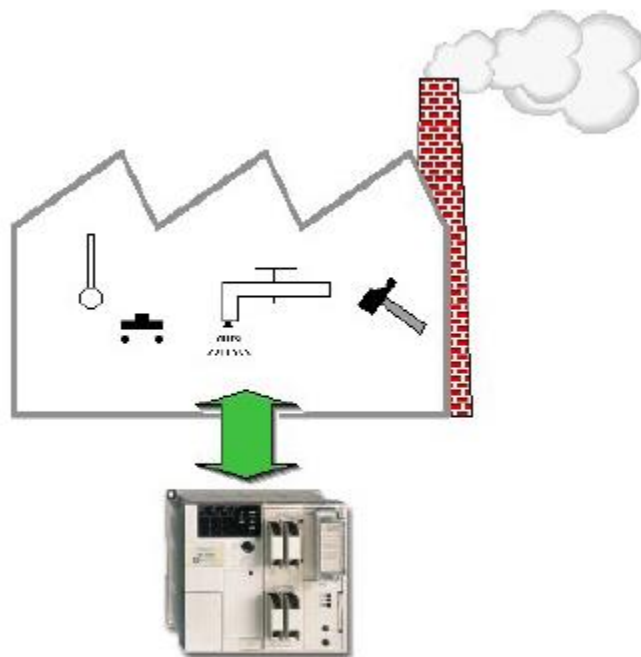


COURS D'INIATION A L'AUTOMATIQUE.



Présenté par :

Mr. Mazoughou GOEPOGUI

Tel: 655 34 42 38 / 669 35 43 10

E-mail: massaleidamagoe@yahoo.fr

I. INTRODUCTION.

I.1. Définition de l'automatique.

Un système est dit *automatique* lorsqu'il peut fonctionner tout seul sans intervention humaine.

Ainsi l'automatique est une branche de la technique et de la technologie qui s'occupe de l'étude et de la fabrication des systèmes automatiques (c'est-à-dire des systèmes pour lesquels l'homme n'intervient quasiment pas, sauf pour donner ses ordres ou consignes).

L'automatisation de la production consiste à transférer tout ou partie des tâches de coordination, auparavant exécutées par des opérateurs humains, dans un ensemble d'objets techniques appelé *partie commande*.

Dans toute la panoplie d'appareils utilisés pour l'automatisation des processus industriels, l'*Automate Programmable Industriel (API)* occupe une place très importante. La création du premier API remonte à la fin des années 60. L'industrie automobile en est la première instigatrice et la première utilisatrice. Jusqu'alors, la commande des automatismes industriels était réalisée à l'aide d'armoires de commande à relais. Les changements annuels de modèle de voiture impliquaient des modifications fréquentes des chaînes de montage et de leurs armoires de commande. Comme ces dernières étaient complexes, leurs modifications étaient difficiles et comportaient un risque élevé d'erreur de branchement. L'industrie automobile a donc amené la création d'un appareil programmable capable de remplacer les armoires de commande.

Ce fut alors le début d'une grande aventure pour plusieurs compagnies. Les ordinateurs qui étaient principalement utilisés pour faire de la comptabilité furent modifiés afin de répondre aux exigences de la commande industrielle. Petit à petit la technique s'améliora et gagna plus d'adeptes. Il a fallu cependant attendre une bonne décennie avant que le concept soit introduit, de façon systématique, dans l'industrie. Aujourd'hui, l'API est le principal système de commande utilisé dans l'industrie.

L'automatique est une branche de synthèse qui nécessite des connaissances dans de nombreux domaines techniques et scientifiques.

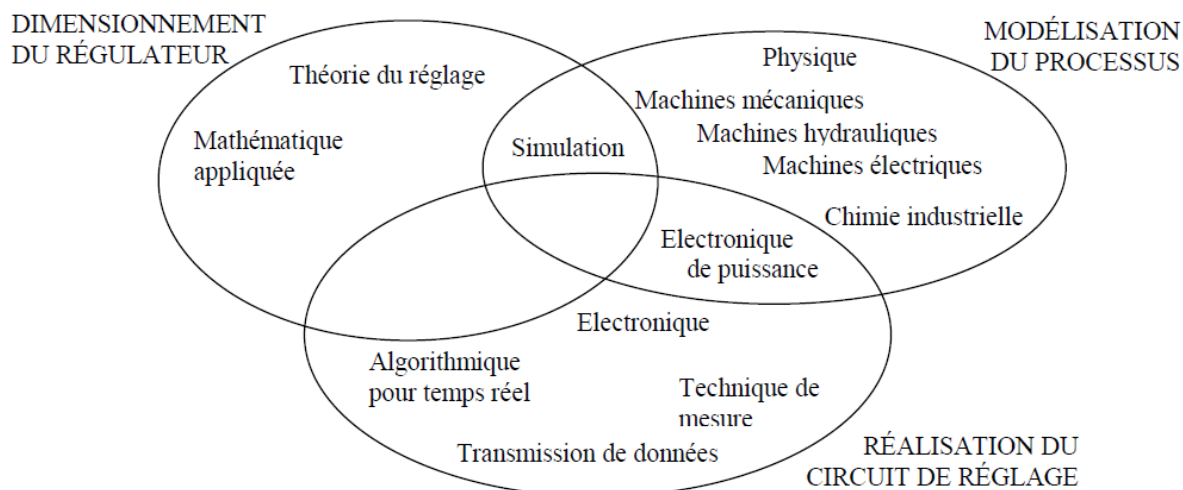


Fig. 1.2 Pluridisciplinarités des systèmes asservis.

Si on n'a pas trop d'exigences, il est vrai qu'on peut réaliser une installation réglée sans grande connaissances mathématiques et physiques: par exemple le réglage de la chaufferie

d'un bâtiment. Il suffit de procéder par essais et ajustages successifs des paramètres de réglage.

Si on spécifie simplement qu'on veut optimiser la consommation de mazout, il faut déjà une connaissance approfondie du processus et on doit faire appel à la théorie du réglage. Pour la majorité des systèmes, la méthode par tâtonnement est trop coûteuse en temps et n'aboutit qu'à des résultats médiocres. Pour certaines applications (centrales nucléaires, astronautique, ...) cette méthode est même très dangereuse.

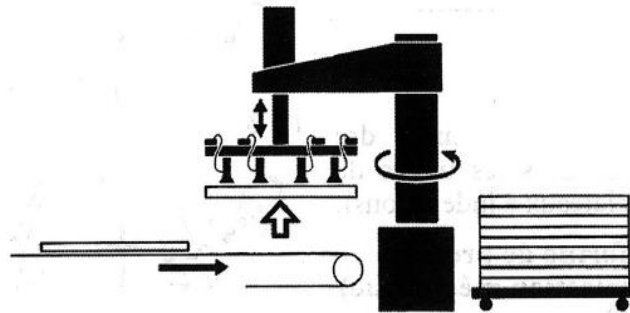
Comme le suggère l'ensemble "modélisation" sur la figure ci-dessus, les systèmes asservis s'appliquent à tous les domaines de la technique. A y regarder de près, on constate que beaucoup de phénomènes naturels peuvent être décrits avec la théorie du réglage.

I.2. Domaine d'application de l'automatique.

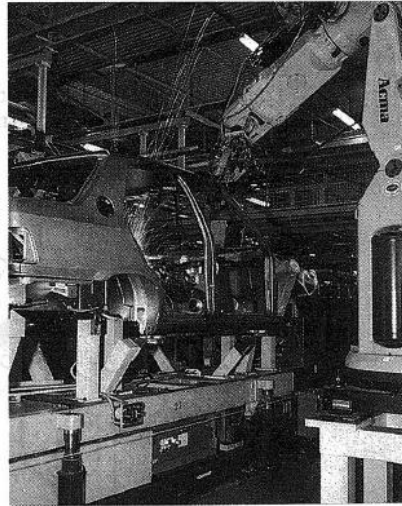
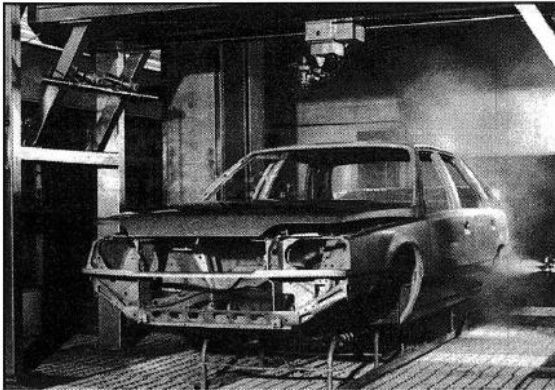
Aujourd'hui, il serait difficile de concevoir un système de production sans avoir recours aux différentes technologies et composants qui forment les systèmes automatisés. Ainsi, l'automatique, loin d'être un champ scientifique et technique fermé, est transversal dans ses applications : allant de la régulation de l'économie à celle de la machine électrique en passant par le pilotage des lanceurs, l'asservissement des têtes de lecture dans les disques durs, etc. Il emprunte ses outils aux mathématiques, électricité, électronique, informatique.

Exemples de domaines d'applications.

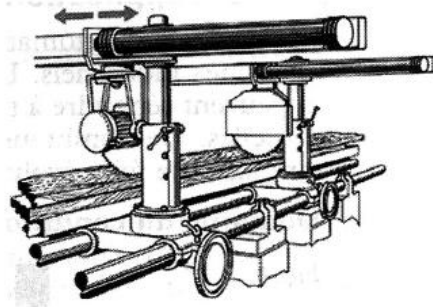
1. Conditionnement sur palette après emballage.



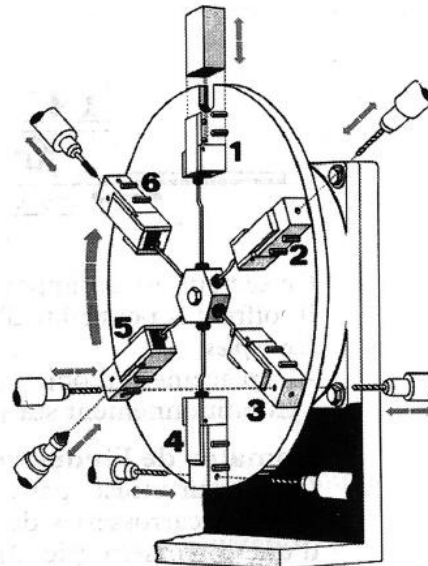
2. L'industrie automobile avec l'utilisation de robots industriels pour effectuer l'assemblage et la peinture des carrosseries



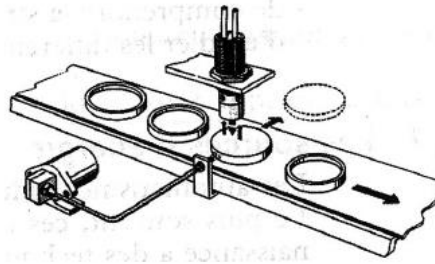
3. L'industrie du bois avec les opérations de débit, de sciage et d'usinage du bois.



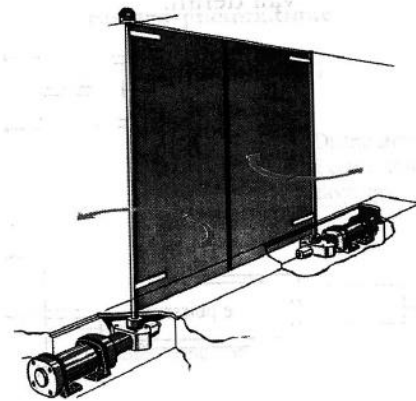
4. Machine-outil dans les unités de perçage.



5. Contrôle de produits : Détection de défauts en bout de chaîne de production.



6. *Automatisation de services : ouvertures programmées de portes et fenêtres, gestion centralisée de bâtiment.*



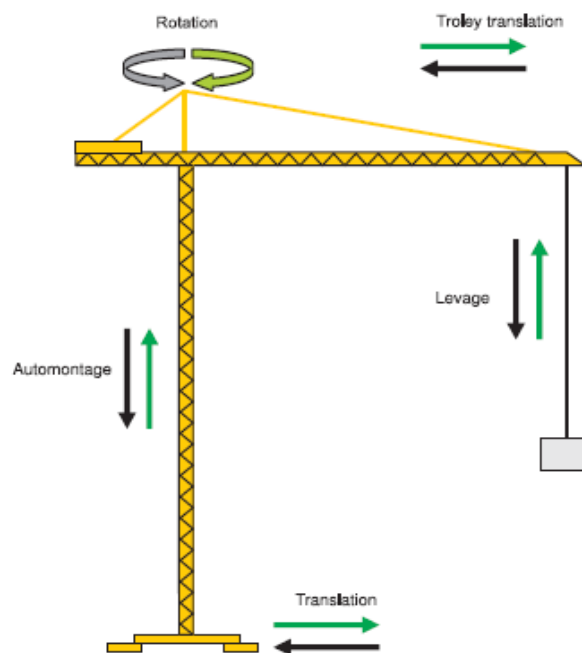
7. *Pompe GPL*



8. *Positionnement offshore.*



9. Grue de chantier.



I.3. Avantages et inconvénients de l'automatique.

1.3.1. Les avantage.

- L'augmentation de la production ;
- L'aptitude à convenir à tous les milieux de production ;
- La souplesse d'utilisation ;
- Une meilleure rentabilité ;
- Une meilleure compétitivité ;
- Améliorer la flexibilité de production ;
- S'adapter à des contextes particuliers ;

- Adaptation à des environnements hostiles pour l'homme (milieu salin, spatial, nucléaire...);
- Adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées...);
- Augmenter la sécurité, etc...

I.3.2 Les inconvénients.

- Le coût élevé du matériel, principalement avec les systèmes hydrauliques ;
- La maintenance doit être structurée ;
- La suppression d'emplois.

I.4. Différents types de systèmes automatisés.

Il existe deux types de systèmes automatisés :

1. **Les systèmes automatisés continus (ou analogique)** pour asservir et/ou commander des grandeurs physiques de façon précise et sans aide extérieure (l'angle d'une fusée, la vitesse de rotation d'un lecteur CD, la position du bras d'un robot, le pilotage automatique d'un avion).
2. **Les systèmes automatisés à événements discrets (ou numérique)** pour les commandes en tout ou rien (les distributeurs automatiques, les ascenseurs, le montage automatique dans le milieu industriel, les feux de croisement, les passages à niveaux). Ces systèmes se divisent en deux parties :

- a) **Les systèmes automatisés à logique câblée** pour lesquels la commande est effectuée uniquement selon câblage électrique du circuit. Là également on distingue deux types :

- **Les systèmes automatisés combinatoires.** Ces systèmes n'utilisent aucun mécanisme de mémorisation. A une combinaison des entrées ne correspond qu'une seule combinaison des sorties. La logique associée est la logique combinatoire. Les outils utilisés pour les concevoir sont l'algèbre de Boole, les tables de vérité, les tableaux de Karnaugh.

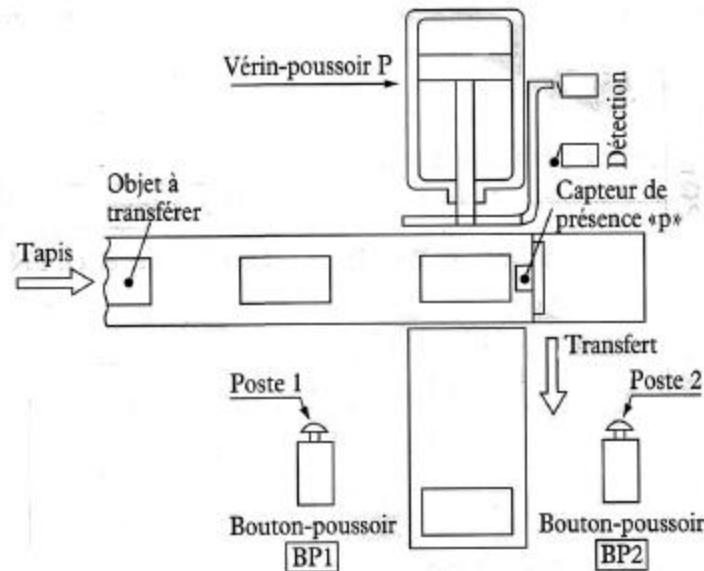
Exemple : Sur la figure ci-dessous, si la présence de l'objet à transférer est détectée par le capteur de présence "p", alors le vérin-poussoir P entrera en fonction si l'opérateur du poste 1 ou celui du poste 2, appuie sur le bouton poussoir correspondant (BP1 ou BP2).

Les systèmes automatisés utilisant la technique "combinatoire" sont aujourd'hui très peu utilisés. Ils peuvent encore se concevoir sur des mécanismes simples où le nombre d'actions à effectuer est limité (ex : pilotage de 2 vérins). Ils présentent en outre l'avantage de n'utiliser que très peu de composants.

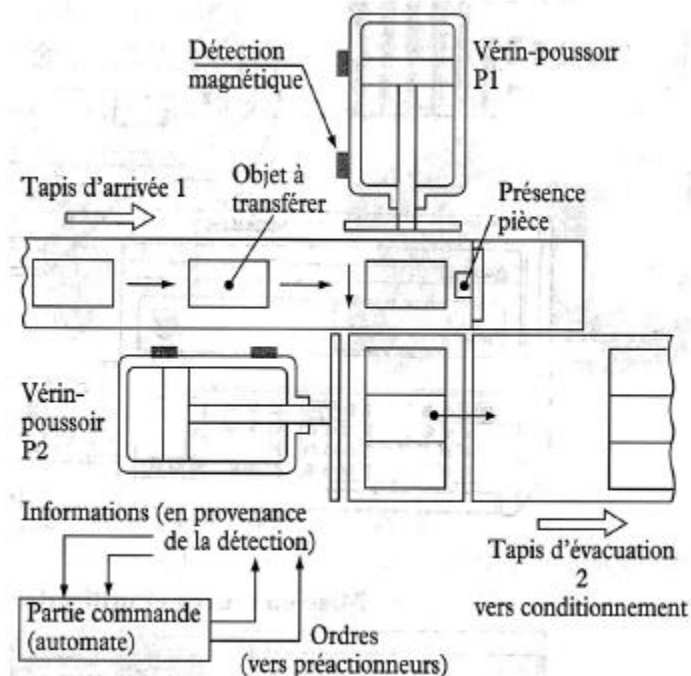
- **Les systèmes automatisés séquentiels.** Pour ces systèmes, le déroulement du cycle s'effectue étape par étape. A une situation des entrées peuvent correspondre plusieurs situations de sortie. La sélection d'une étape ou d'une autre dépend de la situation antérieure du dispositif.

Exemple : Sur la figure ci-dessous, le système place les objets côte à côte 3 par 3, sur un tapis d'évacuation 2. Ainsi placés, les objets sont emmenés vers un dispositif de conditionnement sous film plastique rétractable.

- b) **Les systèmes automatisés programmables.** Ces systèmes sont les plus répandus dans le domaine industriel. L'élément principal s'appelle l'Automate Programmable Industriel ou l'API. Le pilotage des actionneurs se fait selon le programme installé dans la mémoire de l'automate. Il existe sur le marché de nombreuses marques d'automates : Télémécanique, Siemens, Omron, Allen Bradley, Cegetel, etc.



Exemple de système automatisé combinatoire.



Exemple de système automatisé séquentiel.

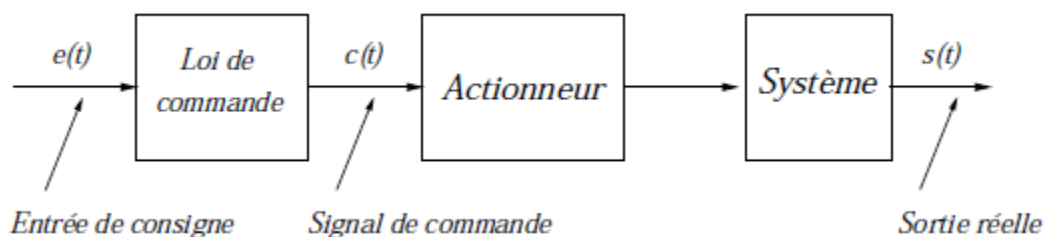
I.5. Structure d'un système automatique.

Quelle que soit la nature du système commandé, il est toujours possible de classer les différentes structures de commande en deux grandes familles. Les structures de commande en **boucle ouverte** et les structures de commande à **contre-réaction** appelées également structures de commande en **boucle fermée**.

I.5.1. Commande en boucle ouverte.

En l'absence d'entrées perturbatrices et en supposant que le modèle mathématique du système est parfait, il est imaginable de générer un signal de commande produisant le signal de sortie souhaité. Cela constitue le principe de la commande en **boucle ouverte**. Les signaux d'entrée ne sont pas influencés par la connaissance des signaux de sortie.

Cette solution est envisageable dans le cas où le système est parfaitement connu et modélisé et dans le cas où l'obtention d'une mesure de la sortie n'est pas économiquement possible.



Commande en boucle ouverte

Exemple (machine à laver).

L'exemple typique de ce type de structure est constitué par la machine à laver fonctionnant sur la base de cycles préprogrammés ne possédant pas d'informations mesurées concernant le degré de propreté du linge.

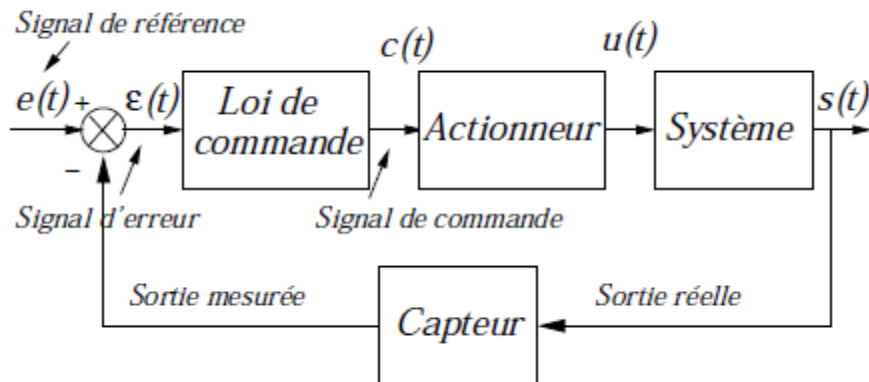
Toutefois, si le système à commander n'est pas parfaitement connu ou si des perturbations l'affectent, les signaux de sortie ne seront pas ceux souhaités.

I.5.2. Commande en boucle fermée.

L'introduction d'un retour d'information sur les sorties mesurées s'avère nécessaire pour pallier aux inconvénients évoqués ci-haut. Le principe de commande en boucle fermée est illustré sur la figure ci-dessous et définit la structure de **commande à contre-réaction** (**feedback** en anglais). On parle alors de **système bouclé** (figure I.28), par opposition aux **systèmes en boucle ouverte**.

Un système bouclé vérifie en quelque sorte que la réponse du système correspond à l'entrée de référence tandis qu'un système en boucle ouverte commande sans contrôler l'effet de son action. Les systèmes de commande en boucle fermée sont ainsi préférables quand des perturbations non modélisables et/ou des variations imprévisibles des paramètres sont présentes. Cette structure de commande permet ainsi d'améliorer les performances dynamiques du système commandé (rapidité, rejet de perturbation, meilleur suivi de consignes, moindre sensibilité aux variations paramétriques du modèle, stabilisation de systèmes instables en boucle ouverte).

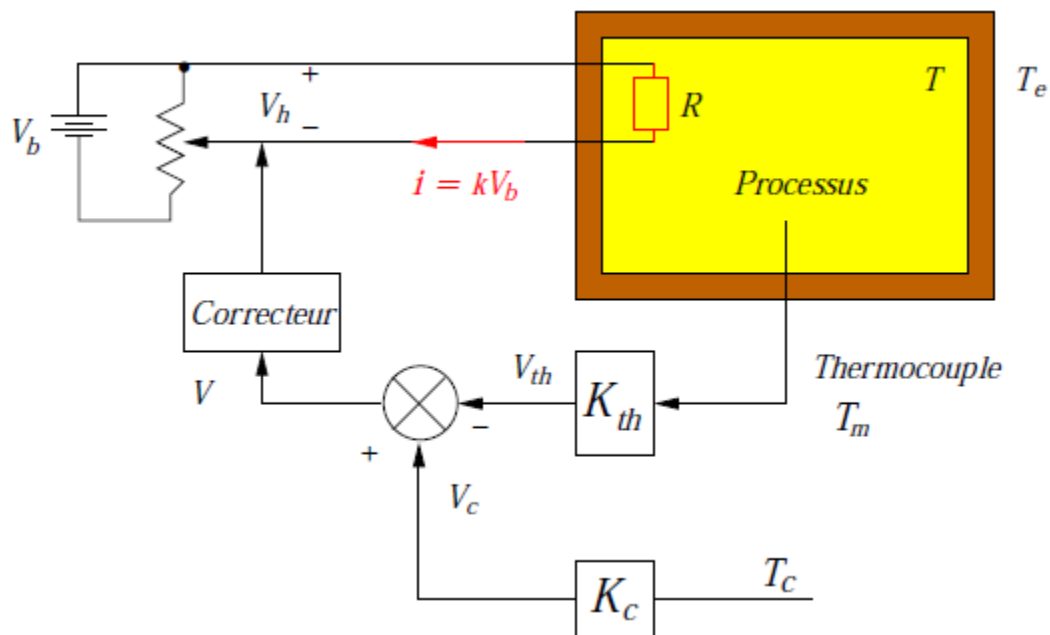
Il est toutefois important de remarquer que cette structure de commande ne présente pas que des avantages. Elle nécessite l'emploi de capteurs qui augmentent le coût d'une installation. D'autre part, le problème de la stabilité et de la précision des systèmes à contre-réaction se pose de manière plus complexe. Le concept de rétroaction est toutefois à la base de tous les développements théoriques de l'Automatique moderne.



Commande en boucle fermée.

Exemple (Régulation de température dans un four).

Un des problèmes les plus fréquemment rencontrés dans l'industrie est celui de la régulation de température dans une enceinte close (four, incubateur...). Un système de contrôle thermique est représenté à la figure ci-dessous. La température T d'un processus donné est régulée dans un four à l'aide d'une résistance chauffante R . Par comparaison entre la tension de consigne V_c et la tension "mesurée", reflet de la température T , une tension de commande V est délivrée à l'actionneur modifiant en conséquence le courant parcourant la résistance R .



Régulation de température.

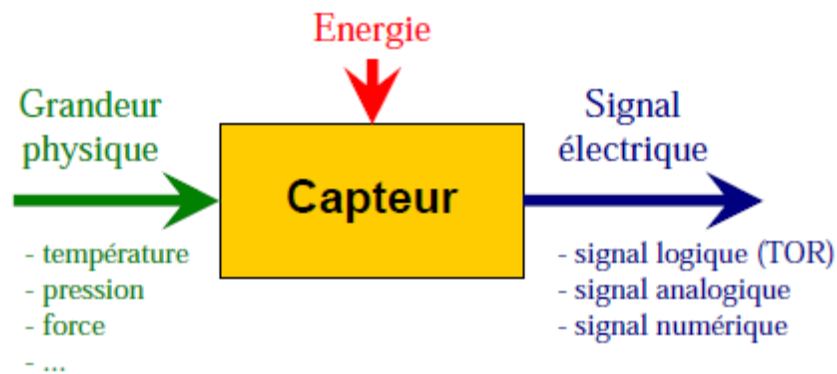
I.6. Les capteurs.

Dans de nombreux domaines (industrie, recherche scientifique, services, loisirs ...), on a besoin de contrôler de nombreux paramètres physiques (température, force, position, vitesse,

luminosité, ...). Le capteur est l'élément indispensable à la mesure et au contrôle de ces grandeurs physiques.

I.6.1. Définition.

Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (très souvent électrique). Cette grandeur représentative de la grandeur prélevée est utilisable à des fins de mesure ou de commande.



I.6.2. Classification.

Si on s'intéresse à la nature de la commande, on distingue **les capteurs à commande mécaniques et les capteurs à commande électronique**.

I.6.2.1. Les capteurs à commande mécanique.

Dès qu'une grandeur physique est détectée (ou change d'état), ils délivrent en sortie un signal électrique ou une pression pneumatique. Ils fonctionnent en Tout Ou Rien (TOR). Par exemple :

- ✓ *Les capteurs à commande manuelle* destinés à l'équipement des pupitres et des postes de commande comme des boutons poussoirs, les boutons à 2 ou 3 positions. La commande est fournie par l'opérateur ;
- ✓ *Les interrupteurs de position* situés sur la partie opérative, ils détectent par contact la présence d'une partie mobile (par ex : la tige d'un vérin).

I.6.2.2. Les capteurs à commande électronique.

Si l'on s'intéresse aux phénomènes physiques mis en jeu dans les capteurs électroniques, on peut les classer en deux catégories : *les capteurs actifs et les capteurs passifs*.

a) Capteurs actifs.

Fonctionnant en **générateur**, un capteur actif est généralement fondé, dans son principe, sur un effet physique qui assure la conversion en énergie électrique de la forme d'énergie propre à la grandeur physique à prélever (énergie thermique, mécanique ou de rayonnement). *Les effets physiques les plus classiques sont :*

- ✓ *Effet thermoélectrique* : Un circuit formé de deux conducteurs de nature chimique différente, dont les jonctions sont à des températures T_1 et T_2 , est le siège d'une force électromotrice d'origine thermique $e(T_1, T_2)$.
- ✓ *Effet piézo-électrique* : L'application d'une contrainte mécanique à certains matériaux dits piézo-électriques (le quartz par exemple) entraîne l'apparition d'une déformation et d'une même charge électrique de signe différent sur les faces opposées.

- ✓ **Effet d'induction électromagnétique** : La variation du flux d'induction magnétique dans un circuit électrique induit une tension électrique (détection de passage d'un objet métallique).
- ✓ **Effet photo-électrique** : La libération de charges électriques dans la matière sous l'influence d'un rayonnement lumineux ou plus généralement d'une onde électromagnétique.
- ✓ **Effet Hall** : Un champ magnétique B et un courant électrique I créent dans le matériau une différence de potentiel U_H .
- ✓ **Effet photovoltaïque** : Des électrons et des trous sont libérés au voisinage d'une jonction PN illuminée, leur déplacement modifie la tension à ses bornes.

Grandeur physique mesurée	Effet utilisé	Grandeur de sortie
Température	Thermoélectricité	Tension
Flux de rayonnement optique	Photo-émission	Courant
	Effet photovoltaïque	Tension
	Effet photo-électrique	Tension
Force	Piézo-électricité	Charge électrique
Pression		
Accélération	Induction électromagnétique	Tension
Vitesse		
Position (Aimant)	Effet Hall	Tension
Courant		

b) Capteurs passifs.

Il s'agit généralement d'**impédance** dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte :

- ✓ Soit d'une variation de dimension du capteur, c'est le principe de fonctionnement d'un grand nombre de capteur de position (potentiomètre, inductance à noyaux mobile, condensateur à armature mobile).
- ✓ Soit d'une déformation résultant de force ou de grandeur s'y ramenant, pression accélération (armature de condensateur soumise à une différence de pression, jauge d'extensomètre liée à une structure déformable).

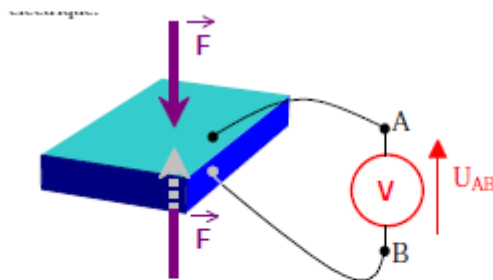
Grandeur mesurée	Caractéristique électrique sensible	Type de matériau utilisé
Température	Résistivité	Métaux : platine, nickel, cuivre ...
Très basse température	Constante diélectrique	Verre
Flux de rayonnement optique	Résistivité	Semi-conducteur
Déformation	Résistivité	Alliage de Nickel, silicium dopé
	Perméabilité magnétique	Alliage ferromagnétique
Position (aimant)	Résistivité	Matériaux magnéto résistants : bismuth, antimoine d'indium
Humidité	Résistivité	Chlorure de lithium

I.6.3. Exemples de capteurs.

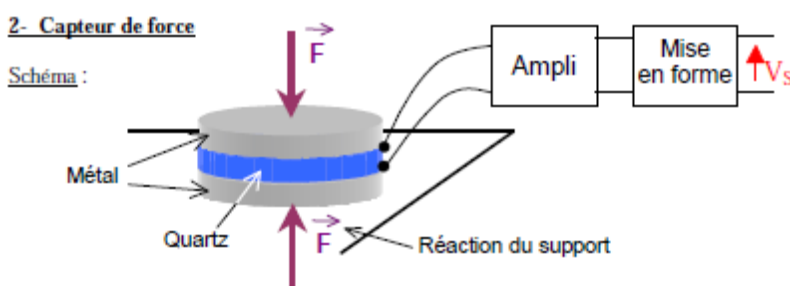
I.6.3.1. Capteurs à effet piézoélectrique.

a) Effet piézoélectrique.

Une force appliquée à une lame de quartz induit une déformation qui donne naissance à une tension électrique.



b) Capteur de force.



c) Capteur de pression.

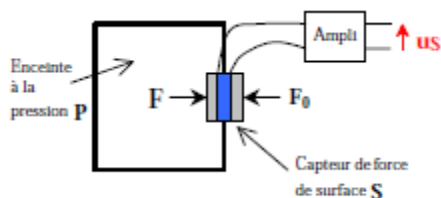
Lorsqu'un corps (gaz, liquide ou solide) exerce une force F sur une paroi S (surface); on peut définir la pression P exercée par ce corps avec la relation ci-dessous.

$$P = \frac{F}{S} \text{ avec les unités : } 1\text{Pascal} = \frac{1\text{Newton}}{1\text{m}^2} \text{ ou } 1\text{Pa} = \frac{1\text{N}}{1\text{m}^2} .$$

On rappelle que 1 kg = 9,81 N.

Unités : 1 bar = 10⁵ Pa = 100 000 N / m² ≈ 10 000 kg / m² = 1 kg / cm²

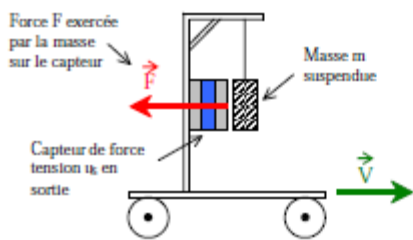
Le capteur de force est inséré dans la paroi d'une enceinte où règne une pression P. Une face du capteur est soumise à la force F (pression P) et l'autre face est soumise à la force F₀ (pression extérieure P₀).



On a $F = P.S$; $F_0 = P_0.S$ et $u_s = k.(F+F_0)$ (capteur de force, k = constante).
 Donc $u_s = k.S (P + P_0) = k' (P + P_0) \Rightarrow u_s = k' (P + P_0)$.

Il s'agit ici d'un capteur de pression qui mesure la somme de la pression extérieure P₀ et de la pression de l'enceinte P.

d) Capteur d'accélération.



L'augmentation de vitesse V du véhicule donne une accélération a qui induit une force F exercée par la masse sur le capteur.

On a donc :

$$F = m.a \text{ mais } u_s = 2k.F$$

$$\text{donc } \boxed{u_s = 2k.m.a}$$

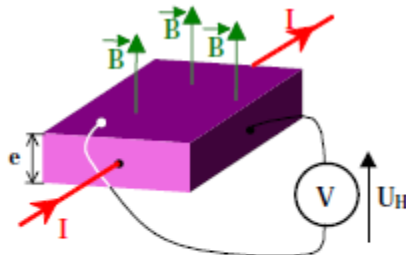
e) Récepteur à ultrason.

La réception d'un son engendre une variation de pression à la surface du récepteur. Un capteur de pression sur cette surface donnera donc une tension image du signal ultrasonore.

I.6.3.2. Capteurs à effet hall.

a) L'effet Hall.

Un barreau de semi-conducteur soumis à un champ magnétique uniforme B et traversé par un courant I , est le siège d'une force électromotrice U_H sur deux de ses faces.



La tension de Hall U_H est définie par la relation ci-dessous :

$$\boxed{U_H = R_H \frac{IB}{e}} \text{ avec : } R_H : \text{ constante de Hall (dépend du semi-conducteur)}$$

I : intensité de la source de courant (A)

B : intensité du champ magnétique (T)

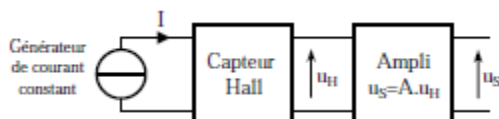
e : épaisseur du barreau de silicium.

Si on maintient le courant I constant, on a donc une tension U_H proportionnelle au

champ magnétique B : $U_H = k.B$ avec k constante égale à $R_H \frac{I}{e}$.

b) Capteur de champ magnétique.

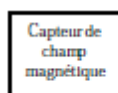
La structure typique d'un capteur de champ magnétique est la suivante :



La sensibilité de ce capteur pourra être ajustée en agissant sur I et sur A .

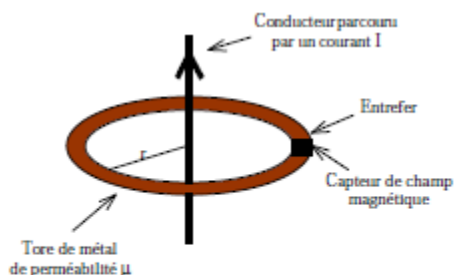
c) Autres applications.

➤ Capteur de proximité.



Le capteur détecte l'approche de l'aimant placé au préalable sur un objet.

- *Mesure de l'intensité d'un courant électrique sans "ouvrir" le circuit.*



Le courant I crée un champ magnétique proportionnel à ce courant : $B = \frac{\mu I}{2\pi r}$.

Le capteur donne une tension $U_s = k.B = k'.I$ avec k et k' constantes.

C'est le principe des pinces ampèremétriques (mesure de forts courants de 1000A et plus).

Avantage :

- Plus de détérioration des ampèremètres "classiques".
- Pas de danger car le fil reste isolé (pas besoin d'ouvrir le circuit).
- Rapidité d'intervention.

I.6.3.3. Capteurs à effet photoélectrique.

a) L'effet photoélectrique.

Un semi-conducteur est un matériau pauvre en porteurs de charges électriques (isolant). Lorsqu'un photon d'énergie suffisante excite un atome du matériau, celui-ci libère plus facilement un électron qui participera à la conduction.

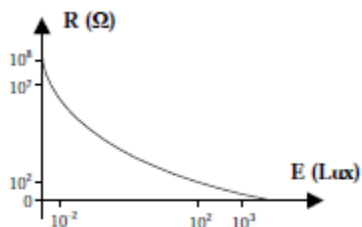
b) Les photorésistances.

Une photorésistance est une résistance dont la valeur varie en fonction du flux lumineux qu'elle reçoit.

Exemple :

- Obscurité $R_0 = 20M\Omega$ (0lux).
- Lumière naturelle $R_1 = 100k\Omega$ (500lux).
- Lumière intense $R_2 = 100\Omega$ (10000lux).

Courbe :



Avantage :

- Bonne sensibilité.
- Faible coût et robustesse.

Inconvénients :

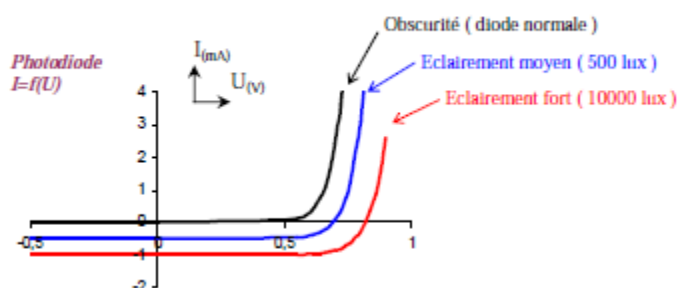
- Temps de réponse élevé ;
- Bande passante étroite ;
- Sensible à la chaleur.

Utilisation : Détection des changements obscurité-lumière (éclairage public).

c) Les photodiodes.

Une photodiode est une diode dont la jonction PN peut être soumise à un éclairage lumineux.

Courbe : Le graphe $I = f(U)$ pour une photodiode dépend de l'éclairement (Lux) de la jonction PN.



On constate que lorsque la diode est éclairée, elle peut se comporter en générateur ($I = 0 \Rightarrow U = 0,7V$ pour 1000lux). On a donc affaire à une photopile (effet photovoltaïque).

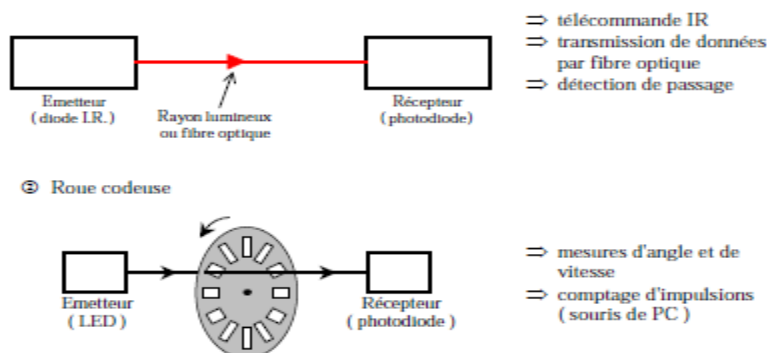
Avantage :

- Bonne sensibilité ;
- Faible temps de réponse (bande passante élevée).

Inconvénients :

- Coût plus élevé qu'une photorésistance ;
- Nécessite un circuit de polarisation précis.

Utilisations : Transmission de données.



d) Codes à barres.

Ce procédé de détection, dont l'invention remonte en 1950, permet une identification précise et sans erreur d'un produit. Il est très employé dans les domaines agro-alimentaires,

pharmaceutiques et industriels. Une étiquette, composée de barres noires et blanches alternées verticales, est lue par un crayon optique. Il existe deux familles de codes :

- *Les codes dits "alimentaires"* composés de barres et d'espaces dont la largeur peut varier de 1 à 3 ;
- *Les codes industriels* toujours composés de barres et d'espaces larges représentant le 1 logique et de barres et d'espaces minces représentant le 0 logique ;

Représentation des étiquettes



EAN 8



UPCA



EAN 13



7 UPCE

Codes EAN en Europe
Codes UPC aux Etats-Unis



2/5 entrelacé



Code 128

XGV-SD 20



Code 39

*VISION



MONARCH

1 2 3 4 5 6 7 8 9 0

Codes numériques : 2 parmi 5
2 parmi 5 entrelacé
Codes alphanunériques 39, 128.

Le lecteur/décodeur a pour rôle de lire, de décoder puis de transmettre l'information d'une étiquette particulière vers une unité de traitement. Il fonctionne comme une caméra vidéo analysant une image électronique.

Avantage :

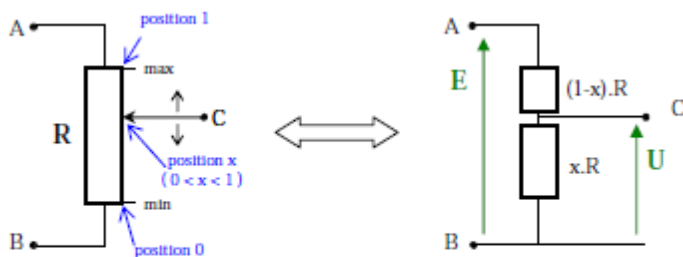
L'information portée par un code à barres est fiable (un code mal imprimé demeure lisible), facile à imprimer, économique, se limitant à une étiquette donc simple.

I.6.3.4. Capteurs à résistance variable par déformation.

a) Capteurs potentiométriques de déplacement.

Principe.

Pour mesurer la position d'un objet, il suffit de le relier mécaniquement au curseur C d'un potentiomètre (schéma ci-dessous).



On applique une tension continue E entre les extrémités A et B du potentiomètre.

La tension U en sortie aura l'expression suivante : $U = E \frac{x.R}{R} = x.E$

La tension U en sortie est donc proportionnelle à la position x du curseur.

Avantages.

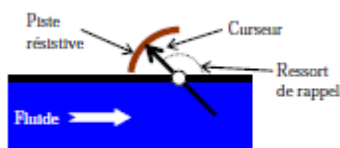
- Simplicité d'utilisation ;
- Faible coût.

Inconvénient.

- Usure mécanique (utilisation déconseillée dans les asservissements très dynamiques).

Utilisations.

- Mesures de déplacements rectilignes (potentiomètre rectiligne) ;
- Mesures d'angles de rotations (potentiomètre rotatif mono-tour ou multi-tour) ;
- Mesure de débit de fluide : Le débit du fluide exerce une force sur un clapet relié au curseur d'un potentiomètre. La tension en sortie du potentiomètre augmente avec la vitesse d'écoulement.



b) Capteur à jauges d'extensiométrie.

2- Capteurs à jauges d'extensiométrie

a- Principe

La résistance d'un conducteur est donnée par la relation :

$$R = \rho \frac{l}{S}$$

résistivité $\Omega \cdot m$
longueur m
surface m^2

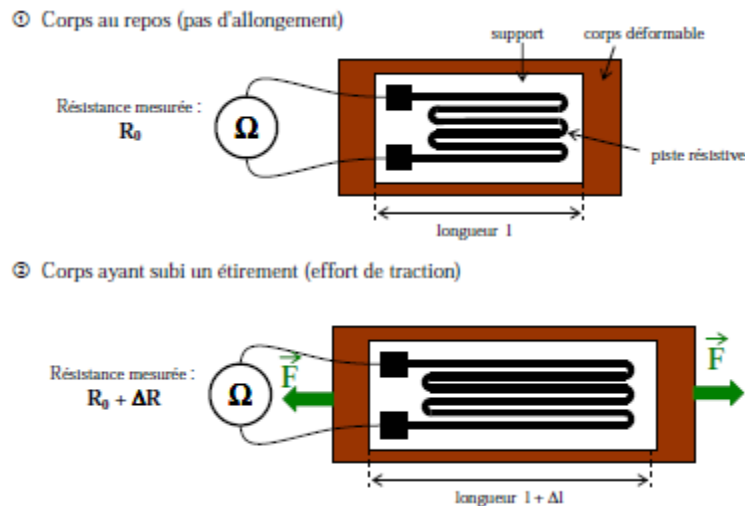
La déformation du conducteur (jauge) modifie la longueur l entraînant une variation de la résistance R .

La relation générale pour les jauges est $\frac{\Delta R}{R_0} = K \frac{\Delta l}{l}$ où K est le facteur de jauge.

Fonctionnement d'une jauge simple.

La jauge est constituée d'une piste résistive collée sur un support en résine. Le tout est collé sur le corps dont on veut mesurer la déformation.

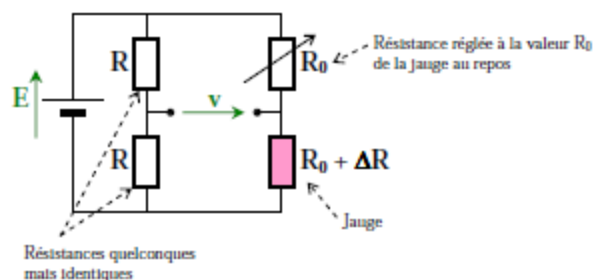
Corps au repos (pas d'allongement).



Remarque : Dans le cas d'une contraction, la résistance de la jauge serait $R_0 - \Delta R$.

Conditionneur de signal (pont de Wheatstone).

La jauge étant un composant purement résistif, il faut l'associer à un circuit électrique pour obtenir une tension image de la déformation. Le circuit souvent utilisé est appelé "pont de Wheatstone". Il est ici constitué d'un générateur de tension associé à 4 résistances dont une est la jauge (schéma ci-dessous) :



La tension de sortie v du pont a l'expression suivante :

$$v = E \left[\frac{R_0 + \Delta R}{R_0 + R_0 + \Delta R} - \frac{R}{R + R} \right] = E \left[\frac{R_0 + \Delta R}{2R_0 + \Delta R} - \frac{1}{2} \right] = E \left[\frac{2R_0 + 2\Delta R - 2R_0 - \Delta R}{4R_0 + 2\Delta R} \right]$$

$$\Rightarrow v = E \frac{\Delta R}{4R_0 + \Delta R}$$

En général, la variation ΔR est petite devant R_0 ; la relation se simplifie alors pour devenir quasi-linéaire :

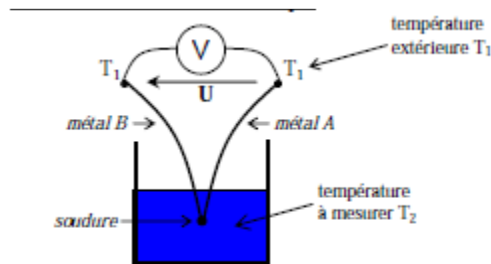
$$v = E \frac{\Delta R}{4R_0}$$

Remarque :

On peut améliorer la sensibilité et la linéarité du dispositif en utilisant un pont à 2 résistances et 2 jauges symétriques $R_0 + \Delta R$ et $R_0 - \Delta R$. Il est même possible d'utiliser un pont à 4 jauges symétriques pour avoir une parfaite linéarité.

I.6.3.5. Capteurs de température.

a) Thermomètre à thermocouple.



On constate que si la température T_2 est différente de T_1 alors il apparaît une tension U aux bornes des deux fils soumis à la température T_1 . Le phénomène inverse est aussi vrai : si on applique une tension, alors il y aura un échauffement ou un refroidissement au point de liaison des deux conducteurs (modules à effet Peltier).

Application : Mesure des hautes températures (900 à 1300°C).

b) Thermistance.

Une thermistance est un composant dont la résistance varie en fonction de la température. En première approximation, la relation entre résistance et température est la suivante :

$$R_{\theta} = R_0 (1 + a\theta)$$

R_{θ} est la résistance à la température θ

R_0 est la résistance à la température 0°C

a est le coefficient de température.

Remarque : si $a > 0$ alors on a une thermistance CTP ($R \nearrow$ quand $\theta \nearrow$)
si $a < 0$ alors on a une thermistance CTN ($R \searrow$ quand $\theta \nearrow$).

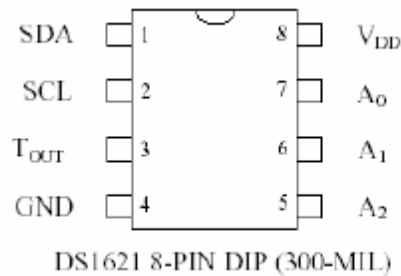
Utilisation : On insère la thermistance dans un pont de jauge.
On obtient ainsi une tension V en sortie du pont $V = k (\theta - \theta_0)$.
Si on prend $\theta_0 = 0^\circ\text{C}$, on obtient $V = k\theta$.

On peut aussi alimenter la thermistance avec un générateur de courant.
La tension à ses bornes sera donc proportionnelle à la résistance.

c) Capteurs à sortie numérique directe.

On trouve actuellement sur le marché, des capteurs de température à sortie numérique directe de type série. Il s'agit notamment des capteurs *DALLAS* qui sont classés en deux catégories :

- Les capteurs à sortie I2C (2 fils) DS1621. Ce capteur DS1621 peut mesurer une température variant de -55°C à 125°C avec une précision de 0,5°C. Pour transmettre la mesure (9 bits), il utilise la norme I2C qui consiste à transmettre en série les bits de mesure sur la ligne SDA en synchronisation avec la ligne SCL (horloge).

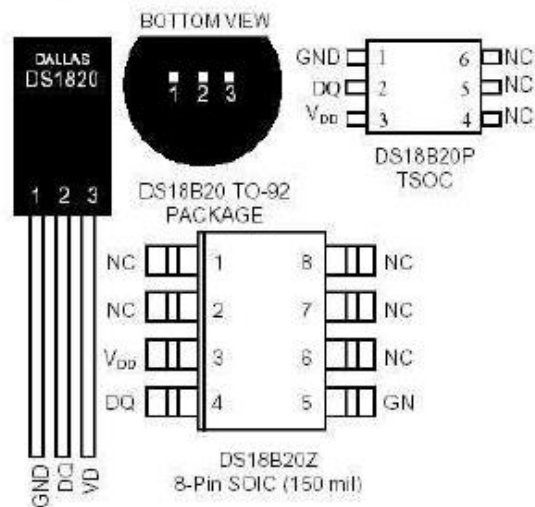
**Table 2. TEMPERATURE/DATA RELATIONSHIPS**

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	01111101 00000000	7D00h
+25°C	00011001 00000000	1900h
+½°C	00000000 10000000	0080h
+0°C	00000000 00000000	0000h
-½°C	11111111 10000000	FF80h
-25°C	11100111 00000000	E700h
-55°C	11001001 00000000	C900h

Le DS1621 possède aussi d'autres fonctions :

- ✓ Il est adressable physiquement sur 3 bits (A0, A1 et A2), ce qui permet d'en utiliser 8 sur la même ligne SDA-SCL.
 - ✓ Il possède une fonction thermostat qui permet de commander un chauffage (températures TH et TL) par l'intermédiaire de la ligne T_{OUT} même lorsque le capteur est déconnecté du matériel informatique.
- *Les capteurs 1 Wire ou i-button* (1 fil) DS1820. Ce capteur DS1820 peut mesurer une température variant de -55°C à 125°C avec une précision maximale de 0,125°C. Pour transmettre la mesure (résolution réglable de 9 à 12 bits), il utilise la norme i-button qui consiste à transmettre en série sur un seul fil, le résultat de la mesure. La ligne VD peut être connectée à la masse GND et la ligne DQ supportera à la fois l'alimentation et la transmission des données, d'où l'appellation 1 Wire. Il suffit donc de deux fils (DQ et GND) pour alimenter et communiquer avec ce capteur. Le DS1820 possède aussi d'autres fonctions :
- ✓ Il est doté d'une adresse (numéro de série) affectée en usine et définitive. Elle est codée sur 8 octets ce qui permet d'utiliser, en théorie, un très grand nombre de DS1820 sur la même ligne.
 - ✓ Une alarme de température peut être paramétrée et la consultation de celle-ci se fait par lecture d'une zone mémoire (adresse – donnée).

PIN ASSIGNMENT



PIN DESCRIPTION

GND - Ground
 DQ - Data In/Out
 V_{DD} - Power Supply Voltage
 NC - No Connect

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C	0000 0101 0101 0000	0550h*
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FF6Fh
-55°C	1111 1100 1001 0000	FC90h

*The power on reset register value is +85°C.

II. SYSTEMES ASSERVIS.

II.1. Introduction.

Le principal de base des systèmes asservis est la rétroaction ou "feedback" : réagir en fonction de ce qui est réalisé, connaissant ce qui est demandé.

Ce principe nous l'utilisons tous les jours dans la plupart de nos actions. Pour conduire, nous devons regarder la route et sans cesse corriger la direction de la voiture même s'il n'y a pas de virages.

II.1.1 Notion de système, boucle ouverte, boucle fermée.

L'automatique peut s'appliquer à tout ce qui bouge, fonctionne, se transforme. L'objet d'application de l'automatique est appelé système. Un système se caractérise par ses grandeurs d'entrée et de sortie. Les grandeurs d'entrée sont les grandeurs qui agissent sur le système. Il en existe de deux types de grandeurs :

- **Commandes** : celles que l'on peut maîtriser ;
- **Perturbations** : celles que l'on ne peut pas maîtriser.

Un système est en boucle ouverte lorsque la commande est élaborée sans l'aide de la connaissance des grandeurs de sortie : il n'y a pas de feedback.

Dans le cas contraire, le système est dit en boucle fermée. La commande est alors fonction de la consigne (la valeur souhaitée en sortie) et de la sortie. Pour observer les grandeurs de sortie, on utilise des capteurs. C'est l'information de ces capteurs qui va permettre d'élaborer la commande.

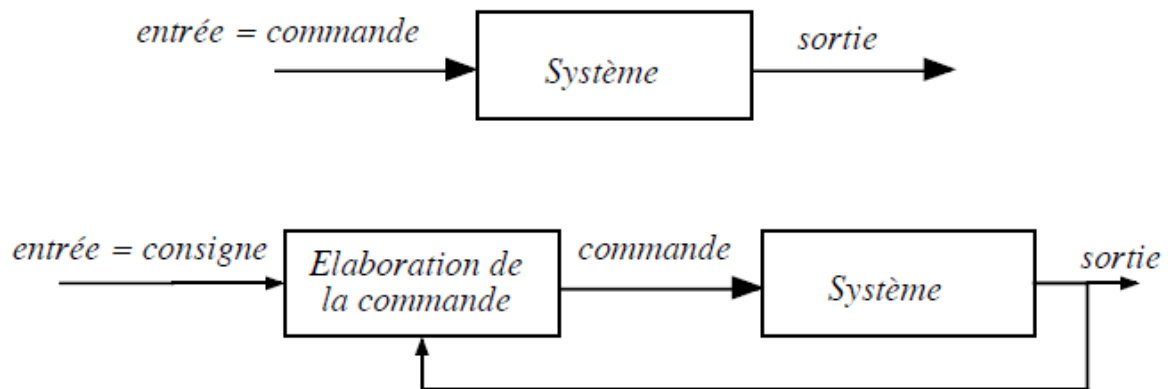


Schéma d'un système en Boucle Ouverte (en haut) et en Boucle Fermée (en bas).

Ce que nous avons vu permet de donner cette autre définition de l'automatique : c'est une science et une technique qui permet de maîtriser le comportement d'un système (traduit par ses grandeurs de sortie), en agissant de manière adéquate sur ses grandeurs d'entrée.

Exemples (Chauffage d'une salle).

Considérons le chauffage électrique d'une salle. Le système est constitué par l'ensemble chauffage + salle. La sortie de ce système est la température de la pièce. La commande du système est la position 0 ou 1 de l'interrupteur. Les perturbations peuvent être l'ouverture d'une fenêtre, de la porte ou les rayons du soleil. En boucle ouverte, la commande est insensible à la sortie.

Pour créer un feedback ou rétroaction, on peut utiliser un thermostat. La commande est alors élaborée en fonction de la consigne (température souhaitée) et de la sortie (température de la pièce).



Schéma de la régulation de la température d'une pièce par un thermostat.

Exceptionnellement, le système de commande peut opérer en boucle ouverte à partir du seul signal de consigne. Mais la boucle fermée (contre réaction) est capable de :

- Stabiliser un système instable en BO ;
- Compenser les perturbations externes ;
- Compenser les incertitudes internes au processus lui-même ;

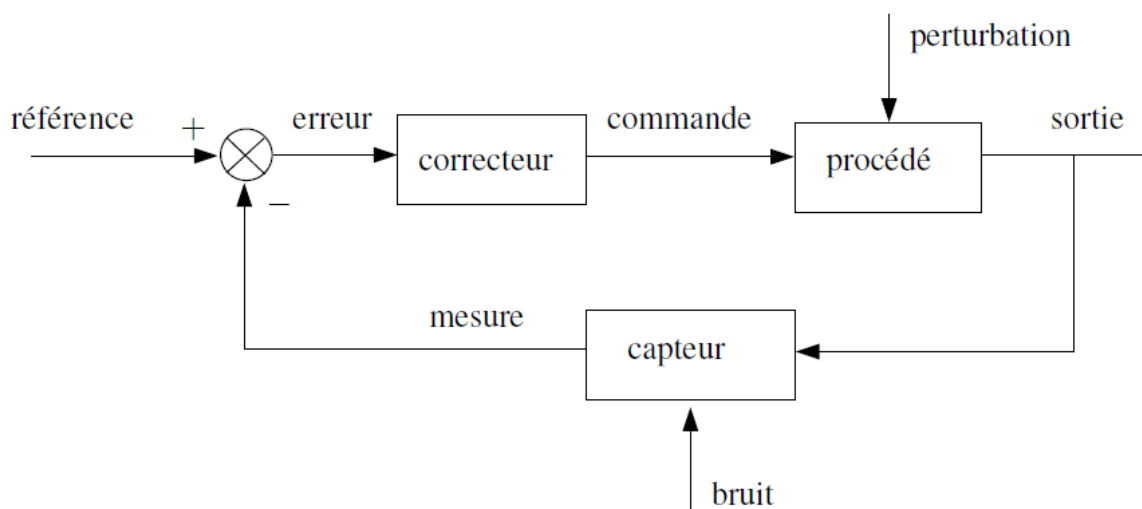
Un système de commande peut réaliser deux fonctions distinctes :

- L'asservissement c'est à dire la poursuite par la sortie d'une consigne variable dans le temps ;
- La régulation c'est à dire la compensation de l'effet de perturbations variables sur la sortie (la consigne restant fixe).

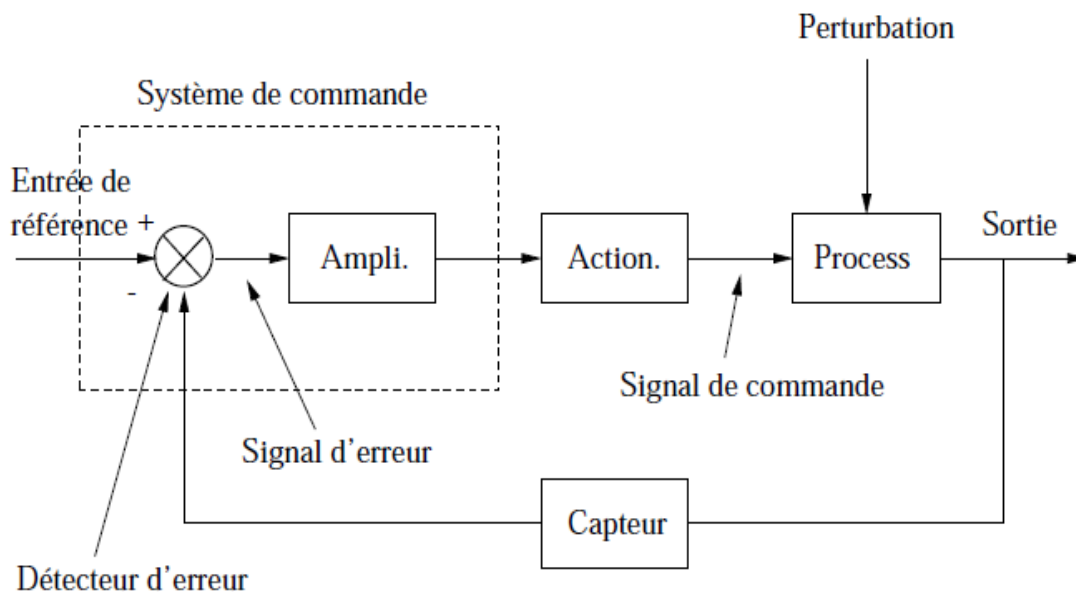
II.2. Constitution d'un système asservi.

En général, l'élaboration de la commande est basée sur :

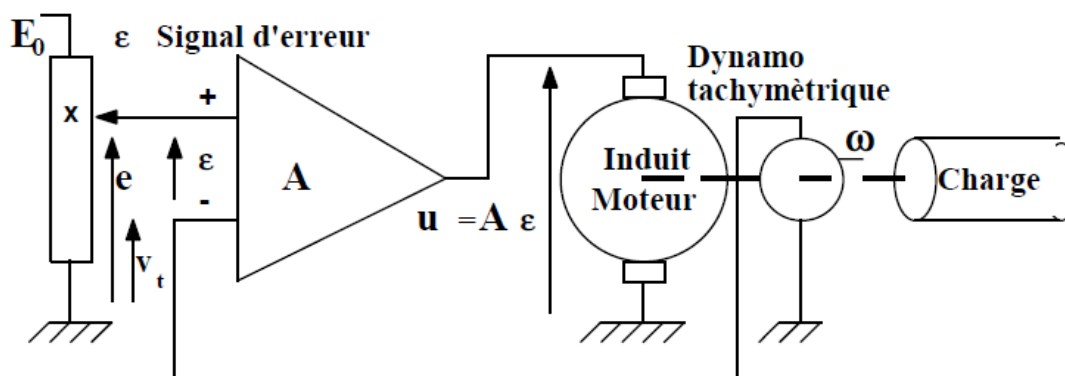
- Un capteur pour mesurer la sortie ;
- Un comparateur entre la consigne et la sortie ;
- Un correcteur qui élabore la commande en fonction de la comparaison précédente, ce qui peut se représenter par la figure ci-dessous.



Le schéma de principe d'un système de commande automatique de vitesse de croisière d'un véhicule est le suivant.



Exemple : Commande de la vitesse de rotation d'un moteur.



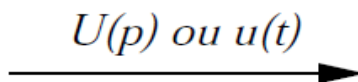
II.3. Schéma fonctionnel d'un système asservi.

Bien qu'utilisés le plus souvent en liaison avec les modèles entrée-sortie du type fonction de transfert, les schémas fonctionnels peuvent aussi servir à donner une image graphique des modèles dans l'espace d'état.

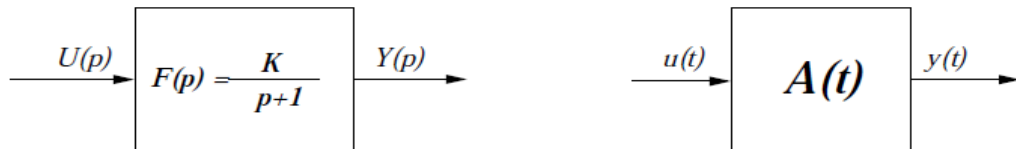
II.3.1. Définition.

Le **schéma fonctionnel** d'un système est une représentation graphique des fonctions de chaque composant élémentaire constituant le système ainsi que le flux des signaux utiles. Un schéma fonctionnel est composé :

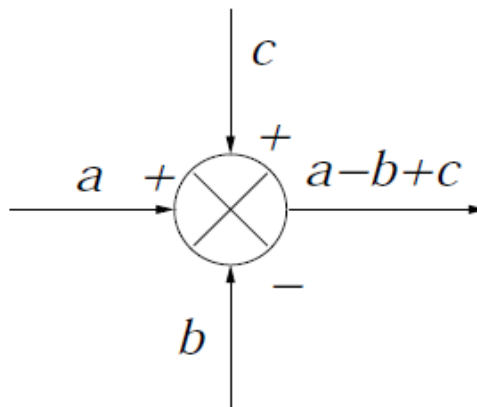
- **D'arcs orientés** qui représente le flux de signaux donnés regroupés dans un vecteur.



- **De blocs fonctionnels** qui sont les symboles représentant l'opération mathématique appliquée à l'entrée du bloc et produisant sa sortie. Cette opération mathématique est très souvent mais pas exclusivement représentée par une fonction de transfert. La figure ci-dessous représente à gauche un bloc fonctionnel associé à une fonction de transfert $Y(p) = F(p)U(p)$ alors que celui de droite est associé à l'opération de multiplication de l'entrée par une matrice $y(t) = A(t)u(t)$.



- **De blocs sommateurs** qui traduit une relation purement algébrique entre les signaux d'entrée et de sortie.



Remarques.

- Un schéma fonctionnel n'est pas unique.
- Il est indépendant de la nature physique du système modélisé.

II.3.2. Schéma fonctionnel et fonctions de transfert.

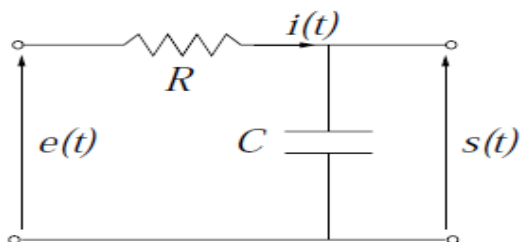
Le schéma fonctionnel est très étroitement associé aux fonctions de transfert. Ainsi, une procédure systématique de tracé du schéma fonctionnel d'un système donné peut être proposée.

Principes de construction.

1. Ecrire les équations de la physique associées à chaque élément constituant le système.
2. En appliquant la transformée de Laplace, calculer la fonction de transfert associée à chaque élément en supposant les conditions initiales nulles.
3. Identifier les relations inter-signaux et les relations signaux-blocs pour tracer le schéma fonctionnel.

Exemple.

Soit le circuit électrique RC.



1- Les équations électriques donnent :

$$e(t) - s(t) = Ri(t)$$

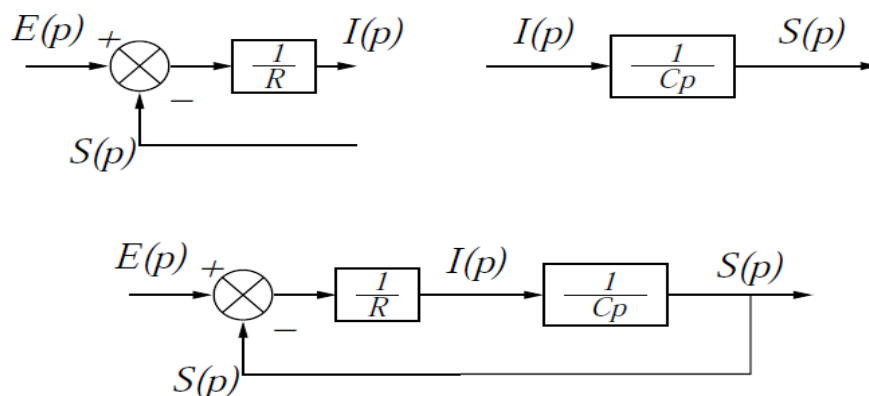
$$s(t) = \frac{1}{C} \int_0^t i(\tau) d\tau$$

2- Application de la transformée de Laplace et détermination des fonctions de transfert élémentaires :

$$V_R(p) = E(p) - S(p) = RI(p)$$

$$S(p) = \frac{1}{Cp} I(p)$$

3- Tracé du schéma fonctionnel :



La représentation d'état linéaire temps-variant suivante a pour équivalent le schéma fonctionnel de la figure ci-dessous.

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t) + D(t)u(t)$$

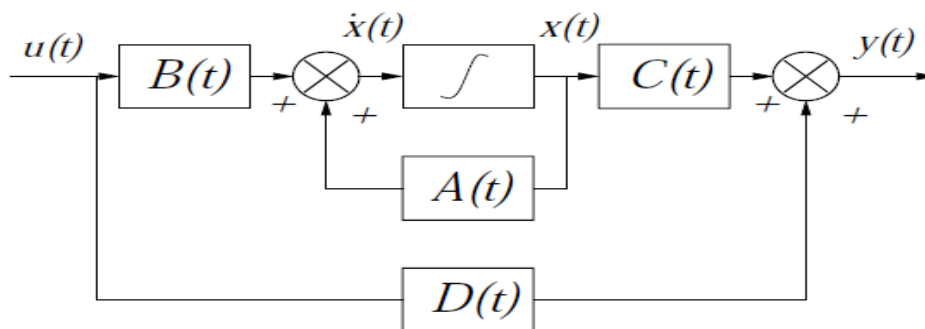


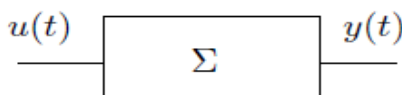
Schéma fonctionnel d'une représentation d'état.

II.4. Fonction de transfert des systèmes linéaires.

Soit un système linéaire invariant d'entrée $u(t)$ et de sortie $y(t)$. La réponse du système peut s'écrire $y(t) = g(t).u(t)$ avec $g(t)$ la réponse impulsionnelle du système. En appliquant la transformée de Laplace à cette équation on obtient $Y(s) = G(s).U(s)$. En variable de Laplace la relation entrée-sortie n'est donc plus un produit de convolution, mais un simple produit.

On appelle fonction de transfert du système la transformée de Laplace $G(s)$ de la réponse impulsionnelle qui est le rapport des transformées de Laplace de la sortie et de l'entrée. Le terme de transmittance, synonyme de fonction de transfert est souvent utilisé.

$$G(s) = \frac{Y(s)}{U(s)}$$



II.4.1. Fonction de transfert en boucle ouverte et en boucle fermée.

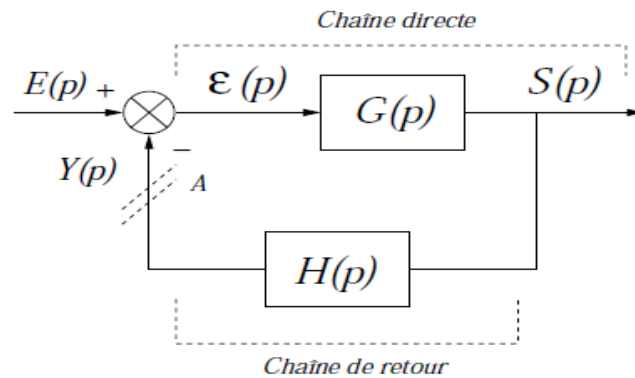
Soit le schéma de la figure ci-dessous. On définit $E(p)$ comme l'entrée de consigne, $Y(p)$ comme la sortie mesurée et p comme l'erreur.

La fonction de transfert en boucle ouverte ou gain de boucle correspond au transfert si la boucle est ouverte en A. Elle est définie comme suit :

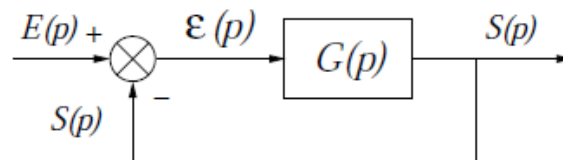
$$\frac{Y(p)}{E(p)} = G(p)H(p)$$

La fonction de transfert en boucle fermée correspond au transfert "global" de la boucle d'asservissement. Elle est définie comme suit :

$$\frac{S(p)}{E(p)} = \frac{G(p)}{1 + G(p)H(p)}$$



Un cas particulier que l'on rencontre fréquemment est celui des systèmes bouclés à retour unitaire.



Un système bouclé est dit à **retour unitaire** si le transfert de la chaîne de retour est égal à un. La fonction de transfert en boucle ouverte est alors égale au transfert de la chaîne directe $G(p)$ et la fonction de transfert en boucle fermée est donnée par :

$$\frac{S(p)}{E(p)} = \frac{G(p)}{1 + G(p)}$$

II.5. Relations fondamentales des systèmes linéaires.

Nous avons ci-dessous les relations fondamentales de quelques systèmes linéaires.

2.1.1 En mécanique

Exemple 2.1.1 (Wagon). D'après le principe fondamental de la mécanique, l'équation du système est :

$$m\ddot{x}(t) = u(t).$$

Exemple 2.1.2 (Oscillateur harmonique).

$$m\ddot{x}(t) + kx(t) = u(t).$$

Exemple 2.1.3 (Pendule oscillant (en robotique : bras articulé)).

$$\ddot{\theta}(t) + mgl \sin \theta(t) = u(t).$$

2.1.2 Electricité

Exemple 2.1.8 (Vitesse angulaire d'un rotor).

$$I\dot{\omega}(t) = u(t).$$

Exemple 2.1.9 (Circuit RLC).

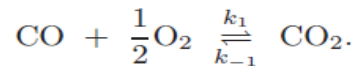
$$L \frac{di}{dt} + Ri + \frac{q}{C} = u,$$

où $q(t) = \int^t i$ est la charge du condensateur. D'où :

$$\begin{cases} \frac{dq}{dt} = i, \\ \frac{di}{dt} = -\frac{R}{L}i - \frac{1}{LC}q + \frac{1}{L}u. \end{cases}$$

2.1.3 Chimie

Exemple 2.1.11 (Cinétique chimique). Considérons l'équation



Les équations de la cinétique chimique sont :

$$\begin{cases} \frac{d[\text{CO}]}{dt} = -k_1[\text{CO}][\text{O}_2]^{1/2} + k_{-1}[\text{CO}_2], \\ \frac{d[\text{O}_2]}{dt} = -\frac{1}{2}k_1[\text{CO}][\text{O}_2]^{1/2} + \frac{1}{2}k_{-1}[\text{CO}_2], \\ \frac{d[\text{CO}_2]}{dt} = k_1[\text{CO}][\text{O}_2]^{1/2} - k_{-1}[\text{CO}_2], \end{cases}$$

où k_1 et k_{-1} dépendent de la température T , et le contrôle est $u = \frac{dT}{dt}$.

2.1.4 Biologie

Exemple 2.1.12 (Système proies-prédateurs).

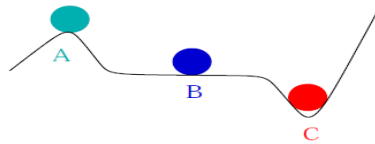
$$\begin{cases} \dot{x} = x(1 - y) + u, \\ \dot{y} = -y(1 - x). \end{cases}$$

Exemple 2.1.13 (Bioréacteur).

$$\begin{cases} \dot{x} = \text{croissance} - \text{sortie} = \mu(s) - Dx, \\ \dot{y} = \text{consommation} - \text{sortie} + \text{alimentation} = -\nu(s)x - Ds + Ds_{in}. \end{cases}$$

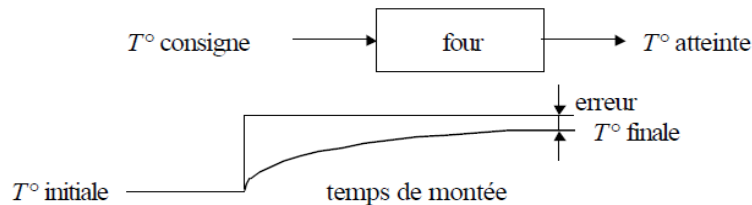
II.6. Stabilité.

Ces concepts de stabilité d'un point d'équilibre peuvent être illustrés sur la figure ci-dessous. La position A correspond à un point d'équilibre instable, la position B à un point d'équilibre stable et la position C à une position d'équilibre asymptotiquement stable.



Quand le système est mis en équation, il est beaucoup plus facile d'évaluer ses performances. Les performances du système sont obtenues en mesurant la **rapidité** de prise en compte de la commande et la **précision** atteinte en sortie.

Exemple:



II.6.1. Réponse impulsionnelle.

C'est la réponse du système lorsque le signal d'entrée est une impulsion. On note cette réponse $h(t)$. Elle contient toute l'information nécessaire sur le système. Cependant, l'utilisation de cette réponse s'accompagne de quelques difficultés pratiques car on ne sait pas toujours réaliser une impulsion de durée très brève vis à vis des constantes de temps mises en jeu dans le système étudié.

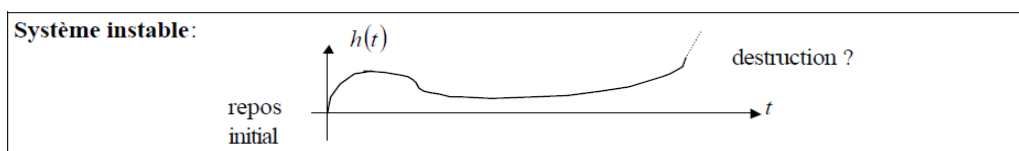
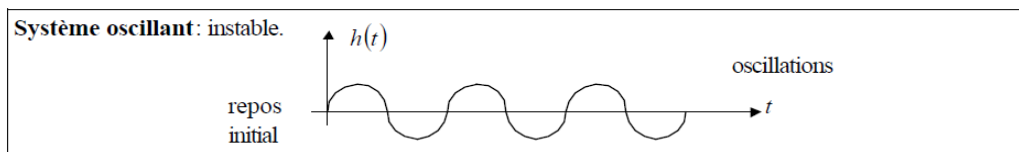
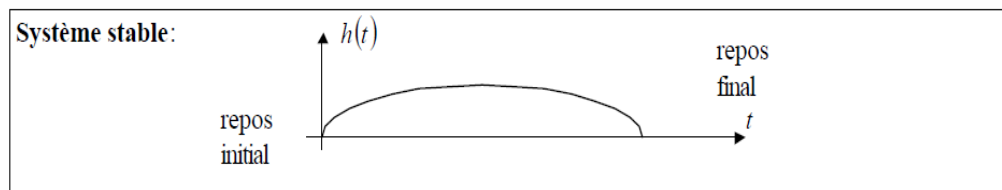
Ainsi, malgré que la réponse impulsionnelle ne permet pas d'évaluer les performances d'un système, Elle permet d'introduire la notion de stabilité.

II.6.2. Stabilité.

Un système est dit stable, si sa réponse impulsionnelle est le siège d'un régime amorti :

$$\lim_{t \rightarrow \infty} h(t) = 0$$

Exemples:



On ne doit pas oublier, en effet, que commander correctement un système, c'est avant tout éviter qu'il ne devienne instable.

III. SYSTEME AUTOMATISE COMBINATOIRE.

Tous les circuits numériques fonctionnent en mode binaire, un mode dans lequel les signaux ne peuvent prendre que deux valeurs : 0 et 1. Ces valeurs correspondent à des plages de tension définies à l'avance. Pour étudier les systèmes logiques, il est nécessaire donc de posséder:

- Un outil mathématique, c'est l'algèbre de Boole. Dans cet ensemble de lois mathématiques, il n'y a que deux constantes que nous désignerons par 0 et 1. Ces symboles 0 et 1 représentent deux ETATS et non deux chiffres. On utilise aussi H pour high (haut) et L pour low (bas).
- Un outil physique pour matérialiser les fonctions de base utilisées, ce sont les circuits logiques.

Les variables sont des grandeurs qui ne peuvent prendre que deux états (0 ou 1). Comme en algèbre ordinaire, on symbolise ces variables par des lettres, par exemple :

- Variables d'entrée : A, B, C, D, X, Y, etc...
- Variables de sortie : F ⇒ Fonction S ⇒ Sortie L ⇒ Lampe M ⇒ Moteur, etc...

Une expression booléenne est une association de variables liées par des signes d'opérations.

Exemple: $S = A.B + C$. Lire S égale A et B ou C car en algèbre de Boole la multiplication est égale à ET, alors que la somme repr+ = OU (sera expliqué plus loin).

Lorsque l'état des sorties d'un système logique ne dépend uniquement que de l'état des entrées et non du passé du système, on parle de **LOGIQUE COMBINATOIRE**, dans le cas contraire, on parle de **LOGIQUE SEQUENTIELLE**.

III.1. Systèmes de numération.

Le système de numération est la représentation d'une grandeur numérique par des symboles. Le nombre de symboles utilisés caractérise le numéro de la base. Celui que nous connaissons le mieux est le système décimal mais nous allons aussi définir les systèmes binaire, octal, hexadécimal.

Le système de numération binaire est le plus important pour les circuits numériques ; car il est le seul que ces circuits soit capable d'utiliser. Il ne faut pour autant pas négliger l'importance des autres systèmes. Le système décimal revêt de l'importance en raison de son utilisation universelle pour représenter les grandeurs du monde courant.

III.1.1. Le système de numération binaire ou base 2.

Les systèmes électriques et électroniques sont caractérisés par deux états :

- interrupteur : **ouvert** ou **fermé** ;
- transistor : **bloqué** ou **saturé**.

De cette constatation est née l'idée d'utiliser le système à base 2 ou système binaire. La base 2 n'utilise que deux symboles : 0 et 1. L'équivalence avec les circuits électriques se fera ainsi :

- 0 = interrupteur ouvert aucun courant ne peut circuler, la lampe est éteinte correspond au transistor bloqué.
- 1 = interrupteur fermé le courant peut circuler, la lampe est allumée correspond au transistor saturé.

III.1.2. Le système hexadécimal.

Il est très employé, surtout en informatique. C'est un système numérique ayant pour base 16. On l'utilise pour l'écriture condensée de nombres binaires. Les 16 symboles sont composés de dix chiffres (de 0 à 9) et de six lettres majuscules (de A à F).

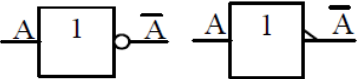
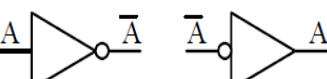
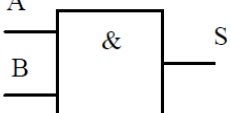
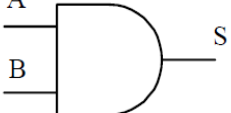
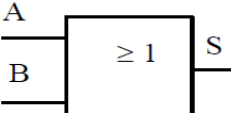
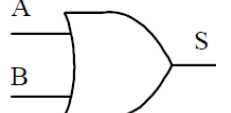
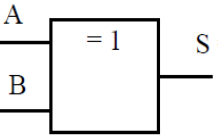
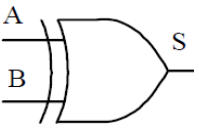
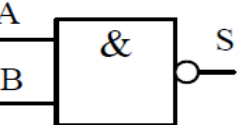
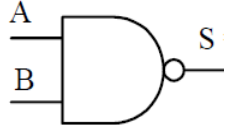
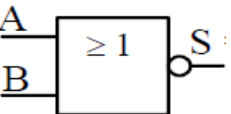
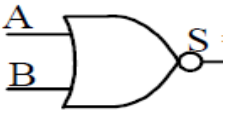
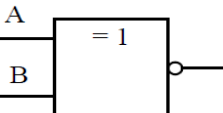
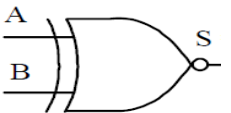
Les symboles hexadécimaux A à F correspondent aux valeurs décimales 10 à 15. Un caractère hexadécimal représente un mot binaire de 4 bits. Cette écriture est de loin plus pratique qu'une suite de 1 et de 0.

Exemple : $01101011_2 = 6B_{16}$;

$11111111_2 = FF_{16}$

III.2. Les opérateurs logiques de base.

L'opérateur de base de tout circuit numérique est la porte logique. C'est un circuit, représenté par un symbole, qui exprime une relation d'opération entre ses entrées. Il existe deux normes de représentation des symboles des portes logiques : la norme US (américaine) et la norme CEI (commission électrotechnique internationale).

Symboles		Equation	Table de Vérité															
US	CEI																	
NO 	NON 	$S = \bar{A}$	<table border="1"> <thead> <tr> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	S	0	1	1	0									
A	S																	
0	1																	
1	0																	
AND 	ET 	$S = A.B$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1
A	B	S																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR 	OU 	$S = A + B$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1
A	B	S																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
XOR 	OU Exclusif 	$S = A \oplus B$ $= \bar{A}.B + A.\bar{B}$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0
A	B	S																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
NND 	NON ET 	$S = \overline{A.B}$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0
A	B	S																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR 	NON OU 	$S = \overline{A + B}$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0
A	B	S																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XNOR 	NON OU Ex 	$S = \overline{A \oplus B}$ $= \bar{A}.\bar{B} + A.B$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	1
A	B	S																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

III.3. Algèbre de Boole.

L'algèbre Boole élabore les règles qu'on utilise pour simplifier les expressions logiques et, par le fait même, les circuits logiques.

III.3.1. Théorèmes de Boole.

$$\begin{array}{llll} X \cdot X = X ; & X \cdot \bar{X} = 0 ; & X + 1 = 1 ; & X + \bar{X} = 1 ; \\ X + XY = X ; & X + YZ = (X + Y)(X + Z) ; & X + \bar{X}Y = X + Y \end{array}$$

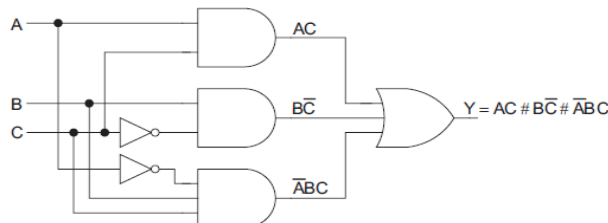
III.3.1. Théorèmes de De Morgan.

$$\overline{X + Y} = \bar{X} \cdot \bar{Y} ; \quad \overline{X \cdot Y} = \bar{X} + \bar{Y}$$

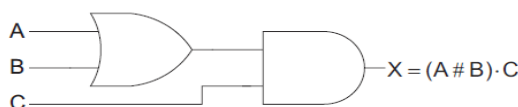
III.4. Elaboration d'un logigramme.

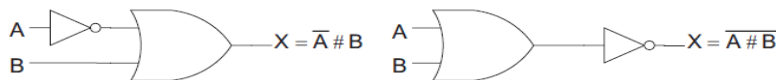
Il est possible de tracer directement un diagramme logique à partir d'une expression booléenne. Si l'opération d'un circuit est définie par une expression booléenne, il est possible de tracer directement un diagramme logique à partir de cette expression. Par exemple, si on a besoin d'un circuit tel que $X = ABC$, on sait immédiatement qu'il nous faut une porte ET à trois entrées. Le raisonnement qui nous a servi pour ces cas simples peut être étendu à des circuits plus complexes.

Supposons que l'on veuille construire un circuit dont la sortie est $S = AC + B\bar{C} + \bar{A}BC$. Cette expression booléenne est constituée de trois termes : AC ; $B\bar{C}$; $\bar{A}BC$ qui sont additionnés logiquement. On déduit qu'il nous faut une porte OU à trois entrées auxquelles sont appliqués respectivement les signaux AC ; $B\bar{C}$; $\bar{A}BC$. Chaque entrée de la porte OU est un produit logique, ce qui signifie qu'il a fallu trois portes ET avec les entrées appropriées pour produire ces termes. C'est ce qu'on peut voir à la figure ci-dessous où est tracé le schéma final du circuit.



Inversement, tout circuit logique, quelle que soit sa complexité, peut être décrit au moyen des opérations booléennes déjà décrites. A titre d'exemple, considérons le circuit de la figure ci-dessous comprenant trois entrées A, B et C et une seule sortie X. En recourant à l'expression booléenne de chacune des portes, on peut facilement trouver l'équation correspondant à la sortie.





III.5. Synthèse d'un système logique combinatoire.

Les étapes de conception d'un circuit logique combinatoire à partir d'un cahier de charge sont les suivantes :

1. Construire la table de vérité selon le cahier de charge.
2. Ecrire l'expression Booléenne relative à la table de vérité.
3. Simplifier l'équation.
4. Réaliser le schéma de l'équation simplifiée.

III.5.1. Construction de la table de vérité.

Fréquemment, les cahiers de charge d'un circuit logique à concevoir sont donné sous forme textuelle. Cette forme n'est pas utilisable pour déterminer l'équation du circuit. La table de vérité permet de spécifier le fonctionnement du circuit. Construire une table de vérité c'est donc traduire une donnée de problème ou un cahier de charge d'un client en un tableau attribuant des 1 ou des 0 aux comme valeur des variables d'entrée et de sortie. Il faut être donc capable d'identifier les entrées et sorties du système ensuite prévoir toutes les éventualités sur les variables d'entrée. On prendra donc soin d'analyser toutes les possibilités des variables d'entrées. C'est certainement le point le plus important de la conception. Une table de vérité ne peut être correcte que si les données sont parfaitement comprises.

Les variables d'entrée ne sont autres que les conditions liées au fonctionnement du système. Les variables de sorties sont quant à les résultats attendus. Un système qui a n variables d'entrées présentera un maximum de 2^n éventualités (ou combinaisons).

III.5.2. Ecriture de l'expression Booléenne.

La procédure est la suivante :

- a) Pour chaque cas de la table qui donne 1 en sortie, on écrit le produit logique (terme ET) qui lui correspond. On doit retrouver toutes les variables d'entrée dans chaque terme ET soit sous forme directe soit sous forme complémentée. Dans un cas particulier, si la variable est 0, alors son symbole est complémenté dans le terme ET correspondant.
- b) On somme logiquement (opérateur OU) ensuite tous les produits logiques constitués, ce qui donne l'expression définitive de la sortie.

Exemple :

No	C	B	A	X	Equation minterme
0	0	0	0	0	
1	0	0	1	0	
2	0	1	0	1	$\bar{C}B\bar{A}$
3	0	1	1	1	$\bar{C}BA$
4	1	0	0	0	
5	1	0	1	0	
6	1	1	0	0	
7	1	1	1	1	CBA

$$X = \bar{C}B\bar{A} + \bar{C}BA + CBA$$

III.8.3. Simplification d'une équation logique.

Dès qu'on dispose de l'expression d'un circuit logique, il peut être possible de la minimiser pour obtenir une équation comptant moins de termes ou moins de variables par terme. Cette

nouvelle équation peut alors servir de modèle pour construire un circuit entièrement équivalent au circuit original mais qui requiert moins de portes. L'objectif est donc de réaliser le circuit le plus simple possible. Il sera moins cher, moins encombrant, moins gourmand en énergie et souvent plus rapide!

Une première façon est fondée sur l'application des théorèmes de l'algèbre booléenne (simplification algébrique); cette façon, comme nous le verrons, dépend énormément de l'instinct et de l'expérience. L'autre façon (les tables de Karnaugh), au contraire, suit une démarche systématique (*voir cours circuit logique combinatoire*).

La simplification algébrique se fait en utilisant les théorèmes de Boole et de De Morgan.

L'équation $Y = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$ peut par exemple être simplifiée de la manière suivante: Le terme ABC comporte deux variables communes avec chacun des autres termes. L'algèbre de Boole nous autorise à ajouter deux autres termes ABC et de mettre en facteur:

$$y = \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

$$y = BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$$

$$y = BC + AC + AB$$

$$y = C(B + A) + AB$$

III.6. Les fonctions standards combinatoires.

Lorsque le problème à résoudre comprend plus de 4 ou 5 entrées la méthodologie avec la table de vérité et la table de Karnaugh n'est plus applicable car il sera aussi difficile d'établir une table de vérité lorsque le nombre d'entrées devient important. Si nous avons un système avec 10 entrées, la table de vérité aura 1024 lignes! Nous devons donc adapter notre méthodologie. Il s'agira ainsi de décomposer notre problème en identifiant des sous-fonctions qui sont les principales fonctions standards combinatoires. Ces fonctions sont nommées aussi fonctions MSI pour "Medium Scale Integration". Il s'agit de fonctions plus complexes que les simples portes logiques. Ces fonctions ont été très rapidement intégrées dans des circuits des familles TTL ou CMOS. Elles présentaient l'avantage d'être moins onéreuses sous la forme d'un circuit.

Voici une liste non exhaustive des principales fonctions standards combinatoires :

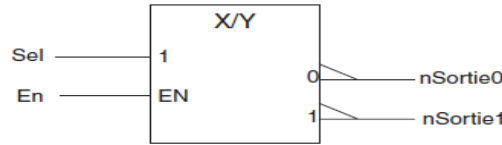
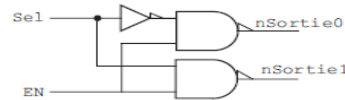
- ✓ Le décodeur (X/Y) ;
- ✓ L'encodage de priorité ;
- ✓ le transcodage de nombres (BIN→BCD, BCD→BIN, BCD→7SEG, etc.) ;
- ✓ Le multiplexeur (MUX) ;
- ✓ Le démultiplexeur (DMUX) ;
- ✓ Le comparateur (COMP) ;
- ✓ les opérateurs arithmétiques (additionneur, soustracteur, ...).

III.6.1. Décodeur (X/Y).

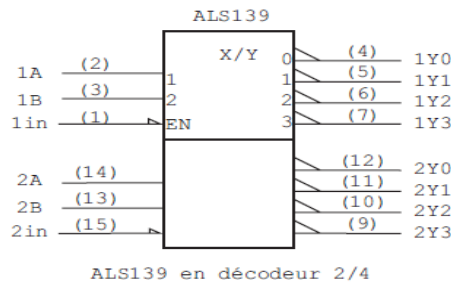
Un décodeur permet d'identifier la combinaison qui est active. Nous parlerons du décodage de la combinaison d'entrée X. Il comporte une entrée de X bits à décoder (sélection), une entrée de validation (*enable*), et $2^X (=Y)$ sorties. La sortie, dont le numéro correspond à la valeur codée donnée en entrée, sera activée si l'*enable* est actif. Toutes les autres sorties sont inactives. Le décodeur n'a qu'une seule sortie active à la fois. Dans le cas où l'entrée de validation est inactive toutes les sorties sont alors inactives.

La figure ci-dessous nous donne un exemple de décodeur 1 à 2. Ce décodeur dispose d'une entrée de 1 bit et d'une entrée de validation. Il dispose donc 2 sorties (2^1).

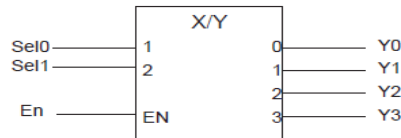
EN	Sel	Sortie1	Sortie0
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0



Exemple de décodeur de commerce : Le 74139 de la figure ci-dessous avec sa table de vérité. Il est alimenté à partir des broches 8 (GND) et 16 (V_{CC}).



EN	Sel1	Sel0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

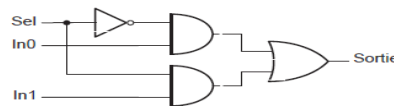


III.6.2. Multiplexeur (MUX).

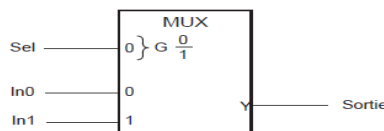
Un multiplexeur est un système combinatoire qui met sur sa sortie unique la valeur d'une de ses 2ⁿ entrées de données, le numéro de l'entrée sélectionnée étant fourni sur les n entrées de commande.

Voici la table de vérité d'un multiplexeur 2 vers 1 (2 to 1), son schéma logique et le symbole CEI de cette fonction.

Sel	In1	In0	Sortie
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



TDV compacte	
Sel	Sortie
0	In0
1	In1

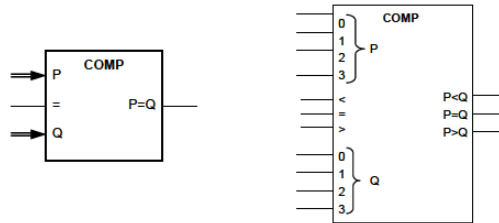


III.9.3. Comparateur.

La fonction de comparaison de deux nombres binaires est très fréquemment utilisée. Un comparateur est un circuit qui indique si deux nombres binaires sont plus grands, égaux ou

plus petits. Dans le cas d'un circuit modulaire, il y a trois entrées afin de savoir si les bits précédents sont plus grands, égaux ou plus petits.

La **figure 5- 20 a)** nous donne le symbole CEI d'un comparateur d'égalité avec l'entrée '=' pour la chaînage. La **figure b)** nous donne le symbole CEI d'un comparateur 4 bits modulaire avec les trois sorties '<', '=' et '>'.



III.6.4. Additionneur binaire.

Les ordinateurs ne peuvent additionner que deux nombres binaires à la fois, chacun de ces nombres pouvant avoir plusieurs bits. La figure ci-dessous illustre l'addition de deux nombres de 5 bits.

cumulande	1	0	1	0	1	Mémo
						risé dans l'accumulateur
	+					
cumulateur	0	0	1	1	1	Mémo
						risé dans le registre B
somme	1	1	1	0	0	
report	0	0	1	1	1	

IV. LES SYSTEMES AUTOMATISES SEQUENTIELS.

Faisant suite à notre étude de la logique combinatoire, nous abordons maintenant la logique SEQUENTIELLE. Souvenons-nous qu'en logique combinatoire, nous avons la situation suivante :

- A chaque état des variables d'entrée correspond un seul état des variables de sortie et inversement.

La situation est différente avec les circuits séquentiels :

- A un état des variables d'entrée peut correspondre plusieurs états différents des variables de sortie parce que le circuit se souvient de ses états précédents.

Ainsi les sorties dépendent non seulement des entrées mais également de l'état du système.

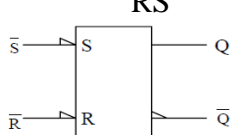
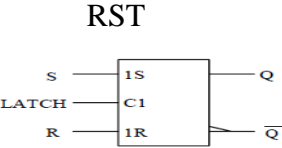
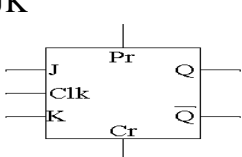
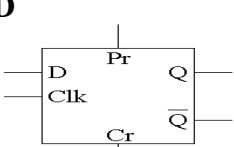
La logique séquentielle permet donc de réaliser des circuits dont le comportement est variable avec le temps. L'état d'un système constitue ainsi une mémoire du passé.

IV.1. Fonctions logiques séquentielles de base.

Les circuits séquentiels complexes sont constitués de circuits séquentiels élémentaires et de circuits combinatoires. Parmi les principaux, citons :

- ✓ Les fonctions mémoires ;
- ✓ Les fonctions de comptage ;

La fonction mémoire élémentaire est réalisée par les BASCULES. Lorsque les changements d'état des divers composants d'un circuit séquentiel se produisent à des instants qui dépendent des temps de réponse des autres composants et des temps de propagation des signaux on parle de logique séquentielle asynchrone. Cependant les retards peuvent ne pas être identiques pour toutes les variables binaires et conduire à certains aléas. Ceux-ci peuvent être évités en synchronisant la séquence des diverses opérations logiques sur les signaux périodiques provenant d'une horloge. La logique séquentielle est alors dite synchrone : tous les changements d'état sont synchronisés sur un signal de contrôle.

Symboles de la bascule	Table de Vérité																				
<p>RS</p> 	<table border="1"> <thead> <tr> <th>S</th> <th>R</th> <th>Q</th> <th>\bar{Q}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q</td> <td>\bar{Q}</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table> <p>Sorties inchangées Set : Remise à Un : RAU Reset : Remise à Zéro : RAZ A proscrire</p>	S	R	Q	\bar{Q}	0	0	Q	\bar{Q}	1	0	1	0	0	1	0	1	1	1	0	0
S	R	Q	\bar{Q}																		
0	0	Q	\bar{Q}																		
1	0	1	0																		
0	1	0	1																		
1	1	0	0																		
<p>RST</p> 	<table border="1"> <thead> <tr> <th>S_n</th> <th>R_n</th> <th>Q_{n+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q_n</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>?</td> </tr> </tbody> </table>	S_n	R_n	Q_{n+1}	0	0	Q_n	1	0	1	0	1	0	1	1	?					
S_n	R_n	Q_{n+1}																			
0	0	Q_n																			
1	0	1																			
0	1	0																			
1	1	?																			
<p>JK</p> 	<table border="1"> <thead> <tr> <th>J_n</th> <th>K_n</th> <th>Q_{n+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q_n</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>\bar{Q}_n</td> </tr> </tbody> </table>	J_n	K_n	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	\bar{Q}_n					
J_n	K_n	Q_{n+1}																			
0	0	Q_n																			
0	1	0																			
1	0	1																			
1	1	\bar{Q}_n																			
<p>D</p> 	<table border="1"> <thead> <tr> <th>D</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </tbody> </table>	D	Q	0	0	0	1														
D	Q																				
0	0																				
0	1																				

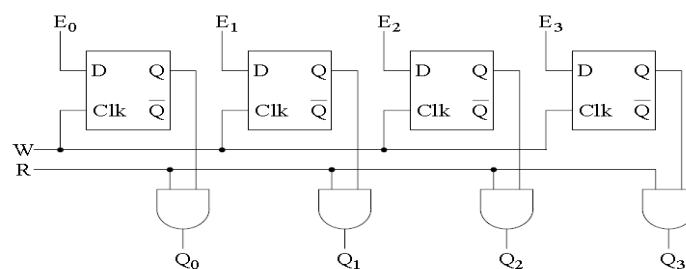
IV.2. Compteurs et Registres.

IV.2.1. Registres.

a) Registre de mémorisation.

Un registre permet la mémorisation de n bits. Il est donc constitué de n bascules, mémorisant chacune un bit. L'information est emmagasinée sur un signal de commande et ensuite conservée et disponible en lecture. La figure 24 donne un exemple de registre 4 bits réalisé avec quatre bascules D.

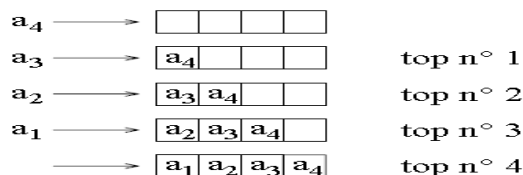
En synchronisme avec le signal d'écriture W le registre mémorise les données présentées sur les entrées E_0 , E_1 , E_2 et E_3 . Elles sont conservées jusqu'au prochain signal de commande W . Dans cet exemple les états mémorisés peuvent être lus sur les sorties Q_0 , Q_1 , Q_2 et Q_3 en coïncidence avec un signal de validation R . Lorsque ces sorties sont connectées à un bus, les portes ET en coïncidence avec ce signal de lecture sont remplacées par des portes à trois états.



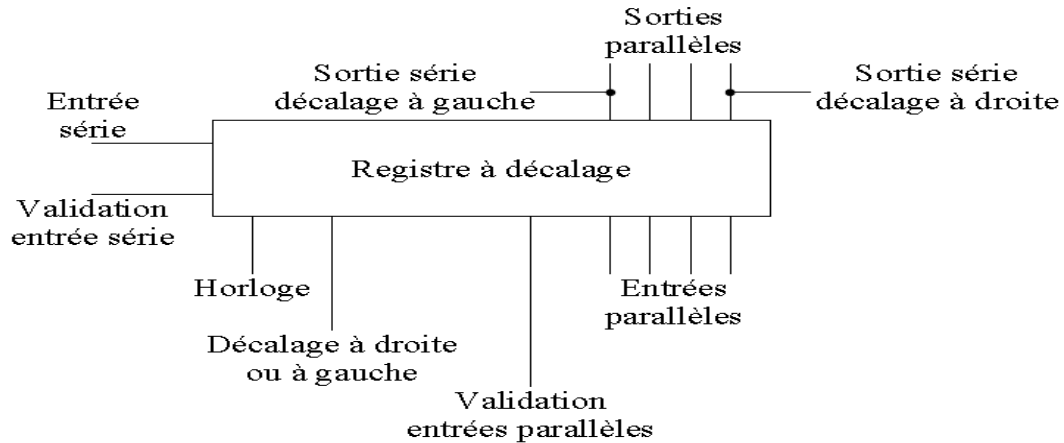
b) Registre à décalage.

Dans un registre à décalage les bascules sont interconnectées de façon à ce que l'état logique de la bascule de rang i puisse être transmis à la bascule de rang $i+1$ (ou $i-1$) quand un signal d'horloge est appliqué à l'ensemble des bascules. L'information peut être chargée de deux manières dans ce type de registre.

- Entrée parallèle : comme dans le cas d'un registre de mémorisation. En général une porte d'inhibition est nécessaire pour éviter tout risque de décalage pendant le chargement parallèle.
- Entrée série : l'information est présentée séquentiellement bit après bit à l'entrée de la première bascule. A chaque signal d'horloge un nouveau bit est introduit pendant que ceux déjà mémorisés sont décalés d'un niveau dans le registre. La figure ci-dessous schématise le chargement d'un registre 4 bits en quatre coups d'horloge.



De même l'information peut être lue en série ou en parallèle. D'autre part, certains registres peuvent être capables de décaler à gauche et à droite. Un registre à décalage universel serait donc constitué des entrées, des sorties et des commandes suivantes :



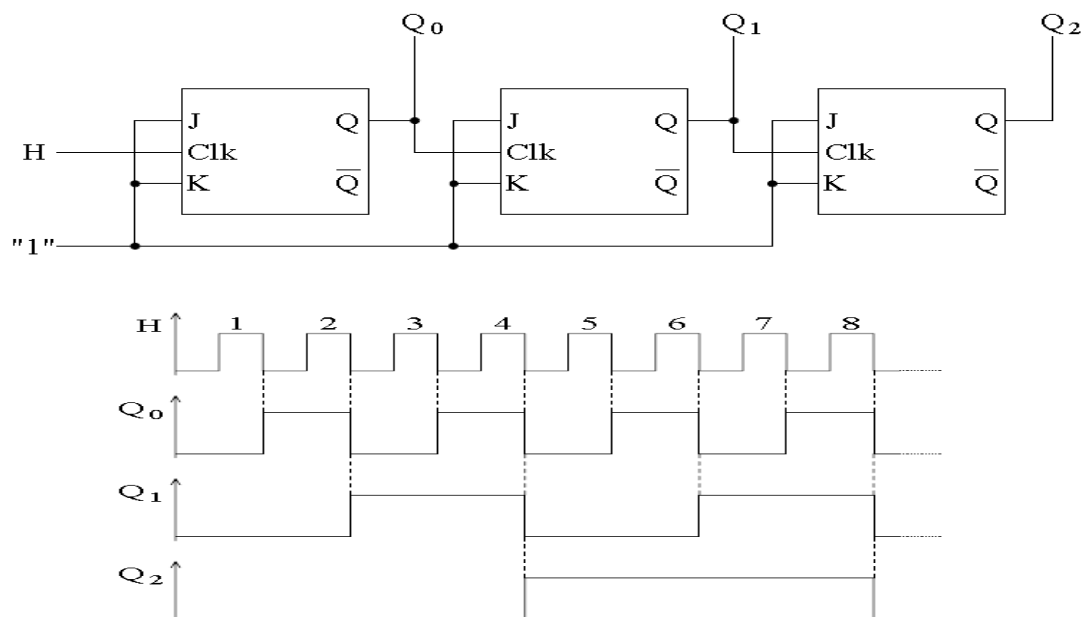
IV.2.2. Compteurs.

Un compteur est un ensemble de n bascules interconnectées par des portes logiques de sorte que le contenu s'incrémente d'une unité à chaque impulsion d'horloge. Un compteur de n bascules ne peut avoir au maximum que 2^n combinaisons. Le nombre total N des combinaisons successives est appelé le modulo du compteur. On a $N \leq 2^n$. Si $N < 2^n$ un certain nombre d'états ne sont jamais utilisés.

Une distinction essentielle des compteurs porte sur le cadencement : asynchrone (en cascade, ripple) ou synchrone. Ce dernier type cadence toutes les bascules simultanément, alors que dans la disposition en cascade, chaque étage attaque le suivant.

a) Compteurs asynchrones.

Un compteur asynchrone est constitué de n bascules J-K fonctionnant en mode T. Le signal d'horloge n'est reçu que par le premier étage (bascule LSB : Least Significant Bit). Pour chacune des autres bascules le signal d'horloge est fourni par une sortie de la bascule de rang immédiatement inférieur.

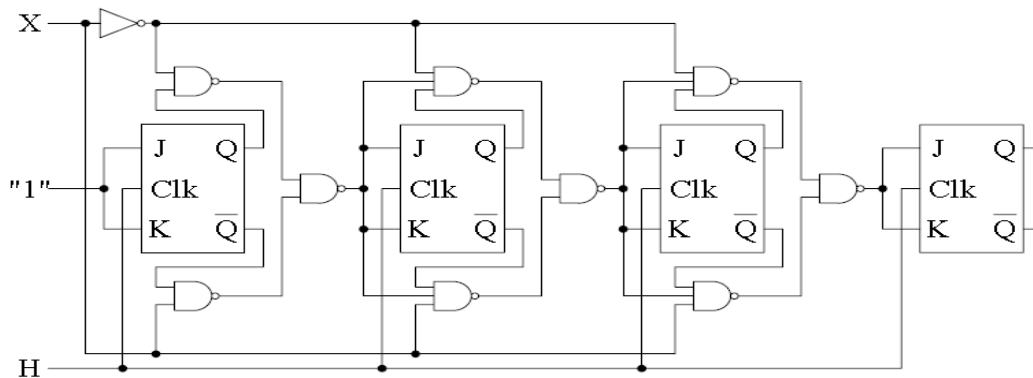


A partir de ce chronogramme nous pouvons écrire la liste des états successifs des trois sorties :

Impulsion	Q_2	Q_1	Q_0
état initial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

b) Compteurs synchrones.

Dans un compteur synchrone toutes les bascules reçoivent en parallèle le même signal d'horloge.

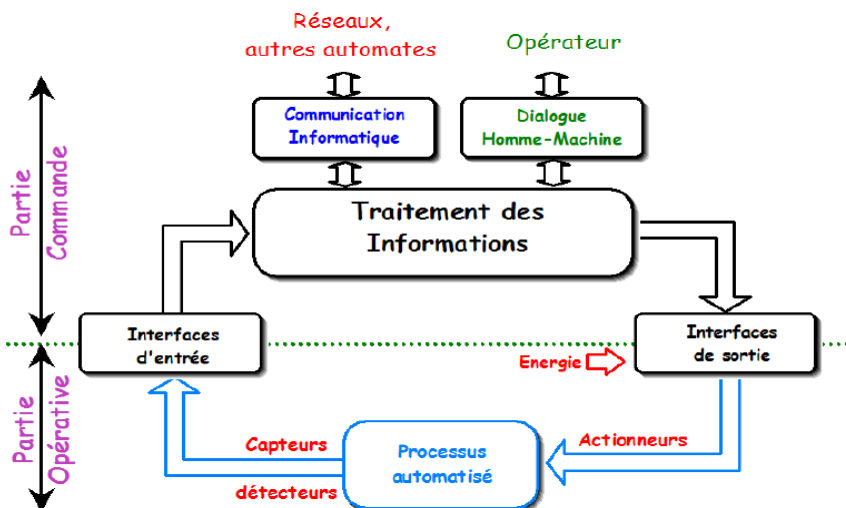
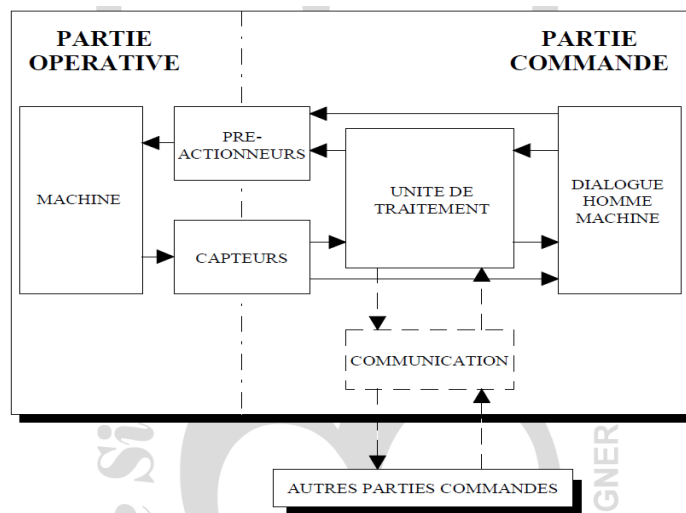


V. LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API).

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

V.1. Architecture.

Les systèmes automatisés, utilisés dans le secteur industriel, possèdent une structure de base identique. Ils sont constitués de plusieurs parties plus ou moins complexes reliées entre elles. La structure générale comprend deux grandes parties qui sont : la **partie opérative (PO)** et la **partie commande (PC)**.



V.1.1. La partie opérative.

C'est la partie visible du système. Elle comporte les éléments du procédé, c'est à dire :

- Les pré-actionneurs (distributeurs, contacteurs) qui reçoivent des ordres de la partie commande ;

- Les actionneurs (vérins, moteurs, vannes) qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique (ce sont des moteurs) ;
- Les capteurs qui informent la partie commande de l'exécution du travail. Leur rôle est donc de contrôler, mesurer, surveiller et informer la PC sur l'évolution du système.

V.1.2. La partie commande.

Ce secteur de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la Partie Opérative, et les restitue vers cette même Partie en direction des pré-actionneurs et des actionneurs.

La Partie Commande d'un système isolé est un ensemble de composants et de constituants de traitement de l'information, destiné à :

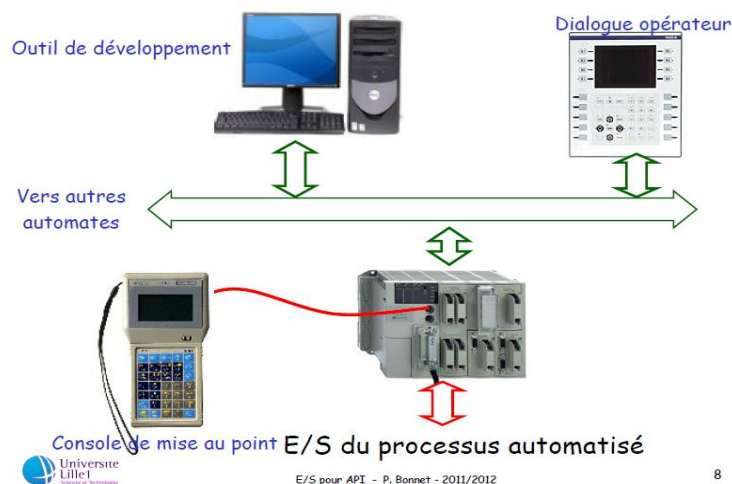
- Coordonner la succession des actions sur la Partie Opérative ;
- Surveiller son bon fonctionnement ;
- Gérer les dialogues avec les intervenants ;
- Gérer les communications avec d'autres systèmes ;
- Assurer le traitement des données et des résultats relatifs au procédé, aux matières d'œuvre, aux temps de production, à la consommation énergétique (gestion technique).

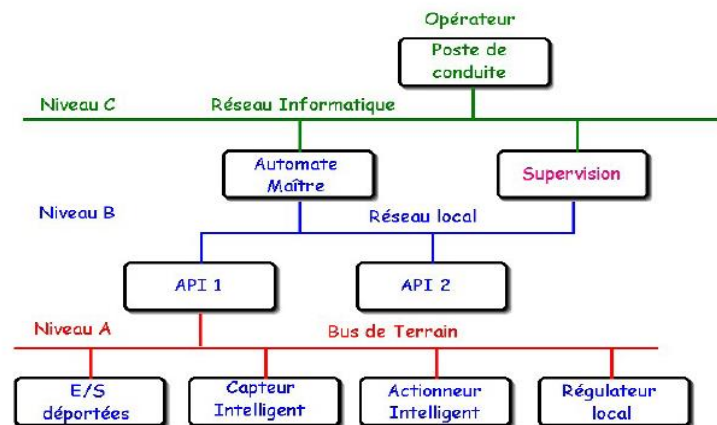
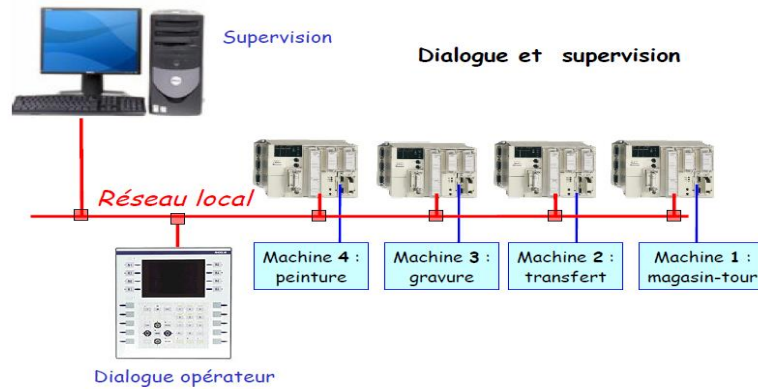
L'outil de description de la partie commande s'appelle le GRAPhe Fonctionnel de Commande Etape / Transition (GRAFCET).

Une autre partie, appelée **partie relation**, est incorporée à la partie commande. Sa complexité dépend de l'importance du système. Elle regroupe les différentes commandes nécessaires au bon fonctionnement du procédé, c'est à dire marche/arrêt, arrêt d'urgence, marche automatique, etc. L'outil de description s'appelle le Guide d'Etudes des Modes de Marches et d'Arrêts (GEMMA).

Les outils graphiques, que sont le GRAFCET et le GEMMA, sont utilisés par les automaticiens et les techniciens de maintenance.

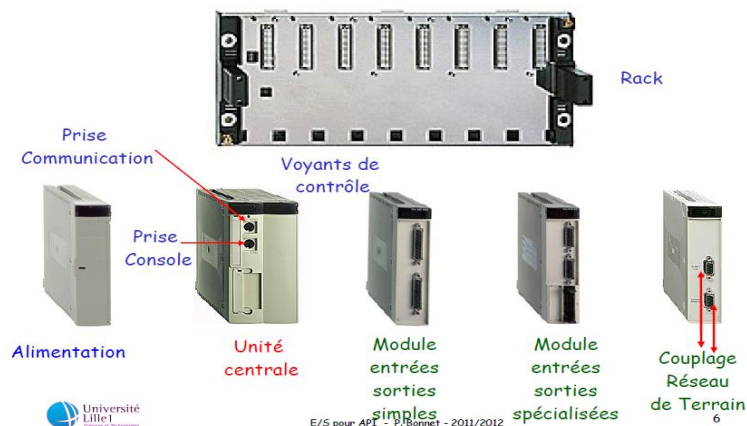
Dans certains domaines d'applications (surtout à l'industrie) il est parfois possible qu'un API communique avec d'autres API. Dans les API sont mis en série comme l'indique la figure ci-dessous.





Système automatisé à structure hiérarchisée

V.2. Description des éléments d'un API.

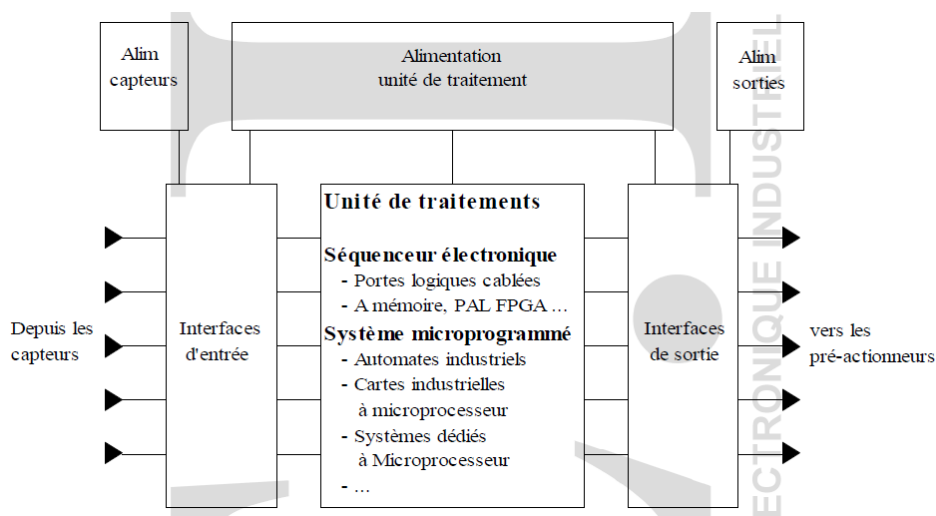


V.2.1. Unité centrale.

L'unité centrale se présente sous la forme d'un ensemble de circuit intégré réunissant tous les éléments nécessaires au traitement d'un processus automatisé. Voici généralement ce que l'on trouve à l'intérieur d'un tel composant :

- ♦ Un microprocesseur (C.P.U.),
- ♦ De la mémoire de donnée (RAM et EEPROM),
- ♦ De la mémoire programme (ROM, OTPROM, UVPRM ou EEPROM),
- ♦ Des interfaces parallèles pour la connexion des entrées / sorties,

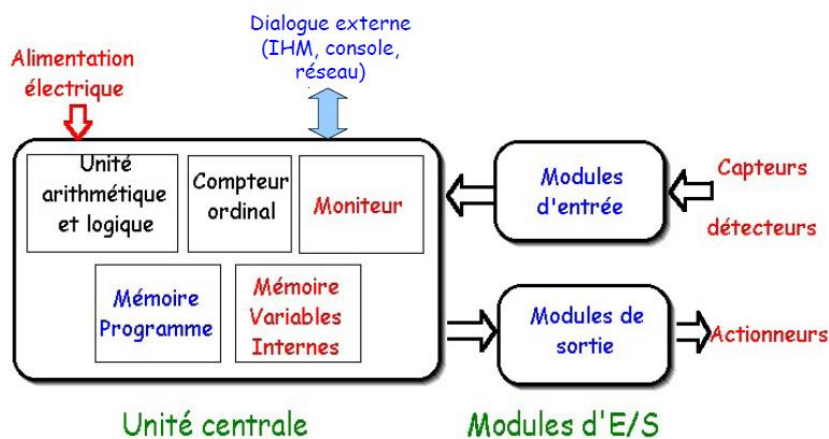
- ♦ Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités,
- ♦ Des timers pour générer ou mesurer des signaux avec une grande précision temporelle,
- ♦ Des convertisseurs analogique / numérique pour le traitement de signaux analogiques.



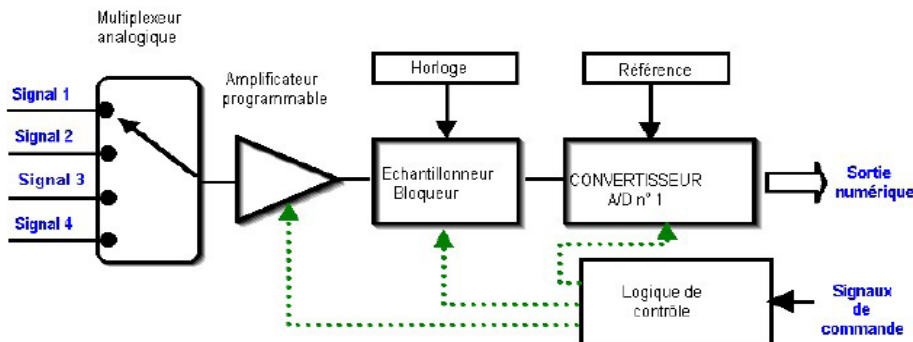
a) CPU (Central Process Unit : unité centrale) ou microprocesseur.

Un microprocesseur exécute séquentiellement les instructions stockées dans la mémoire programme. Il est capable d'opérer sur des mots binaires dont la taille, en bits, est celle du bus des données (parfois le double pour certains microcontrôleurs). Il est généralement constitué des éléments suivants :

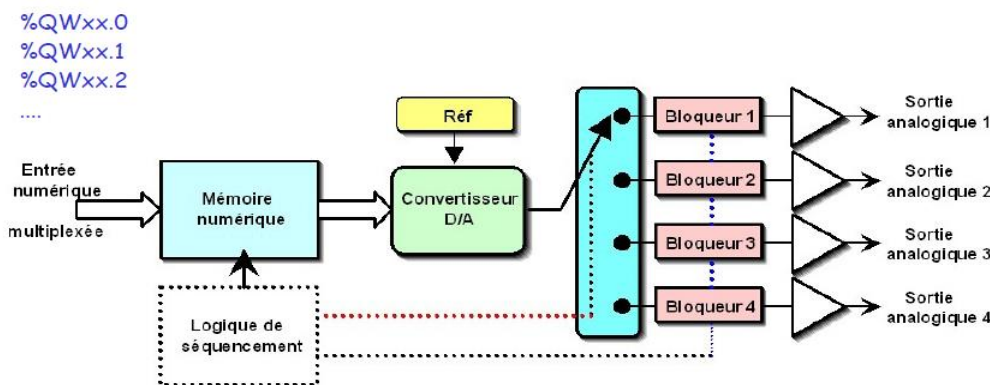
- ♦ Un ou plusieurs registres **accumulateurs** contenant temporairement les opérandes ainsi que les résultats des opérations,
- ♦ Des **registres auxiliaires** permettant de relayer les accumulateurs,
- ♦ Des **registres d'index** pour le mode d'adressage indirect,
- ♦ Un **compteur programme** pointant l'adresse de la prochaine instruction à exécuter, sa taille est celle du bus des adresses,
- ♦ Une unité arithmétique et logique (**ALU**) permettant d'effectuer des opérations entre l'accumulateur et une opérande,
- ♦ Un **registre code condition** indiquant certaines particularités en ce qui concerne le résultat de la dernière opération (retenu, zéro, interruption...).



• entrées multiplexées



• sorties démultiplexées

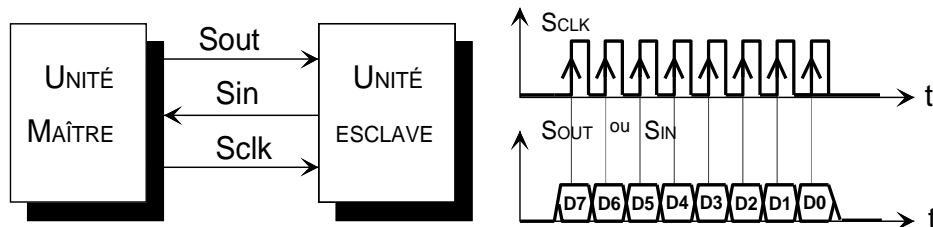


e) L'interface série.

Ce type d'interface permet à l'automate de communiquer avec d'autres systèmes à base de microprocesseur. Les données envoyées ou reçues se présentent sous la forme d'une succession temporelle (sur un seul bit) de valeurs binaires images d'un mots. Il y a 2 types de liaison série : synchrone et asynchrone.

➤ Liaison série synchrone.

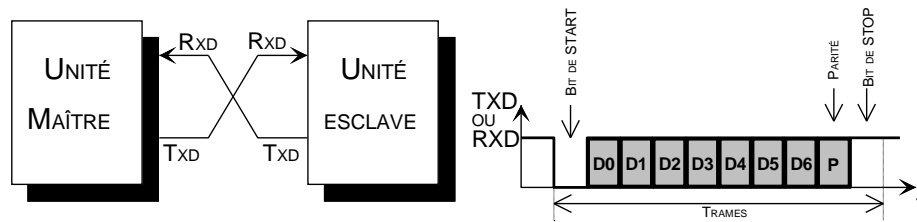
Dans ce dispositif la transmission est synchronisée par un signal d'horloge émis par l'unité maître.



➤ Liaison série asynchrone.

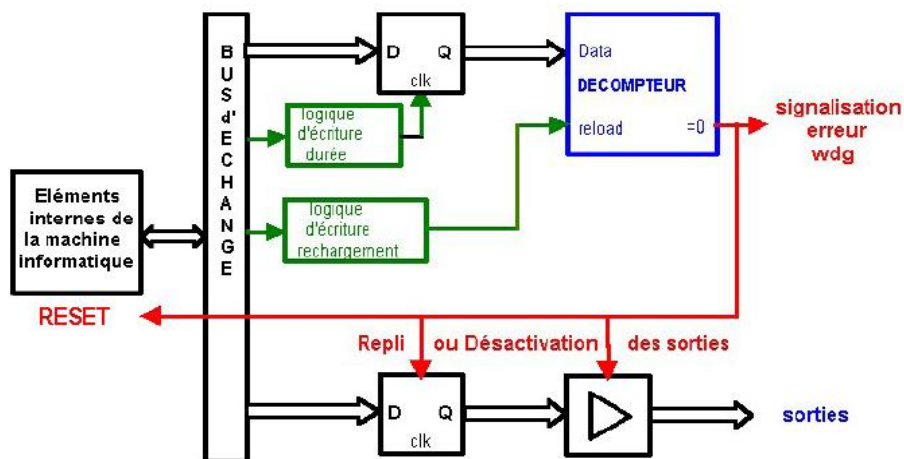
Ce dispositif ne possède pas de signal d'horloge de synchronisation. Les unités en liaison possèdent chacune une horloge interne cadencée à la même fréquence. Lorsqu'une unité veut émettre un mot binaire, elle génère un front descendant sur sa

ligne émettrice. A la fin de l'émission de ce mot, la ligne repasse au niveau haut. La donnée à transmettre peut contenir un bit supplémentaire appelé "parité" et servant à la correction d'erreurs.



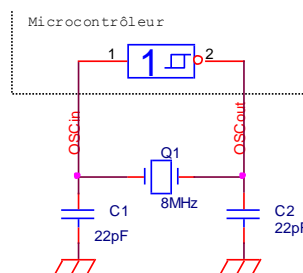
f) Le chien de garde (watch dog timer).

Ce dispositif est un système anti-plantage de l'automate. Il s'assure qu'il n'y ait pas d'exécution prolongée d'une même suite d'instruction. Un compteur préchargeable se décrémente régulièrement au rythme de la fréquence d'horloge. Si aucun préchargement n'est effectué avant qu'il n'atteigne la valeur "0" un Reset est généré relançant ainsi l'automate. Il faut donc penser à précharger régulièrement ce chien de garde par programme lorsqu'il est activé.



g) Le signal d'horloge.

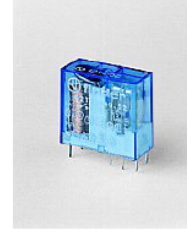
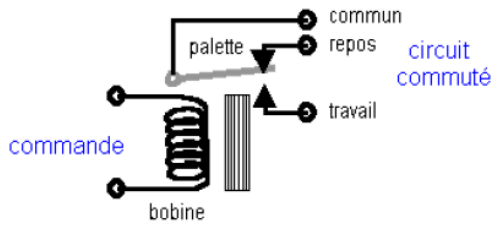
Le signal d'horloge permet de cadencer le fonctionnement de l'automate. Ce dernier intègre généralement une porte Trigger de Schmitt afin de réaliser un oscillateur. Pour l'obtenir on place un quartz entre les deux broches "Osc In" et "Osc Out" comme l'indique le schéma suivant :



V.2.2. Pré-actionneur.

Constituant de gestion d'énergie qui, sur ordre de la commande, distribue une énergie disponible vers un actionneur (ex : relais, distributeur, hacheur).

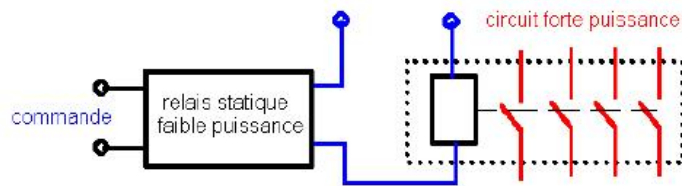
⇒ Relais électromécanique



⇒ Relais statique



⇒ Cascade de pré-actionneurs

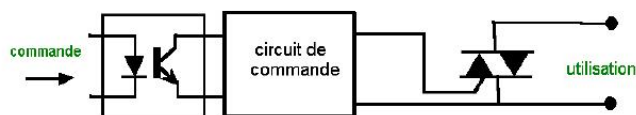


+

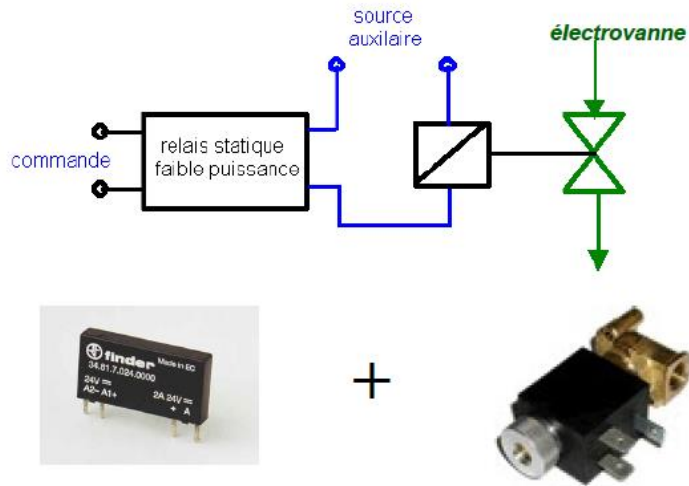


⇒ Relais statique à isolation galvanique

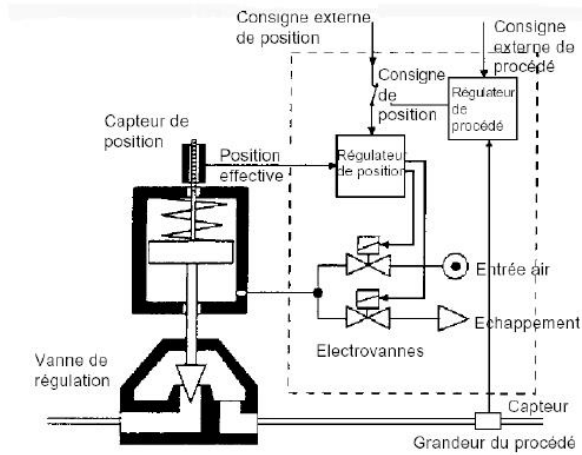
Module pour sortie à courant alternatif



⇒ Cascade de pré-actionneurs



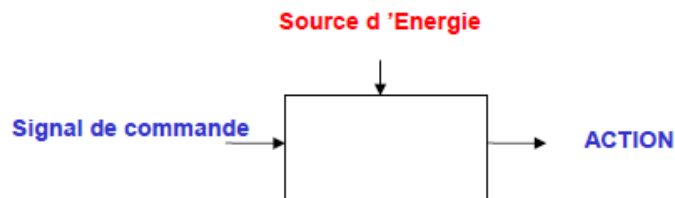
Exemple: Servovanne à commande pneumatique



⇒ Mesure réelle du débit ou de la pression

V.2.3. Actionneurs.

Dispositif qui transforme le signal de commande en une grandeur physique d'une autre nature.



Il existe deux types d'actionneurs : actionneur TOR et actionneur proportionnel.

➤ *Actionneurs TOR.*

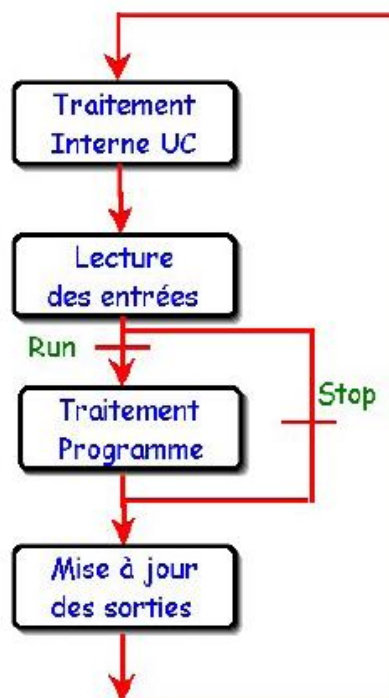
L'actionneur TOR possède 2 états : actif ou inactif. Exemple : allumage d'un voyant, déclenchement d'une sirène, mise en marche d'un chauffage, ouverture d'une électrovanne, déplacement d'un vérin, avance continue d'un tapis...

➤ **Actionneur proportionnel.**

L'action est proportionnelle au signal de commande. Exemple : ouverture d'une vanne, réglage de la puissance d'un chauffage, réglage d'une pression, contrôle de l'angle d'un rotor...

V.3. Activité logicielle d'un automate.

Un automate est en permanence en activité dès sa mise sous tension. En mode « stop », il reste actif mais seule l'exécution des tâches (logique, grafcet) est suspendue tandis que les fonctions réseau restent actives (échange avec les autres automates, surveillances des E/S ...). Le Traitement est exécuté selon les instructions définies par le programmeur.



V.4. Connexion d'un API.

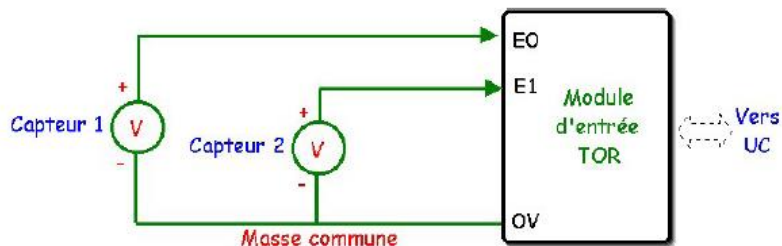
V.4.1. Raccordement des entrées logiques de l'unité de traitement.

Le principe de raccordement des entrées est conforme à un certain nombre de règles comme indiqué sur les schémas ci-dessous.

Dans la pratique, les différentes entrées, d'un point de vue de l'information, ont un point électrique commun (la masse ou mise à terre). L'alimentation des différents capteurs est fournie soit par l'automate soit par une source de tension externe (la tension d'alimentation des automates est presque toujours de 24 V continu).

Connexion des entrées :

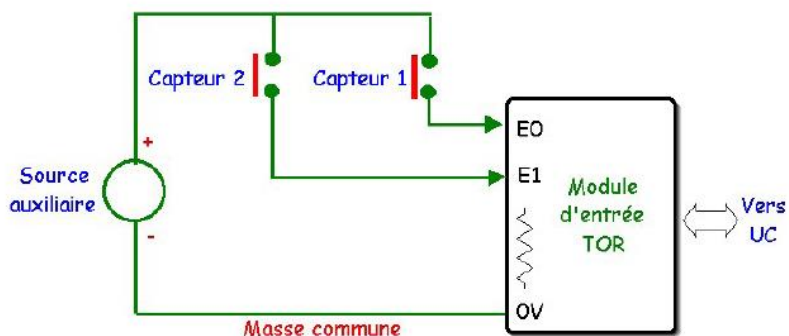
- cas des sources de tension



fonctionnement dit en « logique positive »

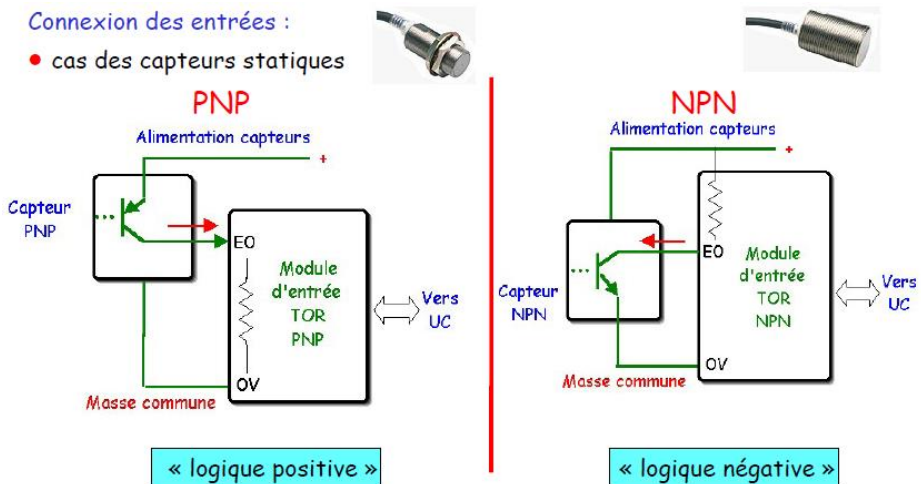
Connexion des entrées :

- cas des contacts « secs »



Connexion des entrées :

- cas des capteurs statiques



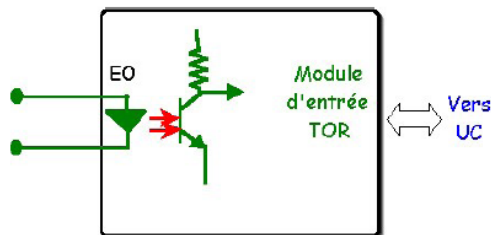
« logique positive »

« logique négative »

couplage électrique entre les entrées et l'automate

Connexion des entrées :

- entrées opto-couplées

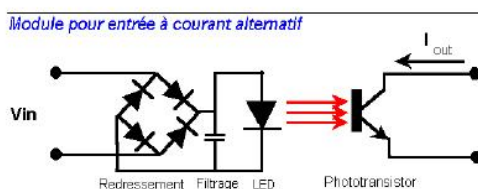
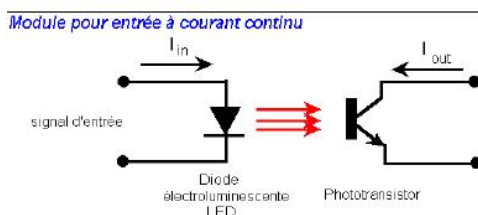


isolation galvanique

- entre l'entrée et l'automate
- entre les entrées

L'isolation implique une source d'alimentation pour chaque structure isolée

Modules isolés pour entrées TOR :



V.4.2. Raccordement des entrées spécialisées.

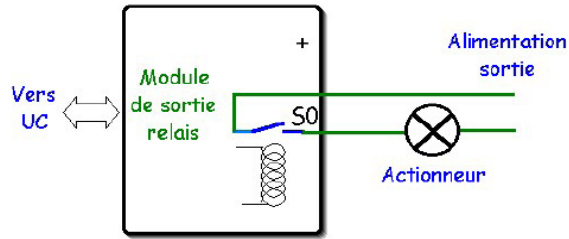
Il existe, selon les constructeurs, différents coupleurs disponibles dans une gamme d'automates programmables industriels (compteur rapide, gestion de codeur incrémental ou absolu, asservissement, contrôle d'axe de robot, gestionnaire d'imprimante...). Seule une étude, la documentation spécifique permet la mise en œuvre de ces différents modules.

V.4.3. Raccordement des sorties logiques.

Le principe de raccordement des sorties est conforme aux schémas ci-dessous. Dans la pratique, le choix d'une interface de sortie se fait suivant : le type de charge (DC/AC, tension, courant), l'isolation souhaitée, la cadence de fonctionnement souhaitée et le nombre de manœuvres souhaité.

Connexion des sorties :

- sorties « relais »

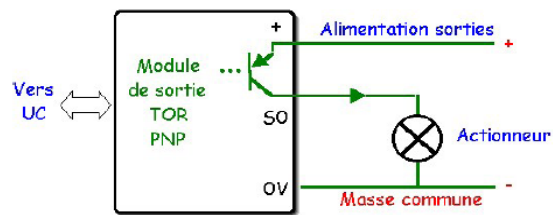


Isolation entre l'automate et les sorties

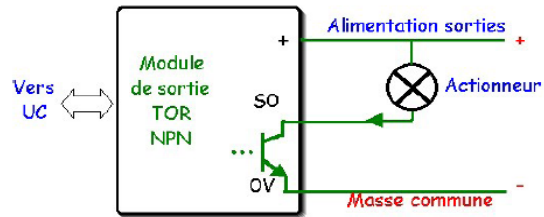
Connexion des sorties :

- sorties « statiques »

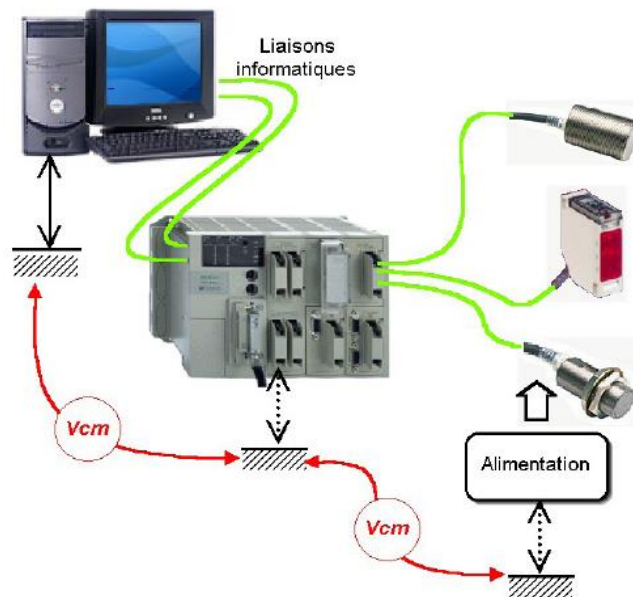
PNP

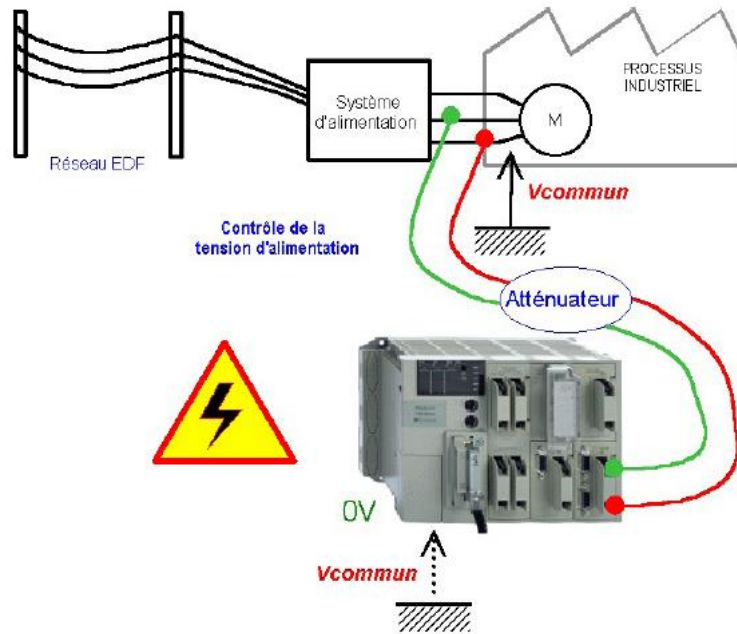


NPN

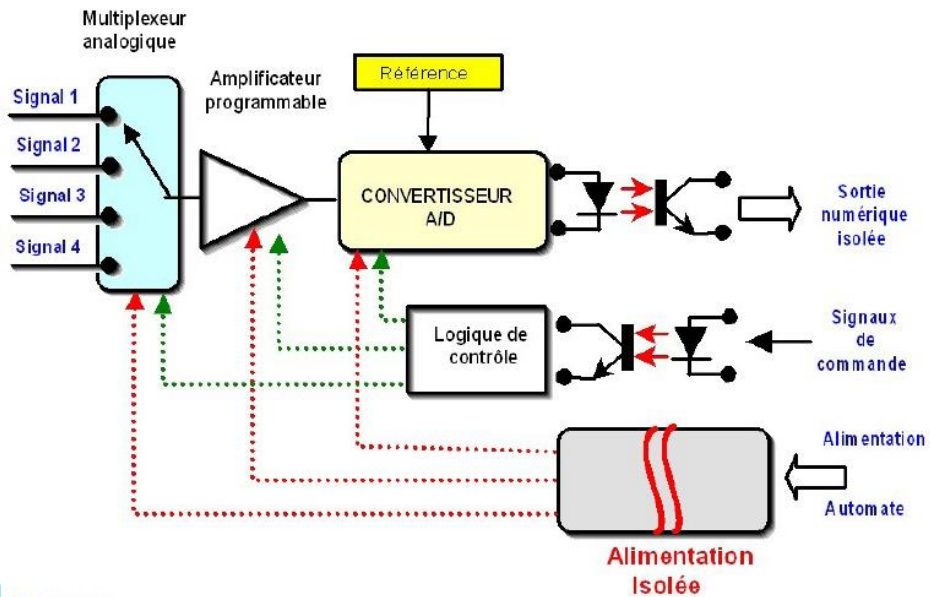


Nécessité de l'isolation :

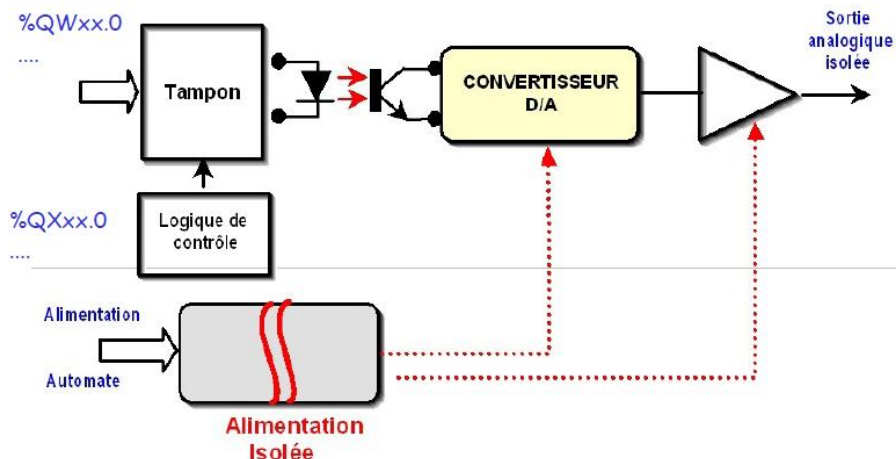




Isolation des Entrées analogiques :

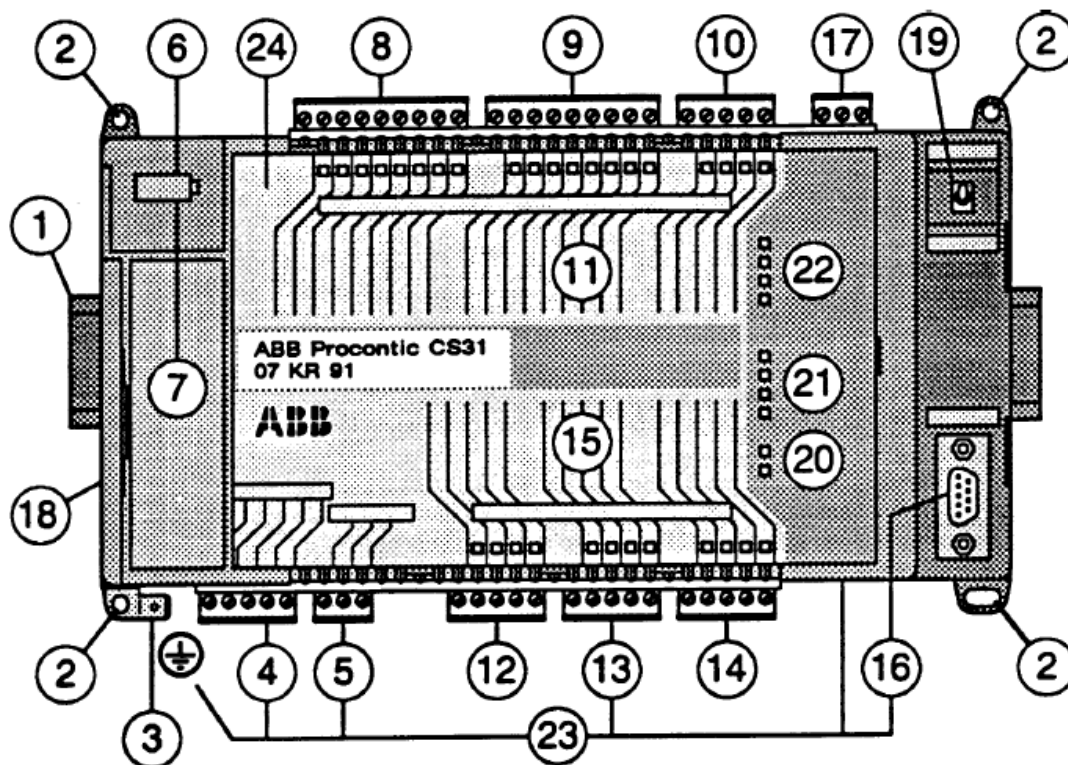


Isolation des Sorties analogiques :



Exemple d'interface de connexion pour API.

L'automate utilisé est un ABB 07 KR 91 de la gamme CS 31 dont voici le schéma.



- 1 : Montage de l'automate sur un rail DIN.
- 2 : Montage de l'automate avec des vis.
- 3 : Connexion de terre.
- 4 : Alimentation de l'automate.
- 5 : Sortie 24 V DC pour alimentation des entrées.
- 6 : Batterie de sauvegarde des données et des programmes.
- 7 : Emplacement pour batterie de sauvegarde.
- 8-10 : 20 entrées réparties en trois groupes isolés entre eux (8 + 8 + 4).

- 11 : Définition écrite des entrées.
- 12-14 : 12 sorties réparties en trois groupes isolés entre eux (4 + 4 + 4).
- 15 : Définition écrite des sorties.
- 16 : Prise série pour programmation et communication.
- 17 : Connecteur pour réseau de terrain CS 31.
- 18 : Trappe d'accès pour EEPROM.
- 19 : Interrupteur programmation/marche.
- 20 : Indicateur led pour alimentation et batterie.
- 21 : Indicateur led pour fonctionnement et erreurs.
- 22 : Indicateur led pour réseau de terrain CS 31.
- 23 : Indications pour isolation et mise à la terre.
- 24 : Film plastique sur face avant amovible pour écriture des entrées et sorties.

V.5. Communication avec un API.

V.5.1. Communication avec le programmeur.

Des outils de programmation permettant l'élaboration, la mise au point et la maintenance des programmes sont disponibles chez les différents constructeurs. Ces outils ont la forme d'une "grosse calculatrice" ou d'un logiciel dans un PC portable pour les interventions sur site (mise au point et maintenance) ou bien d'un logiciel installé sur une station de travail au bureau pour la phase d'étude et de simulation.

V.5.2. Communication avec l'agent d'exploitation.

Des outils de réglage de la taille d'une calculatrice permettent d'affiner certains paramètres pendant l'exploitation (ils sont autorisés par le programmeur en conformité avec le cahier des charges).

V.5.3. Communication locale avec des convertisseurs machines, ...

Des normes de connexion permettent des liaisons avec les différents constituants d'une chaîne de production (RS 232 C, RS 422, USB et autres liaisons propriétaires).

V.5.4. Communication distante avec d'autres automates.

Des réseaux de communication permettent un dialogue permanent entre les différents acteurs d'une chaîne de production. Ils peuvent être ouverts ou appartenir à un fabricant tels Unitelway, Modbus, Jbus et bien d'autres encore.

VI. Le GRAFCET.

En 1975, le groupe de travail Systèmes Logiques de l'AFCEC, décida, sous l'impulsion de Michel BLANCHARD qui en était l'animateur, de créer une commission Normalisation de la représentation du cahier des charges d'un automatisme logique. Les universitaires et industriels réunis dans ce groupe se sont alors attaqués à un problème ardu : tenter de définir un formalisme simple, accepté par tous, adapté à la représentation des évolutions séquentielles d'un système, intelligible à la fois par les concepteurs et les exploitants, et fournissant potentiellement des facilités de passage à une réalisation, à base matérielle et/ou logicielle de l'automatisme ainsi spécifié. Au début, le travail consista à dresser un état de l'art des différentes approches de modélisation du comportement de tels automatismes. Trois grandes classes d'outils de modélisation furent ainsi recensées :

- Les organigrammes ;
- Les représentations de systèmes logiques à évolutions simultanées (dont les Réseaux de Pétri) ;
- Les outils dérivés des graphes d'état.

L'analyse des avantages et inconvénients de ces outils mena, en 1977, à la définition du GRAFCET, ainsi nommé pour, à la fois marquer l'origine de ce nouvel outil de modélisation grAFCEC et son identité (GRAphe Fonctionnel de Commande Etapes-Transitions). Les résultats de ces travaux firent l'objet, d'une part, d'un rapport interne à l'AFCEC, daté d'août 1977, et, d'autre part d'une publication officielle dans la revue "Automatique et Informatique Industrielle" en décembre 1977, date que la communauté considère aujourd'hui comme correspondant à la date de naissance effective du GRAFCET.

Le GRAFCET permet l'établissement des descriptions de la fonction et du comportement des systèmes de commandes en établissant une représentation graphique indépendante de la réalisation technologique.

Le GRAFCET est destiné à représenter des automatismes logiques (ou discrets), c'est à dire des systèmes à événements discrets dans lesquels les informations sont de type booléennes ou peuvent s'y ramener (par exemple la comparaison d'une température avec un seuil). Le GRAFCET est utilisé généralement pour spécifier et concevoir le comportement souhaité de la partie commande d'un système, mais il peut également être utilisé pour spécifier le comportement attendu de la partie opérative ou bien de tout le système de commande.

VI.1. But du GRAFCET.

Lorsque le mot GRAFCET (en lettre capitale) est utilisé, il fait référence à l'outil de modélisation. Lorsque le mot grafcet est écrit en minuscule, il fait alors référence à un modèle obtenu à l'aide des règles du GRAFCET. Présenté sous forme d'organigramme, son but est la description du fonctionnement de l'automatisme contrôlant le procédé. C'est tout d'abord un outil graphique, puissant, directement exploitable, car c'est aussi un langage pour la plupart des API existants sur le marché. Il comprend :

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

VI.2. Description du GRAFCET.

VI.2.1. Les étapes.

L'étape symbolise un état ou une partie de l'état du système. L'étape possède deux états possibles : *active* représentée par un jeton dans l'étape ou *inactive*. L'étape *i*, repérée

numériquement, possède ainsi une variable d'état, appelée variable d'étape X_i . Cette variable est une variable booléenne valant 1 si l'étape est active, 0 sinon.



Étape initiale

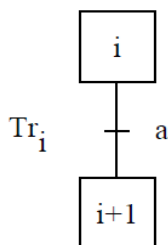
Elle représente le système à l'état de repos initial. Elle est activée au début du cycle.



Étape

A chaque étape est associée une action ou plusieurs, c'est à dire un ordre vers la partie opérative ou vers d'autres grafjets.

VI.2.2. Les transitions.



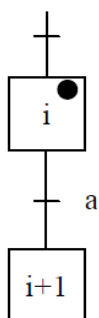
Une transition indique la possibilité d'évolution qui existe entre deux étapes et donc la succession de deux activités dans la partie opérative. Lors de son franchissement, elle va permettre l'évolution du système. A chaque transition est associée une réceptivité qui exprime la condition nécessaire pour passer d'une étape à une autre.

Cette condition est écrite sous forme d'une proposition logique, une fonction combinatoire calculée à partir :

- Des variables d'entrées traduisant l'état des capteurs, des boutons poussoirs, etc. ;
- Du temps ;
- De l'état courant des étapes du grafjet (les X_i).

Si la réceptivité n'est pas précisée, alors cela signifie qu'elle est toujours vraie.

Règle : Si l'étape i est inactive, $X_i = 0$, la transition Tr_i est sans effet. Cependant, attention, valider sans raison une transition peut avoir des conséquences graves, perturbant le cycle dans certains cas !



Si l'étape i est active, $X_i = 1$, la transition a est validée, alors :

Si $a = 0$, alors attente

Si $a = 1$, alors l'étape i est dévalidée $X_i = 0$ et l'étape suivante $i+1$ est activée, $X_{i+1} = 1$.

VI.2.3. Les liaisons orientées.

Une liaison orientée est le lien qui lie une étape à une transition ou l'inverse. Par convention, étapes et transitions sont placées suivant un axe vertical. Les liaisons orientées sont de simples traits verticaux lorsque la liaison est orientée de haut en bas, et sont munis d'une flèche vers le haut lorsque la liaison est orientée vers le haut.

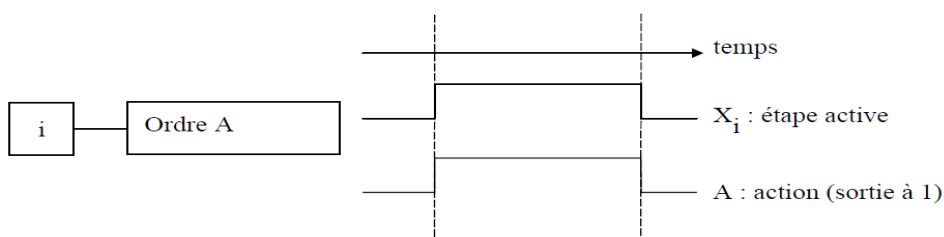
VI.2.4. Notation des entrées/sorties.

Lors de l'établissement du grafcet de spécification (grafcet de niveau 1), on utilise des noms explicites pour les entrées du système modélisé ainsi que pour les sorties. Lors du passage au grafcet de réalisation, on utilise plutôt des noms logiques : **Ei** pour les entrées et **Si** pour les sorties.

VI.2.5. Les actions.

L'action associée à l'étape peut être de 3 types : continue, conditionnelle ou mémorisée.

Actions continues

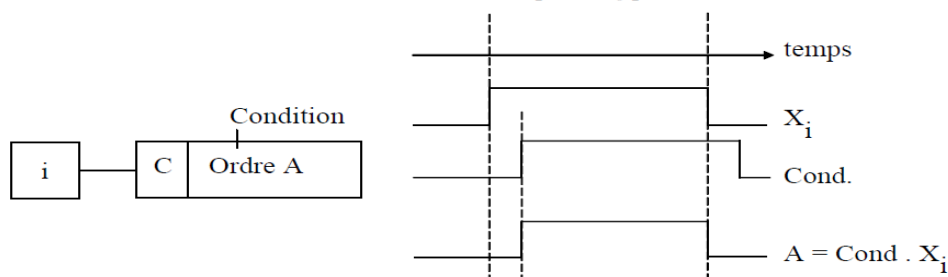


La ou les sorties correspondant à l'ordre A sont mises à 1 tant que l'étape associée est active. Lorsque l'étape devient inactive, la ou les sorties sont mises à 0.

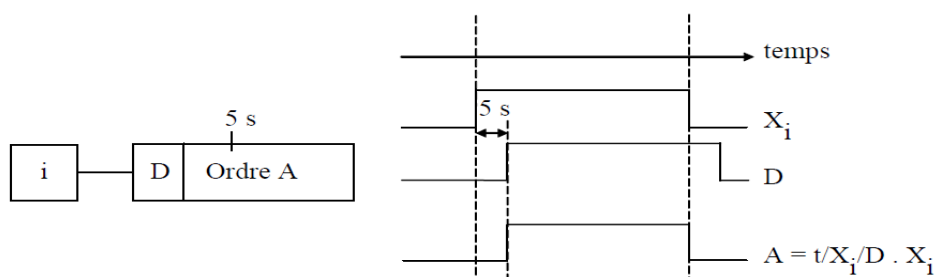
Actions conditionnelles.

Une action conditionnelle n'est exécutée que si l'étape associée est active et si la condition associée est vraie. Elles peuvent être décomposées en 3 cas particuliers.

Action conditionnelle simple : Type C

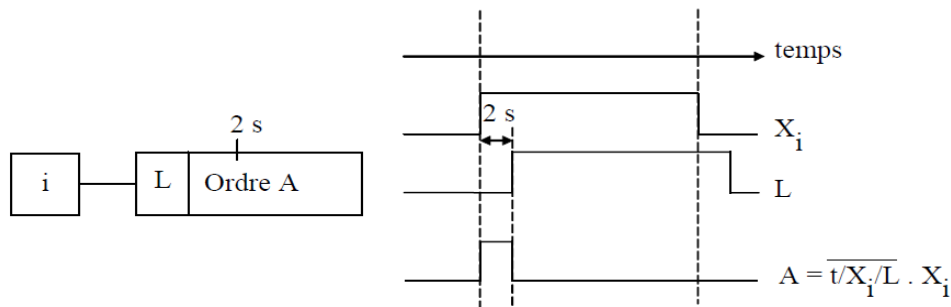


Action retardée : Type D (delay)

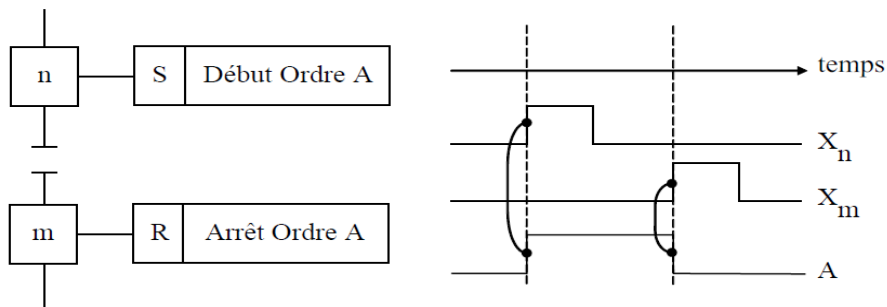


D est la durée de la temporisation, c'est à dire le retard mis pour passer à 1 après que Xi soit passé à 1.

Action limitée dans le temps : Type L (limited)



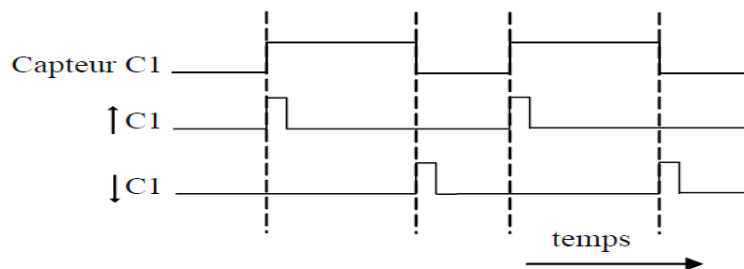
Actions mémorisées



On peut ainsi donner l'équation d'un ordre A en fonction des états des étapes, des conditions éventuelles et du temps.

VI.2.6. Les réceptivités.

Une réceptivité est associée à chaque transition. C'est une fonction booléenne calculée à partir des entrées du graphe, des états des étapes ou des temporisations. Une réceptivité est donc écrite en utilisant les opérateurs ET, OU, NON et front. Le front montant ou descendant d'une variable permet de situer dans le temps le changement de valeur de ce capteur. On les note respectivement \mathbf{E} et $\overline{\mathbf{E}}$. Les fronts ne sont à 1 que durant un délai \mathbf{d} , qui correspond au temps de prise en compte de l'événement, c'est à dire le temps de franchissement d'une transition.

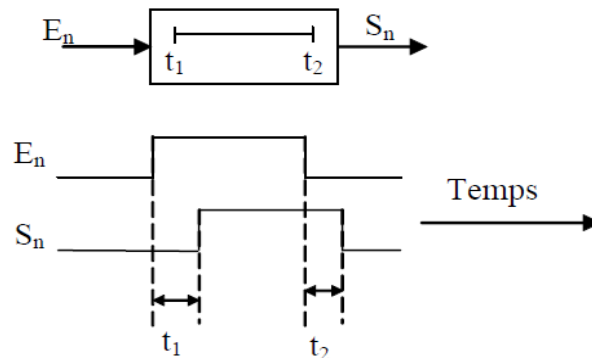


VI.2.7. Les temporisations.

Les temporisations utilisées en GRAFCET font référence aux variables d'étapes (\mathbf{Xn}). Une variable temporisation \mathbf{Sn} est une variable booléenne dont le mode d'évaluation permet de prendre en compte le temps. Elle peut s'écrire de manière générale $\mathbf{Sn} = \mathbf{t1/Xn/t2}$ où :

- \mathbf{En} désigne la variable d'entrée ;
- $\mathbf{t1}$ désigne le retard apporté au changement de l'état logique $0 \rightarrow 1$ de la variable d'entrée \mathbf{En} . Si $\mathbf{t1} = 0$, alors la temporisation s'écrit $\mathbf{Sn} = \mathbf{Xn/t2}$.

- t_2 désigne le retard apporté au changement de l'état logique $1 \rightarrow 0$ de la variable d'entrée E_n . Si $t_2 = 0$, alors la temporisation s'écrit $S_n = t_1/X_n$.



La variable X_n (l'étape n associée) doit être active pendant un temps supérieur ou égal à t_1 pour que S_n puisse être évalué.

VI.3. Les règles d'évolution du GRAFCET.

Règle N°1 : Condition initiale.

A l'instant initial, seules les étapes initiales sont actives.

Règle N°2 : Franchissement d'une transition.

Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, et seulement si la réceptivité associée est vraie.

Règle N°3 : Evolution des étapes actives.

Le franchissement d'une transition entraîne obligatoirement dans cet ordre la désactivation de toutes ces étapes amont et l'activation de ses étapes aval.

Règle N°4 : Franchissement simultané.

Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.

Règle N°5 : Conflit d'activation.

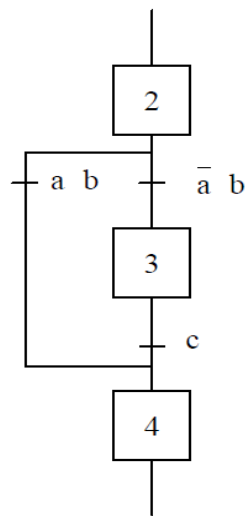
Si une étape doit être simultanément désactivée par le franchissement d'une transition aval, et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes au procédé) non désirées.

Durée de franchissement d'une transition ou d'activité d'une étape.

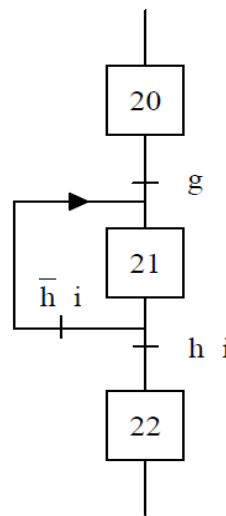
La durée de franchissement d'une transition est considérée théoriquement comme aussi petite que l'on veut, mais non nulle, même si, en pratique, cette durée peut être imposée par la technologie utilisée pour la réalisation de l'automatisme. De même, la durée d'activité d'une étape ne peut pas être nulle, mais suffisante, si besoin est, pour effectuer une action fugitive à la vitesse de la partie commande.

VI.4. Les structures de base.

VI.4.1. Saut d'étape et reprise de séquence.

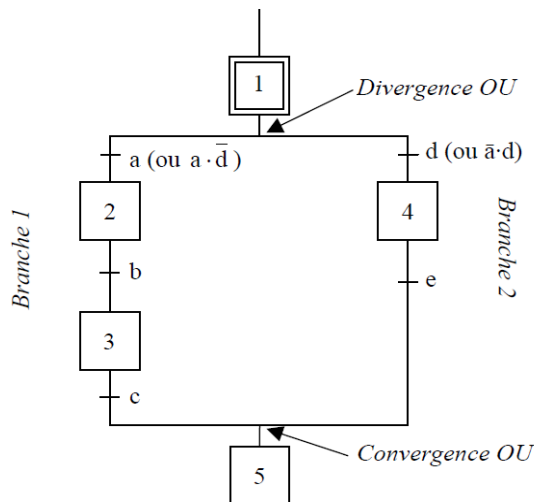


Saut d'étape



Reprise de séquence

VI.4.2. Aiguillage entre deux ou plusieurs séquences.



Soit X_1 active et $a = 0$ $d = 0$
 → attente que a ou d passe à 1

Branche 1 :

- si $a = 1 \rightarrow X_2$
- si $b = 1 \rightarrow X_3$
- si $c = 1 \rightarrow X_5$

Branche 2 :

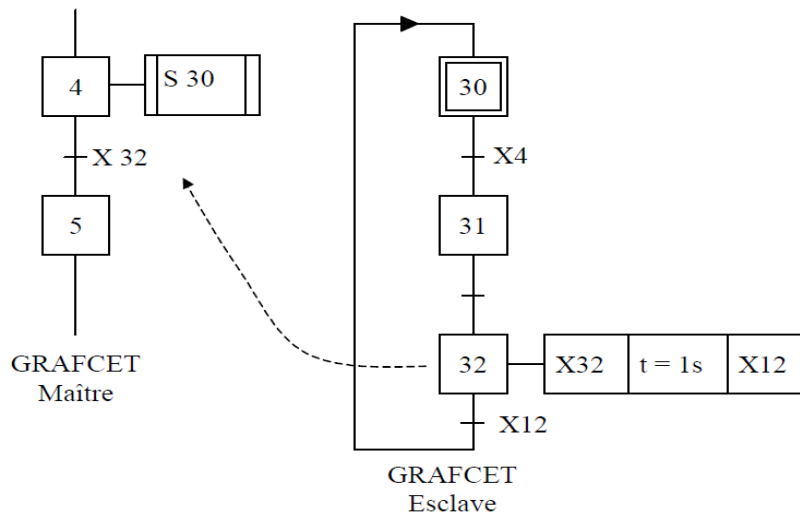
- si $d = 1 \rightarrow X_4$
- si $e = 1 \rightarrow X_5$

Remarque : chaque branche de la divergence peut être terminée par un saut, auquel cas il n'y a pas de convergence en OU.

Attention : Si a et d sont à 1 simultanément, alors nous sommes confronté à un cas de parallélisme interprété. Les étapes 2 et 4 vont devenir actives simultanément et donc provoquer l'exécution d'actions en parallèle, situation non voulue par le concepteur. a et d doivent être des conditions exclusives : l'une ou l'autre sera vraie, mais pas les deux à la fois.

VI.4.3. Parallélisme.

Au contraire de l'aiguillage où ne peut se dérouler qu'une seule activité à la fois, il peut y avoir plusieurs activités indépendantes pouvant se dérouler en parallèle. On parle alors de parallélisme structurel.



Les grafjets esclaves sont très utiles pour les séquences répétées, arrêt d'urgence, commande manuelle. Ils permettent de commander plusieurs automatismes sur un même API.

VI.4.5. Gestion de conflits.

Suivant la structure de programmation choisie, et pour des automatismes donnés, il existe des situations conflictuelles qui peuvent avoir des conséquences graves. Comment y remédier ?

Aiguillage.

Si les conditions d'aiguillage sont toutes validées simultanément, l'automaticien est face à un problème de préséance, qui se résout en doublant l'une des conditions d'aiguillage.

Parallélisme où 3 types de conflits peuvent se présenter.

- Le déroulement simultané de séquences avec attente commune. Ceci se produit, si les durées des actions des séquences simultanées sont différentes. On ajoute alors une étape de synchronisation sans action. C'est ce que l'on appelle la synchronisation de durée.
- L'état d'avancement de plusieurs séquences simultanées. Une séquence a son déroulement tributaire de l'état d'avancement d'une autre séquence. L'utilisateur a alors besoin d'une synchronisation de décalage temporel qui peut se faire au niveau d'une transition ou d'un grafjet esclave activé en fonction des contraintes de synchronisation.
- La mémorisation de disponibilité d'une partie de la machine. Une séquence particulière n'est utilisée qu'alternativement par plusieurs séquences. Il faut alors faire un choix de priorité ou utiliser des étapes d'attente si la séquence commune est déjà utilisée.