

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Mouloud MAMMERY de Tizi-Ouzou  
Faculté de Génie Electrique et d'Informatique  
Département d'Electrotechnique



# MEMOIRE

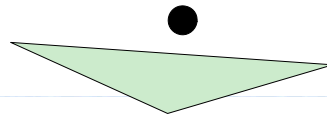
**De fin d'étude**

En vue de l'obtention du diplôme d'ingénieur d'état en Electrotechnique

**Option**

**MACHINES ELECTRIQUES**

**Thème**



**Automatisation d'une chaîne de  
transfert des armoires  
Frigorifiques à  
l'ENIEM**

Dirigé par :

Mr. MANSOURLR

Présenté par :

• BOUAZIZ Djamel

Promotion 2008

# **Remerciements**

*Je tien à exprimer ma reconnaissance et respectueuse gratitude envers mon promoteur Mr. R.MANSOURI et le Co-promoteur Mr A. BOUFARGUENE pour leur disponibilité, leur générosité et leur suivi très instructif tout au long de notre travail.*

*Je remercie également et vivement tous les membres du jury qui nous feront l'honneur d'évaluer notre modeste travail.*

*Mes remerciements vont aussi à l'égard de tous les enseignants qui ont contribué à ma formation.*

*En fin, je n'oublie pas de saluer tous ceux et celles de près ou de loin ont contribué à la réalisation de ce travail*

# **Dédicaces**

*Je dédie ce modeste travail à :*

- *Ma mère Aldja,*
- *Mon père Amar,*
- *Mon frère Chabane et mes sœurs (Samia, Fatiha, Saliha, Tassadit),*
- *Ma grande mère Sadia,*
- *Ma copine Sabrina,*
- *Toutes ma famille,*
- *Mon oncle Ahmed, ses fils et ses filles,*
- *Mon cousin, ami Rachid,*
- *Tous mes amis (es).*

## SOMMAIRE

Introduction générale.....	01
----------------------------	----

### CHAPITRE I : structure d'un système à automatisé

1 Généralités sur les systèmes automatisés.....	04
1.1. Définition du système automatisé.....	04
1.2 Automate programmable industriel.....	04
2 L'outil de modélisation GRAFCET.....	07
2-1. Historique.....	07
2.2 Définition.....	08
2.3 Cahier des charges.....	08
2.4 Constitution du <i>GRAFCET</i> .....	09
2.5 Règles d'évolution.....	13

### CHAPITRE II : les actionneurs et les capteurs

1. Pré actionneur.....	14
1.1 Distributeur pneumatique.....	14
1.2 Commandes manuelles auxiliaires.....	16
1.3 Contacteur.....	16
2. Actionneur.....	18
2.1 Actionneurs pneumatiques.....	18
2.2. Actionneurs électriques.....	22
3. Capteurs.....	27
3.1 Capteur de proximité.....	28
3.2 Capteurs de température.....	30
3.3 Capteur de vitesse.....	31
3.4 Capteurs de position.....	32
3.5 Capteur de pression.....	32

### Chapitre III : présentation du système à automatiser

1. Description et étude du fonctionnement du système.....	34
1.1 Acheminement des armoires.....	35
1.2 Transfert des armoires.....	36
1.3 Évacuation des armoires.....	38
2 Réalisation du GRAFCET de niveau II.....	41
2.1 Description littérale de la technologie du système.....	41
2.2 Mise en équation d'un GRAFCET.....	45
2.3 Liste des capteurs, pré capteurs et actionneurs.....	47
Conclusion.....	52

### Chapitre IV : présentation de l'API S7-300

I Choix d'un automate.....	53
----------------------------	----

2	Présentation de l'automate S7-300.....	53
2.1	Caractéristiques du S7-300.....	54
2.2	Construction de l'automate S7-300.....	54
2.3	Console de programmation (PG) ou PC avec logiciel STEP7.....	56
3	Fonctionnement de l'automate programmable.....	57
3.1	Réception des informations sur les états du système.....	57
3.2	Système d'exploitation.....	57
3.3	Exécution du programme utilisateur .....	57
3.4	Commande du processus.....	58
4	Programmation de l'automate S7-300.....	58
4.1	Création d'un projet STEP 7.....	59
4.2	Adressage des modules du S7-300.....	60
4.3	Traitement du programme par l'automate.....	61
5	Principe de conception d'une structure de programme.....	62
5.1	Rôle du système d'exploitation.....	62
5.2	Rôle du programme utilisateur.....	63
5.3	Blocs du programme utilisateur.....	63
5.4	Fonction (FC).....	64
6	Traitement du programme par la CPU.....	65
6.1	Linéaire .....	65
6.2	Structuré.....	66
6.3	Quelque blocs d'organisation des S7-300.....	66
	Conclusion.....	67

## **Chapitre V : programmation et simulation**

1	programmation du système.....	68
1.1	Affectation des entrées et des sorties.....	68
1.2	Elaboration du programme.....	70
2	Simulations.....	70
2.1	Présentation du S7-PLCSIM.....	70
2.2	Simulation du programme élaboré.....	73
	Conclusion.....	73
	<b>Conclusion générale.....</b>	<b>74</b>

Devant la compétitivité sans cesse croissante et de plus en plus « dure » demandée aux industries, qu'elles soient mécaniques, de transformation de produit ou de grande consommation, les industriels doivent garder leur outil de production performant et fiable. Dans ce domaine, l'automatisation tient une place très importante. Aujourd'hui, il serait difficile de concevoir un système de production sans avoir recours aux différentes technologies et composants que forment les systèmes automatisés de production (*SAP*).

L'introduction de l'informatique, en particulier, dans les processus de fabrication a considérablement accéléré le développement de l'automatisation, en assurant la sécurité et en augmentant la flexibilité des systèmes de production. En effet, avec l'avènement des *API* (Automates Programmables industriels), des machines à commande numérique sont apparues, elles peuvent accomplir plusieurs opérations d'usinage différentes. Les *API* sont utilisés à tous les stades du processus productifs (Conception, Production et contrôle).

Les automates SIEMENS qui sont disponibles sur le marché, de technologie moderne, jouissent d'atouts considérables (documentation disponible, programmation simple), ce qui permet d'assurer une formation rapide du personnel et une meilleure rentabilité de ces équipements.

C'est cette philosophie que l'entreprise nationale des industries électroménagères (*ENIEM*) compte appliquer, dans le but d'atteindre ses objectifs et d'assurer son existence sur le marché en procédant à l'automatisation de ses chaînes de production. [3]

*L'ENIEM* (Entreprise Nationale des Industries ElectroMénagères) est le leader de l'électroménager en Algérie. Elle est spécialisée dans la fabrication et le développement d'appareils électroménagers. Elle assure également la commercialisation et le service après vente de ses appareils. Elle est la première entreprise algérienne à être certifiée aux normes *ISO 9002* en 1998 et *ISO 9001/2000* en 2003 et renouvelée en 2006. [3]

*L'ENIEM* offre une gamme variée de produits électroménagers notamment :

- Réfrigérateurs petits et grands modèles ;
- Cuisinières ;
- Climatiseurs.

Elle offre également une gamme à usage professionnel :

- Réfrigérateurs vitrés ;
- Congélateurs ;
- Conservateurs ;
- Climatiseurs.

Elle offre aussi des produits pour équiper les administrations, entreprises et organismes publics.

*L'ENIEM* est une entreprise publique économique constituée le 2 janvier 1983 mais qui existe depuis 1974 sous tutelle de l'entreprise *SONELEC*. Son siège social se situe au chef lieu de la Wilaya de *Tizi-Ouzou*. Les unités de production : Froid, Cuisson, et Climatisation sont implantées à la zone industrielle *Aissat Idir* à *Oued-Aissi*, distante de 7 km du chef-lieu de wilaya. [4]

Depuis son implantation , *l'ENIEM* n'a cessé de chercher à automatiser ses différentes installations pour concurrencer les leader mondiaux de l'électroménager ; notamment dès ses débuts dans l'industrie, elle a signé des contrats de travail avec des firmes internationales en vu d'améliorer la qualité du produit et de former une classe travailleuse qualifiée, c'est en 1987 que *l'ENIEM* a débuté l'expérience des installations automatisées grâce à *TOSHIBA* après la mise en oeuvre de chaînes entièrement automatiques, ce qui lui a valu la marque *ISO 9002* en 1998 ; c'est dans la même année qu'elle s'est équipée d'une nouvelle installation automatique pilotée par un automate *SIEMENS*. [3]

C'est dans cet environnement de diversité technologique des différentes installations que nous avons élaboré ce modeste travail d'étude d'une automatisation, profitant ainsi de l'expérience des techniciens pour mieux comprendre les procédés industriels et notamment leurs conceptions.

Dans ce cadre, les responsables du département maintenance de l'unité froid de l'*ENIEM*, nous ont confié la tâche consistant à développer une solution programmable d'un équipement de transfert à base d'un *API S7-300* de *SIEMENS*.

Pour aboutir aux objectifs de notre travail, ce mémoire est consisté en cinq chapitres à savoir :

- Etude de la structure générale d'un système automatisé;
- Les différents types d'actionneurs et capteurs ;
- La description générale de fonctionnement du système à automatiser ;
- Etude de l'automate programmable *S7-300* de *SIEMENS* ;
- Elaboration du programme et la simulation via le simulateur de mode physique *S7-PLCSIM*.

Une conclusion générale termine notre modeste travail.

## Introduction

De nos jours, les entreprises sont soumises à de rudes épreuves de concurrence telle qu'il est impératif de proposer des produits de qualité en temps réduit, d'assurer la sécurité des installations et celle du personnel, entre autres contraintes auxquelles elles doivent faire face.

L'évolution remarquable en technologie cautionne l'évolution des méthodes de résolution des problèmes séquentiels, en utilisant moins de composants, une rapidité d'exécution en temps réel et un rendement optimal des appareils. Dans ce cadre, l'automate programmable industriel, en abrégé "API" est le constituant souhaité dans toute installation industrielle.

## 1 Généralités sur les systèmes automatisés

### 1.1 Définition du système automatisé

Un système automatisé est un ensemble d'éléments en interaction, organisés dans le but de produire des produits de qualité et ce pour un fiable coût avec minimum d'interventions humaines. [1]

L'Automatisation d'un procédé consiste à assurer un fonctionnement auparavant exécuté par des opérateurs humains ou, une logique câblée, dans un ensemble d'objets techniques appelé « partie commande ». Cette automatisation a pour but de répondre aux exigences quantitatives et qualitatives, pour que l'industrie puisse atteindre les objectifs suivants :

- Accroître la rentabilité et la compétitivité ;
- Améliorer la flexibilité de la production ;
- Améliorer les conditions du travail ;
- Augmenter la sécurité des ouvriers et biens à fabriquer.

### 1.2 Automate programmable industriel

L'Automate programmable industriel « API » est un dispositif électronique de contrôle de processus de production, il génère des ordres vers les pré-actionneurs de la partie opérative à partir des données d'entrées (capteurs, détecteurs, sondes...) et d'un programme

d'application (partie commande), il est généralement relié à un pupitre (terminal, opérateur de supervision) de commande. [2]

### 1.2.1 Structure générale des automates programmables industriels

L'automate programmable industriel comporte trois parties essentielles :

#### a- Partie opérative (PO)

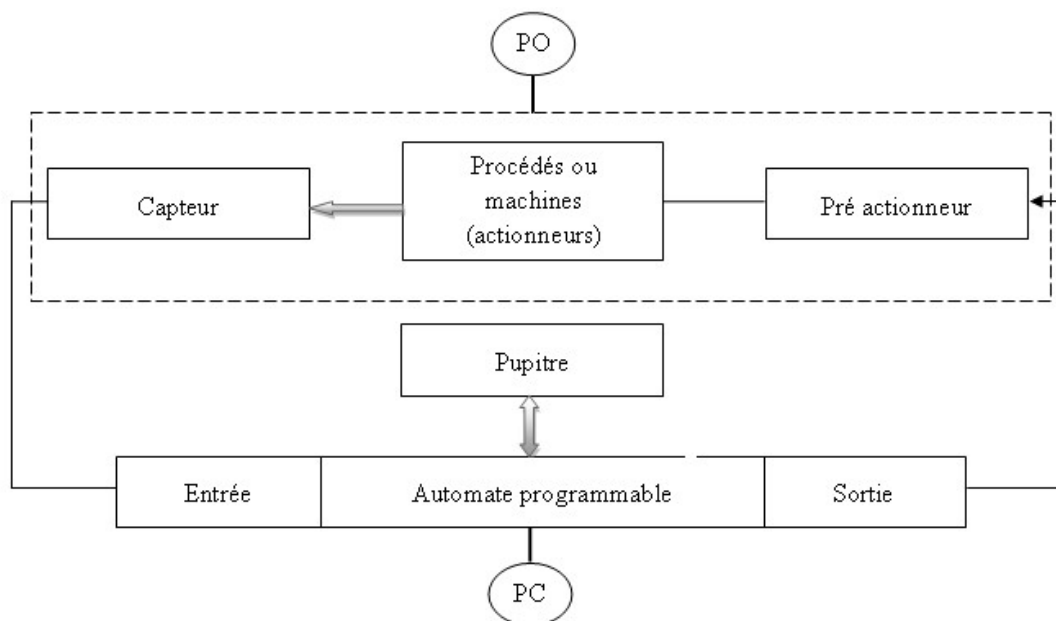
La partie opérative est l'ensemble des moyens matériels opérant physiquement sur les matières d'œuvres en vue d'assurer la production, elle est composée des actionneurs, des pré-actionneurs et des capteurs.

#### b- Partie commande (PC)

La partie commande est l'ensemble des moyens de traitement de l'information et de communication permettant le pilotage des parties opératives du système automatisé. Elle est construite à partir de la technologie électrique et électronique, et associe les techniques programmées (issues de l'informatique) et câblées.

#### c- Partie relation (pupitre de dialogue)

Il regroupe les différentes commandes (supervision) nécessaires au bon fonctionnement du procédé, et ce qui est appelé interface homme machine ou en abrégé IHM. [2]



**Figure I.1 :** Schéma fonctionnel d'une installation automatisée.

## 1.2.2 Architecture des Automates Programmables Industriels

L'automate programmable industriel à une structure externe et une structure interne.

### a- Architecture externe d'un API

Il existe deux types d'architectures externes d'API, les compacts (non modulaires) et les modulaires.

- **Les API compacts** : dans ce type d'API, la CPU, les modules d'entrées/sorties, alimentation, etc. sont intégrés dans un seul boîtier.
- **Les API modulaires** : ce type d'API se présente sous forme de modules indépendants enfichables sur un support appelé rack ou sur une extension et reliés entre eux par un bus. Ces API sont constitués d'une alimentation, d'une CPU et des modules d'entrées/sorties, à ceux-ci peuvent s'ajouter des modules de communication et des modules de fonction qui se chargeront des fonctions spéciales telles que la commande d'un moteur pas à pas, comptage rapide, régulation.

### b- Architecture interne d'un API

Cette structure comporte les parties principales suivantes :

- L'unité centrale ;
- Les modules d'entrées/sorties ;
- Le module d'alimentation ;
- Les modules de communication ;
- Les coupleurs.

Ces parties sont reliées entre elles par des bus (ensemble de fils autorisant le passage des informations entre les secteurs de l'automate). Ces parties sont représentées dans le schéma synoptique de la figure I.2.

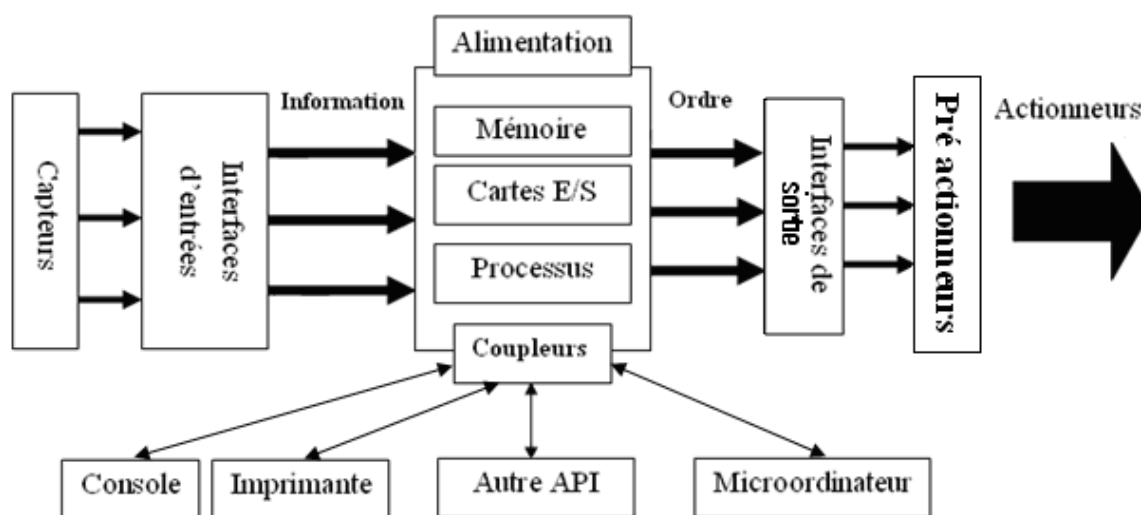


Figure I.2 : Architecture d'un automate programmable industriel.

## 2 L'outil de modélisation GRAFCET

### Introduction

En première étape de toute tâche d'automatisation d'un système on doit définir un cahier des charges ; le langage courant ne permet pas de lever toutes les ambiguïtés, alors il faut un moyen de communication satisfaisant la compréhension du constructeur et son client, d'où la nécessité d'une représentation graphique telle que le *GRAFCET*.

### 2.1 Historique

Le *GRAFCET* est le résultat du travail bénévole d'une commission réunissant, l'*AFCET* (Association Française pour la Cybernétique Économique et Technique), l'*ADEPA* (Agence pour le Développement de la Productique Appliquée à l'industrie) des industriels et des universitaires. Cette commission, créée le 26 juin 1975, a défini les bases du *GRACET* dans son rapport final achevé en avril 1977. Le *GRAFCET* a été conçu comme un système unifié d'expressions qui n'est pas la propriété de personne. Dès 1978 le *GRACET* fait son entrée dans l'éducation nationale. Il est maintenant le pilier des programmes d'automatismes industriels. D'autres outils complémentaires du *GRACET* ont été créés, tels le Guide d'Étude des Modes Marche et d'Arrêt (*GEMMA*1981), les technoguides et les chaînes fonctionnelles.

Depuis 1988, le *GRAFCET* est devenu un outil de description normalisé (norme *CEI 848*) qui fonctionne en logique séquentielle. C'est un outil simple mais extrêmement puissant

qui permet les représentations fonctionnelles opérationnelles et technologique de la plupart des automatismes industriels.

En 1985, *SIEMENS* (leader européen des automatismes) adopte le *GRAF CET* et le promeut en Allemagne. En 1986 *ALLEN BRADLEY* (leader mondial des automates programmables) adopte et développe le *GRAF CET*, y compris pour le marché Américain.

## 2.2 Définition

Un *GRAF CET* (GRAPhe Fonctionnel de Commande Etapes-Transitions), est une méthode de conception et modélisation de commandes séquentielles des systèmes à partir d'un cahier des charges. Elle est basée sur la représentation graphique très détaillée du système avant de faire une quelconque synthèse de l'automatisation. Le développement du *GRAF CET* est étroitement lié à l'évolution du processus de l'automatisation.

## 2.3 Cahier des charges

Un cahier des charges décrit tous les échanges à réaliser entre le processus à automatiser et la partie commande ainsi que les conditions d'utilisation et de fonctionnement de l'automatisme en exploitation du processus. L'élaboration du cahier des charges se fait en deux niveaux.

### 2.3.1 *GRAF CET* de niveau 1 (spécifications fonctionnelles)

Le *GRAF CET* de niveau 1 est la description littérale et la représentation du comportement de la partie commande vis-à-vis de la partie opérative et du monde extérieur. A ce niveau on décrit uniquement, les événements et leurs chronologies, mais pas de technologie (actionneurs, capteurs...).

### 2.3.2 *GRAF CET* de niveau 2 (spécifications technologiques)

Les spécifications technologiques opérationnelles précisent la manière avec laquelle l'automatisme devra s'insérer physiquement, dans l'ensemble que constitue le système automatisé. C'est dans cette étape que doivent intervenir les renseignements, sur la nature exacte des capteurs et des actionneurs.

## 2.4 Constitution du *GRAFCET*

La description du fonctionnement d'un automatisme logique peut être représenté graphiquement par un ensemble :

- D'étapes auxquelles sont associées des actions ;
- De transitions auxquelles sont associées des réceptivités ;
- De liaisons (ou arcs) orientées.

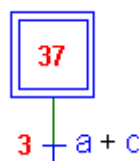
### 2.4.1 Étapes

L'étape correspond à une situation élémentaire ayant un comportement généralement stable. En principe, pendant une étape, les organes de commande ne changent pas d'état. Une étape est soit active soit inactive. Elle se représente par un carré repéré par une variable alphanumérique placée en son centre. Lorsqu'une étape est active, on peut le préciser par un point.

Sur un *GRAFCET* on peut rencontrer différents types d'étapes : étape initiale, macro étape, étape encapsulante, la tache, etc.

#### a- Étape initiale

Certaines étapes sont actives au début de fonctionnement du système automatisé. Ce sont les étapes initiales. Elles se représentent par un double carré.



**Figure I.3 :** Etape initiale

#### b- Macro étape

Proprement parler une macro-étape n'est pas une étape. C'est une représentation unique d'une succession d'étapes et de transitions. On parlera alors d'expansion de macro-étape. Une macro-étape est assimilable, par son fonctionnement, à un déroutement de programme sur interruption. Dans un *GRAFCET*, une macro-étape est unique. On ne pourra l'appeler qu'une seule fois. Il peut y avoir plusieurs macro-étapes dans un *GRAFCET*.

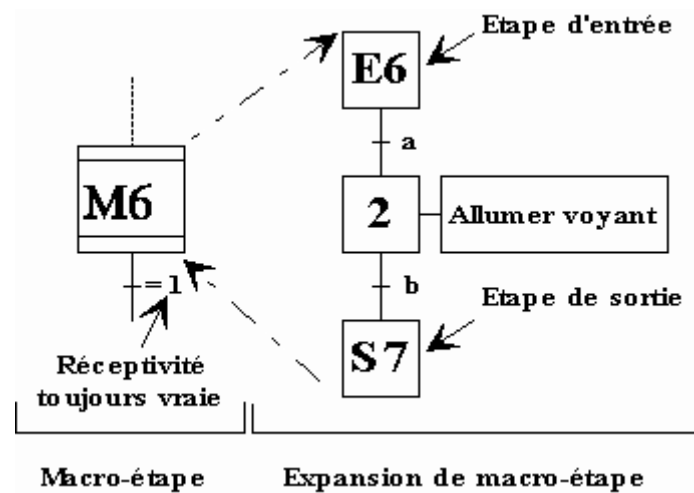


Figure I.4 : Représentation d'une macro-étape

#### d- Tâche

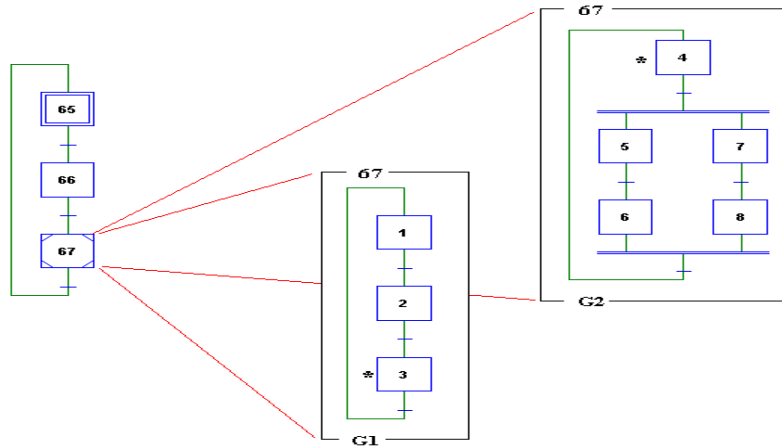
Cette étape se différencie de la macro-étape par sa non unicité. Une tâche est un sous-programme et peut être appelée à plusieurs reprises dans un même *GRAF CET*. La tâche fait appel à des notions d'antériorité gérées par une matrice d'antériorité un peu complexe à manipuler. A la place de la tâche nous utiliserons des *GRAF CET* hiérarchisés plus souples à l'emploi.

#### e- Étapes encapsulantes

Il y a encapsulation d'un ensemble d'étapes, dites encapsulées par une étape, dite encapsulante, si et seulement si lorsque cette étape encapsulante est active, l'une au moins, des étapes encapsulées est active. Le spécificateur peut utiliser l'encapsulation pour structurer de manière hiérarchique un *GRACET*.

Dans l'exemple ci-dessous, l'étape encapsulante 67 possède 2 encapsulations. Ces deux encapsulations sont les *GRAF CET* partiels G1 et G2. L'activation de l'étape 67 provoque l'activation des étapes X3:G1 et X4:G2. La désactivation de l'étape 67 provoque la désactivation de toutes les étapes des *GRAF CET* partiels G1 et G2. On repère une encapsulation par un *GRAF CET* partiel entouré d'un cadre où on place en haut, le nom de l'étape encapsulante (dans notre exemple : 67), en bas le nom du *GRAF CET* partiel (dans

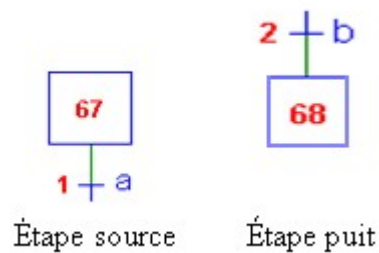
notre exemple : G1 et G2). Dans notre exemple on constate qu'il n'y a pas d'étape initiale pour les *GRAF CET* partiels.



**Figure I.5 :** Représentation d'étapes encapsulantes.

Il existe aussi des étapes sources et des étapes puits. Une étape est appelée "étape source" si elle n'est pas reliée en amont à une transition. De la même façon, une étape est appelée "étape puits" si elle n'est pas reliée en aval à une transition.

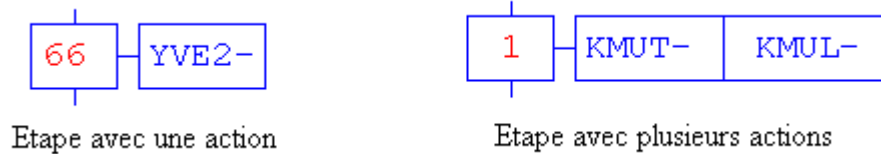
- **Exemples :**



## 2.4.2 Actions associées à l'étape

### a- Actions simples

On précise pour chaque étape, à l'intérieur d'un rectangle l'action ou les actions à effectuer lorsque l'étape est active.



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

### b- Actions conditionnelles

Certaines actions associées à des étapes peuvent être conditionnées par des contraintes externes ou internes.

L'ordre émis peut être :

- Soumis à un retard, noté D (Delay) ;
- D'une durée limitée, noté L (Limited) ;
- Fugitif, noté P (Ponctual) ;
- Maintenu actif, noté S (Set) ;
- Désactivé, noté R (Reset).

### 2.4.3 Transitions

Une transition indique la possibilité d'évolution d'une étape à l'étape suivante. A chaque transition on associe une ou des conditions logiques (booléennes) qui traduisent la notion de réceptivité. La réceptivité est une fonction combinatoire d'informations booléennes telles que :

- L'état des capteurs ;
- Une impulsion sur un bouton poussoir ;
- Action d'un temporisateur, d'un compteur ;
- Etat actif ou inactif d'autres étapes, etc.

### 2.4.4 Liaisons orientées

Les liaisons indiquent les voies d'évolution du *GRAFCET*. Dans le cas général, les liaisons qui se font du haut vers le bas ne comportent pas de flèche. Dans les autres cas, on peut utiliser des flèches pour préciser l'évolution du *GRAFCET* pour éviter la confusion.

## 2.5 Règles d'évolution

Le *GRACET* fonctionne suivant cinq règles d'évolution :

**Première règle :** l'initialisation précise l'étape ou les étapes actives au début du fonctionnement. On la repère en doublant les côtés des symboles correspondants. Il peut y avoir plusieurs étapes initiales dans un *GRAF CET*.

**Remarque :** les étapes initiales sont activées inconditionnellement au début de cycle.

**Deuxième règle :** une transition est soit validée, soit non validée. Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives. Elle ne peut être franchie que lorsqu'elle est validée et que la réceptivité associée à la transition est vraie.

**Troisième règle :** le franchissement d'une transition entraîne l'activation simultanée de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

**Quatrième règle :** plusieurs transitions simultanément franchissables sont simultanément franchies (hors programme).

**Cinquième règle :** si au cours du fonctionnement, une même étape doit être activée et désactivée simultanément, elle reste active (hors programme). [1]

**Remarque :** la durée de franchissement d'une transition ne peut jamais être rigoureusement nulle, même si d'après les troisième et quatrième règles, elle peut être rendue aussi petite que possible. Il en est de même de la durée d'activation d'une étape. La durée d'activation minimale d'une étape est équivalente à un temps de cycle d'automate ( $t_c$ ), soit  $t_c \ll 10 \text{ ms}$  ». La cinquième règle se rencontre très rarement.

## Introduction

Dans le premier chapitre nous avons abordé la structure générale d'un système automatisé. On a dit que se dernier est composé d'une partie opérative, d'une partie commande et d'un pupitre. L'objet de ce chapitre est de décrire les principaux organes qu'on rencontre souvent d'une partie opérative d'un système automatisé.

### 1 Pré actionneur

C'est un composant de gestion d'énergie d'alimentation des actionneurs, il reçoit l'information de commande de la partie commande, puis il délivre en sortie la puissance nécessaire pour l'actionneur.

#### 1.1 Distributeur pneumatique

Le distributeur pneumatique est un dispositif commandé électriquement permettant d'autoriser ou d'interrompre par une action mécanique, la circulation d'un fluide ou d'un gaz dans un circuit. On les retrouve sur les groupes froids, les climatiseurs, les compresseurs, les lave-linge...[4]

Le distributeur pneumatique a pour fonction essentielle de distribuer le fluide (ici l'air) dans les canalisations qui aboutissent aux chambres du vérin.

##### 1.1.1 Constitution

Un distributeur classique comprend 3 éléments principaux comme l'indique la figure suivante : [4]

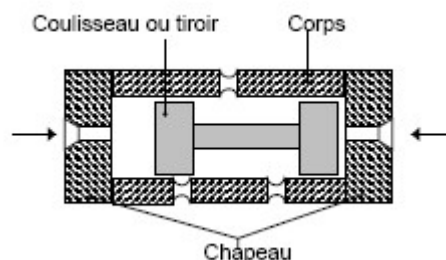


Figure II.1 : Constitution d'un distributeur.

##### 1.1.2 Les principaux distributeurs pneumatiques

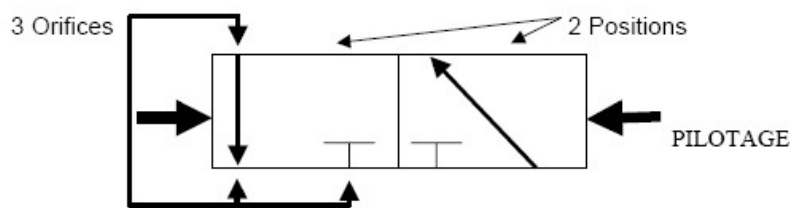
Un distributeur est caractérisé :

- **Par son nombre d'orifices** : c'est-à-dire le nombre de liaisons qu'il peut avoir avec son environnement (pression, sortie(s), échappement),
- **Par son nombre de positions** : qui exprime le nombre de cas d'interconnexion qu'il peut assurer par rapport à son environnement.

Ces deux caractéristiques donnent le nom au distributeur. La représentation des distributeurs dépend de son nom. Chaque position est symbolisée par un carré dans lequel figurent les interconnexions réalisées entre les liaisons.

Pour décoder le fonctionnement, il faut considérer que les liaisons restent fixes et que c'est l'ensemble du carré de position qui se déplace. Ce déplacement est obtenu par le *pilotage* du distributeur.

Un distributeur 3/2 a donc 3 orifices dans un carré, et 2 positions (donc 2 carrés) comme l'indique la figure suivante :



**Figure II.2** : Représentation d'un distributeur 3/2.

Il existe deux types de pilotage électrique suivant le type de distributeur :

- **Le distributeur bistable (2 états stables)** : Il garde sa position en l'absence de signal de pilotage, ses solutions constructives l'ont doté d'une fonction mémoire. On dit qu'il est à *double pilotage*.
- **Le distributeur monostable (1 état stable)** : il est rappelé dès la disparition du signal de pilotage par un ressort, ou par une pression permanente. On dit qu'il est à *simple pilotage*.

## 1.2 Commandes manuelles auxiliaires

Pour faciliter le réglage et le dépannage hors tension, les électrovannes sont munies de commandes manuelles auxiliaires (C M A). Un bouton poussoir de petite dimension permet d'agir directement sur le clapet.

Cette possibilité de manœuvre est critiquable sur le plan sécurité. Toute action inopportune sur l'un des poussoirs, pendant le fonctionnement normal du système automatisé de production, peut entraîner un comportement dangereux de l'installation. Vu la conception actuelle des électrovannes, le seul recours est de les regrouper dans une même armoire à fermeture par clé. L'accès aux CMA ne doit être autorisé qu'aux techniciens de maintenance ou à ceux chargés de la mise au point.

Cependant, dans une installation pneumatique, la commande manuelle auxiliaire associée au pilotage pneumatique et indirecte reste opérationnelle en permanence. Le problème de sécurité qui se pose est de même ordre que pour les électrovannes et appelle le même genre de solution : la mise en armoire verrouillée des distributeurs à CMA. Toutefois, compte tenu de la dimension souvent très réduite des poussoirs, L'utilisateur ne peut, en fait, s'en servir que pour tester la position du tiroir dans le corps du distributeur et non pour déplacer le tiroir.

## 1.3 Contacteur

Le contacteur est un appareil de commande capable d'établir ou d'interrompre le passage de l'énergie électrique entre le réseau d'alimentation et le récepteur (c'est sa fonction). Il a donc un pouvoir de coupure non nul. Ce dernier peut être commandé à distance au moyen de contacts actionnés manuellement (bouton poussoir) ou automatiquement (asservi à une grandeur physique : pression, température, vitesse, etc.). Il est constitué des éléments suivants :

**Le circuit principal :** Il est constitué de l'ensemble des pièces conductrices du contacteur.

Cela comprend en premier les pôles principaux, puis les contacts principaux qui sont insérés dans le circuit principal du contacteur et ce sont eux qui supporte, établissent ou coupent le courant sur l'ensemble du circuit électrique dans lequel s'insère le contacteur.

Il est à noter que ce dispositif contient un système de soufflage de l'arc électrique. C'est grâce à ce dernier qu'il est possible de couper le courant sans causer de dommage au pôle au moment précis où l'arc électrique s'amorce quand on procède à la coupure.

**Le circuit de commande :** ce dernier est constitué de toutes les autres pièces conductrices autres que celles du circuit principal.

Ce circuit a pour fonction de commander l'ouverture ou la fermeture du circuit principal du contacteur.

Il est à noter que ce circuit est alimenté par son propre circuit électrique qui est en tension faible (environ 13V). Son élément principal en est le contact de commande que le contacteur manoeuvre, quand à lui, mécaniquement.

Ce circuit n'a pas de protection particulière puisque étant alimenté en faible tension il n'a pas de risques électrique.

**Le circuit auxiliaire :** ces fonctions sont principalement le verrouillage, la signalisation, l'auto-alimentation et l'asservissement du contacteur en général. Il comporte des contacts auxiliaires manoeuvrés eux aussi mécaniquement par le contacteur. Ces derniers peuvent être temporisés ou instantanés. Ils existent en deux types :

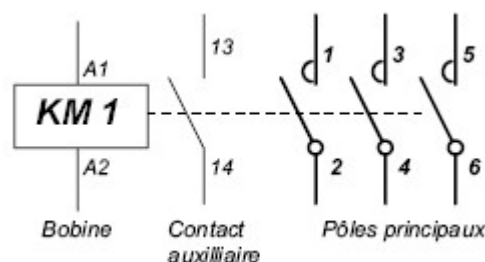
- Contacts à fermeture
- Contacts à ouverture

L'intensité de fonctionnement de ces contacts est assez variable puisqu'elle concerne une plage de 5 à 10 ampères. Par comparaison, les contacts principaux supportent 200 à 3000 ampères.

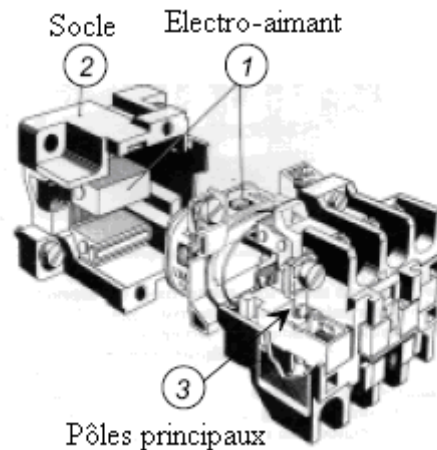
**L'organe moteur :** ce dernier est électromagnétique et est constitué d'un électro-aimant. Il contient :

- Un circuit magnétique déformable
- Une bobine d'attraction

L'aspect de l'électro-aimant sera conditionné par le fait qu'il soit à courant continu ou alternatif.



**Figure II.3 :** Schéma d'un contacteur tripolaire.



**Figure II.4 :** Vue générale d'un contacteur.

## 2. Actionneur

L'actionneur est un organe qui reçoit de l'énergie électrique, pneumatique ou hydraulique venant d'une source extérieure pour la transformer en une action physique, dans le but d'agir sur la matière d'œuvre et obtenir une valeur ajoutée.

### 2.1 Actionneurs pneumatiques

Avec l'air comprimé, on dispose d'une énergie potentielle exploitable sous forme statique ou sous forme dynamique par transformation en énergie cinétique. [4]

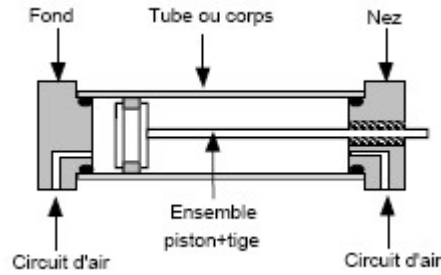
En faisant agir l'air comprimé sur une face immobile, on obtient une force statique  $F_S$  proportionnelle à la pression  $P$  et à surface  $S$ , exprimée par la relation :

$$F_s = P \cdot S$$

Parmi les actionneurs pneumatiques on peut citer :

#### 2.1.1 Vérins pneumatiques

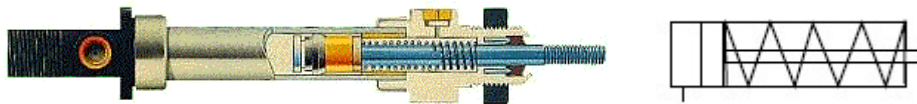
Les vérins pneumatiques (figure II.5) sont des actionneurs qui permettent de transformer l'énergie de l'air comprimé en un travail mécanique. On distingue deux types :



**Figure II.5 :** Constitution d'un vérin pneumatique.

### *a- Vérins simple effet*

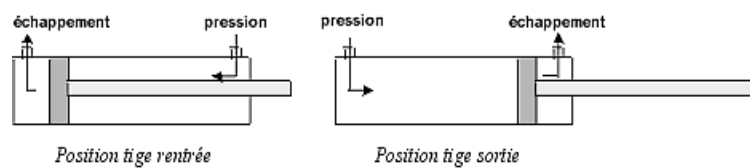
Ce sont des vérins qui effectuent un travail dans un seul sens. Le retour à la position initiale s'effectue par un ressort.



**Figure II.6 :** vérin simple effet.

### *b- Vérins double effet*

Les vérins double effet ont deux alimentations possibles : soit par la chambre arrière, soit par la chambre avant.



**Figure II.7 :** vérin double effet.

Lors de l'alimentation en pression de la chambre arrière le piston se déplace vers l'avant, celui-ci pousse l'air de la chambre avant et lors de l'alimentation de la chambre avant le piston se déplace vers l'arrière, celui-ci pousse l'air de la chambre arrière.

L'air de la chambre à l'échappement doit pouvoir être évacué afin de ne pas s'opposer au déplacement du piston.

Dans un vérin double effet les chambres se trouvent donc alternativement mises à la pression et à l'échappement.

### 2.1.2 Choix fonctionnel d'un distributeur par rapport au vérin

Ce qui change dans les distributeurs présentés avant c'est le nombre de sorties. Donc pour choisir le distributeur on doit connaître le vérin associé.

#### a- Distributeur pour vérin simple effet

Un vérin simple effet ne comporte qu'un seul orifice à alimenter, le distributeur utilisé ne comporte qu'un seul orifice de sortie. D'où l'utilisation d'un *distributeur 3/2* à trois orifices (pression, sortie, échappement) et deux positions. [4]

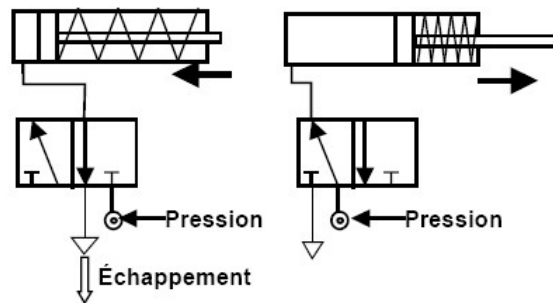


Figure II.8 : Distributeur 3/2 associé à un vérin simple effet.

#### b- Distributeur pour vérin double effet

Un vérin double effet comporte deux orifices sur lesquels il faut alterner les états de pressions et d'échappement, le distributeur utilisé doit comporter deux orifices de sorties.

Deux possibilités sont offertes :

- *Un distributeur 4/2* à quatre orifices (pression, sortie 1, sortie 2, échappement) et deux positions selon la figure ci-dessous.

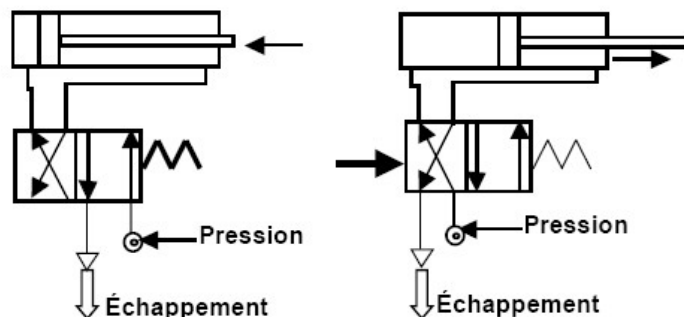
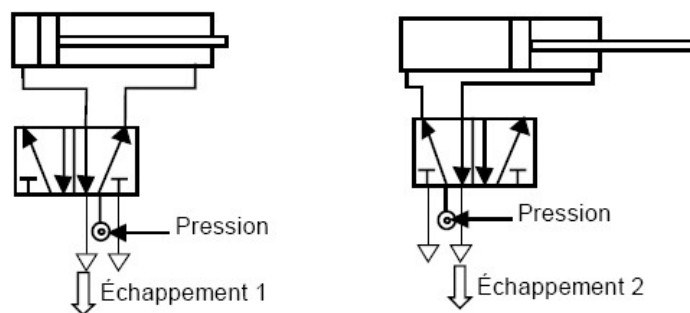


Figure II.9 : Distributeur 4/2 associé à un vérin double effet.

- **Un distributeur 5/2** à cinq orifices (pression, sortie 1, sortie 2, échappement 1, échappement 2) et deux positions.

Cette dernière option qui offre un échappement différent par chambre du vérin permet une régulation différente de la vitesse des deux mouvements de la tige. La figure ci après représente un distributeur 5/2 associé à un vérin double effet.



**Figure II.10** : Distributeur 5/2 associé à un vérin double effet.

### 2.1.3 Les ventouses

Ce sont des éléments de préhension souples destinés à être utilisés avec un générateur de vide. De matière, de forme et de diamètre différents elles permettent de répondre pratiquement à tous les cas d'applications de manutention.

On les retrouve dans les industries de l'agro-alimentaire, du verre, du bois ; la prise de pièces en sortie de presse à injecter, l'imprimerie pour le brochage et le retournement de feuilles, le conditionnement de produits et le chargement de petits transferts rotatifs...

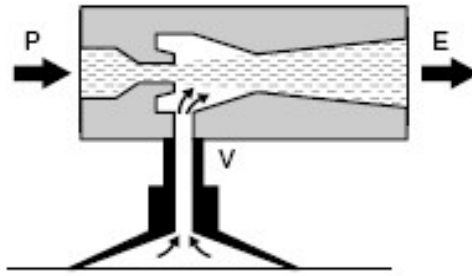
#### **a- Générateur de vide à effet Venturi**

La technique du vide prend une place très importante dans les domaines de la préhension et de la manutention. Parmi les différents moyens de créer une dépression, les générateurs de vide à effet Venturi présentent de nombreux avantages : technique simple et compétitive, pas d'usure (aucune pièce en mouvement), faible encombrement, compact, grande légèreté permettant d'être montés directement sur les systèmes embarqués tel qu'en robotisation.

#### **b- Principe de fonctionnement**

En traversant la buse d'éjection du venturi, l'alimentation d'air comprimé (P) provoque une aspiration (V) et crée ainsi une dépression dans le circuit de préhension (– 920 mbar à

partir de 5 bar) figure (II.11). L'air est évacué à travers d'un silencieux d'échappement placé en (E).



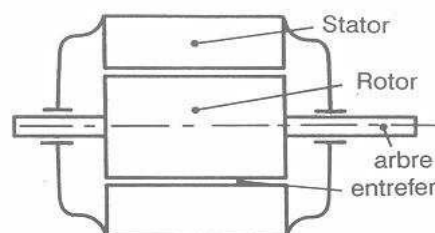
**Figure II.11** : Principe de fonctionnement d'un générateur de vide à effet Venturi.

## 2.2 Actionneurs électriques

### 2.2.1 Présentation des machines

Les machines électriques sont constituées d'un stator et d'un rotor en matériaux ferromagnétiques. Entre ces deux parties se trouve l'entrefer qui est maintenu par des paliers ou roulements (figure II.12). [4]

**Remarque** : Les parties du moteur qui sont soumises à des champs variables doivent être feuilletées pour diminuer les pertes par courant de Foucault.



**Figure II.12** : Constitution d'une machine électrique.

Il existe un nombre important et varié de type de moteurs disponibles, sur le marché, on peut citer :

- Le moteur à courant continu
- le moteur asynchrone triphasé.

### 2.2.2 Moteur à courant continu

Le moteur à courant continu est une machine électrique destinée, à transformer de l'énergie électrique sous tension et courant continu, en énergie mécanique. Comme toutes les machines électriques, la machine à courant continu est une machine réversible, permettant également la transformation énergétique inverse. La machine à courant continu est composée de deux parties principales.

Le stator ou l'inducteur qui porte, le bobinage qui crée le flux.

Le rotor ou l'induit porte le bobinage qui tourne, dans le flux inducteur est le siège de forces électromotrices induites. Le moteur à courant continu est utilisé, pour les vitesses variables et les basses tensions. Il s'impose, dans les faibles puissances.

### 2.2.3 Moteurs asynchrones triphasés

Dans un moteur asynchrone triphasé, le champ magnétique tournant est obtenu à l'aide de trois champs alternatifs produits, par trois bobines à 120 degrés alimentées, par des tensions alternatives.

Les organes électriques de ce moteur sont :

- ◆ Le stator qui est bobiné ;
- ◆ Le rotor qui est, soit en « court circuit » ou (en gage d'écureuil : barres conductrices reliées par anneaux conducteurs), soit bobiné et à bagues.
- ◆ La plaque à bornes (normalisée) permettant, le branchement du moteur.

#### a- Branchement du moteur asynchrone triphasé

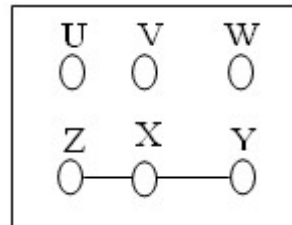
Il existe deux possibilités de couplage, des bobines du stator :

- **Couplage étoile**

- On relie les extrémités x, y, z des bobines entre elles,
- Les extrémités u, v, w sont reliés chacune à une phase du réseau d'alimentation, la tension aux bornes de chaque bobine est alors égale à :

$$u / \sqrt{3} = 380 / \sqrt{3} = 220V .$$

Pour réaliser ce couplage sur la plaque à bornes, raccorde x, y, z à l'aide d'une barrette les bornes u, v, w restant disponible, le branchement sur le réseau figure (II.13).



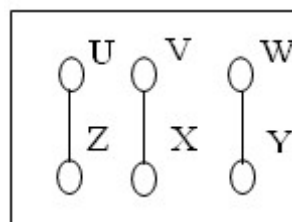
**Figure II.13** : Couplage étoile.

- **Couplage triangle**

On relie les bornes x et v, y et w, z et u figure

La tension aux bornes des bobines est égale à la tension U entre phases du réseau.

Pour réaliser ce couplage sur la plaque à bornes, on relie les bornes indiquées à l'aide de trois barrettes figure (II.14).



**Figure II.14** : Couplage triangle.

### Remarque

Un moteur dont la plaque indique 220/380 V pourra être branché en étoile, sur le 380 V ou en triangle sur le 220V. Ses performances seront identiques, puisque les bobines seront alimentées en 380 V dans les deux cas.

**b- Démarrage du moteur asynchrone**➤ *Démarrage direct*

On choisit, le montage étoile ou tringle en fonction des caractéristiques du moteur et on couple directement, les enroulements du stator au réseau.

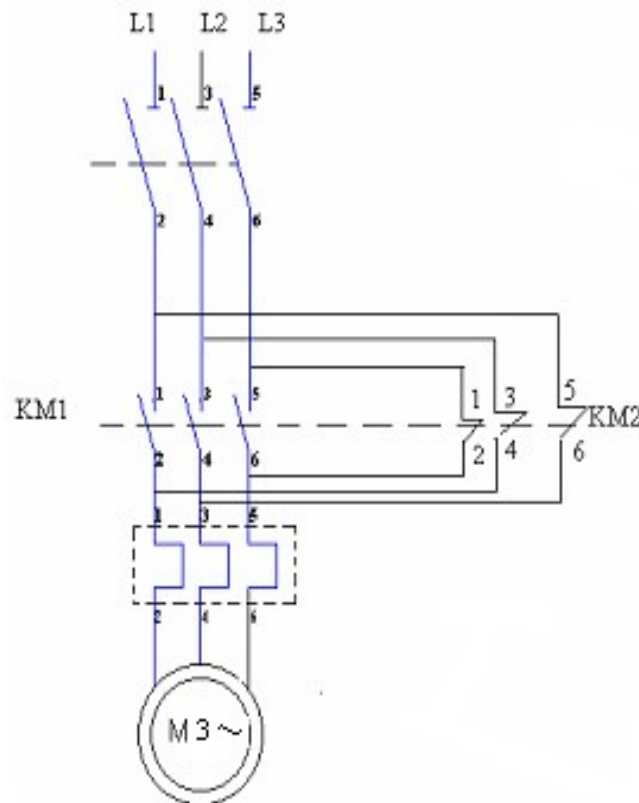
Les moteurs asynchrones triphasés pouvant tourner dans le sens ou dans l'autre, on peut prévoir un montage permettant d'inverser à la demande le sens de rotation (figure II.15).

Ce mode de démarrage présente des Avantages suivants

- Simplicité : peu d'appareillage, câblage simple;
- Couple de démarrage important ;
- Réservé aux moteurs à cage ;
- Démarrage rapide.

Il présente aussi les Inconvénients suivants

- Surintensité importante au moment de démarrage  $I_d = 4 \text{ à } 8 I_n$ .
- Démarrage brusque.



**Figure II.15 :** Schéma de démarrage direct, d'un moteur asynchrone triphasé avec inversion de rotation.

➤ **Démarrage étoile triangle**

Pour pouvoir utiliser ce type de démarrage, il faut :

- Avoir accès, sur la plaque à bornes, à toutes les extrémités d'enroulement x, y, z, u, v, w.
- Que le couplage triangle corresponde à la tension du réseau : moteur 220-380V pour une tension de réseau 220 V.

**Principe :** on effectue le démarrage, en deux étapes :

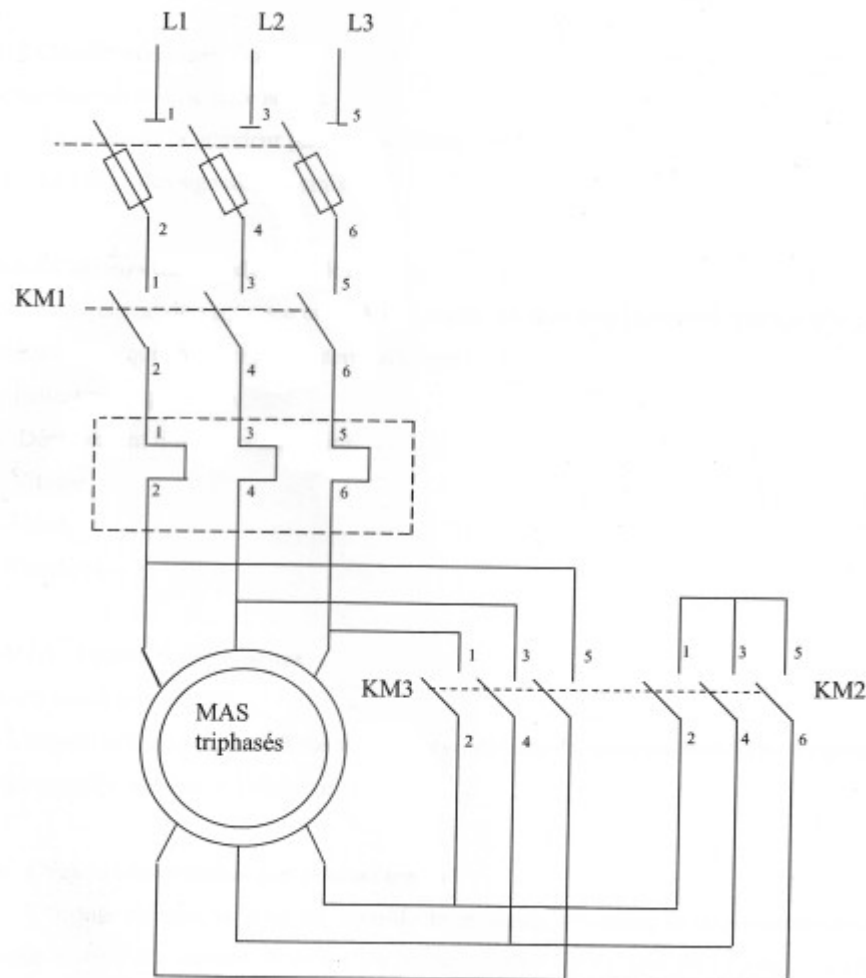
1- couplage étoile du moteur :

Les enroulements sont soumis, à la tension 220V. La tension de démarrage est plus réduite qu'en démarrage direct. Id de l'ordre 1.3 à 2.6 In et Cd= 1/3 Cn faible.

2-couplage triangle du moteur :

Quand le moteur est lancé, on passe au couplage triangle. La surintensité qui en résulte est peu importante, et le moteur atteint, sa vitesse de régime à plain tension.

Le passage de 1 à 2 est réalisé à l'aide d'un contacteur comportant, un contact temporisé qui fixe la durée du couplage étoile (KM2) (figure II.16).



**Figure II.16** : Schéma de démarrage étoile triangle d'un moteur asynchrone triphasé.

### 3 Capteurs

Les capteurs sont des composants destinés à détecter ou à mesurer une grandeur physique et la transformer en une information (analogique, logique ou numérique) exploitable par la partie commande. Ce retour d'information permet à la partie commande de savoir l'état de la partie opérative et de s'assurer de l'exécution des actions générées.

On peut caractériser les capteurs selon deux critères :

En fonction de la grandeur mesurée : on parle alors de capteur de position, de température, de vitesse, de force, de pression, etc.; [4]

En fonction du caractère de l'information délivrée : on parle alors de capteurs logiques appelés aussi capteurs tout ou rien (TOR), de capteurs analogiques ou numériques.

On peut alors les classer en deux catégories, les capteurs à contact qui nécessitent un contact direct avec l'objet à détecter et les capteurs de proximité. Chaque catégorie peut être subdivisée en trois catégories de capteurs : les capteurs mécaniques, électriques, pneumatiques.

### 3.1 Capteur de proximité

#### 3.1.1 Capteur magnéto inductif

Les capteurs inductifs produisent à l'extrémité de leur tête de détection un champ magnétique oscillant. Ce champ est généré par une self et une capacité montée en parallèle. Lorsqu'un objet métallique pénètre dans ce champ, il y a perturbation de ce champ puis atténuation du champ oscillant. Cette variation est exploitée par un amplificateur qui délivre un signal de sortie, le capteur commute. [4]

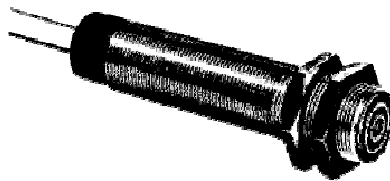


Figure II.17 : Capteur inductif.

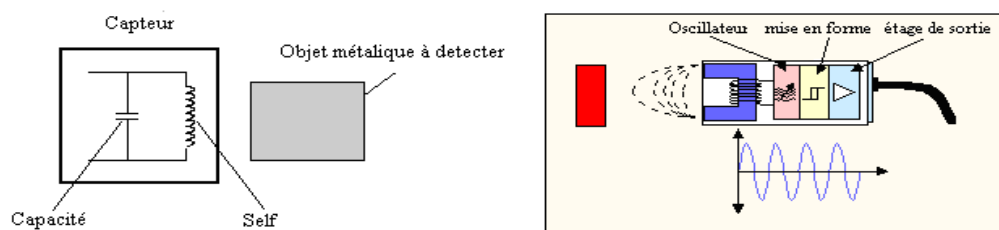
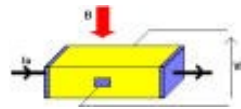


Figure II.18 : Oscillateur.

#### 3.1.2 Capteur à effet Hall

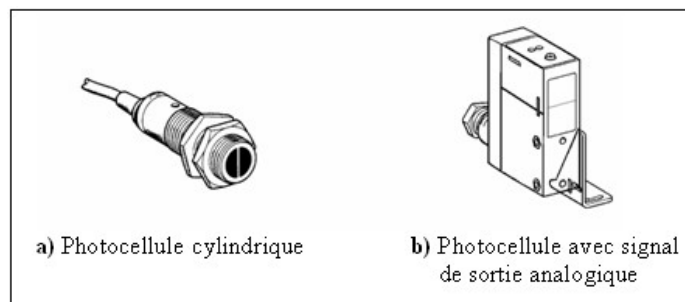
Dans ce cas, il s'agit de mesurer un champ magnétique qui est créée par un aimant (pièce à repérer) ou un inducteur (fonctionnement d'une machine électrique ou d'un transformateur). La plaque en semi-conducteur à effet Hall (figure II.19) placée dans un champ magnétique  $B$  voit apparaître à ses bornes une tension proportionnelle au champ. [4]



**Figure II.19** : Capteur à effet Hall.

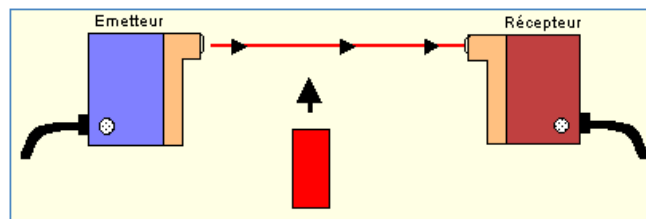
### 3.1.3 Capteur optique

Une photodiode est un capteur de proximité. Il se compose d'un émetteur de lumière associé à un récepteur. La détection d'un objet se fait par coupure ou variation d'un faisceau lumineux. Le signal est amplifié pour être exploité par la partie commande. [4]

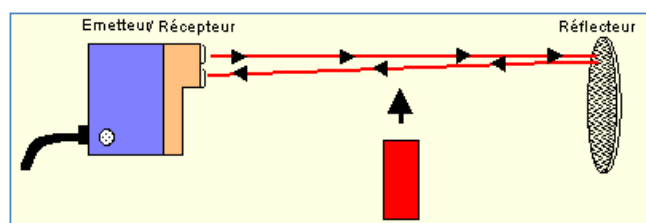


**Figure II. 20:** Capteur optique.

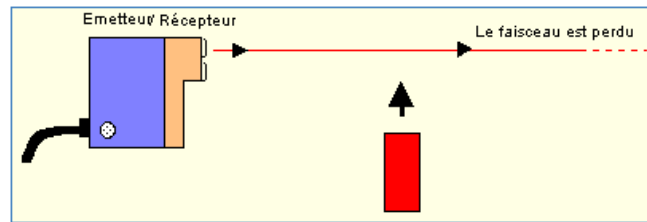
Ces capteurs utilisent différents modes de détection qui sont schématisés par les figures suivantes :



**a)** Système barrage.



**b)** Système reflex.



c) Système de proximité.

## 3.2 Capteurs de température

Un grand nombre de lois physiques sont sensibles à la température. C'est pourquoi il existe une grande diversité de types de capteurs de température.[4]

### 3.2.1 Mesure par thermocouple

Un thermocouple est constitué de deux conducteurs A et B formant entre eux deux jonctions aux températures  $T_1$  et  $T_2$ . La première température est connue et sert de référence et l'autre est placée dans le milieu étudié et représente la température inconnue. Il apparaît une différence de potentiel entre les deux électrodes qui est fonction de la différence de température et du couple de métaux utilisé.[4]

**Figure II. 21** : Exemple de thermocouples industriels.

### 3.2.2 Thermométrie par diodes et transistors

Une jonction silicium diode ou transistor alimenté par un courant constant donne une tension proportionnelle à la température. On utilise cette propriété pour fabriquer des capteurs. En général, on ajoute un circuit intégré pour conditionner le capteur. Il existe aussi des circuits intégrés permettant d'enregistrer la température sur une période longue. Cela est utilisé dans les systèmes de contrôle de la chaîne du froid en introduisant le capteur dans le produit au début de la chaîne (abattoir) et en lisant les données à la fin (magasin).

Il existe aussi des mesures de température basées sur la mesure du bruit de fond qui représente l'agitation thermique des électrons dans un matériau. On peut aussi citer les mesures par quartz dont la fréquence de résonance dépend de la température.

### 3.3 Capteur de vitesse

Les capteurs de vitesse permettent de mesurer une vitesse ou une position relative entre deux points.

#### 3.3.1 Codeurs incrémentaux

Les codeurs incrémentaux sont utilisés dans les applications de commande des machines asynchrones. Trois sorties sont utilisées (A, B, Top0). Les voies A et B sont une succession de 0 et 1. Le codage se fait en comptant les impulsions.

Un décalage entre les deux voies permet de connaître le sens de rotation de la machine. Le Top0 recale à chaque tour la position initiale en cas de raté pendant un tour. Les codeurs classiques sont codés sur 10 bits (1024 positions) et les plus sophistiqués sur 12 bits (4096 positions). Selon le même principe du comptage, on utilise des capteurs de proximité avec des roues crantées. Ce système ne permet pas de connaître le sens de rotation mais est utilisé dans les systèmes ABS.[4]



**Figure II.22 :** Codeur incrémental 1024 points.

#### 3.3.2 Tachymètres

Un machine à courant continu à excitation par aimant délivre à vide une fem proportionnelle à la vitesse. On l'appelle Dynamo tachymétrique. Son principal inconvénient, comme toutes les machines à courant continu, est le frottement des balais sur les collecteurs qui provoque une usure importante. Une variante est l'alternateur tachymétrique qui délivre trois tensions sinusoïdales décalées de  $2\pi/3$  et dont la tension est proportionnelle à la vitesse. La précision de ce type de capteur est moins grande que celle de la dynamo tachymétrique. [4]

### 3.4 Capteurs de position

En génie électrique, on utilise les capteurs de position pour obtenir une position absolue, que ce soit en linéaire ou en angle. Les capteurs de positions relatives seront traités comme capteurs de vitesse.[4]

#### 3.4.1 Capteurs potentiométriques

Un potentiomètre linéaire ou rotatif permet si il est alimenté de déterminer la position. En rotation, ce principe peut être utilisé si le système tourne lentement car le frottement du capteur impose une usure. Nous avons ici plusieurs capteurs de position angulaire. Ce sont des potentiomètres qui n'ont pas de butée de fin de course.

#### 3.4.2 Capteurs digitaux

Le principe est de coder la position sur un certain nombre de bits pour obtenir un nombre fini de position. En augmentant le nombre de bits, on augmente la précision du capteur. Les capteurs classiques sont codés sur 10 bits (1024 positions) et les plus sophistiqués sur 12 bits (4096 positions). Au delà, la précision des secteurs devient difficile à lire même pour un capteur optique par diode et phototransistor.

### 3.5 Capteur de pression

Dans un capteur de pression, on mesure la force qui s'exerce sur la surface constante et connue d'un corps d'épreuve.

#### Capteur à seuil de pression pneumatique

Ce sont des capteurs fin de course qui se montent directement sur les vérins. Pour pouvoir fonctionner correctement, il est nécessaire de les coupler avec une cellule « non inhibition à seuil ».

Le principe de fonctionnement de ce capteur est d'utiliser la contre pression qui existe dans la chambre non soumise à la pression du réseau. Lorsque le piston subit une pression il se déplace. Ce déplacement entraîne une réduction du volume de la chambre qui n'est pas soumise à la pression du réseau. Ceci entraîne une augmentation de la contre pression qui est amplifiée par des régleurs de débit. Lorsque le vérin arrive en fin de course, cette contre pression chute. Lorsqu'elle est inférieure à 1/12<sup>ème</sup> de la pression du réseau le

capteur déclenche. On peut traduire cette information, soit par un signal électrique soit par un signal pneumatique.[4]

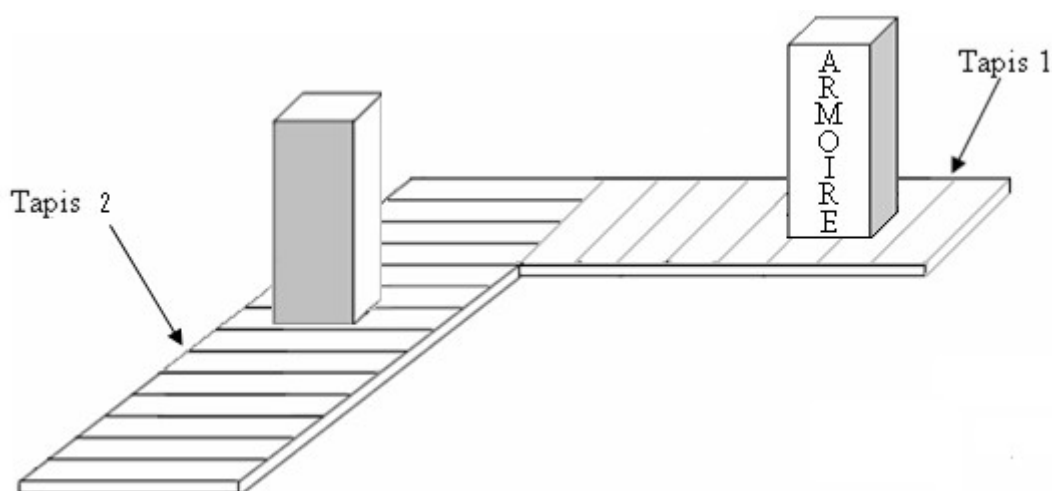
## 1. Description et étude du fonctionnement du système

Le système que nous avons à étudier est un dispositif électromécanique qui permet le transfert automatique des armoires d'une chaîne d'assemblage (1) de réfrigérateur vers une autre chaîne d'assemblage (2) en repositionnant les armoires de façon à ce que leurs faces arrières soient en face des ouvriers permettant ainsi à chacun d'effectuer sa tâche. Ce système assure dans des conditions de fonctionnement optimales le transfert d'une moyenne de 900 armoires par jour, ce qui justifie l'intérêt de cet équipement pour l'entreprise.

La principale question qui se dégage de cette petite introduction est la suivante :

Comment effectuer ce transfert ?

A priori, on peut penser à placer une longue chaîne à écaille entraînée par un moteur électrique permettant ainsi aux ouvriers d'effectuer les différents assemblages, mais pour des raisons diverses telles que : la longueur de la chaîne d'assemblage et une meilleure gestion de l'espace, une autre solution qui consiste à subdiviser cette chaîne en deux placées en équerre s'impose figure (III.1).



**Figure III.1** : Chaîne à écaille subdivisée en deux.

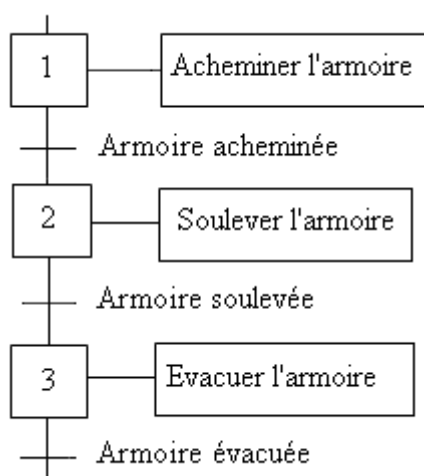
Ceci nous amène à poser la problématique suivante :

De quelle manière faut-il procéder pour transférer l'armoire de la première chaîne vers la deuxième tout en la faisant tourner avec un angle de 90°?

La solution proposée consiste à :

- ◆ Acheminer l'armoire vers le poste de transfert situé à la fin du premier tapis ;
- ◆ Transférer l'armoire vers le poste d'évacuation situé sur le deuxième tapis;
- ◆ Évacuer l'armoire.

Ceci est représenté par le GRAFCET de niveau 1 suivant :

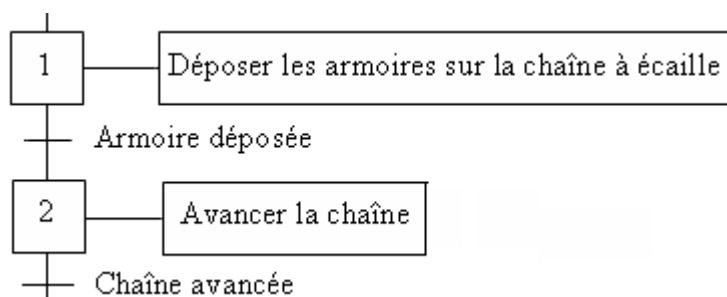


**Figure III.2** : GRAFCET de niveau 1.

### 1.1 Acheminement des armoires

Pour acheminer les armoires transférées par un convoyeur vers le poste de transfert, il faut : Placer les armoires sur la chaîne ;

- ◆ Faire avancer la chaîne ;



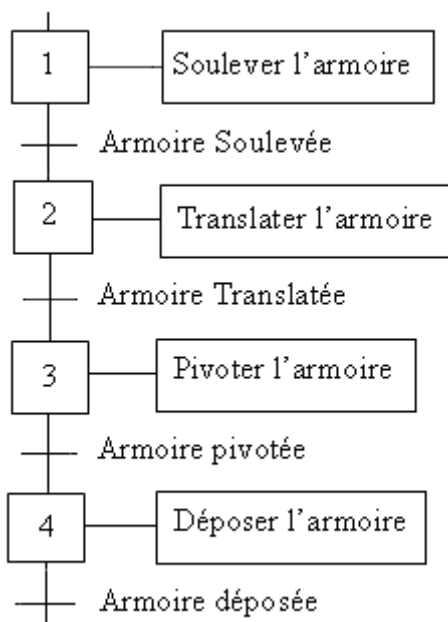
**Figure III.3** : GRAFCET de niveau 1 pour l'acheminement des armoires.

## 1.2 Transfert des armoires

Pour assurer ce transfert, il faut passer par les étapes suivantes :

- ◆ Soulever l'armoire ;
- ◆ Translater l'armoire ;
- ◆ Pivoter l'armoire avec un angle de  $90^\circ$  ;
- ◆ Déposer l'armoire sur la deuxième chaîne.

Ce processus est représenté par le grafcet suivant :

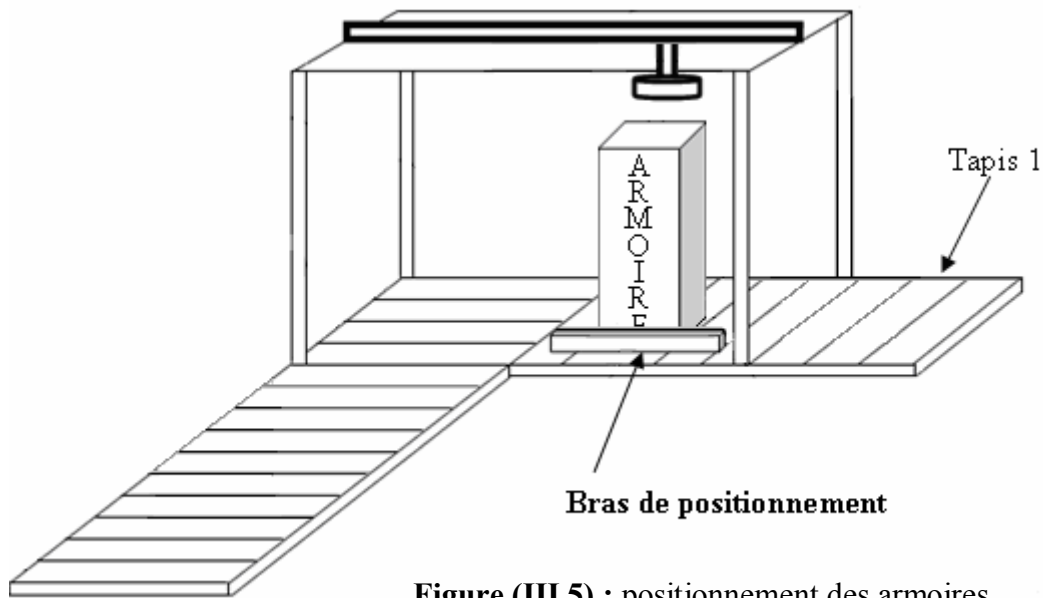


**Figure III.4 :** GRAFCET de niveau 1 pour transfert des armoires.

Pour soulever l'armoire il faut passer par les sous étapes suivantes :

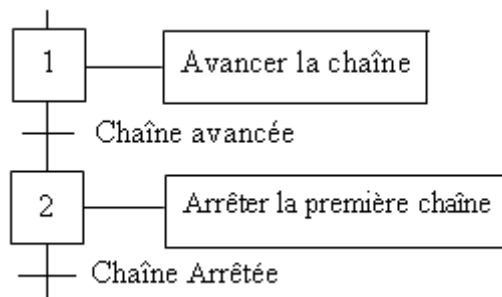
- ◆ Arrêter l'avancement de la première chaîne ;
- ◆ Poser l'armoire dans une position régulière;

Voir La figure (III.5)



**Figure (III.5) :** positionnement des armoires.

Le GRAFCET de cette étape est représenté par la figure suivante :



**Figure III.6 :** GRAFCET de niveau 1 pour positionnement de l'armoire.

En ajoutant ces deux étapes au grafcet de la figure (III.4), il devient :

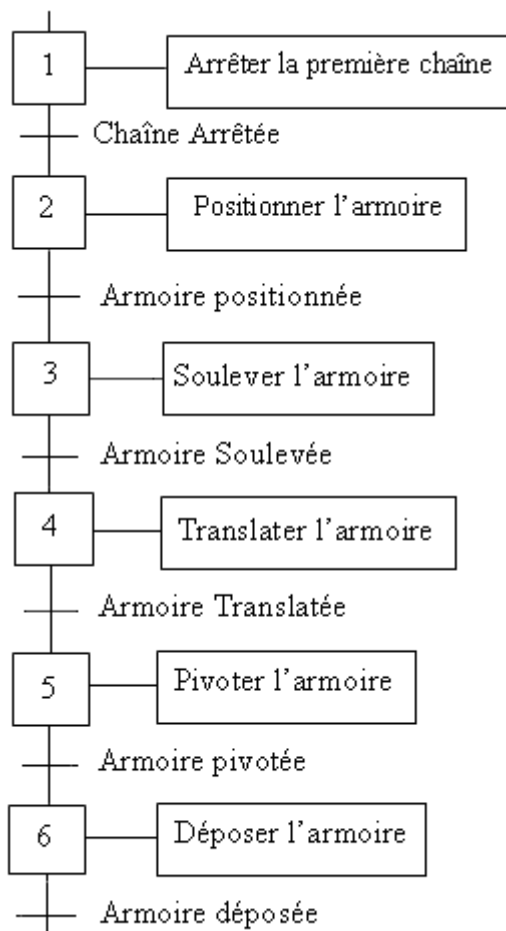
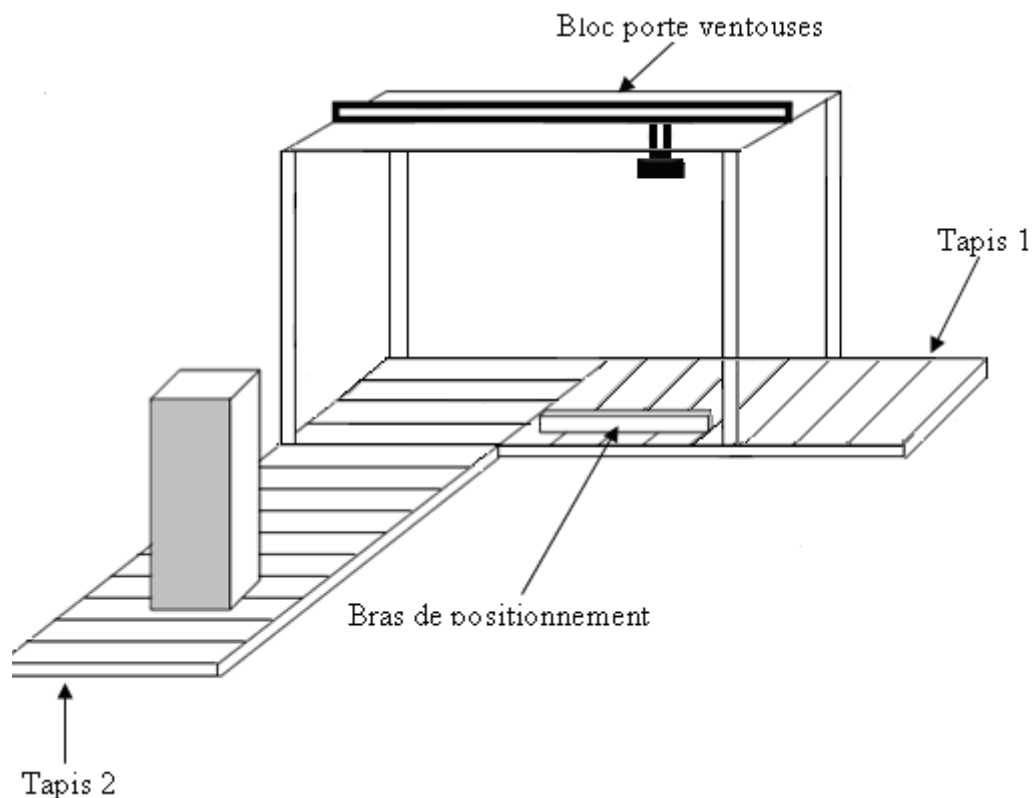


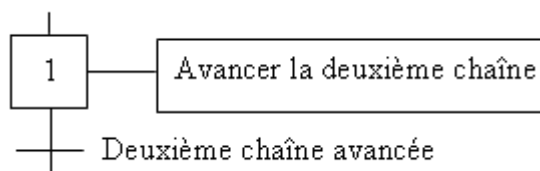
Figure III.7 : GRAFCET de niveau 1 détaillé pour transfert des armoires

### 1.3 Évacuation des armoires

Pour évacuer l'armoire figure (III.8), il suffit de faire avancer la deuxième chaîne.

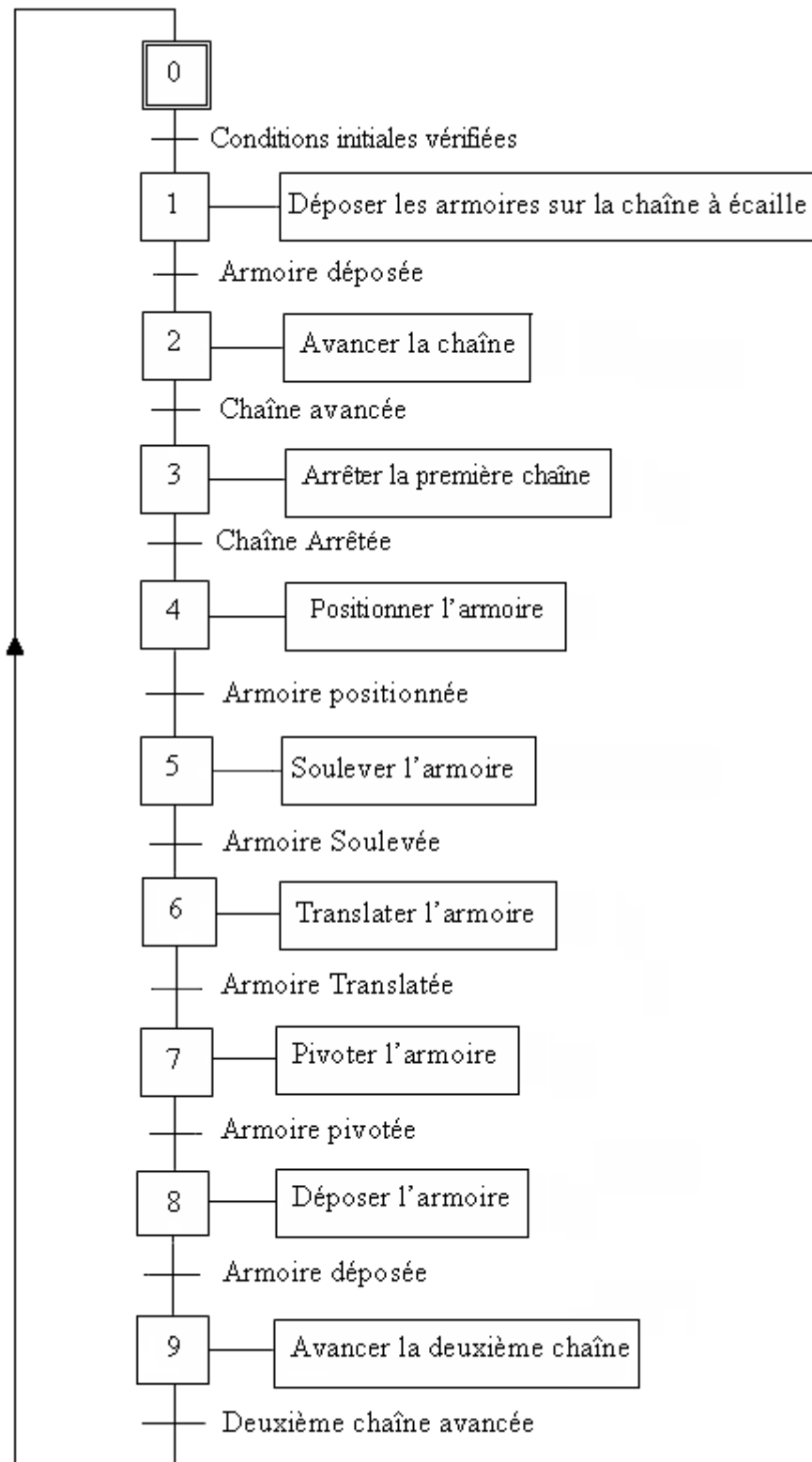


**Figure III.8** : évacuation des armoires.



**Figure III.9** : étape d'avancement de la deuxième chaîne.

Le fonctionnement de notre système est résumé d'une manière globale dans le grafcet de niveau 1 donné ci-dessous :



**Figure III.10** : GRAFCET de niveau 1 détaillé global du système.

## 2 Réalisation du GRAFCET de niveau II

### 2.1 Description littérale de la technologie du système

#### 2.1.1 Acheminement des armoires

La première phase de notre système débute par l'acheminement des armoires transportées par le convoyeur vers le poste de transfert. Pour remplir cette fonction qui paraît très simple, il a fallu placé une sorte de tapis roulant appelé chaîne à écaille entraînée par un moteur asynchrone.

#### 2.1.2 Positionnement des armoires

Les armoires acheminées arrivent à la fin de la première chaîne dans des positions irrégulière. Pour les repositionner, on a placé un bras de positionnement dont la sortie et l'entrée est commandée par un vérin pneumatique à double effet. En plus de ceci, il faut arrêter la chaîne à écaille à chaque fin de parcours de l'armoire.

Toute cette synchronisation est commandée par deux photocellules et un capteur magnéto inductif placés à la fin de la chaîne de transfert ; figure (III.11).

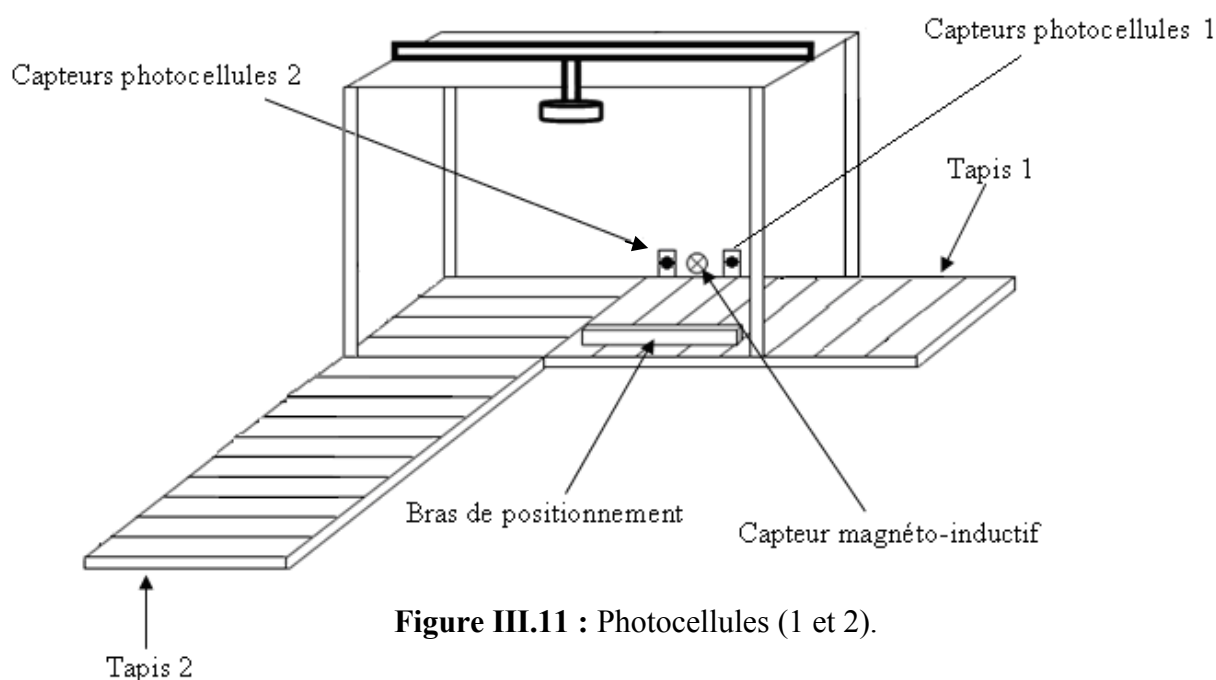


Figure III.11 : Photocellules (1 et 2).

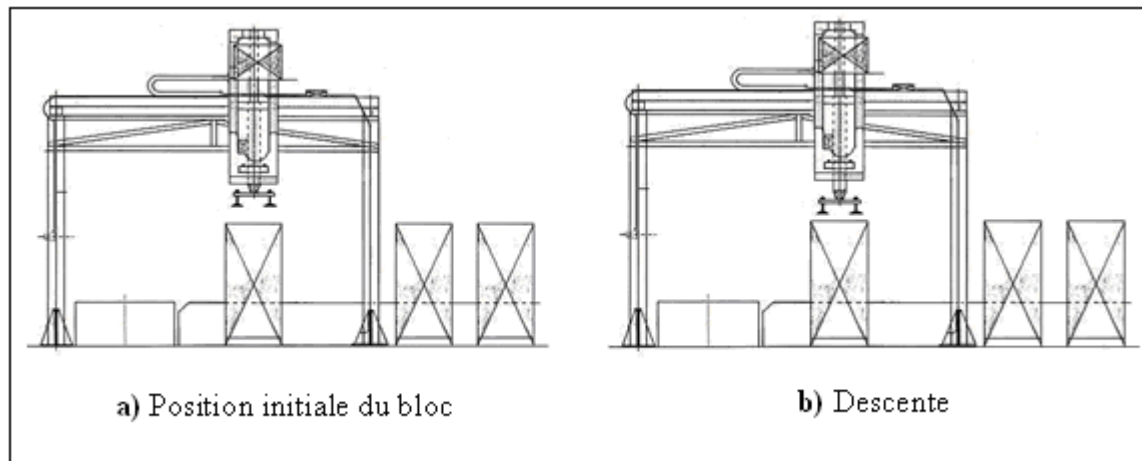
- ◆ La photocellule (1) détecte le passage des armoires qui est indiqué par l'allumage d'une lampe témoin ;
- ◆ La photocellule (2) détecte la présence des armoires induisant un arrêt momentané de la chaîne à écaïlle et la sortie du bras de positionnement ;
- ◆ Le capteur magnétique détecte le bon positionnement des armoires.

### Remarque

Il faut bien souligner que l'arrêt de la chaîne à écaïlle n'entraîne pas l'arrêt du moteur car, ce dernier tourne sans arrêt. Ceci est effectué à l'aide d'un système d'embrayage qui découple la chaîne à écaïlle du moteur.

### 2.1.3 Descente du bloc porte ventouse

La position régulière de l'armoire actionne la sortie d'un vérin pneumatique qui pousse le bloc porte ventouses vers le bas. Dans le cas contraire le vérin ne sera pas actionné.

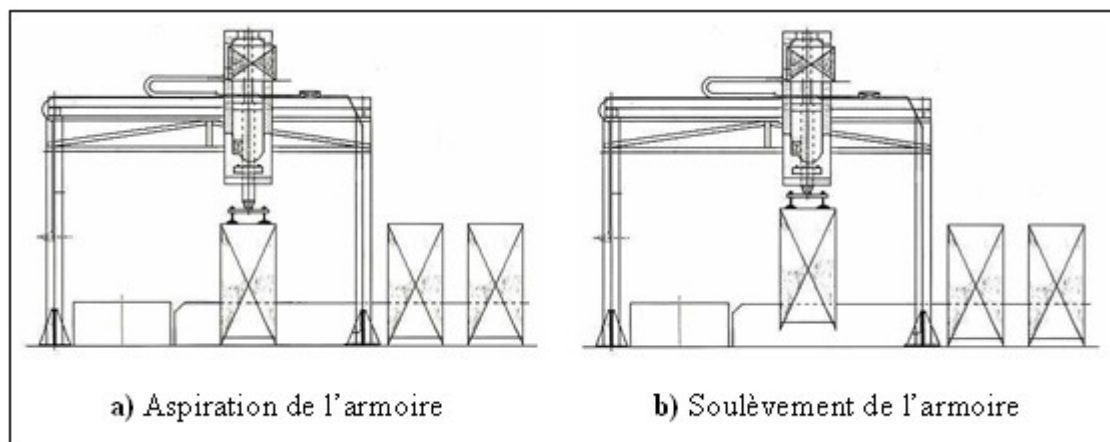


**Figure III.12 :** Descente du bloc porte ventouses.

### 2.1.4. Aspiration et soulèvement des armoires

Après la descente du bloc porte ventouses des capteurs de fin de course à contacte placés en dessous de ces dernières détectent le contact de l'armoire avec les ventouses, si toutes les conditions sont réunies telle la pression, la présence de l'armoire la bonne adhésion des ventouses avec l'armoire..., les ventouses aspirent l'armoire, ce qui actionne l'entrée d'un

vérin vertical qui tire le bloc porte ventouses vers le haut en soulevant l'armoire figure(III.13).



**Figure III.13** : Aspiration et soulèvement de l'armoire.

### 2.1.5 Translation et rotation de l'armoire

Des capteurs de fin de course détectent la limite de montée de l'armoire, si celle-ci est régulière, l'entrée d'un vérin pneumatique horizontal est actionnée. Ceci génère la translation horizontale du bloc qui déplace l'armoire avec lui. Cette translation ne s'effectue que si une photocellule (3) placée au début de la chaîne d'évacuation figure (III.15) détecte la non présence d'une autre armoire.



**Figure III.14** : Photocellules.

Dans ce cas un vérin pneumatique rotatif pivote l'armoire dans le sens horaire avec un angle de 90°. Une autre photocellule (4) figure (III.15) placée juste après la photocellule (3)

confirme alors l'évacuation totale d'une armoire déjà transférée, cette information actionne la sortie du vérin vertical et pousse le bloc porte ventouses vers le bas qui à leur tour lâchent l'armoire, dans le cas contraire ce vérin ne serait pas actionné et l'armoire resterait suspendu.

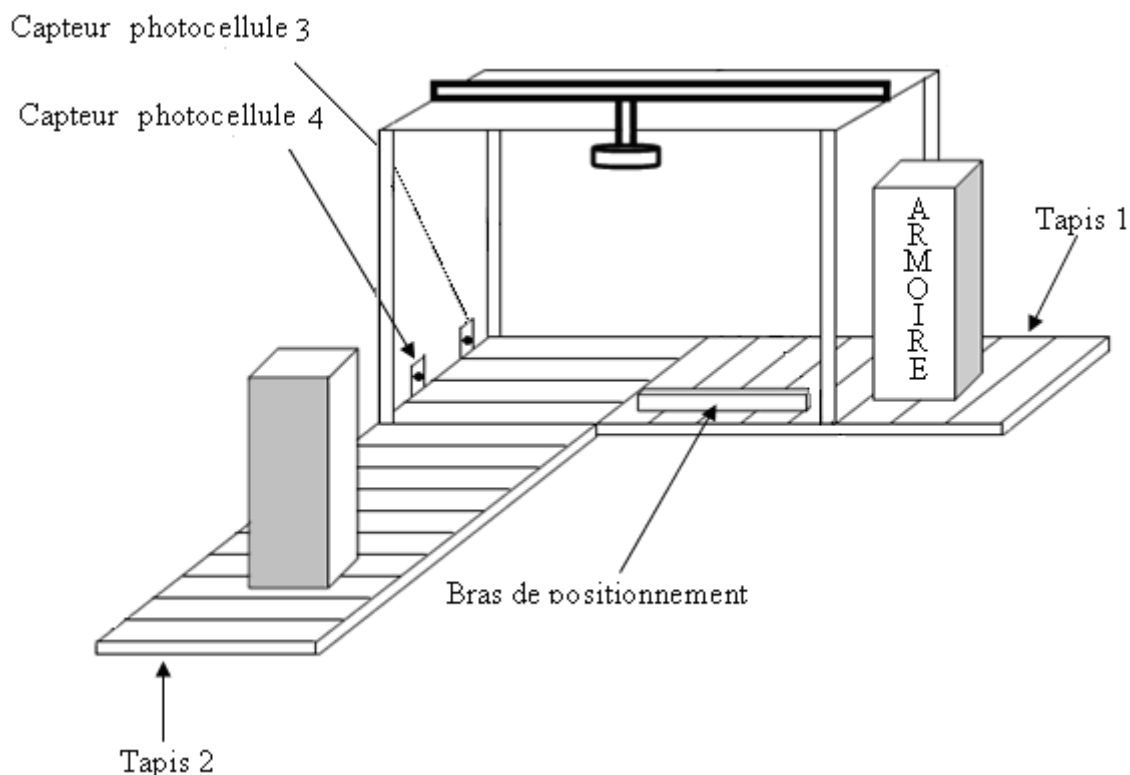


Figure III.15 : photocellules (3et4).

### 2.1.6. Évacuation des armoires

Les armoires transférées sont évacuées à l'aide d'une deuxième chaîne à écaille qui est entraînée par un autre moteur asynchrone.

#### Remarque

Il arrive parfois qu'à la fin de la chaîne à écaille, des armoires dévient leur parcours et créent ainsi un encombrement ou un petit dysfonctionnement de la chaîne de fabrication. Pour éviter ce problème une photocellule (5) dont le rôle est d'arrêter la chaîne dès qu'il y a dépassement est placée à la fin de cette chaîne.

## 2.2 Mise en équation d'un GRAFCET

Pour passer de l'étape de modélisation du procédé par GRAFCET à l'étape de programmation par l'un des langages acceptés par l'automate utilisé, on doit traduire notre GRAFCET sous forme d'équations logiques combinatoires, en utilisant le GRAFCET de niveau 2.

En précisant les conditions d'activation et de désactivation ainsi que les initialisations et les arrêts d'urgence d'une étape et de l'action associée.

L'état d'une étape  $X_n$  peut être notée comme suit :

$X_n=1$  : si l'étape est active ;

$X_n=0$  : si l'étape est inactive ;

De plus, la réceptivité  $t_n$  qui est une variable binaire à pour valeurs :

$t_n=1$  : si la réceptivité associée est vraie ;

$t_n=0$  : si la réceptivité associée est fausse;

Soit le GRAFCET de la figure (III.16).

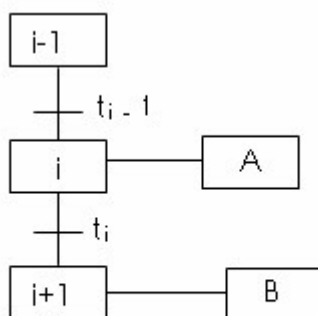


Figure III.16 : exemple de GRAFCET

En tenant compte de la deuxième et la troisième règle d'évolution du GRAFCET on peut déduire les conditions d'activation et de désactivation d'une étape, puis l'équation logique de l'étape.

$CAX_i = X_{i-1}.t_{i-1}$  ; (condition d'activation de l'étape i),

$CDX_i = X_{i+1}$  (condition d'activation de l'étape i),

$A = X_i$  (l'action A associée à l'étape i).

Si les conditions d'activation et de désactivation sont fausses, l'étape i reste dans son état (effet mémoire)

L'équation logique d'une étape sera comme suit :

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

Si on tenant compte de mode marche/ arrêt et arrêt d'urgence ;

INIT = 1 : initialisation du GRAFCET (mode arrêt)

INIT = 0 : écoulement du cycle (mode marche).

AUDoux = 1 de toutes les actions en maintenant les étapes actives.

AUDur = 1 : désactiver les étapes du GRAFCET.

L'équation devienne :

- Pour les étapes initiales :

$$X_i = (CAX_i + X_i \overline{CDX_i} + INIT) \cdot \overline{AUDur}$$

Avec :  $CAX_i = X_{i-1} \cdot t_{i-1} \cdot INIT \cdot \overline{AUDur}$

$$CDX_i = X_{i+1} \cdot \overline{INIT} + AUDur$$

- Pour les étapes non initiales :

$$X_i = (CAX_{i-1} + X_i \overline{CDX_i}) + \overline{INIT} \cdot \overline{AUDur}$$

Avec :  $CAX_i = X_{i-1} \cdot t_{i-1} \cdot \overline{INIT} \cdot \overline{AUDur}$

- $CDX_i = X_{i+1} \cdot \overline{INIT} + AUDur$

Pour les actions :

$$A = X_i \bullet \overline{AUDoux}$$

Pour passer à la programmation on doit écrire les conditions d'activations et de désactivation de chaque étape et leurs actions associées.

## 2.3 Liste des capteurs, pré capteurs et actionneurs

### 2.3.1 Actionneurs

#### a- Moteurs

**MOT1** : moteur triphasé asynchrone 1 commandant l'avance du tapis 1.

$\left\{ \begin{array}{l} MOT1AV : \text{moteur en marche.} \\ MOT1ART : \text{moteur à l'arrêt.} \end{array} \right.$

**MOT2** : moteur triphasé asynchrone 2 commandant l'avance du tapis 2.

$\left\{ \begin{array}{l} MOT2AV : \text{moteur en marche.} \\ MOT2ART : \text{moteur à l'arrêt.} \end{array} \right.$

#### b- Vérins

**Vp** : Vérin double effet pour positionner les armoires.

$\left\{ \begin{array}{l} Vp\_p : \text{sortie du bras de positionnement} \\ Vp\_n : \text{entrée du bras de positionnement} \end{array} \right.$

**Vv** : vérin double effet pour le déplacement vertical du bloc porte ventouses.

$\left\{ \begin{array}{l} Vv\_p : \text{descente du bloc porte ventouses.} \\ Vv\_n : \text{montée du bloc porte ventouses.} \end{array} \right.$

**Vh** : Vérin double effet pour le déplacement horizontal du chariot.

{ **Vh\_p** : translation horizontale à gauche.  
**Vh\_n** : translation horizontale à gauche.

**Vr** : Vérin rotatif pour la rotation du bloc porte ventouses.

{ **Vr\_p** : translation horizontale à gauche.  
**Vr\_n** : translation horizontale à gauche.

### 2.3.2 Pré actionneurs

- *Électrovannes*

**EV1** : électrovanne pour actionner le vérin Vp

**EV2** : électrovanne pour actionner le vérin Vv

**EV3** : électrovanne pour actionner le vérin Vh

**EV4** : électrovanne pour actionner le vérin Vr

### 2.3.3 Capteurs

**CFc1** : capteur photocellule pour l'incrémentation du compteur ;

**CFc2** : capteur photocellule pour l'arrêt du MOT1 et l'actionnement de Vv ;

**Cm** : capteur de bonne position des armoires (capteur magnéto-inductif) ;

**Cfh** : capteur de fin de course de la position haute du Vv ;

**Cfb** : capteur de fin de course de la position basse du Vv ;

**Cfa** : capteur de fin de course du mouvement de l'avance de Vh ;

**Cfr** : capteur de fin de course du mouvement de recul de Vh ;

**Cfrg** : capteur de fin de course gauche du mouvement de rotation de Vr ;

**Cfrd** : capteur de fin de course droite du mouvement de rotation de Vr ;

**CapP** : capteur de bonne adhésion armoires/ventouses ;

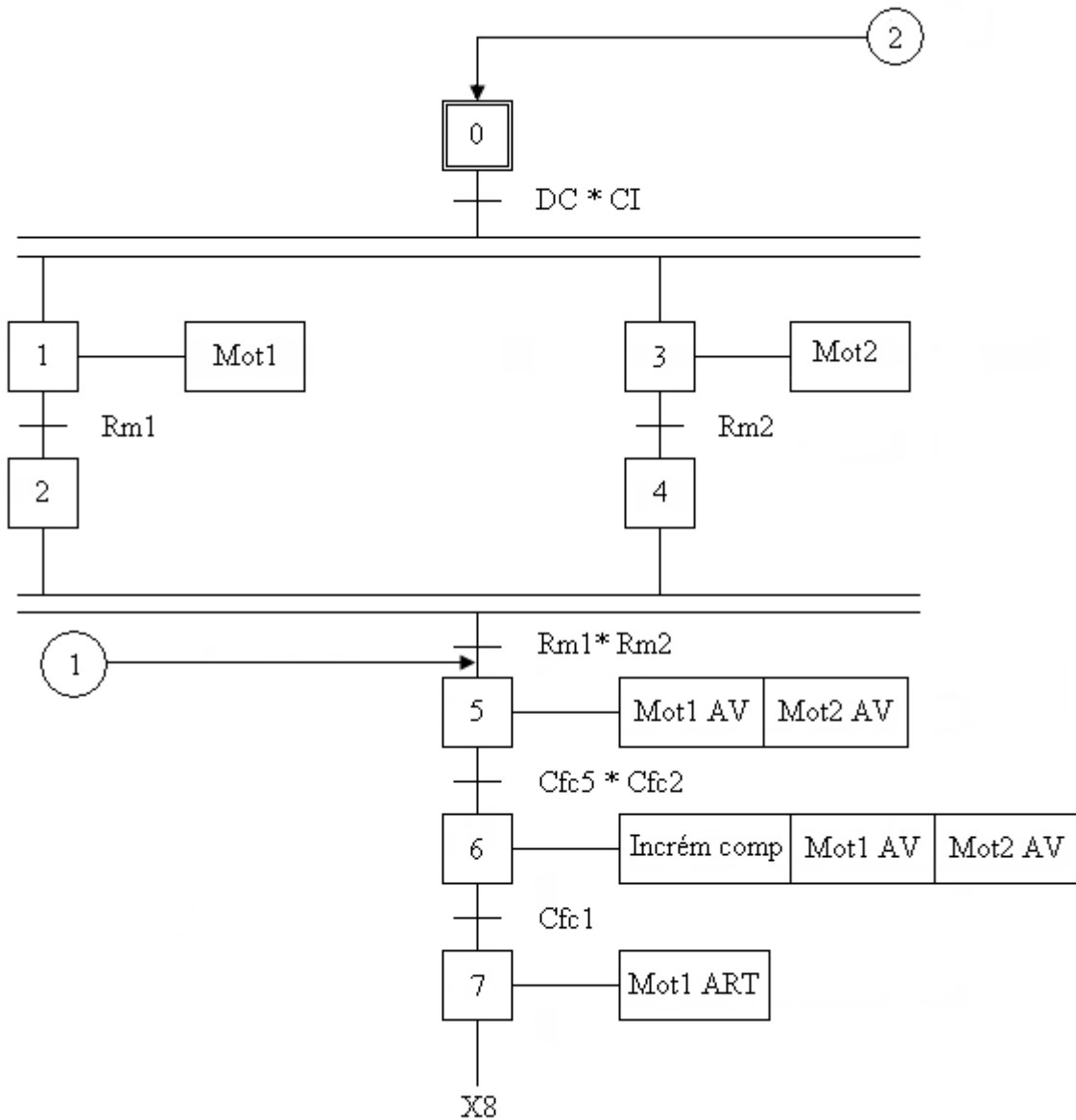
**CFc3** : capteur photocellule pour actionner Mot1 et Vh+ ;

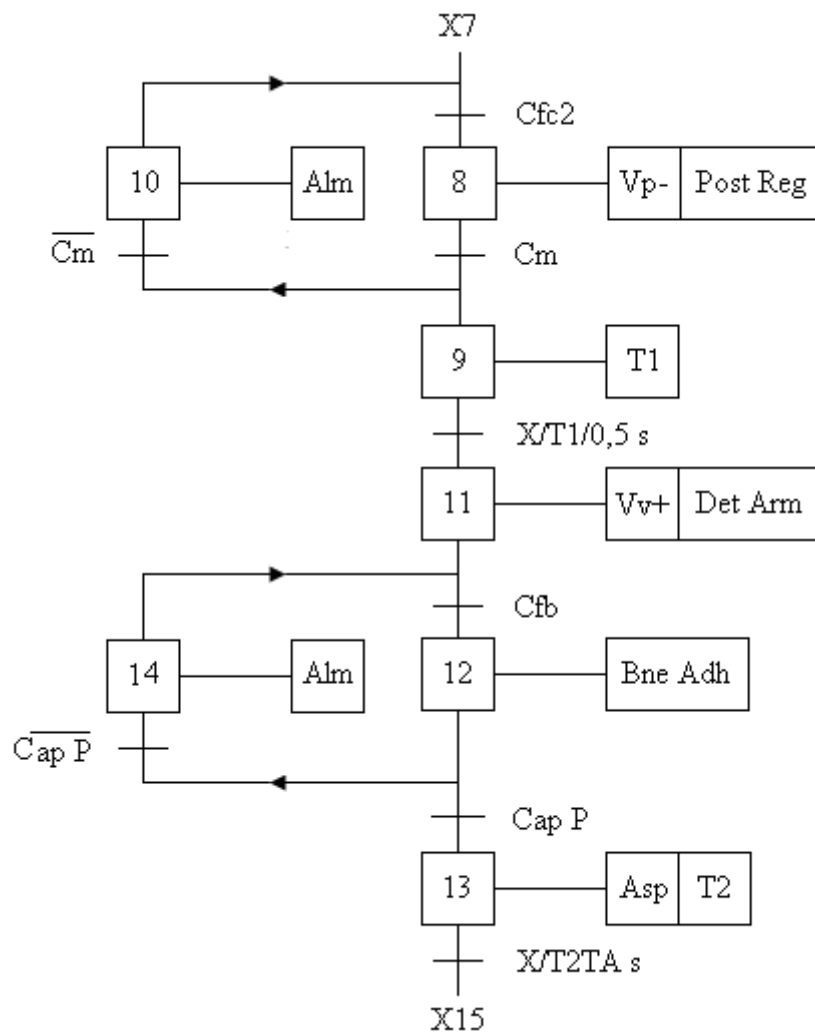
**CFc4** : capteur photocellule pour commander le Tapis 2 ;

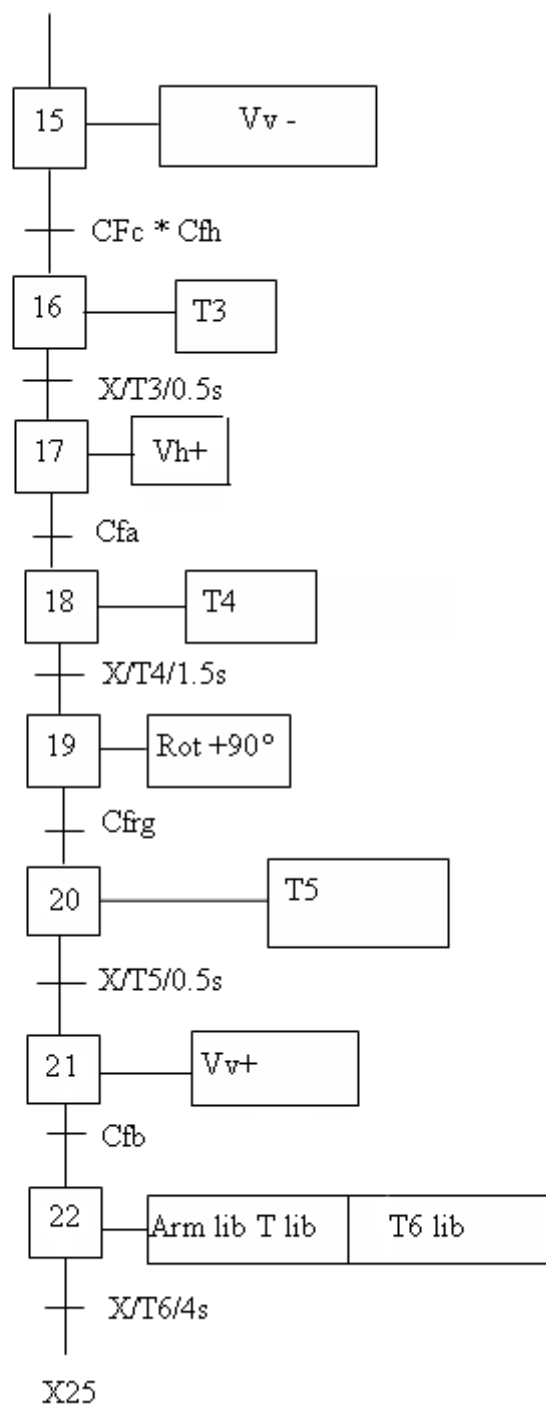
**Str** : bouton poussoir du début du cycle ;

**Cfc5** : capteur photocellule pour la sécurité.

On résumant le GRAFCET de niveau 1 et la technologie appliquée on aura le GRAFCET de niveau 2 représenté dans la figure (III.17).







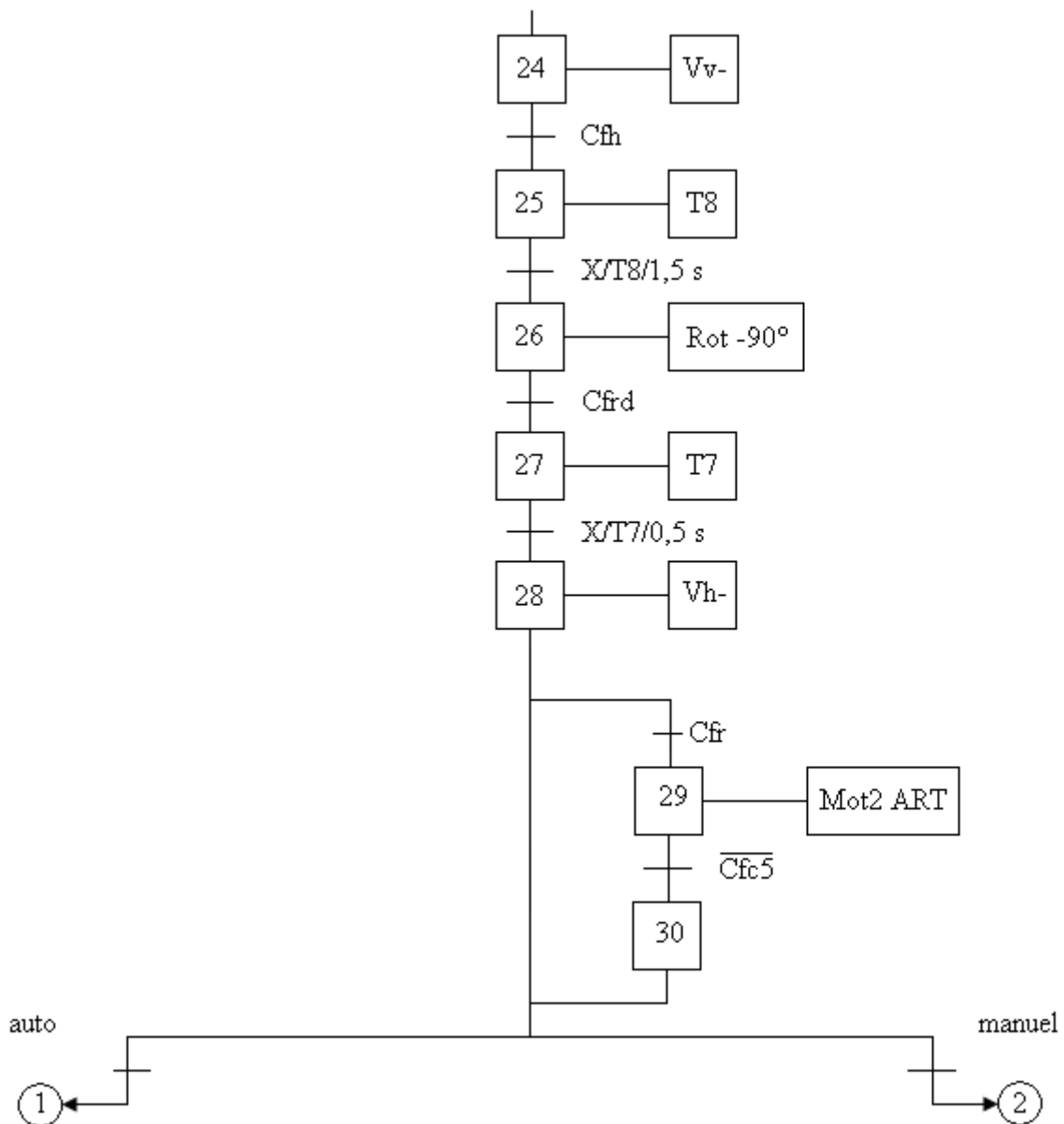


Figure III.17 : GRAFCET de niveau 2 pour le système étudié.

## Conclusion

Après modélisation du système par le GRAFCET qui lui correspond, on passe dans ce qui suit à l'élaboration du programme puis on arrive à l'étape décisive du travail effectuer par simulation à l'aide du logiciel S7-PLCSIM, le bon fonctionnement du système.

## Introduction

Les automates programmables SIEMENS sont des appareils fabriqués en série. Tous les éléments logiques, fonctions de mémoire, temporisations, compteurs....etc. nécessaires à l'automatisation sont prévus par le fabricant et sont intégrés à l'automate.

### 1 Choix d'un automate

D'après le cahier de charges établi, l'automate choisi devra répondre à certains critères qui sont :

- La capacité de traitement du processeur ;
- Le nombre d'entrées/sorties ;
- La nature des entrées/sorties (numériques, analogiques, booléennes) ;
- La fiabilité ;

Pour notre cas, l'unité froid de l'ENIEM, qui est concernée par le projet, a porté son choix sur l'automate programmable industriel « SIMATIC S7-300 »

### 2 Présentation de l'automate S7-300

L'automate S7-300 est constitué d'une alimentation, d'une CPU, de modules d'entrées et des modules de sorties. A ceux-ci peuvent s'ajouter des processeurs de communication et des modules de fonctions spéciales telles que la commande d'un moteur pas à pas, régulation PID ou autre. [2]

Comme tout API, l'automate S7-300 est un système modulaire offrant la gamme de modules suivants :

- Modules d'alimentation (PS) avec 2A, 5A ou 10A.
- Unités centrales (CPU315) travaillant avec une mémoire de 48ko, sa vitesse d'exécution est 0.3 ms/1k instructions par exemple.
- Modules de signaux (SM) pour entrées et sorties TOR et analogiques.
- Modules d'extension (IM) pour configuration multi- rangée de S7-300

- Modules de fonction (FM) pour fonctions spéciales (par exemple l'activation d'un moteur pas à pas).
- Processeurs de communication (CP) pour la mise en réseau [7]

## 2.1 Caractéristiques du S7-300

L'automate S7-300 offre les caractéristiques suivantes :

1. gamme diversifiée de CPU ;
2. gamme complète de modules ;
3. possibilité d'exécution jusqu'à 32 modules ;
4. bus de fond de panier intégré au module ;
5. possibilité de mise en réseau avec MPI, PROFIBUS ou INDUSTRIAL ETHERNET ;
6. raccordement central de la PG avec accès à tous les modules ;
7. liberté de montage aux différents emplacements ;
8. configuration et paramétrages à l'aide de l'outil configuration matérielle.

Des automates S7-300 peuvent communiquer entre eux au moyen d'un câble PROFIBUS

## 2.2 Construction de l'automate S7-300

### 2.2.1. Module d'alimentation

Tout réseau 24 volt industriel peut être utilisé pour alimenter la CPU du S7-300. Les modules d'alimentation suivants de la gamme S7 sont prévus pour être utilisés.

Désignation	CS	Tension à la sortie	Tension à l'entrée
PS 307	2A	DC 24 v	AC 120v/230v
PS 307	5A	DC 24 v	AC 120v/230v
PS 307	10A	DC 24 v	AC 120v/230v

### 2.2.2. Unité centrale (CPU)

La CPU (Central Processing Unit) est le cerveau de l'automate. Elle lit les états des entrées, ensuite elle exécute le programme utilisateur en mémoire et enfin, elle commande les sorties (actions). Elle comporte une unité de commande et de calcul, des mémoires, un programme système et des interfaces vers les modules de signaux.

La gamme S7-300 offre une grande variété de CPU telles que la CPU312, 314, 314IFM, 315, 315-2dp,...etc. et chaque CPU possède certaines caractéristiques différentes des autres. Par conséquent le choix de la CPU pour un problème d'automatisation donné est conditionné par les caractéristiques offertes par la CPU choisie.

La CPU 315-2DP peut aussi être utilisée dans un sous-réseau PROFIBUS :

- comme maître DP
- comme esclave DP raccordé à un maître DP/M7 ou à un autre maître DP

### ***a- Interface (MPI)***

MPI (interface Multipoint) est l'interface de la CPU vers les PG/OP ou pour la communication avec plusieurs stations au sein d'un sous-réseau MPI

### ***b- Signalisation d'état***

La CPU comporte des LED de signalisation suivante :

- SF (rouge) : signalisation groupée de défauts.
- BATF (rouge) : défaut de la pile.
- DC5v (verte) : alimentation 5Vcc pour la CPU et le bus S7-300.
- FRCE (jaune) : forçage permanent.
- RUN (verte) : état de fonctionnement RUN.
- STOP (jaune) : état de fonctionnement STOP.

### ***c- Commutateur de mode***

Le commutateur de mode et les éléments d'affichage de toutes les CPU sont identiques. Leurs rôles et leurs fonctions sont également identiques. On définit ici les quatre positions principales de la CPU de S7-300.

- RUN-P (mode de fonctionnement RUN programme) : la CPU traite le programme utilisateur et la clé ne peut être retirée.
- RUN (mode de fonctionnement RUN) : la CPU traite le programme utilisateur. Dans cette position la clé peut être retirée.
- STOP (mode de fonctionnement STOP) : la CPU ne traite aucun programme utilisateur, la clé peut être retirée pour éviter le changement de mode inattendu mais on peut lire et écrire dans la CPU.
- MRES : effacement général.

L'arrêt est réalisé pour : STOP/MRES et la marche pour RUN/RUN-P

#### ***d- Carte mémoire***

Une carte mémoire conserve le contenu du programme en cas de coupure de courant, même en l'absence de pile.

#### ***e- La pile***

Elle permet de sauvegarder le contenu de la RAM en cas de coupure du courant.

### **2.2.3 Coupleur (IM)**

Les coupleurs peuvent utiliser un couplage sur de courtes distances. Pour un couplage sur de longues distances, il est recommandé d'émettre les signaux via le bus PROFIBUS.

### **2.2.4 Modules des signaux (SM)**

Il comporte plusieurs types tels que : STOR, ETOR, SANA, EANA ou E/SANA, E/STOR, ils ont comme fonction d'adaptation des niveaux de signaux entre le processus et le S7-300.

### **2.2.5 Module de fonction (FM)**

Il a pour rôle l'exécution de tâches de traitement des signaux du processus à temps critique et nécessitant une importante capacité mémoire comme le comptage, positionnement, la régulation.

### **2.2.6 Module de communication (CP)**

Il permet de établir des liaisons homme-machine et machine-homme qui sont effectuées à l'aide des interfaces de communication :

- Point à point (PPI)
- PROFIBUS
- Industriel Ethernet

## **2.3 Console de programmation (PG) ou PC avec logiciel STEP7**

Elle permet :

- D'écrire le programme, de le compiler et de le transférer vers l'automate.
- D'exécuter le programme pas à pas et de le visualiser.

- De forcer ou de modifier les données telles que les entrées, les sorties, les temporisations et les compteurs.

### **3 Fonctionnement de l'automate programmable**

L'automate lors de son fonctionnement exécute le programme cyclique, qui commence par l'acquisition des entrées issues des capteurs sur l'état du processus et finit par l'envoi des sorties aux actionneurs.

#### **3.1 Réception des informations sur les états du système**

Le S7-300 reçoit des informations sur l'état du processus via les capteurs de signaux reliés aux entrées.

Le S7-300 va mettre à jour la mémoire image des entrées au début de chaque cycle de programme, en transférant le nouveau état des signaux d'entrées des modules vers la mémoire image des entrées ce qui permet à la CPU de savoir l'état du processus.

#### **3.2 Système d'exploitation**

Le système d'exploitation contenu dans la CPU, organise toutes les fonctions et procédures dans la CPU qui ne sont pas liées à une tâche d'automatisation spécifique. Ce système gère :

- Le déroulement du démarrage et du redémarrage ;
- L'actualisation de la mémoire image des entrées et l'émission de la mémoire image des sorties ;
- L'appel du programme utilisateur ;
- L'enregistrement des alarmes et l'appel des OB d'alarmes ;
- La détection et le traitement d'erreurs ;
- La gestion des zones mémoire ;
- La communication avec des consoles de programmation et d'autres partenaires de communication.

#### **3.3 Exécution du programme utilisateur**

Après avoir acquis les informations d'entrée et exécuté le système d'exploitation, la CPU passe à l'exécution du programme utilisateur, qui contient la liste d'instructions à

exécuter pour faire fonctionner le processeur. Il est composé essentiellement de bloc de données de code et de bloc d'organisation.

### 3.4 Commande du processus

Pour commander le processus, on doit agir sur les actionneurs. Ces derniers reçoivent l'ordre via le module de sorties du S7-300. L'état des sorties est donc connu après l'exécution du programme utilisateur par la CPU, puis met à jour la mémoire image des sorties pour communiquer au processus le nouvel état.

## 4. Programmation de l'automate S7-300

La programmation des automates de la famille S7 se fait sur la console de programmation qui est actuellement le PC et sous environnement WINDOWS. Le langage de Programmation est le *STEP 7*.

Le logiciel STEP 7 est un outil de base pour la configuration et la programmation des systèmes d'automatisation SIMATIC. [5]

Le STEP 7 offre les fonctions suivantes pour l'automatisation d'une installation :

- Configuration et paramétrage du matériel et de la communication ;
- Création et la gestion de projet ;
- Gestion des mnémoniques ;
- Création de programmes pour systèmes cible ;
- Test de l'installation d'automatisation.

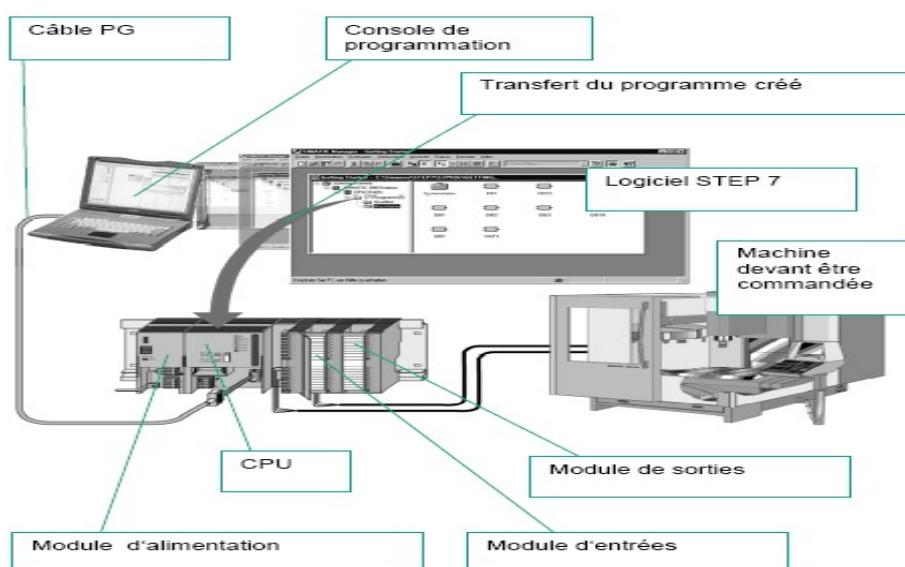


Figure IV.1: Vue d'ensemble de l'automatisme.

La programmation en STEP 7 présente trois modes possibles qui peuvent être combinés dans une même application :

1. le schéma logique (LOG) ;
2. le schéma contact (CONT) ;
3. liste d'instruction (LIST).

Chaque mode de représentation du programme a ses avantages mais aussi ses limites. Si les règles de programmation ont été respectées lors de la programmation, la compilation est possible dans les trois modes de représentation. Les programmes d'automatisation programmés en CONT ou LOG sont en principe toujours traduisible en LIST.

Dans la mémoire de programme de l'automate, le programme est toujours stocké en LIST (plus exactement en langage machine).

#### 4.1 Création d'un projet STEP 7

Pour créer un projet STEP 7 on dispose d'une certaine liberté d'action. En effet on a deux solutions possibles [5] :

- Solution 1 : commencer par la configuration matérielle ;
- Solution 2 : commencer par la création du programme.

Après la validation de la CPU, une fenêtre qui apparaît permet de choisir les modules, Il est recommandé de commencer par la configuration matérielle pour les installations qui contiennent beaucoup d'entrées et de sorties. L'application de la configuration matérielle de STEP 7 présente l'avantage de la sélection automatique des adresses.

Si on commence par la création du programme, il faudra rechercher les adresses en fonction des constituants choisis, dans ce cas on ne bénéficie pas de la fonction d'adressage automatique de STEP 7.

Les procédures qui vont nous permettre la création du projet sous le logiciel STEP 7 sont comme suit :

1. double cliquer sur l'icône SIMATIC Manager. Ceci lance l'assistant de STEP 7
2. la fenêtre suivante permet la création d'un nouveau projet.
3. on clique sur la fenêtre suivante, ce qui permet de choisir la CPU.



4. insérer, et choisir le langage de programmation (LIST, CONT et LOG)
5. en cliquant sur suivant, une dernière fenêtre pour la création du projet apparaît pour le nommer
6. on clique sur créer, la fenêtre suivante apparaît

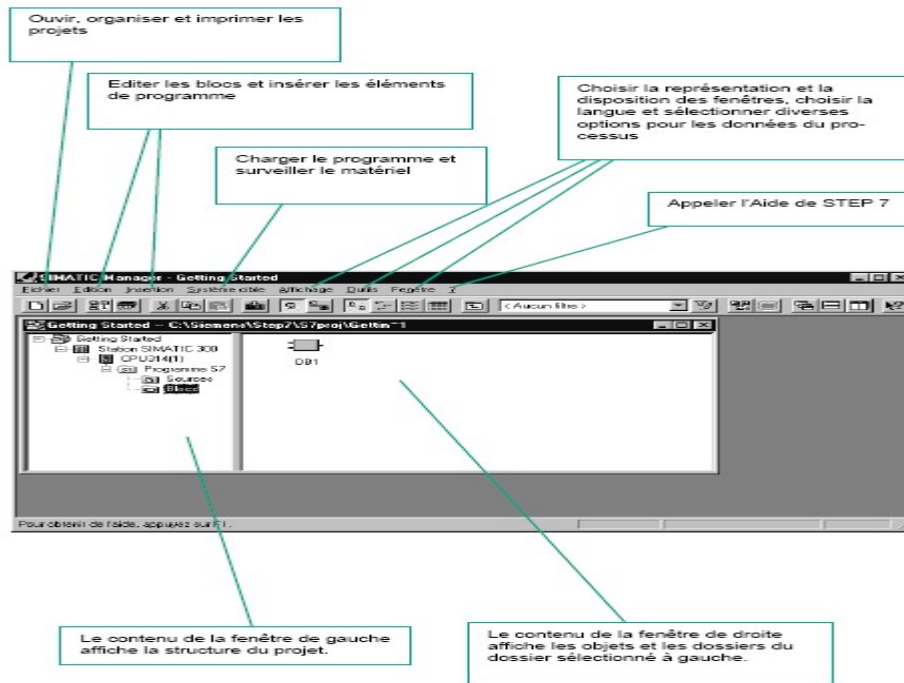


Figure IV.2 : vue des composants d'un projet s7.

## 4.2 Adressage des modules du S7-300

On a deux types d'adressage :

### 4.2.1 Adressage des modules lié à l'emplacement

Dans ce cas d'adressage (adressage par défaut), une adresse de début de module est affectée à chaque numéro d'emplacement sur le profil support (châssis).

### 4.2.2 Adressage absolu des modules de signaux

Ce type d'adressage est nécessaire pour adresser les voies des modules de signaux dans le programme utilisateur. Chaque sortie possède une adresse absolue déterminée par la configuration matérielle.

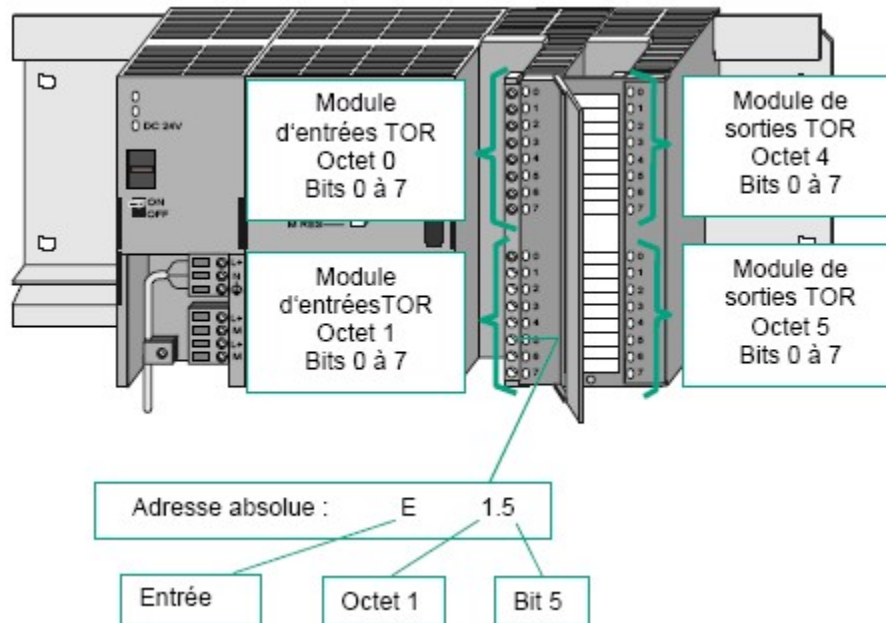
#### a- Adressage des modules TOR

L'adressage d'une entrée ou d'une sortie est constitué d'une adresse d'octet et d'une adresse de bit.

- L'adressage d'octet dépend de l'adresse de début de module.

- L'adressage de bit est indiqué sur le module

La figure IV.3 montre un exemple d'adressage absolu ou direct.



**Figure IV.3** : Exemple d'adressage absolu d'un module TOR.

#### b- Adressage des modules analogiques

L'adressage d'une voie d'entrée ou sortie analogique est toujours une adresse de mot. L'adressage de la voie dépend de l'adresse de début de module. Si le premier module analogique occupe l'emplacement 4, l'adresse de début de chaque module analogique suivant est incrémentée de 16 par emplacement. Les voies d'entrée et de sortie analogique d'un module d'entrées/sorties analogique ont la même adresse de début.

#### 4.2.3 Mémentos

Les mémentos sont utilisés pour les opérations internes à l'automate pour lesquelles l'émission d'un signal n'est pas nécessaire. Les mémentos sont des bistables servant à mémoriser les états logique "0" ou "1".

Chaque automate programmable dispose d'un grand nombre de mémentos (S7-300 dispose de 2048 bits de mémentos). En cas de panne de la tension de service, le contenu sauvegardé des mémentos sera perdu.

### 4.3 Traitement du programme par l'automate

La CPU traite le programme d'une manière cyclique en plusieurs phases :

**Phase (1) :** le système d'exploitation démarre la surveillance du temps de cycle.

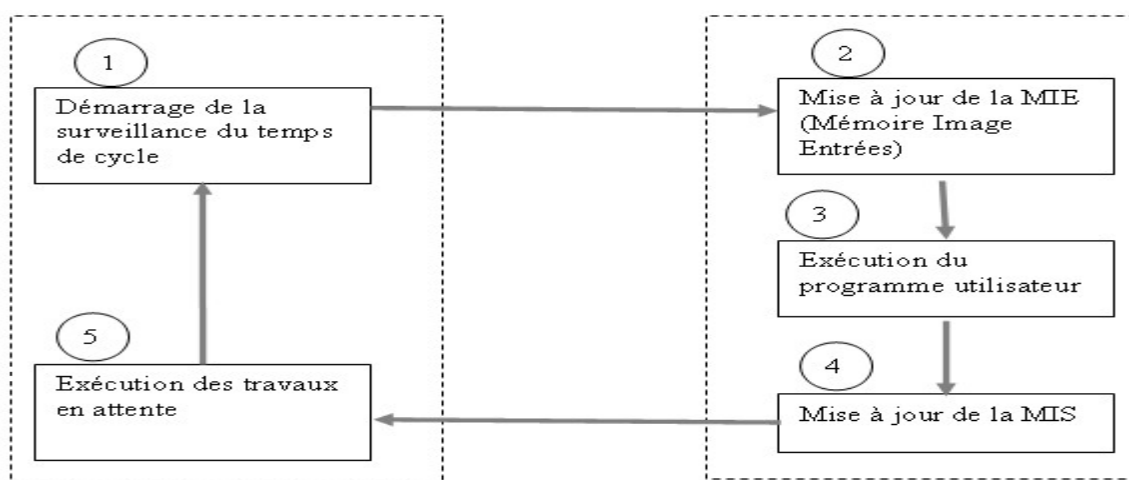
**Phase (2) :** la CPU lit l'état des entrées dans les modules d'entrée et met à jour la mémoire image des entrées.

**Phase (3) :** à cette étape la CPU exécute les instructions du programme utilisateur.

**Phase (4) :** la CPU écrit les résultats dans la mémoire image des sorties, puis elle transfère ces derniers vers les modules de sorties.

**Phase (5) :** à la fin du cycle, le système d'exploitation exécute les travaux en attente, tels que le chargement et l'effacement des blocs, ou la réception de données globales.

**Phase (6) :** la CPU revient alors au début du cycle et démarre à nouveau la surveillance du temps de cycle, un schéma illustratif est donné par la figure IV.4



**Figure IV.4 :** Exécution cyclique du programme dans l'API.

## 5 Principe de conception d'une structure de programme

Pour chaque solution d'automatisation, la CPU exécute deux types de programme différents, le système d'exploitation et le programme utilisateur.

### 5.1 Rôle du système d'exploitation

Le système d'exploitation, contenu dans la CPU, organise toutes les fonctions et procédures qui ne sont pas liées à une tâche d'automatisation spécifique, ces tâches sont les suivantes :

- Le déroulement du démarrage et du redémarrage ;
- L'actualisation de la mémoire image des entrées et l'émission de la mémoire image des sorties ;
- L'appel du programme utilisateur ;
- L'enregistrement des alarmes et l'appel des OB d'alarmes ;
- La détection et le traitement d'erreurs ;
- La gestion des zones de mémoire ;
- La communication avec des consoles de programmation et d'autres partenaires de communication (interface).

## 5.2 Rôle du programme utilisateur

Parmi les tâches d'automatisation, on trouve la création du programme utilisateur et son chargement dans la CPU, ce programme contient toutes les fonctions nécessaires au traitement d'une tâche d'automatisation, et il doit prendre en charge :

- La détermination des conditions pour la démarrage et le redémarrage de la CPU (ex : initialisation des signaux) ;
- La réaction aux alarmes ;
- Le traitement des données du processus ;
- Le traitement des perturbations dans le déroulement normal du programme.

## 5.3 Blocs du programme utilisateur

Il faut avoir l'aptitude à subdiviser le procédé à automatiser en différentes tâches. Les parties d'un programme utilisateur structuré correspondant à ces différentes tâches sont les blocs de programme. [2]

Le STEP7 offre la possibilité de structurer le programme utilisateur, c'est-à-dire le subdiviser en différentes parties autonomes qui donnent les avantages suivants :

- Ecrire des programmes importants et clairs ;
- Standardiser certaines parties du programme ;
- Simplification de l'organisation du programme ;
- Simplifier le test du programme, car on peut l'exécuter section par section ;
- Faciliter la mise en service.

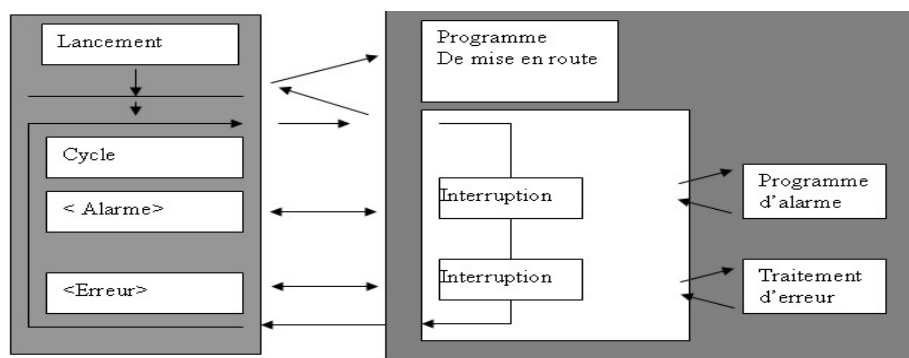
Le logiciel de base STEP7 dans ses différents langages de programmation possède un nombre important de blocs d'utilisateurs, destiné à structurer le programme utilisateur.

### 5.3.1. Bloc d'organisation (OB)

Un OB est une structure importante d'un programme utilisateur, il est appelé cycliquement par le système d'exploitation. Il gère le traitement du programme cyclique et déclenche les alarmes, ainsi que le comportement de la mise en route de l'automate programmable et le traitement des erreurs. Nous pouvons programmer les blocs d'organisation et déterminer ainsi le comportement de la CPU. En résumé, l'OB constitue l'interface entre le système d'exploitation et le programme utilisateur.

Les OB définissent l'ordre dans lequel les différentes parties du programme sont traitées.

L'exécution d'un OB peut être interrompue par l'appel d'un autre OB. Cette interruption se fait selon une priorité. Une illustration est donnée à la figure (III.9). [2]



**Figure IV.5 :** Traitement du programme avec possibilité d'interruption.

### 5.3.2 Bloc fonctionnel (FB)

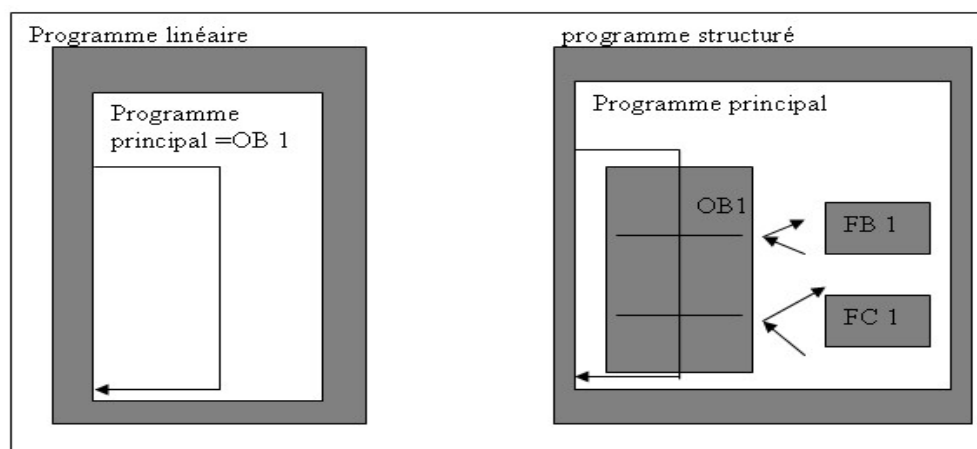
Un bloc fonctionnel contient un programme qui est exécuté dès son appel par un autre bloc de code (OB). Il facilite la programmation de fonction complexe, comme la commande de moteur (rampe d'accélération,....etc.).

## 5.4 Fonction (FC)

Les fonctions font partie des opérations que le concepteur programme. Elles ne possèdent pas de mémoires. Les variables temporaires d'une fonction sont sauvegardées dans la pile de données locale. Ces données sont perdues après exécution de la fonction. Les

fonctions peuvent faire appel à des blocs de données globaux pour la sauvegarde de données.

[2]



**Figure IV.6** : schéma illustrant le traitement du programme par la CPU.

Une fonction contient un programme qui est exécuté lorsqu'elle est appelée par un autre bloc de code. Elle peut être utilisée pour :

- Renvoyer une valeur de fonction au bloc appelant (exp : fonction mathématique) ;
- Exécuter une fonction technologique (exp : commande individuelle avec combinaison)

### **Bloc de données (DB)**

Les DB sont utilisés pour la mise à disposition de l'espace mémoire pour des variables de type données, on a deux types de bloc.

Tout les FB, FC, OB peuvent lire les données contenues dans un DB global ou écrire des données dans un DB global. Ces données sont conservées dans le bloc de données même lorsqu'on le quitte.

## **6 Traitement du programme par la CPU**

Selon la manière de programmation choisie pour le problème d'automatisation, le traitement du programme par la CPU peut être :

### **6.1 Linéaire**

La CPU exécute le cycle habituel en appelant le bloc OB1 dans le programme principal une fois par boucle. Ce type de traitement est requis lors de la commande de procédés simples exigeant des programmes de tailles et de complexité réduite.

## 6.2 Structuré

Dans le cas d'un procédé complexe possédant plusieurs fonctions (postes), il est recommandé de partager le programme utilisateur en fonctions principales que l'on programme à l'aide des blocs de codes (OB, FB, FC) d'une façon à faciliter la programmation et le test des parties de programme ainsi que sa compréhension et son diagnostic.

Dans ce cas la CPU exécute le programme principal dans OB1 et fait appel aux autres blocs (FC, FB).

## 6.3 Quelques blocs d'organisation des S7-300

Les S7-300 présentent un effectif important d'OB, ils constituent une interface entre le système d'exploitation de la CPU et le programme utilisateur.

### 6.3.1 Bloc d'organisation pour le traitement cyclique (OB1)

Il constitue l'infrastructure du programme utilisateur, il est exécuté d'une manière cyclique par le système d'exploitation.

### 6.3.2 Bloc d'organisation pour le traitement des erreurs

Les erreurs que les CPU S7 détectent et auxquelles elles peuvent réagir à l'aide de blocs d'organisation sont classables en deux catégories.

- Erreur synchrone : elle apparaît pendant le traitement d'une opération dans le programme utilisateur, exp : OB80 à OB87 ;
- Erreur asynchrone : elle ne peut être associée directement au programme utilisateur, il s'agit d'erreur de classe de priorité ou d'erreur dans l'automate (exp. Module défaillant, OB121, OB122).

### 6.3.3 Bloc d'organisation de programme déclenché par alarme

Le STEP7 met à la disposition du programmeur différents types d'OB qui interrompent le traitement de l'OB1, à des intervalles de temps précis ou à des apparitions d'événements donnés, on peut citer :

- Blocs d'alarmes horaires OB10 à OB17

- Blocs d'alarmes temporisées OB20 à OB23
- Blocs d'alarmes cycliques OB30 à OB47
- Blocs d'alarmes de processus OB40 à OB47
- Bloc d'alarme multiprocesseur OB 60.

## 7 Conclusion

Au terme de ce chapitre nous concluons que l'automate de la gamme S7-300 est un API parfait même pour l'automatisation partielle d'une installation, vu les avantages suivants :

- sa modularité lui permet de réaliser diverses fonctions, mais aussi la simplicité de câblage avec une énorme possibilité d'exploitation ;
- son langage de programmation STEP7 qui peut être représenté en trois modes, offre les possibilités suivantes :
  - configuration et paramétrage du matériel ;
  - Test, facilité de mise en service et maintenance de l'installation ;
  - Modification du programme facile à effectuer par rapport à une logique câblée.
- une console de programmation qui permet non seulement d'écrire le programme, et le compiler et de le transférer à l'automate mais aussi de l'exécuter pas à pas et de visualiser l'état des entrées sorties.

# 1 programmation du système

Après avoir modélisé notre système par un GRAFCET, on est passé à sa programmation qui débute par l'affectation des entrées et les sorties.

## 1.1 Affectation des entrées et des sorties

### 1.1.1 Affectation des Entrées

Entrées	Notions	Natures
E 0.0	INIT	Bouton Poussoir
E 0.1	AUDur	//
E 0.2	AUDoux1	//
E 0.3	AUDoux i	//
E 0.4	Str	//
E 1.0	Rm1	Relais
E 1.1	Rm2	//
E 1.3	Cm	Capteur Magnéto inductif
E 1.4	Cfrg	//
E 1.5	Cfrd	//
E 1.6	Cfb	Capteur de fin de course à contact
E 1.7	Cfh	//
E 1.2	Cap P	Capteur de pression
E 2.0	Cfa	Capteur de fin de course à contact
E 2.1	Cfr	//
E 2.2	fC1	Capteur photocellule
E 2.3	FC2	//
E 2.4	FC3	//
E 2.5	FC4	//
E 2.6	FC5	//

## 1.1.2 Affectation des sorties

Sorties	Notions	Désignations
A 2.0	Mot 1.	Excitation du moteur 1.
A 2.1	Mot 2.	Excitation du moteur 2.
A 2.2	M1AV.	Rotation du moteur 1 (Avance du tapis 1).
A 2.3	M2AV.	Rotation du moteur 2(Avance du tapis 2).
A 2.4	M1ART	Arrêt du moteur 1 (Fin D'avance du tapis 1).
A 2.5	M2ART	Arrêt du moteur 2 (Fin d'avance du tapis 2).
A 2.6	ICmp	Incrémentation du Compteur
A 2.7	PRég	Position régulière de l'armoire.
A 3.0	V <sub>p-n</sub>	Entrée du vérin du bras de positionnement
A 3.1	A1m	Alarme
A 3.2	V <sub>v_p</sub>	Descente du vérin vertical.
A 3.3	V <sub>v_n</sub>	Montée du vérin vertical (Bloc porte ventouses)
A 3.4	Arm Dét.	Détection de l'armoire
A 3.5	Bne Adh.	Bonne adhésion (armoires/Ventouses)
A 3.6	Asp	Aspiration de l'armoire par les ventouses.
A3.7	Arm Lib	Libération de l'armoire
A 4.0	V <sub>h_n</sub>	Recul du vérin horizontal (chariot)
A 4.1	V <sub>h_p</sub>	Avance du vérin horizontal (chariot)
A 4.2	V <sub>r_p</sub>	Rotation à gauche de 90° du vérin rotatif (Bloc porte ventouses).
A 4.3	V <sub>r_n</sub>	Rotation à droite de 90° du vérin rotatif (Bloc porte ventouses).

## 1.2 Elaboration du programme

(Voir l'annexe)

## 2 Simulations

Après l'élaboration du programme de l'automate, nous arrivons à l'étape décisive du travail effectué. Cette étape est la validation du programme par simulation et la vérification du bon fonctionnement de notre système avec la nouvelle solution programmable développée.

Pour cela, nous avons utilisé le logiciel S7-PLCSIM qui est le logiciel optionnel de STEP 7 et son installation requiert l'installation de STEP 7 au préalable.

L'application de ce logiciel est le test des programmes sous STEP 7 pour les automates S7-300 et S7-400 qu'on ne peut tester immédiatement sur l'installation et ceci pour différentes raisons telles que :

Lorsque on ne dispose pas d'un automate.

L'application est critique, car elle peut occasionner des dommages matériels ou blessures corporelles en cas d'erreurs de programmation mais la simulation permet de supprimer ces erreurs pendant le test de simulation.

### 2.1 Présentation du S7-PLCSIM

L'application de simulation de modules S7-PLCSIM nous permet d'exécuter et de tester le programme dans un automate programmable (API) que nous simulons dans un ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux). L'API S7 de simulation nous permet de tester des programmes destinés aux CPU S7-300 et aux CPU S7-400, et ce afin de remédier à d'éventuelles erreurs.

Le S7-PLCSIM dispose d'une interface simple qui nous permet de visualiser et de forcer les différents paramètres utilisés par le programme (comme, par exemple, d'activer ou désactiver des entrées). Tout en exécutant le programme dans l'API de simulation, nous avons également la possibilité de mettre en œuvre les diverses applications du logiciel STEP 7 comme, par exemple, le test de bloc afin de visualiser les variables d'entrées et de sorties.[2]

### 2.1.1 États de fonctionnement de la CPU

#### *État de marche (RUN-P)*

La CPU exécute le programme tout en nous permettant de le modifier, de même que ses paramètres. Afin de pouvoir utiliser les application de STEP 7 pour forcer un paramètre quelconque du programme durant son exécution, nous devons mettre la CPU à l'état RUN-P.

Nous avons toujours la possibilité d'utiliser les "fenêtres" créées dans l'application de simulation de modules S7-PLCSIM pour modifier une donnée quelconque utilisée par le programme.

#### *État de marche (RUN)*

La CPU exécute le programme en lisant les entrées, exécutant le programme, puis en actualisant les sorties. Lorsque la CPU se trouve à l'état de marche (RUN), nous ne pouvons ni charger un programme, ni utiliser des applications de STEP 7 pour forcer un paramètre quelconque (comme les valeurs d'entrées). Nous pouvons uniquement utiliser les fenêtres créées dans l'application de simulation des modules S7-PLCSIM pour modifier une donnée quelconque utilisée par le programme.

#### *État d'arrêt (STOP)*

La CPU n'exécute pas le programme. Contrairement à l'état d'arrêt (STOP) des CPU réelles, les sorties ne prennent pas de valeurs (de "sécurité") prédéfinies, mais conservent les états dans lesquels elles étaient lorsque la CPU est passée à l'état d'arrêt (STOP). Nous pouvons charger des programmes dans la CPU lorsqu'elle est à l'arrêt. Le passage à l'arrêt (STOP) à celui de marche (RUN) démarre l'exécution du programme à partir de la première opération.

#### *Effacement général de la CPU*


Les états de fonctionnement de la CPU, les indicateurs de la CPU et le bouton MRES (effacement général) sont tous représentés dans la fenêtre "CPU". Nous pouvons définir l'état de fonctionnement de la CPU avec la commande position du commutateur à clé. Nous pouvons suspendre l'exécution du programme de l'API de simulation lorsque la CPU est à l'état RUN ou RUN-P.

### 2.1.2 Mise en route du logiciel S7-PLCSIM






Le mode simulation est disponible à partir du gestionnaire de projet SIMATIC Manager à condition qu'aucune liaison à des API réels ne soit établie. On peut suivre la procédure suivante pour la mise en route du logiciel S7-PLCSIM.[7]

La procédure à suivre est :

1)- Ouvrir le gestionnaire de projet SIMATIC Manager ;

2)- Cliquez sur  ou sélectionnez la commande *Outils > Simulation de modules*

Cela lance l'application S7-PLCSIM et ouvre une fenêtre CPU ;

- 3)- Dans le gestionnaire de projets SIMATIC Manager, chercher le projet ;
- 4)- Dans le projet-, chercher le dossier Blocs (se reporter à l'aide en ligne du gestionnaire de projets SIMATIC Manager pour une présentation des objets de STEP7) ;
- 5)- Dans le gestionnaire de projets SIMATIC Manager, cliquer sur  ou choisir la commande *Système cible > Charger* pour charger le dossier Blocs dans l'API de simulation ;
- 6)- Dans l'application S7-PLCSIM on crée de nouvelles fenêtres pour visualiser les informations provenant de l'API de simulation :
  - Cliquer sur  ou sélectionner la commande *Insertion>Entrée*. La fenêtre affiche *EB0* (Octet d'entrée 0) avec la possibilité de modifier le numéro d'octet à condition de respecter la configuration matérielle déclarée ;
  - Cliquer sur  ou sélectionner la commande *Insertion>Sortie* pour afficher une seconde fenêtre, *AB0* (octet de sortie 0) avec la possibilité de modifier le numéro d'octet à condition de respecter la configuration matérielle déclarée.
  - Cliquer sur  ou sélectionner la commande *Insertion>Temporisation* pour afficher trois fenêtres de temporisations. Taper 1, 2, 3 et 4 (pour les temporisation T1, T2, T3 et T4) dans les fenêtres respectives, en appuyant sur la touche Entrée après chaque valeur ;
- 7)- Choisir le menu *CPU* dans S7-PLCSIM et vérifier que la commande *Mettre sous tension* est activée (•) ;
- 8)- choisir la commande *Exécution>Mode* d'exécution et vérifier que la commande *Cycle continu* est activée (•) ;
- 9)- Mettre la CPU de simulation en marche en cliquant sur l'une des cases à cocher RUN ou RUN-P ;
- 10)- Cliquer sur le bit 0 de EB0 pour simuler la mise à 1 de l'entrée 0.0 et observez la réaction des temporisations et de AB0 ;
- 11)- Cliquer sur  ou sélectionner la commande *Fichier>Enregistrer CPU sous...* pour enregistrer l'état actuel de l'AP de simulation dans un nouveau fichier.

Une fois que toutes les fenêtres d'entrée et de sorties sont prêtes, on active les entrées voulues pour lire l'état des sorties.

### 2.1.3 Visualisation l'état du programme

Après le chargement du programme dans la CPU du simulation et la mise de cette dernière en mode "RUN", le STEP7 nous permet de visualiser l'état du programme soit en cliquant sur l'icône ou



on sélectionne la commande *Test>Visualiser*.

## 2.2 Simulation du programme élaboré

C'est l'étape décisive du travail. La simulation du programme grâce au logiciel S7-PLCSIM, nous a permis de vérifier le bon fonctionnement (fiable et sûr) du système et de valider notre solution programmée.

## Conclusion

Dans le cas où on ne disposerait pas d'un automate réel, le logiciel S7-PLCSIM devient l'outil indispensable à la validation de la solution programmée et des concepts de commande automatisés avant leur implémentation sur le système réel.

Le logiciel S7-PLCSIM nous a permis de visualiser le comportement des sorties et simuler le programme du procédé avec succès.

A l'issue de mon modeste travail avec la coordination de mon enseignant et des techniciens de l'ENIEM je viens à constater la grande utilité des automatismes dans les systèmes de productions (la productique) à savoir l'introduction des automates programmables industriels dans leurs différentes gammes et différents constructeurs tels que Siemens, Schneider Electric et Allen Bradley. L'entreprise a opté pour le premier.

Aussi, la solution d'automatisation proposée pour chaque système passe inévitablement par une étape de modélisation du processus. Dans notre cas présent, puisque il s'agit d'un système à évolution séquentielle, l'outil de modélisation utilisé est le GRAFCET : un outil graphique représentant le fonctionnement global d'une manière très simple.

Vient ensuite la phase de programmation dans un logiciel d'automatisme qui est Step7, ce logiciel permet l'écriture de programmes simples et facilement lisibles par l'automaticien ou le technicien de maintenance pour le diagnostic. Il offre également trois manières d'écriture, à savoir le langage à contact proche de l'électricien, le logigramme en se servant des portes logiques et le List par liste d'instructions comme pour le cas de l'assembleur et le Pascal.

## BIBLIOGRAPHIE

[1] **K. Nordine, SACI Farhat** Mémoire de fin d'étude « étude de l'automatisation des quatre premiers postes de la chaîne de fabrication de portes de réfrigérateurs à l'ENIEM » université Mouloud Mammeri de Tizi-Ouzou promotion 2001/2002.

[2] **M. Rachid, C. Madjid** Mémoire de fin d'étude « développement d'une solution programmable d'un équipement de transfert à l'aide d'un API S7-300 à l'unité froid de l'ENIEM » université Mouloud Mammeri de Tizi-Ouzou promotion 2006/2007.

[3] **www.ENIEM-dz.com**

[4] Cours Capteurs et Actionneurs Master SIS.1, Michel Bensoam Université Paul Cézanne Aix Marseille III janvier 2005

[5] **SIMATIC S7-GRAPH** pour S7-300/400 Programmation de commandes séquentielles Getting started promotion 2002/2004