...
- ✓ L'électronique grand public : téléviseur, téléphone portable, lecteur DVD, mais aussi les systèmes d'émission et de réception par satellite des signaux audios et vidéos
- ✓ La domotique : store, ouvre-portail, climatisation...
- ✓ Le service : distributeur de billets, horodateur, station de lavage, ascenseur...
- ✓ Le médical : respirateurs, pompes, rein artificiel...

2.2- Selon le contexte technico-économique.

Suivant le contexte technico-économique, la conception des systèmes se fait différemment.

On distingue :

- **Les systèmes de diffusion limitée rencontrés dans les usines de production industrielle.**

Ces systèmes, souvent unitaires, font l'objet d'études réduites et sont constitués d'éléments standards assemblés.

- **Les systèmes de grande diffusion rencontrés dans l'électroménager, la domotique, la distribution.**

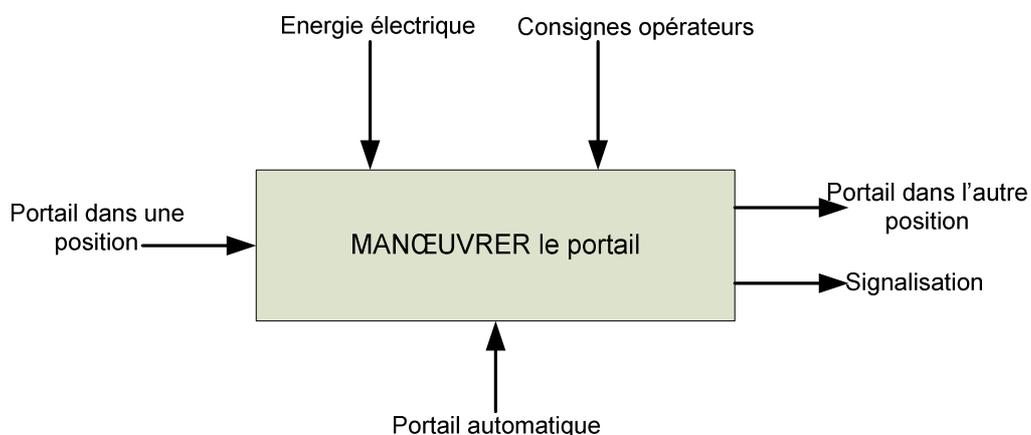
Ces systèmes, produits en grande série, font l'objet d'études poussées pour optimiser les coûts. Le design est souvent un élément important.

3. Organisation structurelle d'un système automatisé.

Étude sur un exemple : Fermeture et ouverture automatique d'un portail :

A la réception du signal de commande radio, la carte électronique commande :

- Soit l'ouverture (déverrouillage de la serrure, puis action des 2 motoréducteurs)
- Soit la fermeture (action des 2 motoréducteurs, puis verrouillage de la serrure).



3.1- Constitution d'un système automatisé : PE, PC et PO.

Un système automatisé est constitué :

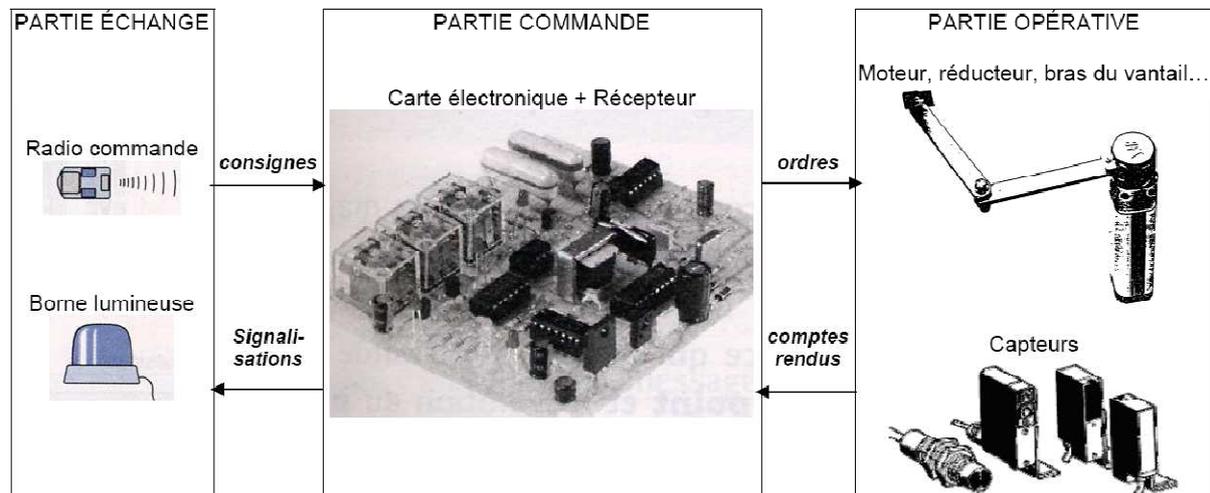
- D'une PARTIE OPERATIVE (PO) en général mécanisée, qui agit physiquement sur la matière d'oeuvre, pour lui apporter la valeur ajoutée.
- D'une PARTIE COMMANDE (PC) qui traite l'information pour assurer le pilotage et la coordination des tâches.
- D'une PARTIE ECHANGE (PE) qui permet à l'opérateur (utilisateur) de dialoguer avec la PC.

3.2- Relations entre PE, PC et PO

(Consignes, ordres, comptes rendus, signalisations).

- La PE émet à la PC des consignes (demande d'ouverture, de fermeture, d'arrêt ...).
- La PC émet à la PO des ordres pour coordonner le déroulement des opérations.
- La PO émet des comptes rendus caractérisant les états des matières d'oeuvres ou des parties mécaniques.

La PC émet à la PE des signalisations pour informer de l'état dans lequel elle se trouve.



4. Organisation fonctionnelle d'un système automatisé : Chaîne fonctionnelle.

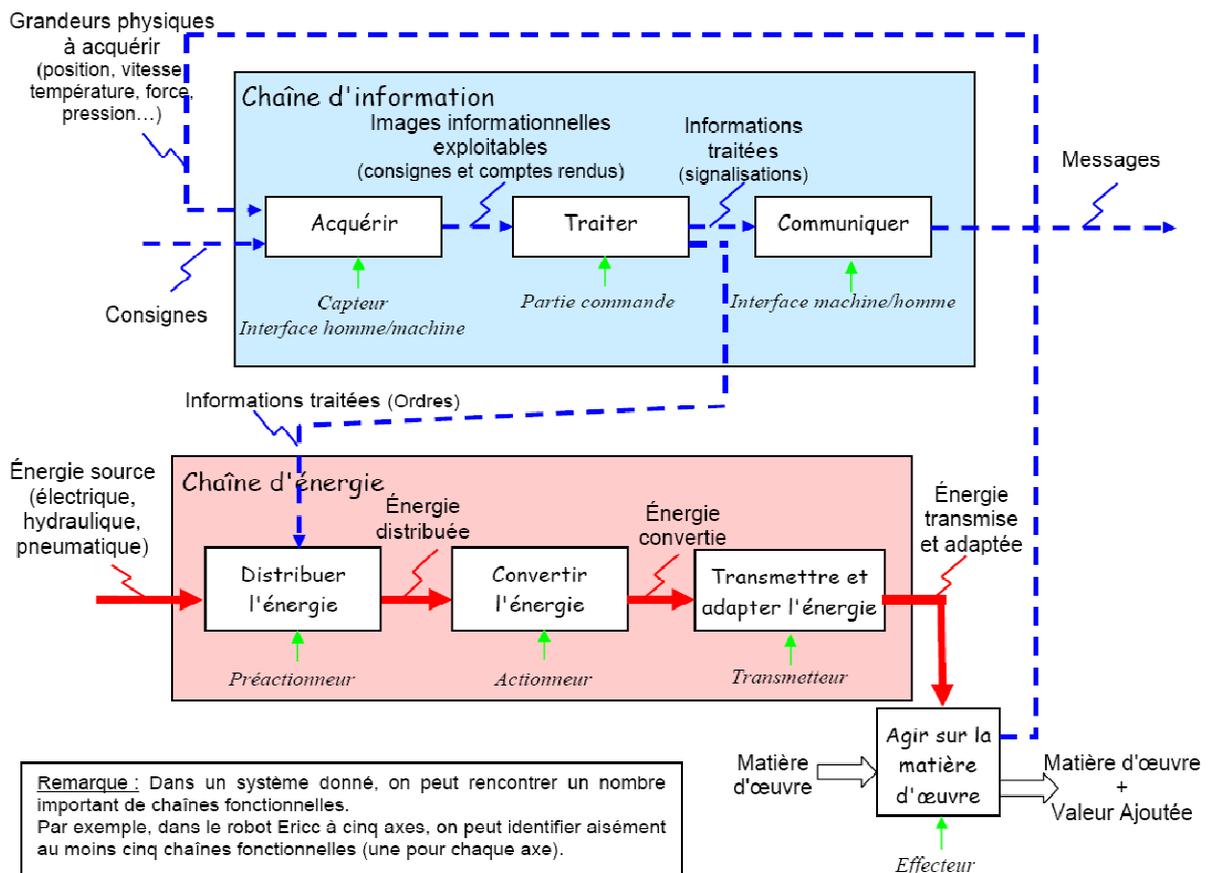
4.1- Chaîne fonctionnelle : Obtention d'une seule fonction.

Une chaîne fonctionnelle est un ENSEMBLE DE CONSTITUANTS ORGANISES en vue de l'obtention d'UNE SEULE FONCTION PRINCIPALE (par exemple : prendre un objet, déplacer une charge, chauffer une pièce...) : c'est un système automatisé élémentaire.

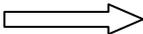
4.2- Chaînes d'énergie et d'information.

Une chaîne fonctionnelle comporte :

- une chaîne d'énergie qui réalise une action à partir d'énergies disponibles.
- une chaîne d'information qui réalise l'acquisition et le traitement d'informations.



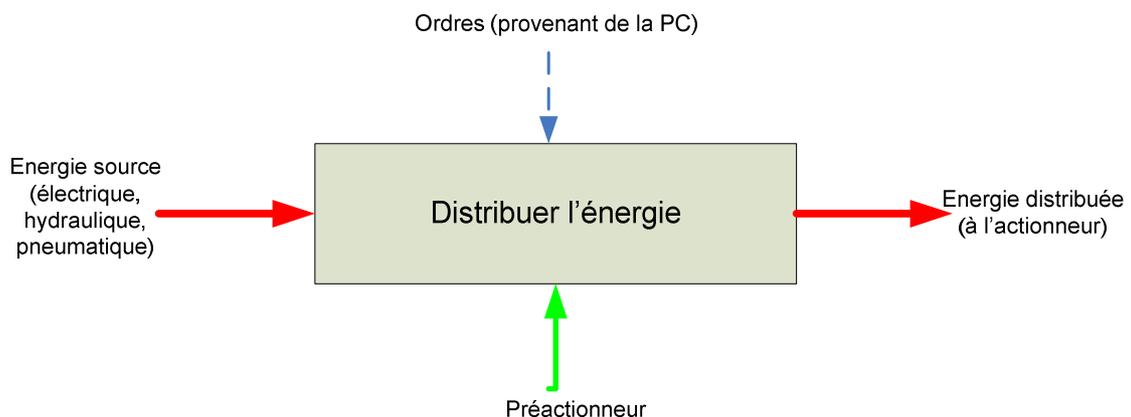
4.2- Flux d'énergie, flux d'information et flux de matière.

Différents types de flux utilisés dans une chaîne fonctionnelle	Représentation	électrique	pneumatique	hydraulique
Dans la chaîne d'énergie où l'énergie utilisée doit être importante pour donner aux matières d'oeuvre la valeur ajoutée attendue, nous parlerons de FLUX D'ENERGIE .		220V	7 bar	250 bar
Dans la chaîne d'information où l'énergie utilisée doit être faible pour véhiculer des signaux, nous parlerons de FLUX D'INFORMATION .		24V	3 bar	10 bar
Au niveau de l' effecteur , nous parlerons de FLUX DE MATIERE (pièce, matériau...).				

5. Constituants de la chaîne d'énergie.

5.1- Les préactionneurs.

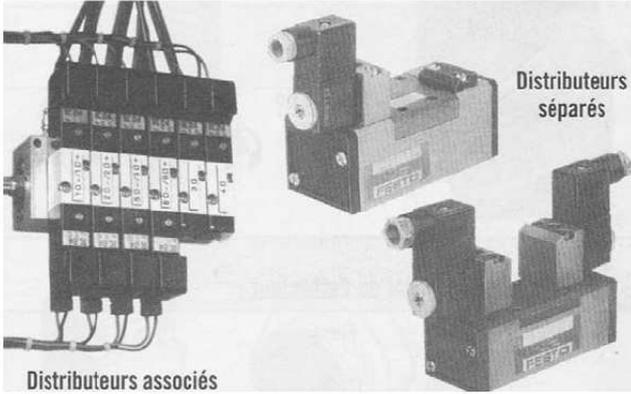
Définition : Élément qui laisse passer l'énergie source à l'actionneur sur ordre de la partie commande. (Le préactionneur réalise l'interface de dialogue (PC ↔ PO))



Explications : La partie commande envoie par l'intermédiaire de son circuit de commande un ordre de faible niveau pour établir ou fermer un circuit de puissance. Suivant la présence de ce signal, le préactionneur distribuera l'énergie source à un actionneur (moteur, résistance

chauffante, lampe... pour une énergie source électrique, ou vérin, générateur de vide... pour une énergie source pneumatique).

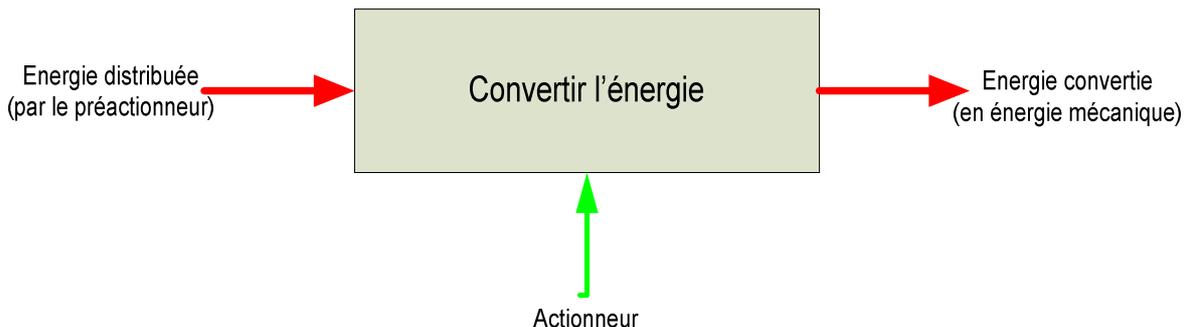
Les composants les plus utilisés.
Si l'actionneur est ELECTRIQUE le préactionneur qui lui est associé est

<p>Le relais électromagnétique pour les faibles puissances (cas de l'ouvre-portail)</p> 	<p>Le contacteur pour les puissances élevées</p> 	<p>Si l'actionneur est PNEUMATIQUE OU HYDRAULIQUE le préactionneur qui lui est associé est</p> <p>Le distributeur</p> 
--	---	---

NB : Les préactionneurs ci-dessus sont appelés PREACTIONNEUR TOUT OU RIEN, c'est-à-dire qu'ils sont une sorte d'interrupteur de la chaîne d'énergie. Mais il existe aussi des préactionneurs qui laissent passer seulement une partie de l'énergie source, c'est-à-dire qu'ils régulent le débit d'énergie source, on parle alors de PREACTIONNEUR PROPORTIONNEL. Exemples : variateur de vitesse (utilisé sur la capsuleuse de bocaux), hacheur (sur le maxpid), carte de puissance (sur la cordeuse de raquette)....

5.2- Les actionneurs.

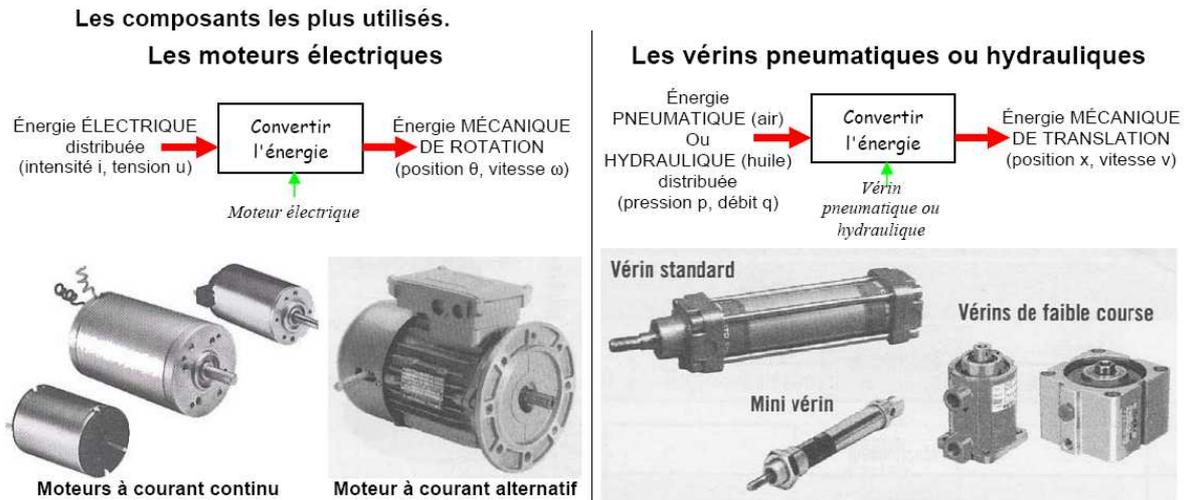
Définition : Élément qui convertit une énergie d'entrée non directement utilisable par les mécanismes agissant sur la matière d'oeuvre en une énergie de sortie utilisable par ces mécanismes pour obtenir une action définie.



Explications :

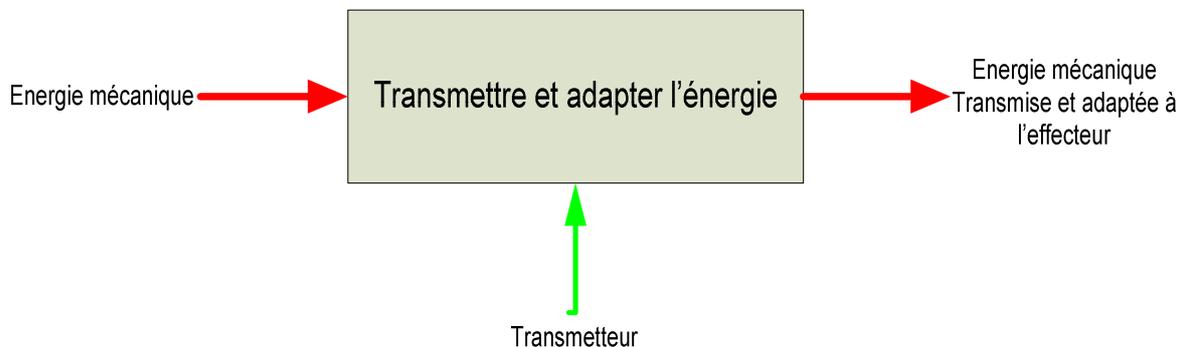
Afin d'agir sur la matière d'oeuvre, la partie opérative a besoin d'énergie de haut niveau.

L'énergie source employée est le plus souvent de nature électrique ou pneumatique, parfois hydraulique. Cette énergie source n'est pas directement utilisable et doit être convertie (en général en énergie mécanique) : c'est la fonction des actionneurs.

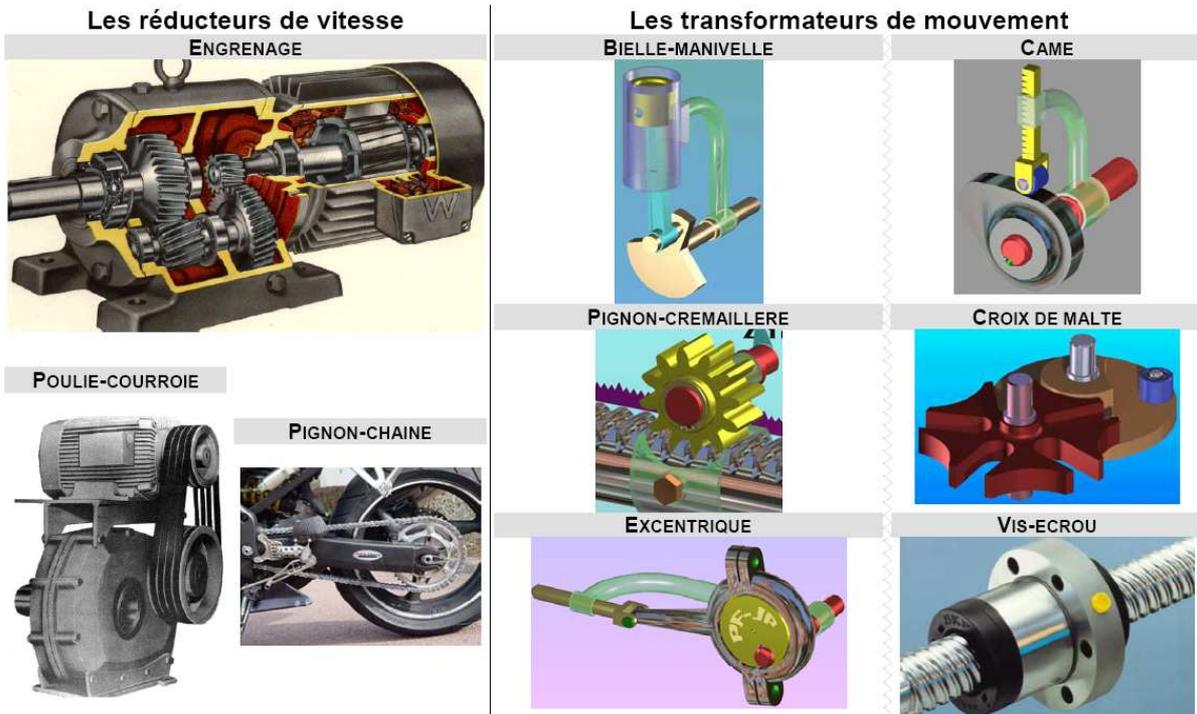


5.3- Les transmetteurs (ou adaptateurs).

Définition : Lorsque l'on veut que l'énergie mécanique produite par l'actionneur ait des caractéristiques bien précises (fréquence de rotation réduite, vitesse linéaire alternative...), on incorpore dans la chaîne d'énergie des adaptateurs d'énergie mécanique.

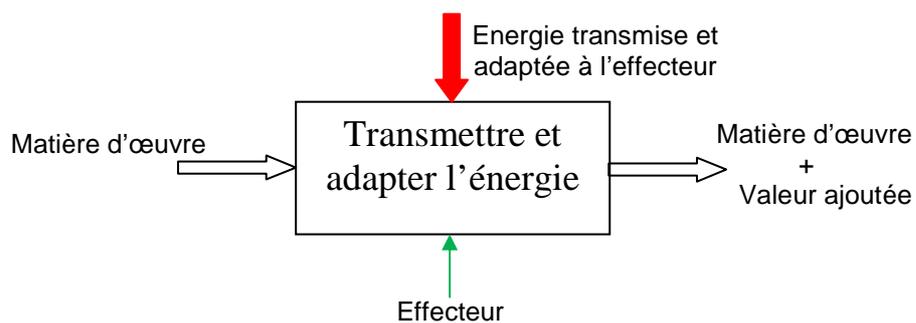


Les composants les plus utilisés.

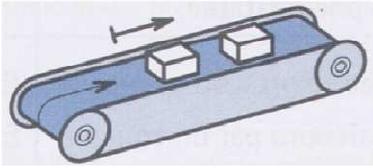
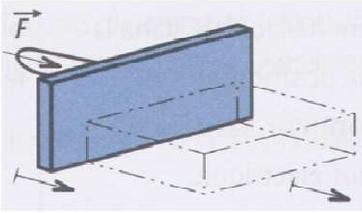
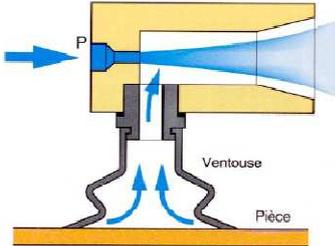


6. Les effecteurs.

Définition : Ce sont les éléments terminaux. Ils agissent directement sur la matière d'œuvre en vue de lui apporter une valeur ajoutée. Ils convertissent l'énergie reçue de l'adaptateur en une opération ou un effet sur la matière d'œuvre.



Effecteurs pour une VA de type déplacement de matières d'œuvres (tapis roulant, ventouse...)

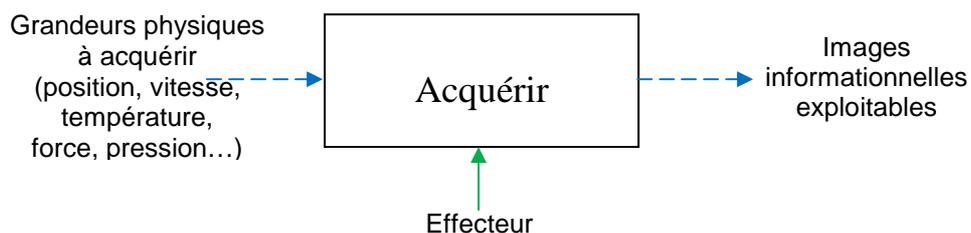
<p>Poser-Entraîner : <i>Tapis roulant ou bande transporteuse ou convoyeur.</i></p>  <p>Principe : Une (ou deux) bande(s) souple(s) entraîne(nt) les pièces par adhérence.</p> <p>Actionneur associé : moteur électrique.</p> <p>Utilisation : Entraînement rapide et continu de pièces ayant une surface plane.</p>	<p>Pousser : <i>Plaque pousseuse.</i></p>  <p>Principe : Une plaque rigide mobile pousse la M.O. selon une direction, dans un seul sens.</p> <p>Actionneur associé : vérin pneumatique.</p> <p>Utilisation : Déplacement en translation, dans un seul sens de pièces prismatiques ou cylindriques.</p>	<p>Saisir-Soulever : <i>Ventouse.</i></p>  <p>Principe : Lorsque la pression d'air P alimente le générateur de vide, le jet d'air turbulent entraîne l'air ambiant (effet Venturi) et le vide ainsi créé permet la saisie de la pièce par la ventouse.</p> <p>Actionneur associé : générateur de vide.</p> <p>Utilisation : Saisie de pièces présentant une surface plane, lisse et non poreuse. Augmenter le diamètre et/ou le nombre de ventouses selon la charge.</p>
--	---	---

7. Constituants de la chaîne d'information

7.1- Les capteurs

Définition : Élément qui :

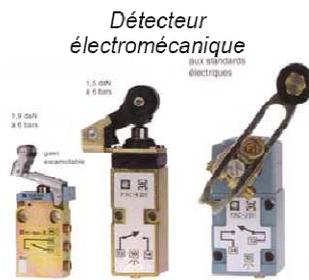
- prélève les états (position, vitesse, température, force, pression...) de la PO ou du milieu extérieur,
- puis les convertit en signaux exploitables (généralement électrique : tension ou intensité) sous forme :
 - logique (0 ou 1),
 - ou analogique,
 - ou numérique,
- et enfin transmet ces signaux à la PC.



Les capteurs TOR (Tout Ou Rien) ou capteurs logiques (2 valeurs).

Ces capteurs transmettent une information de type binaire (donc deux états). Peu coûteux, ce sont généralement des capteurs de position. Par exemple, ils indiquent si une pièce est présente ou non, si une tige de vérin est sortie.... Ils ne permettent pas, en revanche, de mesurer sur toute une plage.

Les détecteurs avec contact



Les détecteurs sans contact



Les détecteurs inductifs : sont sensibles aux matériaux conducteurs. Lorsqu'on approche une pièce métallique du détecteur, le champ magnétique est modifié. Au delà d'un certain seuil, l'objet a été détecté.

Les détecteurs capacitifs : utilisent l'effet d'un condensateur. Pour rappel, un condensateur est simplement deux matériaux conducteurs que l'on met en présence l'un de l'autre mais sans contact. Ce condensateur possède une « capacité » dont la valeur dépend de la géométrie du capteur. Si on vient mettre une pièce entre les deux matériaux, la capacité est modifiée et par la même occasion, le champ électrique. Leur domaine d'application est limité à la détection des liquides car leur coût est élevé.

Les détecteurs photoélectriques : comportent une source lumineuse et un récepteur photosensible. Ils permettent de déceler sans contact tous les matériaux opaques.

Les détecteurs magnétiques ou Interrupteurs à Lame Souple (I.L.S.) : sont disposés sur les extrémités du corps d'un vérin. Un aimant, placé à l'intérieur du piston, attire l'interrupteur (ou la lame souple magnétique) du capteur lorsque le piston est à proximité du capteur.

Les capteurs analogiques (infinité de valeurs).

Ces capteurs transmettent une information prenant une infinité de valeurs. La grandeur de sortie est en relation directe avec la grandeur d'entrée. Le principe est de traduire une modification dimensionnelle (due à un effort, à une pression...) en variation de résistance électrique. Ces capteurs sont linéaires

Jauge d'extensiométrie
(capteurs d'effort)



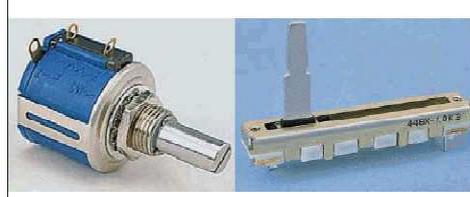
Les pressostats
(capteurs de pression)



Les tachymètres
(capteurs de vitesse)



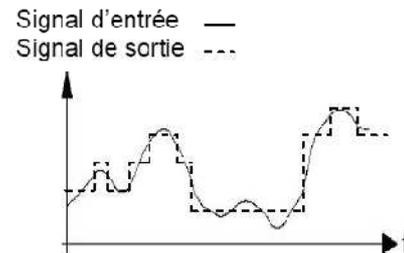
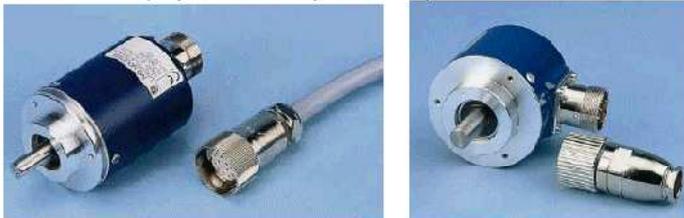
Les potentiomètres résistifs
(capteurs de déplacement)



Les capteurs numériques (nombre limité de valeurs).

Ces capteurs transmettent une information prenant un nombre limité de valeurs distinctes (comme tout signal numérique...).

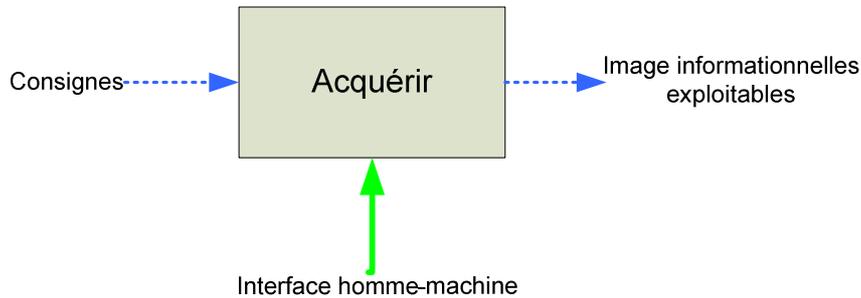
Les codeurs (capteurs de déplacement).



7.2- Les interfaces homme/machine.

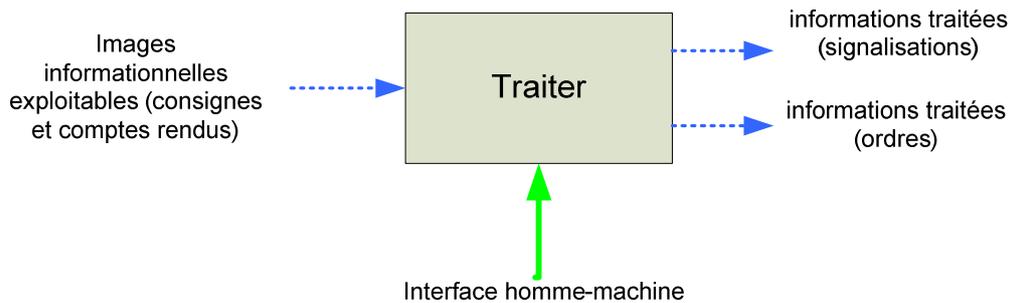
Eléments qui prélèvent, convertissent en signal exploitable et transmettent les consignes à la Partie Commande.

Exemples : **Bouton poussoir, bouton coup de poing, potentiomètre, interrupteur de position, clavier...**



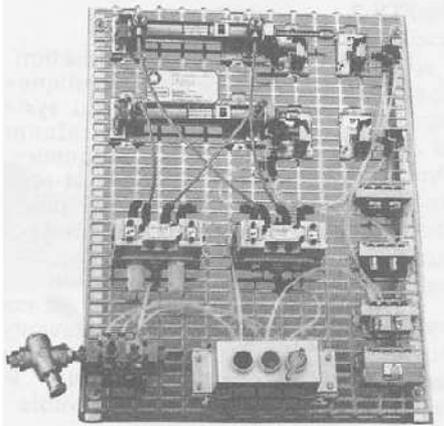
7.3- Les parties commandes.

Eléments qui traitent les informations et gèrent le fonctionnement du système.



La réalisation matérielle des PC peut être effectuée :

en logique câblée si la commande est pneumatique :
association de composants d'automatisme



en logique programmée si la commande est électrique :
Automate programmable industriel (API), ordinateur, carte électronique.



Automates programmables Industriels

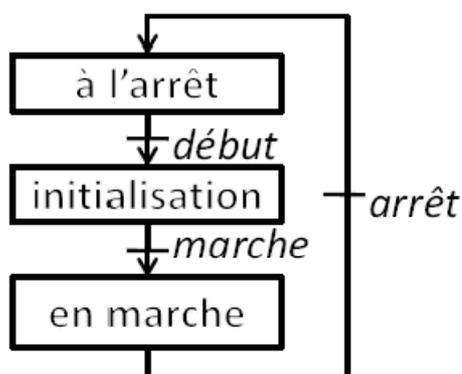
Chapitre 2 : Synthèse des automatismes décrits par GRAFCET

1. Introduction

Le Grafcet est un Système Dynamique à Evénements Discrets

- Système **dynamique** : dont l'état dépend des entrées passées et présentes (en opposition aux systèmes statiques)
- Système à **événements discrets** :
 - l'état du système est discret : à valeur dans un ensemble fini (en marche, à l'arrêt, etc.)
 - le système répond à des événements (capteur de fin de course, mise en marche, arrêt d'urgence, etc.) (en opposition aux systèmes (à états) continus)

Exemple:

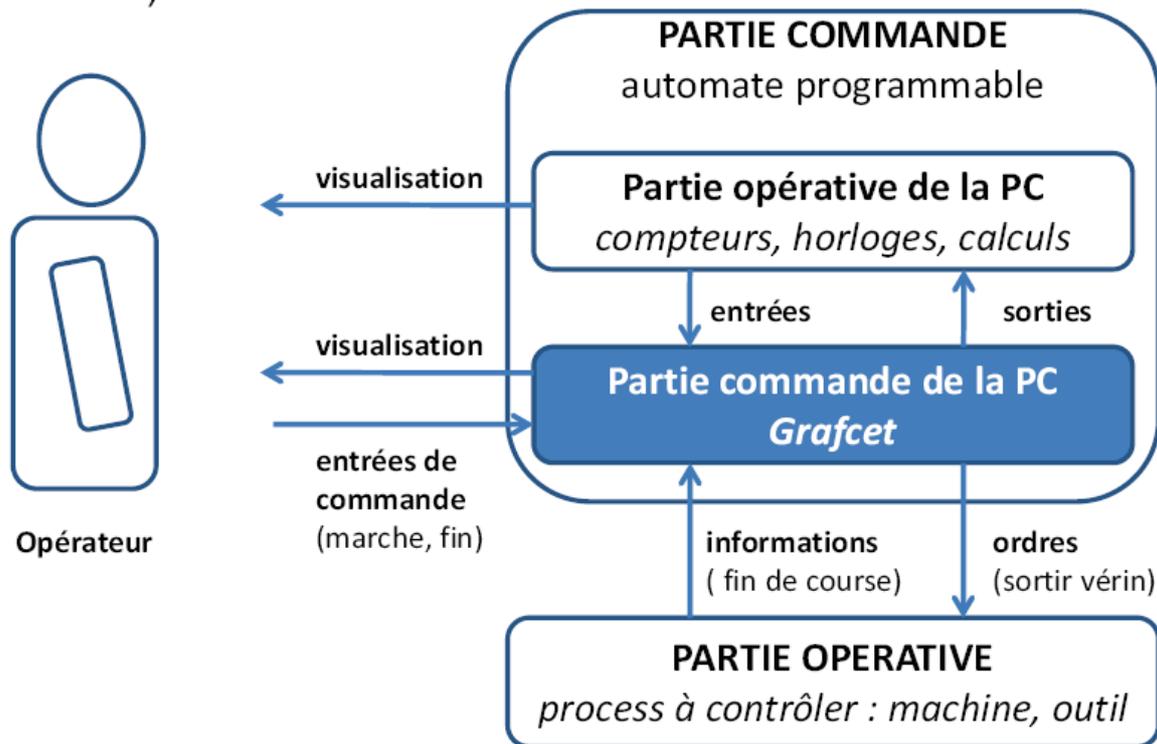


- 3 états : {à l'arrêt, initialisation, en marche}

- 3 événements : {début, marche, arrêt}

- l'occurrence de l'événement *marche* n'a pas le même effet suivant l'état du système

Initialement (1977), le Grafcet servait à la description des Automates Programmables Industriels (API), aujourd'hui il sert aussi à leur programmation



Chaque processus industriel de transformation ou de fabrication est constitué d'un ensemble de machines qui associé réalise le travail voulu.

Chaque machine ou partie opérative, comprend un ensemble de moteurs, vérins, vannes et autres dispositifs qui lui permet de fonctionner: ce sont les actionneurs.

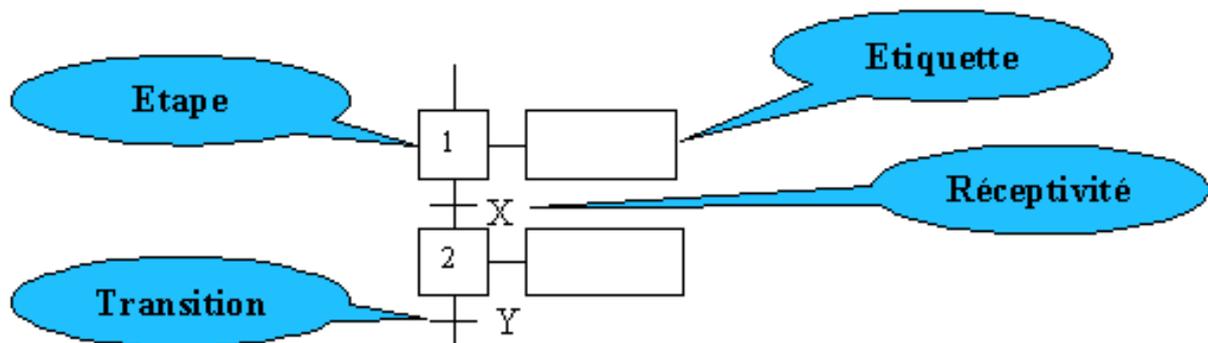
Ils sont pilotés par un automate ou partie commande. Cette partie commande élabore des ordres transmis aux actionneurs à partir d'informations fournies par la machine au moyen d'interrupteurs de position, de thermostats et autres dispositifs appelés capteurs.

Entre la partie commande et l'homme se trouve la partie dialogue. Elle permet à l'utilisateur de transmettre des informations au moyen de dispositifs adaptés (boutons poussoirs, commutateurs, etc...). De même la partie dialogue retourne vers l'homme des informations sur l'état du système sous des formes compréhensible par lui (voyant, afficheur, cadran, etc...).

2. Définition

Le GRAFCET (GRAPhe Fonctionnel de Commande des étapes et Transitions) est l'outil de représentation graphique d'un cahier des charges. Il a été proposé par l'ADEPA (en 1977 et normalisé en 1982 par la NF C03-190).

Le GRAFCET est une représentation alternée d'**étapes** et de **transitions**. Une seule transition doit séparer deux étapes.



Un grafcet est un graphe orienté biparti défini par $G=(E,T,A,M_0)$

- des étapes E
- des transitions T
- des arcs A
- un marquage initial M_0

ETAPE :

Un grafcet comporte un nombre fini d'étapes. Une **étape** se trouve entre 2 transitions, elle est représentée par un carrée et représente l'état du système

On définit un **vecteur d'état booléen** X :

$X_i = 0$, si l'étape i est inactive

$X_i = 1$, si l'étape i est active

À une étape peut être associée une action : l'action est effectuée quand l'étape est active, elle est représentée par un rectangle à droite de l'étape

TRANSITION :

Une transition se trouve entre 2 étapes et :

- est représentée par un trait horizontal
- est validée si l'étape amont est active
- représente les possibilités d'évolution du système

Aux transitions sont associées des réceptivités, notées R_i , dépendant:

- de variables internes ou externes
- d'événements internes ou externes
- de combinaisons des 2

ARC :

Un grafcet comporte un nombre fini d'arcs

Un arc est représenté par :

- un trait vertical de haut en bas
- une flèche s'il va de bas en haut

Un arc est orienté :

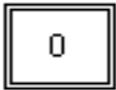
- d'une étape vers une transition
- d'une transition vers une étape
- d'une étape vers plusieurs transitions

MARQUAGE INITIAL M_0 :

Le marquage initial : ensemble fini non vide d'étapes actives à l'instant initial, représentées par des doubles rectangles.

3. Règles d'évolution

Règle N°1 : situation initiale



Cette représentation indique que l'étape est initialement activée (à la mise sous tension de la partie commande). La situation initiale, choisie par le concepteur, est la **situation à l'instant initial**.

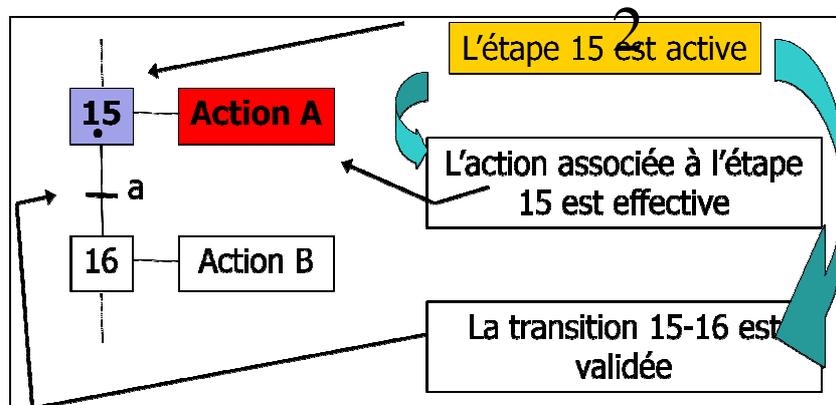
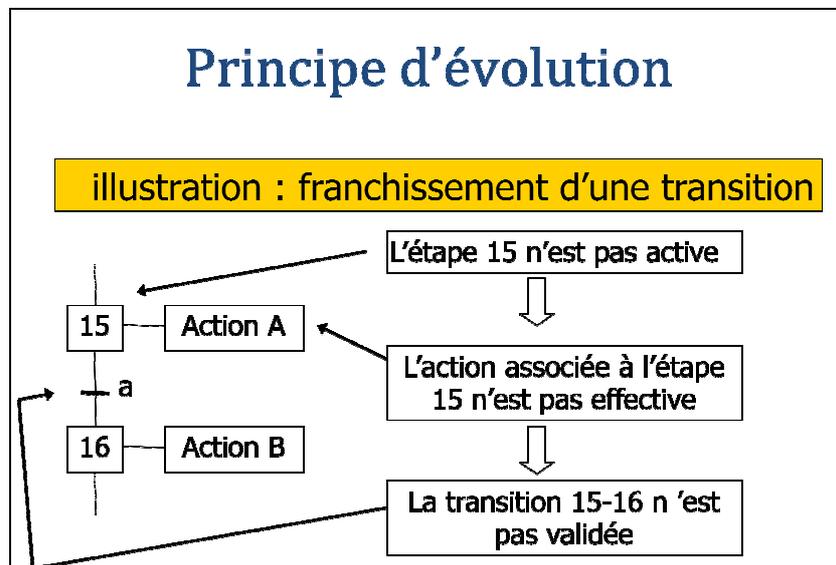
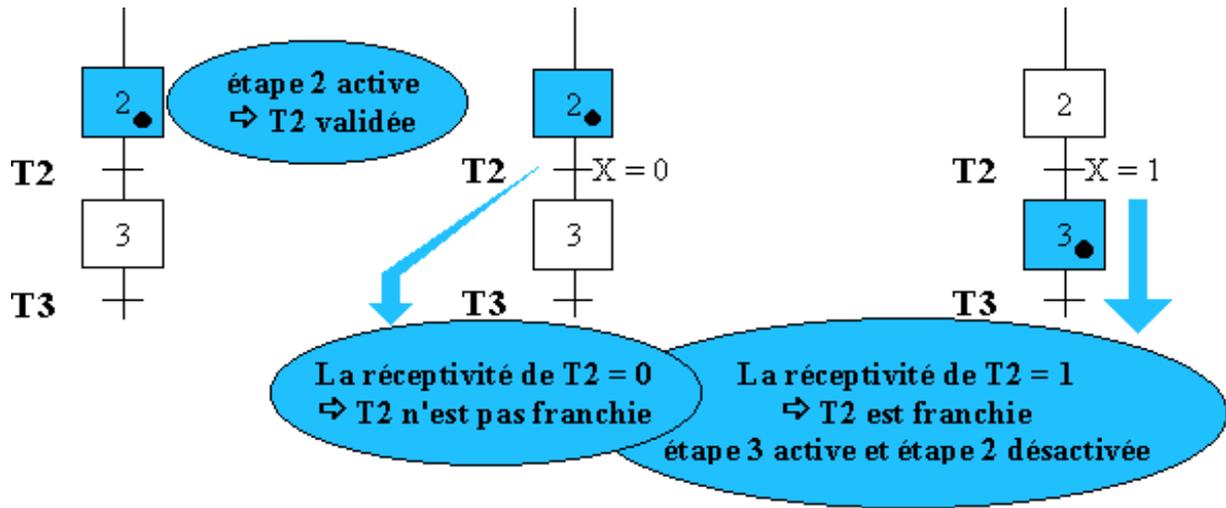
Règle N°2 : franchissement d'une transition

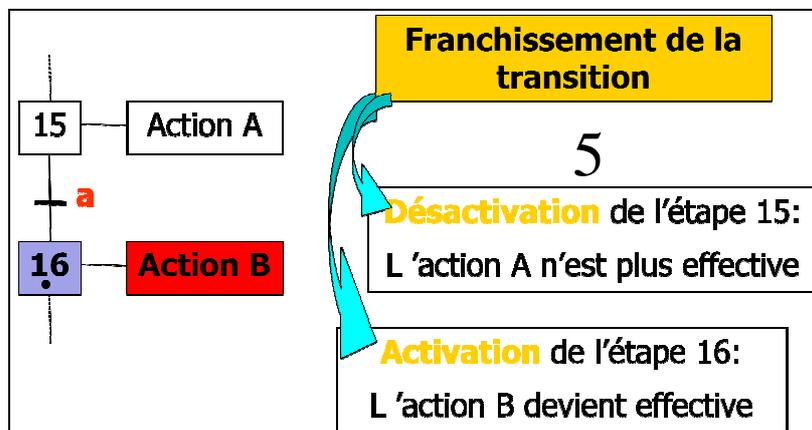
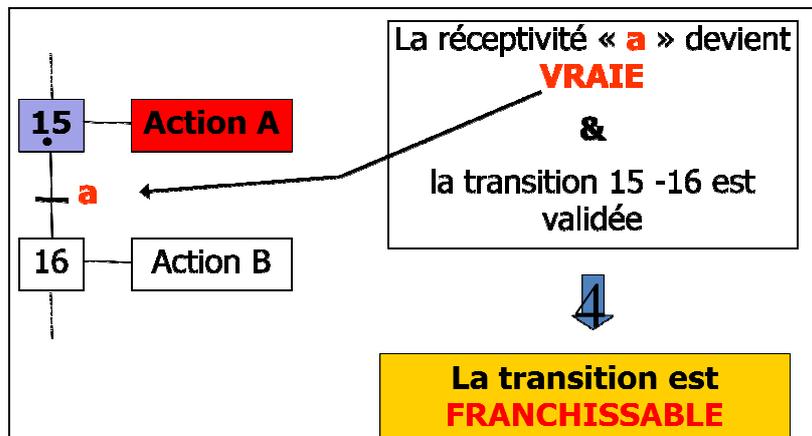
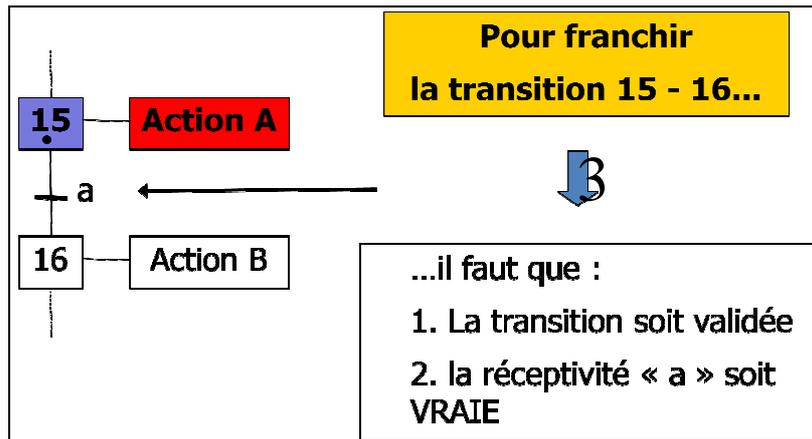
Une transition est franchie lorsque l'étape associée est active et la réceptivité associée à cette transition est vraie.

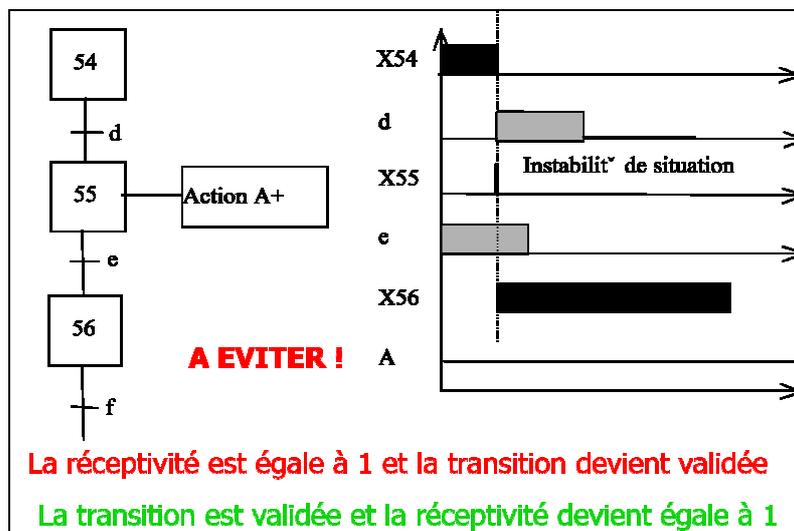
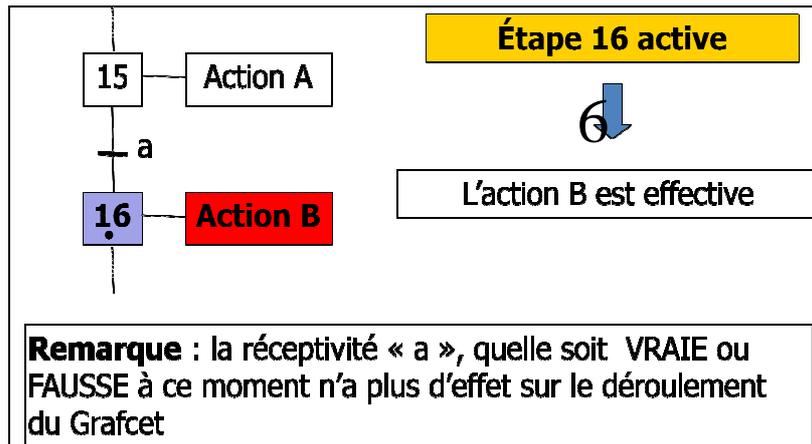
Règle N°3 : Evolution des étapes actives

Le franchissement d'une transition provoque simultanément :

- la **désactivation** de toutes les étapes immédiatement précédentes reliées à cette transition,
- l'**activation** de toutes les étapes immédiatement suivantes reliées à cette transition.







Règle N°4 : Transitions simultanées

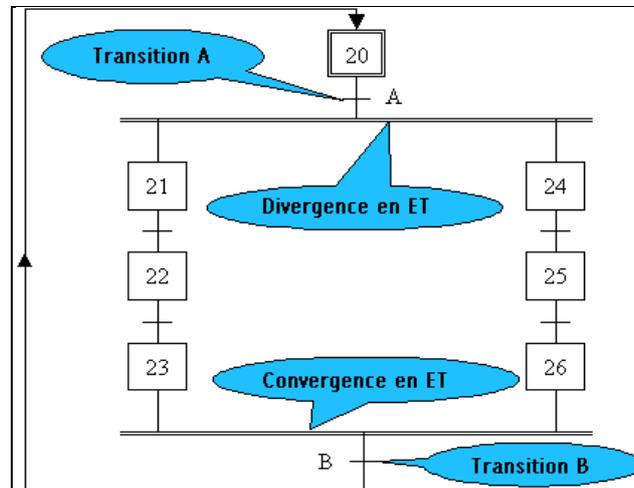
Plusieurs transitions simultanément franchissables sont simultanément franchies.

Règle N°5 : Activation et désactivation simultanées

Une étape à la fois activée et désactivée reste active.

4. Structures de base

4.1- Divergence et convergence en ET (séquences simultanées)



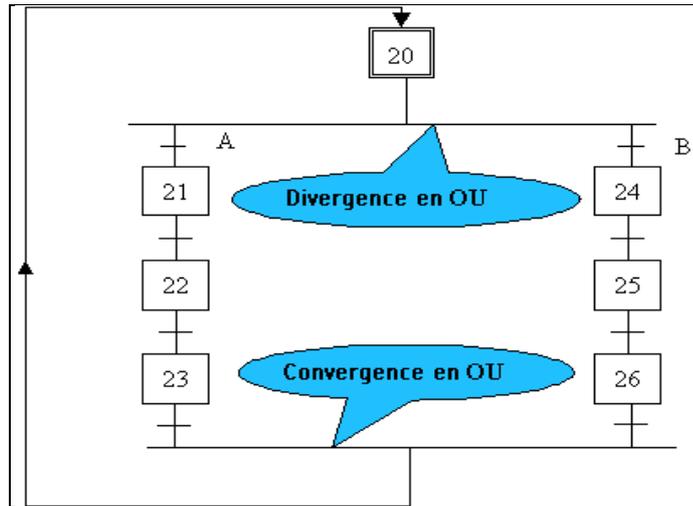
Divergence en ET : lorsque la transition A est franchie, les étapes 21 et 24 sont actives.

Convergence en ET : la transition B sera validée lorsque les étapes 23 et 26 seront actives. Si la réceptivité associée à cette transition est vraie, alors celle-ci est franchie.

REMARQUE :

- Après une divergence en ET, on trouve une convergence en ET.
- Le nombre de branches parallèles peut-être supérieur à 2.
- La réceptivité associée à la convergence peut-être de la forme = 1. Dans ce cas la transition est franchie dès qu'elle est active.

4.2- Divergence et convergence en OU (aiguillage)



Divergence en OU : l'évolution du système vers une branche dépend des réceptivités A et B associées aux transitions.

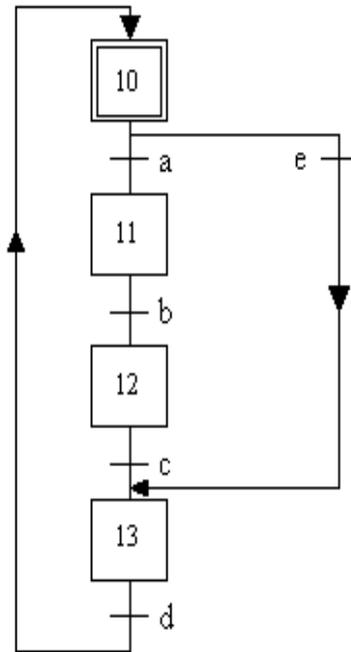
Convergence en OU : après l'évolution dans une branche, il y a convergence vers une étape commune.

REMARQUE :

- A et B ne peuvent être vrais simultanément (conflit).
- Après une divergence en OU, on trouve une convergence en OU.
- Le nombre de branches peut-être supérieur à 2.
- La convergence de toutes les branches ne se fait pas obligatoirement au même endroit.

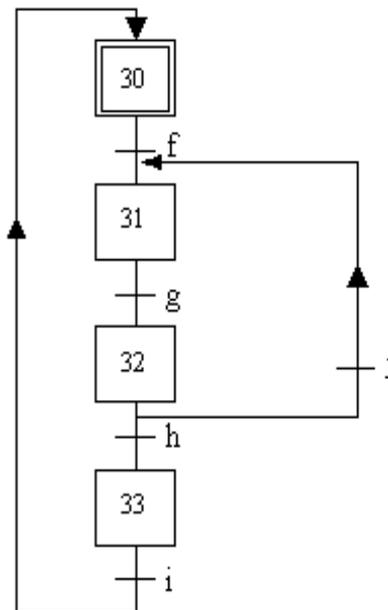
4.3- Saut en avant (saut de phase)

Le saut en avant permet de sauter une ou plusieurs étapes lorsque les actions à réaliser deviennent inutiles.



4.4- Saut en avant (saut de phase)

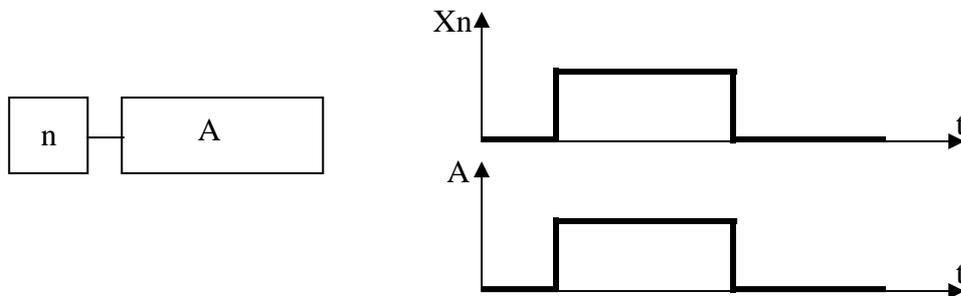
Le saut en arrière permet de reprendre une séquence lorsque les actions à réaliser sont répétitives.



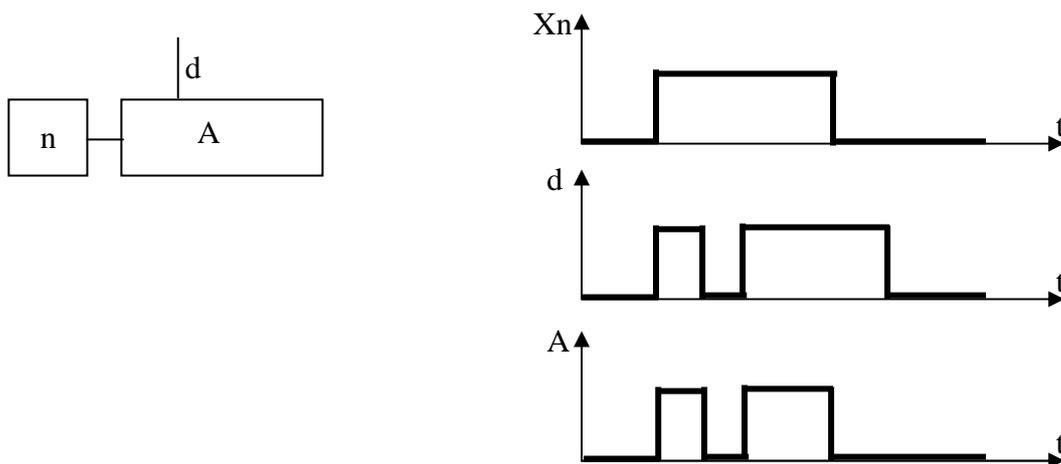
5. Classification des actions

Une action est un élément du langage GRAFCET associée à une étape, l'action indique le comportement d'une variable de sortie

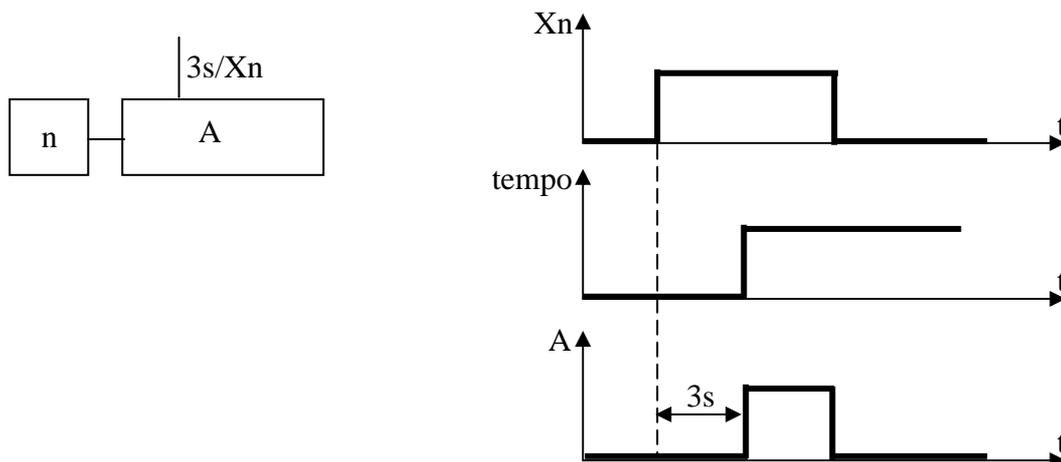
Action continue : Une action continue est nécessairement associée à une étape. Plusieurs actions peuvent être associées à une même étape.



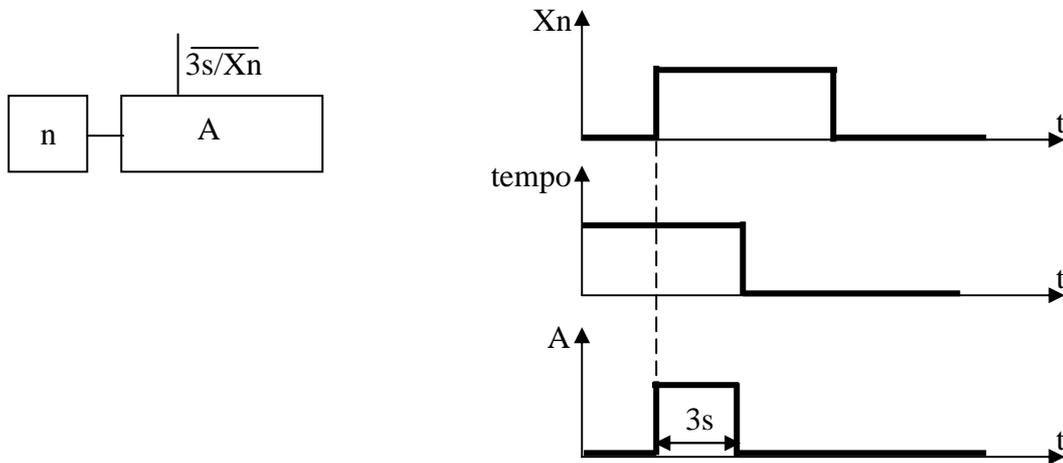
Action conditionnelle : Une action continue est nécessairement associée à une étape. Plusieurs actions peuvent être associées à une même étape.



Action retardée : L'action retardée est une action continue dont la condition d'assignation n'est vraie qu'après une durée t_1 spécifiée depuis l'activation de l'étape associée n , dans le but de retarder l'assignation à la valeur vraie de la sortie correspondante.

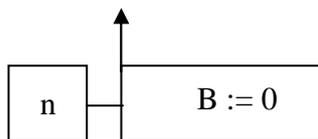


Action limitée dans le temps : L'action limitée dans le temps est une action continue dont la condition d'assignation n'est vraie que pendant une durée t_1 spécifiée depuis l'activation de l'étape n à laquelle elle est associée, dans le but de limiter la durée d'assignation à la valeur vraie de la sortie correspondante.



Action mémorisée : Une action mémorisée possède un libellé qui décrit comment la variable de sortie est affectée à une valeur déterminée selon la règle d'affectation

Action à l'activation : Une action à l'activation est une action mémorisée associée à l'ensemble des événements internes qui ont chacun pour conséquence l'activation de l'étape liée à cette action.



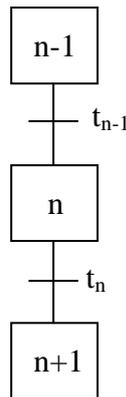
La variable booléenne B est affectée à la valeur 0 lorsque l'un des événements, conduisant à l'activation de l'étape n , se produit.

Chapitre 3 : Mise en œuvre du GRAFCET

1. Mise en équation du GRAFCET

1.1-Objectif

Soit la partie du grafcet représenté par la figure ci-contre.



Pour décrire l'activité de l'étape n, nous utilisons la notation suivante :

$X_n = 1$: si l'étape n est active

$X_n = 0$: si l'étape n est inactive

La réceptivité t_n étant une variable binaire, a pour valeur

$t_n = 1$: si la réceptivité est vraie

$t_n = 0$: si la réceptivité est fausse

Les règles d'évolution du GRAFCET sont le point de départ des équations logiques

1.2.- Mise en équation d'une étape

❖ 2^{ème} règle :

Une transition est soit validée, soit non validée. Elle 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 est VRAIE

La traduction de cette règle donne la **Condition d'Activation** de l'étape n

$$CA X_n = X_{n-1} \cdot t_{n-1}$$

❖ 3^{ème} règle :

Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes
 La traduction de cette règle donne la **Condition de désactivation** de l'étape n

$$CDX_n = X_n \cdot t_n = X_{n+1}$$

❖ Effet mémoire :

Si la CA et la CD de l'étape n son fausses, l'étape n reste dans son état. C'est ce qu'on appelle effet mémoire. C'est-à-dire que l'état de X_n à l'instant $t+\delta t$ dépend de l'état précédent de X_n à l'instant t

D'après ces trois points précédents on peut écrire : $X_n = f(CAX_n, CDX_n, X_n)$

Il est alors possible d'écrire la table de vérité de l'activité de l'étape n

$X_n(T)$	CAX_n	$CDX_n(T)$	$X_n(T+\delta T)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Le tableau de Karnaugh associé est le suivant

		$CAX_n \ CDX_n$			
		00	01	11	10
X_n	0	0	0	1	1
	1	1	0	1	1

L'équation de X_n :

$$X_n = CA X_n + \overline{CDX_n} \cdot X_n$$

$$X_n = X_{n-1} \cdot \overline{t_{n-1}} + X_{n+1} \cdot X_n$$

1.3- Gestion des modes Marche/Arrêt et des arrêts d'urgences

A l'initialisation du GRAFCET, toutes les étapes autres que les étapes initiales sont désactivées. Seules sont activées les étapes initiales.

Soit la variable Init telle que :

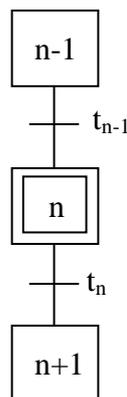
- Init = 1 : initialisation du GRAFCET mode ARRET
- Init = 0 : déroulement du cycle mode MARCHE

Soit les variables Arrêt d'urgence (AUDUR et AUDOUX) telles que :

- AUDUR = 1 : désactivation de toutes les étapes
- AUDOUX = 1 : désactivation des actions les étapes restant actives

Généralisation :

❖ Equation d'une étape initiale



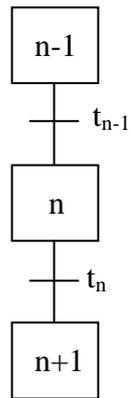
L'équation de X_n :

$$CAX_n = X_{n-1} \cdot \overline{t_{n-1}} + \overline{Init}$$

$$CDX_n = X_{n+1} \cdot \overline{Init}$$

$$X_n = (CA X_n + \overline{CDX_n} \cdot X_n + \overline{Init}) \cdot \overline{AUDUR}$$

❖ Equation d'une étape non initiale



L'équation de X_n :

$$CA X_n = X_{n-1} \cdot t_{n-1} \overline{Init}$$

$$CD X_n = X_{n+1} + Init$$

$$X_n = (CA X_n + CD X_n \cdot X_n) \overline{Init} \cdot \overline{AUDUR}$$

Chapitre 4 : Matérialisation des automatismes décrits par grafcet à l'aide de séquenceurs.

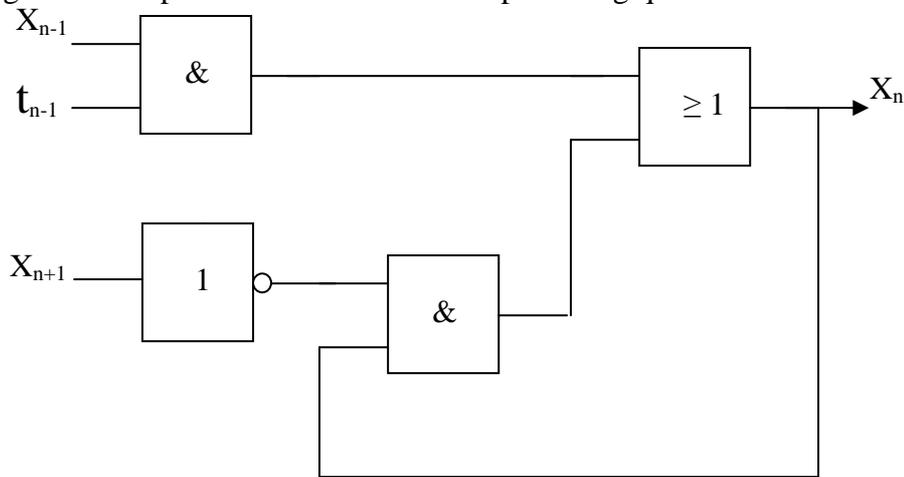
1. Réalisation par câblage

1.1- Elément de la logique combinatoire (portes logiques)

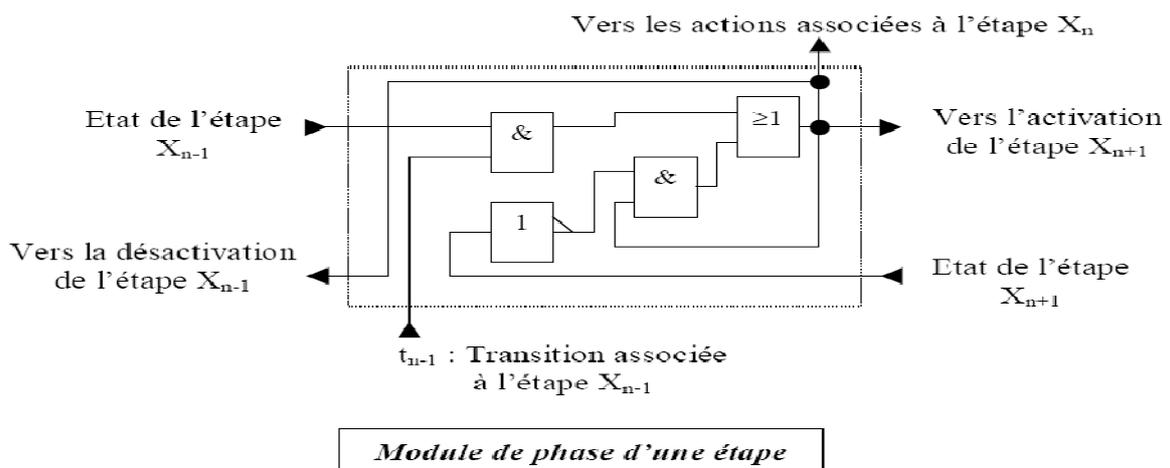
L'équation d'une étape :

$$X_n = X_{n-1} \cdot t_{n-1} + \overline{X_{n+1}} \cdot X_n$$

Le câblage d'une étape est réalisé à l'aide de 4 portes logiques.

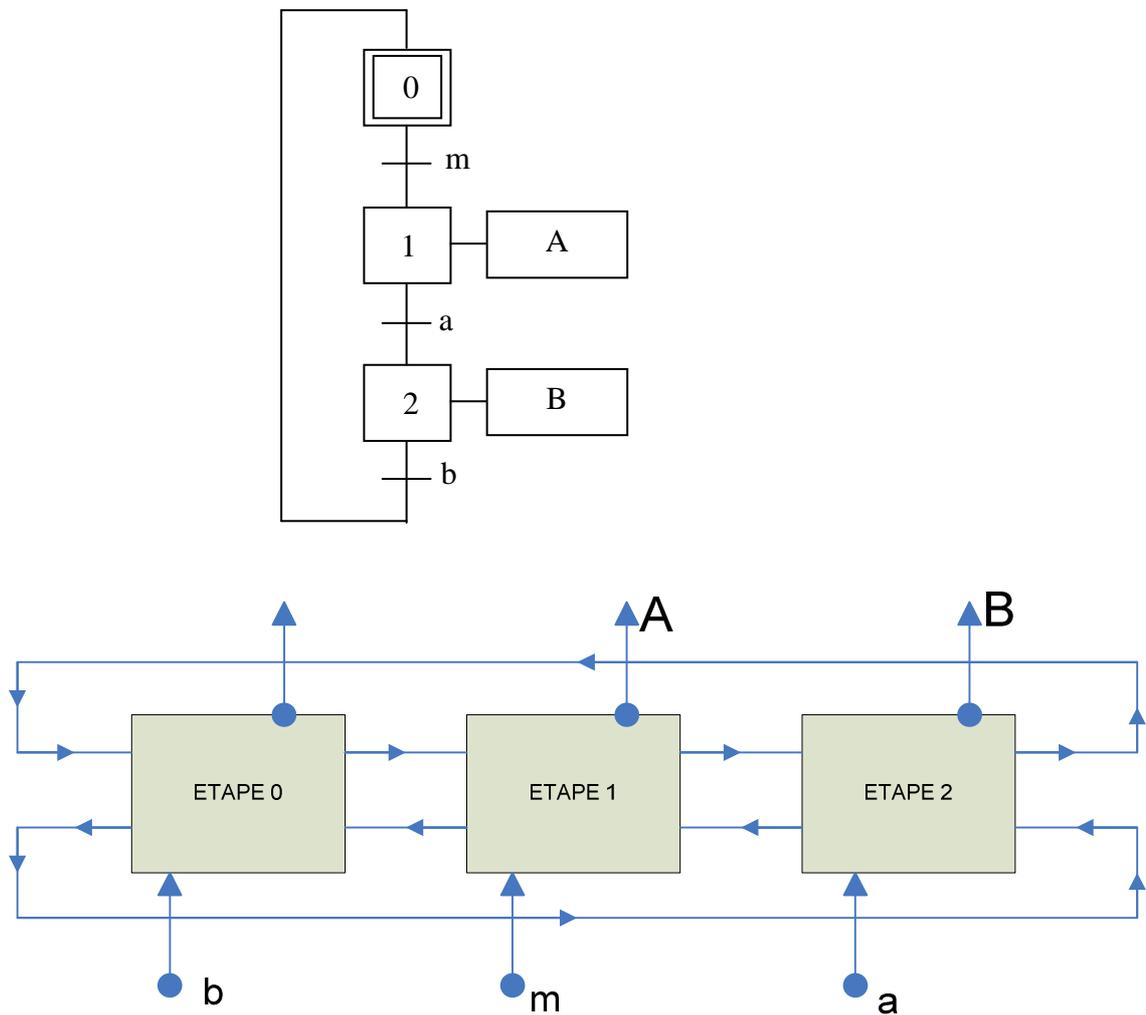


Une étape de GRAFCET se symbolise alors sous forme d'un module de phase :



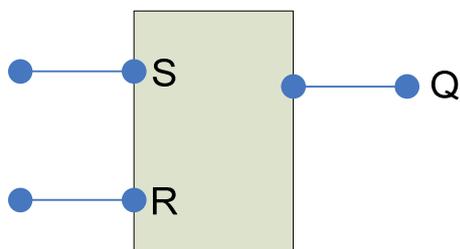
Exemple : Cas d'un GRAFCET à séquence unique

Chaque étape du GRAFCET Sera câblée comme le module de phase décrit précédemment.
 On réalise alors un séquenceur électronique à base des portes logiques.



1.2- Elément de la logique Séquentielle (bascule RS)

- Rappel

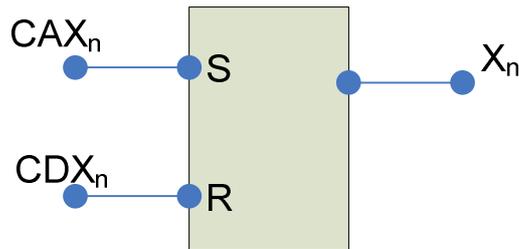


R	S	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	-

- **Application au GRAFCET**

La **Condition d'activation** est câblée sur l'entrée **SET** de la bascule

La **Condition de désactivation** est câblée sur l'entrée **RESET** de la bascule

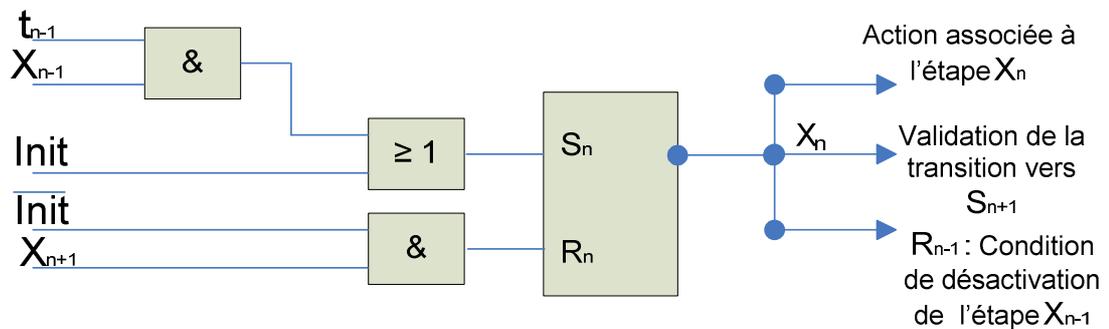


On ne tient pas compte dans le câblage suivant des arrêts d'urgence (AUDUR et AUDOUX)

Câblage d'une étape initiale :

$$CAX_n = X_{n-1} \cdot t_{n-1} + \overline{Init}$$

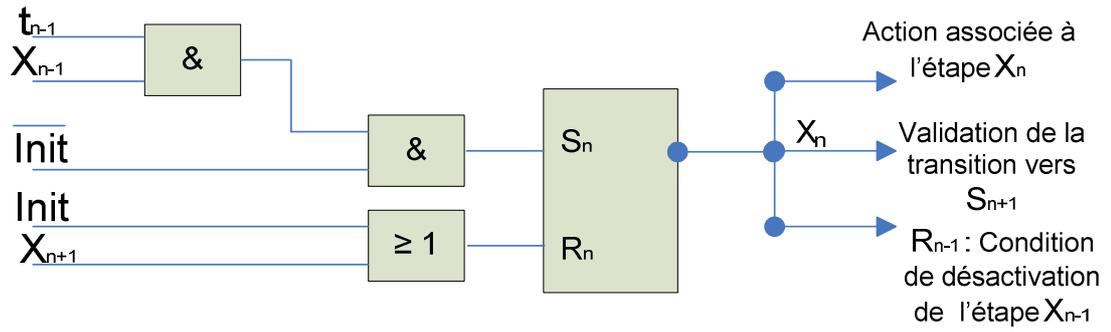
$$CDX_n = X_{n+1} \cdot \overline{Init}$$



Câblage d'une étape non initiale :

$$CAX_n = X_{n-1} \cdot t_{n-1} \cdot \overline{Init}$$

$$CDX_n = X_{n+1} + Init$$



Exemple : GRAFCET à séquence unique

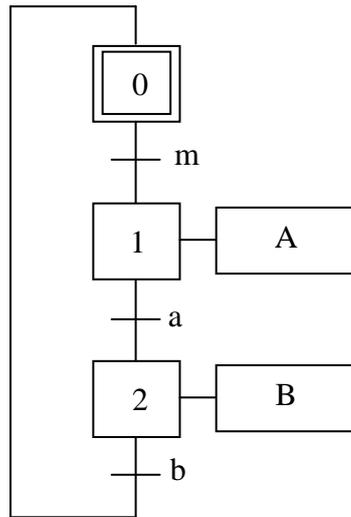
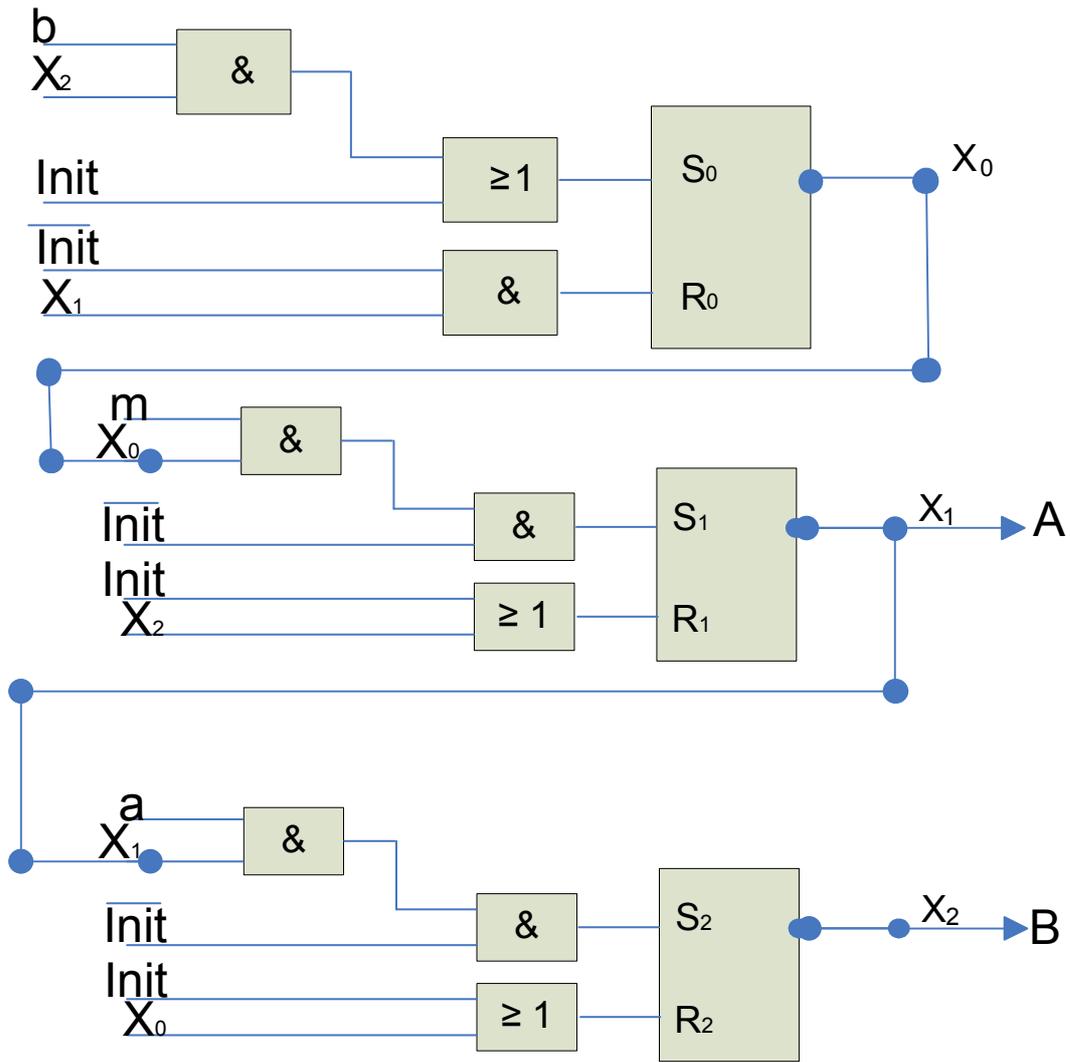


Tableau des conditions d'activation et de désactivation des étapes

X_n	CAX_n	CDX_n
0	$X_2 \cdot b + Init$	$X_1 \cdot \overline{Init}$
1	$X_0 \cdot m \cdot \overline{Init}$	$X_2 + Init$
2	$X_1 \cdot a \cdot \overline{Init}$	$X_0 + Init$



2. Séquenceur pneumatique

2.1- Introduction

Le séquenceur pneumatique est une association linéaire de modules d'étape.

- Description du séquenceur pneumatique :

Un **module d'étape** correspond à une étape du Grafcet et son fonctionnement traduit les règles d'évolution du Grafcet.

Un **mémoire pneumatique bistable** est l'élément logique de base de la réalisation d'un module d'étape.

• **Principe de base du fonctionnement d'un module d'étape :**

Le module d'étape est construit à partir d'une mémoire bistable et de portes logiques pour respecter les règles d'évolution du GRAFCET. Pour que le fonctionnement du module d'étape soit **conforme aux règles d'évolution** d'une étape du GRAFCET, il faut tenir compte des conditions suivantes lors de sa réalisation :

- L'action associée à une étape a lieu si l'étape est **active**.
- Une étape est considérée activée si l'étape précédente est active et que la réceptivité associée à la transition est **logiquement "VRAIE"**.
- Une étape doit être désactivée si l'étape suivante est **active**.

Une mémoire pneumatique bistable dispose de deux signaux d'entrée (signal de mise à 1 et signal de mise à 0) et de deux sorties complémentaires.

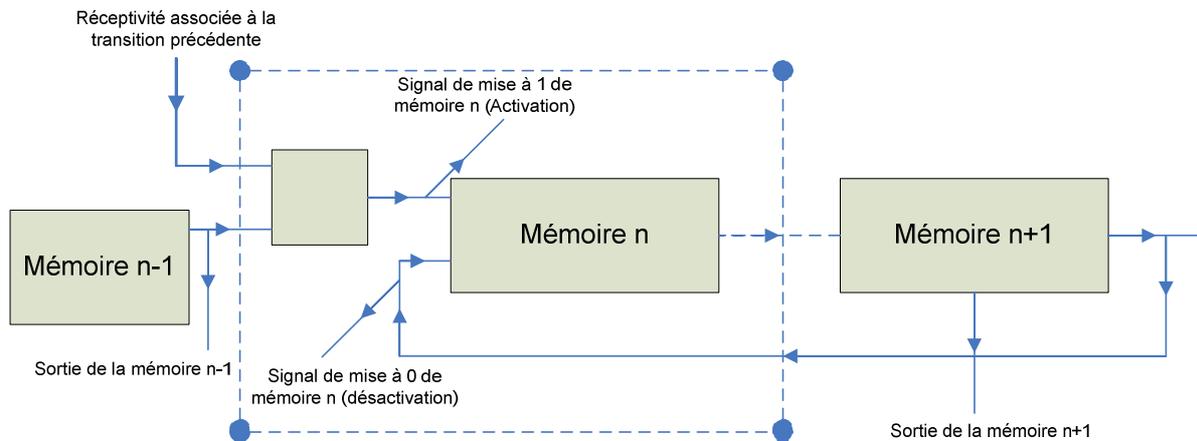
L'activation de la mémoire se fait quand le signal de mise à 1 est à l'état logique 1 alors que sa désactivation est possible quand le signal de mise à 0 est à l'état logique 1.

Ainsi, les conditions de fonctionnement du module d'étape qu'il faut respecter peuvent se traduire de la façon suivante :

- **La mémoire** délivrera un signal de sortie si elle est activée. Ce signal servira à commander l'action.
- **Le signal de mise à 1** de la mémoire sera constitué par la multiplication logique du signal de l'étape précédente et la réceptivité associée à la transition précédente. De cette manière, l'activation de la mémoire se fera à la seule condition que l'étape précédente soit active et que la réceptivité associée à la transition précédente soit logiquement "VRAIE" conformément aux règles d'évolution du Grafcet.
- **La mémoire doit être désactivée** lorsque la mémoire suivante devient active. Ceci peut être réalisé en connectant la sortie de la mémoire suivante au signal de mise à 0 de la mémoire. De cette manière, la mémoire est désactivée quand la mémoire suivante est activée.

La figure suivante présente la description schématique de la réalisation d'un module d'étape.

Description schématique de la réalisation d'un module d'étape :



Sur cette figure, vous remarquez que **la mémoire "n - 1"** représente l'étape précédente alors que **la mémoire "n + 1"** désigne l'étape suivante de l'étape "n".

L'activation de la mémoire du module d'étape "n" ne sera possible que lorsque le signal à la sortie de la mémoire "n - 1" est à l'état **logique 1** et que la réceptivité de la transition entre l'étape "n - 1" et l'étape "n" est logiquement **"VRAIE"**.

C'est une **cellule logique "ET"** qui réalise cette condition logique. Cette cellule admet donc comme entrée, la sortie de la mémoire "n - 1" et un signal de capteur qui indique si la réceptivité associée à la transition entre l'étape "n - 1" et "n" est logiquement **"VRAIE"**.

La sortie de cette cellule logique "ET" constitue le signal de mise à 1 de la mémoire "n". Le signal de mise à 0 de cette mémoire est constitué par la sortie de la mémoire "n + 1". Ainsi, quand la sortie de la mémoire "n + 1" sera active, la mémoire "n" doit être désactivée conformément aux règles d'évolution du Grafcet.

Les modules d'étape commerciaux peuvent présenter certaines propriétés particulières selon leur fabricant. Par exemple, le module d'étape de Parker permet une remise à 0 générale (RAZ). Cette **remise à 0** générale est assurée par un signal qui agit sur les entrées de mise à 0 de chacune des mémoires des modules d'étape.

Dans ce module d'étape, **une cellule logique "OU"** est ajoutée au schéma du principe de base. Cette cellule admet comme entrées le signal de remise à 0 général (RAZ) et la sortie de la mémoire suivante. De cette façon, la mémoire présente peut être désactivée quand la mémoire suivante est activée ou quand le signal de remise à 0 général (RAZ) est à l'état logique 1. Le schéma de ce module d'étape est donné à la figure suivante.

2.2- Montage d'un séquenceur pneumatique à partir de modules d'étape

La réalisation pratique d'un séquenceur pneumatique est possible par la mise en cascade de plusieurs modules d'étape. On réalise ainsi **une chaîne de modules d'étape**.

Chaque module d'étape représente **une étape du Grafset** à réaliser.

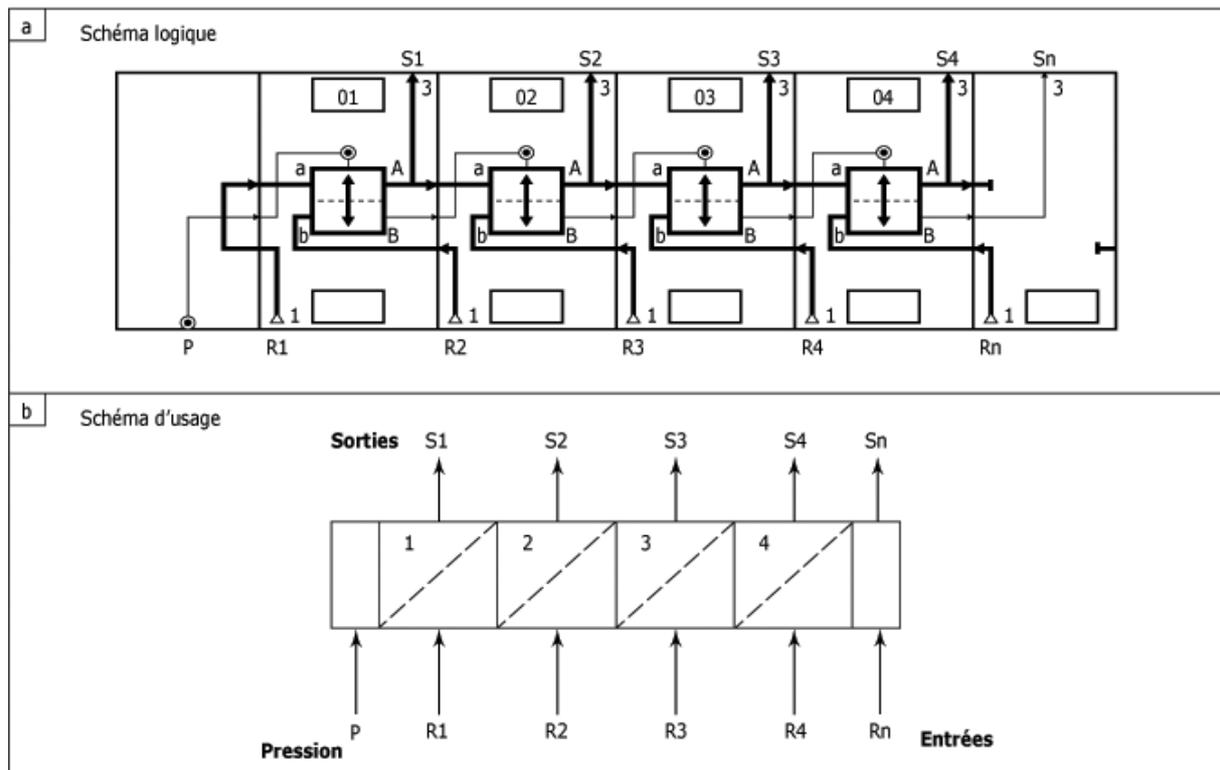
Ainsi, pour la réalisation d'un **Grafset à quatre étapes**, on aura besoin d'un séquenceur à **quatre modules d'étape**. Les modules d'étape sont montés sur des plaques d'embase alors que deux plaques d'extrémité doivent être placées au début de la chaîne et à sa fin.

La plaque d'extrémité du début de la chaîne sert généralement à alimenter la chaîne en pression commune et à diriger le signal du début du cycle vers le premier module d'étape.

La plaque de la fin de la chaîne capte le signal de la fin du cycle et le retourne vers le signal de début du cycle.

Les fabricants de séquenceurs pneumatiques ont mis en place un standard de **symbolisation logique** des séquenceurs. Les séquenceurs commerciaux présentent des particularités différentes selon le fabricant. La figure suivante montre le schéma logique et le schéma d'usage d'un séquenceur Climax (système Polylog) sans remise à zéro générale à quatre modules d'étape.

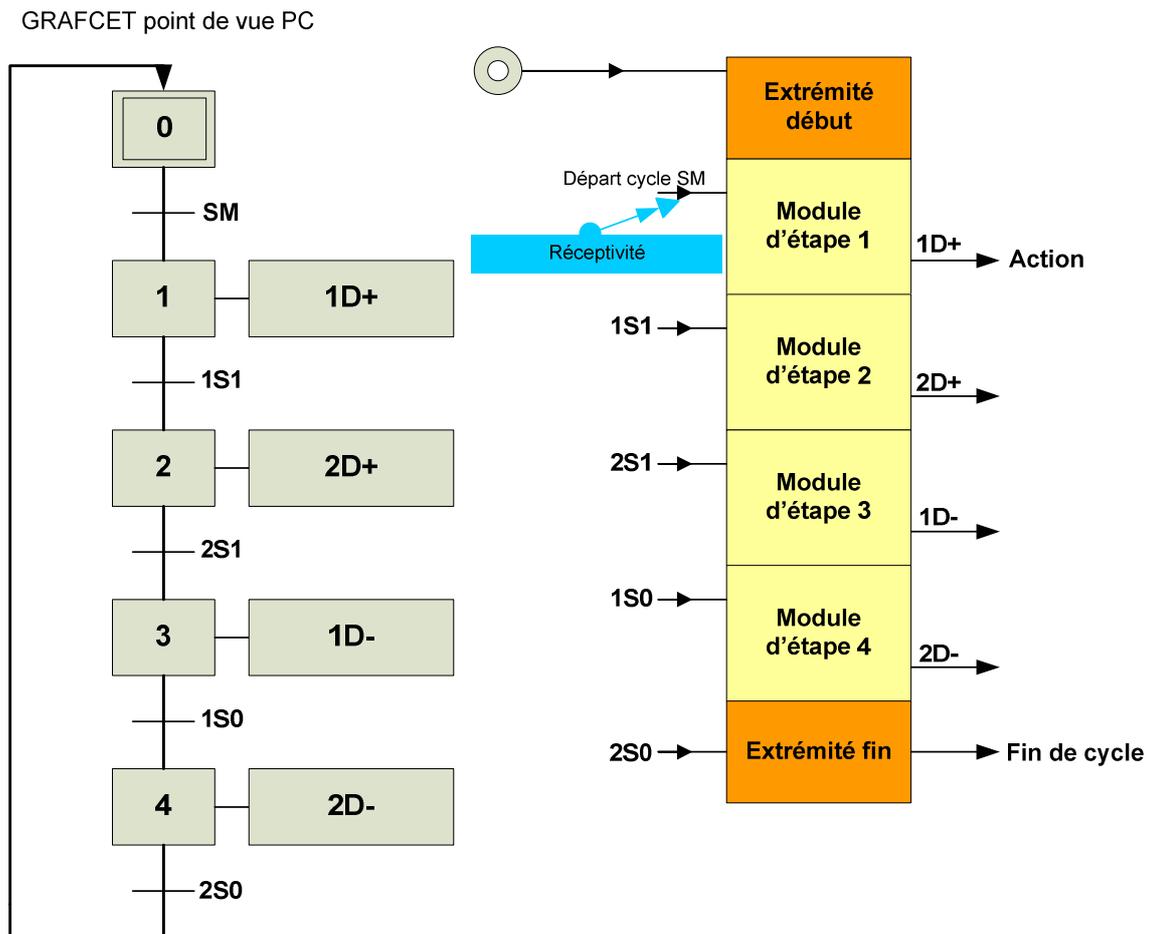
Schéma logique et schéma d'usage d'un séquenceur Climax (système Polylog) à quatre modules d'étape :



Exemple

Le séquenceur pneumatique est l'élément idéal pour la réalisation pratique des Grafcet à séquence unique. Le séquenceur est un ensemble modulaire, facile à installer et il permet de visualiser l'évolution des étapes actives du Grafcet. A partir de l'exemple d'un Grafcet à séquence unique, vous allez comprendre le schéma de branchement nécessaire à la réalisation de l'automatisme.

Considérez le Grafcet des mouvements (c'est une autre représentation du Grafcet point de vue partie opérative)

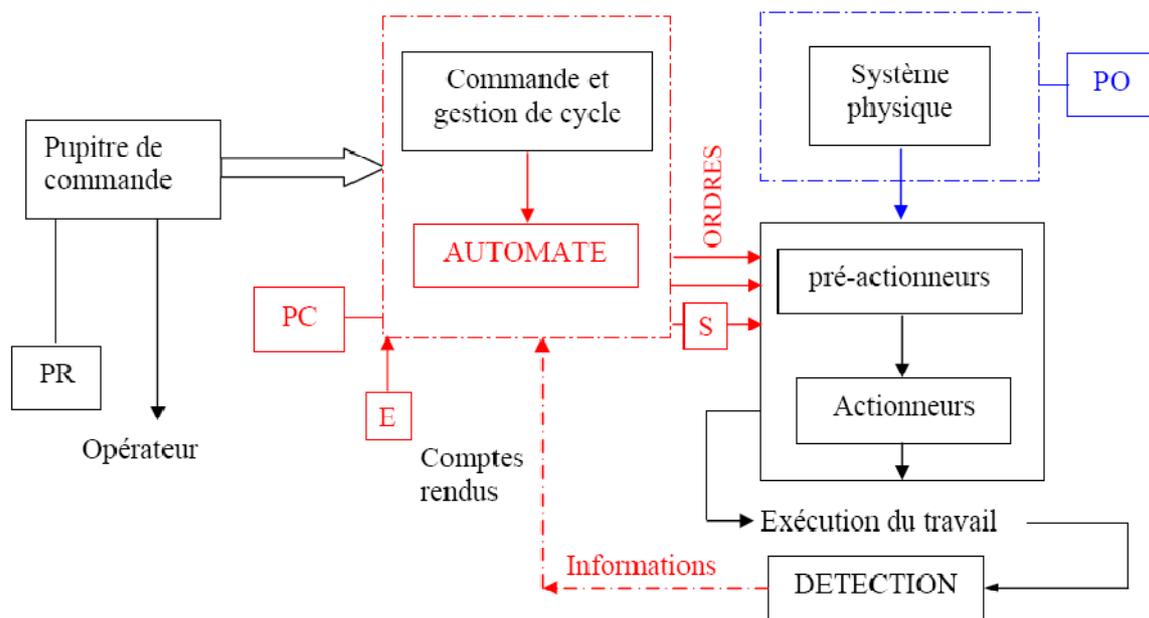


Chapitre 5 : Automates Programmables Industriels (API)

1. Introduction

Les **Automates Programmables Industriels (API)** sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

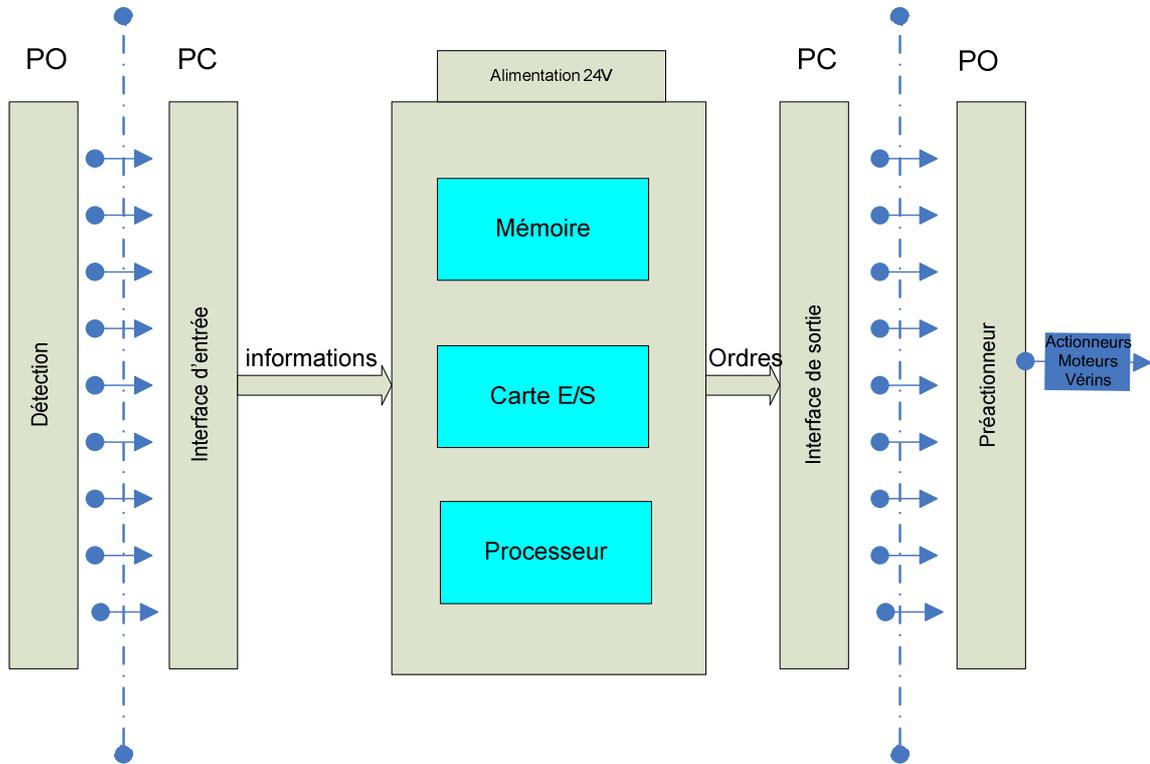
Un **Automate Programmable Industriel (API)** est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels. Un automate programmable est adaptable à un maximum d'application, d'un point de vue traitement, composants, langage. C'est pour cela qu'il est de construction modulaire.



2. Structure d'un API

Cet ensemble électronique gère et assure la commande d'un système automatisé. Il se compose de plusieurs parties et notamment d'une mémoire programmable dans laquelle l'opérateur écrit, dans un langage propre à l'automate, des directives concernant le déroulement du processus à automatiser. Son rôle consiste donc à fournir des ordres à la partie opérative en vue d'exécuter un travail précis comme par exemple la sortie ou la rentrée d'une

tige de vérin, l'ouverture ou la fermeture d'une vanne. La partie opérative lui donnera en retour des informations relatives à l'exécution dudit travail.



Les API comportent quatre parties principales :

- Une mémoire
- Un processeur
- Des interfaces d'Entrées/Sorties
- Une alimentation (240 Vac ! 24 Vcc).

Ces quatre parties sont reliées entre elles par des bus (ensemble câblé autorisant le passage de l'information entre ces 4 secteurs de l'API). Ces quatre parties réunies forment un ensemble compact appelé automate.

La structure interne d'un **automate programmable industriel (API)** est assez voisine de celle d'un système informatique simple, L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

Deux types de mémoire cohabitent :

- **La mémoire Programme** où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte)

- **La mémoire de données** utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle fait partie du système entrées-sorties. Elle fige les valeurs (0 ou 1) présentes sur les lignes d'entrées, à chaque prise en compte cyclique de celle-ci, elle mémorise les valeurs calculées à placer sur les sorties.

2.1- Description des éléments d'un API

a- Le processeur :

Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'E/S et d'autre part à gérer les instructions du programme.

b- La mémoire :

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système que sont le terminal de programmation (PC ou console) et le processeur, qui lui gère et exécute le programme. Elle reçoit également des informations en provenance des capteurs. Il existe dans les automates plusieurs types de mémoires qui remplissent des fonctions différentes :

- ✓ La conception et l'élaboration du programme font appel à la RAM et l'EEPROM;
- ✓ La conservation du programme pendant l'exécution de celui-ci fait appel à une EPROM.

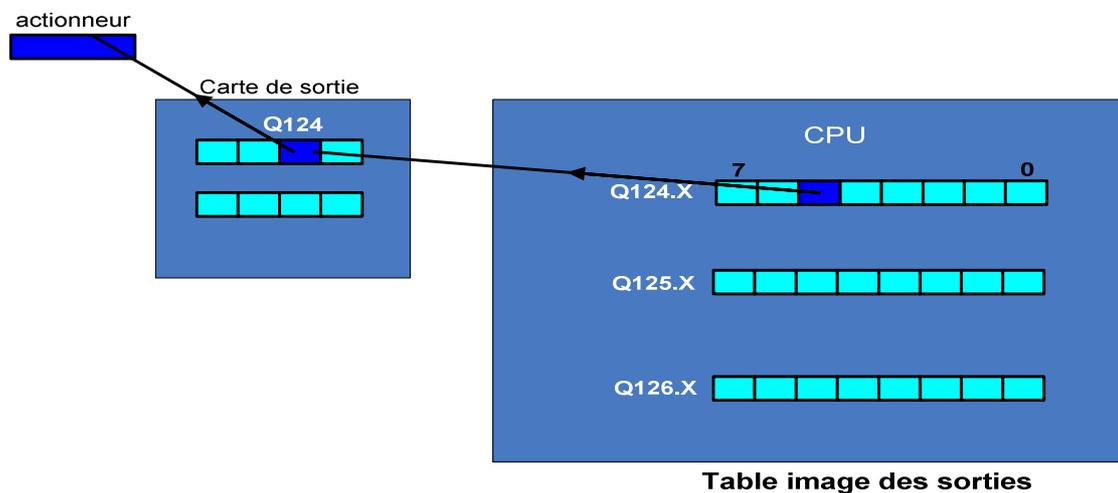
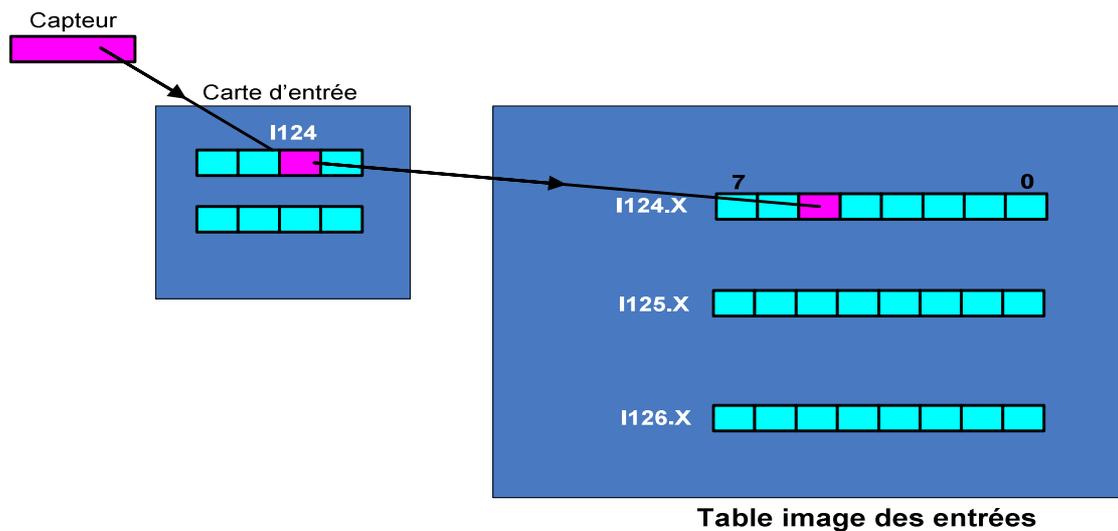
c- L'alimentation :

Tous les automates actuels utilisent un bloc d'alimentation alimenté en 240 Vac et délivrant une tension de 24 Vcc.

d- Les interfaces :

L'interface d'Entrées comporte des adresses d'entrée, une pour chaque capteur relié. L'interface de Sorties comporte des adresses de sorties, une pour chaque pré-actionneur. Le nombre d'E/S varie suivant le type d'automate.

Les cartes d'E/S ont une modularité de 8, 16 ou 32 voies. Elles admettent ou délivrent des tensions continues 0 - 24 Vcc.



Fonctionnement

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. Généralement les automates programmables industriels ont un fonctionnement cyclique

Le **microprocesseur** réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul... Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons **parallèles** appelées '**BUS**' qui véhiculent les informations sous forme binaire.. Lorsque le fonctionnement est dit synchrone par rapport aux entrées et aux sorties, le cycle de traitement commence par la prise en compte des entrées qui sont figées en mémoire pour tout le cycle.

2.2- Le choix de l'unité centrale

Le choix de l'unité centrale (UC) d'un API est conditionnée par son prix, ses performances ou son temps de réponse, ses possibilités de connexion à des réseaux, ses capacités de calcul et de la taille de sa mémoire.

a- Le prix de l'UC :

Le prix de l'UC varie en fonction des capacités qu'elle possède. Le prix intervient dans le choix de l'API, mais il reste très faible en comparaison avec l'ensemble du projet.

b- Le temps de réponse :

Le temps de réponse est fonction de la vitesse du microprocesseur, du temps d'accès à la RAM et de la taille du programme. Le constructeur donne généralement la valeur du temps de traitement d'un bit ou d'un mot. Par exemple la CPU 215 possède un temps de traitement d'un bit de 0.18 μ s et d'un mot de 0.78 μ s. Le temps de scrutation du programme se déduit par la multiplication du nombre de traitement de bits et de mots par le temps de traitement d'un bit et d'un mot

c- La connexion à des réseaux :

Les API modulaires de haute gamme ont la possibilité d'être connectés à des réseaux. Des cartes additionnelles, coupleurs de communication, assurent le dialogue avec d'autres équipements branchés à des réseaux PROFIBUS, ETHERNET

d- Les possibilités de calculs des UC :

Les API de bas de gamme intègrent des fonctions de calcul simples. Les API de haute gamme offrent la possibilité de faire des calculs complexes sur des variables réelles, de traiter des chaînes de caractères, d'utiliser des fonctions de régulation (PID), de dialoguer avec d'autres équipements

e- La taille mémoire des UC :

La mémoire des API est encore très chère. Elle doit être nécessaire et suffisante pour accepter le programme. En général, la taille mémoire, utilisée pour le programme utilisateur, évolue de 48KB à 512KB. Certains UC sont dotés de mémoire flash. Cette mémoire sauvegarde le programme. A la mise sous tension de l'API, le programme résidant dans la mémoire flash, est transféré dans la RAM de l'API. Elle peut servir à stocker des données de production.

2.3- Le temps de réponse d'un API

Le temps de réponse est le retard entre l'apparition d'un signal d'entrée et l'apparition du signal de sortie. On parle de causalité. Le temps de réponse d'un API dépend de la période de scrutation qui peut être facilement mesuré.

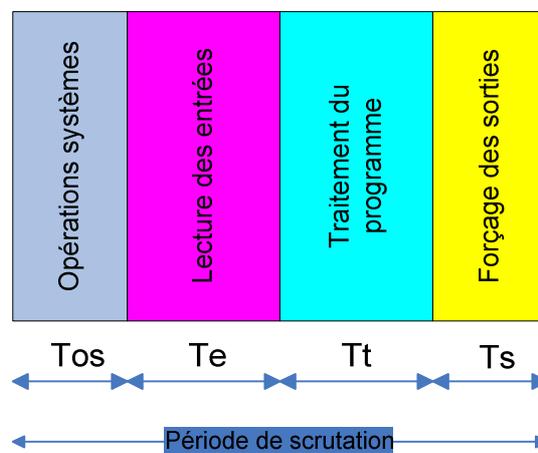
La période de scrutation (P_s) :

Le temps de réponse est lié à la rapidité du microprocesseur, au temps de filtrages des entrées, à la taille du programme et au retard des sorties. Toutes les équations décrites dans le programme sont lues et effectuées, au cours d'un cycle de scrutation. La période d'exploitation, pour la lecture des entrées, pour le traitement du programme et pour le forçage des sorties.

- **Opérations système d'exploitation (T_{os}) :** Temps de traitement des tâches propres au fonctionnement de l'API, traitement des blocs d'organisation (OB).
- **Lecture des entrées (T_e) :** Temps de forçage des bits images des entrées selon l'état des entrées physiques.

- **Traitement du programme (Tt) :** Temps pour effectuer les équations décrites dans le programme utilisateur.
- **Forçage des sorties (Ts) :** temps de forçages des sorties physiques selon l'état des bits images des sorties.

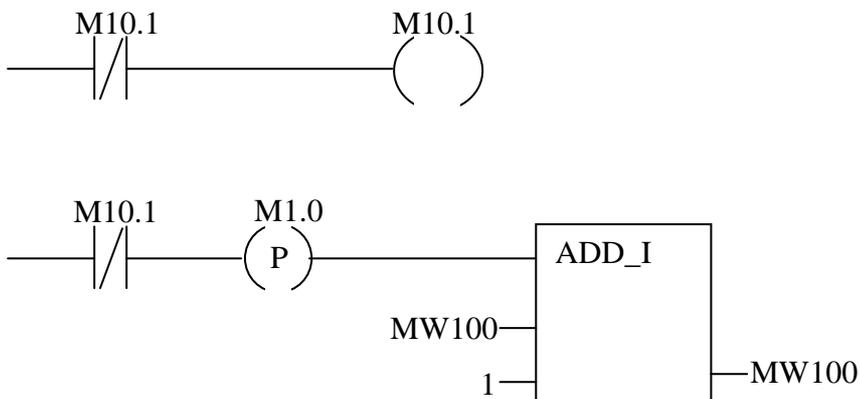
La période de scrutation ($P_s = T_{os} + T_e + T_t + T_s$) est comprise entre quelque millisecondes et quelques dixièmes de secondes. Un bit interne nommé chien de darde passe à l'état 1 lorsque le temps d'un cycle dépasse le temps maximal de scrutation. Dans ce cas on considère que l'API n'à plus de contrôle. L'API passe en mode STOP.



Cycle API

Mesure de la période de scrutation (Ps) :

La mesure de la période de scrutation peut être facilement effectuée en comptant les permutations d'un bit mis en oscillation, pendant un laps du temps fini.



Le bit M10.1 à l'état 1 au cours d'une scrutation et passe à l'état 0 au cours de la scrutation suivante. A chaque front montant du bit M10.1, le mot MW100 s'incrémente. Deux scrutations sont nécessaires pour que le bit M10.1 passe de 0 à 1. La fréquence de scrutation est la valeur de MW100 au bout d'une seconde. Multiplié par 2.

3. Langages de programmation pour les API

Chaque automate possède son propre langage de programmation. Mais par contre, les constructeurs proposent tous une interface répondant à la norme CEI 11313. Cette norme définit cinq langages de programmation utilisables, qui sont :

- **GRAFCET ou SFC** : ce langage de programmation de haut niveau permet la programmation aisée de tous les procédés séquentiels.
- **Schéma par blocs ou FBD** : ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables.
- **Schéma à relais ou LD** : ce langage graphique est essentiellement dédié à la programmation d'équations booléennes.
- **Texte structuré ou ST** : ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.
- **Liste d'instructions ou IL** : ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur

Pour programmer l'automate, l'automaticien peut utiliser :

- une console de programmation ayant pour avantage la portabilité.
- un PC avec lequel la programmation est plus conviviale, communiquant avec l'automate par le biais d'une liaison série RS232 ou RS485 ou d'un réseau de terrain.

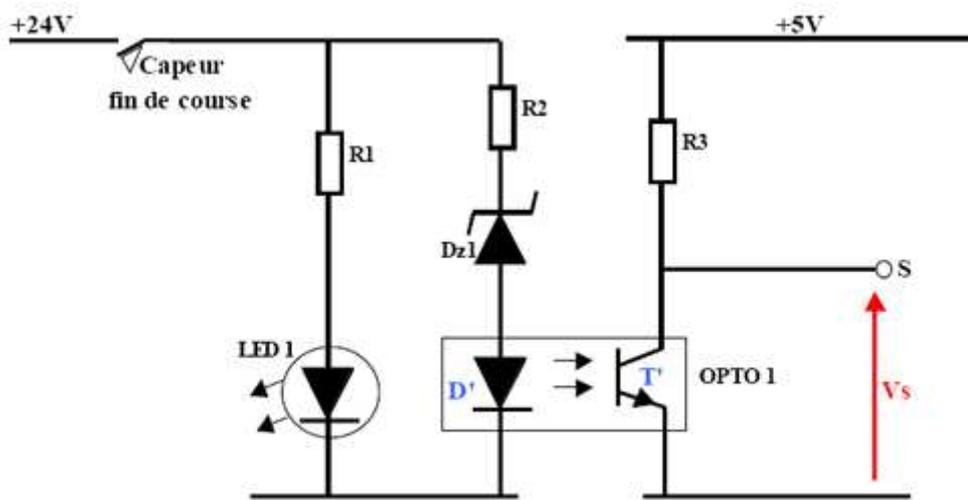
4. Etude des différentes interfaces

4.1- Les cartes d'entrées TOR

Les cartes d'entrées Tout Ou Rien (TOR) sont utilisées pour l'acquisition des signaux binaires. Un bit interne passe à l'état 1 lorsque la tension d'entrée est supérieure à un seuil.

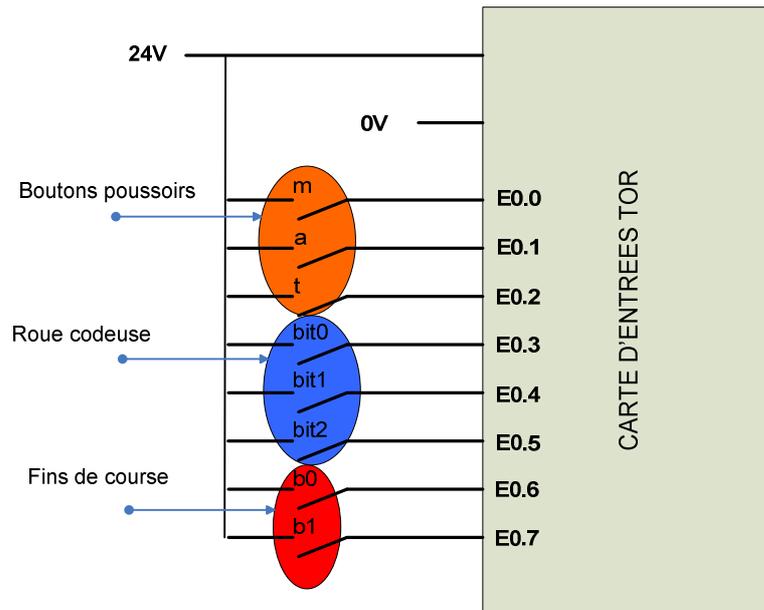
Les cartes d'entrées TOR acceptent des composants fournissant une tension, comme des boutons poussoirs, interrupteurs, roues codeuses, capteurs de fin courses.

La plupart des entrées TOR, des API sont PNP, c'est-à-dire par établissement du +24V, certains capteurs sont à sorties NPN, par établissement de masse.



Les entrées d'un API sont protégées grâce à un opto-coupleurs qu'isole l'électronique de l'API avec l'extérieur.

Un parasite sur une ligne d'entrées ne doit pas être considéré comme une entrée effective. La plupart des cartes d'entrées TOR ne force les bits images des entrées qu'au bout d'un certain temps T_e . Un parasite de durée plus faible ne sera pas pris en compte. Mais ce filtrage crée un retard T_e entre l'apparition de l'entrée et son bit image.

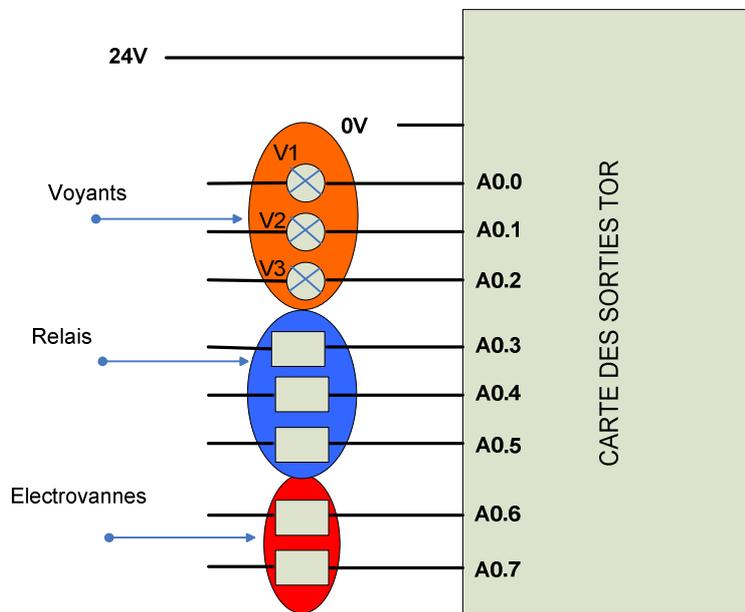
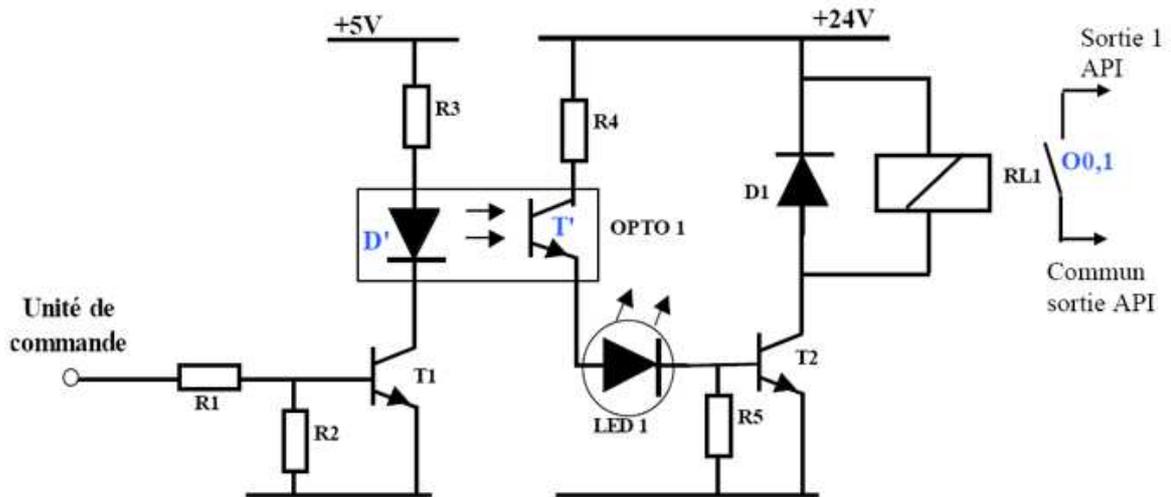


4.2- Les cartes de sorties TOR

Les cartes de sorties TOR commandent des actionneurs Tout ou Rien. Dès le forçage d'un bit de sortie, la carte fournit une tension, à 1 ou à 0, sur la sortie. Des composants comme des contacteurs, des électrovannes, des voyants, peuvent être commandés par ces sorties. Le temps de réponse d'une sortie (T_s), est le temps entre l'apparition d'un bit image de la sortie et la présence de tension sur cette sortie API.

Le retard de commutation (T_s) d'un relais peut être supérieur à 30ms. Par conséquent il ne sera pas possible de faire clignoter une sortie de ce type à une fréquence supérieure à 15Hz. Les sorties à transistors sont plus rapides, le temps (T_s) peut être inférieur à 1ms.

Les sorties TOR peuvent être aussi réalisées avec des transistors ou des triacs, les premières sont utilisées pour les actionneurs dont la tension d'alimentation est à courant continu, les second pour ceux à courant alternatifs.



4.3- Les cartes d'entrées analogiques

La carte d'entrée analogique est constituée d'un convertisseur analogique numérique (CAN). Elle est utilisée pour l'acquisition d'une tension (0V à 10V ou -5V à +5V) ou d'un courant (4 mA à 20 mA) qui varie entre deux bornes sans discontinuité. Un mot interne d'entrée est l'image de la tension ou du courant d'entrée. Tous les capteurs analogiques (potentiomètres, par exemple) peuvent y être câblés. La résolution est directement liée au nombre de bits du CAN.

Le mot image d'entrée analogique :

La valeur image d'entrée est stockée dans un mot de 16 bits d'entrées EWn (n: adresse de l'octet d'entrée).

Si le convertisseur analogique numérique (CAN) possède 12 bits et si la tension varie de -10V à +10V, le mot image évolue entre -2048 et 2047

La résolution d'un CAN:

La résolution est le plus petit écart de tension mesurable. Le mot image d'entrée est un entier.

La résolution est le division de la plage mesurée par la plage du mot interne :

$$R = \text{Plage} / \text{Mot}$$

Certains API convertissent ces 12 bits en un mot de 16 bits, dans ce cas entre -32768 à +32767.

Par exemple un capteur LVDT donne, pour un déplacement de son palpeur de 0 à 8,19mm, une tension évoluant entre 0 et 10V. la sortie analogique de ce capteur est reliée à l'entrée d'une carte CAN 12 bits qui donnent un mot image compris entre 0 et 4095. La résolution est de 10 / 4095, soit environ 0,25 mV et correspond à un déplacement de 0,002mm du palpeur

4.3- Les cartes de sorties analogiques

Les cartes de sorties analogiques, sont constituées d'un convertisseur numérique analogique (CNA). Ses sorties fournissent une tension proportionnelle à un mot de sortie image, interne à la carte, pour commander un actionneur (ou un pré actionneur), accéder par une consigne de tension. Elles conviennent à la commande des variateurs de moteur. La résolution de tension obtenue est dépendante du nombre de bits du CNA.

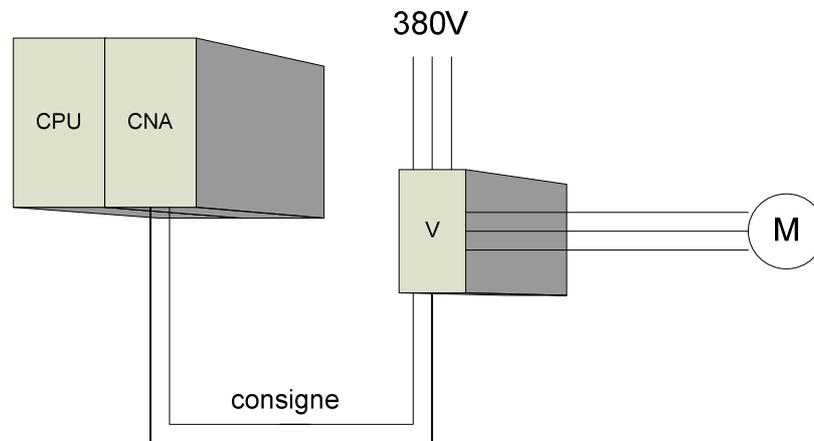
Le mot image de sortie analogique :

Comme une sortie TOR, un mot interne de sortie AWn (n : adresse de l'octet de sortie) est forcé à une valeur par le programme. Par exemple si ce mot est codé sur 16 bits, si AWn est forcé à la valeur 0 la sortie de la CNA prend la valeur nulle. Si ce mot est forcé à la valeur +32767, la sortie prend la valeur maximale (10V)

La résolution d'un CAN:

Le nombre de bits d'un CAN peut varier entre 8 et 16. Si le CNA possède 16 bits et fournit une tension de -10V à 10V, la résolution est de 20V divisée par 65535, c'est-à-dire de 0,3mV pour un point.

Exemple : la figure représente le câblage d'un moteur avec son variateur (V), une carte CNA envoie une tension de consigne proportionnelle à la vitesse du moteur (M) désirée.



Une carte CNA de 12 bits, est affectée à l'adresse 30. Le mot AW30, interne à la CNA évolue donc entre -2047 à + 2047. Il peut moduler la tension, envoyée à la consigne d'entrée du variateur, entre -10V et +10V. Si la plage de vitesse du moteur est de -2000 à +2000 tr/mn, la résolution est de 0,976 tr/mn. (4000/4096).

4.3- La carte de comptage rapide

Les cartes d'entrée TOR ne permettent pas le comptage d'impulsions de grandes fréquences. Certains API monoblocs possèdent des entrées rapides, d'autres modulaires, peuvent être dotés de cartes de comptage rapide (CCR).

Les limites en fréquence d'une entrée:

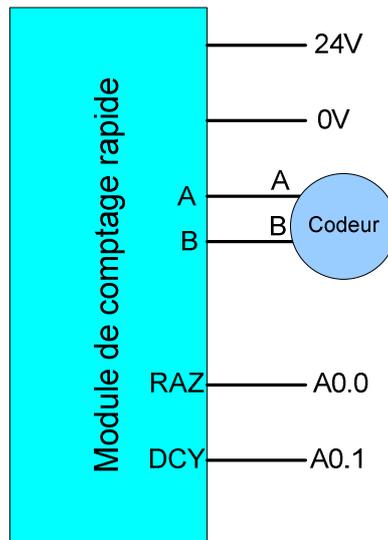
En général, la fréquence d'une entrée TOR ne dépasse pas 50Hz. Le théorème de SHANNON impose, pour ne perdre l'information, une fréquence de comptage 2 fois supérieure à la fréquence à compter.

La fréquence maximale de rotation d'un codeur incrémentale de 100 points par tour (Nc), câblée sur une entrée TOR est de seulement 15 tr/mn.

$$N_{max} < (60 \cdot 50) / 2 \cdot N_c = 15 \text{ tr / mn.}$$

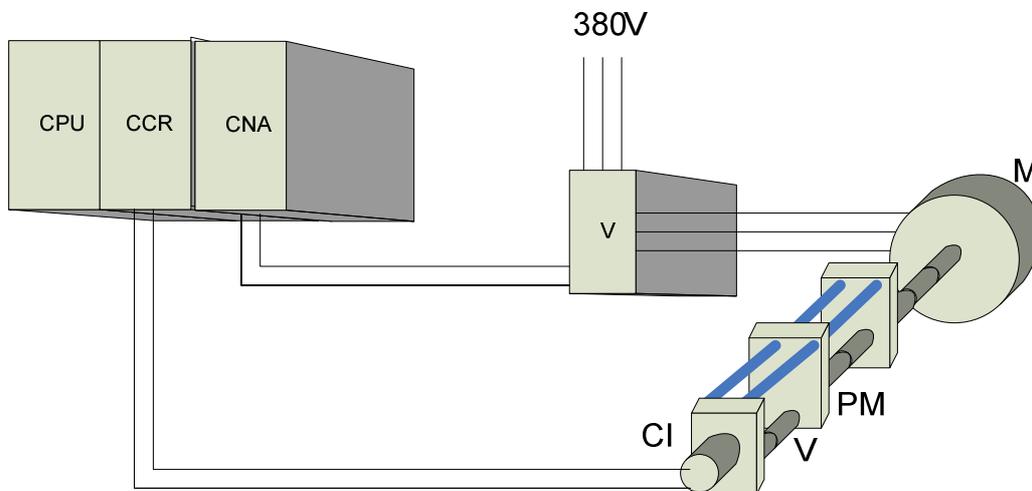
Le module de comptage rapide:

Dans un slot d'un API modulaires, il est possible d'ajouter un module de comptage rapide. Ce module compte les permutations d'une entrée, qui oscille à des fréquences de 1KHz à 1MHz.



Un mot interne EWn est incrémenté à chaque front montant sur l'entrée A. La remise à zéro du mot interne se fait par l'entrée RAZ. Certaines cartes possèdent une entrée DCY, dans ce cas le mot interne EWn n'évolue que si DCY est à 1.

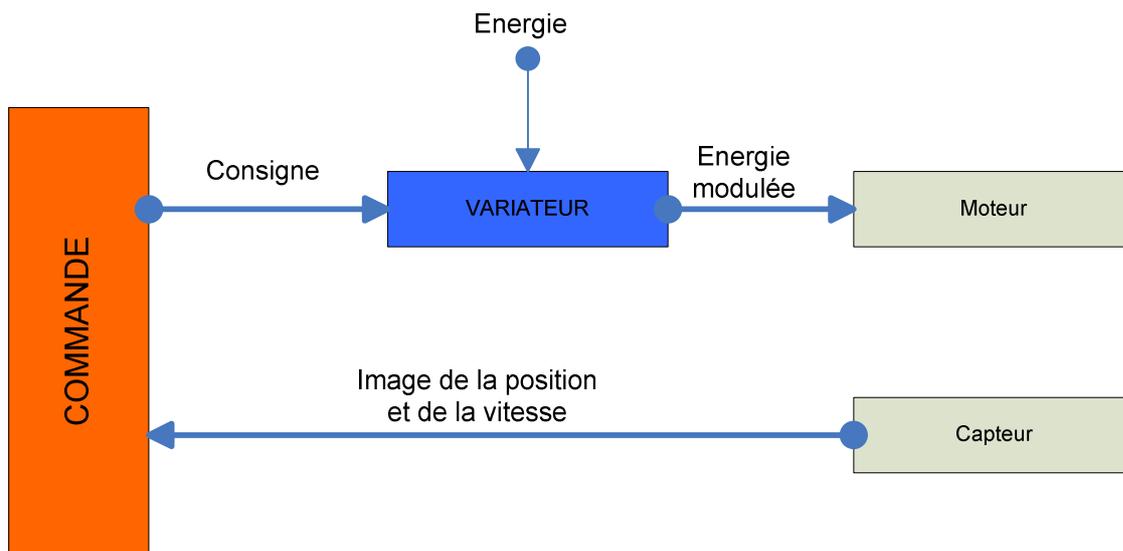
Exemple : l'exemple suivant montre une partie mobile PM guidée en translation. Le moteur entraine une vis V de pas égal à 5mm.



4.4- La commande d'axe

La commande d'axe peut être faite, au moyen de cartes d'entrées et de sorties analogiques, mais l'utilisation des cartes d'axes simplifie fortement la programmation.

Un axe est une partie mécanique composée d'un moteur, d'un guidage et d'un ou plusieurs capteurs, liés à l'arbre du moteur, qui donnent l'image de la position et de la vitesse de la partie mobile.



La commande d'un axe par cartes d'axes.

La programmation d'axe se limite à la définition des paramètres du mouvement, accélération, Vitesse maxi, position à atteindre. Ces cartes acceptent, plusieurs types de capteurs de position, analogique, codeurs incrémentaux et absolus.