

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

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D' INFORMATIQUE  
DEPARTEMENT D'AUTOMATIQUE

## **Mémoire de Fin d'Etudes de MASTER PROFESSIONNEL**

Domaine : Sciences et Technologies

Filière : Génie électrique

Spécialité : **Automatique et informatique  
industrielles**

*Présenté par*

**Merzouk KHALDI**

**Aissa BOUSTA**

Thème

## **Conception, modélisation et supervision d'un charriot pour une enrouleuse installé à l'unité froid de l'ENIEM**

*Mémoire soutenu publiquement le 10/07/2017. devant le jury composé de :*

**M. KASRI.**

MCB UMMTO, Président

**M.CHELLI**

MCB, UMMTO, Encadreur

**A.SI SALEM**

ENIEM, Co-Encadreur

**M.LAKHLEF**

MAA, UMMTO, Examineur

# Remerciements

*Nous remercions avant tout le bon dieu de nous avoir donné la santé, le courage et la volonté pour finir ce travail.*

*Nous tenons à remercier notre promoteur Mr CHELLI pour son aide, le temps qu'il nous a consacré et ses orientations et surtout pour sa patience tout au long de ce travail.*

*Nous remercions également notre co-promoteur Mr SI SALEM « chef du département maintenance », qui nous a orienté et facilité notre intégration dans le domaine industriel et qui nous a accueillis avec beaucoup de gentillesse et de patience.*

*Nous sommes aussi reconnaissants à tous les enseignants qui ont contribué à notre réussite.*

*Nous remercions également les membres de jury qui feront l'honneur de juger notre travail, d'apporter leurs réflexions et suggestions scientifiques.*

*Nos remerciements les plus chaleureux s'adressent à nos familles et surtout nos parents qui sont la source de cette réussite et qui nous ont soutenu et encourager pour aller au bout de ce travail.*

*Nos derniers remerciements s'adressent à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.*

# DEDICACE

*Je dédie ce modeste travail*

*A mes très chers parents.*

*A ma fiancé et ces parents.*

*A mes très chers frères .*

*A ma soeur.*

*A la mémoire de ma grand-mère.*

*A la mémoire de mes grands-pères.*

*A mon binôme.*

*A tout mes amis.*

*A tous ceux qui ont su croire en moi.*

*A tous ceux qui me sont chers.*

*Aïssa*

# DEDICACE

*Je dédie ce modeste travail*

*A mes très chers parents.*

*A mes très chers frères .*

*A mes soeurs.*

*A la mémoire de ma grand-mère.*

*A la mémoire de mes grands-pères.*

*A mon binôme.*

*A tout mes amis.*

*A tous ceux qui ont su croire en moi.*

*A tous ceux qui me sont chers.*

*Merzouk*

# Sommaire

<b>Introduction générale .....</b>	<b>1</b>
------------------------------------	----------

## **Chapitre I : Automatisation des systèmes Industriels**

.1. Introduction .....	3
2. Objectif de l'automatisation d'un système de production .....	4
3. Structure d'un système automatisé .....	4
4. Le rôle d'un automate dans les systèmes automatisés industriels .....	6
5. Conclusion .....	12

## **Chapitre II : Etude et présentation fonctionnelle du système**

1. Introduction .....	13
2. Description de l'enrouleuse .....	15
3. Constitution de la partie opérative de l'enrouleuse .....	16
4. Partie commande du système .....	31
5. Conclusion .....	33

## **Chapitre III : Modélisation du système à l'aide du GRAFCET**

1. Introduction .....	34
2. Définition .....	34
3. les concepts de base d'un GRAFCET .....	34
4. Niveau d'un grafcet .....	40
5. Modélisation par l'outil grafcet .....	41
6. Conclusion .....	45

## **Chapitre IV : Développement de la solution programmable**

1. Introduction .....	46
2. Définition d'un automate programmable industriel (API) .....	46
3. Objectifs de l'automate dans les systèmes automatisés industriels .....	46

4. choix d'un automate programmable industriel .....	48
5. Présentation générale de l'automate S7-300 .....	49
6. Caractéristique de l'automate S7-300.....	50
7. Constitution de l'automate S7-300 .....	50
8. Programmation de l'automate S7-300 .....	50
9. Test et simulation du programme .....	62
10. Conclusion .....	65

## **Chapitre V : Supervision du système**

1. Introduction .....	66
2. Présentation du logiciel Win CC flexible 2008.....	66
3. Définition de la supervision industrielle .....	66
4. constitution d'un système de supervision .....	67
5. Etapes de mise en œuvre .....	68
6. Création d'un projet sous WinCC flexible 2008.....	69
7. Les vues du projet .....	72
8. compilation et simulation .....	74
9.Conclusion .....	75
<b>Conclusion générale</b> .....	76

## **Bibliographie**

## **Annexes**

## Introduction Générale

Devant la compétitivité croissante et de plus en plus dure que subissent les entreprises industrielles, dans des domaines très importants comme la mécanique l'informatique et l'automatisation..., les industries doivent garder leur outil de production performant et fiable, car cela reflète parfaitement leurs aptitudes à affronter le marché mondial.

L'évolution de l'informatique, en particulier, dans l'industrie, a permis la progression de l'automatisation des systèmes de production. D'où l'avènement des API (Automates Programmables Industriels) qui jouent un rôle très important dans l'amélioration des procédés de production et leur sécurité.

L'automate programmable industriel (API) est aujourd'hui, le constituant le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie, car il répond à tous les besoins d'adaptation et de flexibilité pour un grand nombre d'opérations. Cette émergence est due en grande partie, à la puissance de son environnement de développement et aux larges possibilités d'interconnexions.

Le SIMATIC constitue une vaste plate forme d'automatisation offrant des solutions à des problèmes complexes pour tous les secteurs d'activité. Le logiciel STEP7 a été conçu dans un souci d'homogénéité et de complémentarité avec un système de contrôle et de commande, offrant des fonctions conviviales de conduite et de simulation du processus, ce qui simplifie d'une manière considérable la mise en œuvre de nombreuses caractéristiques du système de commande, notamment la gestion de base de données communes.

Notre projet de fin d'étude, s'est déroulé au sein l'Entreprise Nationale des Industries Electroménagères (ENIEM) qui est une entreprise publique de droit algérien constituée le 2 janvier 1983 mais qui existe depuis 1974 sous tutelle de l'entreprise SONELEC.

Son siège se situe au chef lieu de la wilaya de Tizi-Ouzou. Les unités de production, froid, cuisson et, climatisation sont implantées dans la zone industrielle Aissat Idir à Oued Aissi, distant de 7km du chef lieu de la wilaya. Les filiale sanitaire et lampe sont installées respectivement à Miliana de la wilaya de Ain-Defla et Mohamadia, wilaya de Mascara.

Et comme toute autre entreprise l'ENIEM se confronte aux contraintes de progrès technologique par son rythme accélérer, actuellement elle pense a remplacé ses anciens séquenceurs par de nouveaux séquenceurs plus fiable, facile à programmer et qui offrent des gammes diversifiées de la CPU ...etc.

Pour notre projet de fin d'étude, les responsables du département maintenance de l'unité froid de l'ENIEM, nous ont confié la modernisation de la commande d'une machine enrouleuse tube, en ajoutant un charriot transporteur de cuve ainsi la sécurité de la station.

Pour ce faire nous avons élaboré le plan de travail suivant :

- Le premier chapitre comprend une vue globale sur l'automatisation des systèmes industriels.
- Le deuxième chapitre consacré à l'étude et la présentation fonctionnelle du charriot de l'enrouleuse, où sont présentées ses différentes parties constitutives.
- Le troisième chapitre est consacré à la modélisation du système en faisant appel à l'outil de modélisation qui est le GRAFCET (Graphe Fonctionnel de Commande Etape-Transition).
- Le quatrième chapitre consistera en développement d'une solution programmable.
- Le cinquième chapitre traite la simulation du système avec Win CC flexible.

ET nous terminerons notre travail par une conclusion générale.



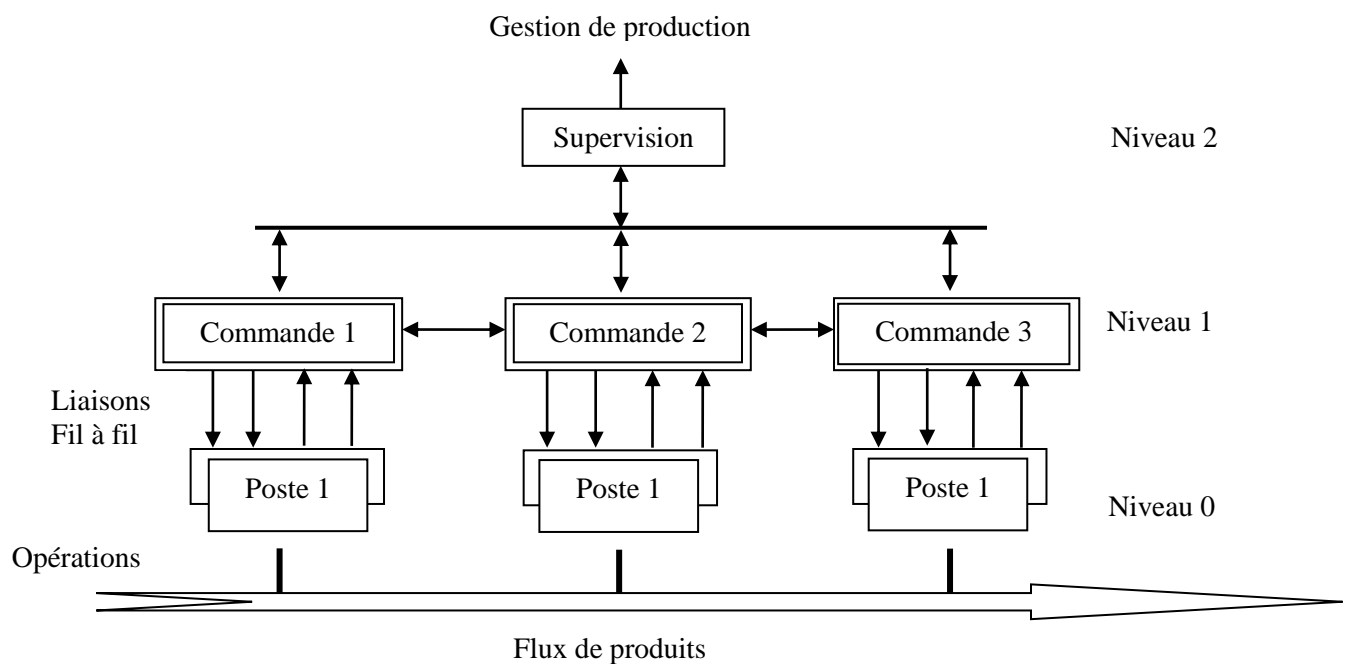
# Chapitre I

## Automatisation des Système Industriels

## I.1. Introduction

Un système automatisé de production (SAP) est un système de gestion de production qui reçoit un flux de matière ou de produits et génère un flux de produits plus élaborés (moulés, usinés, assemblés, testés...etc.). Il doit aussi gérer l'alimentation en énergie, ainsi que des flux auxiliaires tels les consommables et les déchets. Tout cela, ajouté à des exigences sans cesse accrues de qualité, qui entraîne un accroissement des besoins, en particulier la manipulation d'un grand nombre de variables et la gestion de véritables flux de communication. Cela explique que les systèmes câblés deviennent trop volumineux, moins fiable et trop rigides pour de telles applications, et que l'on se tourne donc vers des solutions utilisant les techniques de traitement de l'information par processeurs programmables.

La figure. I.1 illustre le fonctionnement d'un système automatisé de production, tel que l'on en rencontre notamment, mais pas exclusivement, dans l'industrie manufacturière



**Figure I.1 :** Système Automatisé de Production.

## **I.2. Objectif de l'automatisation d'un système de production**

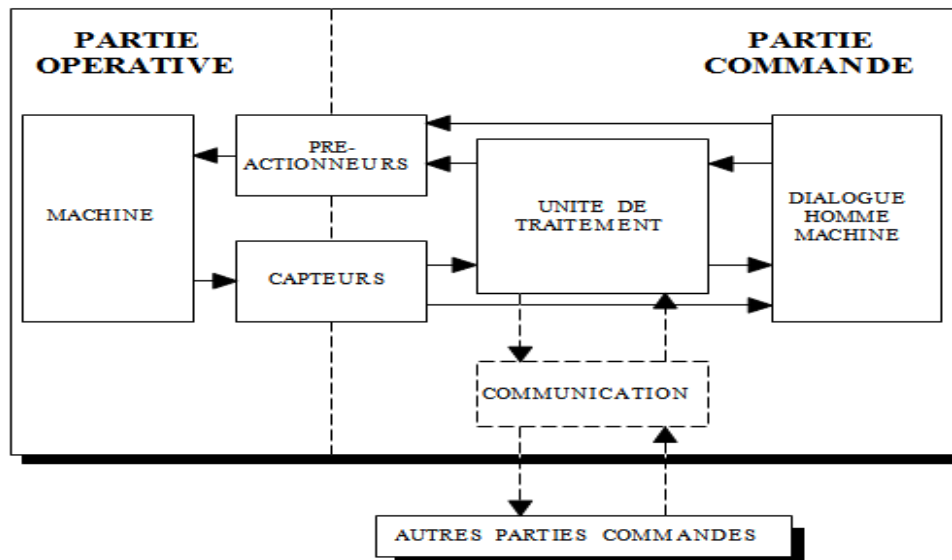
L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée du système. Ces éléments sont exprimables en termes d'objectifs par :

- Accroître la productivité du système, c'est-à-dire augmenter la quantité de produits élaborés pendant une durée donnée. Cet accroissement de productivité exprime un gain de valeur ajoutée sous forme :
  - d'une meilleure rentabilité,
  - d'une meilleure compétitivité (par rapport à la concurrence) .
  - améliorer la qualité du produit grâce à la précision.
- S'adapter à des contextes particuliers :
  - adaptation à des environnements hostiles pour l'homme (milieu salin, spatial, nucléaire...).
  - adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges).
  - augmenter la sécurité.

## **I.3. Structure d'un système automatisé**

Tout système automatisé comporte :

- Une partie Opérative (P.O.).
- Une partie Commande (P.C.).



**Figure. I.2 :** Structure d'un système automatisé industriel

### I.3.1. La partie opérative (OP)

C'est la partie visible du système. Elle comporte les éléments mécaniques du système on trouve :

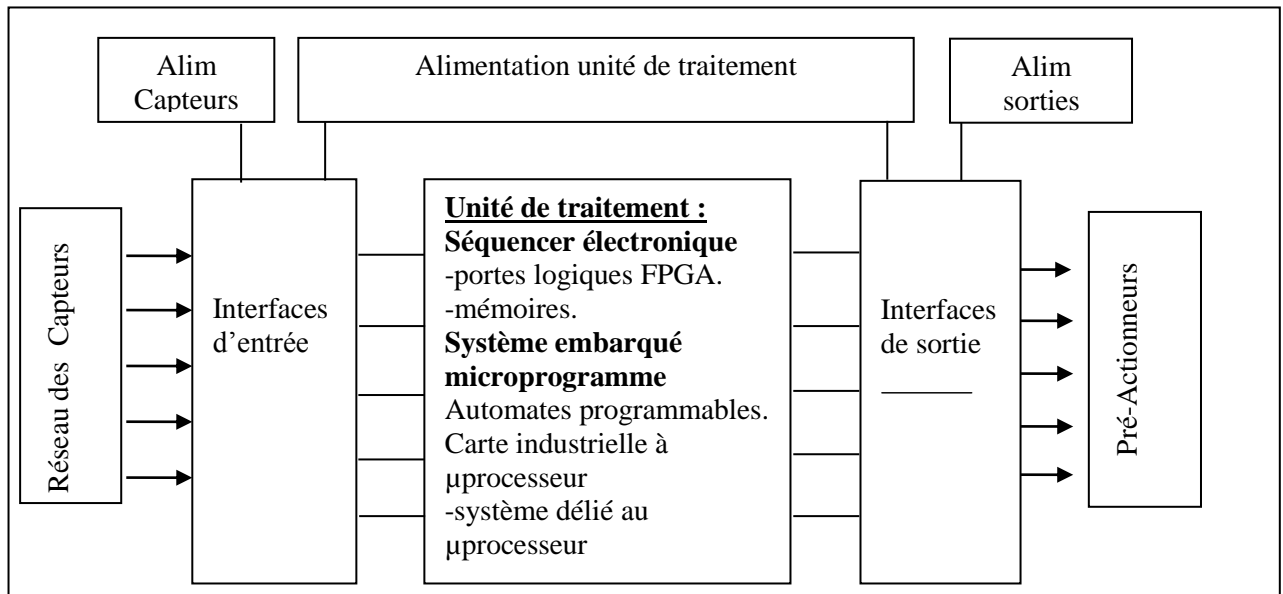
- ✓ des pré-actionneurs (distributeur, contacteurs), qui reçoivent des ordres de la partie commande.
- ✓ Des actionneurs (vérins, moteurs) qui ont pour rôle d'exécuter ces ordres, ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression), ou électrique en énergie mécanique.
- ✓ D'une détection (capteurs) qui informe la partie commande de l'exécution des états de la machine.

Dans un SAP (Système de Production Automatisé), le détecteur indique l'état des actions de service de surveillance (des renseignements sur le mécanisme). Il mesure, surveille et informe la PC sur l'évolution du système.

### I.3.2. La partie commande (PC)

La Partie Commande d'un système est un ensemble de composants constituant le traitement d'information (L'unité de traitement), destiné à coordonner la succession des actions de la partie opérative et à surveiller son bon fonctionnement, elle permet aussi de gérer le dialogue avec les intervenants et la communication avec d'autres systèmes. Elle assure le traitement des données et des résultats relatifs aux procédés, en matière d'œuvres, temps de production et à la consommation énergétique.

La structure de l'unité de commande est représentée sur la figure (I.3). [5]



**Figure. I.3.** Structure de l'unité de commande

#### **I.4. Le rôle d'un automate programmable dans les systèmes automatisés industriels**

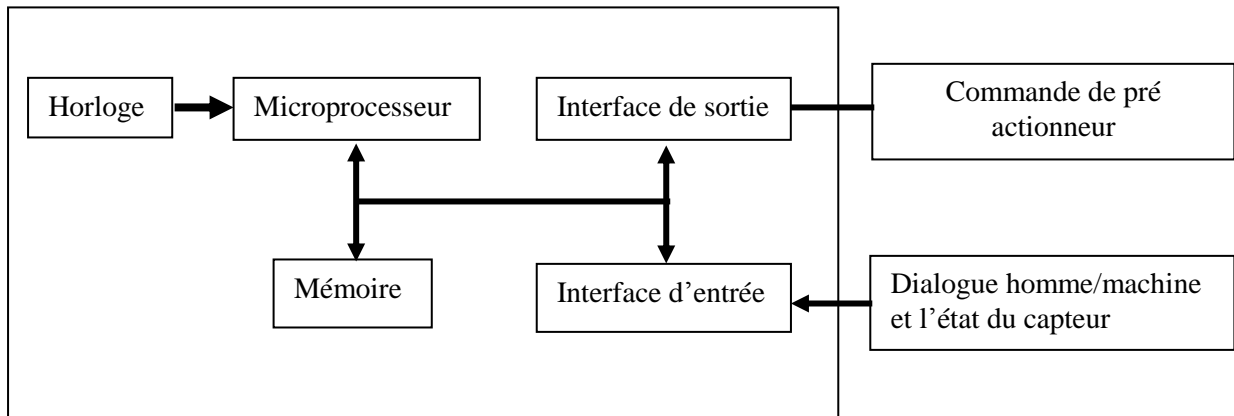
Les automates programmables industriels (API) représentent l'outil de base de l'automatisation des systèmes de production. Un API permet de piloter un système de production conformément à un programme placé dans sa mémoire. La flexibilité d'un API explique son large domaine d'utilisation, il est généralement placé en ambiance industrielle, où il représente le cœur de la partie commande d'un système de production. Il est en relation avec les autres parties du système grâce à son interface d'entrée-sortie (figure. I.4).

Une grande partie des API du marché possèdent un moniteur d'exécution mono-tâche.

Cependant, pour des raisons de performance, de plus en plus les constructeurs proposent des modèles dont le moniteur d'exécution supporte les multitâches et les interruptions. Dans un API cyclique, le programme s'exécute dans une boucle permanente. Dans chaque itération de cette boucle ou cycle, trois types d'actions (l'acquisition des entrées, l'exécution du programme et l'affectation des sorties) sont effectuées. [5]

### I.4.1. L'architecture interne d'un automate programmable

La structure interne d'un API peut se présenter comme indiqué dans la figure. I.4 :

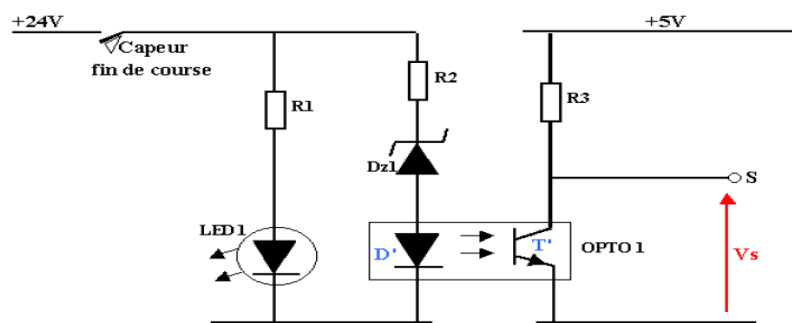


**Figure. I. 4.** Architecture interne d'un API.

L'automate programmable reçoit les informations relatives sur l'état du système et puis commande les pré-actionneurs suivant le programme inscrit dans sa mémoire

Un API se compose donc de trois grandes parties :

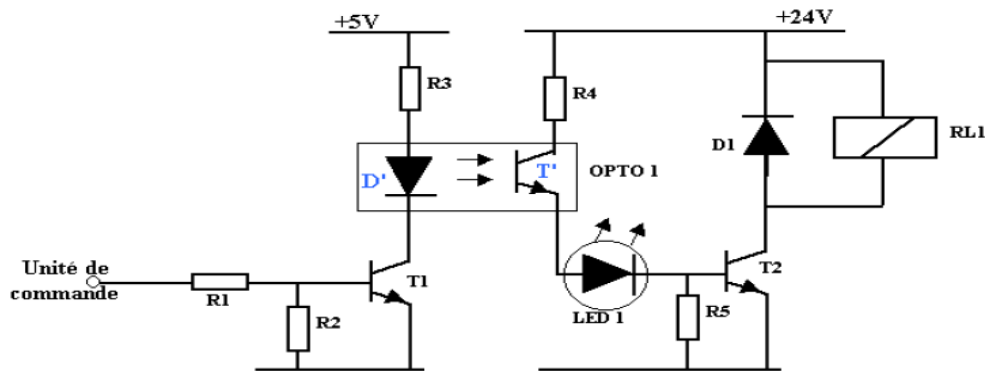
- Le processeur.
- La zone mémoire.
- Les interfaces Entrées/sorties.



**Fig. I.5.** L'interface d'entrée d'un API.

Les interfaces d'entrées reçoivent l'information en provenance du capteur, éliminent les parasites et isolent électriquement l'unité de commande de la partie opérative. (Figure. I.5)

Les interfaces de sortie commandent les pré-actionneurs et éléments de signalisation du système, et adaptent les niveaux de tension de l'unité de commande à celle de la partie opérative du système en garantissant une isolation galvanique entre ces derniers (figure. I.6).



**Fig. I.6** L'interface de sortie d'un API

#### I.4.2. Les langages de programmation des API

Un API est programmé à l'aide de langages spécialisés, fournis par son constructeur et utilisables au travers d'une interface (un logiciel sur PC...). Un standard définit cinq langages correspondant aux familles de langages les plus utilisées pour la programmation des API :

- Instruction List(IL) : C'est un langage textuel de type assembleur
- Structured Text (ST) : C'est un langage textuel structuré similaire au pascal.
- Ladder Diagram (LD) : C'est un langage graphique, très utilisé au milieu industriel, car il s'inspire des circuits de commande basés sur la logique « électrique », les équations combinatoires étant câblées à l'aide de contacts et de relais. Un programme est décrit par un diagramme sous forme d'échelle. Chaque échelon de l'échelle contient un ensemble de symboles graphiques qui peuvent être des contacts ou des bobines. Un contact permet la lecture d'une variable booléenne tandis qu'une bobine permet d'affecter une valeur à une variable booléenne.
- Sequential Function Charts (SFC): C'est un langage graphique permettant de structurer tout comportement séquentiel pouvant être décrit dans l'un des trois autres langages de la norme.
- Function Block Diagram (FBD): C'est un langage graphique permettant d'exprimer le comportement des fonctions, des blocs fonctionnels ou des programmes comme un ensemble de boîtes noires interconnectées (à la manière des portes logiques en électronique IL, ST, LD).

### **I.4.3. Les aspects temporisés dans la programmation des API**

Les API sont souvent utilisés comme plates formes matérielles d'exécution d'applications temps-réel, c'est-à-dire d'applications où des contraintes quantitatives sur les temps de traitement aux entrées doivent être satisfaites

#### **I.4.3.1. Durée du cycle automate**

La durée d'un cycle peut être exprimée comme la somme de la durée de la phase d'entrées- sorties et de la phase d'exécution du programme.

- Durée de la phase d'entrées-sorties :

Elle dépend de la technologie des interfaces entrées et des sorties (transistors, relais,...). Sa durée est généralement constante. Elle se situe typiquement entre quelques 10 microsecondes et 100 millisecondes.

- Durée de la phase d'exécution du programme :

La durée de cette phase est variable, cette variation est due aux différents chemins susceptibles d'être pris au cours de l'exécution du programme.

Sa valeur se situe entre 1 microseconde et 500 millisecondes (en fonction des caractéristiques de l'unité de traitement et de la taille du programme), phase d'exécution du programme. [5]

#### **I.4.3.2. Les constructions temporisées**

Les langages de programmation des API, et en particulier dans leurs descriptions, offrent un certain nombre de constructions qui font intervenir le temps «physique» de façon explicite. Les constructions les plus utilisées sont les temporisateurs.

La temporisation est une fonction que l'on retrouve dans un grand nombre d'applications. Elle est utilisée principalement afin de différer, d'une durée choisie fixée à l'avance, l'activation ou la désactivation d'une sortie. Elle est implémentée de façon logicielle dans la quasi-totalité des API existant sur le marché. On définit trois types de temporisateurs sous forme de blocs fonctionnels. [5]

##### **I.4.3.2.1. Le temporisateur d'enclenchement**

Le temporisateur d'enclenchement, ou *Timer On-delay* (TON) possède deux entrées et deux sorties (Figure. I.7).



➤ Les entrées :

IN, de type BOOL, qui permet de lancer ou d'annuler la temporisation.

PT (Preset Time), de type TIME, permet de spécifier la durée de temporisation.

➤ Les sorties :

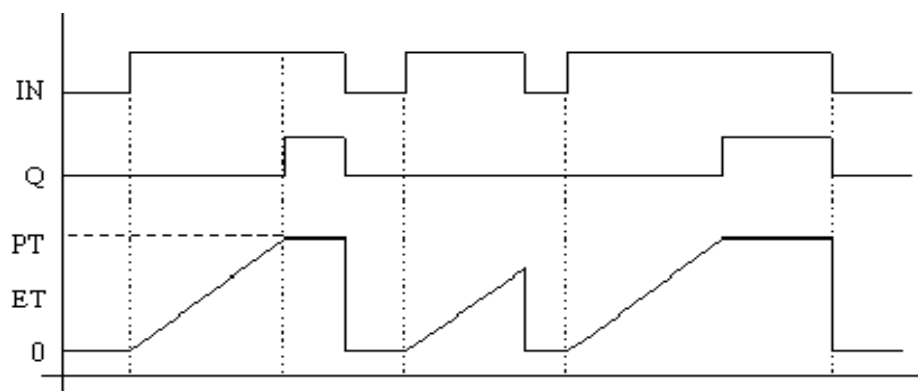
Q est de type BOOL, qui indique si la durée de temporisation a expiré.

ET (Elapsed Time), est de type TIME, indique le temps écoulé depuis le début de la temporisation.



**Figure. I.7** : un temporisateur TON.

Le comportement du temporisateur TON est illustré sur le chronogramme de la figure I.8.



**Figure1.8.** Le chronogramme du temporisateur

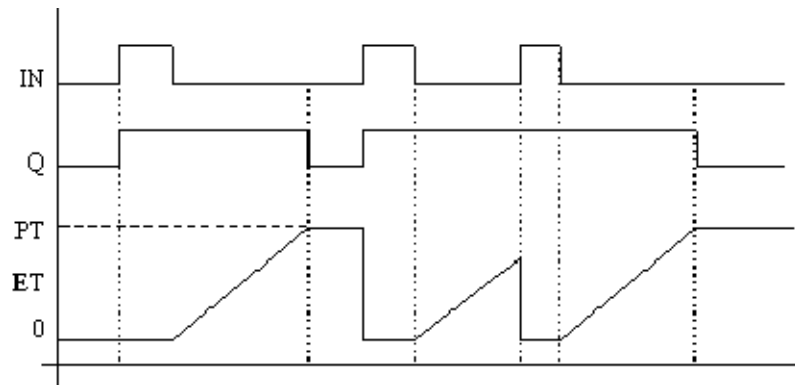
#### I.4.3.2.1. Le temporisateur de déclenchement

Le bloc TOF possède les même entrées et sorties que le bloc TON (Figure. I.9),



**Figure. I.9** : un temporisateur TOF.

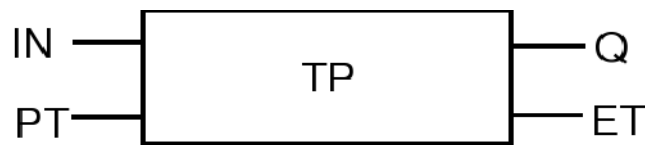
Cependant son comportement est différent, comme illustré sur le chronogramme de la figure. I.10.



**Figure. I.10** : le chronogramme du temporisateur de déclenchement

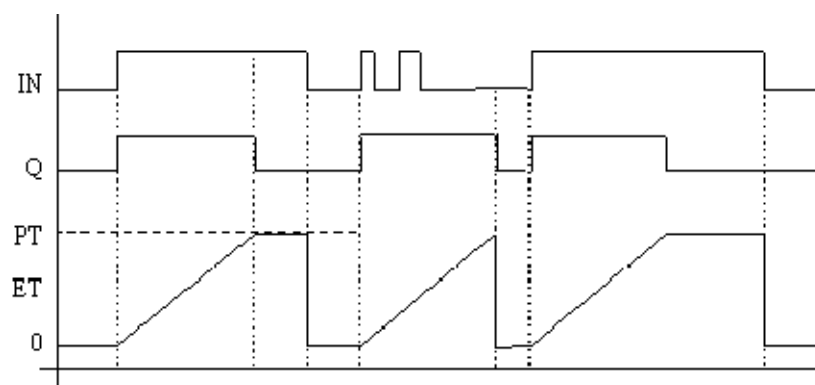
#### I.4.3.2.3. Le temporisateur impulsion

Le temporisateur impulsion possède les mêmes entrées et les même sorties que les temporisateurs TON et TOF (Figure. I.11).



**Figure. I.11** : un temporisateur TP.

Ce temporisateur permet de créer une impulsion de durée PT, en réponse à un front montant du signal d'entrée, et à condition qu'au moment du front montant de l'entrée IN, la sortie Q soit à 0



**Figure I.12** : Le chronogramme de temporisateur impulsion

## **I.5. Conclusion**

Les systèmes automatisés de production devient indispensable pour obtenir une compétitivité des produits fabriqués de haute qualité. Dans ce chapitre on a vu en générale la structure des systèmes de production et les appareils essentiels liés à ces systèmes pour la communication, la distribution d'énergie. Nous avons présenté les différentes parties de contrôle, puissance, communication et d'interface d'un système automatisé.

# Chapitre II

## Etude et description fonctionnelle du système

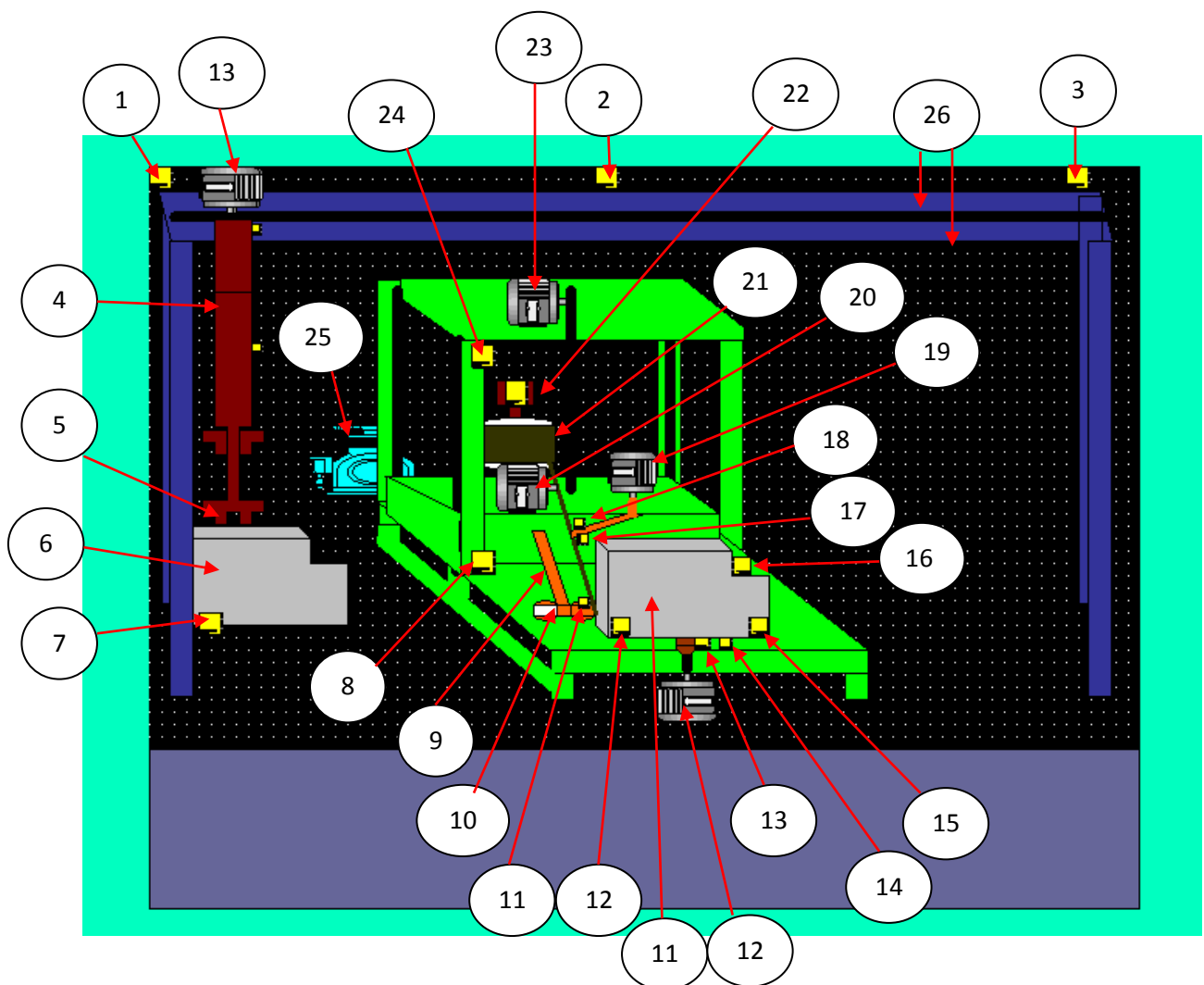
## II.1.Introduction

Les enrouleuses sont des machines très répandues dans l'industrie, on les trouve dans l'industrie de fabrication de congélateur, de tuyaux, de la gaines électriques, des tubes de PVC, des papiers rouleaux et d'autres produits en rouleaux.

La machine étudiée est une Enrouleuse tube sert envelopper et coller avec du ruban adhésif le tube en aluminium de diamètre 8 sur la chambre raide des congélateurs a caisse.

Cet appareil peut exécuter l'enveloppement des chambres modèle CF 1301 (modèle). Elle est semi-automatique et fonctionne à l'aide d'un séquenceur S7 de la firme SIEMENS. Elle est fabriquée sur commande de l'ENIEM à l'entreprise MANZONI en Italie dans le but de moderniser les équipements de production. Elle occupe une place très importante dans l'unité froid.

Dans ce chapitre, nous décrivons de l'enrouleuse et ces différentes parties ainsi son fonctionnement. Voici la machine :

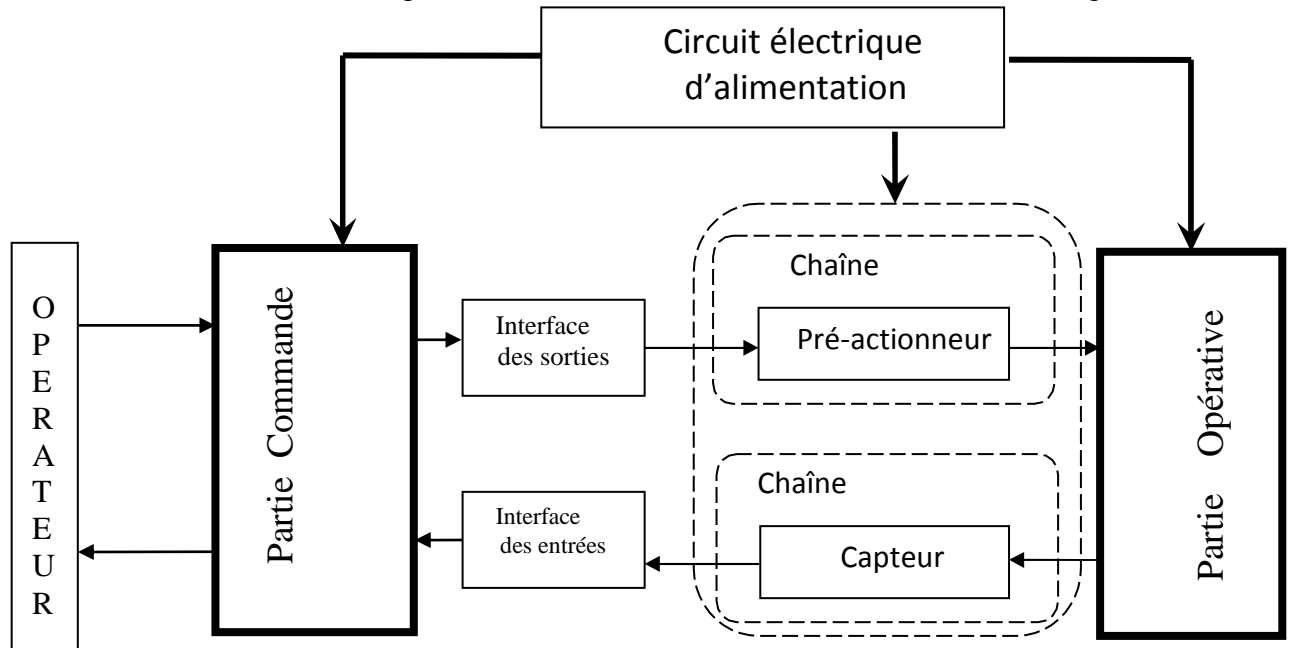


**Figure 1 :** schémas synoptiques de la machine en 3D.

- 1 : capteur de position c1 du charriot1 (p initial)
- 2 : capteur de position c2 du charriot1 (p milieu)
- 3 : capteur de position c3 du charriot1 (p final)
- 4 : vérin 1
- 5 : ventouse
- 6 : cuve
- 7 : capteur de présence de cuve
- 8 : position bas du charriot (2)
- 9 : bras de serrage de la cuve
- 10 : ruban adhésif
- 11 : capteur de présence de tube au niveau de moule
- 12 : fin de course de l'état de moule est ouvert
- 13 : capteur de position initial de moule
- 14 : capteur de compteur
- 15 : fin de course de l'état de moule est fermé
- 16 : capteur de présence de la cuve sur le moule
- 17 : fin de course indique que le tube est coupé
- 18 : fin de course indique que le tube n'est coupé
- 19 : moteur de coupage tube
- 20 : moteur d'attraction tube
- 21 : rouleau de tube
- 22 : vérin qui freine le rouleau de tube
- 23 : moteur de translation horizontale du bras
- 24 : position haute du charriot (2)
- 25 : source d'énergie (pression)
- 26 : raille de déplacement de charriot

## II.2.description de l'enrouleuse

L'enrouleuse est composée de différents modules comme illustré dans la figure 2.



**Figure 2 :** Modules composants de l'enrouleuse.

### ➤ La partie opérative (PO)

C'est la partie puissance, celle qui opère et agit, elle regroupe l'ensemble des actionneurs, des pré-actionneurs, de la structure mécanique, des capteurs et du circuit pneumatique.

### ➤ La partie commande (PC)

C'est la partie qui commande la partie opérative, elle est constituée de l'armoire de commande. Elle transmet les ordres aux actionneurs à partir d'une programmation chronologique d'informations délivrés par les capteurs et les consignes données par l'opérateur. Elle reçoit des informations des capteurs sous formes de signaux électriques adaptés.

### ➤ Les interfaces d'entrées/sorties

Une interface (I), souvent intégrée à la partie commande, qui assure le dialogue entre celle-ci et la partie opérative.

### ➤ Le circuit électrique d'alimentation

En automatique, on s'intéresse à la partie commande, son élaboration, sa description et sa validation. Dans les milieux industriels, ces parties commande (systèmes séquentiels) se concrétisent le plus souvent par un automate programmable.

## II.3. Constitution de la partie opérative de l'enrouleuse

### II.3.1. partie mécanique

Les éléments mécaniques constituant la machine sont décrits ci-après :

- **Rouleau de tube**

C'est un cylindre à rebords, autour duquel est enroulé le tuyau, il est monté sur un châssis fixé au sol, comme le montre la figure 3. [1]

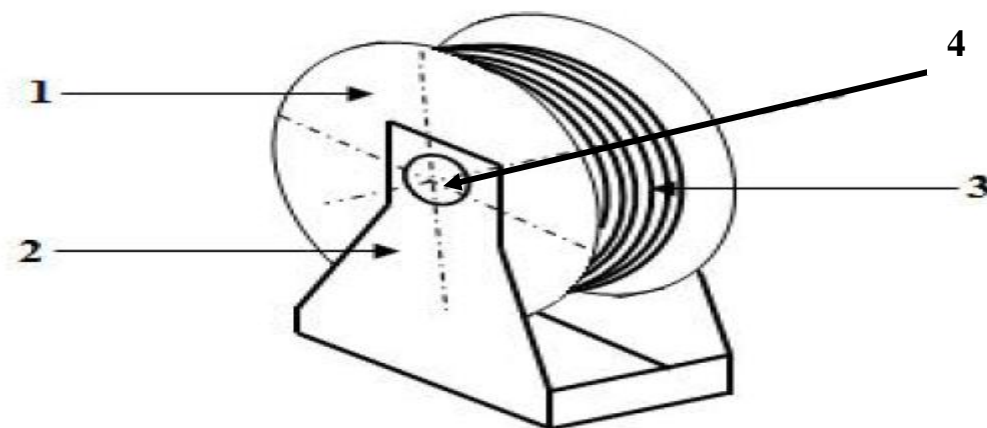


Figure 3 : Rouleau de tube

1 : Rouleau

2 : Châssis

3 : Tube

4 : axe de rotation

- **Galets :**

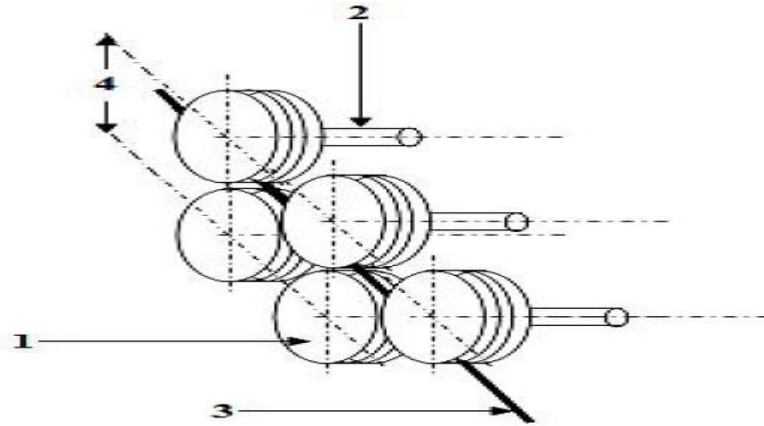
Ce sont des petites roues en mouvement rotatif, servant d'organes de contact avec le tuyau.

On retrouve, à cet effet :



### ✓ Galets de redressement :

Ils permettent d'éliminer d'éventuels défauts de planéité du tuyau, ils sont montés horizontalement ou verticalement selon la déformation du tuyau, leur principe de fonctionnement est illustré par la figure 4.



**Figure 4 : Galets de redressement**

1 : Galet

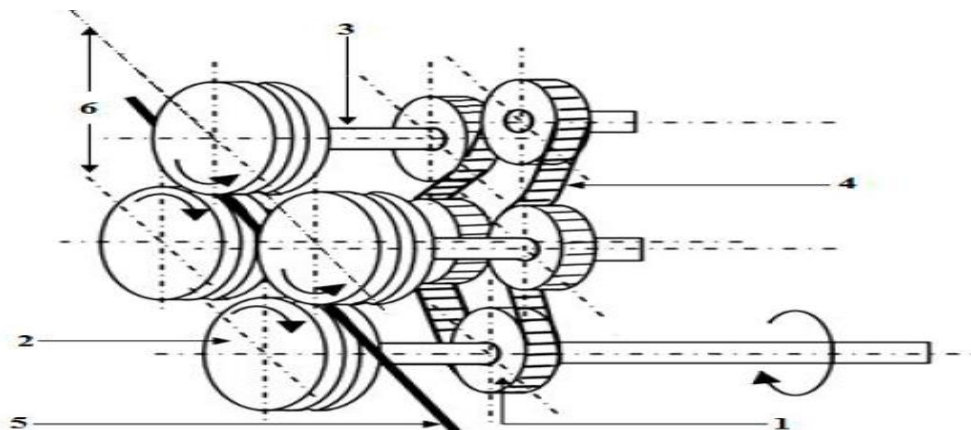
2 : Axe de fixation, sur lequel tourne le galet

3 : Tuyau

4 : Distance entre les axes des galets supérieurs et inférieurs. Celle-ci est réglée selon les défauts de planéité du tuyau. [1][4]

### ✓ Galets d'entraînement

Ils sont couplés au mouvement du moteur à CC (Mx) de l'axe (x) par un réducteur et une chaîne de transmission et assurent le mouvement du tuyau selon cet axe, leur principe de fonctionnement est illustré par la figure 5.



**Figure 5 : Galets d'entraînement.**

**1 : Pignon de transmission**

**2 : Galet d'entraînement**

**3 : Axe de fixation**

**4 : Chaîne de transmission à courroie**

**5 : Tuyau**

**6 : La distance entre les galets supérieurs et inférieurs, celle-ci permet d'agir sur la force d'entraînement. [1]**

- **Chariot :**

C'est le mécanisme qui englobe tout l'ensemble des : moteurs translation horizontale, et moteurs de translation verticale. [1]

- **Rails :**

L'ensemble des voies parallèles sur lesquelles se déplace le chariot. [1]

### **II.3.2. Prés-actionneurs**

Un pré-actionneur est un composant de gestion de l'énergie de commande d'actionneur, il transfère un signal de commande en un signal de puissance, et distribue l'énergie utile aux actionneurs selon la régulation de vitesse et de sens de rotation. A toute action est associé un pré-actionneur indispensable pour assurer le fonctionnement de l'actionneur.

#### **II.3.2.1. Electrovanne**

C'est un pré-actionneur électropneumatique tout ou rien (TOR) permettant le passage d'air issue de circuit pneumatique (Alimentation Pneumatique).

L'électrovanne est constituée principalement d'un corps de vanne où circule l'air. Elle est munie d'une bobine alimentée électriquement et engendrant une force magnétique qui déplace le noyau mobile qui agit sur l'orifice de passage d'air. La bobine doit être alimentée de façon continue pour maintenir le noyau dans la position consigne. [1]

#### **II.3.2.2. Distributeur**

C'est un organe dont le rôle est d'établir ou d'interrompre la communication entre la source d'air (pression) et les vérins ; ainsi il est inséré entre la source et les organes moteurs. Il est caractérisé par :

- Le nombre des orifices : 2, 3, 4 ou 5.
- Le nombre des modes de distribution ou position (case) : 2 ou 3.
- Le type de commande ou de pilotage assurant le changement de position : Simple pilotage avec rappel par ressort, ou double pilotage avec rappel au centre par ressort dans le cas des distributeurs à trois positions.
- La technologie de pilotage : Pneumatique ou électropneumatique.

Dans chaque case ou position, les voies sont figurées par des flèches indiquant le sens de circulation d'air voir figure 6.

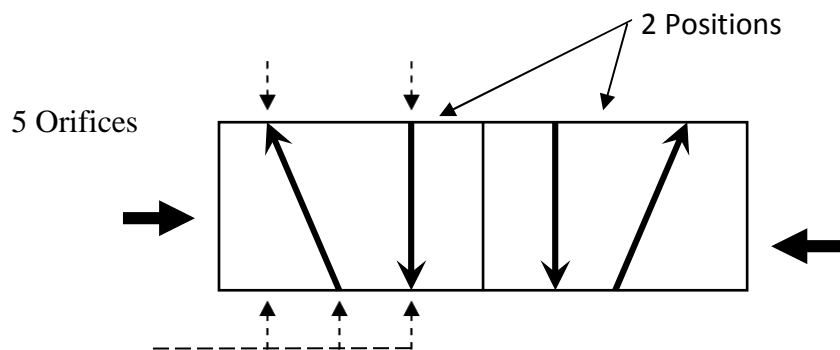


Figure 6 : Distributeur

La commande du changement de position est obtenue par déplacement du tiroir ou des clapets qui sont les éléments mobiles essentiels des distributeurs.

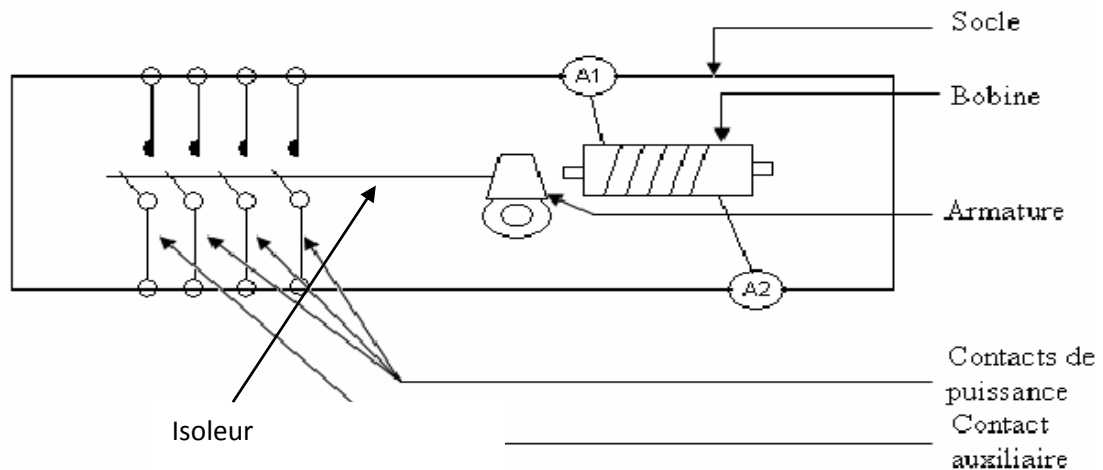
Dans la chaîne de production étudiée, des distributeurs 5/2 sont utilisés.

### II.3.2.3. Contacteur

Un contacteur est un relais de haute puissance comportant des contacts à double rupture pour s'assurer de pouvoir de coupure des tensions et la circulation des courants élevés. Les contacteurs sont utilisés pour commander de moyennes ou grandes charges électriques. Dès que l'on envisage de commander un moteur, on devrait utiliser un contacteur.

Il se compose d'une bobine qui est l'organe de commande, des contacts principaux et de contacts auxiliaires. Lorsque la bobine est alimentée en courant, l'armature est attirée et ferme les contacts. Lorsque la tension entre A1 et A2 (figure 5) est supprimée les contacts reviennent à la position initiale par l'action d'un ressort de rappel.

Le contacteur protège des chutes de tensions et des coupures de courant. En cas de coupure de courant ou de tension, le contacteur s'ouvre et seule une commande volontaire permet de le refermer. Le contacteur protège aussi les récepteurs des surcharges : le contacteur associé à un dispositif de détection (relais thermique) est capable de supporter et d'interrompre des courants de surcharges. [1][4]



**Figure 7 : Contacteur.**

### II.3.3. Actionneur

#### II.3.3.1 Actionneur électrique

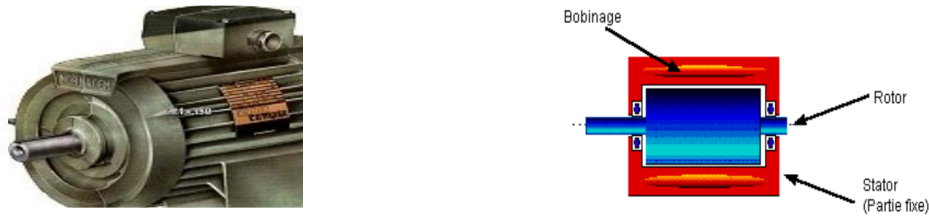
##### II.3.3.1.1. Les moteurs électriques

Le moteur électrique est un dispositif qui transforme l'énergie électrique en énergie mécanique (rotation). Il existe plusieurs types, les moteurs asynchrones, les moteurs synchrones, les moteurs à courant continu et les moteurs pas à pas. Sur cette machine on trouve deux moteurs triphasés asynchrones et deux moteurs à courant continu.

Plusieurs critères entrent en jeu pour le choix de type de moteur à utiliser une première sélection est faite sur la base de couple vitesse.

### II.3.3.1.2. Généralités :

Il est constitué d'une partie fixe, le stator qui comporte le bobinage, et d'une partie rotative, le rotor qui est bobiné en cage d'écureuil.



**Figure 6 :** moteur asynchrone

### II.3.3.1.3. Principe de fonctionnement :

Le principe des moteurs à courants alternatifs réside dans l'utilisation d'un champ magnétique tournant, produit par des tensions alternatives. La circulation d'un courant dans une bobine crée un champ magnétique  $\mathbf{B}$ . Ce champ est dans l'axe de la bobine, sa direction et son intensité sont fonction du courant  $\mathbf{I}$ . C'est une grandeur vectorielle.



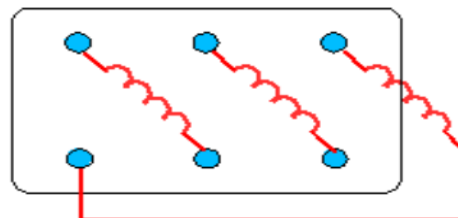
**Figure 8 :** sens de courant et de champ magnétique dans la bobine de moteur asynchrone.

### II.3.3.1.4. Le bobinage

Les bobines sont logées dans les encoches du stator. S'il y a une paire de pôles magnétique pour chacune des trois phases, la fréquence de synchronisme est alors de 3000 tr/mn. Si on augmente le nombre de paires de pôles, il est possible d'obtenir des moteurs avec des fréquences de rotation différentes.



**Figure 9 :** le stator.



**Figure 10 :** branchement des bobines.

Le branchement des bobines sur le réseau se fait au niveau de la plaque à borne située sur le dessus du moteur. On dispose ainsi de 6 connexions, une pour chacune des extrémités des trois bobines.

### II.3.3.1.5. Branchement étoile ou triangle

Il y a deux possibilités de branchement du moteur au réseau électrique triphasé. Le montage en étoile et le montage en triangle. Avec un branchement en étoile, la tension aux bornes de chacune des bobines est d'environ 230V. Dans le montage en triangle, chacune des bobines est alimentée avec la tension nominale du réseau (400V). On utilise le montage étoile si un moteur de 230V doit être relié sur un réseau 400V ou pour démarrer un moteur à puissance réduite dans le cas d'une charge avec une forte inertie mécanique.

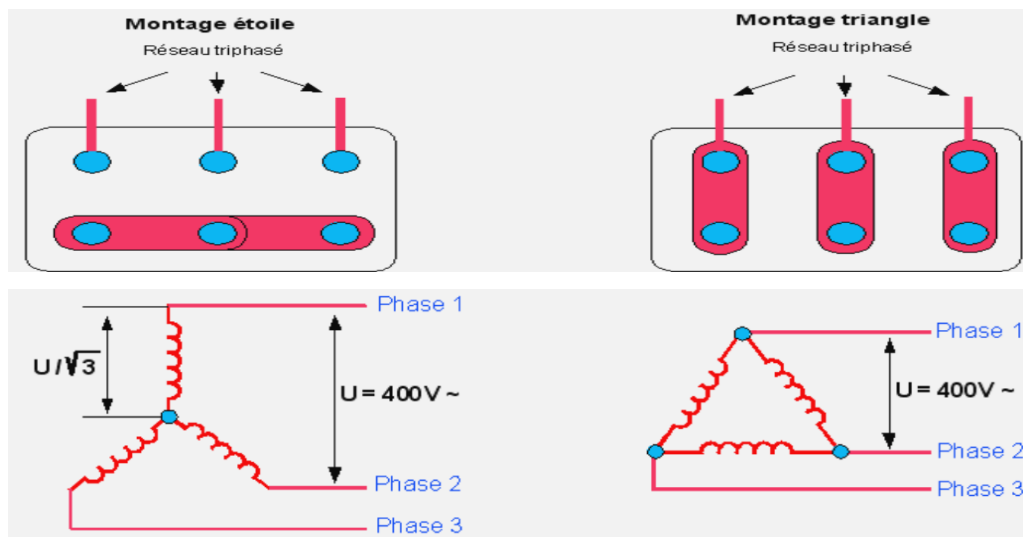


Figure 11 : montage en étoile et en triangle.

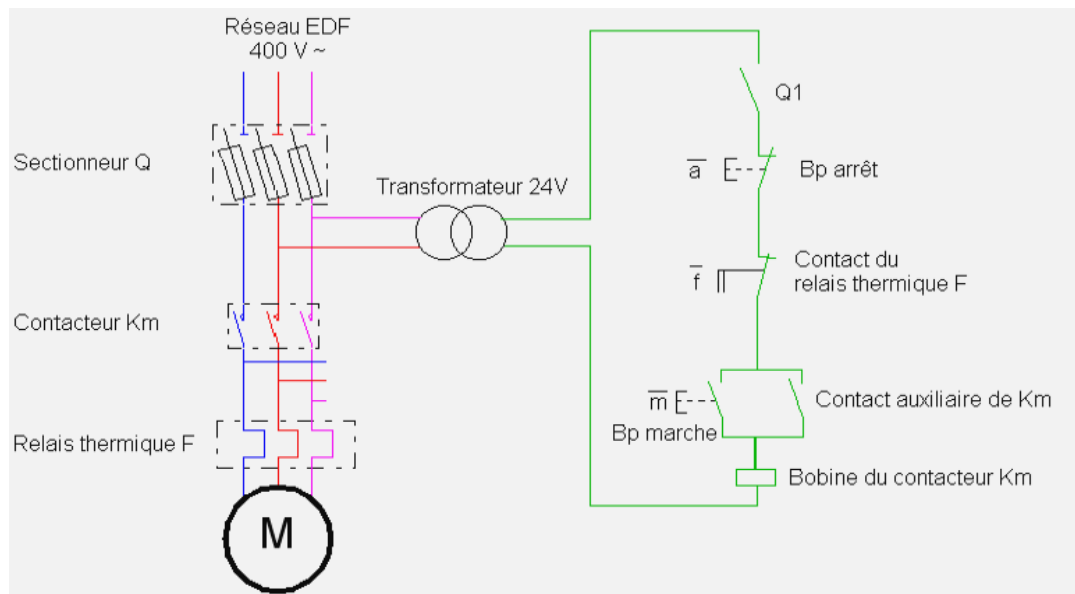
### II.3.3.1.6. Démarrage direct du moteur asynchrone

#### A. Démarrage direct du moteur asynchrone à un seul sens de rotation

Dans ce cas les enroulements du stator sont couplés directement sur le réseau, le moteur démarre et atteint sa vitesse nominale.

#### Fonctionnement

Après avoir fermé le sectionneur Q, l'action sur BP marche excite la bobine du contacteur Km qui s'autoalimente par le contact auxiliaire de Km.

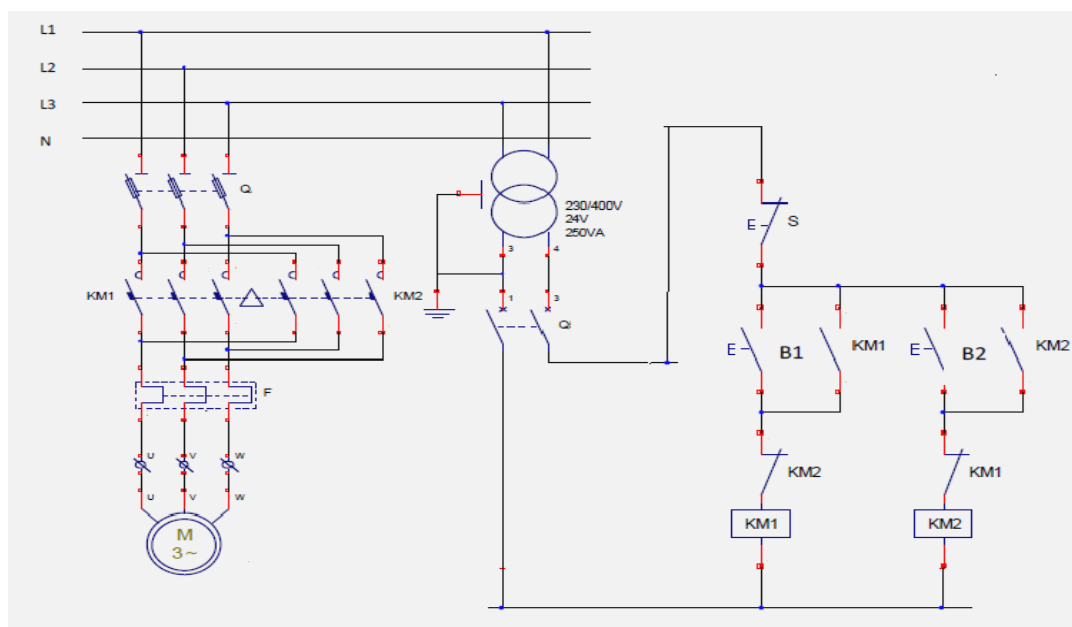


Le circuit de puissance

le circuit de commande.

**Figure 12 :** démarrage direct d'un moteur triphasé.**B. Démarrage direct des moteurs asynchrone à deux sens de rotation**

L'action sur B1 excite la bobine du contacteur KM1 qui s'autoalimente, Après l'arrêt du premier sens l'action sur B2 excite la bobine de KM2 donc le deuxième sens de rotation, un verrouillage entre les deux sens de rotation assure la sécurité (voir figure13).



Le circuit de puissance

le circuit de commande

**Figure 13 :** démarrage direct d'un moteur triphasé à deux sens de rotations.

**Le sectionneur (Q):** Il permet de déconnecter le moteur du réseau pour des opérations de maintenance, protège également le dispositif en aval des risques de court circuit grâce aux fusibles.

**Le relais thermique (F):** Protège le circuit contre les surcharges de courant, l'intensité maximale est réglable. Son action différentielle permet de détecter une différence du courant entre les phases en cas de coupure d'une liaison par exemple.

**Le contacteur (Km) :** Permet d'alimenter le moteur avec une commande manuelle ou automatique avec un automate programmable.



Figure 14 : présentation de pré-actionneur

### C. Plaques signalétiques :

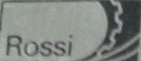

 Rossi		a company of the Habasit group www.rossi-group.com		IEC 60034-1 <b>IE1</b> made in Italy			
MOT.3 ~ N. 23025/11 5/11				IP 55	AMB. 40°C IC 411		
HBZ 80B 4 B5				kg 13	I.CL. F S 1 CONT.		
Freno Brake	Nm	V~/Hz		A	#B#	V=	
BZ04	11	110-480/50-60		0.20	RM1	103	
Esecuzione Execution							
A VY		Hz	A	kW		min <sup>-1</sup>	cos φ
230/400		50	3.3 / 1.9	0.75		1410	0.77
265/460		60	3.0 / 1.7	0.75 SF 1.15		1720	0.75
50HZ IE1 74.7 100% 74.2 75% 70.5 50%							
60HZ NEMA NOM.EFF 78.5% IHP DES.C 000E K							

Figure 15 : Plaque signalétique.



### II.3.3.2. Actionneurs pneumatiques

Un actionneur est un organe qui reçoit de l'énergie électrique pneumatique ou hydraulique venant d'une source extérieure pour la transformation en une action mécanique de translation (vérin), de rotation (moteur) ou d'aspiration (ventouse).

Leurs principales caractéristiques sont : la course, la force et la vitesse.

Les actionneurs pneumatiques sont constitués de plusieurs organes qui sont :

#### II.3.3.2.1. Vérin

C'est un actionneur linéaire qui transforme une énergie pneumatique ou hydraulique en un travail mécanique. Cet actionneur de conception robuste et de simplicité de mise en œuvre est utilisé dans toutes les industries manufacturières. Il permet de reproduire les actions manuelles d'un opérateur telles que : soulever, pousser, tirer, plier, serrer...etc.

Le vérin est constitué d'un piston solidaire, de la tige qui peut se déplacer à l'intérieur du corps (cylindre) (Figure 16). Le cylindre est délimité par le nez et le fond dans lesquels sont aménagés les orifices d'alimentation en air comprimé. Les espaces vides qui peuvent être remplis d'air comprimé s'appellent les chambres. La tige de piston du vérin à double effet est mise en mouvement par l'alimentation en air comprimée d'un côté ou de l'autre du piston. [1]

Sur le piston du vérin se trouve un aimant permanent dont le champ magnétique permet d'actionner des capteurs de proximité (fin de course).

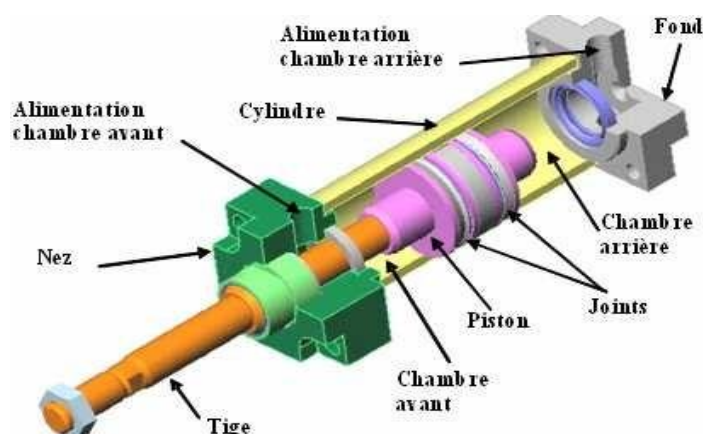
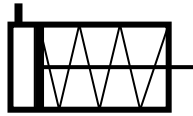


Figure 16 : Vue en coupe d'un vérin.

➤ **Vérin simple effet**

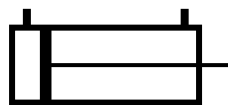
Le vérin simple effet (figure 17) est un composant monostable (stable dans une seule position). Ce type de vérin ne peut produire un effort significatif que dans un seul sens, le rappel de tige est assuré par un ressort. L'air comprimé est admissible sur une seule face du piston.



**Figure 17 :** Vérin simple effet.

➤ **Vérin double effet**

Le vérin double effet (figure 18) est un composant bistable (stable dans deux positions). Ce type de vérin peut produire un effort significatif dans les deux sens, le rappel de la tige est obtenu par inversion de l'alimentation des deux chambres. On l'utilise quand on a besoin d'un effort important dans les deux sens de déplacement. L'air comprimé est admissible sur les deux faces du piston.



**Figure 18 :** Vérin double effet.

#### **II.3.3.2.2. Compresseur (source d'énergie pneumatique) :**

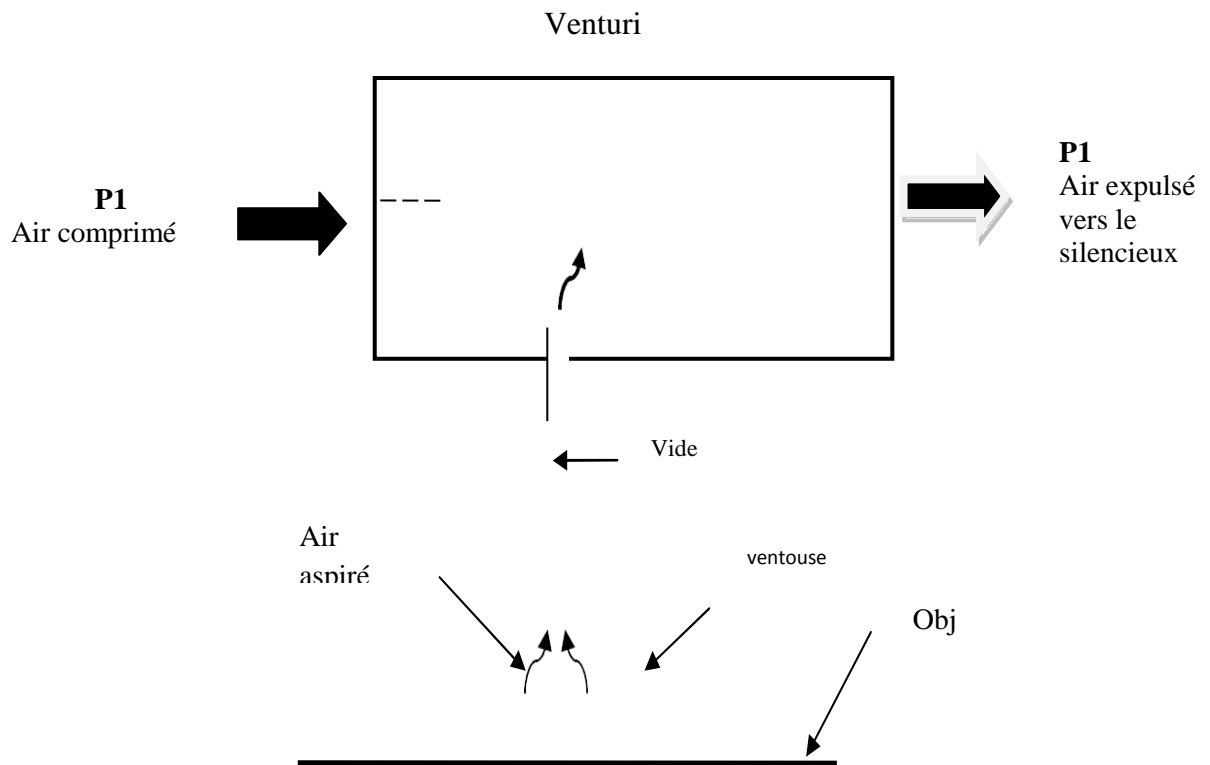
La plupart des ateliers industriels sont équipés d'un réseau de distribution d'air comprimé par un compresseur animé par un moteur électrique. Il comprime l'air ambiant à une pression de 8 à 10 bars. Sa caractéristique principale est son débit qui doit être au moins égal au débit maximal de l'installation. Il est constitué d'un filtre, du système de compression de l'air, d'un refroidisseur assècheur et d'un dernier filtre. La pression d'utilisation est souvent de 6 bars environs :  $1 \text{ bars} = 10^5 \text{ pascals} = 1 \text{ daN/cm}^2$

➤ **Avantages de l'air comprimé**

Energie propre facile à mettre en œuvre, puissance développée élevée, grande vitesse, sécurité pour les interventions humaines.

### II.3.3.2.3. Ventouse

Le schéma synoptique de la ventouse est donné par la figure 16. Le passage de l'air dans la buse d'éjection venturi (rétrécissement) augmente la vitesse de l'air comprimé arrivant en P1 et diminue sa pression en P2 ( $P2 < P1$ ), en provoquant alors une dépression qui permet d'aspirer l'air de la ventouse en V et de saisir en plaquant contre elle les objets à déplacer en les aspirants. Ce phénomène s'appelle l'effet Venturi. L'air est évacué à travers d'un silencieux d'échappement placé en sortie P2. [3][1]



**Figure 19 :** Effet venturi simple étage.

- Le venturi est dimensionné par rapport aux nombres de ventouses et leurs tailles.
- On dispose aussi de venturi à doubles étages qui permet, en début de dépression, des débits d'aspiration très élevés et donc de réduire les temps de mise en dépression.

#### **Inconvénient :**

S'il y a des fuites au niveau de la zone de contact, celles-ci entraînent une perte d'effectivité.

Avec le temps elles se dégradent, elles deviennent moins étanches, il faut donc prévoir leur changement.

### II.3.4. Capteurs

Un capteur est un opérateur technique qui transforme une grandeur physique liée au fonctionnement de l'automatisme, en un signal électrique (exemple : ILS (interrupteur à lames souples), capteur fin de course, ...).

Un capteur prélève une information sur le comportement de la partie opérative et la transforme en une information exploitable par la partie commande. L'information délivrée par un capteur peut être logique (deux états 1 ou 0), numérique (valeur discrète) ou analogique (dans ce cas il faudrait adjoindre à la partie commande un module de conversion analogique numérique).

Les capteurs peuvent être classés suivant les deux critères suivants :

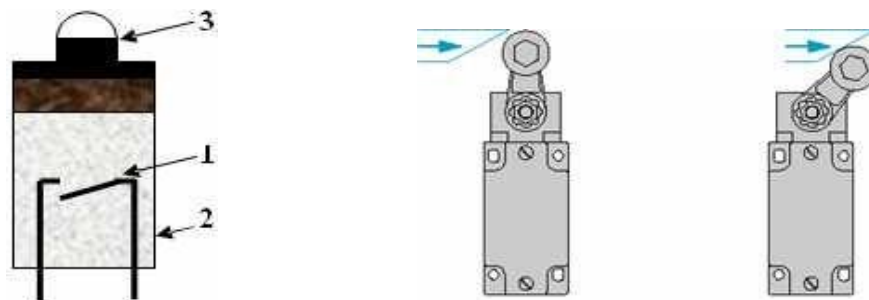
- La grandeur mesurée : on parle alors de capteurs de position.

Le caractère de l'information délivrée : on parle alors de capteurs logiques appelés aussi capteurs Tout ou Rien (TOR), de capteurs analogiques ou numériques. [3]

#### II.3.4.1. Capteur de position à action mécanique

Les interrupteurs de position sont constitués de trois éléments de base suivants :

- Un contact électrique (1),
- Un corps (2),
- Une tête de commande avec son dispositif d'attaque (3).



**Figure 20 :** Capteur à contact.

La détection de présence est réalisée lorsque l'objet à détecter entre en contact avec la tête de commande au niveau de son dispositif d'attaque. Le mouvement engendré sur la tête d'attaque provoque la fermeture du contact électrique situé dans le corps du capteur.

### II.3.4.2. Détecteurs de pressions (pressostat)

Ils sont destinés à contrôler une pression ou une dépression dans un circuit pneumatique ou hydraulique.

L'appareil transforme un changement de pression en un signal électrique ; lorsque la pression ou la dépression atteint la ou les valeurs de réglage, le contact électrique change d'état.

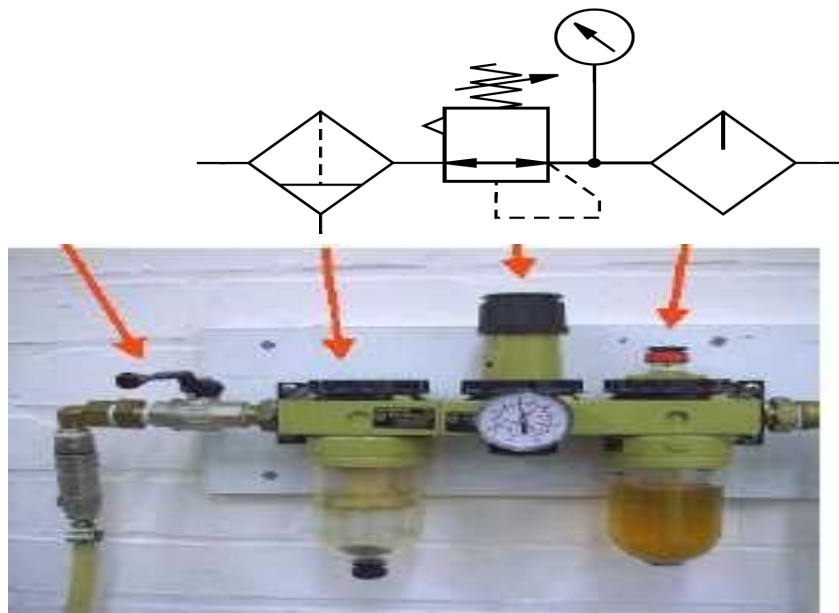


**Figure 21 :** Détecteur de pression.

### II.3.5. Adaptation de l'énergie pneumatique

Pour utiliser l'air comprimé disponible sur le réseau, il est nécessaire de le filtrer pour enlever la poussière, réguler pour maintenir la pression d'utilisation constante, et le lubrifier pour pulvériser de l'huile dans l'air comprimé.

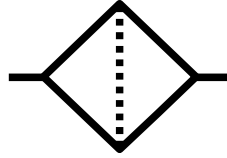
Ceci est conditionné par un dispositif de traitement d'air comprimé (figure 22).



**Figure 22 :** Unité de conditionnement (FRL)

### II.3.5.1. Le filtrage de l'air comprimé (filtre)

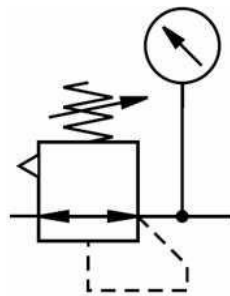
Le filtre sert à assécher les gouttes d'eau ou d'huile présentes dans l'air comprimé et élimine les particules solides en suspension dans l'air. Il sert aussi à filtrer les poussières (figure 23).



**Figure 23 :** Symbole d'un filtre.

### II.3.5.2. La régulation de la pression

Le régulateur de pression permet de régler la pression de fonctionnement d'un poste de travail et de faire retourner au bac le débit excédentaire. On lui associe toujours un indicateur de pression (figure 24).



**Figure 24 :** Symbole d'un régulateur de pression.

### II.3.5.3. La lubrification de l'air

Le lubrificateur (figure 25) permet de pulvériser de l'huile dans l'air comprimé. L'air comprimé est lubrifié :

- Pour des mouvements extrêmement rapides,

Pour des vérins au diamètre d'alésage important.

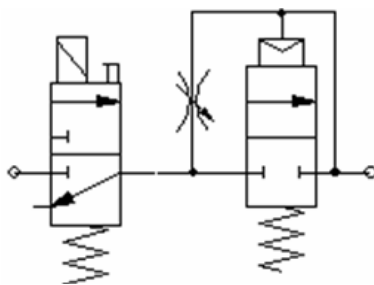
Notons que l'huile est agressive. Elle entraîne des dysfonctionnements et des grippages des composants pneumatiques.

#### II.3.5.4. Mise en pression progressive

Pour des raisons évidentes de sécurité :

- La mise en pression de la machine doit être progressive.
- La mise hors pression de la machine doit être très rapide.

Le dispositif de mise en pression progressive assure une montée progressive de la pression dans l'installation en agissant sur la vitesse de remplissage du circuit. Monté en sortie du FRL (filtrage, régulateur et lubrificateur) et avant le sectionneur général, il protège les personnes d'une brusque remise en service des actionneurs.



**Figure 26 :** Mise en pression progressive.

### II.4. Partie commande du système

#### II.4.1. Cahier de charge

Le travail qui nous a été demandé au niveau de l'unité froid de l'entreprise ENIEM est d'étudier l'enrouleuse tube et de concevoir un système automatisé qui permet de transporter les cuves de congélateur afin d'augmenter la production ainsi que pour assurer la sécurité des opérateurs.

Les causes qui ont motivées ce travail sont :

- Le nombre de cuve fabriqué insuffisant.
- Amélioration des sécurités et de la cadence de production..
- Politique suivie par l'entreprise qui est la modernisation de ses équipements de production.

### II.4.2. Le fonctionnement de la machine

Avant la mise en marche en mode automatique, on doit procéder à la phase de préparation, qui consiste à mettre sous-alimentation les différents organes électriques, pneumatiques de la machine. Une fois la préparation est accomplie, il faut vérifier toutes les positions initiales de chaque bloc de la machine, Choisir un mode de fonctionnement et contrôler la pression de service.

En appuyant sur le bouton poussoir « sectionneur général et la vanne d'air » l'API s'allume et un capteur détecte la présence de pression, une fois que l'API soit allumé en appuyant sur les trois boutons poussoir, l'auxiliaires pour que la table des boutons poussoirs s'allume, reset pour mettre le programme à zéro, star cycle pour que notre programme soit prêt à l'utiliser.

L'opérateur décharge la cuve dans la position c1, un capteur à contact détecte la présence de la cuve, le charriot (01) se déplace horizontalement a l'aide d'un moteur jusqu'à la position c1, le vérin vertical (01) descend en portant les ventouses jusqu'à détection du cuve indiqué par le capteur à contact et enclenche l'aspiration des ventouses en permettant de le saisir.

Une fois la cuve est bien saisi, le pressostat d'air émet un signal et provoque la rentrée du vérin vertical. Le capteur fin de course du vérin (01) détecte la rentrée complète du vérin vertical et le charriot(01) se déplace horizontalement vers le position(02) au-dessus de moule en provoquant l'arrêt du charriot(01), si le moule est vide et il est a sa position initiale cela provoque la sortie du vérin(01) en enclenchant le relâchement de la cuve par les ventouses, un capteur a contacté détecte la présence de la cuve sur le moule cela provoque la rentrée du vérin(01) et la sortie de vérins(02) pour qu'il serre la cuve, et le tube avance automatiquement vers le moule.

Une fois que le tube active le capteur à contacte du moule provoque l'arrêt du moteur traction tube.

En appuyant sur les boutons Start et l'avance tube les quatre opérations commence :

Le moteur traction tube et la fermeture de vérin(03) (celui qui est fixer sur le bras d'enrouleuse) avec une temporisation et le moteur du charriot(02) monte jusqu'à la partie haute de la machine et le moteur de moule fait tournée le moule avec un compteur qui compte le nombre de tours.

Dès que les quatre opérations finisse le vérin(04) freine la bobine de tube et le



vérin(05) sort et le moteur de rotation des lames tourne autour de tube , ce dernier enclenche l'ouverture du vérin(03) et ça provoque la descente de charriot(02) complètement jusqu'il soit détecté par un capteur a contact. Le vérin vertical (01) équipé d'un système de freinage, descend en portant les ventouses jusqu'à détection de la cuve indiquée par le capteur de contact et enclenche l'aspiration des ventouses en permettant de le saisir.

Une fois la cuve est bien saisi, le pressostat d'air émet un signal et provoque-la rentrée du vérin vertical. Le capteur fin de course du vérin (01) détecte la rentrée complète du vérin vertical et le charriot(01) se déplace horizontalement vers la position(03) en provoquant l'arrêt du charriot(01), et le vérin(01) vertical descend complètement en enclenchant le relâchement de la cuve par les ventouses.

Une fois que la tâche est bien saisi le pressostat d'air émet un signal et provoque-la rentrée du vérin vertical ainsi le déplacement de charriot(01) vers la position initial c1.

## **II.5. Conclusion**

Dans ce chapitre, nous avons décrit la machine « enrouleuse de cuve » et les composants essentiels qui la constituent. Ainsi que son principe de fonctionnement afin de préparer à la modélisation que nous allons étudier dans le chapitre suivant.

# Chapitre III

Modélisation du système à l'aide du GRAFCET

### **III.1. Introduction**

Pour la conception d'un système automatisé industriel nous devons disposer d'un cahier de charges, établi en collaboration avec les différents services utilisant ce système. En plus des contraintes techniques, il comporte des instructions impératives reliant la partie commande à la partie opérative, ainsi que le dialogue avec l'opérateur.

La conception, l'étude et la réalisation d'un automatisme nécessite une démarche structurée qui fait appel à un outil de description des systèmes automatisés séquentiels dans l'ordre chronologique des étapes tels que : le chronogramme, l'organigramme ou encore le GRAFCET.

Afin de modéliser notre système, nous avons choisi d'utiliser le GRAFCET. Qui est considéré comme un outil simple mais puissant, permettant de modéliser parfaitement le système en tenant compte des contraintes de fonctionnement.

### **III.2. Définition**

Le GRAFCET (Graphe Fonctionnel de Commande par Etape et Transition) est un diagramme fonctionnel dont le but est de décrire graphiquement, suivant un cahier de charges, les différents comportements de l'évolution d'un système séquentiel. Il est parfois simple à utiliser et rigoureux sur le plan formel et constitue un outil de dialogue entre toutes les personnes collaborant à la conception, à l'utilisation ou à la maintenance de la machine à automatiser.

Le mot GRAFCET (en lettres majuscules) est utilisé pour faire référence à l'outil de modélisation. Alors que le mot grafcet (en lettres minuscules) fait référence à un model obtenu à l'aide des règles de GRAFCET.

Le GRAFCET permet de construire des modèles ayant une structure graphique (représentation statique) à laquelle on associe une interprétation (elle correspond à l'aspect fonctionnel du grafcet). De plus, ce modèle possède un comportement dicté par des règles d'évolution (représentation dynamique), complétées pour l'implantation par des algorithmes d'application de ces règles.

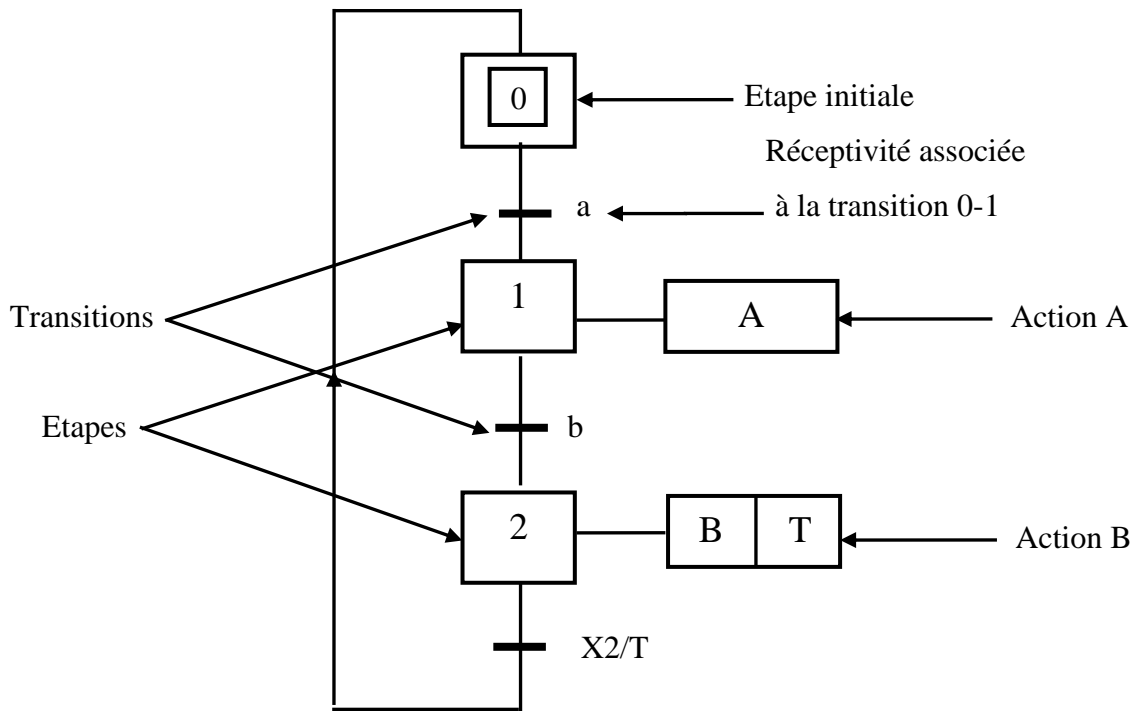
### **III.3. les concepts de base d'un GRAFCET**

Le GRAFCET se compose d'un ensemble :

- D'étapes auxquelles sont associées des actions (activités).
- De transition auxquelles sont associées des réceptivités.

- Des liaisons orientées reliant les étapes aux transitions et les transitions aux étapes.

La figure III.1 montre les éléments de bases d'un GRAFCET.



**Figure III.1 :** Symbolisation d'un grafcet.

Par convention, le sens naturel d'évolution est du haut en bas. Dans un cas différent, il faut montrer le sens d'évolution par une flèche.

### III.3.1. Règles d'évolution du grafcet

Les règles d'évolution réglementent la situation initiale, le franchissement des transitions, l'évolution des étapes actives, les évolutions simultanées, l'activation et la désactivation.

- **Règle 1**

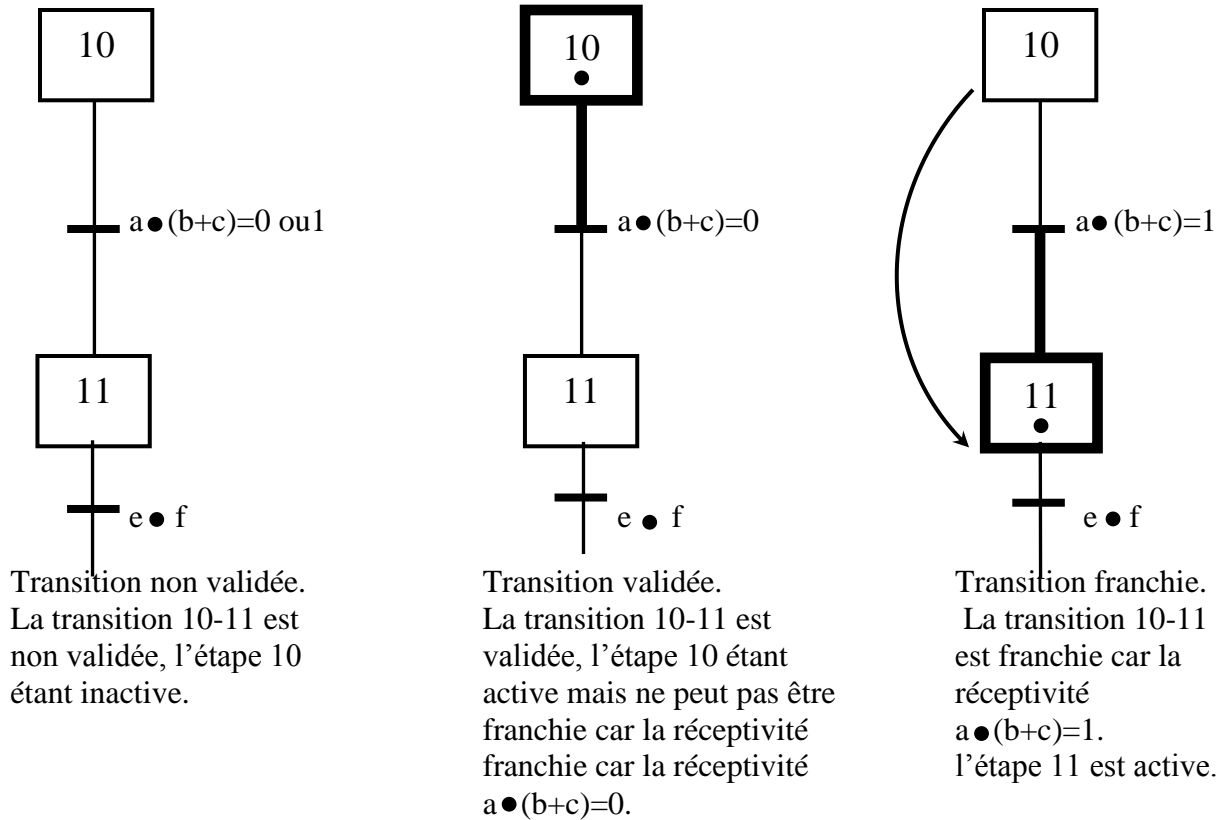
La situation initiale caractérise le comportement initial de la partie commande vis-à-vis de la partie opérative. Elle correspond aux étapes d'actives au début de fonctionnement (étapes initiales ou étapes d'attente).

Dans un grafcet, il doit y avoir une seule étape initiale et elle est unique.

- **Règle 2 : Franchissement d'une transition**

Une transition peut être, soit validée soit non validée. Elle est validée lorsque toutes les étapes immédiatement précédentes sont activées. Elle ne peut être franchie que lorsqu'elle est validée et que la réceptivité associée à la transition est vraie.

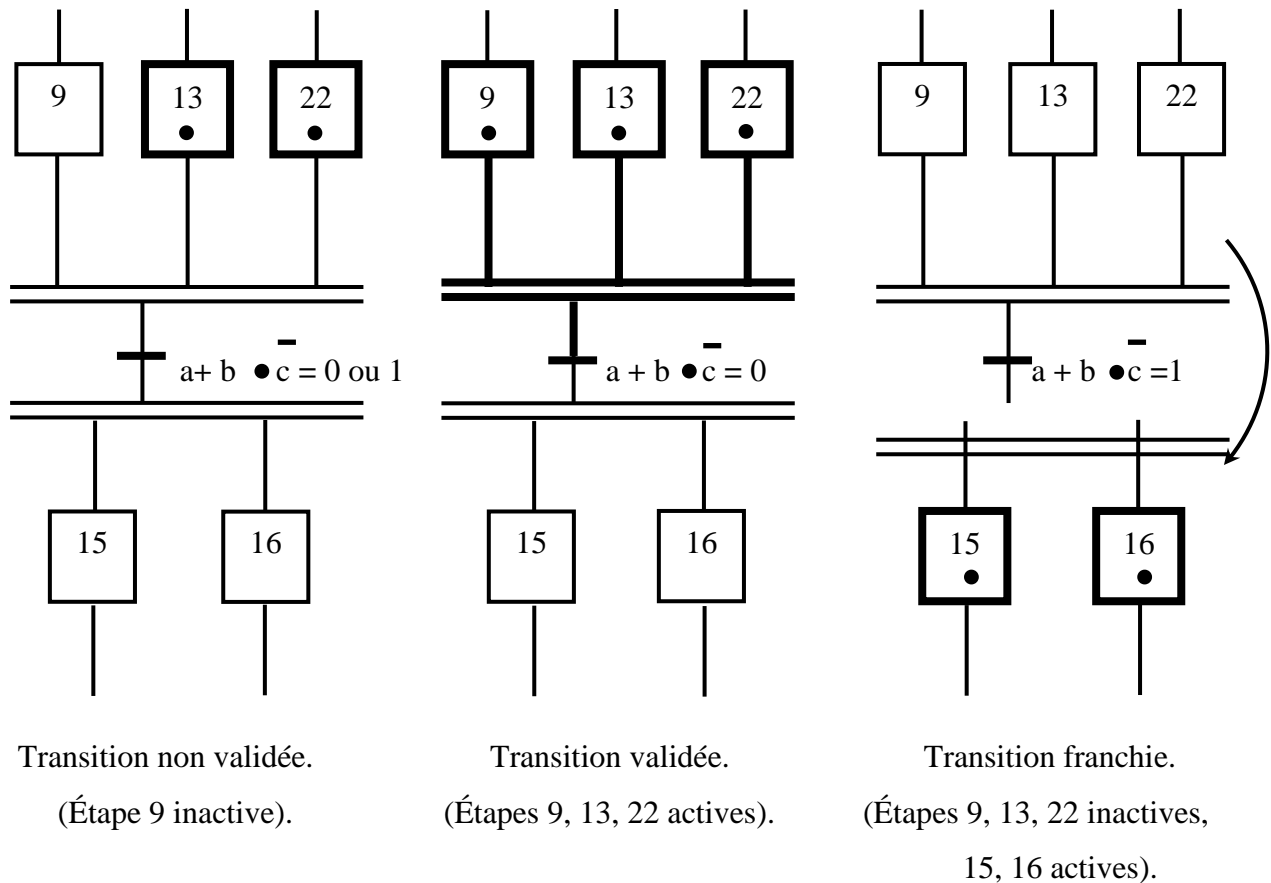
La transition est alors obligatoirement franchie.



- **Règle 3 : évolution des étapes**

Cette règle s'applique dans le cas d'un grafcet à une ou plusieurs séquences.

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



**Figure III.3 :** Évolution des étapes actives.

#### Règle 4

Plusieurs transitions simultanément franchissables, sont simultanément franchies.

#### Règle 5

Si au cours du fonctionnement, une même étape doit être désactivée et activée simultanément, elle reste activée. L'activation doit être prioritaire sur la désactivation au niveau d'une même étape.

#### Remarque

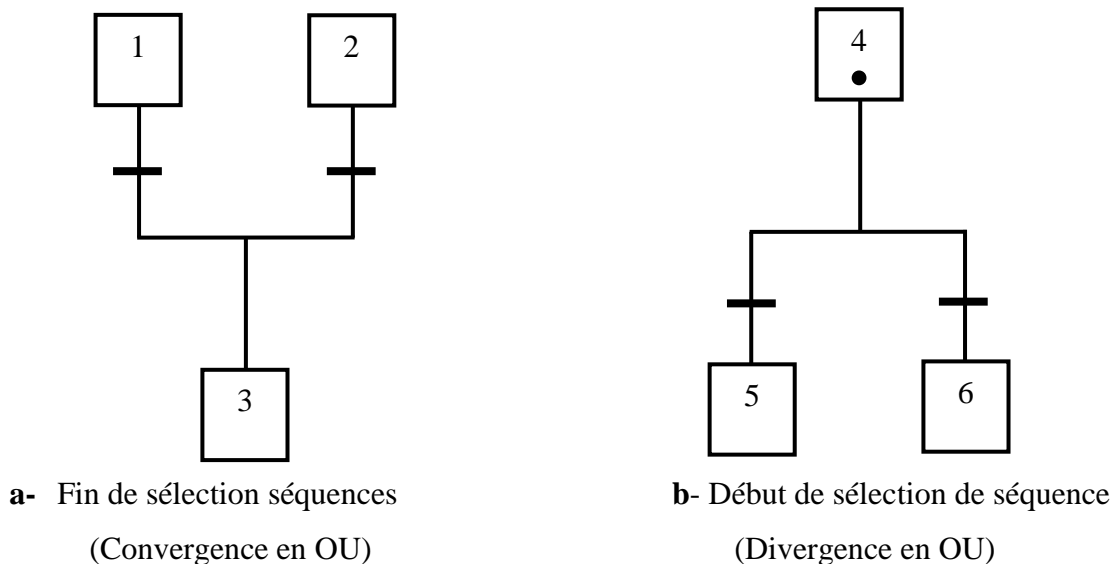
La durée de franchissement d'une transition ne peut jamais être rigoureusement nulle, même si, théoriquement (règles 3 et 4), elle peut être rendu aussi petite que possible. Il en est de même de la durée d'activation d'une étape. En outre, la règle 5 se rencontre très rarement dans la pratique.

### III.3.2. Sélection de séquences et séquences simultanées

Le GRAFCET présente deux structures particulières : la sélection de séquences et les séquences simultanées.

#### a) Sélection de séquences

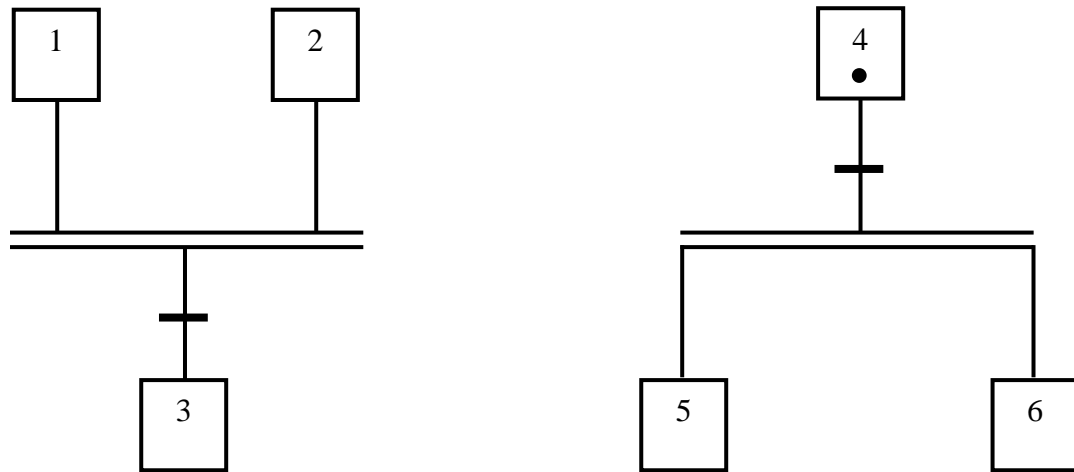
La sélection de séquences dans un grafcet permet de choisir une suite d'étapes plutôt qu'une autre. Cette structure est composée d'une seule étape en amont et de plusieurs transitions en aval qui permettront le choix de la séquence. Elle est représentée à l'aide d'un simple trait horizontal. La fin d'une sélection de séquence permet la reprise d'une séquence unique (Figure III.4).



**Figure III.4 :** Représentation graphique d'une sélection de séquences.

#### b) Séquence simultanées

Cette structure est composée d'une seule étape et d'une seule transition en amont qui permet de déclencher simultanément plusieurs séquences d'étapes. Elle est représentée à l'aide d'un double trait horizontal. A la fin d'une série de séquences simultanées, on trouve, en général, un double trait suivi d'une seule transition.

**a-** Fin de séquences simultanées

(Convergence en ET)

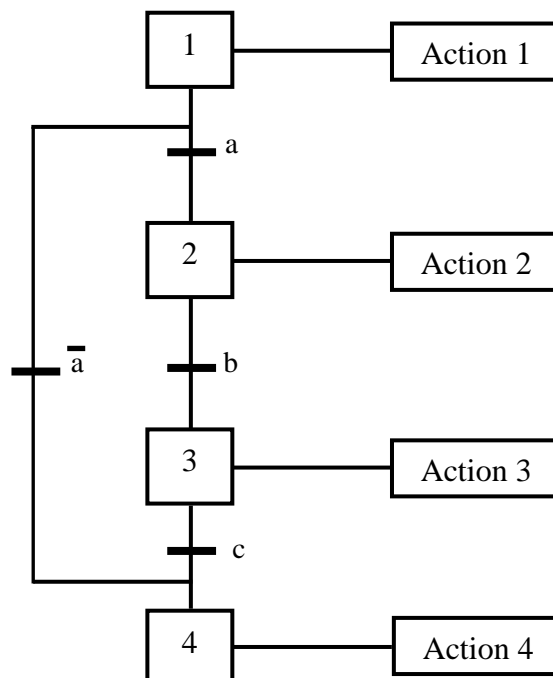
**b-** Début de séquences simultanées

(Divergence en ET)

**Figure III.5 :** Représentation graphique d'une sélection de séquence.

### III.3.3 : Saut d'étape

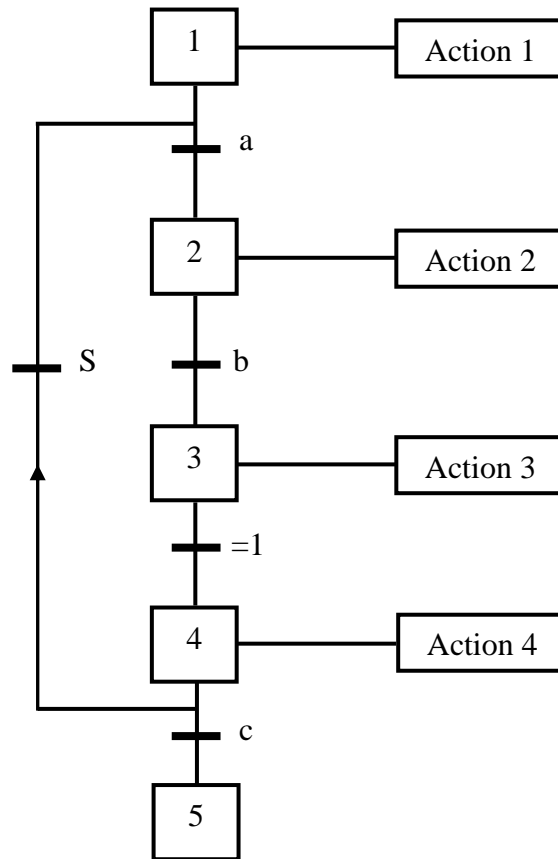
Le saut permet de sauter une ou plusieurs étapes lorsque les actions associées à ces étapes deviennent inutiles (Figure III.6).

**Figure III.6 :** Représentation graphique du saut d'étape.



### III.3.4. Reprise de séquence

Permet de recommencer plusieurs fois la même séquence tant que la condition fixée n'est pas obtenue (Figure III.7).



**Figure III.6 :** Représentation graphique d'une reprise de séquence.

## III.4. Niveau d'un grafcet

### III.4.1. Grafcet niveau 1

Appelé aussi niveau de la partie commande. Il décrit l'aspect fonctionnel du système et les actions à faire par la partie commande en réaction aux informations provenant de la partie opérative indépendamment de la technologie utilisée. Les réceptivités sont décrites en mots et non en abréviation. On associe le verbe à l'infinitif pour les actions.

### III.4.2. Grafcet niveau 2

Appelé aussi niveau de la partie opérative. Il tient compte de plus de détails des actionneurs, des pré-actionneurs et des capteurs, la représentation des actions et réceptivités

sont écrits en abréviations et non en texte, on associe une lettre majuscule à l'action et une lettre minuscule à la réceptivité.

### III.4.3. Grafcet niveau 3

Dans ce cas on reprend le grafcet de niveau 2, en affectant les informations aux étiquettes d'entrée de l'automate et les ordres aux étiquettes de sortie de l'automate. Il s'adapte aux caractéristiques de traitement d'un automate programmable industriel donnée, de façon à pouvoir élaborer le programme, procéder à la mise en œuvre et assurer son évolution.

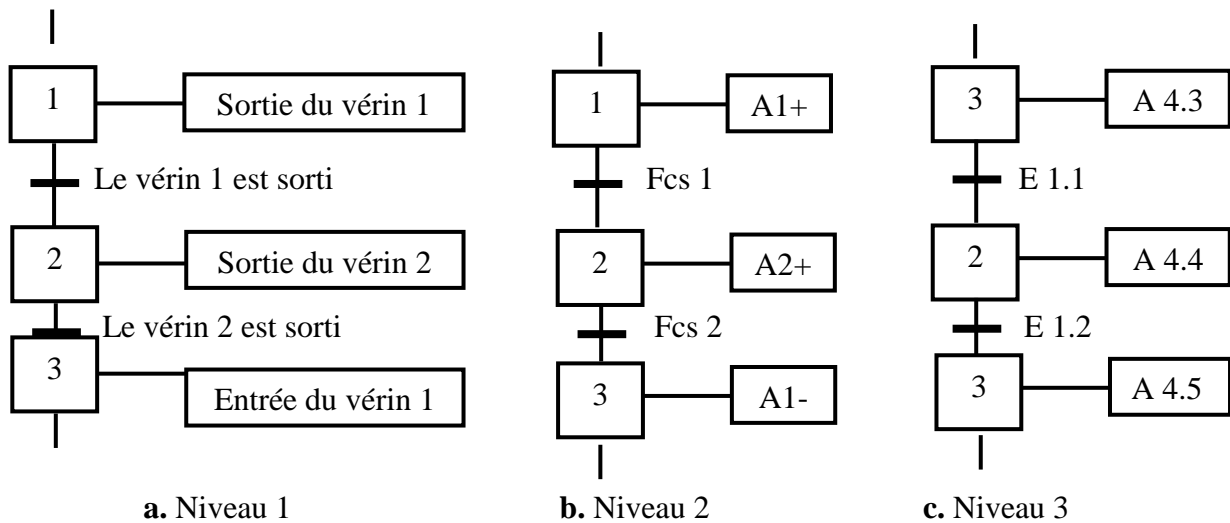


Figure III.7 : Les niveaux de grafcet.

## III.5. Modélisation par l'outil GRAFCET

### III.5.1. GRAFCET du système

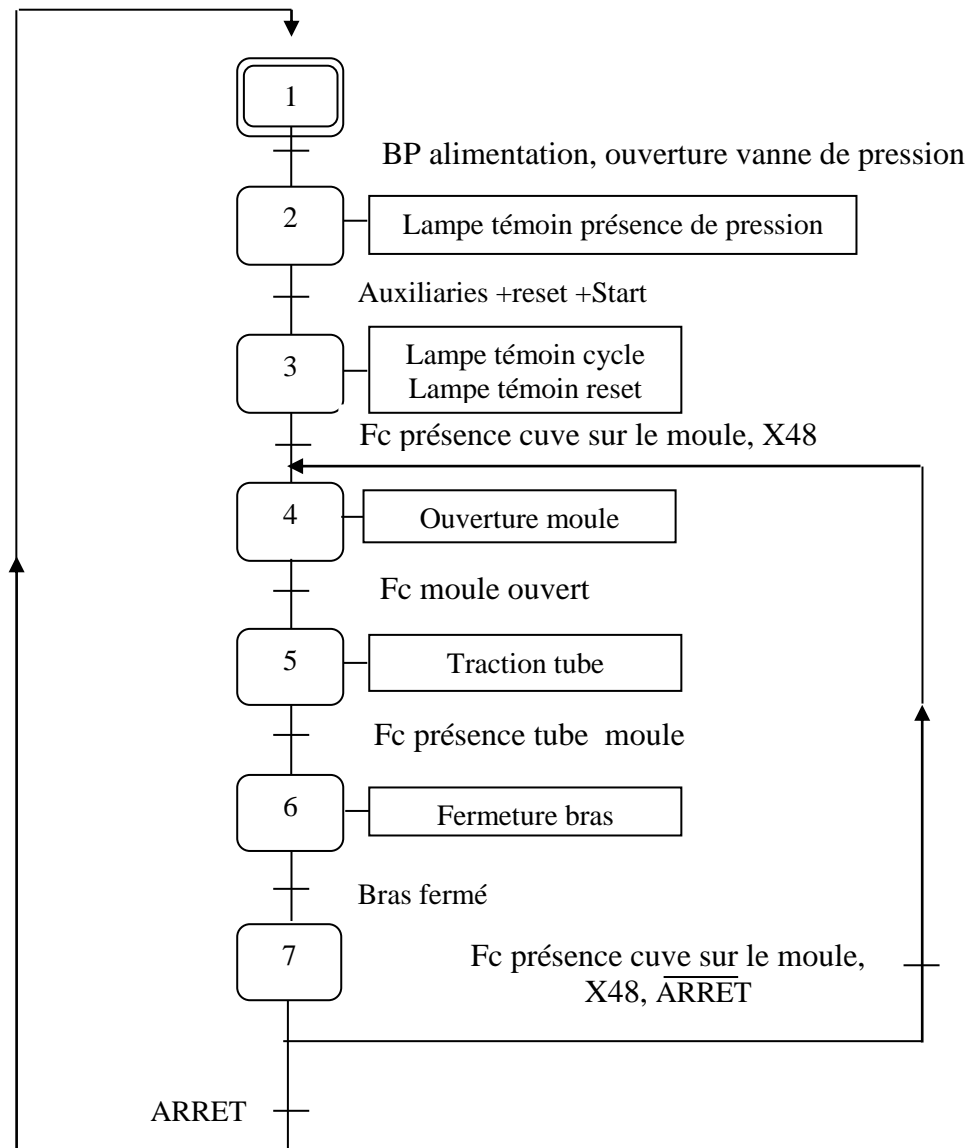
#### ❖ Niveau 1

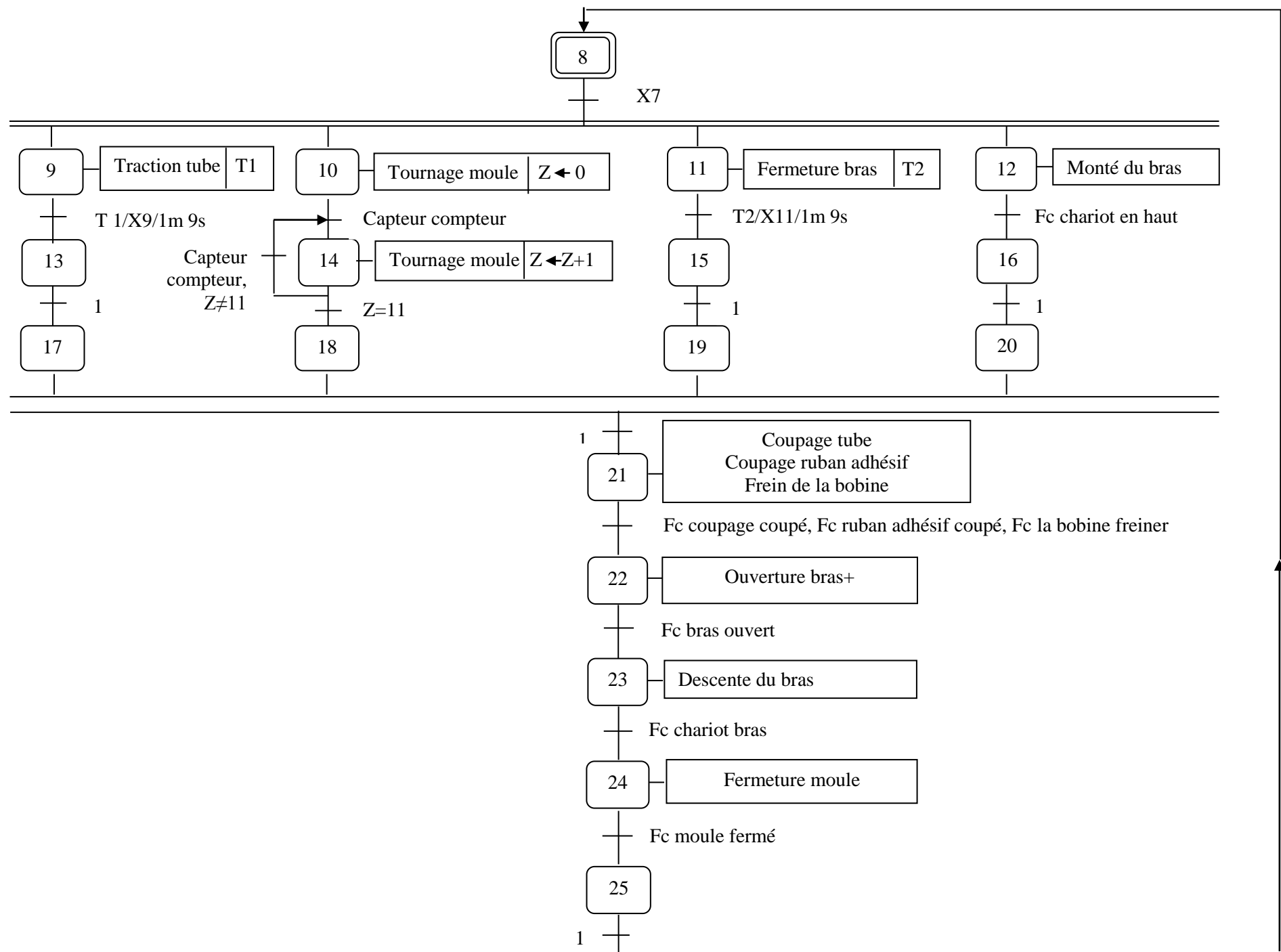
##### Conditions initiales :

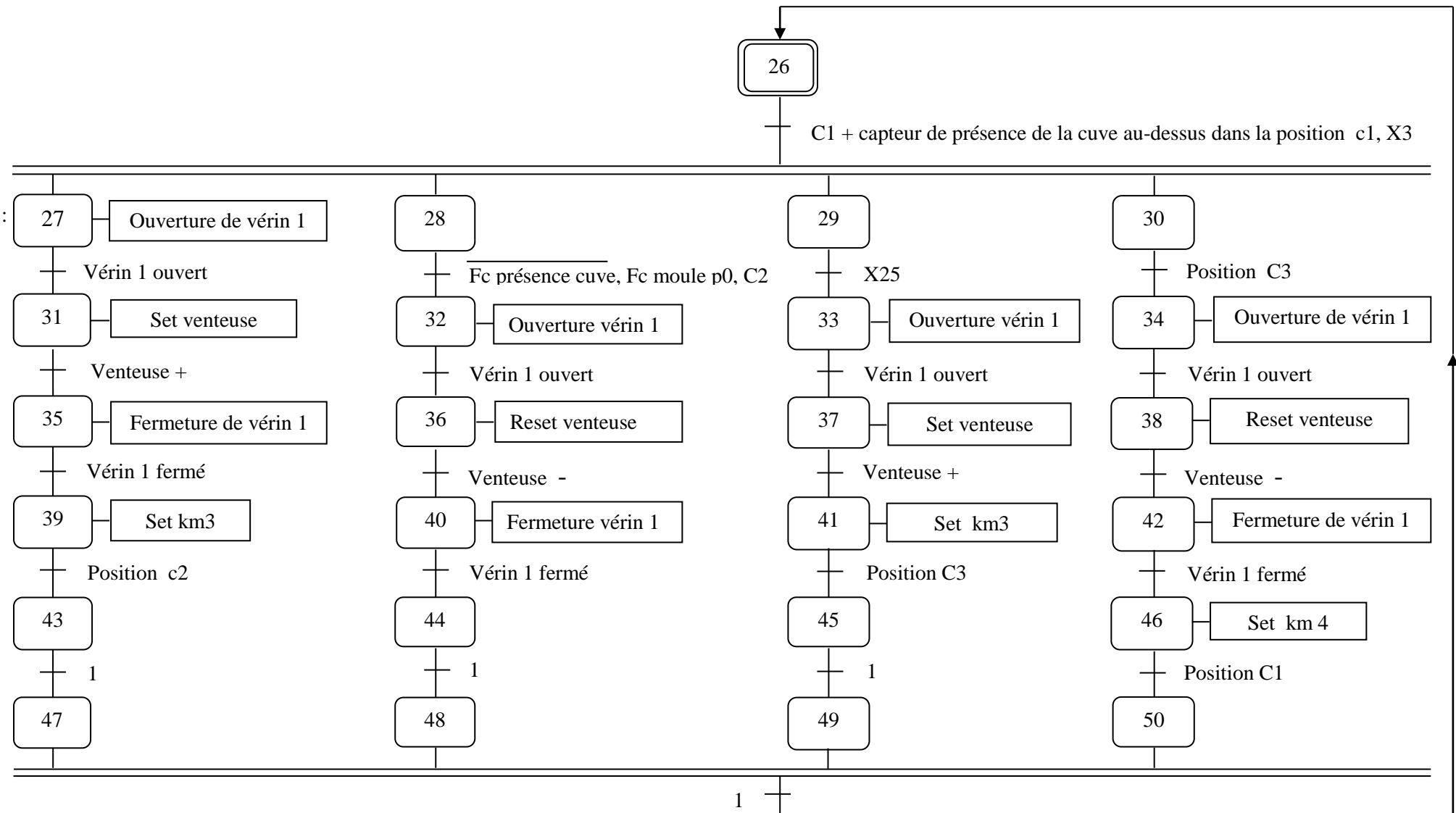
- Présence de la cuve.
- Présence de l'énergie (source) pneumatique.
- Vérin de frein fermé.
- Vérin de bras ouvert.
- Vérins 1 fermé.
- Vérin de coupage tube fermé.
- Vérin de coupage ruban adhésif fermé.
- Vérin de moule fermé.
- Charriot 1 en position initial.

- Charriot 2 en position initial.
- Moule en position zéro.
- Présence de tube.
- Capteur de présence d'air actionné.

### A. Grafcet Niveau 1







### **III.6. Conclusion**

Au terme de ce chapitre nous concluons que le GRAFCET est un outil de modélisation qui permet facilement le passage d'un cahier des charges fonctionnel à un langage d'implantation optionnel, il permet la description du comportement attendu de la partie commande d'un système automatisé, comme il permet aussi de créer un lien entre la partie commande et la partie opérative.

Ainsi, le GRAFCET facilite considérablement le passage de la description à la modélisation et nous permettra au chapitre suivant d'aborder la programmation de la partie opérative qui pilotera le procédé et ce à l'aide du Step7.

# Chapitre IV

Développement de la solution programmable

## **IV.1. Introduction**

L'automate programmable industriel (API) est aujourd'hui, le constituant le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie, car il répond à tous les besoins d'adaptation et de flexibilité pour un grand nombre d'opérations. Cette émergence est due en grande partie, à la puissance de son environnement de développement et aux larges possibilités d'interconnexions.

L'automatisation de n'importe quel processus a pour but de rendre le système plus rapide, fiable et peut agir devant n'importe quel type de contraintes ou de problème aléatoire.

L'automatisation consiste à rendre automatique les opérations qui exigeaient auparavant l'intervention humaine.

## **IV.2. Définition d'un automate programmable industriel (API)**

L'automate programmable industriel (API) ou Programmable Logic Controller (PLC) est un appareil électronique programmable.

Il est défini, suivant la norme française EN-61131-1, adapté à l'environnement industriel, et réalise des fonctions d'automatisme pour assurer la commande de prés-actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques, par exemple, on cite les automates SIMATIC S7.

La gamme SIMATIC S7 comprend les systèmes d'automatisation suivants :

- S7-200 : Un micro-automate compact de l'entrée de gamme.
- S7-300 : Un micro-automate modulaire de milieu de gamme.
- S7-400 : Il couvre le haut et très haut de gamme.

## **IV.3. Objectifs de l'automate dans les systèmes automatisés industriels**

Un système automatisé est un ensemble de moyens matériels et logiciels constituant la partie automatisme, communication et de production d'une installation industrielle.

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée par le système. Ces éléments sont exprimables en termes d'objectif pour :

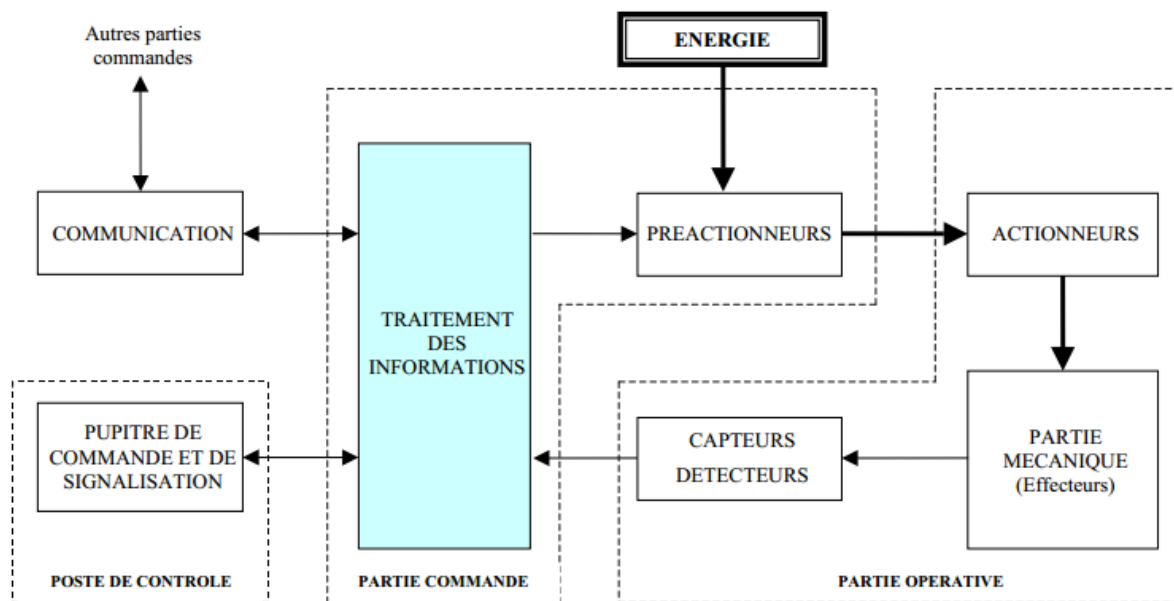
- Une meilleure rentabilité.
- Une meilleure compétitivité.
- Améliorer la flexibilité de production.



- Améliorer la qualité du produit grâce à une meilleure respectabilité de la valeur ajoutée.
- Faciliter la maintenance de l'installation par un diagnostic rapide.
- Adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation des charges lourdes, etc.).

### IV.3.1. Structure générale d'un système automatisé

Tout un système automatisé peut se décomposer selon le schéma de la figure IV.1 [8].



**Figure IV.1 :** Structure d'un système automatisé

Les systèmes automatisés utilisés dans les secteurs industriels possèdent une structure de base identique, ils sont constitués de plusieurs parties et qui sont reliées entre elles, à savoir de :

- **Partie opérative(PO) :** Cette partie puissante comporte les éléments (pré actionneurs, détecteurs).
- **Partie commande (PC) :** Elle élabore les ordres nécessaires à l'exécution d'un processus et reçoit en retour des comptes rendus qui l'informent sur l'état des opérations effectuées.
- **Partie relation (PR) :** Ou pupitre de dialogue, c'est un appareil de contrôle qui permet une intervention rapide sur la machine, par exemple : présélection de valeur de consigne, la lecture des données machines.

### **IV.3.2. Système de commande**

L'automaticien dispos de nombreux outils technologiques pour réaliser l'organe de commande de son système qui est classé habituellement en deux catégories soit :

#### **a) La logique câblée**

C'est celle utilisée au niveau de la l'ENIEM, sa mise en œuvre de la théorie de l'expérience. Cette technologie est simple connue croisée, mais toute modification dans le choix de fonctionnement de l'installation, entraîne :

- Une augmentation du nombre de relais et de fils (cout élève).
- Une intervention dans le câblage (main d'œuvre).

#### **b) La logique programmée**

C'est la solution proposée, elle utilise un automate programmable industriel (API). L'encombrement se trouve réduit et la recherche de panne et plus ou moins facile.

- Main d'œuvre réduit lors du câblage.
- Modification possible sans intervention sur le câblage.
- Erreur de correction réduite.
- Cout faible.

### **IV.4. choix d'un automate programmable industriel**

Les critères de choix d'un API sont [1] :

- Le nombre et la nature des entrée/sorties.
- La nature de traitement (temporisation, comptage..).
- Les capacités de traitement du processus (vitesse, taille du programme, opérations, temps réel,...).
- Le dialogue (la console détermine le langage de programmation).
- Les moyens de sauvegarde du programme (disquettes, cassettes..).
- La fiabilité et la robustesse.
- L'immunité aux parasites.

## IV.5. Présentation générale de l'automate S7-300

### IV.5.1. Aspect externe

L'automate S7-300 est fabriqué par la famille SIMATIC. Il est de conception modulaire, une vaste gamme de module est disponible. Ces modules peuvent être combinés selon les besoins lors de la conception d'une solution d'automatisation [8].

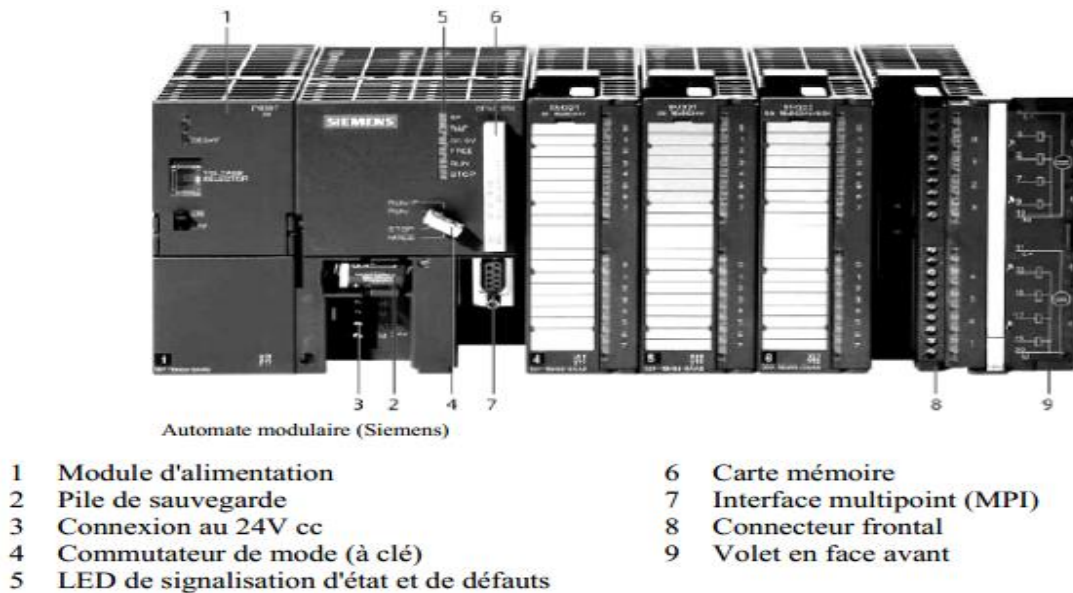


Figure IV.2 : Automate modulaire SIEMENS

### IV.5.2. Aspect interne

L'architecture interne de l'automate est donnée par la figure ci-dessous [8] :

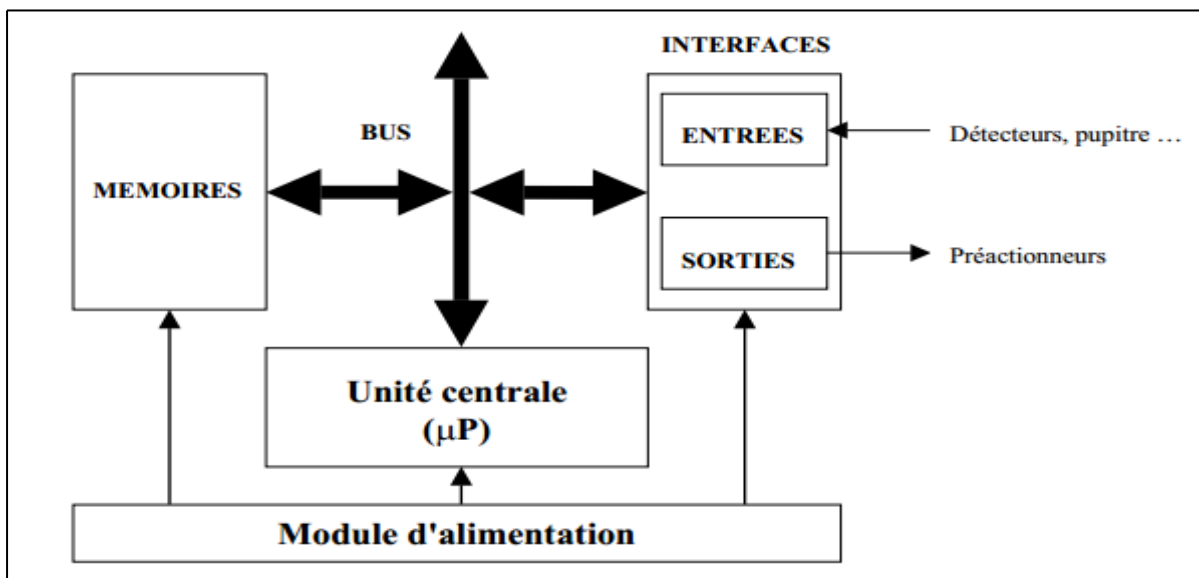


Figure IV.3 : Architecture interne d'un automate

## IV.6. Caractéristique de l'automate S7-300

L'automate S7-300 est spécifié par les caractéristiques suivantes :

- Gamme diversifiée de la CPU
- Gamme complète du module
- Possibilité d'extension jusqu'à 32 modules
- Bus de fond de panier intégré en module
- Possibilité de mise en réseaux avec MPI, PROFIBUS, ETHERNET, PRFINET.
- Raccordement central de la PG avec accès à tous les modules.
- Liberté de montage aux différents emplacements.
- Configuration et paramétrage à l'aide de l'outil configuration matériels.

## IV.7. Constitution de l'automate S7-300

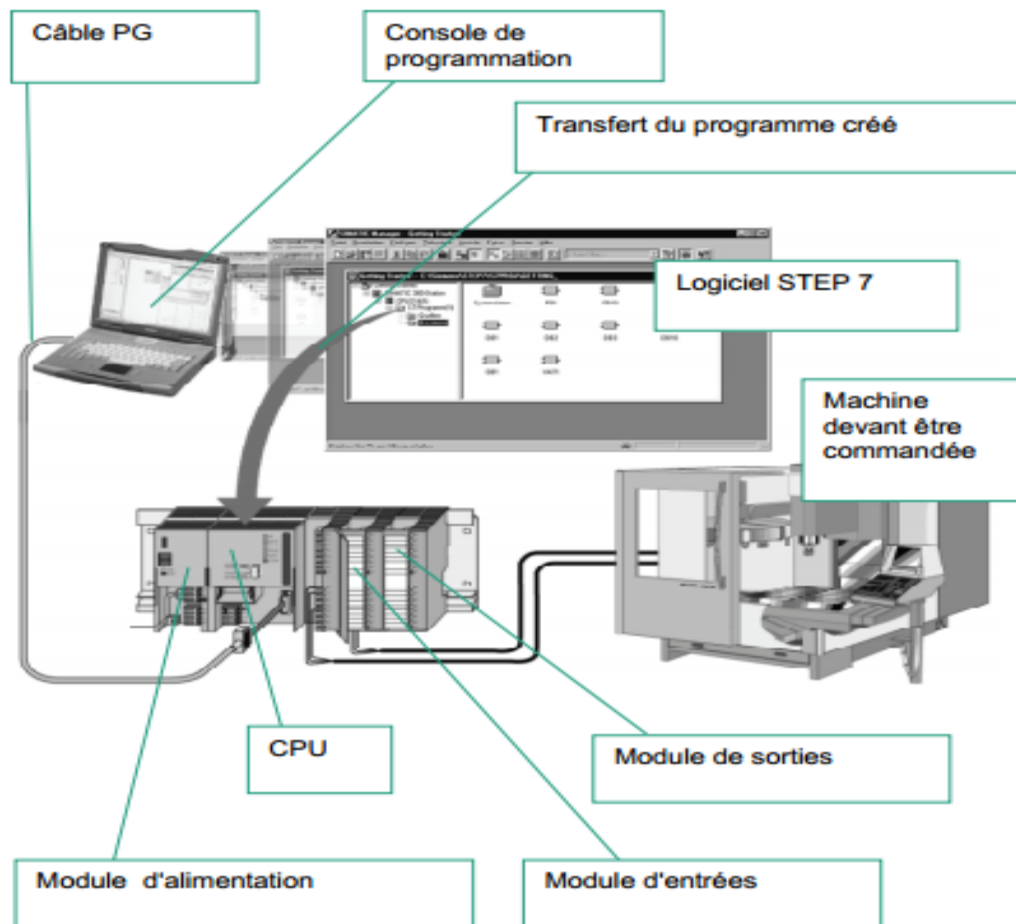
L'automate programmable S7-300 est un système d'automatisation modulaire offrant la gamme des modules suivants :

- Module d'alimentation (PS) 2A, 5A, 10A.
- Unité centrale CPU traitement doté d'une mémoire.
- Module de signaux (SM) entrées et de sorties TOR et analogique.
- Module de simulation (SM 374).
- Le châssis (rack).
- Module coupleur (IM) pour configuration multi rangée du S7-300.
- Module pour fonction (FM) spéciales (coupleur rapide 500khz).
- Processeur de communication (CP) pour la communication avec d'autres éléments de réseau.

## IV.8. Programmation de l'automate S7-300

A chaque famille d'automates correspond à un langage spécifique de programmation.

Les automates de la famille SIEMENS sont programmés grâce au logiciel STEP 7 via une console de programmation au PC et sous un environnement WINDOWS. La figure suivante illustre l'ensemble des éléments entrant dans l'automatisation d'un système [6] :



**Figure IV.4 :** Vue d'ensemble des éléments entrant dans l'automatisation d'un système

Le logiciel STEP 7 offre les possibilités suivantes :

- Configuration et paramétrage du matériel et de la communication.
- Création et gestion des projets.
- Gestion des mnémoniques
- Test de l'installation d'automatisation.

#### **IV.8.1. Configuration et paramétrage de l'automate**

Au sens logiciel du terme, on entend par configuration, l'organisation des modules dans une table de configuration. C'est-à-dire on définit les modules mis en œuvre dans la solution d'automatisation, ainsi que les adresses permettant d'y accéder depuis le programme utilisateur.

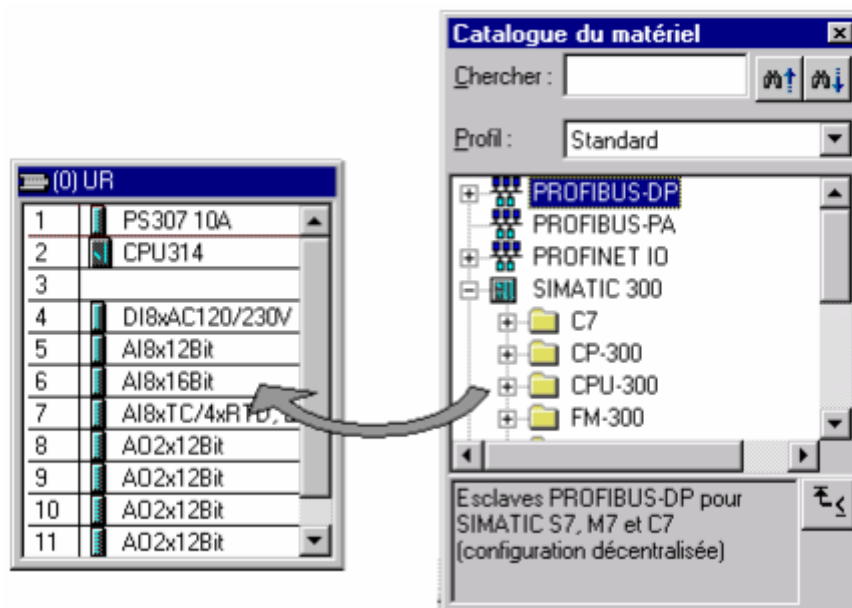
A l'image de la réalité matérielle de notre S7-300, il s'agit de ranger les modules sur un châssis de façon virtuelle avec le logiciel STEP 7. L'emplacement dans la table de configuration doit correspondre à l'emplacement réel sur le profilé-support [6], [9].

### IV.8.2. Marche à suivre pour la configuration d'une station :

Quelle que soit la technique de configuration d'une station, il faut toujours vous en tenir aux étapes suivantes pour la configurer :

1. sélectionnez un composant matériel dans la fenêtre "catalogue du matériel".
2. amenez le composant sélectionné dans la fenêtre de station en utilisant la fonction glisser-lâcher.

La figure suivante illustre les manipulations de base :



**Figure IV.5 :** Manipulation de base pour une configuration matérielle

#### IV.8.2.1. Adressage des modules du S7-300

On a deux types d'adressage :

- Adressage lié à l'emplacement : Il s'agit du mode d'adressage par défaut, c'est-à-dire que le STEP 7 effectue à chaque numéro d'emplacement une adresse de défaut de modèle fixé à l'avance.
- Adresse libre : Dans ce mode d'adressage, il faut effectuer à chaque mode une adresse de notre choix, pourvue qu'elle soit continue dans la page d'adressage possible de la CPU.

#### **IV.8.2.2. Les 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 logiques 0 ou 1.

Chaque automate programmable dispose d'un grand nombre de mémentos (S7-300 dispose de 2048 bits de mémentos). On programme ces derniers comme des sorties. En cas de panne de la tension de service, le contenu sauvegarde dans les mémentos est perdu.

#### **IV.8.2.3. Représentation de programme**

En STEP 7, un programme dans l'outil 'programme des blocs CONT/LIST/LOG' peut être programmé et représenté dans trois catégories différentes :

- Schéma à contact CONT.
- Plan logique LOG.
- Liste d'instruction LIST.

##### **a).Schéma à contact CONT**

La présentation en mode de programmation CONT, s'inspire des schémas de circuit à contact. Ils sont rassemblés dans des réseaux en segment formant un bloc de code.

##### **b).Plan logique LOG**

C'est un langage de programmation graphique qui utilise des symboles de l'algèbre de BOOL pour représenter les instructions.

##### **c).Liste d'instruction LIST**

C'est un langage de programmation textuel proche de la machine. Les différentes instructions correspondent aux étapes par lesquelles la CPU traite le programme.

Le tableau IV.1 illustre les trois modes de langage de programmation :

Schéma a contact	Liste d'instruction	Schéma logique
	<pre> U E 0.0 U E 0.1 U E 0.2 O E 0.3 = A 0.0 </pre>	

Tableau IV.1 : Les modes de programmation

### IV.8.3. programmation sous STEP7

Deux programmes différents s'exécutent dans une CPU à savoir :

#### IV.8.3.1. Systèmes d'exploitation

Le contenu dans chaque CPU, organise toutes les fonctions et les procédures qui ne sont pas liées une tâche d'automatisation spécifique. Ces tâches sont :

- Le déroulement de démarrage et de redémarrage.
- L'actualisation de la mémoire image des entres et l'émission de la mémoire d'image des sorties.
- L'appel du programme utilisateur.
- L'enregistrement des alarmes et l'appel des OB (boc d'organisation) d'alarmes.
- La détection et le traitement des erreurs.
- La gestion des zones de mémoires.
- La communication avec des consoles de programmation et avec d'autres partenaires de communication.
- La modification des paramètres du système d'exploitation.

#### IV.8.3.2. Programme utilisateur

On doit créer notre programme utilisateur et le charge dans le CPU. Il contient toutes les fonctions nécessaires au traitement de notre tâche d'automatisation spécifique. Il doit par conséquent [8] :

- Déterminer les conditions pour le démarrage et le redémarrage du CPU.



- Traiter les données du processus.
- Réagir aux alarmes.
- Traiter les perturbations dans le déroulement normal du programme.

#### **a).bloc dans le programme utilisateur**

Un bloc correspond à une partie du programme utilisateur, délimitée par sa fonction, sa structure et par son application.

Dans STEP7 il existe d'une part :

- Des blocs des codes (FB, FC, OB, SFB, SFC).
- Des blocs des données (DB, SDB).
- Et d'autres types de données utilisateurs (UDT).

Les blocs sont rangés dans le dossier « blocs » sous le programme S7 :

#### **❖ Bloc de code**

Il existe les blocs de code suivants :

##### **Bloc d'organisation (OB)**

Il constitue l'interface entre le système d'exploitation du CPU et le programme utilisateur, c'est dans ce bloc que l'on définit l'ordre de traitement du programme utilisateur. Cela revient à dire que l'exécution d'un OB peut être interrompue par l'appel d'un autre OB plus prioritaire.

##### **Bloc fonctionnel (FB)**

Une FB permet de transmettre des paramètres dans le programme utilisateur. Il contient un programme qui est exécuté quand ce bloc qui est appelé par autre bloc de code. Le bloc FB facilite la programmation des fonctions complexes souvent utilisées. Un FB possède une mémoire (bloc de donnée d'instance).

##### **Bloc de donnée d'instance (SDB)**

Un DB d'instance mémorise les paramètres formels et les données statiques de blocs fonctionnels. Il peut être associé à un appel de FB ou à une hiérarchie d'appel de bloc fonctionnel.

##### **Fonction (FC)**

Une fonction est un bloc sans mémoire. Les données sont perdues à l'achèvement de la fonction. Les FC peuvent faire appel à des blocs de données globaux pour la sauvegarde des données.

### Les SFB et SFC

Les blocs fonctionnels systèmes (SFB) et fonction système sont intégrés au système d'exploitation de la CPU-S7 pour créer des messages.

#### ❖ Bloc des données

Il constitue les blocs suivant :

##### Bloc des données (BD)

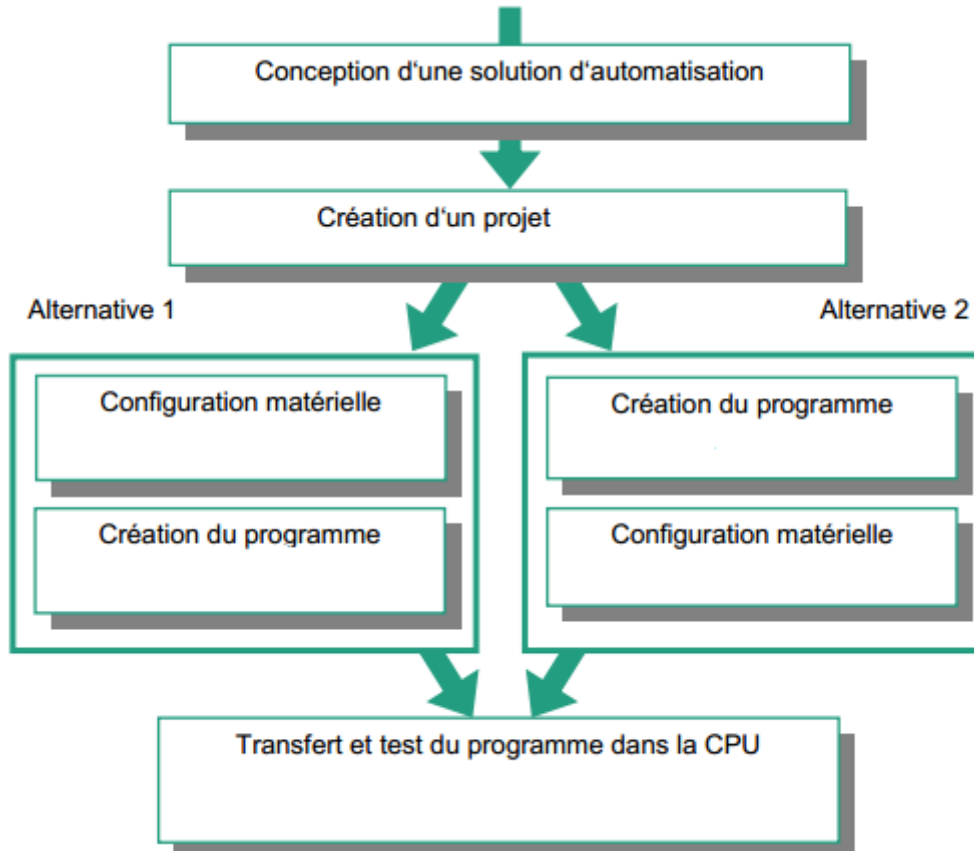
Il s'agit d'une zone dans le programme utilisateur qui contient des données utilisateurs.

##### Bloc de donnée système (SDB)

C'est une zone de mémoire dans la CPU contenant des paramètres et des paramètres de bloc.

#### IV.8.4. Création du projet

Avant de créer un projet, deux approches sont possibles comme le montre la figure suivante [6], [9] :



**Figure IV.6 :** Approches possibles pour créer un projet

Un projet contient la description complète de l'automatisme. Il comporte donc deux grandes parties : la description du matériel et la description du fonctionnement (le programme).

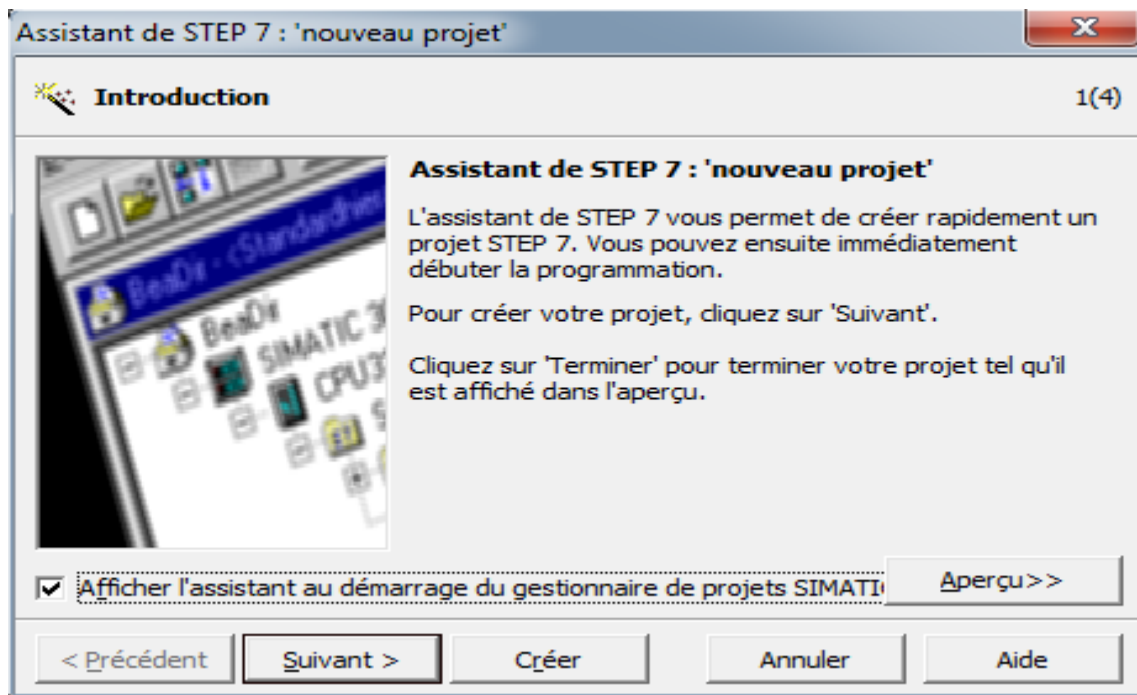
En entrant dans STEP 7 (double clic l'icône SIMATIC Manager qui va lancer l'assistant de STEP 7), l'assistant propose de créer un nouveau projet.

Dans notre cas, nous avons procédé comme suit :



1. lancer SIMATIC Manager par un double clic sur son icône.

2. la fenêtre suivante permet la création d'un projet.



**Figure IV.7 :** Fenêtre de création d'un projet

3. On clique sur suivante, la fenêtre suivante nous permet de choisir la CPU comme il est montré par la figure ci-dessous.

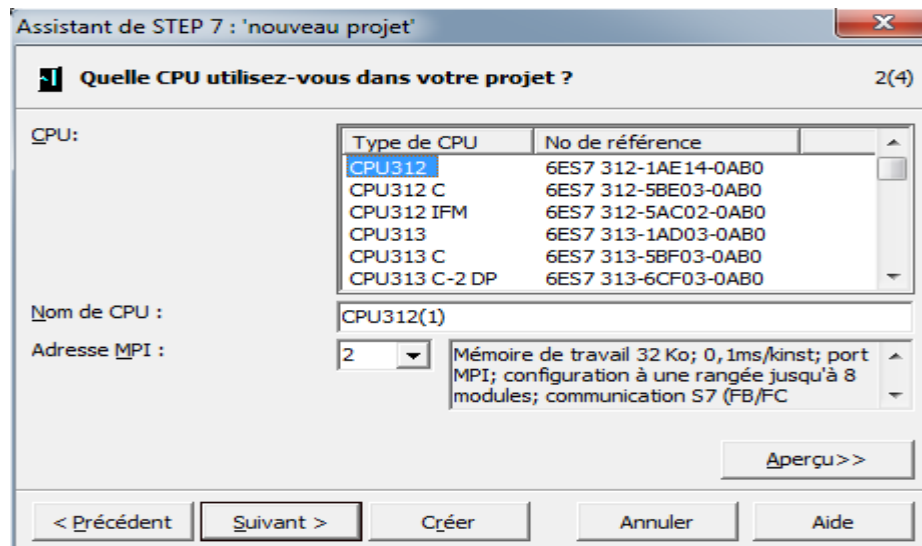


Figure IV.8 : CPU312 sélectionnée

4. Après validation de la CPU, une fenêtre apparaît, qui permet de choisir le bloc et le langage de programmation à insérer.

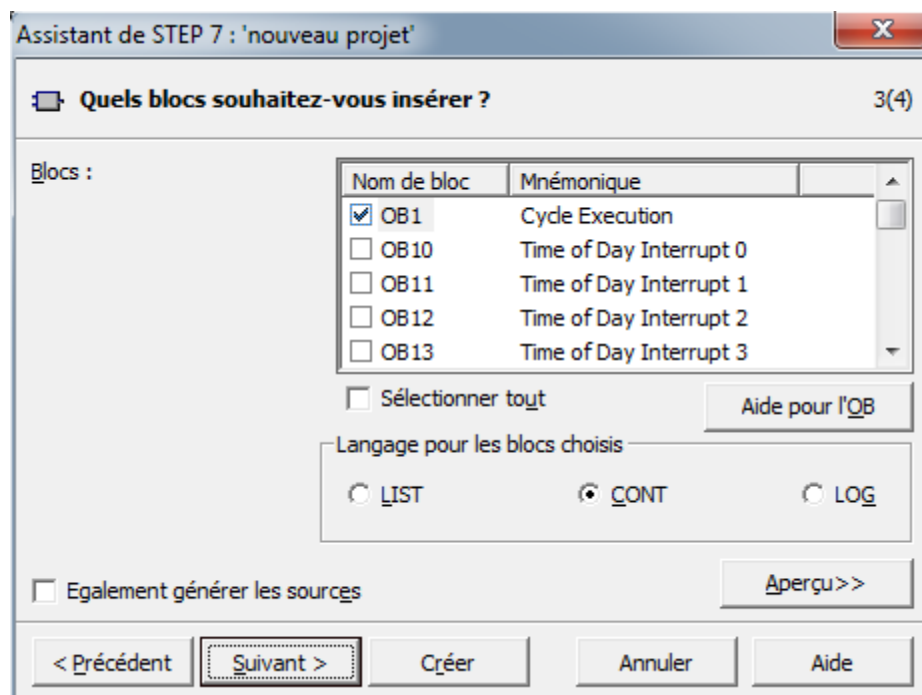


Figure IV.9 : Sélection des blocs et du langage de programmation (CONT)

5. Nomination de projet.

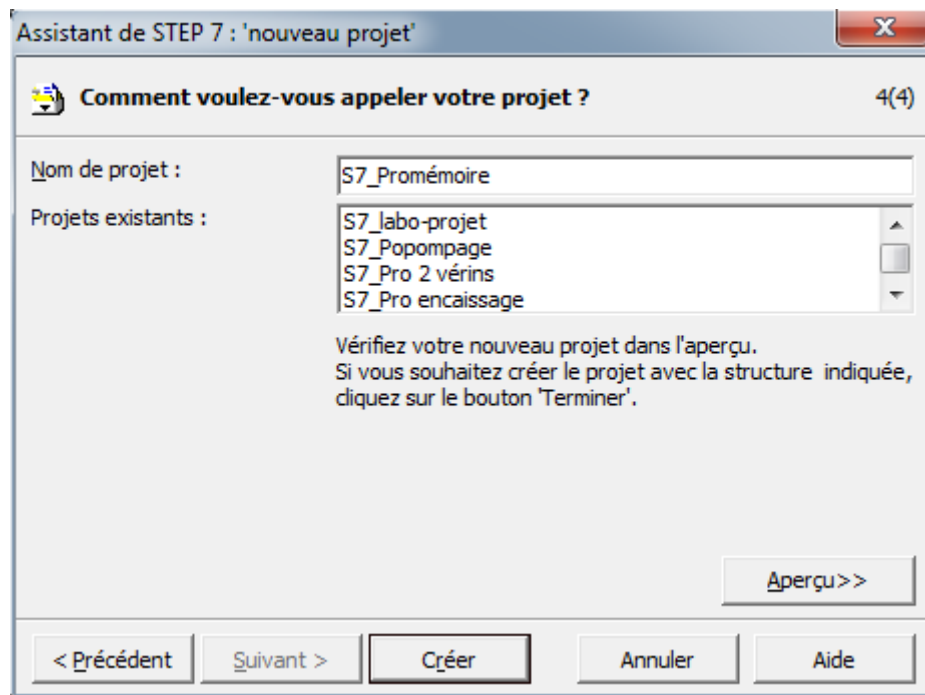


Figure IV.10 : Nomination de projet

6. Une fois le projet créé, il est nécessaire de configurer le matériel à utiliser comme le montre la figure suivante.

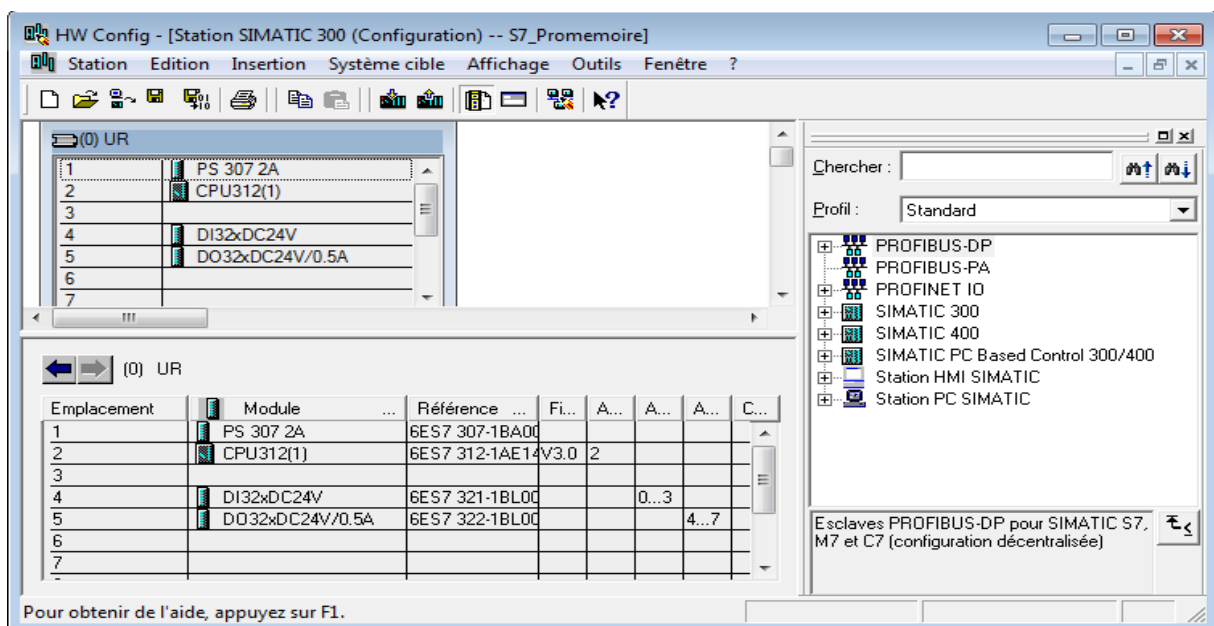
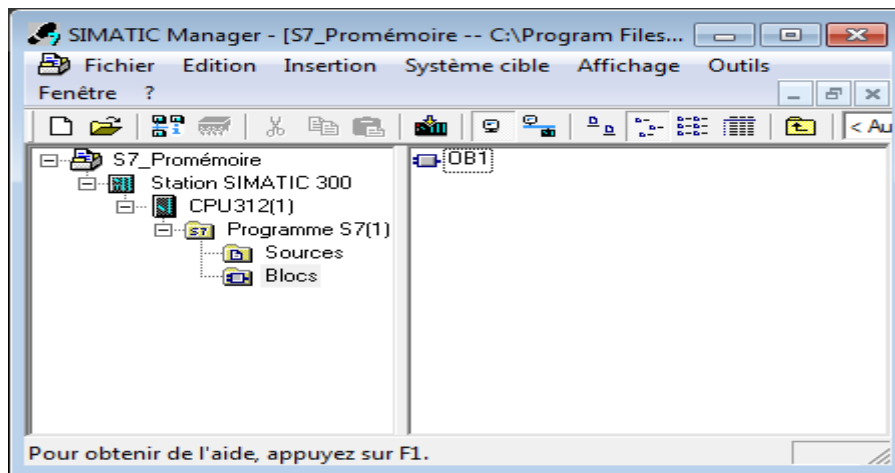


Figure IV.11 : Configuration matériels

7. Ensuite on passe au programme utilisateur que nous avons écrit pour commander l'enrouleuse, ce dernier est composé d'objets définis dans l'environnement de STEP 7.



**Figure IV.12 :** Vue des composants de notre projet

#### IV.8.5. Saisis des mnémoniques

Saisir les mnémoniques est très utile, il vaut mieux saisir un programme entièrement en symbole qu'en adressage absolu, c'est beaucoup plus lisible et compréhensible.

Il suffit d'aller dans la table des mnémoniques et faire entrer les différents éléments, le nom du symbole, son adresse réel, son type et son commentaire. (Voir la figure IV.13).

	Etat	Mnémonique	Opéra /	Type de d	Commentaire
1		lampe temoine cycle	A 4.0	BOOL	
2		lampe temoine reset	A 4.1	BOOL	
3		moteur traction tube	A 4.2	BOOL	
4		moteur coupage tube	A 4.3	BOOL	
5		frein de la bobine	A 4.4	BOOL	
6		moteur tournage moule	A 4.5	BOOL	
7		ouverture bras	A 4.6	BOOL	
8		fermeture bras	A 4.7	BOOL	
9		rele reset conte	A 5.0	BOOL	
1		coupage rubain adhesif	A 5.1	BOOL	
1		ouverture moule	A 5.2	BOOL	
1		fermeture moule	A 5.3	BOOL	
1		km1	A 5.4	BOOL	moteur de charriot 1 tourne sens d'horloge
1		km2	A 5.5	BOOL	moteur charriot 1 tourne sens inverse d'horl...
1		km3	A 5.6	BOOL	moteur charriot 2 tourne sens d'horloge
1		km4	A 5.7	BOOL	moteur charriot 2 tourne sens inverse d'horl...
1		ouverture de vérin1	A 6.0	BOOL	
1		fermeture de vérin 1	A 6.1	BOOL	
1		aspiration des venteuse	A 6.2	BOOL	
2		relâchement des venteuse	A 6.3	BOOL	
2		lampe t prés de préssion	A 6.4	BOOL	
2		auxiliaires	E 0.0	BOOL	
2		start cycle	E 0.1	BOOL	
2		stop cycle	E 0.2	BOOL	
2		reset	E 0.3	BOOL	
2		START	E 0.4	BOOL	
2		ouvert vanne de préssion	E 0.5	BOOL	
2		vanne d'air fermé	E 0.6	BOOL	
2		bp d'alimentation	E 0.7	BOOL	
3		fc charriot en bas	E 1.0	BOOL	
3		fc charriot en haut	E 1.1	BOOL	
3		fc tube coupé	E 1.2	BOOL	
3		fc tube non coupé	E 1.3	BOOL	
3		fc bras ouvert	E 1.4	BOOL	
3		fc bras fermé	E 1.5	BOOL	
3		Capteur de prés tube mou	E 1.6	BOOL	
3		capteur compteur moule	E 1.7	BOOL	
3		fc moule fermé	E 2.0	BOOL	
4		fc moule ouvert	E 2.1	BOOL	
4		fc présence cuve sur mou	E 2.2	BOOL	fc presence cuve sur moule
4		fc moule en position 0	E 2.3	BOOL	
4		fc rubain adhesif coupé	E 2.4	BOOL	
4		fc rubain nn coupé	E 2.5	BOOL	
4		fc frein de la bobine +	E 2.6	BOOL	
4		fc frein de la bobine -	E 2.7	BOOL	
4		fc présence cuve dans c1	E 3.0	BOOL	
4		arrêt d'urgence	E 3.1	BOOL	
4		Vérin 1 en sortie	E 3.2	BOOL	
5		Vérin 1 fermé	E 3.3	BOOL	
5		venteuse -	E 3.4	BOOL	
5		venteuse +	E 3.5	BOOL	
5		position de charriot C1	E 3.6	BOOL	
5		position de charriot C2	E 3.7	BOOL	
5		position de charriot C3	E 4.0	BOOL	
5		bp avance tube	E 4.1	BOOL	
5		ARRET	E 4.2	BOOL	
5		m10.0	M 0.0	BOOL	
5		m20.0	M 0.1	BOOL	
6		m30.0	M 0.2	BOOL	
6		m40.0	M 0.3	BOOL	

Figure IV.13 : La table des mnémoniques

## **IV.9. Test et simulation du programme**

### **IV.9.1. Introduction générale sur le S7-PLCSIM**

L'utilisation de simulation de modules S7-PLCSIM, nous permet d'exécuter et de tester notre programme dans un automate programmable (API), que nous simulons dans un ordinateur ou dans une console de programmation.

Le S7-PLCSIM dispose d'une interface simple qui nous permet de visualiser et de forcer les différents paramètres utilisés par le programme (activer ou désactiver une entrée).

Nous avons également la possibilité de mettre en œuvre les divers applications du logiciels STEP 7, tout en exécutant le programme dans l'API de simulation, prenant l'exemple de test de bloc afin de visualiser les variables d'entrées et de sorties [7].

### **IV.9.2. Etats de fonctionnement de la CPU**

#### **IV.9.2.1. Etat de marche (RUN-P)**

La CPU exécute le programme tout en vous permettant de le modifier, de même que ses paramètres. Afin de pouvoir utiliser les applications de STEP7 pour forcer un paramètre quelconque de programme durant son exécution, vous devez mettre la CPU à l'état RUN-P.

#### **IV.9.2.2. Etat 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), vous ne pouvez ni charger un programme, ni utiliser les applications de STEP7 pour forcer un paramètre quelconque.

#### **IV.9.2.3. Etat d'arrêt (STOP)**

La CPU n'exécute pas le programme. Contrairement à l'état d'arrêt (STOP) de CPU réel, les sorties ne prennent pas de valeurs de sécurité prédéfinies, mais elles conservent l'état auquel elles étaient lorsque la CPU a passé à l'état arrêt(STOP).

#### **IV.9.2.4. Indicateurs de la CPU**

La fenêtre CPU dispose d'une série d'indicateurs qui correspondent aux voyants de signalisation sur une CPU réelle :


- SF (erreur système) vous avertit que la CPU a détecté une erreur système, entraînant un changement d'état de fonctionnement.



- DP (périphérique décentralisé ou E/S éloignées) indique l'état de la communication avec les entrées/sorties décentralisées (éloignées).
- DC (alimentation) indique si la CPU sous ou hors tension.
- RUN indique que la CPU se trouve à l'état marche.
- STOP indique que la CPU se trouve à l'état d'arrêt.

### IV.9.3. Mise en route du logiciel S7-PLCSIM

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

- Ouvrir le gestionnaire de projet SIMATIC.
- Cliquez sur  pour simulation de modules.

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

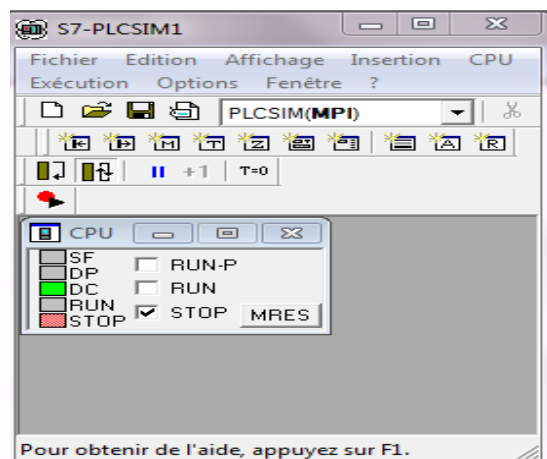







Figure IV.14 : Fenêtre du S7-PLCSIM

- Dans le gestionnaire de projet SIMATIC, cherchez le projet-exemple, projet fin d'étude.
- dans ce projet cherchez le dossier Blocs.
- Dans le gestionnaire de projet SIMATIC, cliquez sur  pour charger le dossier blocs dans l'API de simulation.

Pour la visualisation, on crée de nouvelles fenêtres :

- Cliquez sur  pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des entrées.

- Cliquez sur  pour créer une fenêtre dans laquelle nous pouvons visualiser et forcer des variables dans la zone mémoire des sorties.
- Cliquez sur  pour créer une fenêtre dans laquelle nous pouvons visualiser et forcer les temporisations utilisées par le programme.
- Mettre la CPU de simulation en marche en cliquant sur l'une des cases à cocher RUN ou RUN-P.

Pour sauvegarder la version actuelle de la simulation d'API, cliquez sur .

Une fois toutes les fenêtres d'entrées et de sorties sont prêtes, nous activons les entrées voulues pour lire l'état des sorties. (Voir la figure IV.15).

La figure IV.15 présente le simulateur S7-PLCSIM qui est configuré par entrées, les sorties et les mémentos.

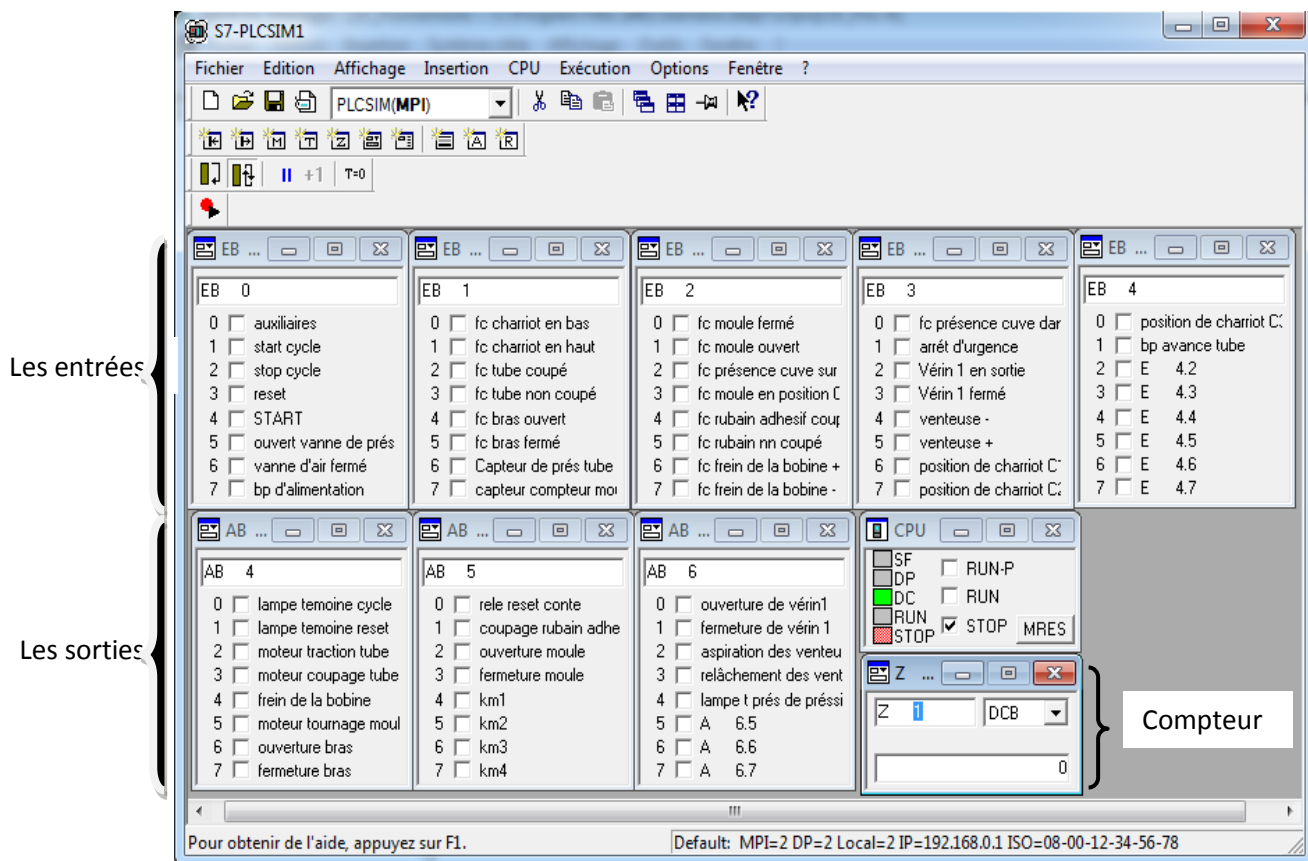


Figure IV.15 : Simulation avec S7-PLCSIM


### IV.9.3.1 Simulation du programme projet fin d'études :

La simulation du projet fin d'études est faite en deux étapes :

- **Etape 1** : Simulation du programme par bocs.
- **Etape 2** : Simulation du cycle complet.

### IV.9.3.2. Visualisation de l'état du programme

Pour le bloc OB1 :

Après le chargement du programme de OB1 dans la CPU du simulateur et la mise de cette dernière en mode « RUN » le STEP 7 nous permet de visualiser l'état du programme, en cliquant sur l'icône . Ceci est illustré par la figure IV.16.

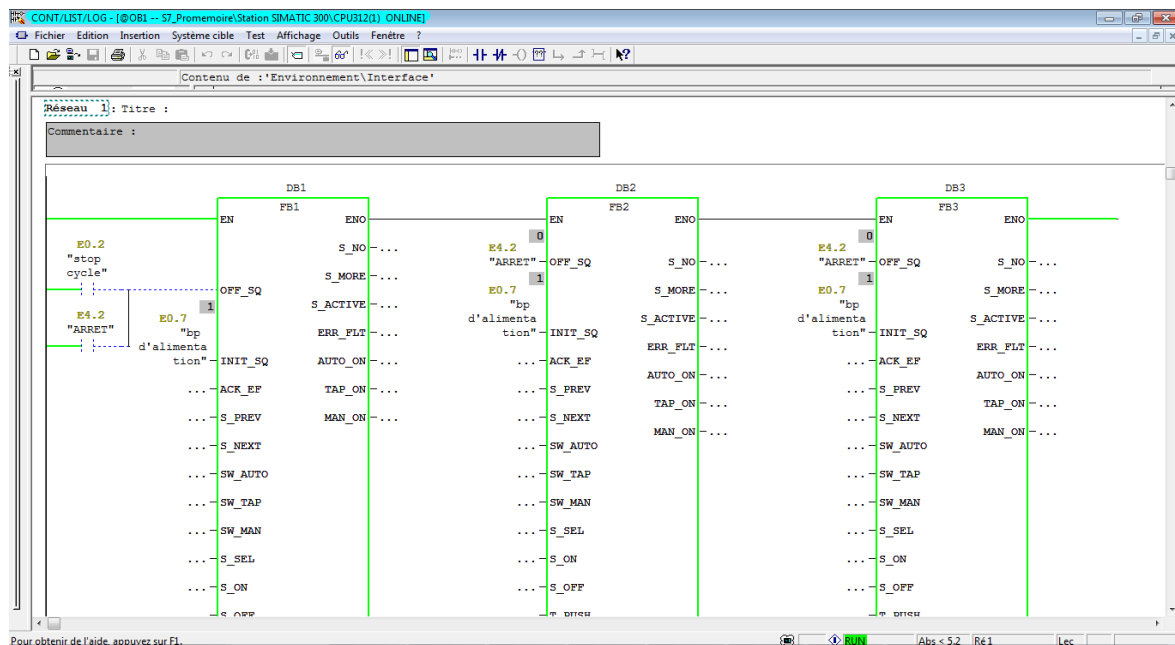


Figure IV.16 : Visualisation de l'état du programme

## IV.10. Conclusion

Les solutions programmées nous procurent plusieurs avantages tels que la flexibilité et la facilité d'extension de ses modules.

Le test de notre programme sur l'enrouleuse a été fait par la fonction de la visualisation et la table des variables sur un S7-300. Pour cela on a utilisé cinq modules d'entrées, trois modules de sorties et quatre mémentos.

Les essais effectués, ont confirmé que la solution programmée nous a permis de donner les résultats envisagés.

# Chapitre V

## Supervision du système avec Win CC

### **V.1. Introduction**

Lorsque la complexité des processus augmente et que les machines et les installations doivent répondre à des spécifications de fonctionnalité toujours plus sévères, l'opérateur a besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'Interface Homme-Machine (IHM). Un système IHM constitue l'interface entre l'homme (opérateur) et le processus (machine/installation). Le contrôle proprement dit du processus est assuré par le système d'automatisation.

Une fois le pupitre mis sous réseau, il permet :

- De visualiser l'état des machines, capteurs.
- D'agir sur les moteurs.

### **V.2. Présentation du logiciel Win CC flexible 2008**

Win CC flexible 2008 est l'Interface Homme-Machine (IHM) idéale pour toutes les applications au pied de la machine et du processus dans la construction d'installations et de machines. De par sa conception générale, WinCC Flexible permet de disposer d'un logiciel d'ingénierie pour tous les terminaux d'exploitation SIMATIC HMI, du plus petit pupitre Micro jusqu'au multi panel, ainsi que d'un logiciel de supervision Runtime pour les solutions monoposte basées sur un PC et tournant sous Windows XP/Vista.

Les projets peuvent être portés sans conversion et sont exécutables sur diverses plateformes IHM. Grâce à des logiciels et à des projets multilingues, le WinCC Flexible peut être utilisé dans le monde entier.

WinCC Flexible comprend des outils d'ingénierie innovants pour la configuration cohérente de tous les terminaux d'exploitation SIMATIC HMI. Il apporte une efficacité de configuration maximale : des bibliothèques contenant des objets préconfigurés, des blocs d'affichage réutilisables, des outils intelligents allant jusqu'à la traduction automatisée des textes dans le cadre de projets multilingues.

### **V.3. Définition de la supervision industrielle**

La supervision est une forme évoluée de dialogue homme/machine. Elle présente beaucoup d'avantages pour le processus industriel de production. Elle facilite à l'opérateur la surveillance de l'état de fonctionnement d'un procédé ainsi que son contrôle. Elle permet grâce à des synoptiques préalable, créés et configurés à l'aide d'un logiciel de supervision, d'intégrer et de visualiser en temps réel toutes les étapes nécessaires à la fabrication d'un

produit et de détecter les problèmes qui peuvent survenir en cours de fonctionnement dans une installation industrielle.

La supervision consiste en de nombreuses fonctions :

- Elle répond à des besoins nécessitant en générale une puissance de traitement importante.
- Elle assure la communication entre les équipements d'automatismes et les outils informatiques d'ordonnancement et de gestion de production.
- Elle coordonne le fonctionnement d'un ensemble de machines enchainées constituant une ligne de production, en assurant l'exécution d'ordres communs et des tâches telles que la synchronisation.
- Elle assiste l'opérateur dans les opérations de diagnostic et de maintenance.
- Elle surveille les procédés industriels à distance.
- Elle permet la simulation de programme avant leur mise en œuvre grâce au logiciel WinCC Flexible.

#### **V.4.constitution d'un système de supervision**

Un système de supervision est généralement composé d'un moteur central (logiciel), auquel se rattachent des données provenant des équipements (automate). Ce moteur central assure l'affichage, le traitement et l'archivage des données. Ainsi que la communication avec d'autres périphériques.

- **Module de visualisation (affichage)**

Il permet d'obtenir et de mettre à la disposition de l'opérateur toutes les informations nécessaires à l'évolution du procédé.

- **Module d'archivage**

Son rôle est de mémoriser les données (alarmes et évènements) pendant une longue période. Il permet l'exploitation des données pour les applications spécifiées à des fins de maintenance ou de gestion de production

- **Module de traitement**

Il permet de mettre en forme les données afin de les présenter via le module de visualisation aux opérateurs sous une forme prédéfinie.

- **Module de communication**

Il assure l'acquisition et le transfert des données. Il gère la communication avec les automates programmables industriels et autres périphériques. Il donne la possibilité de :

- ✓ De modifier la configuration même après la mise en vente.
- ✓ D'avoir la comptabilité avec le réseau internet qui permet de réaliser des solutions basées sur le web (contrôle-commande à distance).

## **V.5. Etapes de mise en œuvre**

Pour créer une interface Homme/Machine, il faut prendre préalablement connaissance des éléments de l'installation ainsi que le logiciel de programmation de l'automate utilisé.

Nous avons créé l'interface pour la supervision à l'aide du logiciel WinCC Flexible qui est le mieux adapté pour le matériel de la gamme SIEMENS.

### **V.5.1. Etablir une liaison directe**

La première chose à effectuer est de créer une liaison directe entre le WinCC et notre automate. Ceci pour que WinCC puisse aller lire les données qui se trouvent dans la mémoire de l'automate. Après avoir créé notre projet WinCC, nous cliquons sur l'onglet liaison afin de créer une nouvelle liaison que nous nommerons « liaison\_1 » nous cliquons ensuite les différents paramètres :

- Interface : MPI/DP : Notre automate est relié par un MPI-DP.
- Adresse : permet de spécifier l'adresse de la station, dans ce cas-là l'adresse MPI.

Par exemple 2

L'éditeur "liaisons" affiche la connexion à l'automate configurée.

### **V.5.2. Création de la table des variables**

Maintenant que la liaison entre notre projet WinCC et l'automate est établie. Nous avons la possibilité d'accéder à toutes les zones mémoire de l'automate.

- Mémoire entrée/sortie.
- Mémento
- Bloc de données.

Les variables permettent de communiquer, c'est-à-dire d'échanger des données entre les composants d'un processus automatisé, ou encore entre un pupitre opérateur et un automate.

Une variable est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture à cette cellule mémoire est possible à partir du pupitre opérateur.

Afin de faire la correspondance entre les données du projet Step7 et les données du projet WinCC, il est possible de faire une table de correspondance des données via l'onglet variable. Chaque ligne correspond à une variable de WinCC. Elle est spécifiée par :

- Son nom.
- La liaison vers l'automate.
- Son type.
- Et le taux de rafraichissement de celle-ci.

Le taux de rafraichissement est le temps que doit mettre WinCC entre deux lectures dans la mémoire de l'automate

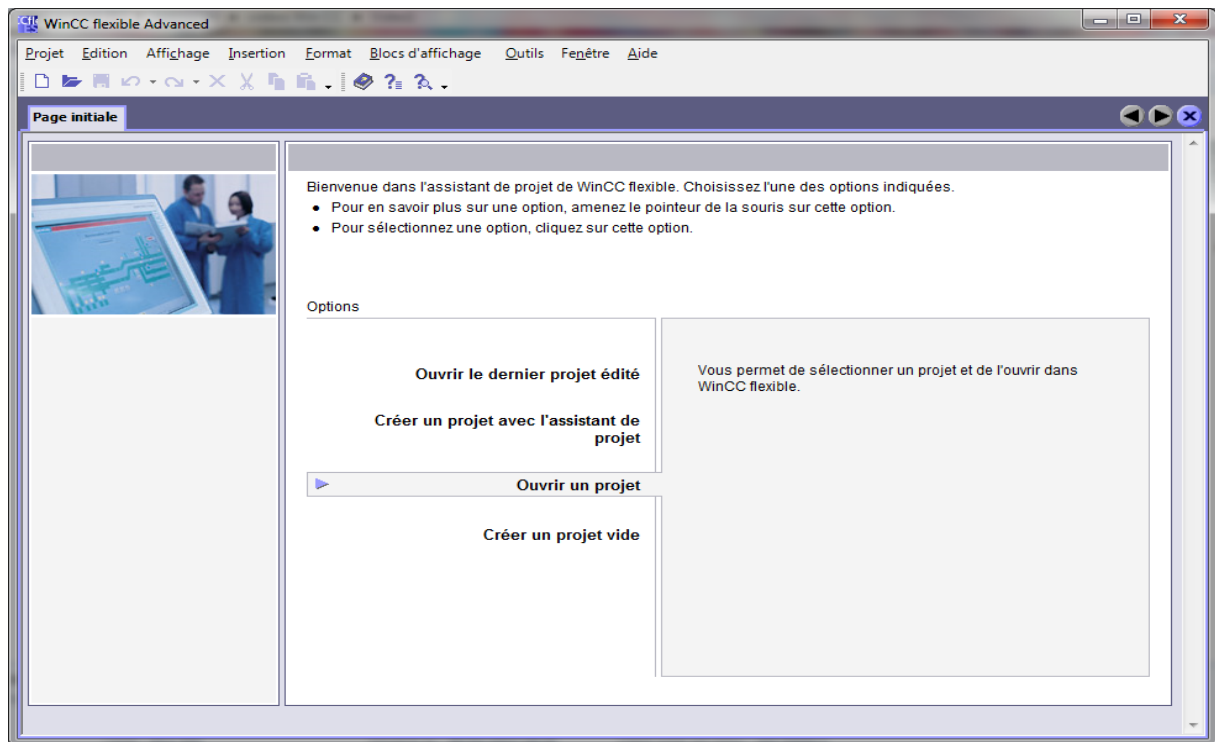
## **V.6. Création d'un projet sous WinCC flexible 2008**

Les principales étapes suivies pour la création de notre application sous WinCC flexible 2008 sont :

### **a) Création d'un nouveau projet**

Au démarrage de WinCC flexible, une fenêtre « WinCC flexible Advanced » s'ouvre avec des choix multiple pour la création de projet. Dans notre cas, on a choisi « créer un projet vide ».

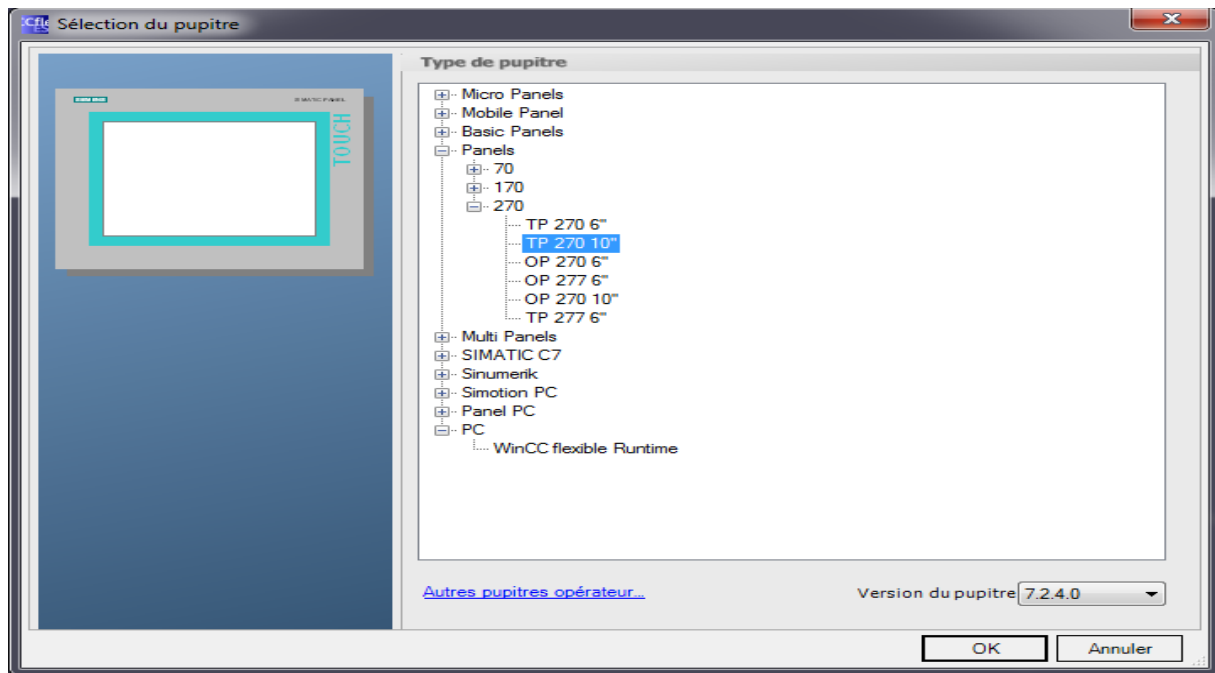




**Figure V.1 :** Création d'un nouveau projet.

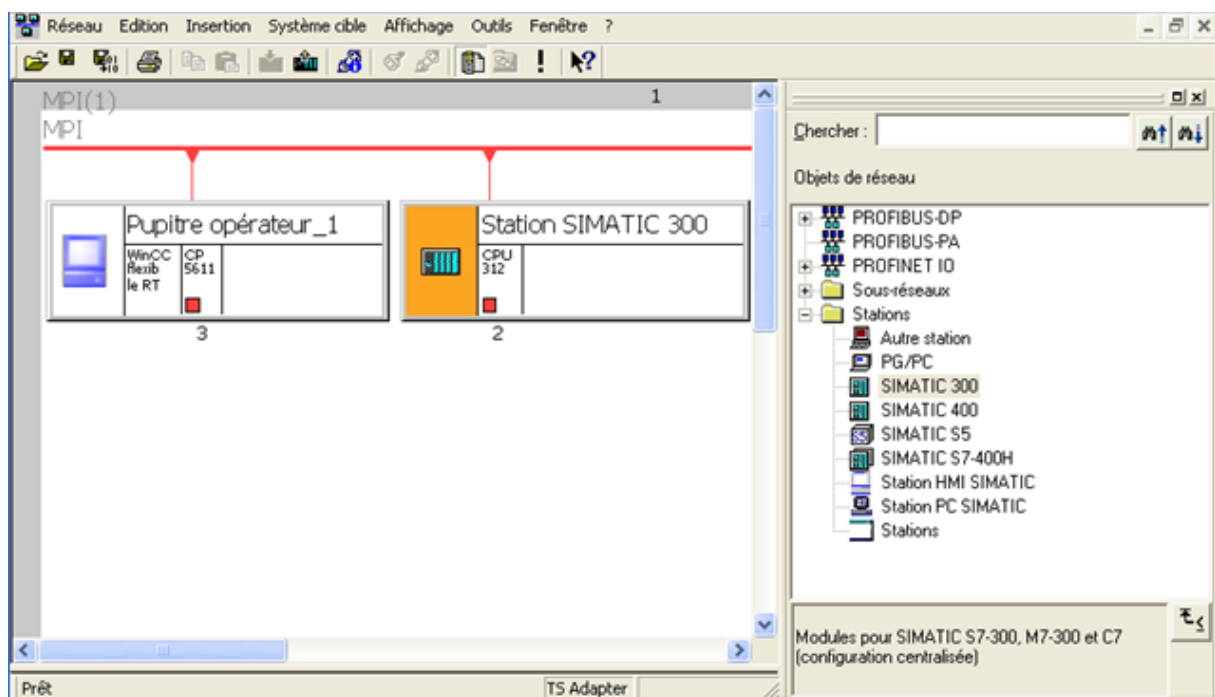
#### **b) Sélection de pupitre**

Après la création d'un nouveau projet, une nouvelle fenêtre s'ouvre pour le choix de pupitre à utiliser pour notre application.



**Figure V.2 :** Fenêtre de choix de pupitre.

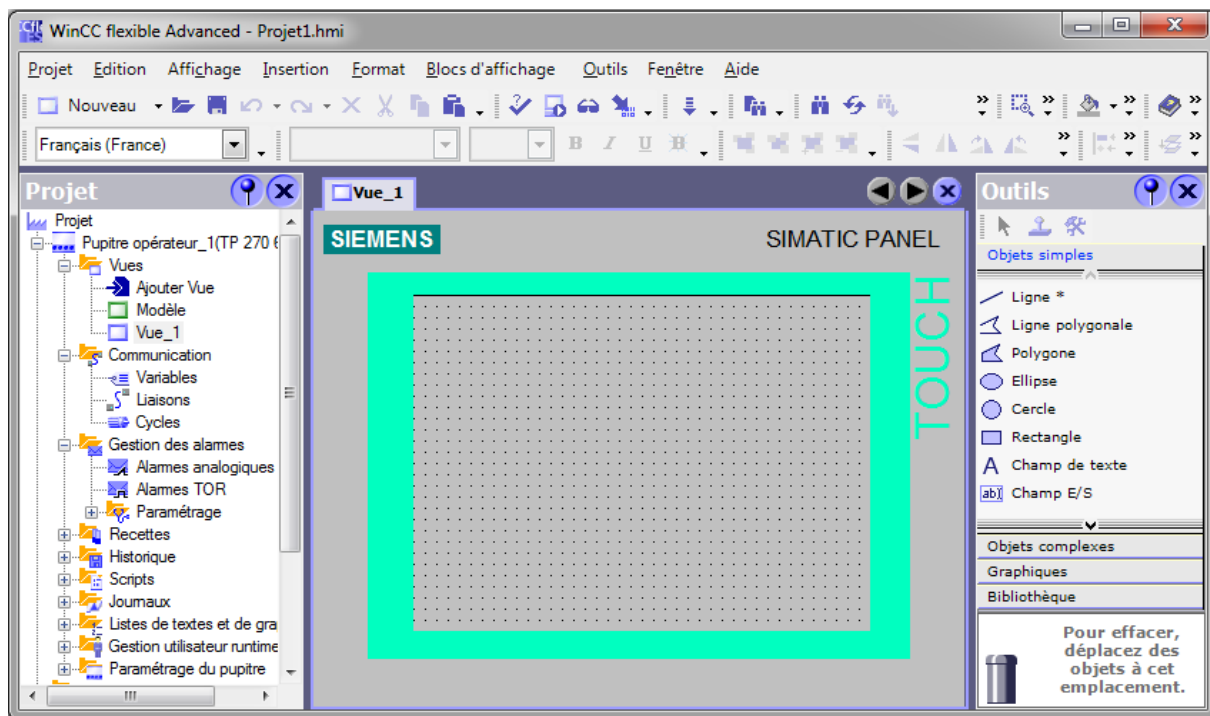
Puis nous avons procédé à la configuration du système de supervision pour assurer la communication entre l'API S7-300 avec le WinCC, pour ce fait nous avons sélectionné la liaison (MPI). (Voir figure V.3).



**Figure V.3 :** Configuration des paramètres de la Machine.

### c) Espace du travail

Après validation du type de pupitre, l'espace du travail WinCC Flexible Advanced-projet-HMI s'ouvre. L'espace de travail de WinCC flexible nous offre tous les outils nécessaires à la présentation d'un quelconque système automatique, mécanique, hydraulique et autres.



**Figure V.4 :** Espace du travail WinCC flexible.

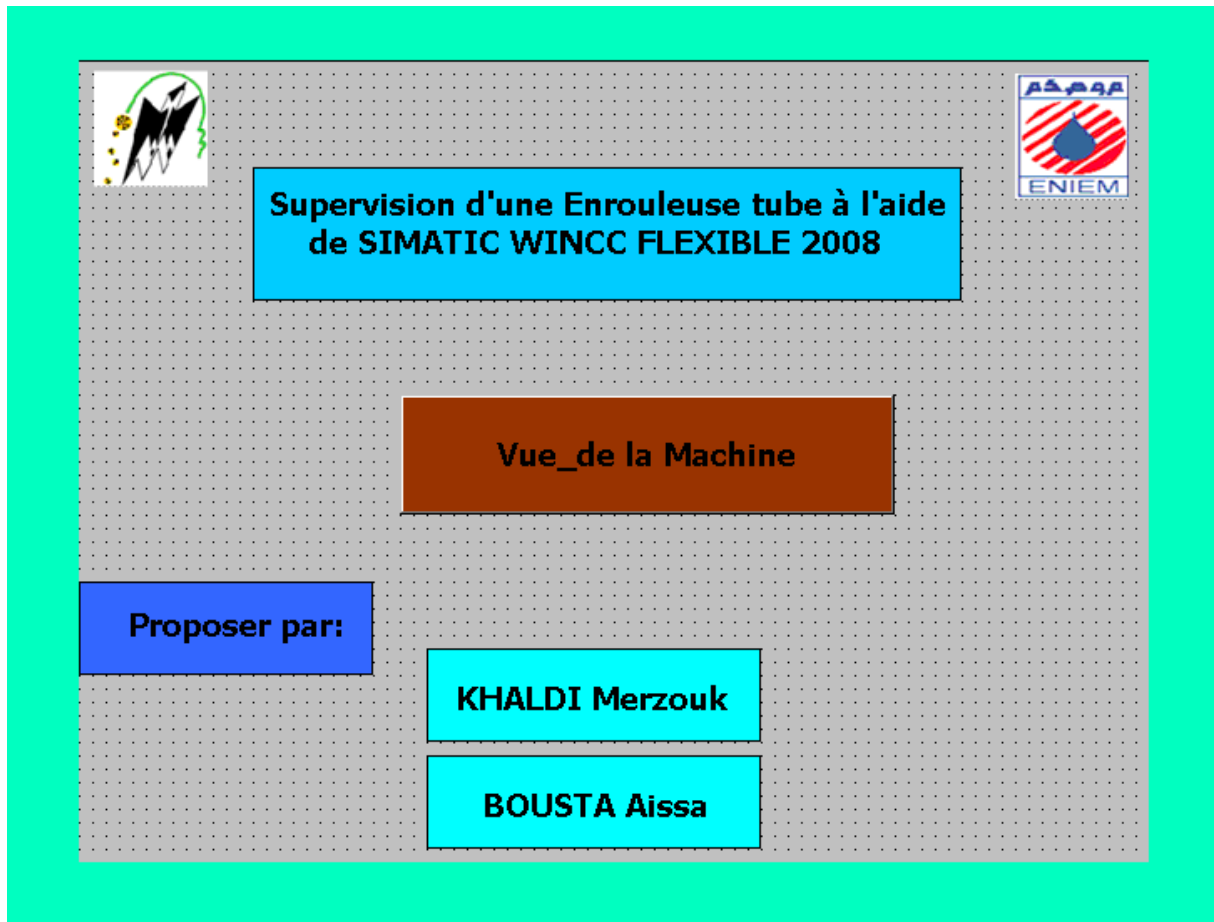
### d) Edition des vues

L'application WinCC que nous avons élaboré renferme toutes les vues contenant tous les éléments de notre système (l'accueil, les phases et pupitre de commande).

## V.7. Les vues du projet

### V.7.1. Vue d'accueil

Nous avons créé la vue d'accueil (Figure V.5), qui contient les boutons de navigation à partir desquels on peut sélectionner la vue à visualiser.



**Figure V.5 :** Vue d'accueil.

### V.7.2. Vue de la Machine

Dans cette vue se trouve notre machine les différents boutons : pour l'arrêt, pour accéder à la vue d'accueil.

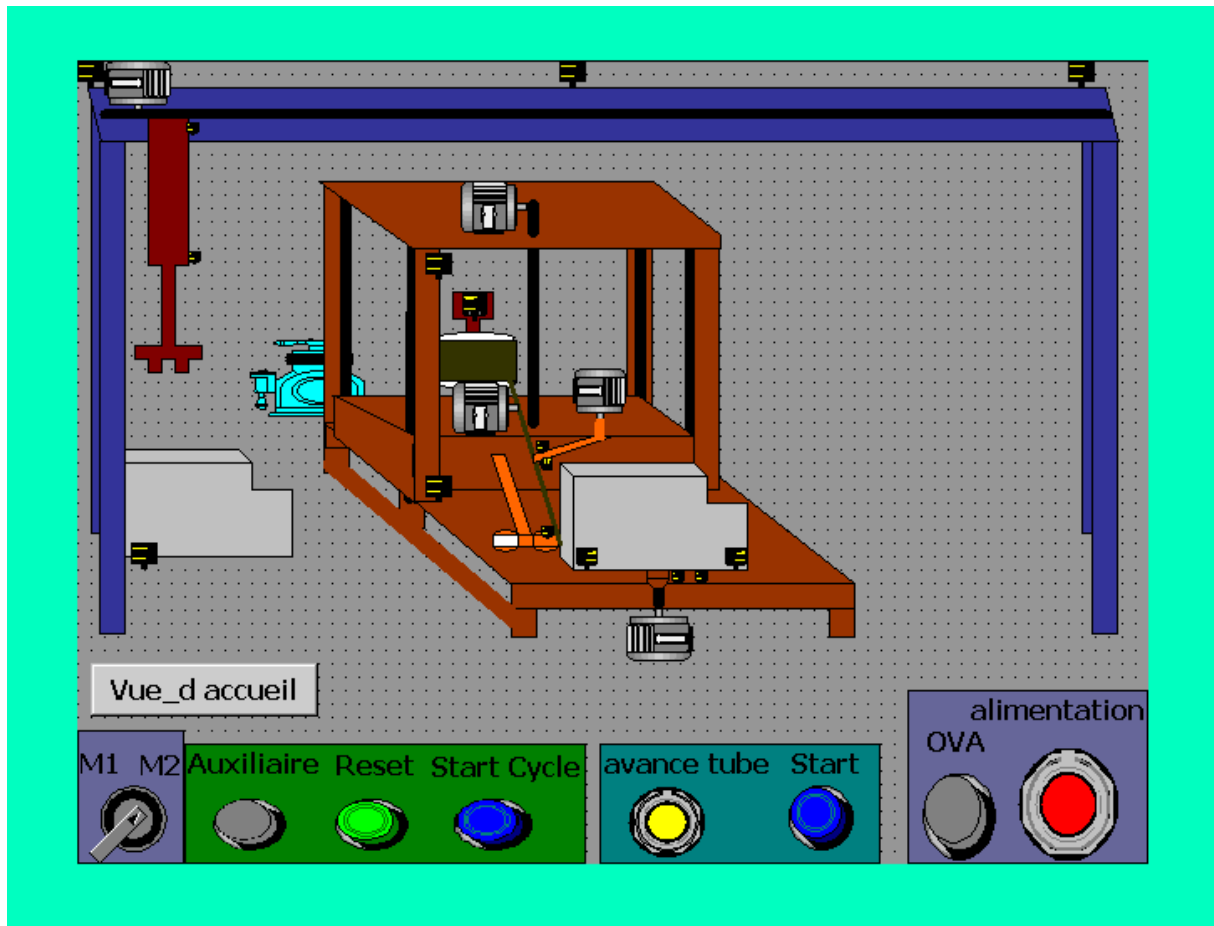


Figure V.6 : Vue 3D de la Machine.

## V.8. compilation et simulation

Après avoir créé le projet et terminé la configuration, il est indispensable de vérifier la cohérence du projet, et de détecter les erreurs, à l'aide de la commande sur la barre du menu "contrôle de la cohérence", après le contrôle de cohérence, le système crée un fichier de projet compilé.

La simulation permet de détecter des erreurs logiques de configuration, par exemple, y a une tache qui se réalise pas ou manque de tube pas de présence de cuve, et cela à l'aide du simulateur Runtime par la commande « démarrer le système Runtime du simulateur ».

**V.9.Conclusion**

Dans ce chapitre nous avons décrit la supervision en précisant sa place dans l'industrie, puis nous avons élaboré sous le logiciel WinCC Flexible les écrans permettant de suivre le fonctionnement de la chaîne.

## **Conclusion :**

Le travail que nous avons effectué dans le cadre de ce projet de fin d'études, avec l'appui du stage pratique à l'entreprise nationale d'Electroménager ENIEM, nous a permis de découvrir la réalité de l'activité d'un complexe industriel, de mettre ainsi en pratique nos connaissances théoriques et de nous familiariser avec les automates programmables industriels.

La commande des processus par un API est la solution recherchée de plus en plus dans l'industrie en raison de la précision (ou l'exactitude) des traitements numériques qu'ils effectuent pour générer la commande adéquate à tout moment et dans toutes les conditions.

Nous avons utilisé comme outil de modélisation le GRAFCET qui est un outil très efficace, qui facilite le passage du modèle à l'implantation technologique de celui-ci dans un automate programmable industriel. Il nous a permis ainsi d'effectuer le passage du langage de spécification au langage d'implémentation utilisé pour la réalisation de l'automatisme.

Le langage de programmation que nous avons utilisé, Le STEP7, permettra de communiquer simplement avec l'enrouleuse, en utilisant ses sources d'une manière efficace.

Notre conviction est d'avoir répondu dans la mesure de nos possibilités aux exigences du cahier de charges imposé.

Les améliorations techniques que nous avons apportées à l'installation assurent un plus haut niveau d'automatisation de l'enrouleuse. Elles libéreront l'opérateur des contraintes qui, présente, se traduisent par des arrêts fréquents du processus, donc des pertes économiques par gaspillage de matières et par baisse de productivité

Nous souhaitons, comme perspective de notre travail, que celui-ci serve de support à d'autres élevées Ingénieurs en Automatique pour réaliser les améliorations que nous avons proposées et une implémentation sur le processus réel à l'Entreprise d'électroménager ENIEM de ce système de commande

Dans notre travail, nous avons élaboré une solution de supervision dont le but est de contrôler le déroulement du processus par l'intermédiaire de graphismes et de schémas en temps réel. Il est donc facile de cibler, en cas de panne, un élément défectueux parmi les capteurs et les actionneurs. Ainsi, l'opérateur peut intervenir et prendre les décisions appropriées pour remédier aux défauts survenus en un temps minime.

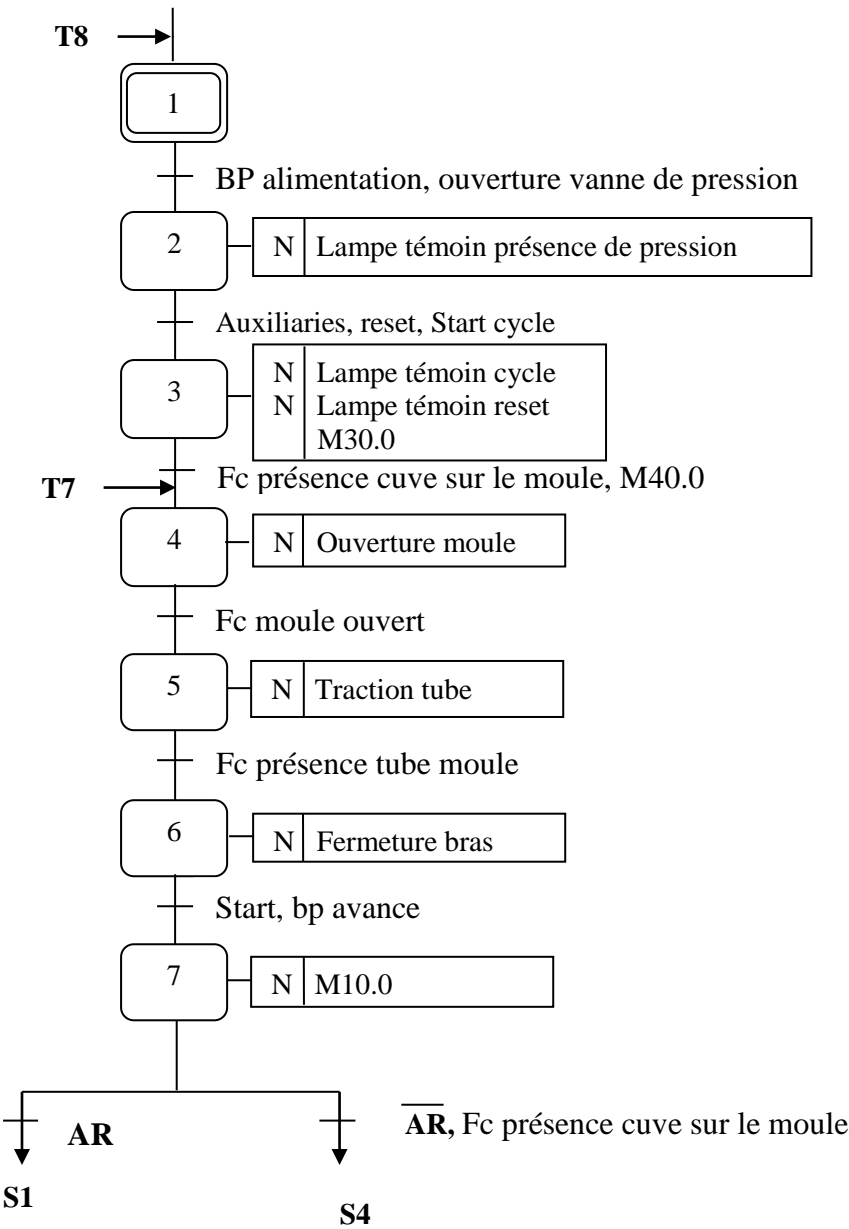
# BIBLIOGRAPHIE

- [1] : Documentations de la chaine de fabrication de congélateur, unité FROID (R2) de l'ENIEM.
- [2] : SIMATIC Programmer avec STEP 7 référencée :6ES7810-4CA10-8CW0.
- [3] : Navigation sur les moteurs de recherche « Google » (Recherche par mots clé).
- [4] : Site de l'usine ENIEM «[www .eniem.dz](http://www.eniem.dz) ».
- [5] : Mémoire de fin d'études U.M.M.T.O département automatique «Etude et automatisation par automates programmables S7-300 d'une presse transfert à l'entreprise l'ENIEM », promotion 2009.
- [6] : SIMATIC STEP 7 V5.1, référencée : 6ES7 810-4CA04-8CA0, Edition 08/2000.
- [7] : G. MICHEL, « Architecture, applications des automates programmables industriels ».
- [8] : A.BELKACEM et BOUKHERROUB, « Etude de l'automatisation par API S7-300 de la chaine de fabrication des armoires frigorifiques de l'ENIEM », département automatique, UMMTO, promotion 2007.
- [9] : Sites internet consultés : [www.ad.siemens.com](http://www.ad.siemens.com)
- [10] : Documentation technique SIMATIC Win CC flexible 2008.



**Grafcet Niveau 2 :**

FB1 :



FB2 :

