Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.
Université Ziane Achour de Djelfa,
Faculté des Sciences et de la Technologie
Département Génie Mécanique

Support de cours

Automatismes

Réalisé Par:
Dr. MERZOUK Imad

Année universitaire 2019 /2020

TABLE DES MATIÈRES

CHAPITRE 4 : LES LANGUAGES DE PROGRAMATION

CHAPITRE 5 : INTRODUCTION AU RESEAU INDUSTRIEL

Introduction

le remplacement de l'être humaine par des machines dans les taches du système de production deviens une nécessité a l'industrie moderne pour garantie la bonne qualité des produits au meilleure prix, donc en parle du systèmes de production automatisé ou bien l'automatismes. A l'heurs actuelle L'Automate Programmable Industriel (API) présente le cœur de chaque système automatisé, cet appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande des machine à partir d'informations logique, analogique ou numérique.

Dans ce cours en s'intéresse a l'étude des systèmes automatisé. commençons par leur architecture détaillé et les différents parties qui rentre dans sont construction, a savoir la partie opérative et la partie commande. Puis la modélisation du système automatisé utilisons le GRAFCET sera présentée, cet outil simple et efficace est le plus utilisé dans le monde industriels. Par la suite les langages de programmation des API sera présenté en donne un exemple de programmation sou logiciel Step 7 qui permet l'accès "de base" aux automates Siemens. Il permet de programmer individuellement un automate (en différents langages). Il prend également en compte le réseau, ce qui permet d'accéder à tout automate du réseau (pour le programmer), et éventuellement aux automates de s'envoyer des messages entre eux. Dans la fin de ce cours les réseaux locaux industriels sont brièvement exposés.

CHAPITRE-1 Généralité Sur Les Systems Automatisée

1-1Définition

Un système est dit automatisé lorsqu'il exécute le même **cyclede travail**.

1-2Architecture

l'architecture du système automatisée est illustré dans la figure 1.1.

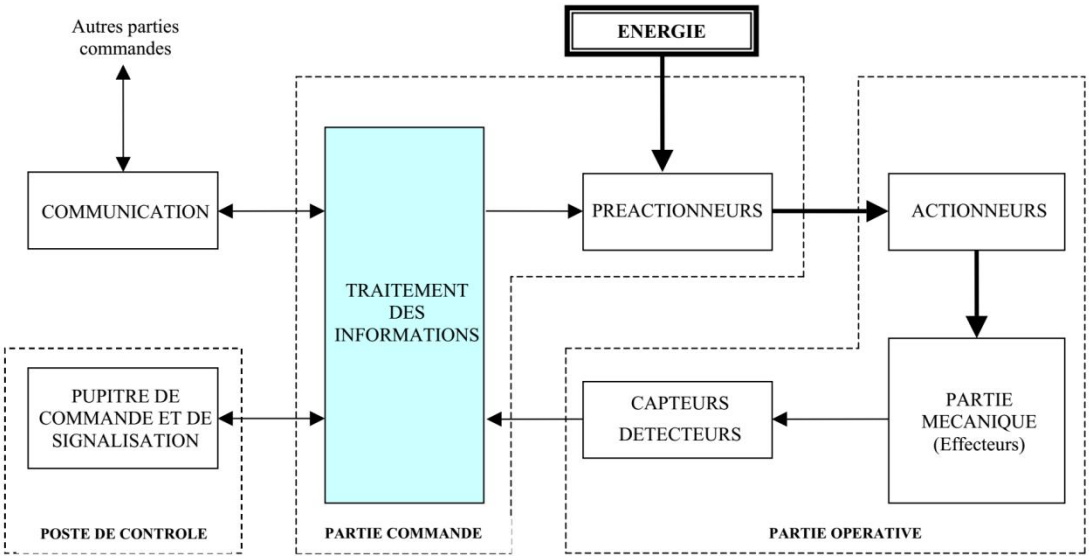


figure1.1 architecture du system automatisée

Tout système automatisé est constitué de 2 parties principales:

La partie opérative ou PO qui assure les modifications de matière d'œuvre et produit ainsi la valeur ajoutée ;la PO est représentative du processus physique à automatiser.

La partie commande ou PC qui gère de façon coordonnée les actionneurs de la PO afin d'obtenir les effetssouhaités à partir d'un modèle de fonctionnement et de diverses consignes

LA CHAINE D'ACTION est constituée des PREACTIONNEURS qui distribuent la PUISSANCE aux ACTIONNEURS. (Si la puissance de l'actionneur est faible on peut se passer de préactionneur ex : ampoule…).

L'EFFECTEUR est l'élément terminal de la chaîne d'action. Il agit directement sur la MO et concrétise lavaleur ajoutée. Il est en général lié à l'actionneur par une chaîne cinématique. (Ex d'effecteurs : Pince, Outil,…).

L'ACTIONNEUR Convertir une énergie d'entrée (énergie de puissance transmise par le préactionneur) en uneénergie de sortie adaptée à l'exécution de la tâcheopérative par l'effecteur. (Ex d'actionneurs : Moteur,Verin,…).

LA CHAINE D'ACQUISITION est constituée des CAPTEURS qui envoient des comptes rendus surl'état de la p.o. a l'unité de traitement.

L'UNITE DE TRAITEMENT est l'organe principal de la PC partie du système automatisé et gère le processus ordonné des tâches de la partie opérative. C'est elle qui renferme le PROGRAMME qui traite les informations reçues de la PO par les capteurs ou du PUPITRE par l'opérateur et qui envoie les ORDRES à laPO (par l'intermédiaire des préactionneurs). (Ex d'unités de traitement : API, Ordinateur, Carte dédiée,…).

LE PUPITRE est l'organe servant d'Interface Homme Machine (HMI). L'opérateur envoie des CONSIGNESà l'unité de traitement et reçoit en retour des informations (Ex d'éléments de pupitre : Bouton, Voyant,Clavier, Ecran,…).

1-3Constituants des chainesfonctionnelle

	Fonction technique	Composant
Chine d'énergie	**Alimentation**	énergie électrique du réseau , prise réseau, raccord réseau pneumatique, piles, accumulateurs
	Distribution	contacteur, relais d'alimentation d'un moteur, de variateur ou encore distributeurs pneumatique ou hydrauliques
	Conversion	Moteur électrique, moteurs thermique, vérins hydraulique, pneumatique,
	Transmission	Transformer la nature du mouvement par exemple les mécanismes poulies-courroies, vis-écrou ou de transformation de mouvement plus généralement (bielle manivelle, etc,…) Adapter l'énergie sans en changer sa nature : il s'agit par exemples des engrenages, des embrayages, des mécanismes poulies courroies, des variateurs de vitesse, etc, …
	Agir	
Chaine d'information	**Acquérir**	capteurs analogiques, numérique mais aussi des interfaces homme/machine et de systèmes numériques d'acquisition de données.
	Traiter	Il peut s'agir d'ordinateurs, d'automates programmables, de microcontrôleurs, de circuits de logiques câblés, voire d'ateliers logiciels (éditeur de modèle de commande , …)
	Communiquer	liaisons informatiques simples entre les deux chaînes (liaison série de l'ordinateur, liaison parallèle, réseau Ethernet), mais aussi de bus plus complexes

1-3-1CHINE D'INFORMATION

a) LES CAPTEURS

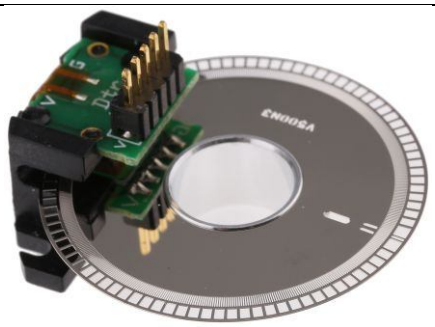
Le capteur fournit à la PC, des comptes rendus sur l'état du système. Il convertit les informations physiques de la PO en grandeurs électriques exploitables par la PC.

Types des capteurs :

TOR : Ce sont les capteurs les plus répandus en automatisation courante : Capteur à contacts mécaniques, détecteurs de proximité, détecteur à distance ..., Ilsdélivrent un signal 0 ou 1 dit tout ou rien. On parle de détecteurs

Analogique Analogique : Les capteurs analogiques traduisent des valeurs de positions, de pressions, de températures ... sous forme d'un signal (tension ou courant) évoluant continûment entre deux valeurs limites

Numérique Numérique : transmettent des valeurs numériques précisant des positions,des pressions,..., pouvant être lus sur 8, 16,32 bits : soit en parallèle sur plusieurs conducteur soit en série sur un seul conducteur .On parle de codeurs.

 capteur de pression	 Détecteur de proximité	 Encodeur

B)LA PARTIE PUPITRE

Le pupitre permet à l'opérateur de dialoguer et de commander la partie opérative. Il comporte :

Des capteurs de commande (marche, arrêt, arrêt d'urgence…).
Des voyants de signalisation (mise sous tension, fonctionnement anormal, buzzer…).
Des appareils de mesure de pression (manomètre), de tension (voltmètre), d'intensité (ampèremètre).

voyant	bouton poussoir

1-3-2 CHINED'ENERGIE

a) LESACTIONNEURS convertissent l'énergie d'entrée disponible sous une certaineforme (électrique, pneumatique, hydraulique) en une énergie utilisable sous uneautre forme, par exemple

- **Energie thermique** destinée à chauffer un four (l'actionneur étant alors une résistance électrique).

- **Energie mécanique** destinée à provoquer une translation de chariot(l'actionneur pouvant être un vérin hydraulique ou pneumatique).

- **Energie électrique** destinée à provoquer une rotation de broche(l'actionneur pouvant être alors un moteur électrique).

 **Moteur asynchrone**	 **Moteur a courant continue**	 **Servomoteur**
 **Vérin pneumatique**	 **Vérin hydraulique**	 **Vérinélectrique**

L'énergie pneumatique

L'énergie pneumatique et fournie par un compresseur d'aire. L'énergie est stockée dans un accumulateur dont la pression est régulée à une dizaine de bars. L'actionneur est un vérin

Les avantages de l'énergie pneumatique sont
-énergie non polluante
-Vitesse importante
- simplicité de mise en œuvre et de maintenance
Les désavantages

- Efforts développés faibles

- Maintien en position peu précis

Il est utilisé pour l'obtention du mouvement simple et rapide. Application type : transfert de pièce, petit outillage, milieu explosif ….

L'énergie hydraulique

L'énergie hydraulique est fournie par une centrale hydraulique qui utilise une énergie primaire (électrique ou thermique). L'actionneur est un vérin

Les avantages

-puissance massique très importante 5kW/kg
-force et couple très élevés

Les désavantages

-Nécessite d'une centrale de production d'énergie hydraulique
-Fluide polluant
-Vitesse limitée

Domaine d'utilisation : travaux publique, aéronautique, engins sous-marins, matériel agricole…..

L'énergie électrique

L'énergie électrique est très souple il peut être

- Redressée
- Mise à niveau (transfo)
- Stocké (accumulateur)

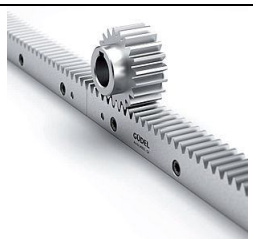
Type d'actionneur électrique : moteur à courant continue, moteur asynchrone, moteur synchrone……..

b)LES PRE-ACTIONNEURS reçoivent les signaux de commande et réalisent la commutation de puissance avec les actionneurs. Les préactionneurs des moteurs électriques sont appelés contacteurs. **Les préactionneurs** des vérins et des moteurs hydrauliques et pneumatiques sont appelés distributeurs (à commande électrique ou pneumatique)..Pour l'énergie pneumatique et hydraulique les pré-actionneurs sont les distributeurs. Pour l'énergie électrique en utilise les contacteurs ou les relais

 **contacteur**	 **distributeur pneumatique**

c)Transmetteur

Son rôle est d'**adapter** et de **transmettre l'énergie mécanique** délivrée par l'actionneurpour la rendre utilisable par l'effecteur.

 **cardan**	 **Crémaillère**
 **Rous d'entrées**	 **poulie courroie**

d)la matière d'œuvre

La matière d'œuvre peut se présenter sous plusieurs formes à savoir :
Produit : liquide, solide, gazeux,

- ✓ Énergie : électrique, thermique, mécaniques, etc.
- ✓ Information : physique, audiovisuel, etc.

La valeur ajoutée est caractérisée par sa nature, sa quantité et sa qualité. Elle peut être soit:

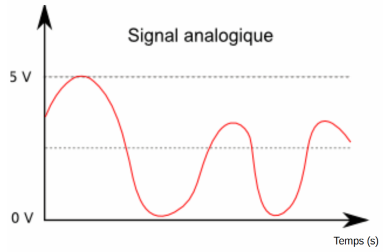
- ✓ Une modification physique: conversion d'énergie, mécanique, etc.
- ✓ Un arrangement particulier: montage, assemblage, etc.
- ✓ Un prélèvement d'information : mesure, contrôle, etc.

1-4 classification des system automatisé

1-4-1 Les différentes natures d'information

On a vu qu'à chaque chaîne fonctionnelle d'un systèmecorrespond une chaîne d'information et une chaîne d'énergie. L'automatique s'intéresse à la **chaîne d'information**. Les systèmes automatisés vont alors être classés en fonction de la nature desinformations de commande ou de mesure, et également en fonction de la nature du traitement de cesinformations.

On distingue deux types d'informations : **analogiques** et **discrètes (logiques)**. **a) information (signal) analogique** est une information qui peut prendretoutes les valeurs possibles dans un intervalle donné. Les grandeurs physiques, comme latempérature, la vitesse, la pression, la tension (…) sont des informations analogiques. Uneinformation analogique peut être représentée par une courbe.



b) information (signal) discrète est constituée d'un ensemble fini de valeurs. On distingue :# Information binaire (0 ou 1, vrai/faux, noir/blanc, Tout Ou Rien (TOR)).

Information numérique sous la forme d'un mot binaire, constitué de plusieurs variablesbinaires (bits1). Information généralement issue d'un traitement d'une information analogique(échantillonnage, codage2)

1-4-2 Classification des systèmes

On peut alors distinguer les systèmes automatisés suivants :

a) Systèmes automatisés à logique combinatoire

Pour un tel système les sorties dépendent exclusivement d'une combinaison des entrées, sansprendre en compte "l'histoire" du système. A un état d'entrées, correspond un et un seul état ensortie. Aucune mémoire des états précédents desentrées et des sorties n'est conservée.

L'information logique est traitée de manière instantanée. Les grandeurs y sont manipulées sous formes d'états binaires, ce qui justifie l'utilisation del'algèbre de BOOLE, et des notions liées au codage de l'information.

b) Systèmes automatisés à logique séquentielle

Qualifié de système à mémoire généralisée, les sorties du système sont élaborées à partir d'unensemble de signaux logiques, mais dépendent aussi de la chronologie des événements logiques."L'histoire" du système est prise en compte.

En effet les états précédents des entrées et des sorties sont mémorisés, et influent sur l'évolutiondu système. A une combinaison d'entrées, peuvent correspondre plusieurs combinaisons des sorties.

c) Variables analogiques ou numériques - Systèmes automatisés continus
Les signaux traités sont analogiques ou numériques, et leurs valeurs ne peuvent êtreprédéterminées. Les sorties (asservies ou non) sont des grandeurs continues pour un processusdonné.

Systèmes automatisés asservis : ils sont l'objet du cours qui suit, pour de tels systèmesune mesure de la sortie est réalisée en permanence et sa valeur comparée à l'entrée (sortiesouhaitée) puis corrigée. Ces systèmes permettent d'obtenir toutes les caractéristiques nécessairesaujourd'hui dans beaucoup de systèmes pluri-techniques [**Rapidité, Précision, Stabilité**].

Lesasservissements sont classés en deux familles : les systèmes **régulateurs** et les systèmes asservis**suiveurs**.

- ✓ Systèmes **régulateurs** : la consigne d'entrée est fixe, ils sont destinés à assurer une sortieconstante.***Exemple : Régulation de température***
- ✓ Systèmes asservis **suiveurs** ou en poursuite d'une loi de référence : la consigne d'entrée varieconstamment et l'objectif est d'ajuster en permanence la sortie au signal d'entrée.

CHAPITRE-2 GRAFCET

2-1 Commande logique de procédés

2-1-1 Opérateurs logiques

Les opérateurs logiques de base sont aussi dits opérateurs élémentaires. Pour cette partie, on notera a ou b les variables d'entrée et S la variable de sortie**.**

a) Les opérateurs de base de l'algèbre binaire

- l'opérateur OU représenter par « + »: opérateur logique de somme.
- l'opérateur ET représenter par « . » : opérateur logique de produit.
- l'opérateur NON représenter par une barre « $\bar{}$ » : opérateur logique de négation uo de complimentation.

OU	**ET**	**NON**
0+0=0 **0+1=1** **1+0=1** **1+1=1**	**0.0=0** **0.1=0** **1.0=0** **1.1=1**	$\bar{1}$**=0** $\bar{0}$**=1**

Définition

- La **somme** logique de a et b, noté **a+b**, realise le OU entre deux variables logique (a+b se lit « a ou b »). On parle aussi d'union logique.
- Le **produit** logique de a et b, noté **a.b** (qu'on note aussi **ab**), Réalise le et entre deux variables logiques (a.b se lit »a et b »). On parle aussi d'intersection logique.
- Le **cpmplémentaire** de a, noté $\bar{a}$, réalise le NON de cette variable logique ($\bar{a}$ se lit « a barre » ou « non a »)

b) Les propriétés de base de l'algèbre binaire

Règles	Fontion OU	Fonction ET
commutativité	a+b=b+a	a.b=b.a
associativité	a+(b+c)=(a+b)+c	a.(b.c)=(a.b).c
distributivité	Ou par rapport à et : a+(b.c)=(a+b).(a+c)	Et par rapport à ou a.(b+c)=(a.b)+(a.c)
Elément neutre	a+0=a	a.1=a
complémentaire	a+$\bar{a}$=1	a.$\bar{a}$=0
Forcage (absorbant)	a+1=1	a.0=0

2-1-2 La fonctionlogiquecombinatoire

a) Table de vérité : une fonction logique combinatoire F(x1 , …,xn) de n variables binaires peut etre définie par la valeur 0 ou 1 pour chacun des 2^n états d'entrée possibles. Cette liste de couples comportant un état d'ntrée et l'etas de sortie correspondant et représenté sous forme d'un tableau applé table de vérité de la fonction F.

Exemple : pour un système a 3 variables d'entrée.

a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Deux variables d'entrée = 2^3 combinaisons

b)Formecanonique

On appelle **minterme** de n variables, un produit logique de ces variables. Avec n variable, on a 2^n mintermes (autant de combinaisons possibles de n éléments binaires).

On appelle **maxterme** de n variables, une somme logique de ces variables. Avec n variable, on a 2^n maxtermes (autant de combinaisons possibles de n éléments binaires).

Exemple : por 2 variables a et b, on a :

4 mintermes : $\bar{a}\bar{b}, \bar{a}b, a\bar{b}, ab$.

4 maxtermes : $\bar{a}+\bar{b}, \bar{a}+b, a+\bar{b}, a+b$

A partir de la table de vérité d'ube fonction logique, il est possible d'obtenir deux expressions algébriques (écrites sous 2 formes différentes mais équivalentes) correspondant à la fonction. Ceux sont les expressions canoniques de la fonction.

- La 1er forme canonique(somme de produits algébriques ou somme de mintermes) est obtenue en considérant les états d'entrée pour lesquels la fonction vaut1.
- La 2éme forme canonique(produit de sommes algébriques ou produit de maxtermes) est obtenue en considérant les états d'entrée pour lesquels la fonction vaut 0.

c)Simplification d'une fonction logique

Une fonction Logique peut être représentée par plusieurs expressions différentes mais totalement équivalentes.

Tableaux de KARNAUGH et simplification.

Soit la fonction f(a,b,c,d)=∑(2,3,5,13,14).

Alors f= $\overline{a}\overline{b}c\bar{d} + \overline{a}\overline{b}cd + \bar{a}b\bar{c}d + ab\bar{c}d + abc\bar{d}$

La fonction logique s'écrit comme une somme de produit logique de 4 variables. Chacun de ces produit logique s'appelle « monôme ».la fonction f

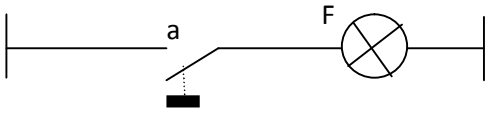
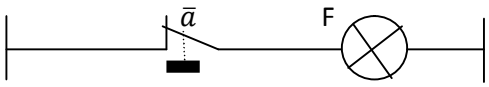
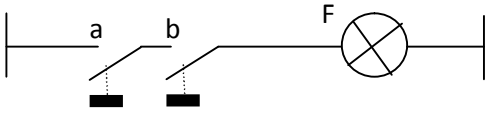
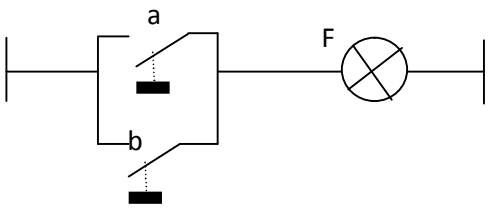
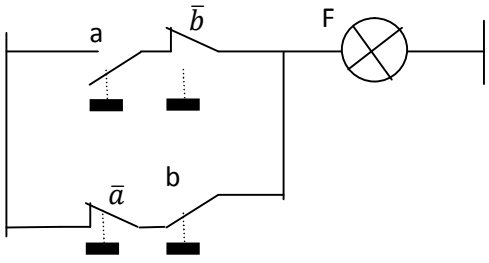
comprend 5 monômes de 4 variables qu'on représente dans la tableau de KARNAUGH suivant.

cd / ab	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	0	1	0	1
10	0	0	0	0

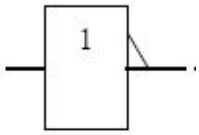
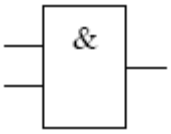
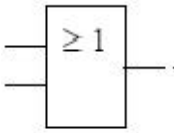
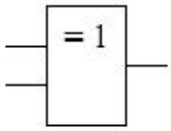
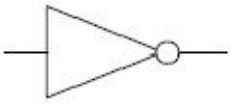
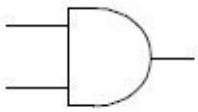
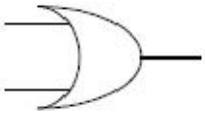
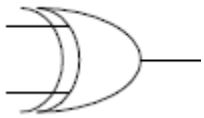
Après simplification f= $\overline{a}\overline{b}c + b\bar{c}d + abc\bar{d}$

2-1-3 Elément de réalisation technologique des portes logiques

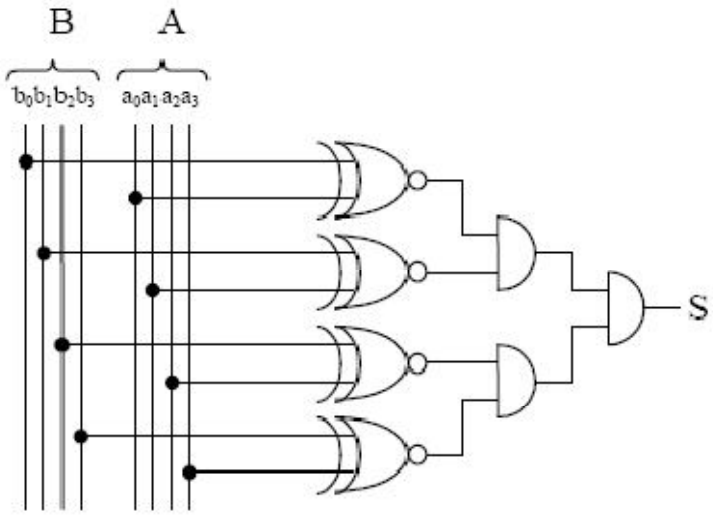
a)Fonction logique a contact

▪ **Fonction OUI (F=a)**	
▪ **Fonction NON « NOT » (F=$\overline{a}$)**	
▪ **Fonction ET « AND » (F=a.b)**	
▪ **Fonction OU (F=a+b)**	
▪ **Fonction OU exclusif (XOR) (F=a+b=$\overline{a}b + a\overline{b}$)**	

b)Les portes logique

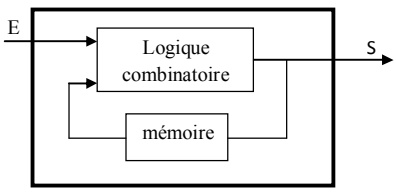
Fonction	NON	ET	OU	OU-EX
Norme européenne				
Norme américaine				

Exemple d'un logigramme.



S= ?

2-1-4 Logique Séquentielle



Dans un système séquentiel, l'états des sorties dépend des états d'entrée en plus de l'histoire (de l'état précédente) : la correspondance entre l'ensemble desentrées et l'état des sorties est variable.

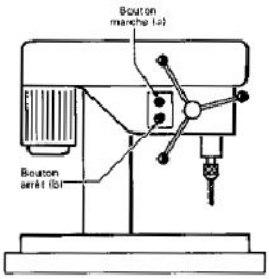
Soient

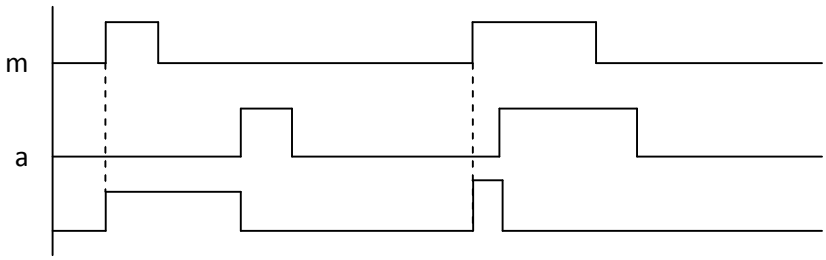
- S(t) la valeur de la sortie du système a l'instant t
- S(t+1) la valeur de la sortie du système a l'instant t+1

Alores

S(t+1)=f[e,s(t)] : la sortie à l'instant t+1 dépand de l'entrée e et de la sortie à l'instant t.

Exemple : on veut commander la mise en marche et l'arret d'uuneperseuse. On dispose de deux boutons poussoirs : « m » marche et « a » arret.





Chronogramme de fonctionnement :

Table de vérité

Action chronologique	« a »	« m »	Etat moteur	
1, on branche l'appareil	0	0	0	Le moteur ne tourne pas
2, appui sur « marche »	0	1	1	Le moteur démarre et tourne
3, relache « marche »	0	0	1	Le moteur tourne
4,Appui sur « arret »	1	0	0	Le moteur s'arrete
5,Relache »arret »	0	0	0	Le moteur est toujours arrété

Remarque : pour deux étas identiques des variable « m » et « n », on a moteur=0 ou1.

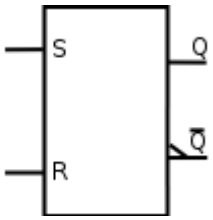
Dans la logique séquentielle, on tient compt de la combinaison des variables d'entrées et de la succession de celles-ci. Dans cet exemple, on a réalisé la fonction mémoire.

a) Bascule RS:

Un verrou RS possède deux entrées de contrôle : Set et Reset, et n'a pas d'entrée de donnée. Les deux signaux de sortie Q et non Q sont présents. Le fonctionnement de cette bascule est le suivant :

- mise à 1 de S (Set) : la sortie Q passe à 1 ;
- mise à 1 de R (Reset) : la sortie Q passe à 0 ;
- R = S = 0 : état mémoire : la sortie Q maintient sa valeur précédente q.

R	S	Qt+1
0	0	Qt
0	1	1
1	0	0
1	1	-

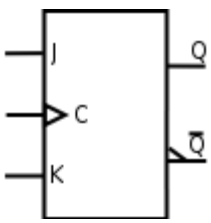


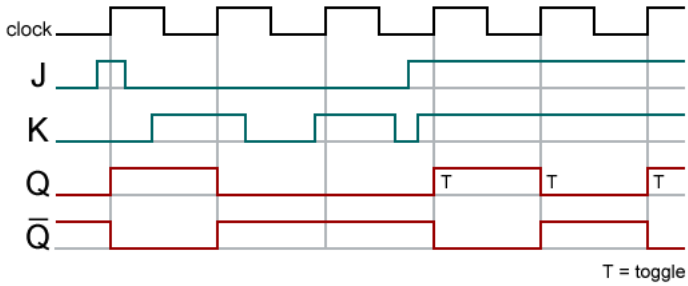
b) Bascule JK:

On réalise une bascule JK (maître-esclave). Cette bascule comporte deux entrées de contrôle : J (**J**ack) et K (**K**ing). S'agissant d'une bascule, le fonctionnement est synchrone à une entrée d'horloge H, c'est-à-dire que la valeur de sortie ne peut changer qu'au moment d'un front d'horloge, montant ou descendant selon les modèles.

- Pour J = K = 0, il y a conservation du dernier état logique Q_{n-1} indépendamment de l'horloge : état mémoire.
- Pour J = K = 1, le système bascule à chaque front d'horloge.
- Pour J différent de K, la sortie Q recopie l'entrée J et la sortie non Q recopie l'entrée K à chaque front d'horloge.

j	k	Qt+1
0	0	Qt
0	1	1
1	0	0
1	1	$\overline{Qt}$





chronogramme de fonctionnement

2-2 GRAFCET

2-2-1définition

Le terme Grafcet signifie Graphe Fonctionnel de Commande Etape Transition. C'est un outil dedescription graphique du cahier des charges d'un système automatisé.En effet, le processus du fonctionnement d'une machine est défini par une succession d'actions et de situations dans lesquelles se trouve le système. Le Grafcet est une représentation graphique permettant de décrire le fonctionnement du système. Il est caractérisé par une succession d'étapes et de transitions associées à leur réactivité (réceptivité)

2-2-2Element Du Grafcet

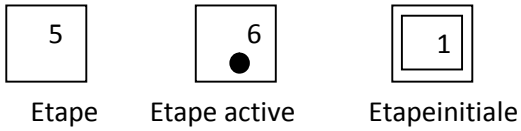
Le fonctionnement d'un système automatique peut être représenté graphiquement par un ensemble

- ✓ **ETAPES** aux quelles son associées des ACTIONS
- ✓ **TRANSITIONS** aux quelles son associées des RECEPTIVITES
- ✓ **LIAISON ORIENTEES** entre les étapes et les transitions, et des transitions aux étapes.

a) ETAPE.

L'étape correspond à une situation élémentaire ayant un comportement stable : pendant une étape, les organes de commande et les capteurs ne change pas d'état.

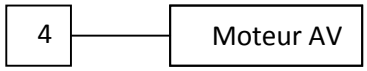
L'étape se représente par un carré repéré par un chiffre placé de préférence dans la moitié supérieur



Lorsqu'une étape est active, on peut le préciser par un point. Les étapes initialement actives en début de fonctionnement se représentent par un double carré.

b) Action Associees A L'etape

On précise pour chaque étape, à l'intérieur d'un rectangle, les actions à effectuée lorsque l'étape et active.

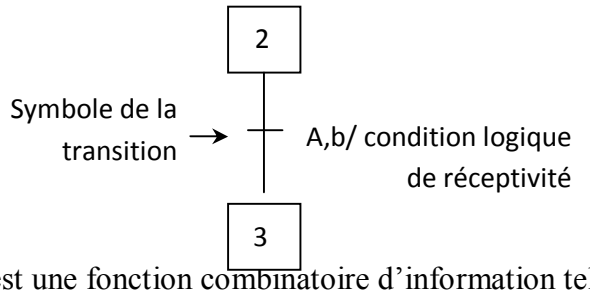


Les actions peuvent être de natures diverses, le rectangle peut avoir des dimensions quelconque et comporter plusieurs actions.

c) Transition

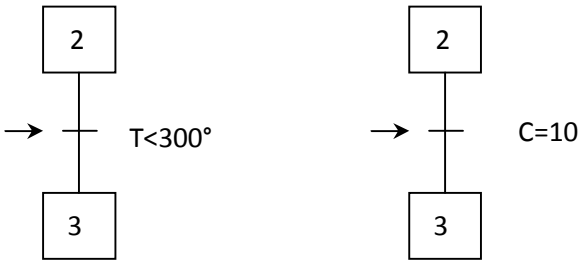
Une transition indique la possibilité d'évolution d'une étape à l'étape suivante.

A chaque transition on associe une ou des conditions logiques qui traduisent la notion de RECEPTIVITE.



La réceptivité est une fonction combinatoire d'information telle que

- ✓ Etats de capteurs ;
- ✓ Action de bouton poussoir par l'opérateur ;
- ✓ Action d'un temporisateur, d'un capteur ;
- ✓ Etat actif ou inactif d'autres étapes.



Température< 300° Valeur de consigne d'un compteur =10

d) Liaison Orientees

Les liaisons indique la vois d'évolution du GRAFCET. Dans les cas général, les liaisons qui se font du haut vers le bas ne comportent pas de flèche. Dans les autres cas, il faut utiliser des flèches.

2-2-3 Regles D'evolution

1er règle : l'initialisation précise l'étape active au début du fonctionnement. On repère en doublant les cotés des symboles correspondants. Il peut y avoir plusieurs étapes initiales.

NB. L'étape initiale est activée inconditionnellement en début de cycle.

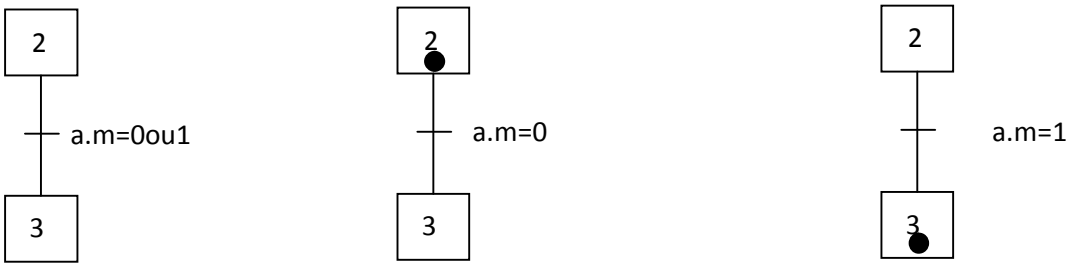
2eme règle : une transition et soit validée, soit non validée.

Elle est validée lorsque toutes les étapes qui immédiatement précédentes sont actives.

Elle ne peut être franchie que :

- lorsqu' elle est validée,

- et que la réceptivité associée à la transition est vraie.

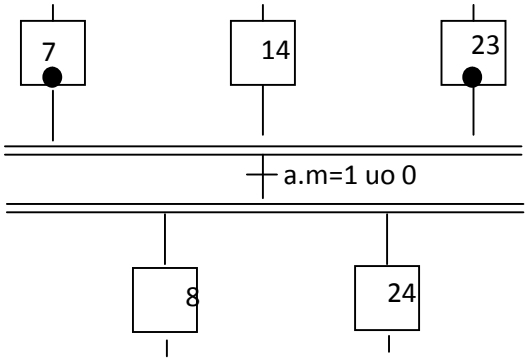


Transition non validée Etape 2 n'étant pas activée

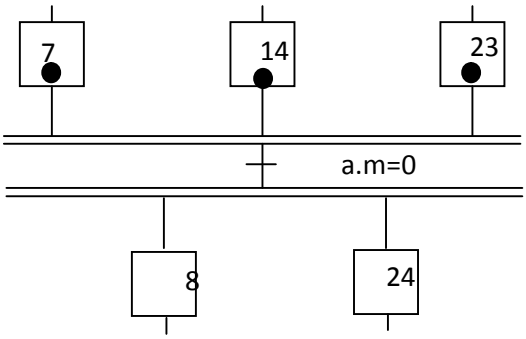
Transition validée Mais ne peut être franchie Réceptivitéestnulle

Transition franchie

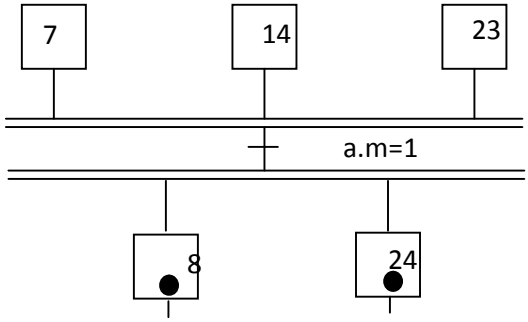
3eme Règle : la franchissement d'une transition entraine simultanément l'activation de toutes les étapes immédiatement suivantes et désactivation de toutes les étapes immédiatement précédentes.



Transition non validée :
L'étape 14 est inactive



Transition validée :
Les étapes 7,14,23 sont actives



Transition franchie :
a.m=1 entraine l'activation de 8 et 24 et la désactivation de 7, 14et 23

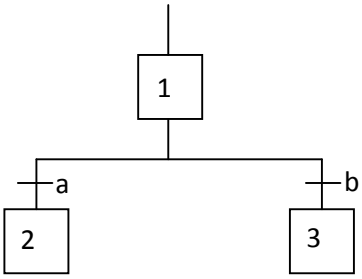
Règle 4 : évolutions simultanées

Plusieurs transitions simultanément franchissables sont simultanément franchies. Cette règle permet de décomposer un GRAFCET en plusieurs parties en assurant leurs intercon-nexions. Dans ce cas, il est indispensable de faire intervenir, dans les réceptivités, les états actifs ou inactifs des étapes.

Règle 5 : activation et désactivation simultanées Si, au cours du fonctionnement, une même étape doit être désactivée et activée simultanément, elle reste activée.

2-2-4 séquences multiples

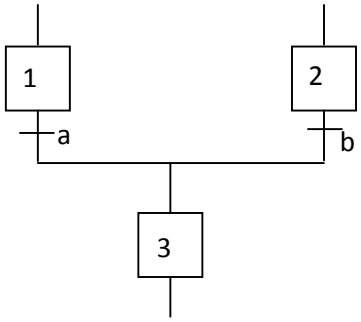
a) Divergence en ou (structure alternative)



Si 1 active et si a seul, alors désactivation de 1 et activation de 3, 2 inchangé.

Si a et b puis 1 active alors désactivation 1, activation 2 et 3 quel que soit leur état précédent.

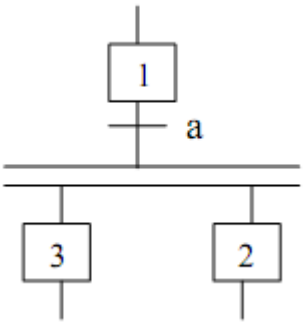
b) Convergence en ou (structure alternative)



Si 1 active et a sans b, alors activation de 3 et désactivation de 1, 2 reste inchangé

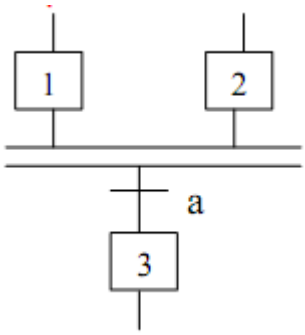
Si 1 et 2 et a et b alors 3 seule active.

c) Divergence en ET (structure simultanée)



Si 1 active et si a, alors désactivation de 1 et activation de 2 ET 3.

d) Convergence en ET (structure simultanée)



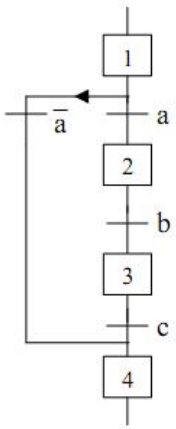
Si 1 active seule et a, alors aucun changement.

Si 1 ET 2 et a, alors activation de 3 et désactivation de 1 et 2.

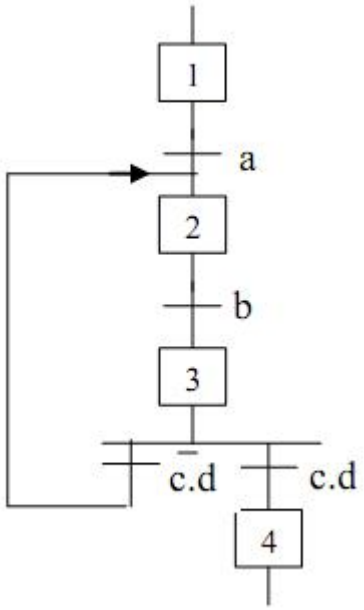
e)Saut d'étape

Il permet de sauter une ou plusieurs étapes :

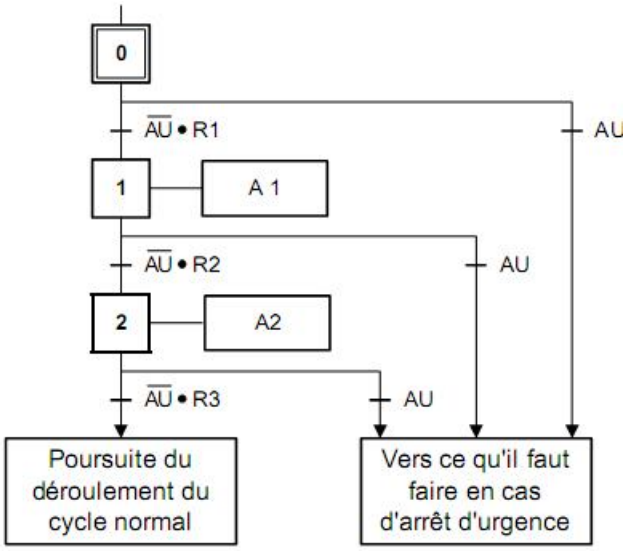
f) Boucle Si Alors



G) Boucle Répéter Tant que

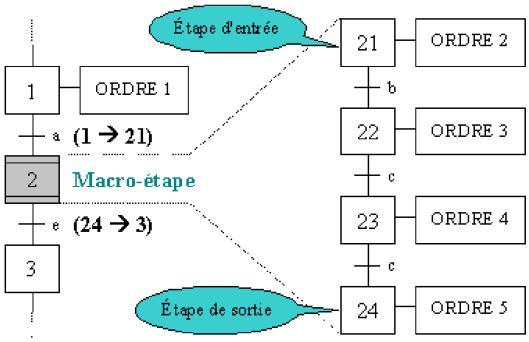


H) Séquenced'urgence



I) Macro-étape

Une macro-étape est le **résumé de la représentation d'un ensemble d'étapes et de transitions**, afin de ne pas **surcharger** le GRAFCET. L'ensemble de la macro-étape est défini dans le **graphisme d'expansion de la macro-étape**.



2-3 Choix de la logique de réalisation des SAP

Pour choisie la meilleur technologie pour un automatisme donné, on utilisegénéralement deux critères: la **faisabilité** et l'**optimisation**.

Critère 1: ***Faisabilité*** :

« La réalisation avec une technologie donnée est-elle possible ou non ? »

Critère 2: ***Optimisation*** :

« La réalisation avec une technologie donnée conduit-elle au coût global le plus bas ? »

- ✓ Le choix de la technologie peut être résumé comme suit :
  Le micro-ordinateur est utilisé surtout dans les cas des systèmes complexes (nombred'E/S assez grand, calculs sur les réels, etc.).
- ✓ Les cartes électriques spécifiques sont utilisées pour résoudre un problème bine défini. Elles sont appelées uniquement dans le cas où le nombre d'exemplaire est supérieur à 100 car leur coût est assez élevé.
- ✓ Les cartes électroniques standards sont utilisées dans les automatismes grand public : distributeurs, parking, etc.
- ✓ Les automates programmables sont utilisés dans les cas des systèmes complexes, flexible et évolutifs.

2-3-1 La logique câblée

La technologie câblée consiste à raccordé des modules par des liaisons matérielles selon un schéma fourni par la description. Ces modules peuvent être électromagnétiques, électriques, pneumatiques ou fluidiques.

En électricité ou en électronique, les liaisons sont faites par câble électrique. En pneumatique et fluidique, il s'agit de canalisations reliant les différentes composants.

Les outils câblés sont utilisés dans l'industrie où l'on apprécie leurs qualités éprouvées; à savoir la **rapidité** et le **parallélisme**. Ils souffrent cependant d'un certain nombre de limitations parmi lesquelles nous citons:

- ✓ Leur encombrement (poids et volume),
- ✓ leur manque de souplesse vis-à-vis de la mise au point des commandes et de l'évolution de celles-ci (améliorations, nouvelles fonctions, modification, etc.): Toute modification impose la modification de câblage voire un changement de composants,
- ✓ Leur difficulté de maîtriser des problèmes complexes,
- ✓ La complexité de recherche des pannes et donc du dépannage,
- ✓ Leur coût élevé pour les systèmes complexes.

2-3-2 La logique programmé

La technologie programmable consiste à substituer le fonctionnement de l'automatise par un programme chargé sur un constituant programmable c'est-à-dire des machines destinées à traiter de l'information. Leur utilisation en gestion et en calcul scientifique est connue. Alors, les applications techniques relèvent de l'informatique industrielle. L'informatique industrielle est une discipline conjuguant les théories de l'automatique et les moyens de l'informatique dans le but de résoudre des problèmes de nature industrielle.

L'informatique offre donc une alternative technologique à l'automaticien et lui ouvre des possibilités nouvelles liées à la puissance de traitement et aux facilités de mémorisation de l'information. En termes d'avantage, nous citons:

- ✓ Moins de câble et d'encombrement
- ✓ Fiabilité de l'automatisme.
- ✓ Facilité de modification
- ✓ Flexibilité
- ✓ Résolution des problèmes complexes.

Cependant, elle souffre de problème de **parallélisme**. Le constituant programmable peut être soitun micro-ordinateur, soit une carte électronique ou bien un automate programmable.

2-4 Mise en Oeuvre du GRAFCET

2-4-1 Réalisation par câblage

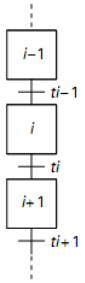
1. EQUATION LOGIQUE D'UNE ETAPE

Pour décrire l'activité d'une étape, nous utilisons la notion proposée par la norme NF c03-190.

Xi =1 signifie que l'étape i est active

Xi=0 signifie que l'étape i est inactive

Soit le partie du GRAFCET représenté si dessous



Détermination quelles sont les variable qui intervienne dans l'activité de l'étape i :

Xi=f(?)

une transition est franchissable si (règle 2) :

— elle est validée (toutes les étapes immédiatement précédente sont actives) ;

— la réceptivité associée à la transition est vraie.

La traduction de la règle 2 donne la condition d'activation de l'étape i (CAXi) :

CAXi= Xi-1 . ti-1

La traduction de la règle 3 donne la condition de désactivation de l'étape i (CDXi) :

CDXi= Xi+1

Si la condition d'activation et la condition de désactivation de l'étape i sont fausses, alors l'étape i reste dans son état (effet mémoire de la logique séquentielle) ; cela signifie que l'état de Xi dépend aussi de Xi. Donc :

Xi=f(Xi,CAXi,CDXi)

Il est possible de définir la table de vérité de **Xi** (tableau 1), et à partir du tableau de Karnaugh associé, de déduire l'équation logique d'une étape :

Table de vérité d'une étape

Xi(t)	CAXi	CDXi	Xi-(t+Δt)
0	0	0	0(2)
0	0	1	0
0	1	0	1
0	1	1	1 (1)
1	0	0	1(2)
1	0	1	0
1	1	0	1
1	1	1	1(1)

(1) Ces lignes garantissent la priorité à l'activation afin de respecter la règle 5.

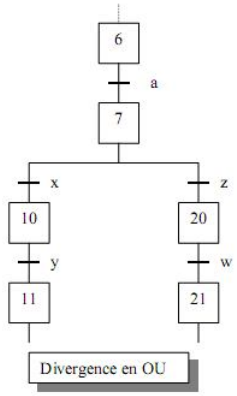
(2) Ces lignes correspondent à l'effet mémoire.

Tableau de Karnaugh de l'équation d'une étape

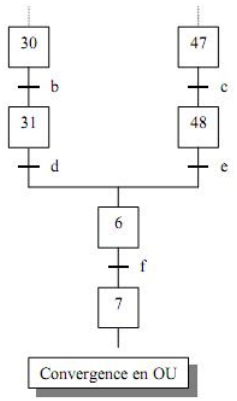
	CAXi.CDXi			
Xi	00	01	11	10
0	0	0	1	1
1	1	0	1	1

$$X_i = CAX_i + \overline{CAD}X_i$$

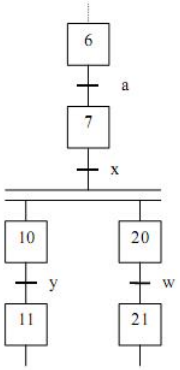
b) Cas d'un GRAFCET avec sélection de séquence



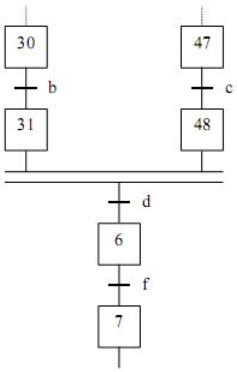
Etape	CAXi	CDXi
7	X6.a	X10+X20
10	X7.x	X11
20	X7.z	X21



Etape	CAXi	CDXi
31	X30.b	X.6
48	X47.c	X6
6	X31.d+X48.e	X7



Etape	CAXi	CDXi
7	X6.a	X10.x20
10	X7.x	X11
20	X7.x	X21



Etape	CAXi	CDXi
31	X30.b	X.6
48	X47.c	X6
6	X31.X48.d	X7

c) Gestion de mode marche/arrêt

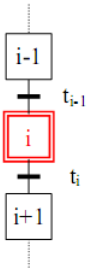
A l'initialisation du FRAFCET, toutes les étapes autre que les étapes initiales sont désactivées.

Soit la variable Init telle que

Init=1 : initialisation du GRAFCET : Mode ARRET

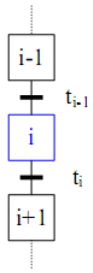
Init=0 : déroulement du cycle : Mode MARCHE

Equation d'uneétape « i » initial



CAXi	CDXi	Equation de Xi
$X_{i-1}.t_{i-1} + Init$	$X_{i+1}.\overline{Init}$	$X_i = CAX_i + \overline{CAD}X_i + Init$

Equation d'une étape « i » non initial



CAXi	CDXi	Equation de Xi
$X_{i-1}.t_{i-1} + \overline{Init}$	$X_{i+1} + Init$	$X_i = (CAX_i + \overline{CADX_i}).\overline{Init}$

d) Realisation Du Grafcet Par Bascules RS (reset/set)

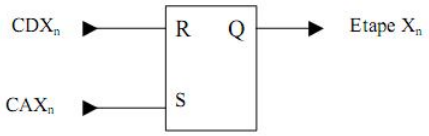
Table de vérité d'une bascule RS

R	S	Qt+1
0	0	Qt
0	1	1
1	0	0
1	1	-

Si la bascule RS est appliquée au GRAFCET.

La condition d'activation d'une étape est alors câblée sur le SET de la bascule.

La condition désactivation d'une étape est alors câblée sur le RESET de la bascule.



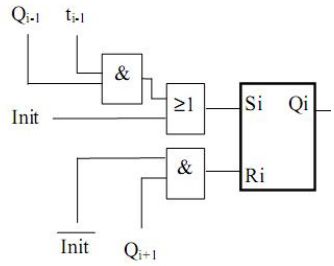
Cablage d'une etape initiale :

Qui permet de faire passer cet etape a l'etat 1 ? l'étape anterieur et la reciptivité de l'etape anterieur ou l'entrée initialisation Init.

Qui permet de faire passer cet etape a l'etat 0 ? l'etape suivante.

$$CAX_i = X_{i-1}.t_{i-1} + Init$$

$$CDX_i = X_{i+1} + \overline{Init}$$



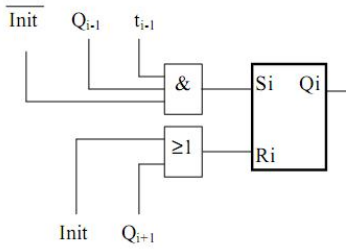
Cas d'une etape classique :

Qui permet de faire passer cet etape a l'etat 1 ? l'étape anterieur et la reciptivité de l'etape anterieur.

Qui permet de faire passer cet etape a l'etat 0 ? l'etape suivante ou aussi l'entrée initialisation Init.

$CAX_i = X_{i-1}.t_{i-1}.\overline{Init}$

$CDX_i = X_{i+1} + Init$



Exemple.

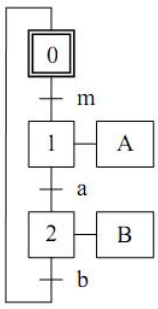
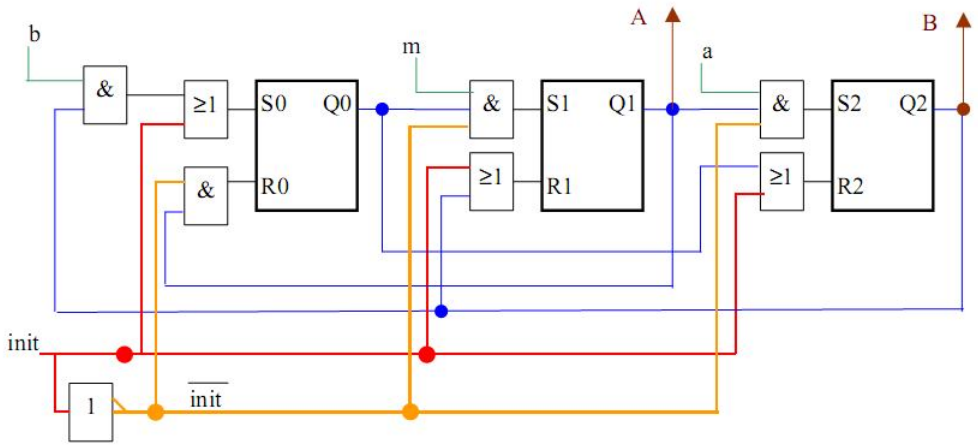


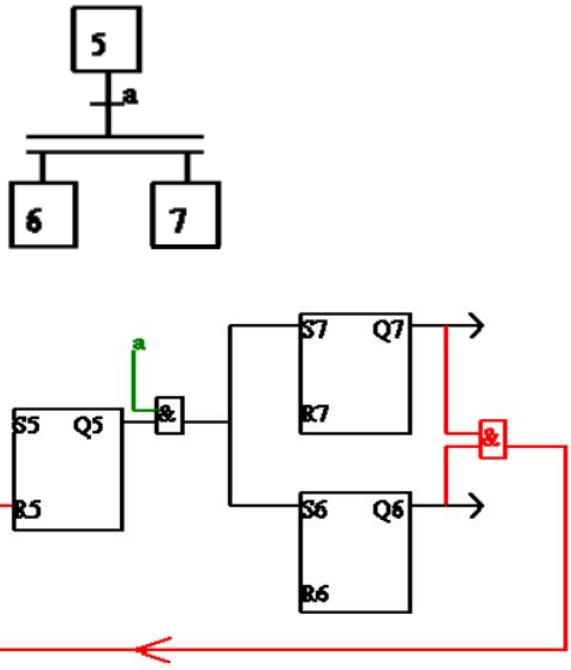
Table de condition da'ctivation et désactivation des etape :

Xn	CAXn	CDXn
0	$X_2.b + Init$	$X_1\overline{Init}$
1	$X_0.m.\overline{Init}$	$X_2 + Init$
2	$X_1.a.\overline{Init}$	$X_0 + Init$



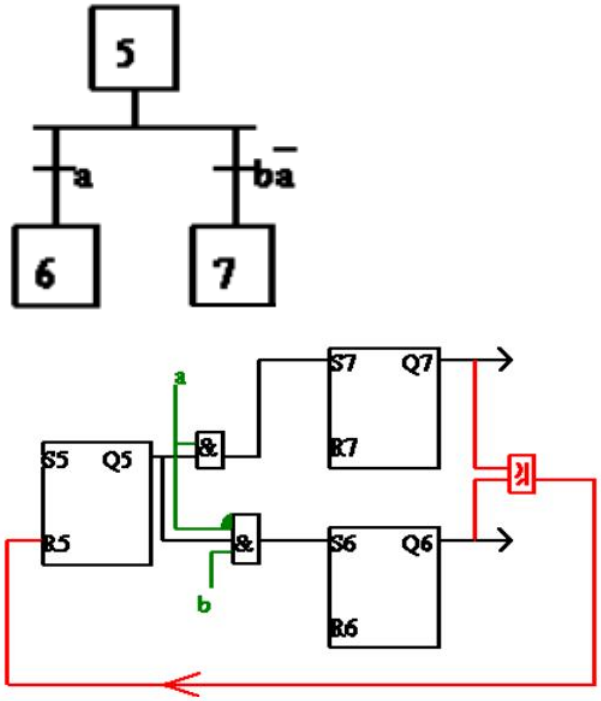
Divergence simple en ET

Quand la transition est franchissable, il suffit d'allumer deux étapes au lieu d'une. Le seul problème est la désactivation de l'étape précédente : il faut être sûr que les deux étapes suivantes ont eu le temps de prendre en compte l'information d'activation avant de désactiver la précédente (si l'on désactive dès qu'une des deux est active, la seconde ne s'activera plus).



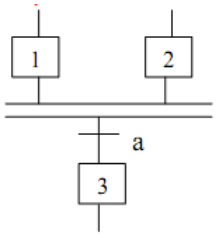
Divergence exclusive en OU

Quand la transition et franchissable, il suffit d'allumer une étape au lieu de deux.



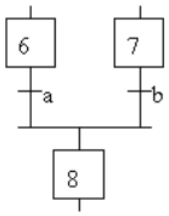
Convergence en et

Il faut que les deux étapes suivante soit active, et que la réceptivité vraie, pour activer l'étape suivante, celle-ci désactivant les étapes précédentes.



Convergence en ou

On allume 8 si (6 et a) ou (7 et b). On éteint 6 et 7 tant que l'on a 8. Evidement ceci ne fonctionne que si l'on ne peut pas avoir simultanément 6 et 7 actives, mais j'ai bien dit (dans le titre ci-dessus) que je ne traite que le cas simple, qui de plus se trouve être aussi le plus courant.



2-4-2 Utilisation d'un automate

Un API (automate programmable industriel) est un matériel programmable pouvant être placé directement dans un environnement de production industrielle (sans aller jusqu'à accepter des jets de solvants). Ils possèdent un premier étage d'interfaçage (entres - sorties en 24 V par exemple, même quelquefois 220 V si l'on n'utilise qu'une faible intensité), un second étage spécifique à chaque actionneur devra être installé (mais disponible dans le commerce pour les cas habituels) : distributeurs , thyristors, relais... Les prix s'étalent de quelques kF à quelques centaines de kF.En entrée de gamme, on trouve des automates avec quelques entrées - sorties ToR (Tout ou Rien), et un langage permettant de simuler les composants de base (ET, OU, NON, bascule, tempo, compteur). A un niveau de prix supérieur, on trouvera des systèmes avec un environnement de programmation de haut niveau (PC par exemple), simplifiant grandement la programmation et les tests (en particulier la possibilité deprogrammer directement l'automate en Grafcet). En haut de gamme on a des API modulaires : on les compose sur mesure, en ajoutant suivant les besoins des cartes d'E/S, des cartes numériques, des conversions numérique - analogique, des asservissements (DSP par exemple)... Excepté en entrée de gamme, les API permettent désormais des dialogues en réseaux d'automates, ce qui permet d'utiliser un superviseur qui ne fait que commander d'autres automates dans lesquels on a décentralisé les tâches (l'usage des réseau estgrandement facilité par l'utilisation d'automates compatibles entre eux, en général de la même marque). On utilisera un API pour des automatismes unitaires, le câblage n'étant intéressant que pour une fabrication en série de produits automatisés.

CHAPITRE 3:Automate programmable industriel (API)

3-1 Contraintes du monde industriel :

3-1-1 Influences externes :

- ✓ *poussières,*
- ✓ *température,*
- ✓ *humidité,*
- ✓ *vibrations,*
- ✓ *parasites électromagnétiques, ...*

3-1-2 Personnel :

- ✓ *mise en œuvre du matériel aisée (pas de langage deprogrammation complexe)*
- ✓ *dépannage possible par des techniciens de formation électromécaniqu.*
- ✓ *possibilité de modifier le système en cours de fonctionnement*

3-1-3 Matériel :

- ✓ *évolutif*
- ✓ *modulaire*
- ✓ *implantation aisée*

3-2 Définition d'un Automate Programmable

Un automate programmable est un appareil dédié au contrôle d'une machine ou d'un processus industriel, constitué de composants électroniques, comportant une mémoire programmable par un utilisateur non informaticien, à l'aide d'un langage adapté. En d'autres termes, un automate programmable est un calculateur logique, ou ordinateur, au jeu d'instructions volontairement réduit, destiné à la conduite et la surveillance en temps réel de processus industriels

Trois caractérises fondamentales distinguent totalement l'Automate Programmable Industriel (API) des outils informatiques tels que les ordinateurs (PC industriel ou autres) :

- il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses entrées/sorties industrielles**.**
- il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, micro-coupures de la tension d'alimentation, parasites, etc.)

- et enfin, sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme fait en sorte que sa mise en oeuvre et son exploitation ne nécessitent aucune connaissance en informatique

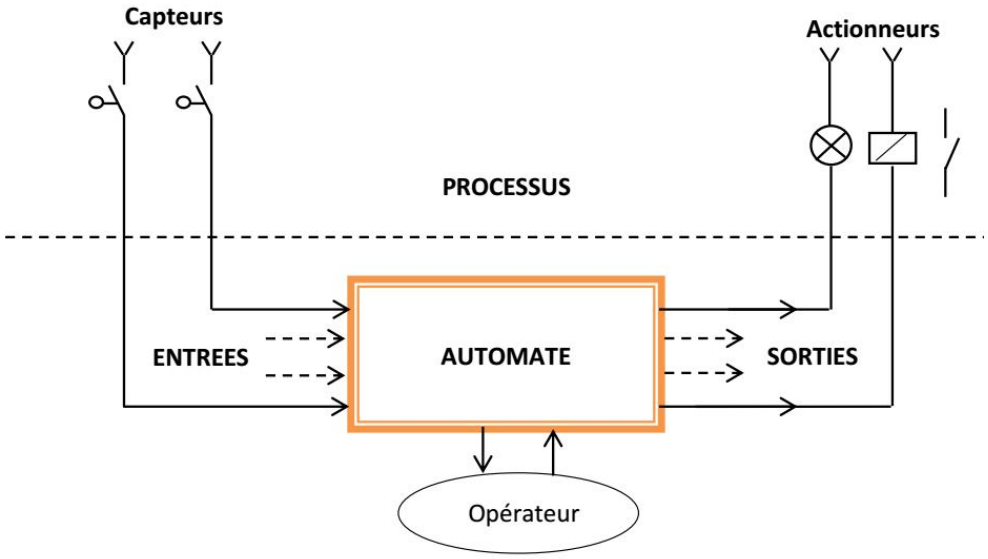


Figure 3.1.

3-3Architecture d'un API

La structure interne d'un API peut se représenter comme suit:

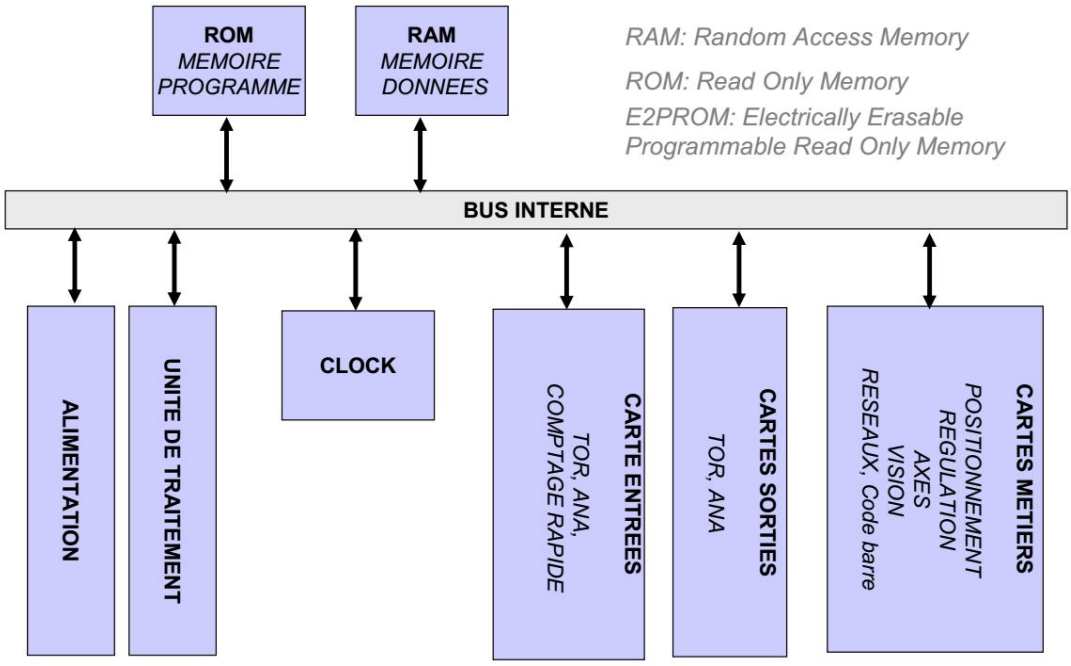


Figure 3.2. architecture interne de L'API

3-3-1 Le processeur

Le processeur, ou unité centrale (UC), a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement de l'application (les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul PID, etc..). Mais en dehors de cette tâche de base, il réalise également d'autres fonctions

- Gestion des entrées/sorties.
- Surveillance et diagnostic de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement.
- Dialogue avec le terminal de programmation, aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications des données.

Un ou plusieurs processeurs exécutent ces fonctions grâce à un micro logiciel préprogrammé dans une mémoire de commande, ou mémoire système. Cette mémoire morte définit les fonctionnalités de l'automate. Elle n'est pas accessible à l'utilisateur.

3-3-2 La mémoire

Elle est destinée au stockage des instructions qui constituent le programme de fonctionnement de l'automatisme, ainsi que des données qui peuvent être**.**

- Des informations susceptibles d'évoluer en cours de fonctionnement de l'application. C'est le cas par exemple de résultats de traitements effectués par le processeur et rangés dans l'attente d'une utilisation ultérieure. Ces données sont appelées variables internes ou mots internes**.**
- Des informations qui n'évoluent pas au cours de fonctionnement, mais qui peuvent en cas de besoin être modifiées par l'utilisateur : textes à afficher, valeurs de présélection, etc.. Ce sont des mots constants**.**
- Les mémoires d'état des entrées/sorties, mises à jour par le processeur à chaque tour de scrutation du programme**.**

Deux familles de mémoires sont utilisées dans les automates programmables**:**

- Les mémoires vives, ou mémoires à accès aléatoire « Random Access Memory (RAM) ». Le contenu de ces mémoires peut être lu et modifié à volonté, mais il est perdu en cas de manque de tension (mémoire volatiles). Elles nécessitent par conséquent une sauvegarde par batterie. Les mémoires vives sont utilisées pour l'écriture et la mise au point du programme, et pour le stockage des données**.** Elles sont à lecture seule, les informations ne sont pas perdues lors de la coupure de l'alimentation des circuits. On peut citer les types suivants :
  - ✓ **ROM « Read Only Memory » :** Elle est programmée par le constructeur et son programme ne peut être modifié**.**
  - ✓ **PROM « Programmable ROM » :** Elle est livrée non enregistrée par le fabricant. Lorsque celle-ci est programmée, on ne peut pas l'effacer**.**
  - ✓ **EPROM « Erasable PROM » :** C'est une mémoire PROM effaçable par un rayonnement ultraviolet intense**r.**
  - ✓ **EEPROM « Electrically EPROM » :** C'est une mémoire PROM programmable plusieurs fois et effaçable électriquement**.**
  - ✓ **Mémoire Flash :** C'est une mémoire EEPROM rapide en programmation. L'utilisateur peut effacer un bloc de cases ou toute la mémoire**.**

La mémoire morte est destinée à la mémorisation du programme après la phase de mise au point. La mémoire programme est contenue dans une ou plusieurs cartouches qui viennent s'insérer sur le module processeur ou sur un module d'extension mémoire**.**

3-3-3 Les interfaces entrées/sorties

Les entrées/sorties TOR (Tout ou Rien) assurent l'intégration directe de l'automate dans son environnement industriel en réalisant la liaison entre le processeur et le processus. Elles ont toutes, de base, une double fonction**:**

- Une fonction d'interface pour la réception et la mise en forme de signaux provenant de l'extérieur (capteurs, boutons poussoirs, etc.) et pour

l'émission de signaux vers l'extérieur (commande de pré-actionneurs, de voyants de signalisation, etc.). La conception de ces interfaces avec un isolement galvanique ou un découplage opto-électronique assure la protection de l'automate contre les signaux parasites**.**

- Une fonction de communication pour l'échange des signaux avec l'unité centrale par l'intermédiaire du bus d'entrées/sorties**.**

Le fonctionnement de l'interface d'entrée peut être résumé comme suit**:**

Lors de la fermeture du capteur.

- ✓ La « **Led 1** » signale que l'entrée de l'API est actionnée**.**
- ✓ La « **Led D'** » de l'optocoupleur « **Opto 1** » s'éclaire**.**
- ✓ Le phototransistor « **T'** » de l'optocoupleur « **Opto 1** » devient passant**.**
- ✓ La tension Vs=0V**.**

Donc lors de l'activation d'une entrée de l'automate, l'interface d'entrée envoie un « 0 » logique à l'unité de traitement et un « 1 » logique lors de l'ouverture du contact du capteur (entrée non actionnée).

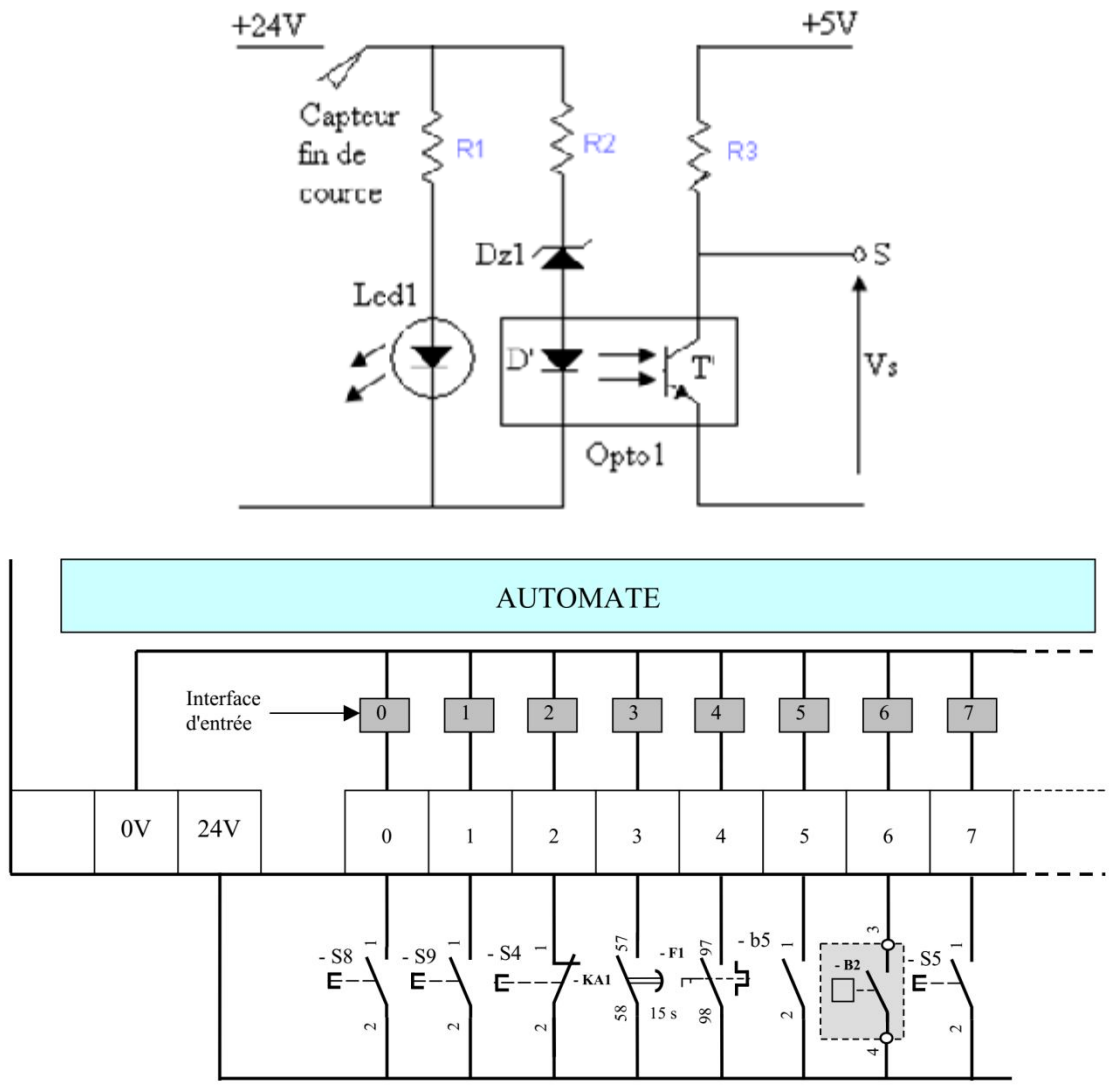


Figure 3.3. principe de fonctionnement de l'interface d'entrée

Le fonctionnement de l'interface de sortie peut être résumé comme suit

Lors de commande d'une sortie automate

- ✓ L'unité de commande envoi un « **1** » logique (5V)
- ✓ « **T1** » devient passant, donc la « **Led D'** » s'éclaire
- ✓ Le photo-transistor « **T'** » de l'optocoupleur « **Opto1** » devient passant.
- ✓ La « **Led1** » s'éclaire.
- ✓ « **T2** » devient passant.
- ✓ La bobine « **RL1** » devient sous tension et commande la fermeture du contact de la sortie « **Q0.1** ».

Donc pour commander un API, l'unité de commande doit envoyer

- ✓ Un « **1** » logique pour actionner une sortie API
- ✓ Un « **0** » logique pour stopper la commande d'une sortie API

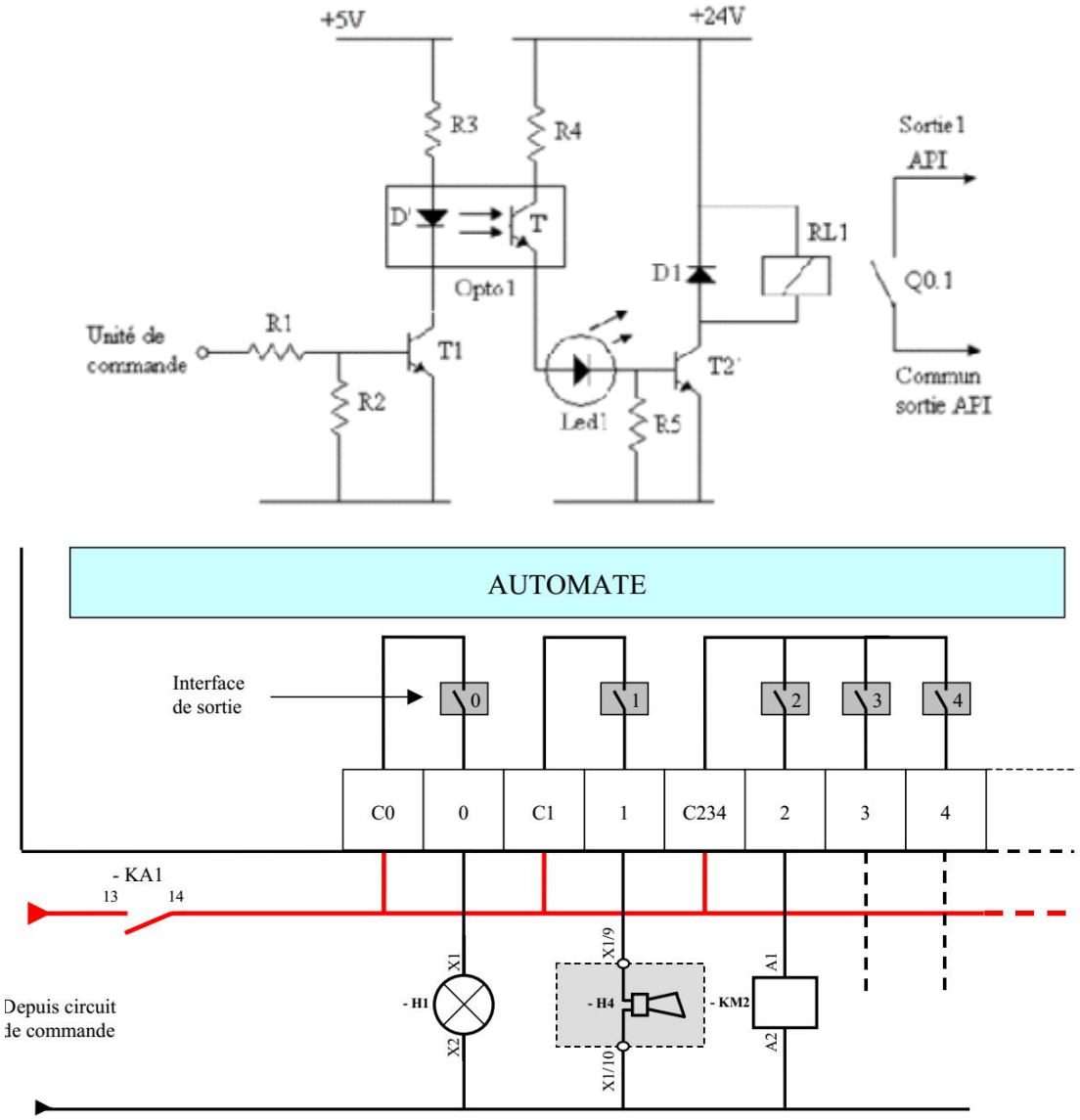


Figure 3.4 principe de fonctionnement de l4interface sortie.

3-3-4 Le Bus

C'est un ensemble de conducteurs qui réalisent la liaison entre les différents éléments de l'automate. Dans un automate modulaire, il se présente sous forme d'un circuit imprimé situé au fond du bac et supporte des connecteurs sur lesquels viennent s'enficher les différents modules : processeur, extension mémoire, interfaces et coupleurs**.**

Le bus est organisé en plusieurs sous ensembles destinés chacun à véhiculer un type bien défini d'informations**.**

- ✓ Bus de données**.**
- ✓ Bus d'adresses**.**
- ✓ Bus de contrôle pour les signaux de service tels que tops de synchronisation, sens des échanges, contrôle de validité des échanges, etc**...**
- ✓ Bus de distribution des tensions issues du bloc d'alimentation**.**

3-3-5 Alimentation

Elle élabore à partir d'un réseau 220V en courant alternatif, ou d'une source 24V en courant continu, les tensions internes distribuées aux modules de l'automate**.**

A fin d'assurer le niveau de sûreté requis, elle comporte des dispositifs de détection de baisse ou de coupure de la tension réseau, et de surveillance des tensions internes. En cas de défaut, ces dispositifs peuvent lancer une procédure prioritaire de sauvegarde**.**

3-3-6 Les Cartes

a)Cartes d'entrées / sorties : Au nombre de 4, 8, 16 ou 32, elles peuvent aussi bien réaliser des fonctionsd'entrées, de sorties ou les deux.Ce sont les plus utilisées et les tensions disponibles sont normalisées (24, 48, 110 ou 230V continu oualternatif ...).Les voies peuvent être indépendantes ou posséder des "communs".

Les cartes d'entrées permettent de recueillir l'information des capteurs, boutons ... qui lui sontraccordés et de la matérialiser par un **bit image** de l'état du capteur.

Les cartes de sorties offrent deux types de technologies : les sorties à relais électromagnétiques(bobine plus contact) et les sorties statiques (à base de transistors ou de triacs).

b)Cartes de comptage rapide : elles permettent d'acquérir des informations de fréquences élevéesincompatibles avec le temps de traitement de l'automate. Exemple : signal issu d'un codeur de position.

c)Cartes de commande d'axe : Elles permettent d'assurer le positionnement avec précision d'élémentmécanique selon un ou plusieurs axes. La carte permet par exemple de piloter un servomoteur et derecevoir les informations de positionnement par un codeur. L'asservissement de position pouvant êtreréalisé en boucle fermée.

d)Cartes d'entrées / sorties analogiques : Elles permettent de réaliser l'acquisition d'un signal analogiqueet sa conversion numérique (CAN) indispensable pour assurer un traitement par le microprocesseur.La fonction inverse (sortie analogique) est également réalisée.Les grandeurs analogique sontnormalisées : 0-10V ou 4-20mA.

e) Cartes de régulation PI

Cette carte comporte souvent plusieurs entrées analogiques permettant de recevoir le signal de mesure issu des transmetteurs 4-20mA, ainsi que plusieurs sorties analogiquespermettant de piloter les vannes de régulation.Le microprocesseur local traite le programme élaboré à partir des différents algorithmes derégulation implantés sur la carte même (PID, Sommation, Racine carrée,..).

f) Cartes de communication (Ethernet ...)

La **carte réseau** (appelée *Network Interface Card* en anglais et notée **NIC**) constitue l'interface entre l'automat et le câble du réseau. La fonction d'une carte réseau est de préparer, d'envoyer et de contrôler les données sur le réseau.

La carte réseau possède généralement deux témoins lumineux (LEDs) :

- ✓ La LED verte correspond à l'alimentation de la carte ;
- ✓ La LED orange (10 Mb/s) ou rouge (100 Mb/s) indique une activité du réseau (envoi ou réception de données).

Pour préparer les données à envoyer, la carte réseau utilise un transceiver qui transforme les données parallèles en données séries. Chaque carte dispose d'une adresse unique, appelée adresse MAC, affectée par le constructeur de la carte, ce qui lui permet d'être identifiée de façon unique dans le monde parmi toutes les autres cartes réseau.

La plupart des cartes réseau destinées au grand public sont des cartes Ethernet. Elles utilisent comme support de communication des paires torsadées (8 fils encuivre),disposant à chaque extrémité de prises RJ45.

Les trois standards Ethernet (norme 802.3) les plus courants correspondent aux trois débits les plus fréquemment rencontrés :

- ✓ Le **10Base-T** permet un débit maximal de 10 Mbit/s. Le câble RJ45 peut alors mesurer jusqu'à une centaine de mètres et seuls 4 des 8 fils sont utilisés.
- ✓ Le **100Base-TX** permet un débit maximal de 100 Mbit/s. Il est également appelé Fast Ethernet et est désormais supporté par la quasi-totalité des cartes réseau. Comme pour le 10Base-T, le câble RJ45 peut alors mesurer jusqu'à une centaine de mètres et seuls 4 des 8 fils sont utilisés.
- ✓ Le **1000Base-T** permet un débit maximal de 1 000 Mbit/s. Il est également appelé Gigabit Ethernet et se démocratise rapidement. Pour que le réseau fonctionne correctement, le câble RJ45 peut toujours mesurer jusqu'à 100 m, mais doit être de bonne qualité. Cette fois, les 8 fils sont utilisés.

3.4Types des automates

Les automates peuvent être de type compact ou modulaire.

3.4.1 De type compact

On distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouzet ...) des microautomates.

Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes.

Automate compact (Allen-bradley)

3.4.2 De type modulaire

le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs).

Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.

Automate modulaire (Modicon)

3.5 Choix d'un API

Le choix d'un API est fonction de la partie commande à programmer. On doit tenir compte de plusieurs critères**.**

- Nombre d'entrées / sorties : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé.
- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- Fonctions ou modules spéciaux : certaines cartes permettront de "soulager" le processeur et devront offrir les caractéristiques souhaitées.
- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards

CHAPITRE 4: LES LANGAGES DE PROGRAMATION

4-1 Function Block Diagram (FBD)

Le langage FBD (function block diagram) est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standard, de fonctions ou de blocs fonctionnels.

Les principales fonctions sont:

- l'énonce **RETURN** (peut apparaître comme une sortie du diagramme, si liaison connectée prend l'état booléen **TRUE**, la fin du diagramme n'est pas interprétée.
- Les étiquettes et les sauts conditionnels sont utilisés pour contrôler l'exécution du diagramme. Aucune connexion ne peut être réalisée à droite d'un symbole d'étiquette ou de saut.
- saut à une étiquette (le nom de l'étiquette est « **LAB** »).

Si la liaison à gauche du symbole de saut prend l'état booléen **TRUE**, l'exécution du programme est déroutée après l'étiquette correspondante.

L'inversion booléenne est représentée par un petit cercle.

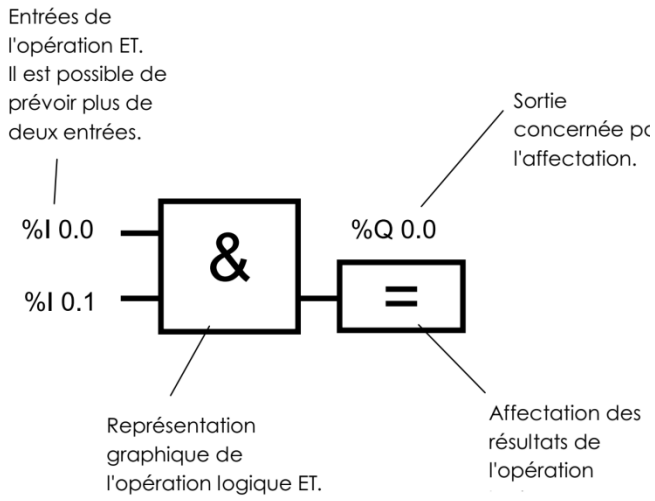
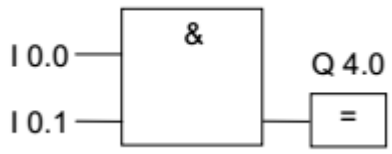


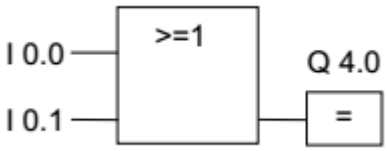
Figure 4.1 . schéma du porte et en langage FBD

4-1-1 Le ET logique



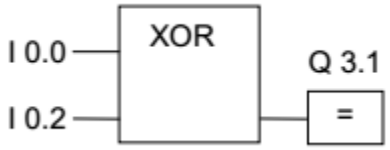
la sortie Q4.0 =1 lorsque les deux entrées =1.

4-1-2 Le OU logique



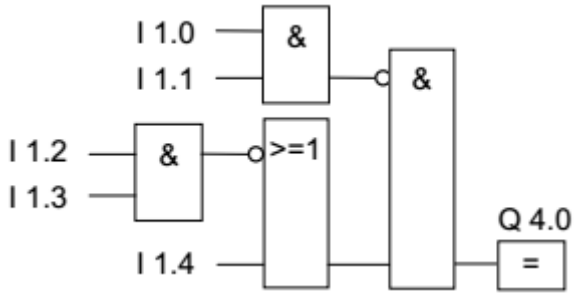
la sortie Q4.0 =1 lorsque deux entrées =1.

4-1-3 Le OU exclusif (XOR)



la sortie Q3.1 =1 lorsque deux entrées =1 et n'est pas les deux.

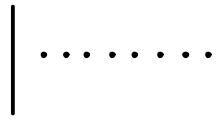
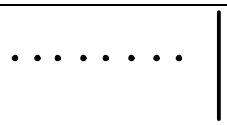
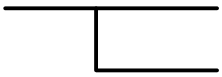
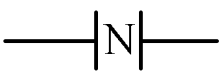
4-1-4 La Négation logique ()

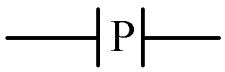
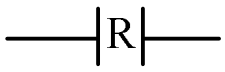


4-2 Ladder Diagram (LD)

Le langage LD (ladder diagram) est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation.

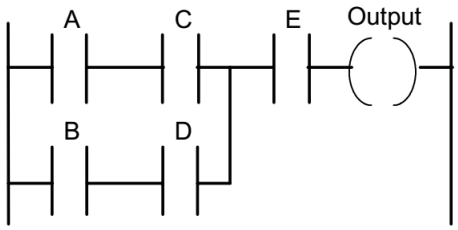
Les composants graphiques élémentaires d'un diagramme LD sont :

Symbole	Fonction
	Barre d'alimentation a gauche
	Barre d'alimentation a droite
	Arc de liaison horizontal
	Arc de liaison vertical
	Liaison multiple horizontal et vertical
	Contact associe a une variable
	Relais associe a une variable interne ou de sortie
	Contact inverse
	Contact a détection front descendant

	Contact a détection front descendant
	Relai a action Set
	Relai a action Reset

Exemple

la fonction logique ((A.C)+(B.D)).C se representer en lqdder comme suite.



4.3-Structured Text (ST)

Le langage **ST** (Structured Text) est un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques. C'est le langage par défaut pour la programmation des actions dans les étapes et des conditions associées aux transitions du langage **SFC.**

Un programme **ST** est une suite d'énoncés. Chaque énoncé est terminé par un point virgule (« ; »). Les noms utilisés dans le code source (identificateurs de variables, constantes, mots clés du langage...) sont délimités par des séparateurs passifs ou des séparateurs actifs, qui ont un rôle d'opérateur. Des commentaires peuvent être librement insérés dans la programmation**.**

Les types d'énoncés standard sont :

- assignation (variable := expression;)**.**
- appel de fonction

- appel de bloc fonctionnel
- énoncés de sélection (**IF, THEN, ELSE, CASE**)
- énoncés d'itération (**FOR, WHILE, REPEAT**)
- énoncés de contrôle (**RETURN, EXIT**)
- opérateurs booléens (**NOT, AND, OR, XOR**).
- énoncés spéciaux pour le lien avec le langage **SFC.**

Il est recommandé de respecter les règles suivantes quand on utilise les séparateurs passifs, pour assurer une bonne lisibilité du code source**.**

- ne pas écrire plusieurs énoncés sur la même ligne
- utiliser les tabulations pour indenter les structures de contrôle**.**
- insérer des commentaires**.**

Exemple

IF (I:000/00=1)THEN
O: 000/00 :-1;
ELSE
O: 000/00 =0;
End_IF;

4.4 instruction list (IL)

C'est un langage textuel, qui est le plus proche du comportement interne de l'automate. Leprogramme se compose d'une suite de lignes, chacune spécifiant un code opération suivi d'unseul opérande.

En langage LIST vous disposez des opérations de base suivantes :

- «**U** »ET« **LAB** »).
- «**UN**»ET NON
- «**O**»OU
- «**ON**»OU NON
- «**X**»OU exclusif
- «**XN**»OU NON exclusif

Les opérations suivantes permettent de combiner des parties de séquence combinatoire figurantentre parenthèses :

- «**U(**»ET d'une expression
- «**UN(**»ET NON d'une expression
- «**O(**»OU d'une expression

- «**ON(**»OU NON d'une expression
- «**X(**»OU exclusif d'une expression
- «**XN(**»OU NON exclusif d'une expression**)** Fermer la parenthèse d'une expression

Les opérations suivantes mettent fin à une séquence combinatoire :

- «= »Affectation (Cette opération sauvegarde le RLG dans le bit en accès)
- «**R** »Mettre à 0 (Cette opération écrit 0 dans le bit en accès si le RLG est égale à 1)
- «**S** »Mettre à 1 (Cette opération écrit 1 dans le bit en accès si le RLG est égale à 1)

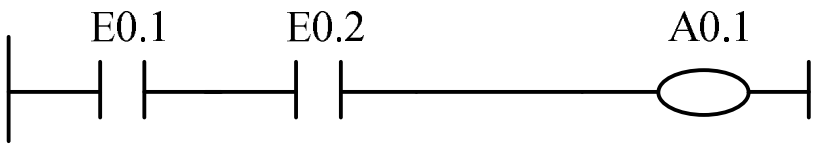
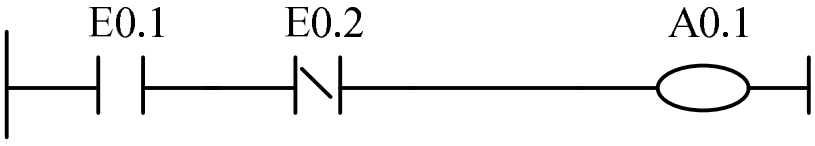
Les opérations suivantes vous permettent de modifier le résultat logique RLG :

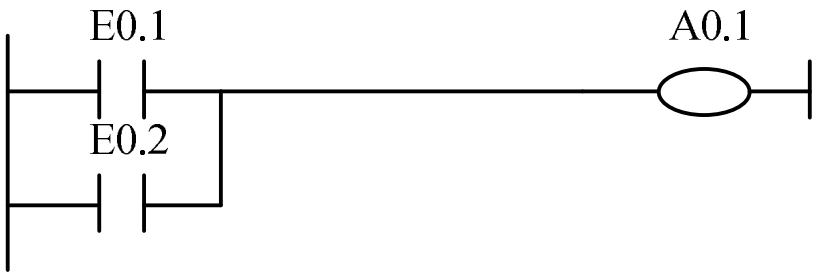
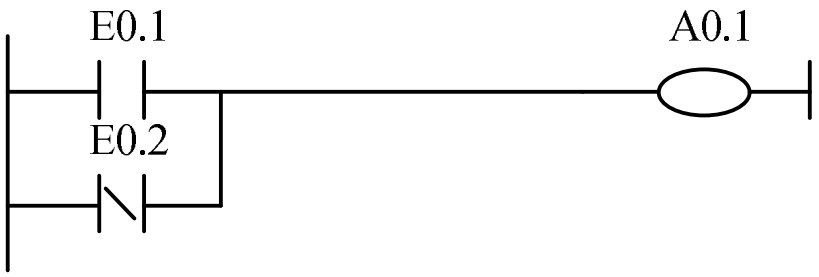
- «**NOT**»Négation du RLG
- «**SET** »Mettre RLG à 1
- «**CLR** »Mettre RLG à 0
- «**SAVE** »Sauvegarder RLG dans le bit RB

Les opérations suivantes détectent les transitions dans le résultat logique RLG et y réagissent :

- «**FN** »Front descendant
- «**FP** »Front montant

exemple

LIST	LADDER
U E 0.1 U E 0.2 = A 0.1	
U E 0.1 UN E 0.2 = A 0.1	

O E 0.1 O E 0.2 = A 0.1	
O E 0.1 ON E 0.2 = A 0.1	

4.5- Sequential Flow Chart (SFC)

Le langage **SFC** (Sequential Function Chart), ou **GRAFCET**, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages **ST, IL, LD** ou **FBD**.

Les principales règles graphiques sont:

- un programme **SFC** doit contenir au moins une étape initiale.
- une étape ne peut pas être suivie d'une autre étape.
- une transition ne peut pas être suivi d'une autre transition.

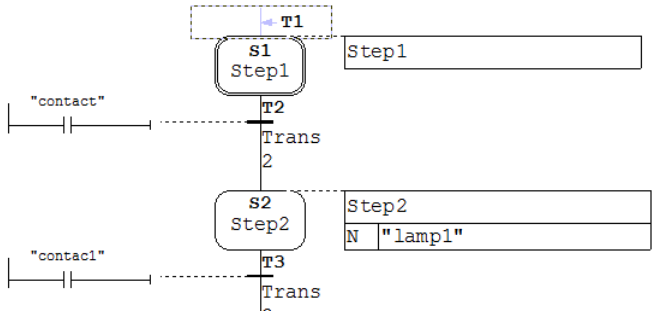
Les composants de base (symboles graphiques) du graphique SFC sont:

- étapes et étapes initiales.
- transitions.
- liaisons orientées.
- renvoi à une étape.

Les différents types d'action sont :

- action booléenne (Elle est forcée à chaque fois que le signal d'activité de l'étape change d'état).

- action impulsionnelle programmée en **ST**, **LD** ou **IL** (c'est une liste d'instructions **ST, IL** ou **LD**, exécutée à chaque cycle pendant toute la durée d'activité de l'étape).
- action impulsionnelle programmée en **ST**, **LD** ou **IL** (c'est une liste d'instructions **ST, IL** ou **LD**, exécutée à chaque cycle pendant toute la durée d'activité de l'étape).
- action normale programmée en **ST**, **LD** ou **IL**.
- action **SFC** (Une action **SFC** est une séquence fille **SFC**, lancée ou tuée selon les évolutions du signal d'activité de l'étape. Elle peut être décrite avec les qualificatifs d'action **N** (non mémorisée), **S** (set), ou **R** (reset).)
- Plusieurs actions (de même type ou de types différents) peuvent être décrites dans la même étape. Un appel de fonctions ou de blocs fonctionnels permet d'intégrer des traitements décrits dans d'autres langages (**FBD**, **LD**, **ST** ou **IL**).

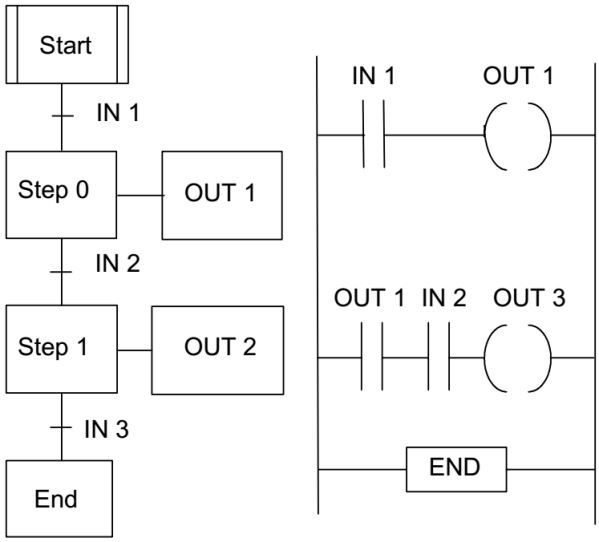


les attributs suivants peuvent être associés aux actions en SFC.

Attribut	Paramètre	Description
N	Aucun	**Action normal**: l'action est activée lorsque l'étape associée est active, et se désactive automatiquement à la sortie de l'étape.
S,R	Aucun	**Action mémorisé** : le (set) permet de mémoriser l'activation de l'action , celle-ci restera active jusqu'a ce que le (Rset) soit rencontré dans le grafcet sur la même action.
L	Durée d'activation	**Action limitée dans le temps**: l'action est activée lorsque l'étape puis désactivée automatiquement au bout de temps
D	Temps de retard	**Action retardée**:l'action est activée avec le retard spécifié après l'entrée dans létape.

Exemple

langage grafcet et sont eauivqlent en ladder.



4-6 Comparisons Entre les Languages

LANGUAGE	AVANTAGES	INCONVENIENTS
LD	facile à lire et à comprendre par la majorité des électriciens langage de base de tout PLC	suppose une programmation bienstructurée
FBD	Très visuel et facile à lire	Peut devenir très lourd lorsque les équations se compliquent
ST	Langage de haut niveau (langage pascal)Pour faire de l'algorithmique	Pas toujours disponible dans lesateliers logiciels
IL	angage de base de tout PLC type assembleur	très lourd et difficile à suivre si leprogramme est complexe. Pas visuel.
SFC	Description du fonctionnement (séquentiel) de l 'automatisme. Gestion des modes de marches Pas toujours accepté dans l'industrie…	Peu flexible

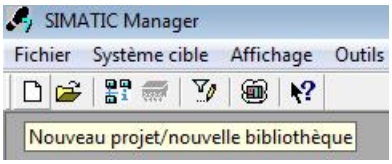
4-7 Projet ladder sur step 7

4-7-1 Configuration De L'automate Siemens

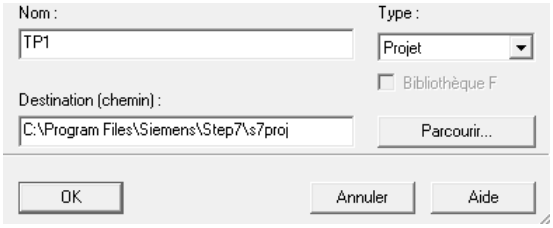
a) Créer un projet

Dans le bureau de Windows, double-cliquer sur l'icône « SIMATIC Manager » :

Cliquer ensuite sur l'icône « nouveau » :

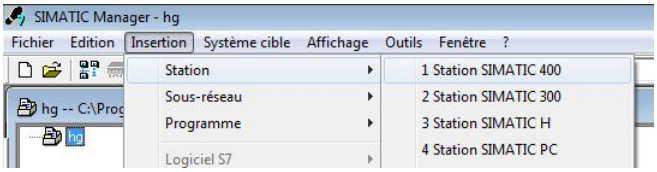


Choisir un nom de projet et valider :



b) Configuration Matérielle

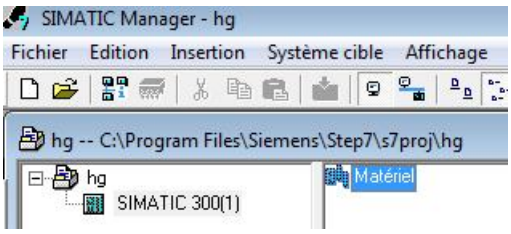
Insérer une « station SIMATIC 300 » :



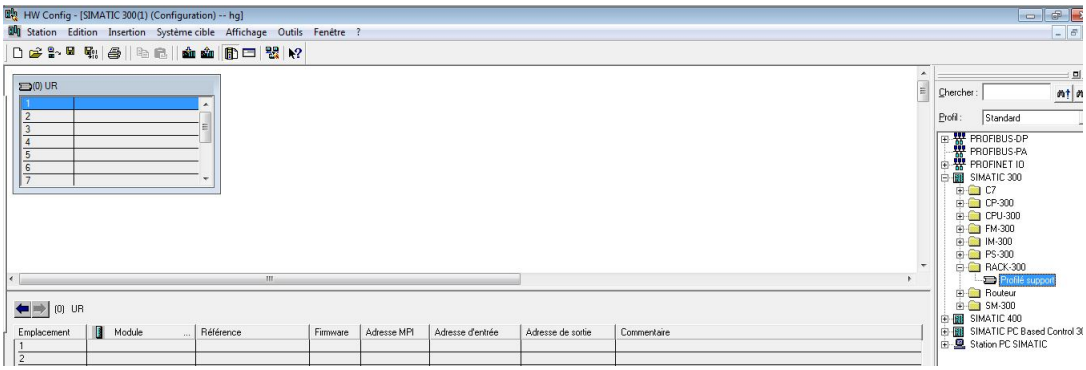
La « station » apparaît dans la partie droite de l'écran, double-cliquer sur cette icône :

Afin d'effectuer la configuration matérielle, double-cliquer sur l'icône « matériel » dans la partie droite

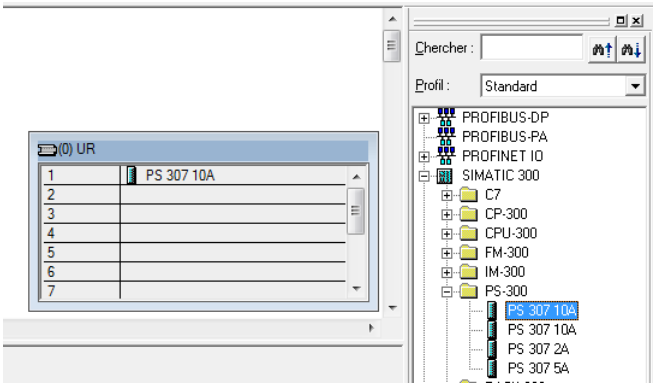
de l'écran :



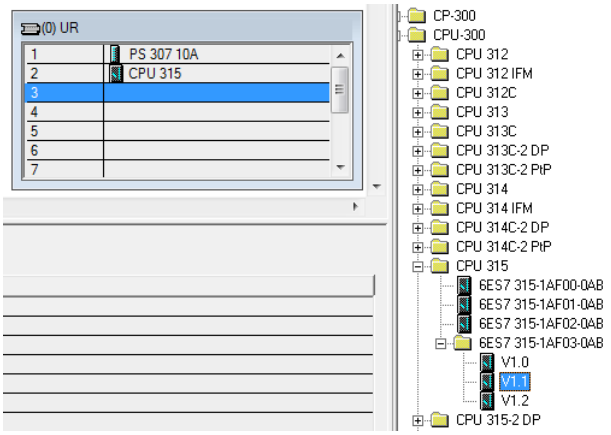
Développer « SIMATIC 300 » (cliquer sur le + devant SIMATIC 300), puis développer « RACK-300 » et double-cliquer sur « Profilé support ». Celui-ci apparaît dans la partie gauche de l'écran :



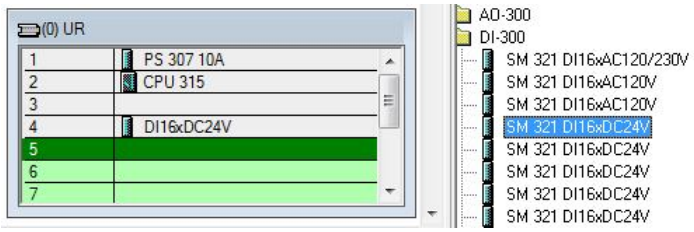
Réduire « RACK-300 » (cliquer sur le – devant « RACK-300 »), développer « PS-300 » et double-cliquer sur « PS 307 1A ». Le module alimentation de 10A se place sur le premier emplacement du rack



Réduire « PS-300 », développer « CPU-300 », puis « CPU 315 » et « 6ES7 315-1AF03 0AB0 », double-cliquer sur « V1.1 ». Le module Unité de Traitement se place sur le deuxième emplacement du rack :



Le module CPU occupe deux emplacements (2 et3). sélectionner l'emplacement 4, réduire « CPU-300 », développer « SM-300 » , puis « DI-300 », double-cliquer sur le troisième module « SM 321DI16xDC24V ».



Réduire « DI-300 », développer « DO-300 », double-cliquer sur le deuxième module « SM 322 DO16xDC24V/0.5A »

Fermer la fenêtre en cliquant en haut à droite et enregistrer les modifications. La configuration matérielle est terminée.

4-7-2 Creation D'un Programme En LADDER

En veut réaliser la fonction logique « **S= (a+b)*c »** en utilisant **SIMATIC** avec langage **LADDER.** pour cela**.**

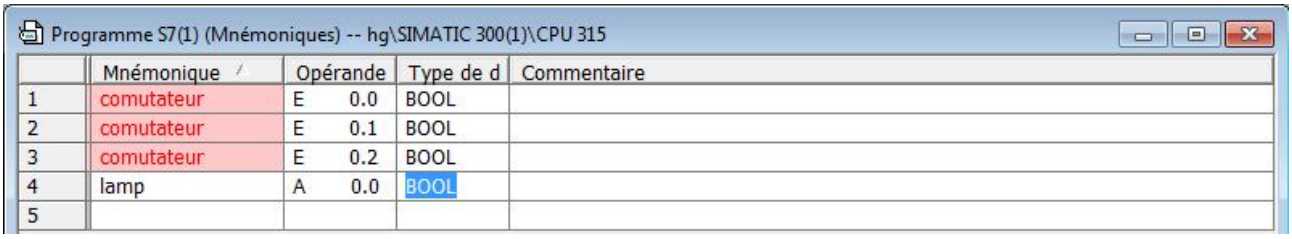
- suivre les même étapes que pour le TP1 (lancer **SIMATIC** , créer un projet, configurer le matérielle)

a) Créer une table de mnémoniques

- Développer « SIMATIC 300 », puis « CPU 315 » et « Programme S7 ». Double-cliquer sur l'icône « mnémoniques » qui apparaît dans la partie droite de l'écran:

Remplir la table :

La colonne « mnémonique » doit contenir une description simple de la variable. Indiquer dans la colonne « opérande » l'adresse de la variable (E pour entrée, A pour sortie…). Le logiciel donne le type de données dans la colonne suivante (ici « BOOL » pour booléen, c'est-à-dire binaire), mais celui-ci peut être modifié en cas de besoin.



	Mnémonique	Opérande	Type de d	Commentaire
1	comutateur	E 0.0	BOOL	
2	comutateur	E 0.1	BOOL	
3	comutateur	E 0.2	BOOL	
4	lamp	A 0.0	BOOL	
5				

Enregistrer la table et revenir dans « SIMATIC Manager ».

b) Programmer un réseau

- Développer « SIMATIC 300 », puis « CPU 315 » et « Programme S7 » et « blocs ». Double-cliquer sur l'icône « OB1 » qui apparaît dans la partie droite de l'écran:

Dans la fenêtre qui s'ouvre, vérifier que le langage de création est bien « CONT » et valider.

Créer le réseau dans la fenêtre qui vient de s'ouvrir. Sur les contacts, on peut indiquer l'adresse ou insérer un mnémonique :

c) Utilisation du simulateur d'automate :

Pour utilisateur le simulateur cliquez sur l'icône  Configurer le simulateur : insérer un bloc d'entrée, lui affecter le numéro 0, insérer un bloc de sortie, lui affecter le numéro 0. Passer en mode RUN

CHAPITRE 5 : INTRODUCTION AU RESEAU INDUSTRIEL

5-1 Architectures d'automatismes

5-1-1 Les automatismes centralisés

Les automatismes centralisés géraient tout un ensemble de fonctions qui n'avaient pas forcement d'interactions entre elles. Lorsqu'il y avait déjà un automate dans l'usine, les automaticiens qui devaient intégrer une fonctionsupplémentaire se posaient simplement la question : l'automate ou le système d'automatisme en placepeut-il gérer les E/S supplémentaires et quelle est la capacité de mémoire disponible ?

Ces automatismes centralisés amenaient des nombreuses contraintes :
• aucune autonomie des différents sous-ensembles,
• mise en service et maintenance lourdes et difficiles à effectuer du fait de la quantité d'E/S gérées,
• arrêt de l'ensemble des fonctions gérées par l'API en cas de défaut système de cet API ou d'arrêt pour
la maintenance du moindre élément de l'outil de production.

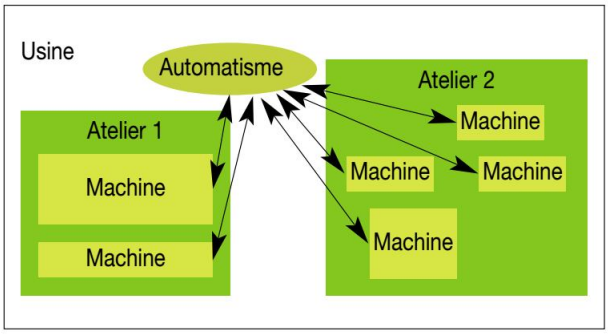


Figure 1. Les automatismes centralisés.

5-1-2 Les automatismes décentralisés

Du fait des contraintes imposées par les systèmes centralisés, les utilisateurs se sont orientés vers une segmentation de l'architecture. Celle-ci a été faite en découpant l'automatisme en entités fonctionnelles. Elle permet de simplifier les automatismes en réduisant le nombre d'E/S gérées et présente donc l'avantage de faciliter la mise en service et la maintenance. Cette segmentation a généré le besoin de communication entre les entités fonctionnelles. La fonction de communication est devenue la clef de voûte de la conception des architectures d'automatismes.

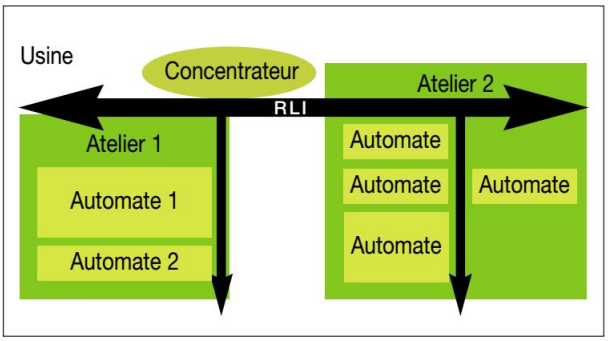
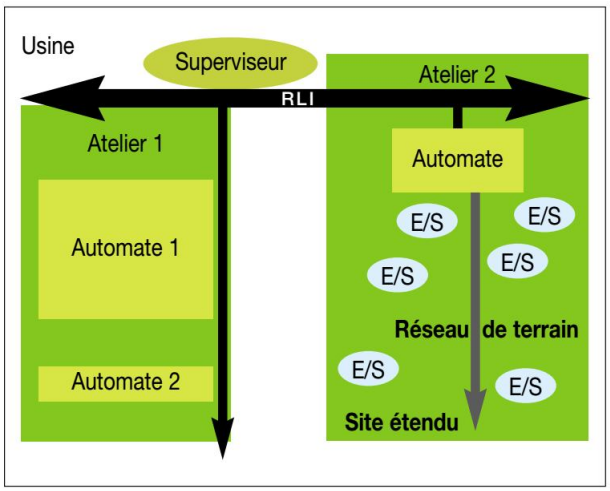


Figure 2. Les automatismes décentralisés.

Les constructeurs d'API ont donc créé des offres de réseaux locaux industriels (RLI) afin d'assurer une communication efficace entre les différents API.

5-1-3 La décentralisation des entrées/sorties et de la périphérie d'automatisme

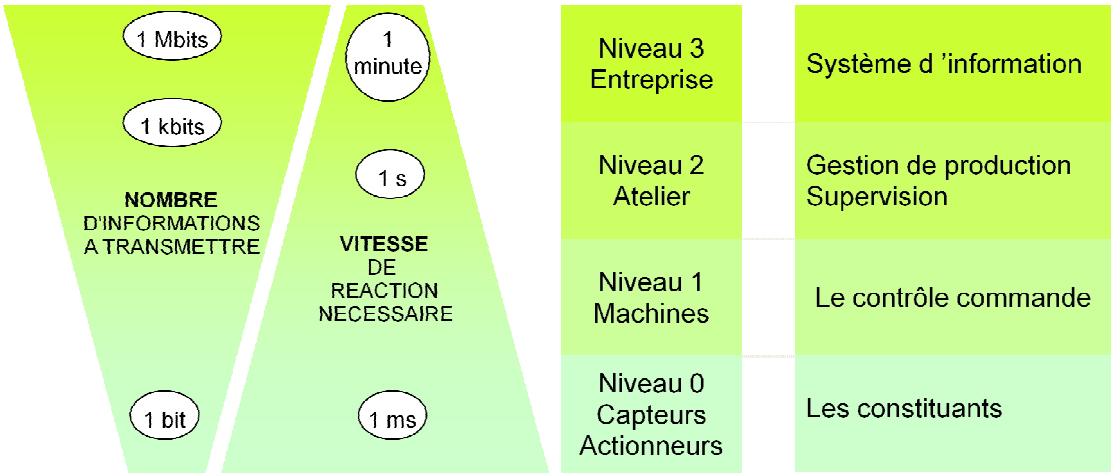
A la demande des utilisateurs finaux, notamment pour faire baisser les coûts de câblage, il a été nécessaire de prendre en compte la topologie des automatismes. Sur des sites plus étendus, il est souvent nécessaire de gérer un nombre de points diffus importants et de prendre en compte les fonctions métier réparties(variation de vitesse, dialogue homme/machine, pesage...). La réponse des constructeurs de produits d'automatismes est arrivée avec les réseaux et bus de terrain. Ceux-ci ont permis de gérer dans unpremier temps des E/S décentralisées puis la périphérie d'automatisme. Ces réseaux de terrain contribuent à réaliser des gains de câblage importants, mais surtout ils permettent de rendre accessibles des services (diagnostic,programmation…) sur tout le site.



5-2Le Concept CIM (Computer Integrated Manufacturing)

5-2-1 Pyramide (CIM)

Le CIM décrit les différents niveaux de communication sous une forme quantitative des données à véhiculer. Le niveau 0, niveau capteur/actionneur, nécessite un transfert performant (quelques millisecondes) mais concernant peu d'informations (données binaires), alors que le niveau 3 nécessite quant à lui de véhiculer de gros paquets de données, des fichiers et la performance n'est plus forcémentuncritère prédominant.

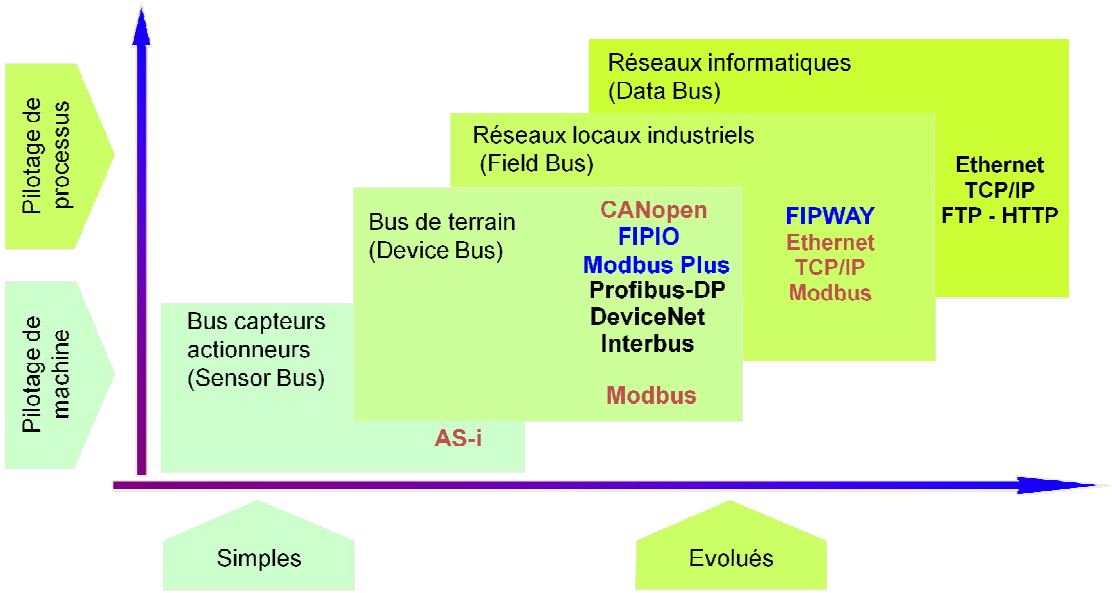


- ***le bus d'usine («data bus») : réseau local industriel (RLI)***.
  permet la communication entre l'automatisme et le monde informatique souvent basé sur Ethernet.
  - Temps non critique (1s à 10 s).
  - Beaucoup de données (fichiers, etc.).
  - Longues distances
- ***le bus de terrain («field bus» et «device bus») :***
  interconnexion des unités de traitement et des périphériques
  -Temps de réaction < 100 ms
  - Quantité de données relativement faible
  - Distance < 1km²
- ***le bus de bas niveau («sensor bus») : buscapteur/actionneur***
  - Temps de réaction < 10 ms
  - Quelques bits
  - Courtes distances < 100 m

5-2-2 Positionnement des principaux réseaux et bus

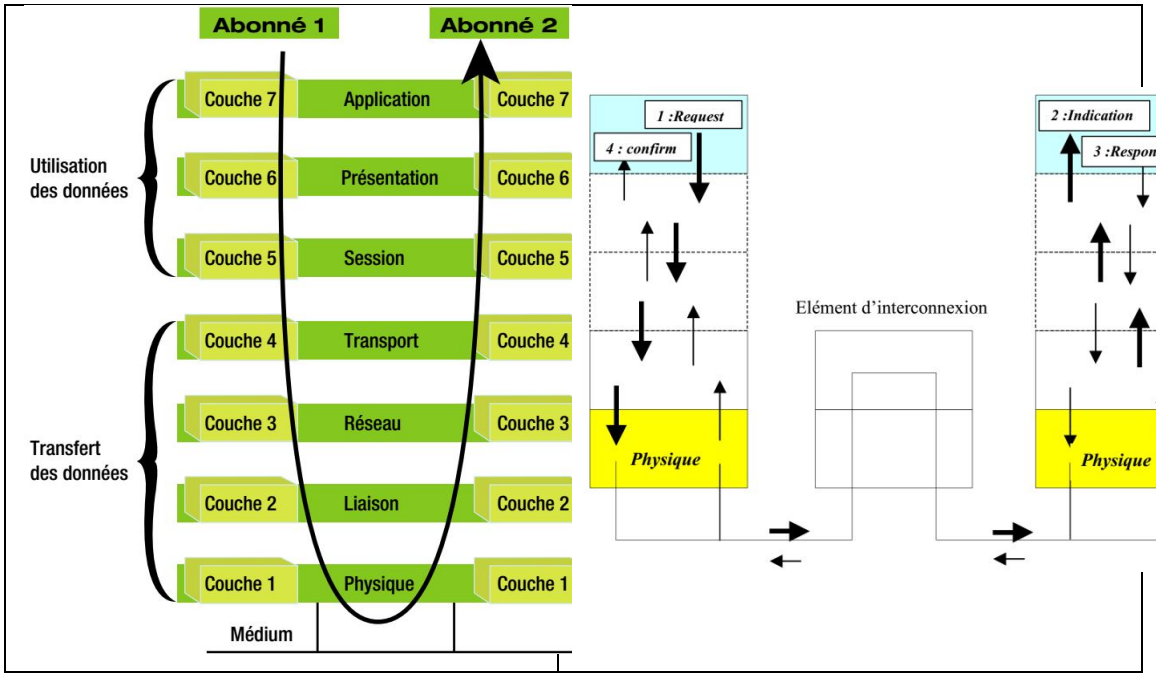
Les constructeurs d'automates programmables ont créé des réseaux et des bus adaptés au besoin. Ainsi à chaque niveau, correspond un bus ou un réseau :

- les "**sensor bus**", bus capteurs et actionneurs unitaires simples,
- les "**device bus**", bus et réseaux pour la périphérie d'automatisme : variateurs, robots, axes
- les "**field bus**", réseaux de communication entre unités de traitement : automates programmables, superviseurs, commandes numériques…
- **les réseaux locaux industriels**, pour l'établissement de la communication entre l'automatisme et le monde informatique.



5-3 Description du modèle OSI (Open SystemInterconnection)

C'est un modèle de communications entre ordinateurs proposé par l'ISO qui décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.



N	Couche ISO	Fonctionnement de la couche
7	**Application**	Elle est l'interface avec l'utilisateur, et fait parvenir les requêtes à la couche de présentation (HTTP)
6	**Présentation**	Elle définit la manière dont les données vont être représentées. Elle converti les données pour assurer leur interprétation par tous les systèmes.
5	**Session**	Elle assure les communications et les liaisons correctes entre les systèmes. Elle définit l'ouverture des sessions sur les équipements du réseau.
4	**Transport**	Elle permet d'établir une communication de bout en bout. Elle gère la segmentation et le ré-assemblage des données, le contrôle du flux ainsi que la détection d'erreur et la reprise sur erreur.
3	**Réseau**	Elle s'occupe de l'acheminement de paquets (datagrammes) à travers le réseau. (IP)
2	**Liaison**	Elle permet d'établir, à partir du support physique, une liaison exempte d'erreurs.
1	**Physique**	Elle définit les protocoles d'échange de bits et les aspects électriques, mécaniques et fonctionnels de l'accès au réseau.

5-4 Modèle OSI réduit pour les RLI

<table>
<tr><th>N</th><th>Couche OSI</th><th colspan="3">Rôle de la couche</th><th>Format de données</th></tr>
<tr><td>7</td><td>Application</td><td>Interface entre le réseau et l'utilisateur</td><td colspan="2">Services permettant de simplifier l'utilisation du réseau.</td><td>Message</td></tr>
<tr><td rowspan="6">2</td><td rowspan="6">Liaison</td><td rowspan="5">Transmission Méthodes d'accès au support</td><td rowspan="2">Aléatoires</td><td>Détection de Collision</td><td rowspan="6">Trame</td></tr>
<tr><td>Evitement de collision</td></tr>
<tr><td rowspan="3">Déterministes</td><td>Maître-esclave</td></tr>
<tr><td>Arbitre de bus</td></tr>
<tr><td>Anneau à jeton</td></tr>
<tr><td>Sécurisation des échanges</td><td colspan="2">En-tête de trame</td></tr>
<tr><td colspan="3">Bit de redondance</td></tr>
<tr><td rowspan="8">3</td><td rowspan="8">Physique</td><td>Codage de l'info.</td><td colspan="2">Tension</td><td rowspan="8">Bit</td></tr>
<tr><td rowspan="3">Topologie</td><td colspan="2">Bus</td></tr>
<tr><td colspan="2">Etoile</td></tr>
<tr><td colspan="2">Anneau</td></tr>
<tr><td rowspan="5">Support de transmission</td><td rowspan="2">Cuivre</td><td>Câble coaxial</td></tr>
<tr><td>Paire torsadée</td></tr>
<tr><td rowspan="2">Fibre optique</td><td>Monomode</td></tr>
<tr><td>Multimode</td></tr>
<tr><td>Autres</td><td>Courant porteur/Radio/Infrarouge</td></tr>
</table>

5-5Les Différentes Topologies

La topologie d'un réseau est caractérisée par le système de câblage du réseau ; c'est la partie physique du réseau.

- ***Point à point***

  C'est la forme la plus élémentaire qui implique deux machines. Elle correspondra souvent à la topologie d'une partie d'un réseau. En général, cela concerne une liaison série dont les vitesses sont vite limitées par la distance, l'utilisation de la fibre optique pouvant compenser cette faiblesse.

- ***Etoile***

  Une approche ancienne, non normalisée, correspondait à une organisation de machines esclaves reliées par liaisons point à point à une machine maître.

  - ✓ Facilite l'ajout de matériel ;
  - ✓ Facilite la localisation des défaillances
  - ✓ Branchement/débranchement à chaud ne pose pas de problème
  - ✓ concentrateur défectueux = réseau panne

- ***Bus***

  C'est la topologie la plus commune aux LAN (Local Access Network) car la plus économique. Chaque nœud est raccordé au bus commun.

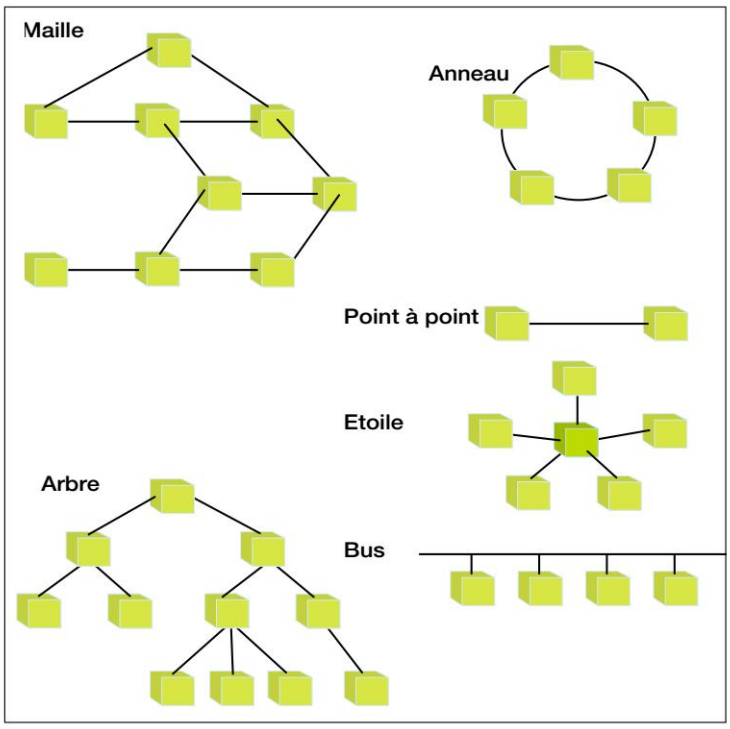
✓ Le réseau n'est pas perturbé lorsqu'unestation est défectueuse

✓ Coût d'installation de l'infrastructure peuélevé

✓ Nombre d'unités limités / la longueur dusupport

- ***Anneau***

  Chaque nœud est relié à ses voisins pour former une boucle fermée et a un rôle actif dans la propagation des échanges. Cette structure est bien adaptée aux LAN, en particulier si l'on recherche la disponibilité.
  Une rupture de liens entre deux nœuds peut être gérée pour garantir la communication. Chaque nœud a la possibilité de régénérer le signal et la structure se prête facilement à l'utilisation de la fibre optique, les distances de couverture pouvant être grandes.



5-6 Les méthodes d'accès

La méthode d'accès constitue la technique employée pour gérer le droit d'accès au média. Elle fait partie des attributions de la couche 2, et plus précisément de la sous-couche MAC.

On distingue deux classes de méthodes :

- **Méthodes statiques** : la bande passante du support est répartie une fois pour toute entre les stations.
- **Méthodes dynamiques** : le support n'est alloué qu'à la station qui veut parler, au moment où elle veut parler.

Dans le cadre des RLI, on utilise surtout les méthodes dynamiques suivantes : accès maître-esclave, accès aléatoire, accès par jeton.

5-6-1 Maitre esclave

Une station spéciale joue le rôle du maître. Les autres stations jouent le rôle des esclaves. Elles peuvent être orateur ou auditeur.

5-6-2 Accès aléatoire

Toutes les stations jouent le même rôle.

Avant d'émettre, une station écoute le réseau. Si aucune transmission n'est en cours,

elle émet son message, tout en l'écoutant. Si une collision intervient (une autre station émet simultanément), elle réitère sa tentative.

5-6-3 Accès par jeton

Toutes les stations jouent le même rôle et une autorisation d'émettre «le jeton» circule sur le réseau :

Avant d'émettre, une station attend un jeton libre. Lorsqu'il passe, elle y attache son message et le propage. S'il contient un message qui lui est destiné, elle en prend connaissance et propage le jeton.

5-7Équipement d'interconnexion de réseau

Les quatre périphériques principaux utilisés dans les réseaux locaux sont les suivants:

• Hubs • Bridges• Switches• Routers

Respectivement au modèle OSI, ces appareils fonctionnent aux couches suivantes:

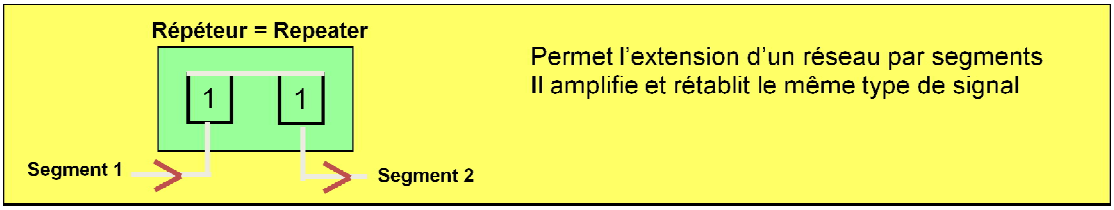
• La couche 1 physique: Hubs, repeaters (hubs are considered to be multiportrepeaters)

• La couche 2 liaison — Bridges, switches

• La couche 3 (Réseau)— Routers

5-7-1 Répéteur

Il s'agit d'un organe non intelligent, qui répète automatiquement les signaux qui lui arrivent et transitent d'un support vers un autre support. Dans le même temps, le répéteur régénère les signaux, ce qui permet de prolonger le support physique vers un nouveau support physique

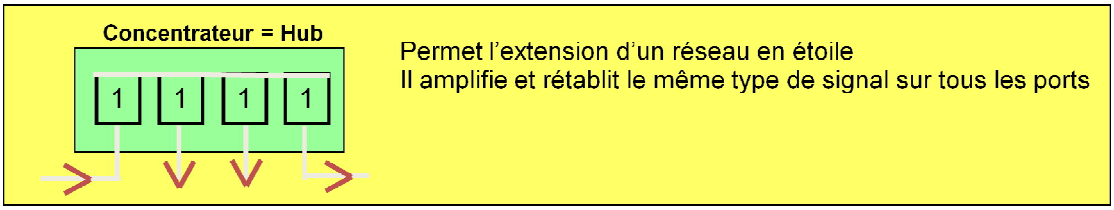


5-7-2 Hubs

Dans un réseau Ethernet ayant une topologie en arbre, un hub est un concentrateur

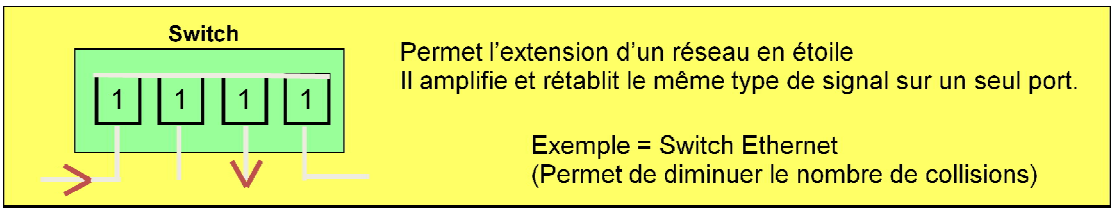
capable de récupérer le signal arrivant par une entrée et de le dupliquer vers

l'ensemble des ports de sortie. Le signal est généralement réamplifié car les données sont enregistrées dans des mémoires de type registre à décalage. Dans ce cas, les hubs sont dits actifs, c'est-à-dire qu'ils possèdent des éléments qui doivent être alimentés électriquement.



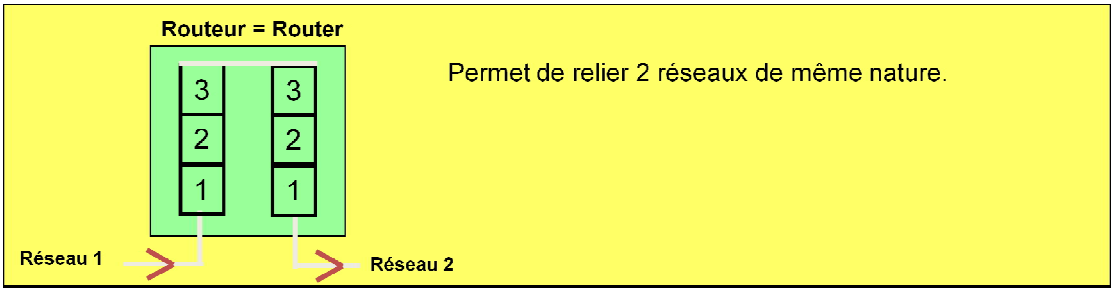
5-7-3 Switch

Un commutateur réseau (ou Network switch en anglais) est un pont multiport c'est-à-dire qu'il s'agit d'un élément actif agissant au niveau de la couche 2 du modèle OSI



5-7-4 Router

C'est un dispositif d'interconnexion de réseaux informatiques permettant d'assurer le routage des paquets entre deux réseaux ou plus afin de déterminer le chemin qu'un paquet de données va emprunter. Ils sont plus puissants : ils sont capables d'interconnecter plusieurs réseaux utilisant le même protocole



5-8 Le support physique de communication (le média)

Le support de communication fait partie de couche 1 du modèle OSI. L'utilisation des supports physiques dépend de la **distance entre les stations** et de l'**environnement** dans lequel sera installé le support On distingue :

- Paire torsadée (blindée ou non)
- Câble coaxial
- Câble électrique (courant porteur).
- Fibre optique
- Ondes radio
- Infra-rouges
- Laser

5- 8-1 Le câble coaxial

Le câble coaxial est un câble cylindrique composé de deux conducteurs électriques concentriques (différents et dits asymétrique).

- Peu coûteux, facilement manipulable
- Peut être utilisé sur de longues distances
- Débit jusque 10Mbit/s

Construction :

Gaine : protection du câble (caoutchouc, PVC ou téflon)

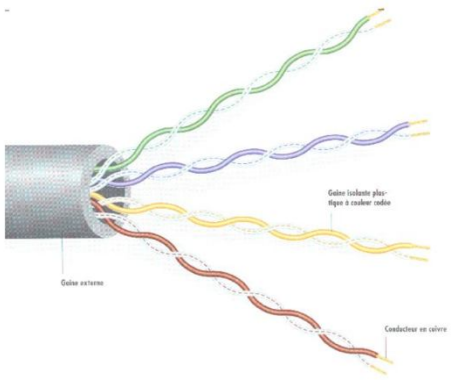
Blindage : partie métallique entourant le câble diminuant le bruit due aux parasites

Isolant : (diélectrique) évite le contact (court-circuit) entre l'âme et le blindage

Âme : brin de cuivre ou brins torsadés transportant les données

5-8-2 La paire torsadée

La paire torsadée est composé de deux fils conducteurs enroulés l'un autour de l'autre (plusieurs paires sont regroupées à l'intérieur d'un même câble).



- Franchissement de la limite des 10Mbits/s
- Plus de bande passante
- Possibilité de travailler en Full Duplex
- Plus d'interruption par coupure du câble
- Gestion plus aisée
- Permet d'avoir un câblage multi-usage (universel Téléphone; Fax Données...)

Inconvénient

- Plus de câbles qu'avec le coaxial
- Câblage plus cher et prend plus de place dans les gaines techniques

5-8-3 Le fibre optique

Utilisation

- Liaison entre répartiteur centraux téléphoniques urbains et interurbains
- Couplage de segments dans une ville, entre deux villes, entre les continents

Avantages

- Légèreté
- Immunité au bruit
- Isolation galvanique parfaite
- Faible atténuation
- Tolère des débits de l'ordre de 100Mbps
- Largeur de bande de quelques dizaines de mégahertz à plusieurs gigahertz (fibre monomode)
- Sécurité (difficile à mettre sur écoute)

Inconvénients

- Peu pratique dans des réseaux locaux (installation difficile)
- Coût relativement élevé
- Relative fragilité
- Distributeur central de la fibre optique

Une fibre optique est composée de 3 éléments principaux

- Le coeur dans lequel se propage les ondes optiques

- La gaine optique d'indice de réfraction inférieur à celui du coeur, qui confie les ondes optiques dans le coeur
- Le revêtement de protection qui assure la protection mécanique de la fibre
- Les fibres (appelées brins au sein d'un câble) sont regroupées dans des câbles par multiples de 2, de 8 ou de 12.

Références

1. M.pinot &al, Du Grafcet aux automates programmables, collection L.P édit Foucher. Paris 1986.
2. William Bolton. Les Automates Programmables Industriels. Edition Dunod 2010.
3. Siemens, SIMATIC. Configuration matérielle et communication dans STEP 7. Manuel Edition 03/2006.
4. Daniel DUPONT, David DUBOIS . Réalisation technologique du GRAFCET. Technique de l'ingénieure S 8 032 - 1 S 8 032 - 24 .
5. Khushdeep Goyal and Deepak Bhandari. Industrial Automation and Robotics. Katson
   Books. 2008.
6. Notes de laboratoire. Pierre Duysinx Geoffray Hutsemekers Henri Lecocq. automatisation et robotisation de la production. université de liège 2009/2010.