

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERI, Tizi-Ouzou**



**Faculté de Génie Electrique et d'Informatique**  
**Département d'Automatique**

## **PROJET DE FIN D'ETUDES**

En vue de l'obtention du diplôme de :

**MASTER PROFESSIONNEL EN AUTOMATIQUE**  
**Option : AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE**

# **Thème**

Automatisation de la machine de moulage, Expanseur  
ACE 25 AR, avec un automate siemens S7-300, ainsi que  
sa supervision winCC.

**Proposé par : Mr. AMEZIANE Djilali**

**Présenté par: Mr. OUBARECHE Salem**

**Dirigé par :**  
**Mr. DJENNOUNE. S**

Soutenu le : 21/ 09 /2016 Devant le jury d'examen composé de :

**Mr. MELLAH. R**  
**Mme. YOUSFI. S**  
**Mme. BEDOUHENE. S**

**Président**  
**Examinatrice**  
**Examinatrice**

*Promotion 2016*

**Mémoire préparé au sein de l'Entreprise Nationale des Industries Electroménagères.**

## Remerciements :

- *En premier lieu j'exprime mes remerciements les plus sincères à ALLAH le tout puissant sans lequel je ne serais arrivé où je suis.*
- *Je remercie mes parents pour leur soutien et pour leurs sacrifices pour me permettre d'être dans de bonnes conditions durant mes cursus, scolaire et universitaire.*
- *Je remercie mon promoteur Mr. DJENNOUNE Saïd pour son soutien et pour sa disponibilité.*
- *Je remercie aussi mon encadreur à l'ENIEM, Mr. AMEZIANE Djilali pour toute l'aide qu'il m'a apportée durant mon stage et pour sa patience, je remercie aussi son équipe de travail.*
- *Je remercie Mr. CHEBALLAH Saïd pour m'avoir facilité les procédures de mon stage à l'ENIEM.*
- *Je tiens à remercier aussi les membres du jury pour le temps qui sera consacré à mon travail.*
- *Je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.*

# Dédicaces :

*Je dédie ce modeste travail :*

*À mes parents*

*À toute ma famille*

*Et à tous mes amis.*

# *Sommaire*

# Sommaire

<b>Préambule</b> .....	<b>1</b>
<b>Introduction générale</b> .....	<b>4</b>

## **Chapitre I : Les systèmes automatisés de production (SAP)**

<b>1. Définition</b> .....	<b>6</b>
<b>2. Structure d'un système automatisé de production</b> .....	<b>6</b>
2.1. La partie opérative (PO) .....	<b>6</b>
2.2. La partie commande (PC) .....	<b>18</b>
2.3. La partie utilisateur (PU).....	<b>20</b>
<b>3. Conclusion</b> .....	<b>20</b>

## **Chapitre II : Description de la machine de moulage par expansion (Expandeur ACE 25AR)**

<b>1. Principe du moulage du polystyrène par expansion</b> .....	<b>21</b>
<b>2. Chaîne de production styropor de l'ENIEM</b> .....	<b>21</b>
2.1. Description de l'expandeur ACE 25 AR.....	<b>21</b>
2.2. Croquis et caractéristiques techniques.....	<b>22</b>
2.3. MODE OPERATOIRE DE L'EXPANSEUR ACE 25AR.....	<b>23</b>
2.3.3. Modes de commande.....	<b>24</b>
a) Commande manuelle.....	<b>24</b>
b) Commande semi-automatique.....	<b>25</b>
c) Commande automatique.....	<b>25</b>
<b>3. Conclusion</b> .....	<b>31</b>

## **Chapitre III : Les automates programmables industriels (API)**

<b>1. Introduction</b> .....	<b>32</b>
<b>2. Constitution d'un API</b> .....	<b>32</b>
2.1. Description des éléments d'un API.....	<b>32</b>
<b>3. Fonctionnement d'un automate programmable industriel</b> .....	<b>33</b>
<b>4. Alimentation de l'automate programmable industriel</b> .....	<b>33</b>
<b>5. Programmation des API</b> .....	<b>34</b>
<b>6. Les réseaux dans l'industrie</b> .....	<b>42</b>
<b>7. Choix d'un automate</b> .....	<b>45</b>
<b>8. L'automate S7-300</b> .....	<b>45</b>
<b>9. Sécurité</b> .....	<b>57</b>
<b>10. Conclusion</b> .....	<b>58</b>

## **Chapitre IV : Insertion du programme de commande et Simulation sous S7-PLCSIM**

1. Introduction.....	59
2. Logiciel de simulation S7-PLCSIM.....	59
2.1. Mise en route du logiciel S7-PLCSIM.....	59
3. Visualisation de l'état du programme.....	61
4. Conclusion.....	62

## **Chapitre V : Supervision de l'EXPANSEUR ACE 25 AR sous WinCC Flexible 2008**

1. Introduction.....	63
2. Généralités sur la supervision.....	63
2.1. Définition de la supervision.....	63
2.2. Constitution d'un système de supervision.....	63
2.3. Avantage de la supervision.....	64
3. Présentation du logiciel de supervision WinCC FLEXIBLE 2008.....	65
3.1. Avantages de WinCC flexible 2008.....	65
3.2. WinCC et SIMATIC STEP 7.....	66
3.3. Communication entre le PC de supervision et l'automate.....	66
4. Développement d'un système de supervision sous WinCC Flexible 2008.....	66
4.1. Réalisation des vues de supervision de l'EXPANSEUR ACE 25 AR.....	66
• Vue d'accueil.....	67
• Vue du Process.....	67
• Vue du circuit des fluides.....	68
5. Conclusion.....	68

<b>Conclusion générale.....</b>	<b>69</b>
---------------------------------	-----------

## **Bibliographie**

### **Annexes :**

**Annexe A :** Programme permettant le fonctionnement de la machine Expandeur ACE 25 AR.

**Annexe B :** Programme permettant la simulation du système de supervision wincc avec son Grafcet.

**Annexe C :** Table des mnémoniques.

# Préambule

**Présentation générale de l'ENIEM :**

**1. Situation géographique :**

L'ENIEM est une entreprise publique économique de droit Algérien constituée le 02 janvier 1983 mais qui existe depuis 1974 sous tutelle de l'entreprise SONELEC.

Son siège social se situe au chef lieu de la Wilaya de Tizi-Ouzou. Les unités de production **froid, cuisson, et climatisation** sont implantées à la zone industrielle Aissat Idir de Oued-Aissi, distante de 7 km du chef-lieu de wilaya.

La filiale sanitaire est installée à Miliana, wilaya de Ain Defla, et la filiale lampe à Mohammadia, wilaya de Mascara.

**2. Historique :**

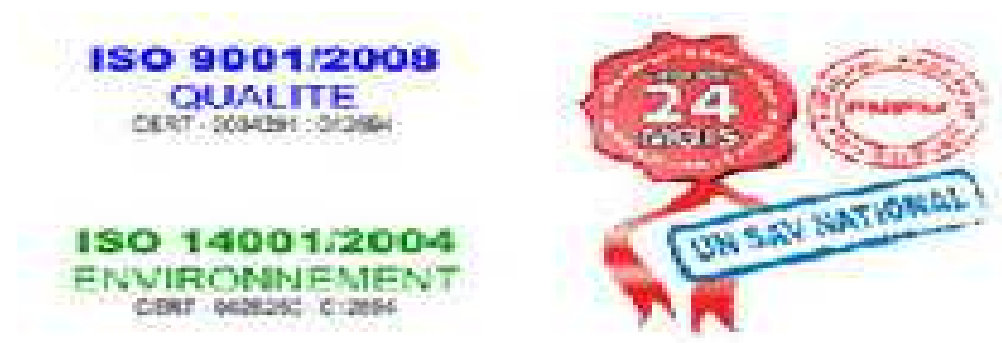
L'ENIEM est entrée en production à partir de janvier 1977 dont l'activité principale est la fabrication et la commercialisation d'appareils électroménagers.



**Figure 1 :** Siège social de l'ENIEM.

L'Entreprise est certifiée :

ISO 9001/2008 QUALITÉ et ISO 14001/2004 ENVIRONNEMENT.



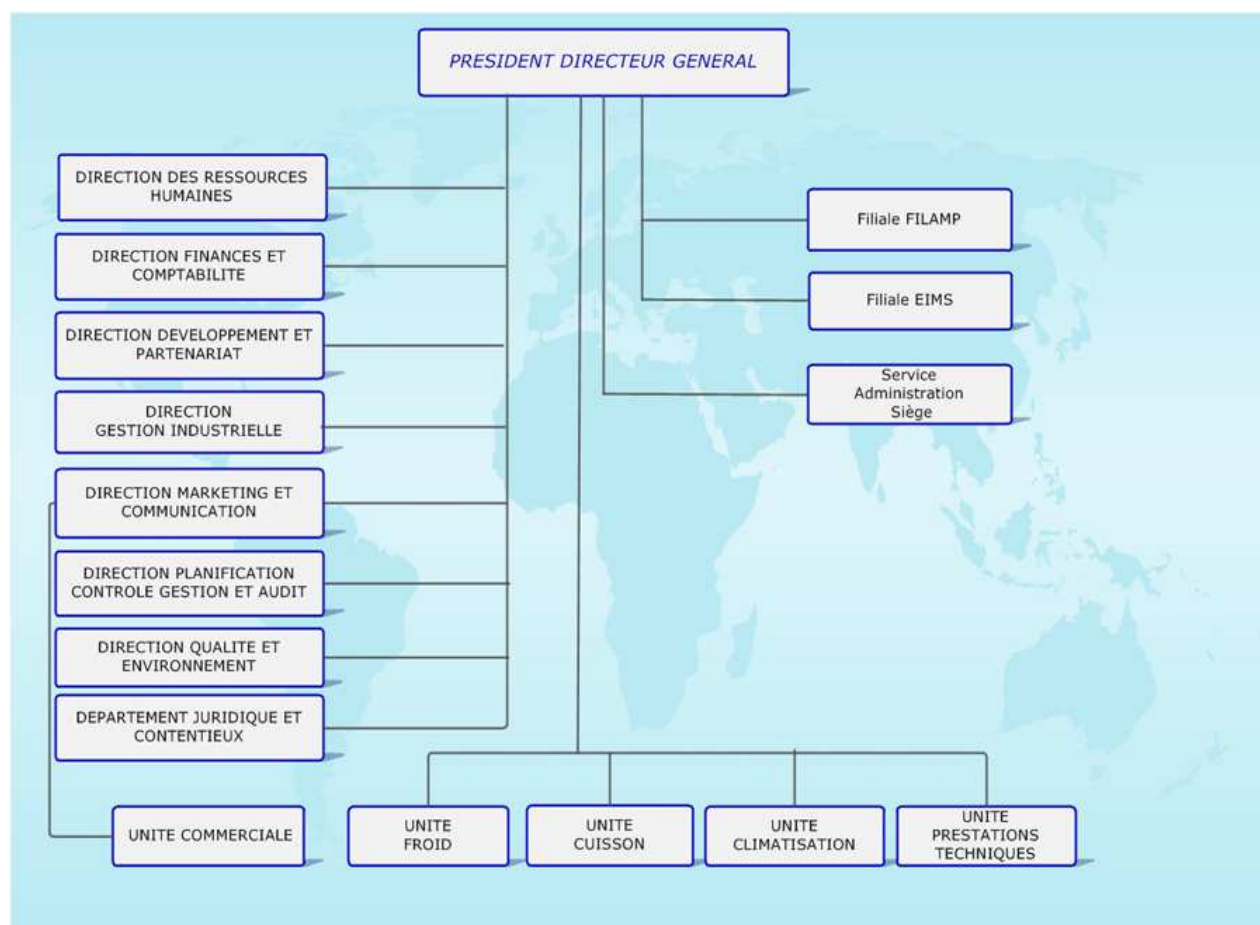
### 3. Objet social et champ d'activité :

L'ENIEM est leader de l'électroménager en Algérie, elle possède des capacités de production et une longue expérience dans le domaine de la fabrication et du développement dans les différentes branches de l'électroménager, notamment :

- Les appareils ménagers domestiques.
- Les appareils de collectivités.
- Les lampes d'éclairage.
- Les produits sanitaires.

### 4. Organisation générale :

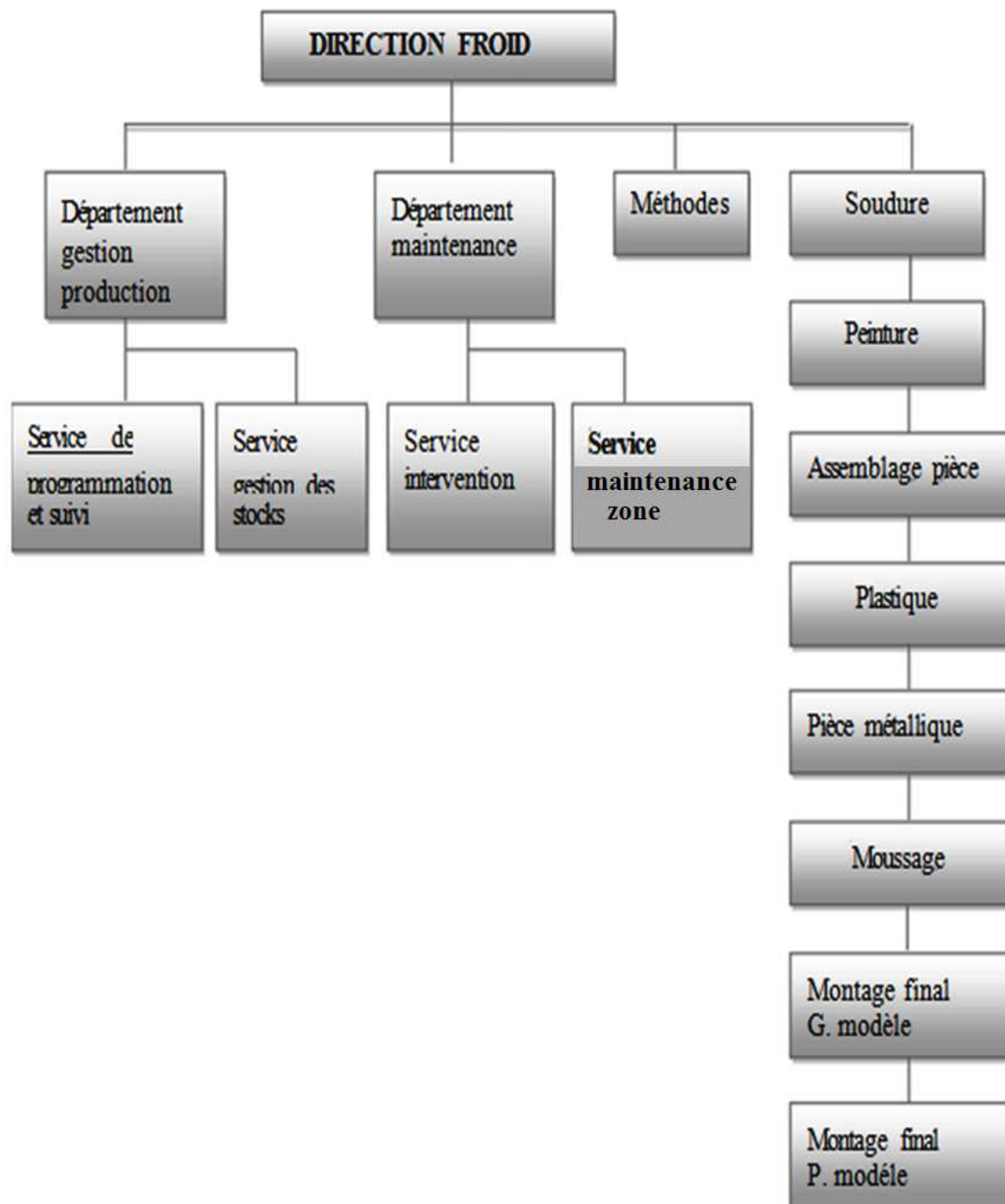
A partir de janvier 1998, l'entreprise s'est réorganisée en centre d'activités stratégiques qui s'articulent autour de la restructuration du complexe d'appareils ménagers créant plusieurs unités de production et de soutien à savoir :



**Figure 2 :** Organisation générale de l'ENIEM.

Le stage a été effectué au service de maintenance du service plastique de l'unité froid.

#### 4.1. Organisation de l'unité « froid » :



**Figure 3** : Organisation de l'unité « froid ».

#### 5. Capital social :

L'ENIEM a été transformée juridiquement en société par actions le 8 Octobre 1989. Son capital social est de 10.279.800.000 DA détenu en totalité par la ELEC ELDJAZAIR.[18]

# **Introduction générale**

De nos jours l'automatique a pris une part importante de notre mode de vie. Nous sommes arrivés à un point où nous ne pouvons plus nous passer des automatismes dans notre vie quotidienne. Les automates sont omniprésents que ce soit dans nos maisons, dans les lieux de travail ou en industrie. En effet avec l'automatisation, l'homme s'est épargné des tâches angoissantes et répétitives et qui lui prennent beaucoup de temps, par un simple clic sur un bouton il confie sa tâche à une machine qui l'exécute soigneusement et en un temps réduit et avec une efficacité et une précision inégalables.

Durant ces dernières années, la course à l'automatisation a connu une ascension fulgurante dans le domaine industriel. Les entreprises concourent à diminuer de plus en plus la main d'œuvre tout en augmentant la production, ce qui a conduit à l'avènement des automates programmables industriels (API).

Ces API sont le cerveau de l'automatisation, avec leur disposition à accomplir des tâches complexes de contrôle et de commande des machines.

Etant dans un développement incessant, les entreprises mondiales sont toujours à la recherche du meilleur API qui assurerait une meilleure conduite des outils de production. Il est admis que de nos jours la survie d'une unité de production dépend du niveau d'automatisation dont elle dispose. En effet, l'automatisation permet d'accélérer et d'amplifier la production à moindre coût afin que l'entreprise soit la première et la plus compétitive sur le marché de vente des produits.

L'entreprise nationale des industries électroménagères est parmi les entreprises qui essayent de survivre sur le marché. Pour ce faire elle doit faire des efforts considérables. En effet la majorité des automates utilisés à l'ENIEM ont disparu sur le marché ce qui cause l'arrêt total des machines pour absence de pièces de rechange en cas de pannes.

De ce fait, l'entreprise se voit dans l'obligation de remplacer les automates en place par de nouveaux automates tels que les automates Siemens S7-300.

Dans ce contexte , le sujet qui m'a été proposé dans le département de maintenance du service plastique de l'unité froid consiste à changer la commande existante (commande avec séquenceur Toshiba ), par une commande avec automate S7-300 en gardant le même programme et en essayant de faire une configuration matérielle semblable à celle déjà en place et d'ajouter un système de supervision WinCC, équipé d'un écran tactile.

Notre projet est réalisé en grande partie dans les ateliers de l'ENIEM. Il a porté sur trois phases essentielles. La première consiste à étudier le fonctionnement processus et le dispositif de son automatisation déjà installé. La seconde phase est une phase de réflexion dans laquelle nous avons remplacé théoriquement l'automate en place par l'API Siemens S7-300 muni d'un système de supervision. Enfin la troisième phase a porté sur l'illustration de la solution proposée en simulation.

Notre mémoire est organisé comme suit :

- Le premier chapitre est consacré aux systèmes automatisés de production, en détaillant ses composants.
- Le deuxième chapitre, est réservé à la description de la machine de moulage par expansion (EXPANSEUR ACE 25 AR) et son mode opératoire.
- Dans le troisième chapitre, on entre dans le vif du sujet, avec une introduction aux automates programmables en général, puis, à l'automate S7-300 en particulier, nous programmerons ce dernier à l'aide du logiciel step7 qu'on présentera.
- Le quatrième chapitre est réservé à l'insertion du programme de commande et à la simulation sous S7-PLC SIM.
- Le cinquième chapitre comporte la supervision de notre application sous WinCC flexible 2008.

Nous terminons ce projet par une conclusion générale.

# **Chapitre I :**

## **Les systèmes automatisés de production(SAP)**

## 1. Définition :

La littérature abonde de définitions relatives aux systèmes automatisés de production, malgré la diversité des approches qui les sous-entendent, elles s'accordent généralement sur le fait qu'un système automatisé de production est constitué essentiellement de trois composantes :

- **Les hommes** : appelés aussi opérateurs, ou utilisateurs, qui exploitent le système.
- **Le processus de production** : appelé aussi le processus physique, qui constitue le support physique du procédé. Le processus physique comprend un certain nombre d'équipements (cuves, tuyaux, réacteurs, machines, centres d'usinage, ...) qui permettent la mise en œuvre du procédé de fabrication. Ce dernier représente la définition du produit à obtenir grâce aux transformations effectuées par le processus.
- **Un système de contrôle de commande** : appelé aussi système d'automatisation ou système informatique temps réel, qui permet le contrôle et la gestion des transformations réalisées dans le processus de production. [1]

## 2. Structure d'un système automatisé de production :

Tout système automatisé peut être décomposé en trois grandes parties :

- **La partie opérative (PO).**
- **La partie commande (PC).**
- **La partie utilisateur (PU) ou bien partie supervision.**

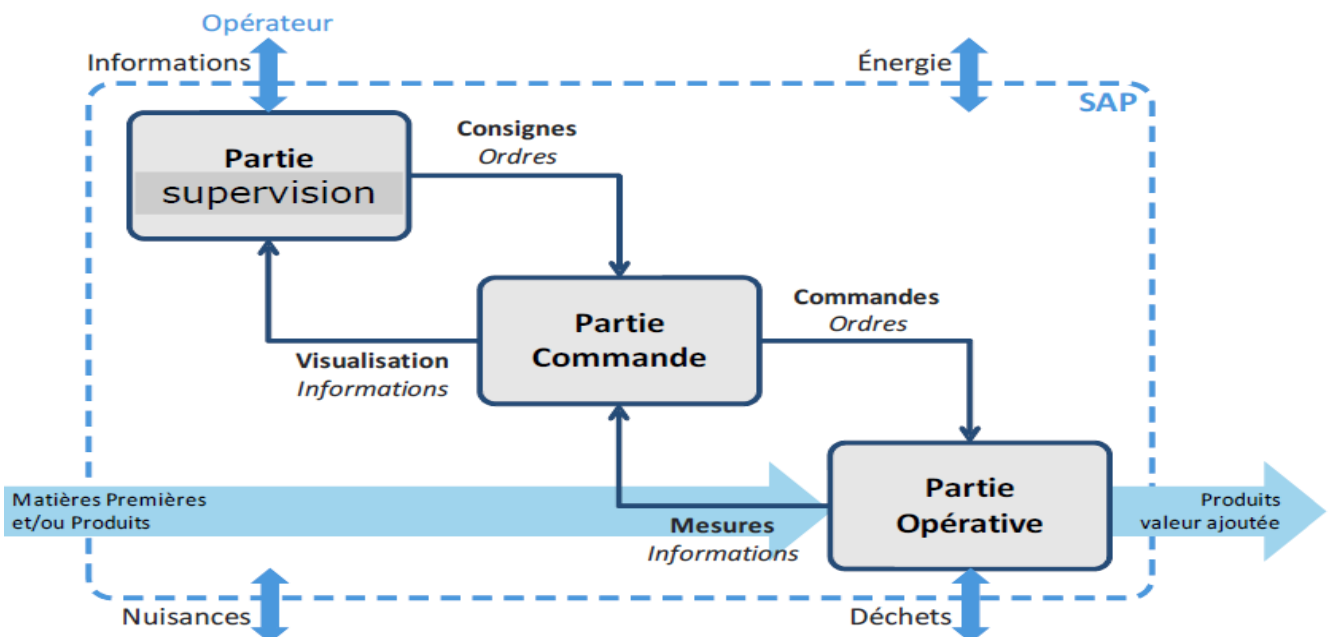


Figure I.1 : représentation de la structure d'un système automatisé. [2]

### 2.1. La partie opérative (PO) :

#### 2.1.1. Définition :

La partie opérative (PO) représente l'ensemble des moyens techniques qui permettent d'apporter la valeur ajoutée à la matière d'œuvre en effectuant directement le processus de leur transformation. [2]

2.1.2. Composants essentiels de la partie opérative (PO) :

2.1.2.1. Les actionneurs :

Définition :

Les actionneurs sont des convertisseurs électromécaniques conçus pour mettre en mouvement des systèmes mécaniques à partir de commande électrique. Ce sont essentiellement des moteurs et des vérins. Ils produisent de l'énergie mécanique à partir d'énergie électrique, hydraulique ou pneumatique mais toujours contrôlés par des signaux de commande électrique.

a) Les actionneurs pneumatiques :

Les actionneurs pneumatiques convertissent l'énergie de puissance pneumatique en énergie mécanique de translation, de rotation ou d'aspiration. Leurs principales caractéristiques sont : la **course**, la **force** et la **vitesse**. Parmi les actionneurs pneumatiques, on retrouve principalement les vérins, les moteurs et les ventouses.

a.1) Les vérins pneumatiques :

Un vérin pneumatique est un actionneur qui permet de transformer l'énergie de l'air comprimé en un travail mécanique. Un vérin pneumatique est soumis à des pressions d'air comprimé qui permettent d'obtenir des mouvements dans un sens puis dans l'autre. Les mouvements obtenus peuvent être linéaires ou rotatifs.

Un **vérin** pneumatique ou hydraulique est un tube cylindrique dans lequel une pièce mobile (le piston) sépare le volume du cylindre en deux chambres isolées l'une de l'autre. Un ou plusieurs orifices permettent d'introduire ou d'évacuer l'air dans l'une des chambres et ainsi de déplacer le piston, le piston muni d'une tige se déplace librement à l'intérieur d'un tube. Pour faire sortir la tige, on applique une pression sur la face avant du piston, et sur la face arrière pour faire rentrer la tige.

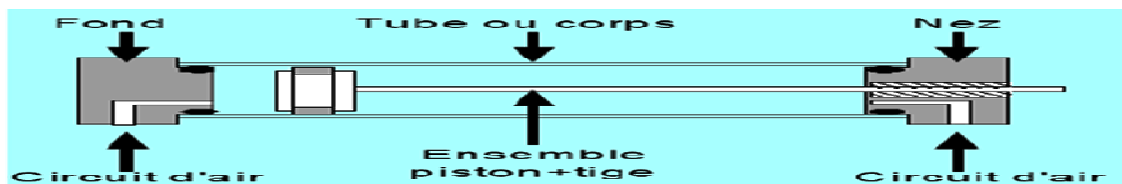


Figure I.2 : Constituants d'un vérin.

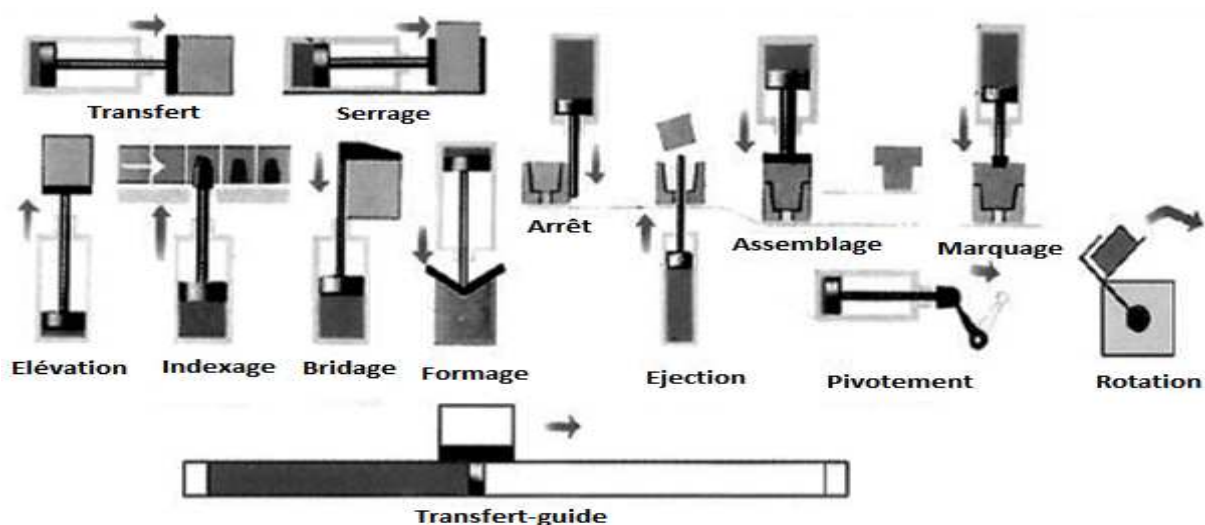


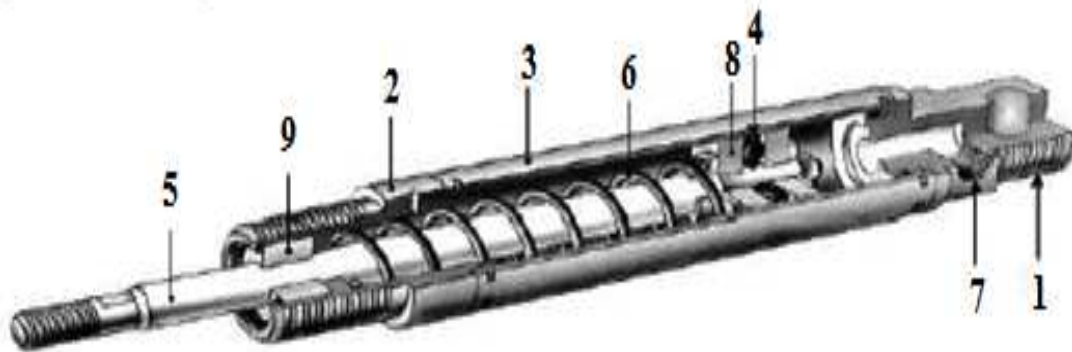
Figure I.3 : principaux domaines d'emplois des vérins pneumatiques.

**Classification des vérins :**

Selon leur mode de travail, dont dépend leur conception, ils se répartissent en trois catégories :

**1) Les vérins à simple effet:**

Dans lesquels l'air d'alimentation ne permet le travail que dans un seul sens : Tirer ou Pousser. L'arrivée de la pression se fait sur un seul orifice d'alimentation, ce qui entraîne le piston dans un sens, le retour s'effectuant sous l'action d'un ressort de rappel.



- 1. Fond
- 2. Nez
- 3. Tube
- 4. Joint de piston
- 5. Tige

- 6. Ressort de rappel
- 7. Entrée d'air
- 8. Piston
- 9. Douille

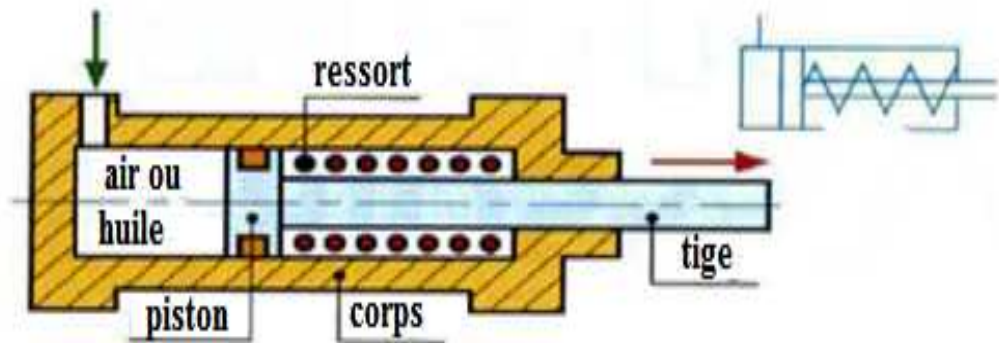


Figure I.4 : Vérin simple effet.

**2) Les vérins à double effet :**

Contrairement à la version à simple effet, ce type de vérin comporte deux orifices répartis sur les deux chambres du vérin et peut donc développer un effort en **tirant** et en **poussant**. La majeure partie de ces vérins peut être équipée de capteurs de position à détection magnétique à condition que le vérin dispose d'un piston magnétique.

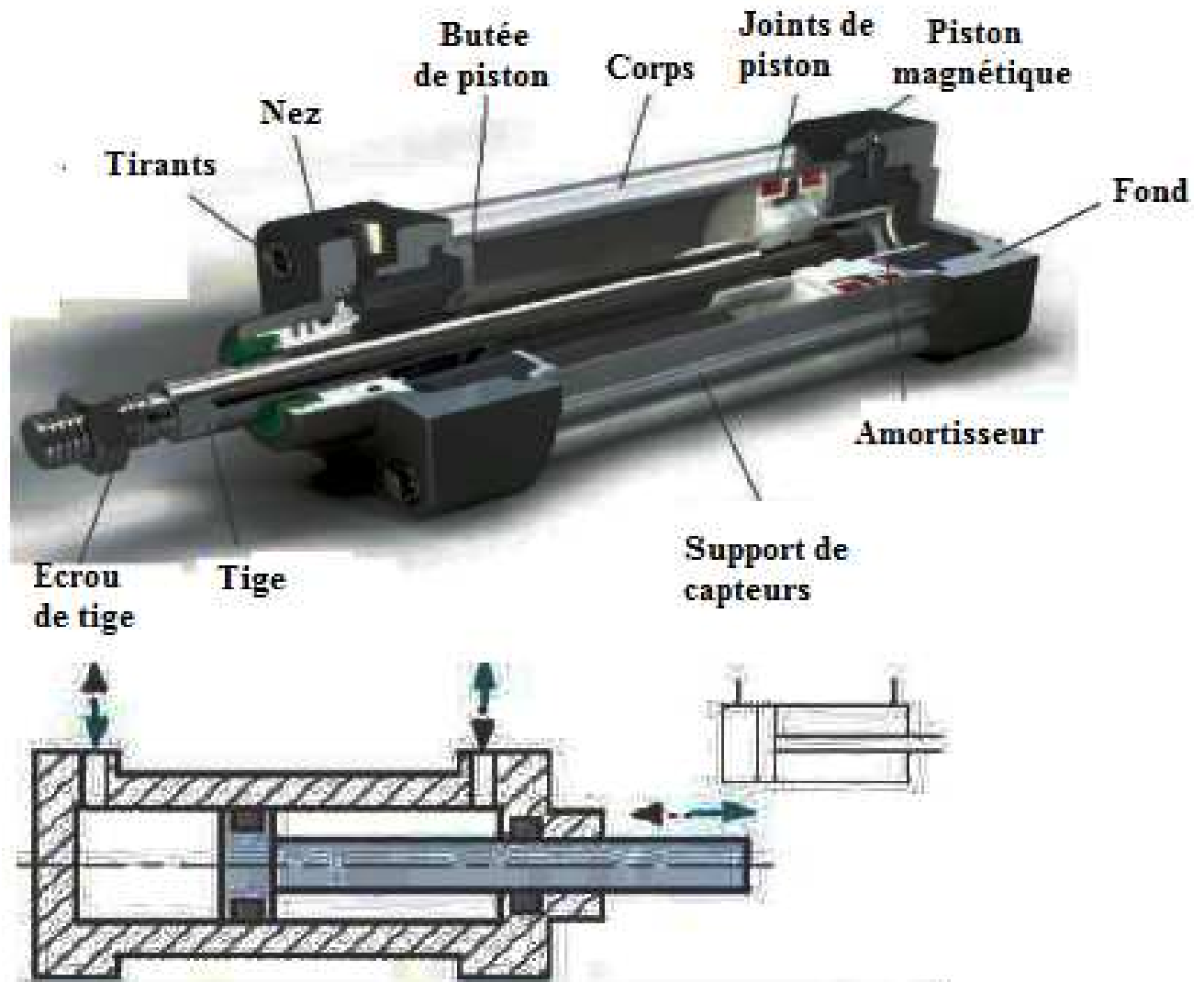


Figure I.5 : Vérin double effet.

### 3) Les vérins spéciaux :

Ils sont conçus à partir des concepts pour simple effet ou double effet, ils sont prévus pour des applications bien spécifiques, voir réalisés à la demande suivant un cahier de charges bien défini. (Vérin à bande, à membrane, souple, ... etc).

#### a.2) Les moteurs pneumatiques :

Les moteurs pneumatiques sont des appareils très utilisés dans les ateliers de production industrielle. Ils trouvent leur utilisation dans des milieux très variés comme les secteurs alimentaires ou pharmaceutiques, le montage automatisé de certains ensembles mécaniques dans l'industrie automobile, l'outillage à main (perceuses, visseuses pneumatiques, etc.).

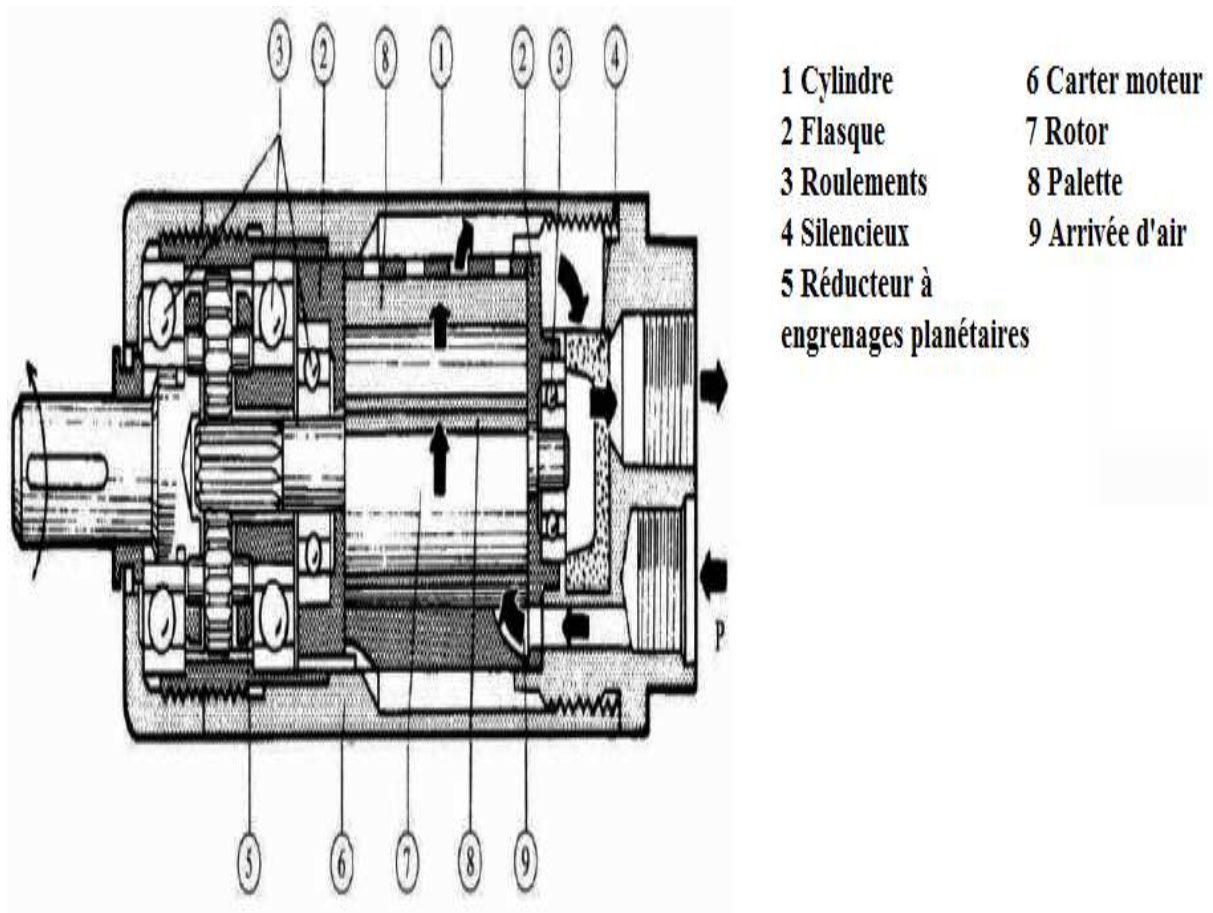


Figure I.6 : Moteur pneumatique.

**b) Les actionneurs hydrauliques:**

**b.1) La centrale hydraulique :**

La centrale hydraulique (appelée aussi groupe hydraulique) est un générateur de débit et non de pression (la pression augmente lorsqu'il y a résistance à l'écoulement). Elle est constituée essentiellement d'un réservoir d'huile, d'un moteur et d'une pompe et d'un système de filtration.

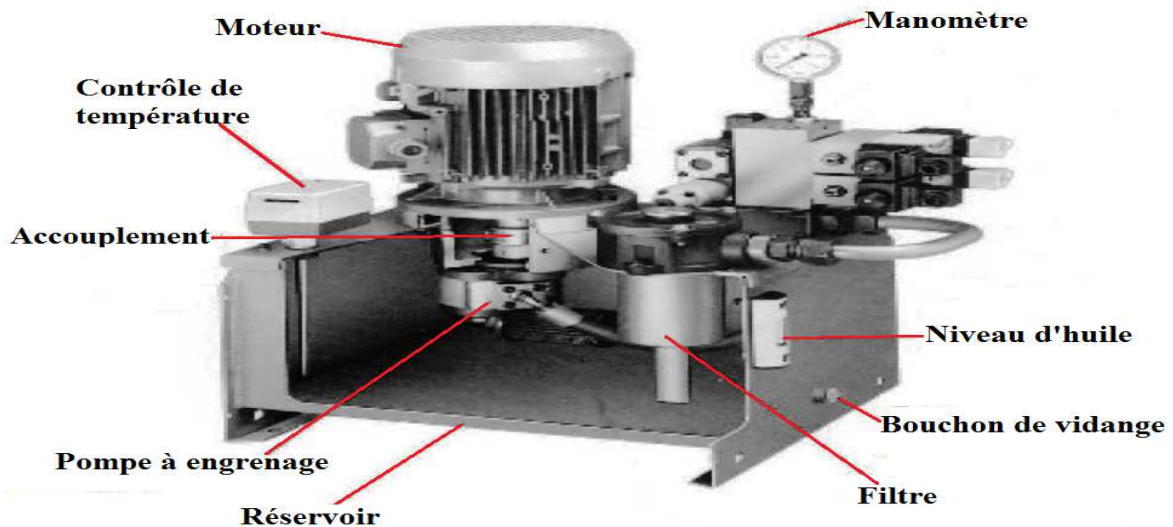


Figure I.7 : Centrale hydraulique.

**Réservoir** : il permet le stockage de l'huile, il le protège contre les éléments qui peuvent le polluer.

**Système de filtration** : il est utilisé pour éliminer les impuretés et les particules solides du fluide.

**Pompe** : sa fonction consiste à générer un débit de liquide et mettre sous pression l'huile sous forme d'énergie hydraulique.

Une centrale hydraulique doit contenir aussi d'autres composants (filtre, limiteur de pression, manomètre, ...).

### b.2) Le vérin hydraulique :

Le vérin hydraulique est un appareil qui transforme une énergie hydraulique en énergie mécanique. Il utilise l'huile sous pression jusqu'à 350 bars, par rapport au vérin pneumatique il est plus couteux et développe des efforts beaucoup plus importants. Les vitesses de tige sont plus précises. Principalement on a deux types : vérins simple effet et vérins double effet.



Figure I.8 : Schéma fonctionnel d'un vérin hydraulique.

### b.3) La pompe hydraulique :

La pompe hydraulique est destinée à transformer une énergie mécanique fournie par un moteur, en énergie hydraulique. Son rôle se limite à aspirer l'huile du réservoir et de la refouler. La pompe fournit un débit. Elle est donc **un générateur de débit**.

### b.4) Les moteurs hydrauliques :

Dans ce type d'actionneur, l'énergie hydraulique fournie par un fluide sous pression est transformée en énergie mécanique. Il en résulte un mouvement de rotation de l'arbre de sortie.

- Les moteurs hydrauliques présentent deux caractéristiques : le couple moteur et la vitesse de rotation.
- Les moteurs sont classés en deux familles : Les moteurs rapides (moteurs à palettes, à engrenages, à pistons axiaux et à pistons radiaux) et Les moteurs lents (cylindrée élevée). [3]

## c) Les actionneurs électriques :

### c.1) La machine à courant continu (MCC) :

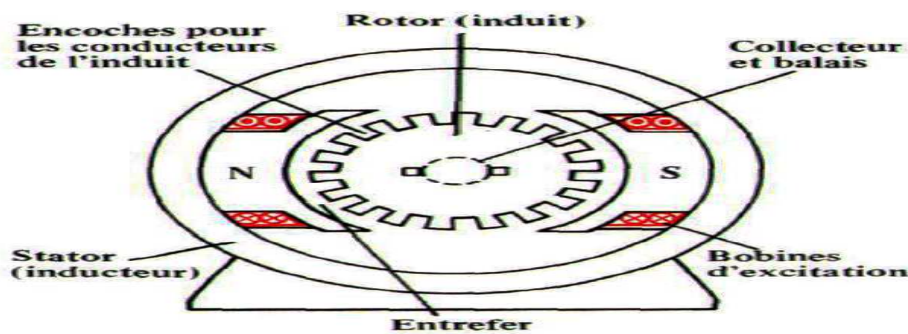
Une machine à courant continu est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique, selon la source d'énergie.

- En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
- En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique (elle peut se comporter comme un frein). Dans ce cas, elle est aussi appelée dynamo.

Cependant, la machine à courant continu étant réversible et susceptible de se comporter soit en « moteur » soit en « générateur » dans les quatre quadrants du plan couple-vitesse<sup>1,2,3</sup>, la distinction moteur/générateur se fait « communément » par rapport à l'usage final de la machine.

Une machine électrique à courant continu est constituée :

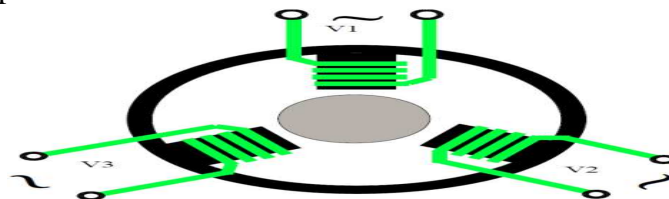
- d'un stator qui est à l'origine de la circulation d'un flux magnétique longitudinal fixe créé soit par des enroulements statoriques (bobinage) soit par des aimants permanents. Il est aussi appelé « inducteur » en référence au fonctionnement en génératrice de cette machine.
- d'un rotor bobiné relié à un collecteur rotatif inversant la polarité de chaque enroulement rotorique au moins une fois par tour de façon à faire circuler un flux magnétique transversal en quadrature avec le flux statorique. Les enroulements rotoriques sont aussi appelés enroulements d'induits, ou communément « induit » en référence au fonctionnement en génératrice de cette machine. [20]



**Figure I.9 :** Vue en coupe simplifiée d'un MCC.

**c.2) la machine asynchrone :**

La machine asynchrone fonctionne différemment par rapport à la MCC. L'alimentation de cette machine ne se fait plus par un générateur de tension constante, mais par un générateur triphasé.



**Figure I.10:** Représentation d'une vue en coupe d'une machine asynchrone.

**c.3) L'effecteur:**

Il représente tout élément de la partie opérative qui agit dans le système (cabine d'ascenseur, convoyeur, wagon, ...). Il est situé à la suite de l'actionneur pour finaliser le travail, il produit l'effet attendu. Par exemple : Les chenilles du robot ou la cabine d'ascenseur. L'effecteur est l'élément terminal de la chaîne d'énergie, sa fonction technique peut se décomposer en plusieurs actions :

- Transmettre l'énergie.
- Transformer l'énergie.
- Adapter l'énergie. [3]

**2.1.2.2. Le Pré-actionneur :****Définition :**

Un pré-actionneur est un composant de gestion de l'énergie de commande des actionneurs, il traduit un signal de commande en un signal de puissance, et distribue l'énergie utile aux actionneurs sur l'ordre de la partie commande. A toute action est associé un pré-actionneur indispensable pour son fonctionnement.

**2.1.2.2.1. L'électrovanne :**

C'est un pré-actionneur électropneumatique tout ou rien (TOR) permettant le passage d'air véhiculé dans le circuit 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. En se déplaçant, le noyau peut permettre ou pas le passage de l'air. La bobine doit être alimentée de façon continue pour maintenir le noyau attiré. [5]

**2.1.2.2.2. Le distributeur :****Définition :**

Le distributeur est un pré-actionneur associé à un vérin pneumatique (ou hydraulique). Il commande la circulation de l'énergie entre la source et l'actionneur.

Il est utilisé pour commuter et contrôler le débit du fluide sous pression, comme des sortes d'aiguillage, à la réception d'un signal de commande qui peut être mécanique, électrique ou pneumatique. Il permet de :

- contrôler le mouvement de la tige d'un vérin ou la rotation d'un moteur hydraulique ou pneumatique (distributeur de puissance).
- choisir le sens de circulation d'un fluide (aiguiller, dériver, etc.).
- exécuter, à partir d'un fluide, des fonctions logiques (fonctions ET, OU, mémoire, etc.).
- démarrer ou arrêter la circulation d'un fluide (robinet d'arrêt, bloqueur, ...).

**Symbolisation des positions d'un distributeur :****Principe de la symbolisation :**

**Nombre de cases :** il représente le nombre de positions de commutation possibles, une case par position. Chaque position du distributeur est symbolisée par un carré.

**Distributeur à deux positions****Distributeur à trois positions**

**Figure I.11 :** Symbolisation des positions d'un distributeur.

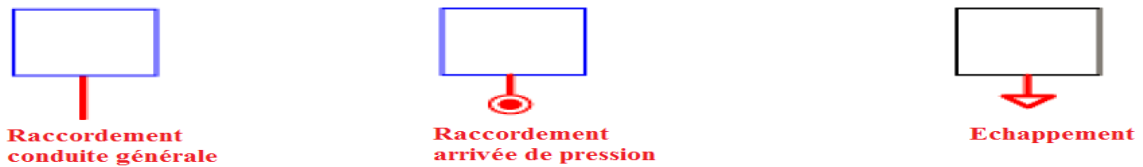
**Flèches :** dans chaque case ou position, les voies sont figurées par des flèches indiquant le sens de circulation du fluide entre les orifices.

**T** : les orifices non utilisés dans une position sont symboliquement obturés par un T droit ou inversé. Le nombre des orifices est déterminé pour une position et est égal pour toutes les positions.



**Figure I.12 :** Symbolisation des flèches et des orifices d'un distributeur.

**Source de pression** : elle est indiquée par un cercle noirci en hydraulique, clair en pneumatique. L'échappement est symbolisé par un triangle noirci en hydraulique, clair en pneumatique.



**Figure I.13 :** Symbolisation de la source de pression et de l'échappement d'un distributeur.

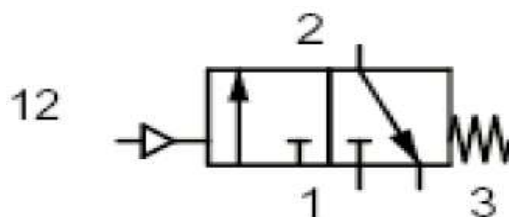
**Les différents types de commande d'un distributeur :**

La représentation des différents types de commande, s'ajoute de chaque coté du symbole de base (**Figure I.14**).



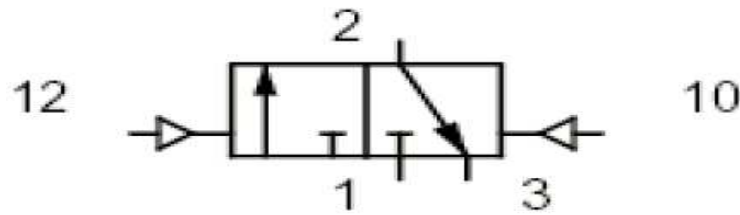
**Figure I.14 :** Symbolisation des différents types de commande d'un distributeur.

**Distributeur MONOSTABLE** : Si le distributeur possède un rappel par ressort, il est dit **monostable**.



**Figure I.15:** Représentation d'un distributeur monostable.

**Distributeur BISTABLE :** Si le distributeur possède deux pilotages, il est dit **bistable**.



**Figure I.16 :** Symbole d'un distributeur bistable.

Un distributeur est désigné par le nombre d'**orifices** qu'il comporte (Nombre d'orifices par case) et par le nombre de **positions** qu'il contient (Nombre de cases)

**Repérage des orifices :** Les orifices sont repérés de la façon suivante :

L'orifice(1) : Alimentation.

Les orifices qui ont des chiffres pairs : branchement vers l'organe à commander.

Les orifices qui ont des chiffres impairs : Echappements. [3]

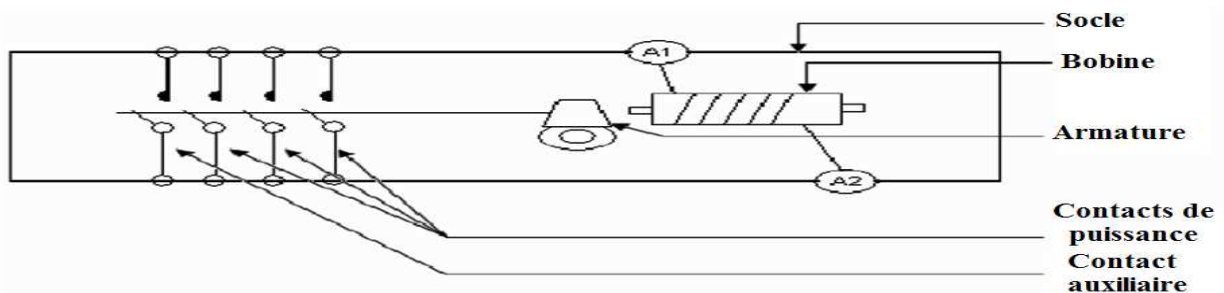
**2.1.2.2.3 Le Contacteur :**

**Définition :**

Un contacteur est un relais de haute puissance comportant des contacts à double rupture pour s'assurer de pouvoir couper des tensions et 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 I.17**) 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. [5]



**Figure I.17 :** Schéma d'un Contacteur.

**2.1.2.3. Les capteurs :**

Les capteurs sont des composants de la chaîne d'acquisition, dans une chaîne fonctionnelle. Ils prélèvent une information sur le comportement de la partie opérative et la transforment en une information exploitable par la partie commande.

Une information est une grandeur abstraite qui précise un événement particulier parmi un ensemble d'événements possibles. Pour pouvoir être traitée, cette information sera portée par

un support physique (énergie), on parlera alors de signal. Les signaux sont généralement de nature électrique ou pneumatique.

En résumé, un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable.

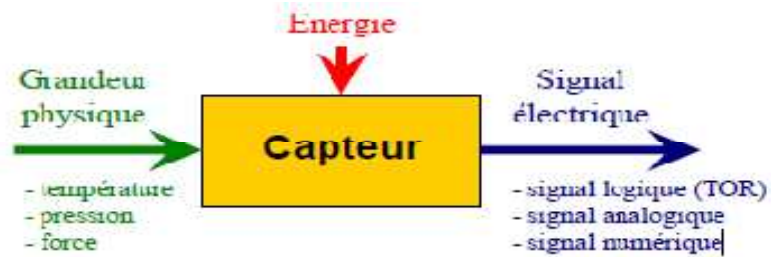


Figure I.18 : Schéma fonctionnel d'un capteur.

### 2.1.2.3.1. Principales caractéristiques des capteurs :

**Etendue de mesure :** C'est la valeur extrême pouvant être mesurée par le capteur.

**Résolution :** C'est la plus petite variation de grandeur mesurable par le capteur.

**Sensibilité :** C'est la variation du signal de sortie par rapport à la variation du signal d'entrée.

Exemple : Le capteur de température LM35 a une sensibilité de 10mV / °C.

**Précision :** Elle représente l'aptitude du capteur à donner une mesure proche de la vraie valeur.

**Rapidité :** Elle représente le temps de réaction du capteur. La rapidité est liée à la bande passante. [3]

### 2.1.2.3.2. Classification des capteurs :

Au delà du simple capteur à contact et à commande mécanique, il existe un grand nombre de modèles afin de répondre aux multiples problèmes posés par la détection.

#### a) Classification en fonction du signal délivré :

On distingue les grandes familles de capteurs par le type de signal qu'ils transmettent.

##### a.1) Signal TOUT ou RIEN (TOR) :

Ce sont les capteurs les plus répandus en automatisation (interrupteurs de position, détecteurs de proximité...). Les capteurs TOR ne délivrant que deux états 0 et 1 sont généralement appelés des DETECTEURS.

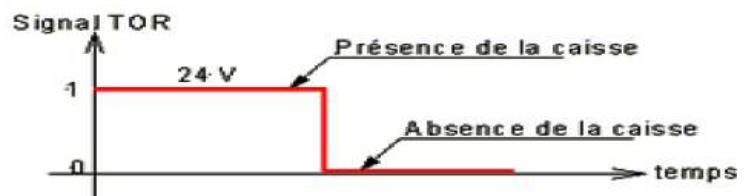


Figure I.19 : Image d'un capteur de présence et de son Signal TOR.

##### a.2) Signal ANALOGIQUE :

Il traduit des valeurs de température, de position, de pression... etc, sous la forme d'un signal évoluant entre deux valeurs limites.

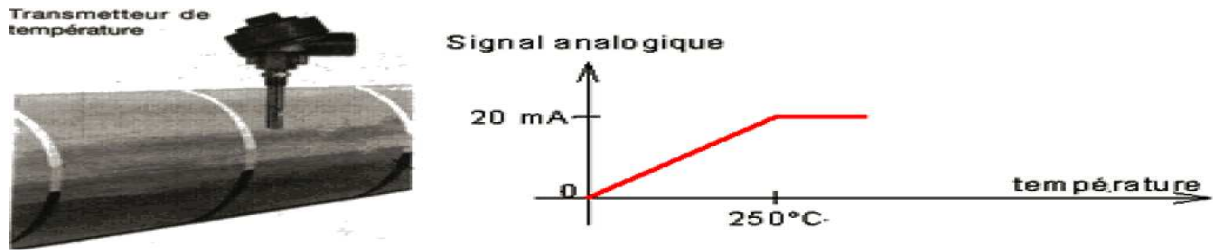


Figure I.20 : Image d'un transmetteur de température et de son signal analogique.

**a.3) Signal NUMERIQUE :**

Le signal délivré est une combinaison de BITS formant un signal numérique.

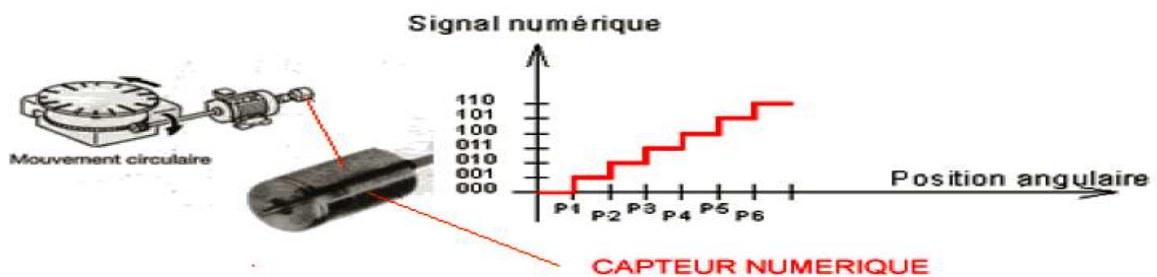


Figure I.21 : Capteur numérique.

**b) Classification en fonction du mode de détection :**

On a trois modes de détection :

**b.1) Détection par CONTACT :**

L'objet à détecter entre en contact avec le dispositif d'attaque.

**Principe de fonctionnement du détecteur de position :**

Les détecteurs de position sont constitués de trois éléments de base :

- (1) un contact électrique.
- (2) un corps.
- (3) une tête de commande avec son dispositif d'attaque.

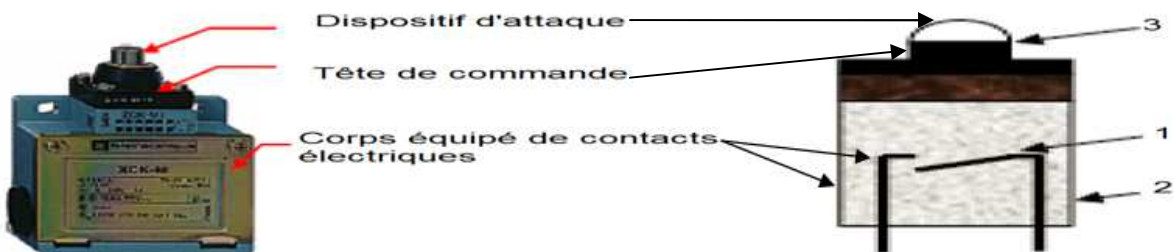


Figure I.22 : Détecteur de position.

**b.2) Détection SANS CONTACT:**

Le détecteur émet un champ magnétique ou électrique qui se trouve perturbé par la proximité d'un objet :

**Métallique :** Détecteur inductif (émission d'un champ magnétique).

**Non métallique :** Détecteur capacitif (émission d'un champ électrique).

### b.3) Détection à DISTANCE d'un objet :

Un rayon lumineux est interrompu par l'objet à détecter, Un photorécepteur traduit cette présence en un signal électrique.

Il existe 3 systèmes de détection photo-électrique :

- Le système BARRAGE.
- Le système REFLEX.
- Le système DE PROXIMITE.

#### b.3.1) Système barrage :

L'émetteur et le récepteur sont dans 2 boîtiers séparés. L'objet est détecté lorsqu'il interrompt le faisceau lumineux.

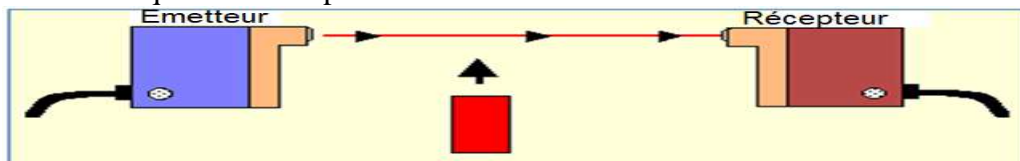


Figure I.23 : Représentation d'un système barrage. [4]

#### b.3.2) Système reflex :

L'émetteur et le récepteur sont dans le même boîtier. Le faisceau émis est renvoyé au récepteur par un réflecteur. La détection se fait par coupure du faisceau.



Figure I.24 : Représentation d'un système reflex.

#### b.3.3) Système de proximité :

L'émetteur et le récepteur sont dans le même boîtier. La détection se fait lorsque le faisceau est réfléchi par la pièce à détecter.[3]

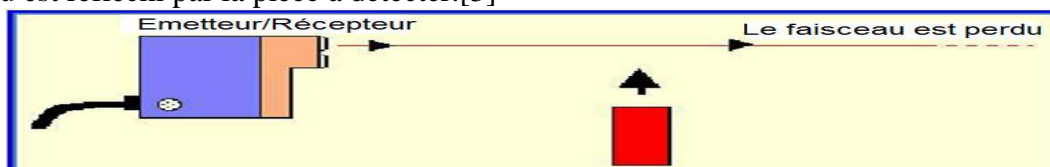


Figure I.25 : Représentation d'un système de proximité.

## 2.2. La partie commande (PC) :

### 2.2.1. Définition :

La partie commande représente l'ensemble des moyens de traitement de l'information qui assurent le pilotage et la coordination des tâches du processus souhaitées par un programme préétabli. [3]

### 2.2.2. Technologie de la commande:

#### a) La logique câblée :

L'élément principal s'appelle le module séquenceur, et l'association de modules constitue un ensemble appelé séquenceur. Le pilotage des distributeurs se fait par une action

de l'air comprimé sur un piston qui fait déplacer le tiroir du distributeur à droite ou à gauche. L'ensemble, appelé tout pneumatique, est homogène et fiable.

**b) La logique programmée:**

C'est une commande électrique dont l'élément principal s'appelle l'Automate Programmable Industriel ou l'API. La détection est électrique. Le pilotage des actionneurs se fait par l'intermédiaire de relais ou de distributeurs. Il existe sur le marché de nombreuses marques d'automates : Télémécanique, Siemens, Omron, Allen Bradley, Cegetel, etc...

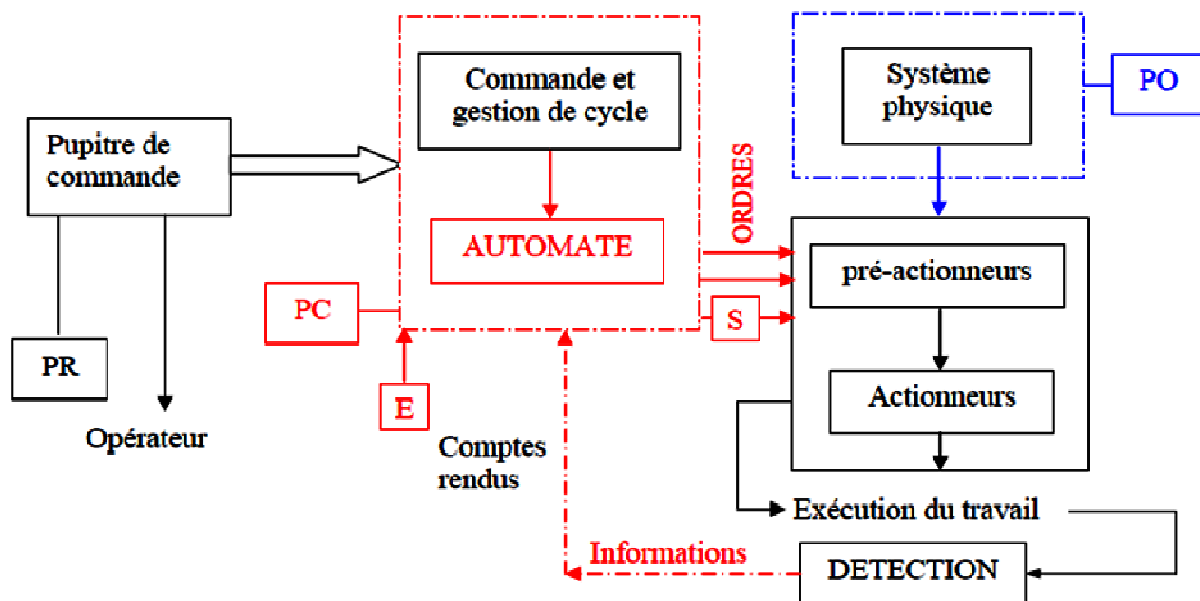
**b.1) L'Automate Programmable Industriel :**

L'Automate Programmable Industriel (API) (en anglais Programmable Logic Controller, PLC) est une machine électronique programmable par un personnel non informaticien. L'API réalise des fonctions d'automatisme pour assurer la commande de pré actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. L'Automate Programmable Industriel (API) peut être de type **compact** ou **modulaire**.

**L'automate compact** intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité.

Pour **l'automate modulaire**, le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (**modules**) et sont fixés sur un ou plusieurs **racks** contenant le "fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où la puissance, la capacité de traitement et la flexibilité sont nécessaires.

Nous verrons l'API en détails dans le chapitre III.



**Figure I.26 :** Situation de l'automate dans un système automatisé de production.

**Structure générale d'un API :**

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

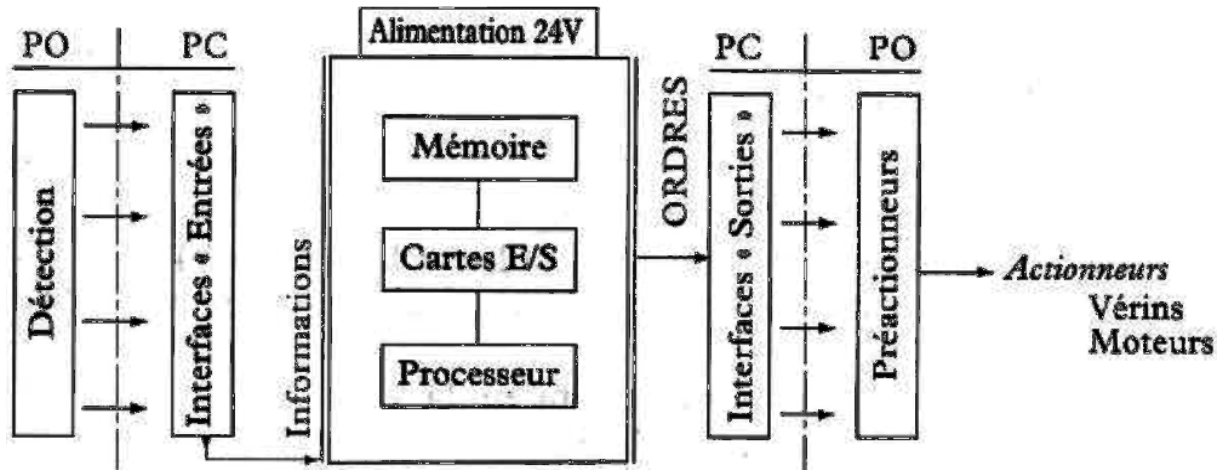


Figure I.27 : Structure générale d'un API.

### 2.3. La partie utilisateur (PU) :

Elle représente l'ensemble des moyens de dialogue qui permettent à l'utilisateur d'échanger des informations avec la partie commande. [3]

Elle est équipée d'organes permettant :

- La mise en hors énergie de l'installation.
- La sélection des modes de marche.
- La commande manuelle des actionneurs.
- Le départ des cycles de fonctionnement.
- L'arrêt d'urgence.
- D'informer l'opérateur de l'état de l'installation : voyants lumineux, afficheurs, écrans vidéos, klaxons... [2]

### 3. Conclusion :

Dans ce premier chapitre, nous avons vu les différents composants d'un système automatisé de production et le rôle de chacun dans ce dernier. Nous essayerons de voir par la suite l'API qui est devenu indispensable dans les systèmes automatisés industriels récents, avec plus de détails.

# Chapitre II :

Description de la machine de moulage par expansion (Expandeur ACE 25AR)

**1. Principe du moulage du polystyrène par expansion:**

Avant d'être expansé, le polystyrène se présente sous forme de petites billes qui renferment des micro-inclusions de gaz (à l'état liquide).

Au contact de la vapeur d'eau, la matière plastique se ramollit et le gaz qu'elle contient se dilate. Les petites billes gonflent, comme le pop-corn gonfle grâce à l'air qu'elles contiennent, en gardant une forme plus régulière.

Cette première expansion est réalisée en usine dans de grandes cuves en inox et permet d'obtenir jusqu'à 30 fois le volume initial des petites billes de polystyrène.

Après séchage des billes pré-expansées dans un silo, cette matière est introduite dans un moule fermé et est soumise à une nouvelle injection de vapeur d'eau.

Les billes reprennent leur expansion, occupent tout l'espace du moule et se soudent entre elles pour obtenir la forme désirée de l'emballage. [19]

**2. Chaîne de production styropor de l'ENIEM :**

L'ENIEM possède ses propre machines de production de pièces de polystyrène permettant la protection de ses produits une fois emballés, parmi elles : l'EXPANSEUR ACE 25 AR.

**2.1. Description de l'expandeur ACE 25 AR :**

L'expandeur est constitué d'une partie fixe et d'une autre mobile, qui sont fortement pressées l'une contre l'autre au moment du moulage puis écartées pour permettre l'éjection de la pièce moulée.

Il est constitué aussi de plusieurs circuits de fluides permettant le chauffage, l'aération et le refroidissement du moule.



**Figure II.1 :** Image photographique de l'expandeur ACE 25 AR de l'ENIEM.

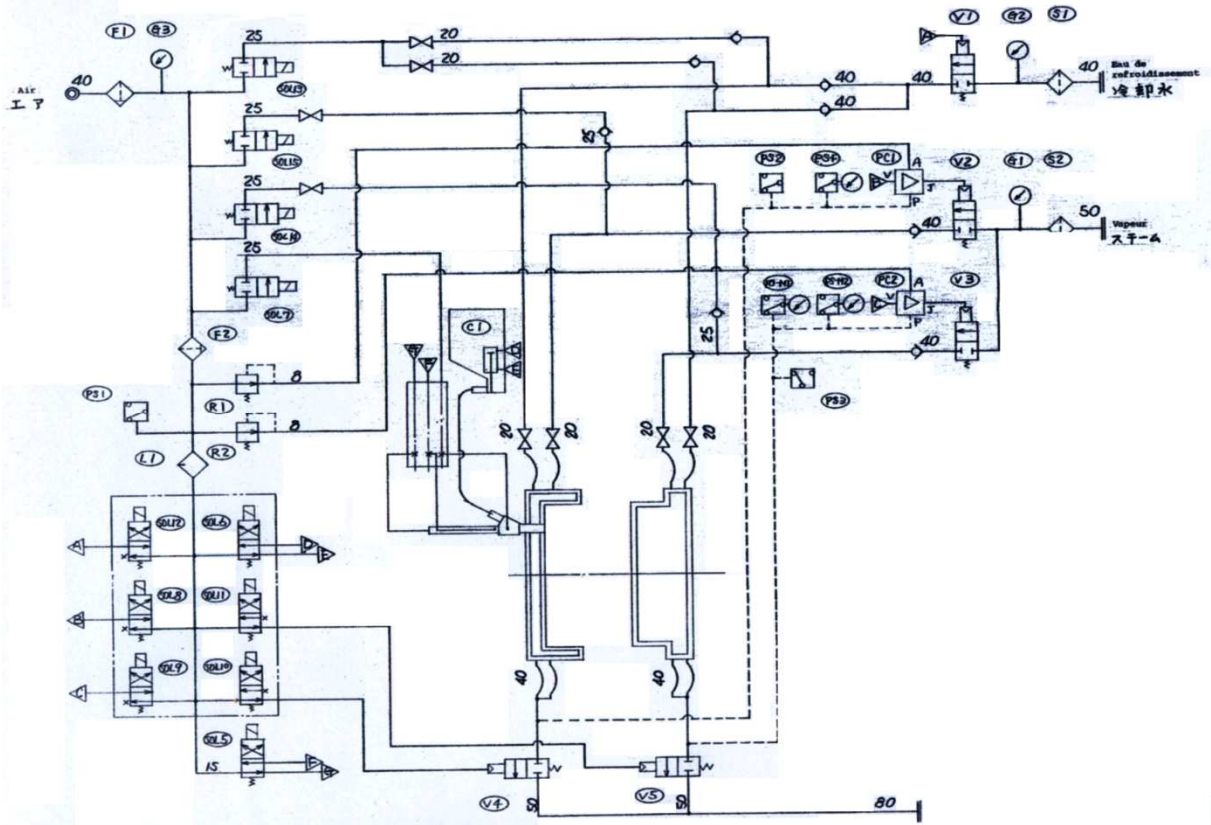


Figure II.2 : Circuit des fluides de l'Expandeur ACE 25 AR.

2.2. Croquis et caractéristiques techniques :

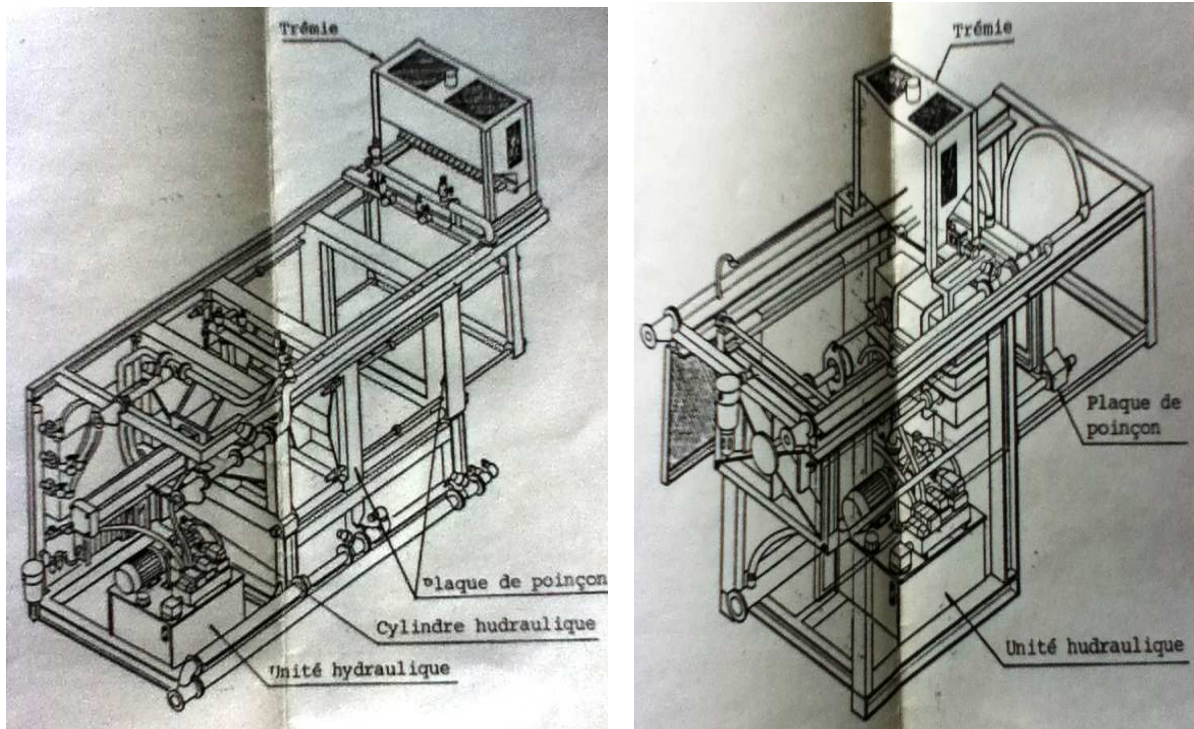


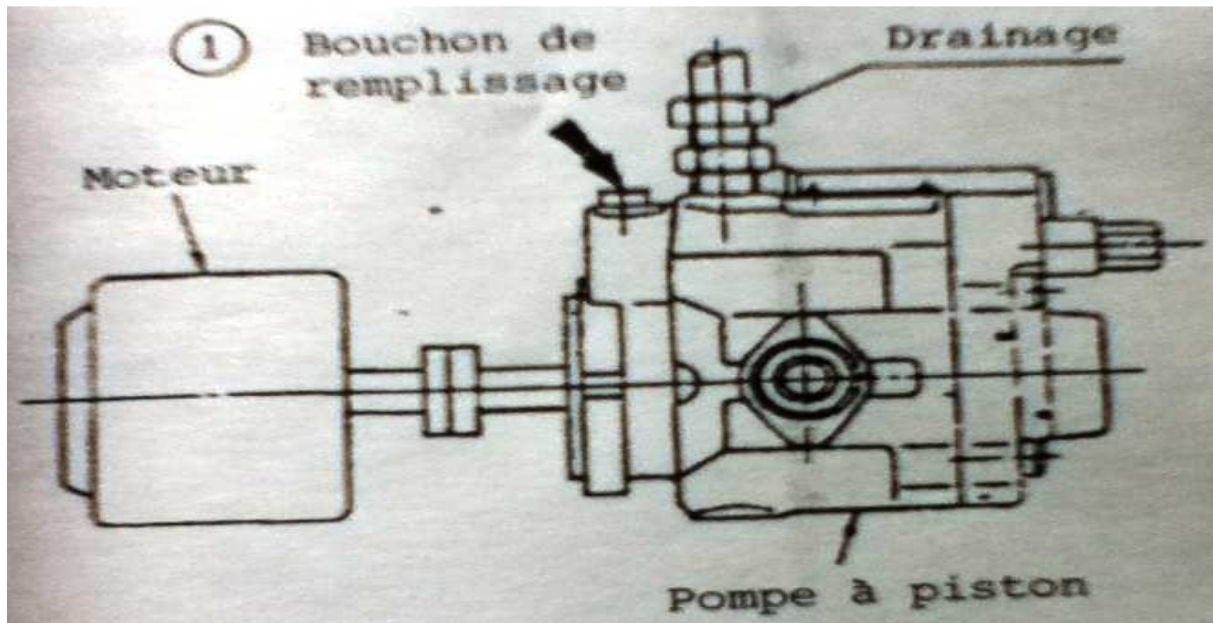
Figure II.3 : Représentation du croquis de l'expandeur ACE 25 AR.

**2.3. MODE OPERATOIRE DE L'EXPANSEUR ACE 25AR:**

**2.3.1. Précautions à prendre à la mise en marche de la pompe hydraulique :**

Lors de la première mise en marche, après montage de l'installation ou après renouvellement du fluide hydraulique, suivre le procédé décrit ci-dessous afin de prévenir un grippage ou un endommagement des pièces composantes, qui peuvent raccourcir la durée de vie de l'équipement :

- 1) Enlever le bouton de remplissage **1** situé à la face supérieure de la pompe.
- 2) Charger le même fluide hydraulique que celui contenu dans le réservoir hydraulique et remplir la pompe du fluide hydraulique. (Quantité : environ 200 ml).
- 3) Visser fermement le bouchon de remplissage **1** après le chargement.
- 4) Répéter la mise en marche et l'arrêt du moteur au moyen des boutons de « Marche » et « Arrêt » disposés sur le tableau de commande.
- 5) En principe, des bruits anormaux discontinus émis au début doivent disparaître par la répétition de mise en marche et d'arrêt, et changent en bruits stables continus.
- 6) Une fois que les bruits de fonctionnement sont stabilisés, l'opération de mise en marche est terminée. Procéder aux opérations normales de la pompe.



**Figure II.4 :** Schéma de la pompe hydraulique reliée à l'expansur ACE 25AR.

**Remarque :**

- Si une des deux portes de sécurité est ouverte, la mise en marche de la pompe est impossible au moyen du bouton de « Marche ».
- Si le mode de fonctionnement n'est pas « Manuel », la mise en marche de la pompe est impossible au moyen du bouton de « Marche ».

**2.3.2. Préparation de la mise en marche :**

1) Ouvrir complètement les robinets de vapeur, d'eau de refroidissement et d'air, et s'assurer que les manomètres de chaque utilité indiquent la pression prévue en haut du tableau de commande.

Vapeur : 6 à 8 kg/cm<sup>2</sup>

Eau de refroidissement : 3 à 8 kg/cm<sup>2</sup>

Air : 4 à 10 kg/cm<sup>2</sup>

2) S'assurer que le raccord aux matrices des flexibles de vapeur, d'eau de refroidissement et de drainage est correct et non desserré.

3) S'assurer que le raccord du chargeur et de la broche d'éjection aux matrices est correct et non desserré.

4) S'assurer que le raccord du tube en nylon des 3 flexibles au chargeur est correct.

5) Normalement, les appareils se trouvent chacun (à l'arrêt) dans l'état suivant : confirmer qu'ils sont dans leur état correct.

- Chargeur : En avance
- Obturateur de trémie : Fermé
- Soupape d'air de remplissage : Fermée
- Robinet de vapeur : Fermé
- Robinet d'eau de refroidissement : Fermé
- Robinet de drainage : Ouvert
- Robinet d'air d'eau d'évacuation : Fermé
- Robinet d'air de démoulage : Fermé
- Robinet d'air d'éjection : Fermé
- Robinet d'évacuation d'air : Fermé (seulement pour ACE7AR)

**2.3.3. Modes de commande :**

Par le réglage du sélecteur de mode de commande (SW-3), l'expansur peut opérer en trois modes différents :

**a) Commande manuelle :**

1. Appuyer sur le bouton de mise en marche. La pompe hydraulique commence à tourner.
2. Appuyer sur le bouton d'ouverture et de fermeture de la matrice. La matrice s'ouvre ou se ferme.
3. Sur la commande du commutateur à ressort sur le tableau de commande, l'électrovanne du chargeur ou de l'obturateur se met en marche.

**b) Commande semi-automatique :**

1. Appuyer sur le bouton « marche ». La machine procède au formage automatique d'un seul cycle puis s'arrête.
2. Appuyer sur le bouton « préchauffage ». La machine procède au préchauffage et puis s'arrête.

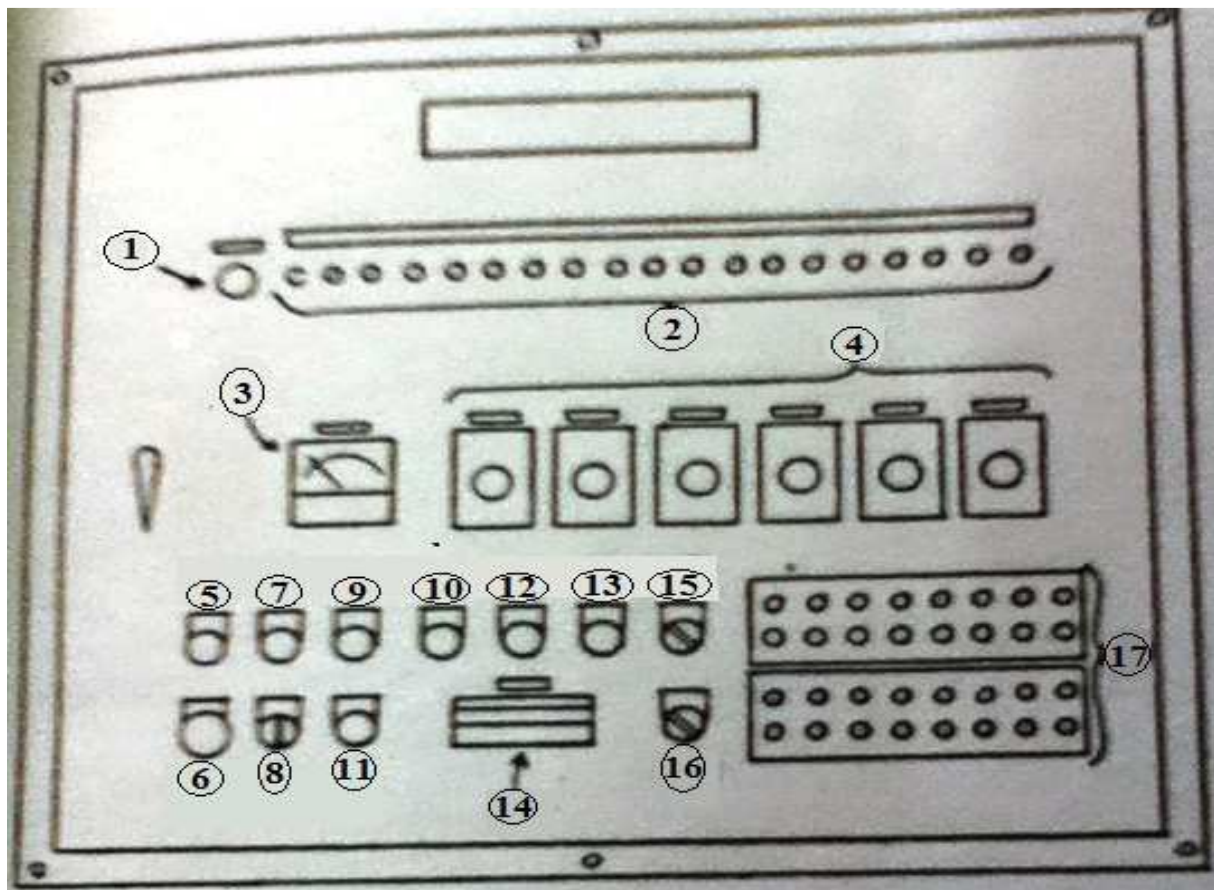
**c) Commande automatique :**

1. Appuyer sur le bouton « marche ». La machine répète le formage automatique jusqu'à ce qu'on appuie sur le bouton « arrêt » ou que le système de sécurité (tel que la porte de sécurité) fonctionne.
2. Appuyer sur le bouton « préchauffage ». La machine procède au préchauffage, et puis, le préchauffage fini, elle procède automatiquement au formage automatique.

**2.3.4. Alimentation électrique :**

Ouvrir le tableau de commande, et mettre le disjoncteur de court-circuitage **1** et le disjoncteur de circuit de commande **2** en position « fermée ». La machine est alors alimentée en énergie électrique et la lampe témoin s'allume sur le tableau.

**2.3.5. Explication sur le tableau de commande :**



**Figure II.5 :** Représentation du tableau de commande.



**Figure II.6 :** Image photographique du tableau de commande.

- 1. Lampe témoin d'alimentation électrique de la machine :** Elle s'allume sur l'alimentation électrique de la machine.
- 2. Lampe témoin d'opération :** En commande automatique, elle indique l'opération que la machine poursuit actuellement.
- 3. Ampèremètre :** Il indique l'ampérage du moteur hydraulique.
- 4. Minuterie :** Elle sert à la commande de diverses opérations. De gauche à droite, « chargement des matières premières », « chauffage des matrices », « chauffage unilatéral contraire », « chauffage bilatéral », « refroidissement à eau » et « refroidissement par radiation ».
- 5. Bouton « mise en marche » :** Le bouton poussoir de mise en marche de la machine. La machine mise en marche, la lampe s'allume. Le moteur hydraulique commence à tourner en mode « manuel », mais ne tourne pas en mode « automatique ».
- 6. Bouton « arrêt d'urgence » :** Ce bouton pressé, la machine s'arrête. En commande automatique, la machine retient l'opération en cours à moins d'une remise à zéro.
- 7. Lampe témoin du moteur de l'unité hydraulique :** Elle témoigne la marche du moteur de l'unité hydraulique.
- 8. Sélecteur de mode :** Il sert à la sélection du mode manuel, semi-automatique ou automatique.

**9. Bouton « commande automatique » :** Il met la machine sous commande automatique. La machine mise sous commande automatique s'allume.

**10. Bouton « préchauffage » :** Met la machine sous préchauffage. La machine mise sous préchauffage s'allume.

**11. Bouton « remise à zéro » :** Il sert à initialiser le mode « automatique », Ce bouton pressé, la machine revient au départ de l'opération en cours. Le bouton est valable seulement en mode « automatique ».

**12. Bouton « fermeture de matrice » :** Il s'allume pendant la fermeture de la matrice. En mode « manuel », la matrice se ferme tant que ce bouton est pressé.

**13. Bouton « ouverture de la matrice » :** Il s'allume pendant l'ouverture de la matrice. En mode « manuel », la matrice s'ouvre tant que ce bouton est pressé.

**14. Compteur :** En mode « automatique », Il compte le nombre de cycles, comptant un par cycle. En mode « semi-automatique » aussi, il compte un cycle, à moins que la machine soit arrêtée en cours d'opération. Il ne compte pas en préchauffage ni en mode « manuel ». On peut en profiter pour pointer les produits.

**15. Sélecteur de vitesse :** Il choisit la vitesse d'ouverture ou de fermeture de la matrice. Il est valable en mode « manuel » et « automatique ». Pour la commande « automatique », il faut choisir « rapide ».

**16. Sélecteur de chauffage unilatéral :** On choisit le sens de chauffage unilatéral. On choisira soit F-M, soit M-F selon les conditions de formage. Choisir d'habitude le sens F-M.

**17. Commutateur à ressort :** En mode manuel, ce commutateur est réglé sur « manuel », l'électrovanne est affichée etc se met en service. La lampe s'allume pendant la marche de l'électrovanne.

En commande « automatique », ne pas oublier de régler ce sélecteur sur « automatique ». S'il est réglé sur « manuel », l'électrovanne ne fonctionne pas et empêche un bon formage.

**Remarques pour la manœuvre :**

1) La « remise à zéro » n'est valable qu'en mode « manuel ».

2) Ne pas oublier de régler le commutateur à ressort sur « automatique » en mode «commande automatique ».

3) Quand on arrête par le bouton d'arrêt d'urgence la machine sous commande « automatique », celle-ci, à sa remise en marche, recommence son travail par l'opération interrompue. La minuterie sera remise à zéro et travaillera pour le temps consigné, et puis passera à l'opération suivante.

### **2.3.6. Commande manuelle :**

- 1) Mettre la machine sous tension, et la mettre en marche en appuyant sur le bouton « mise en marche », Attention : la machine ne se mettra pas en marche si une ou deux des portes de sécurité se trouve ouverte.
- 2) La commande des boutons sur le tableau de commande permet l'ouverture ou la fermeture des matrices ou la mise en service des électrovannes.

### **2.3.7. Commande automatique :**

- 1) La machine procède au formage automatique sous la commande du séquenceur.
- 2) La désignation et le rôle de chaque opération est comme suit :
  1. **Fermeture de matrice** : Ferme la matrice complètement, à 2 ou 3 mm près.
  2. **Pré soufflage** : L'alimentateur souffle la matrice pour enlever les gouttes d'eau qui y restent. La minuterie intérieure du séquenceur commande l'opération.
  3. **Chargement des matières premières** : Ouvrir l'obturateur de la trémie. La minuterie commande l'opération.
  4. **Purge en retour** : Pour éviter le séjour des matières premières dans le chargeur, renvoyer les matières premières à la trémie. L'opération est commandée par la minuterie intérieure du séquenceur.
  5. **Chauffage de matrice** : Injecter la vapeur dans la matrice pour préchauffer celle-ci et évacuer l'air restant.
  6. **Chauffage unilatéral 1** : Faire traverser la vapeur de la cavité au noyau(ou du noyau à la cavité) pour la fusion intérieure du produit formé.
  7. **Chauffage unilatéral 2** : Le manomètre à contacts commande la même opération que celle citée en haut.
  8. **Chauffage unilatéral contraire** : Pour équilibrer l'adhésion entre le côté cavité et le côté noyau, effectuer une opération contraire à l'opération précédente. L'opération est commandée par la minuterie.
  9. **Chauffage bilatéral 1** : Envoyer la vapeur comprimée aux deux faces pour réaliser l'adhésion complète. L'opération est commandée par le manomètre.
  10. **Chauffage bilatéral 2** : La même opération que la précédente est commandée par la minuterie.
  11. **Evacuation d'air** : Décharger la pression intérieure en ouvrant le robinet de drainage. L'opération est commandée par la minuterie.

**12. Refroidissement à l'eau :** L'eau jaillit des buses intérieures de la matrice pour refroidir le produit, L'opération est commandée par la minuterie.

**13. Evacuation d'eau :** Par l'injection d'air d'évacuation, enlever l'eau restante dans la matrice.

**14. Refroidissement par radiation :** Laisser le produit tel quel jusqu'à ce que sa pression d'expansion baisse permettant l'enlèvement du produit.

**15. Démoulage 1 :** Pour laisser le produit formé dans la cavité, injecter l'air à travers le noyau pour faciliter la séparation du produit par rapport au noyau. L'opération est commandée par le manomètre à contacts.

**16. Démoulage 2 :** En ouvrant la matrice, injecter l'air de séparation.

**17. Ouverture de la matrice :** Procéder à l'ouverture de la matrice.

**18. Ejection :** Pour faciliter la séparation de la cavité et du produit, injecter l'air d'éjection en ouvrant la matrice.

**19. Attente de fermeture de la matrice :** Mettre en attente jusqu'à ce que le produit tombe complètement.

3) Sous commande automatique, la machine, à la fin de la 19<sup>ème</sup> opération, revient automatiquement à la 1<sup>ère</sup> opération pour reprendre l'ensemble du cycle.

4) Sous commande semi-automatique, la machine, à la fin de la 19<sup>ème</sup> opération arrête son travail. Si on appuie de nouveau sur le bouton de mise en marche, la machine reprend son travail dès la première opération.

### **2.3.8. Mise à l'arrêt :**

A la fin de l'opération, suivre la procédure suivante pour arrêter la machine :

1) Pour éviter l'endommagement de la matrice et la broche d'éjection, fermer la matrice presque complètement, de 5 à 10 mm prés.

2) Couper l'alimentation électrique de la machine.

3) Fermer complètement les robinets principaux de vapeur, d'eau de refroidissement et d'air.

4) Effectuer le drainage du filtre à air.

### **2.3.9. Inter-verrouillage de pression pneumatique :**

Les vannes et les soupapes de la machine de formage sont commandées par l'air comprimé. Elles ne fonctionnent pas si l'air n'est pas sous pression correcte. Si la pression d'air baisse au cours de la commande automatique de la machine, cela provoque la production de plusieurs produits défectueux. Pour éviter une telle situation, la commande automatique s'arrête si la pression d'air baisse au dessous de 4 Kg/cm<sup>2</sup>. Sur le rétablissement de la

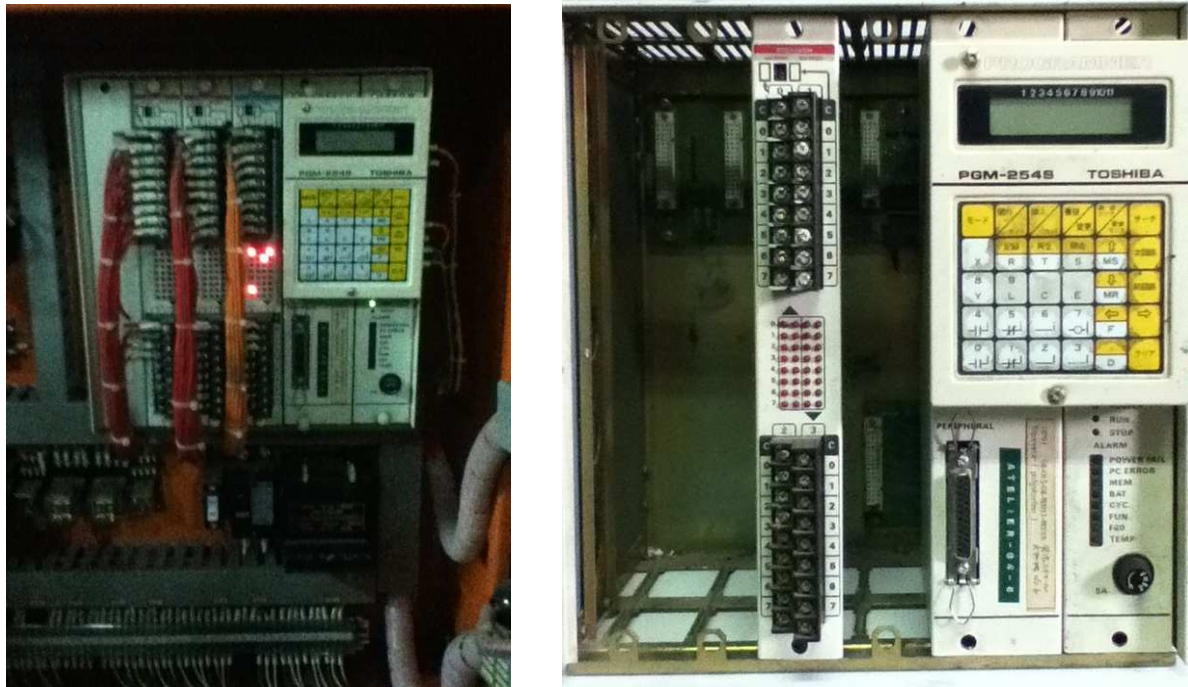
pression d'air, la machine se remet en marche automatiquement en reprenant l'opération interrompue.

**2.3.10. Consignes de sécurité :**

- 1) S'assurer quotidiennement du bon fonctionnement du bouton d'arrêt d'urgence et de la porte de sécurité.
- 2) Ne jamais pénétrer dans la machine en marche, pour n'importe quelle raison.
- 3) En cas de pénétration dans la machine pour remplacement de la matrice etc, prendre de bonnes précautions contre la mise en marche imprévue de la machine par un étranger, en affichant une pancarte ou en postant un surveillant. [15]

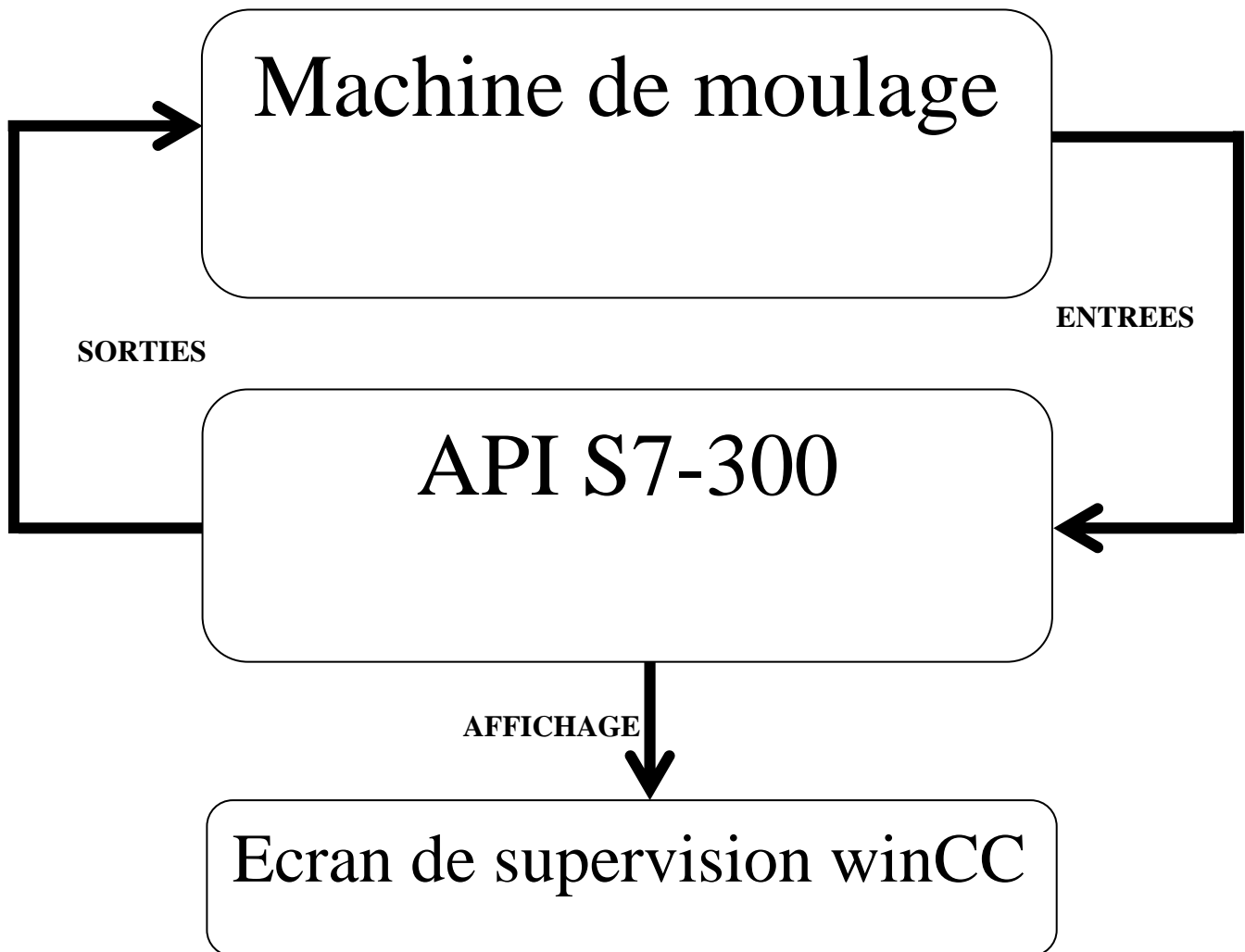
**2.4. Automate assurant la commande de l'expandeur ACE 25 AR :**

L'automate qui assure la commande de notre machine est un vieil automate toshiba, de référence PGM-254S.



**Figure II.7 :** Image photographique de l'automate (Toshiba) assurant la commande de l'expandeur.

Schéma synoptique du travail à réaliser :



**Figure II.8** : Schéma synoptique du travail à réaliser.

### 3. Conclusion :

Dans ce chapitre nous avons présenté la machine de moulage appelée EXPANSEUR ACE 25 AR, on a décrit son mode opératoire et ses différents modes de fonctionnement (automatique, semi-automatique et manuel), on a présenté l'automate assurant la commande de notre machine, et on a constaté la nécessité de son remplacement vu le développement du monde des automates causant la disparition des anciens automates et ainsi l'absence de pièces de remplacement en cas de pannes sur notre automate.

Nous verrons dans le chapitre à venir, l'automate proposé pour remplacer l'automate en place ainsi que sa programmation.

# **Chapitre III :**

**Les automates programmables industriels (API)**

**1. Introduction :**

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

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

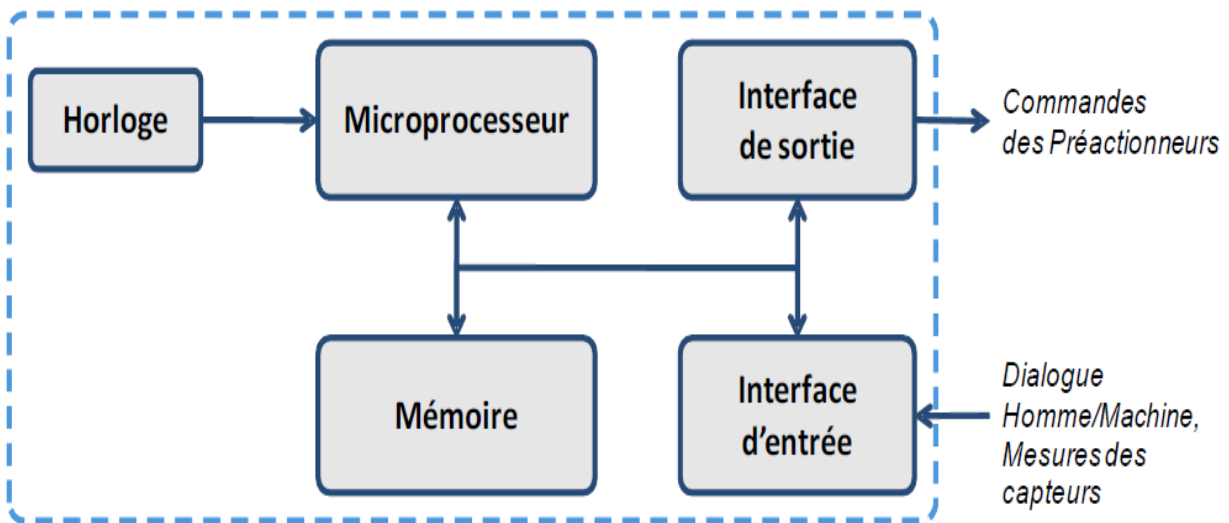
Il est en général manipulé par un personnel électromécanicien. Le développement de l'industrie à entraîner une augmentation constante des fonctions électroniques présentes dans un automatisme c'est pour ça que l'API s'est substitué aux armoires à relais en raison de sa souplesse dans la mise en œuvre, mais aussi parce que les coûts de câblage et de maintenance devenaient trop élevés. [6]

**2. Constitution d'un API :**

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

Un API se compose donc de trois grandes parties :

- Le processeur.
- La zone mémoire.
- Les interfaces Entrées/Sorties. [7]



**Figure III.1:** Structure d'un API. [2]

**2.1. Description des éléments d'un API :**

**2.1.1. Le microprocesseur :**

Le microprocesseur réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul... à partir d'un programme contenu dans sa mémoire. Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons parallèles appelées 'BUS' qui véhiculent les informations sous forme binaire.

### 2.1.2. La zone mémoires :

#### La zone mémoire va permettre :

- De recevoir les informations issues des capteurs d'entrées.
- De recevoir les informations générées par le processeur et destinées à la commande des sorties (valeur des compteurs, des temporisations, ...).
- De recevoir et conserver le programme du processus.

### 2.1.3. Les interfaces d'entrées/sorties :

- Les entrées reçoivent des informations en provenance des éléments de détection (capteurs) et du pupitre opérateur (BP).
- Les sorties transmettent des informations aux pré-actionneurs (relais, électrovannes ...) et aux éléments de signalisation (voyants) du pupitre.

## 3. Fonctionnement d'un automate programmable industriel :

Tous les automates fonctionnent selon le même mode opératoire :

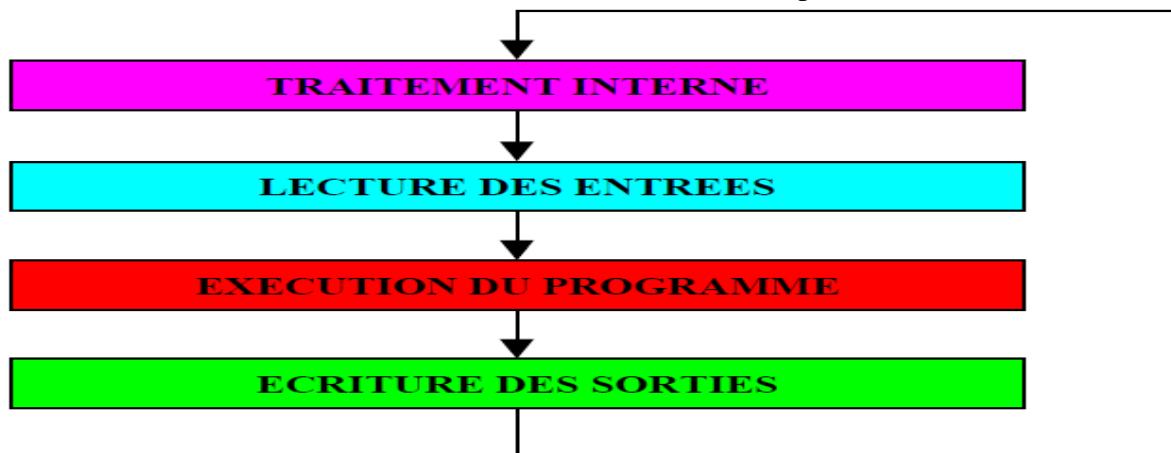


Figure III.2: Schéma récapitulatif du fonctionnement d'un API.

- **Traitement interne** : L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).
- **Lecture des entrées** : L'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.
- **Exécution du programme** : L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.
- **Ecriture des sorties** : L'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique).

## 4. Alimentation de l'automate programmable industriel :

L'automate est alimenté généralement par le réseau monophasé 230V ; 50 Hz, mais d'autres alimentations sont possibles (110V etc ...).

## Chapitre III : Les automates programmables industriels (API)

La protection sera de type magnétothermique (voir les caractéristiques de l'automate et les préconisations du constructeur).

Il est souhaitable d'asservir l'alimentation de l'automate par un circuit de commande spécifique (contacteur KM1).

De même, les sorties seront asservies au circuit de commande et alimentées après validation du chien de garde. [9]

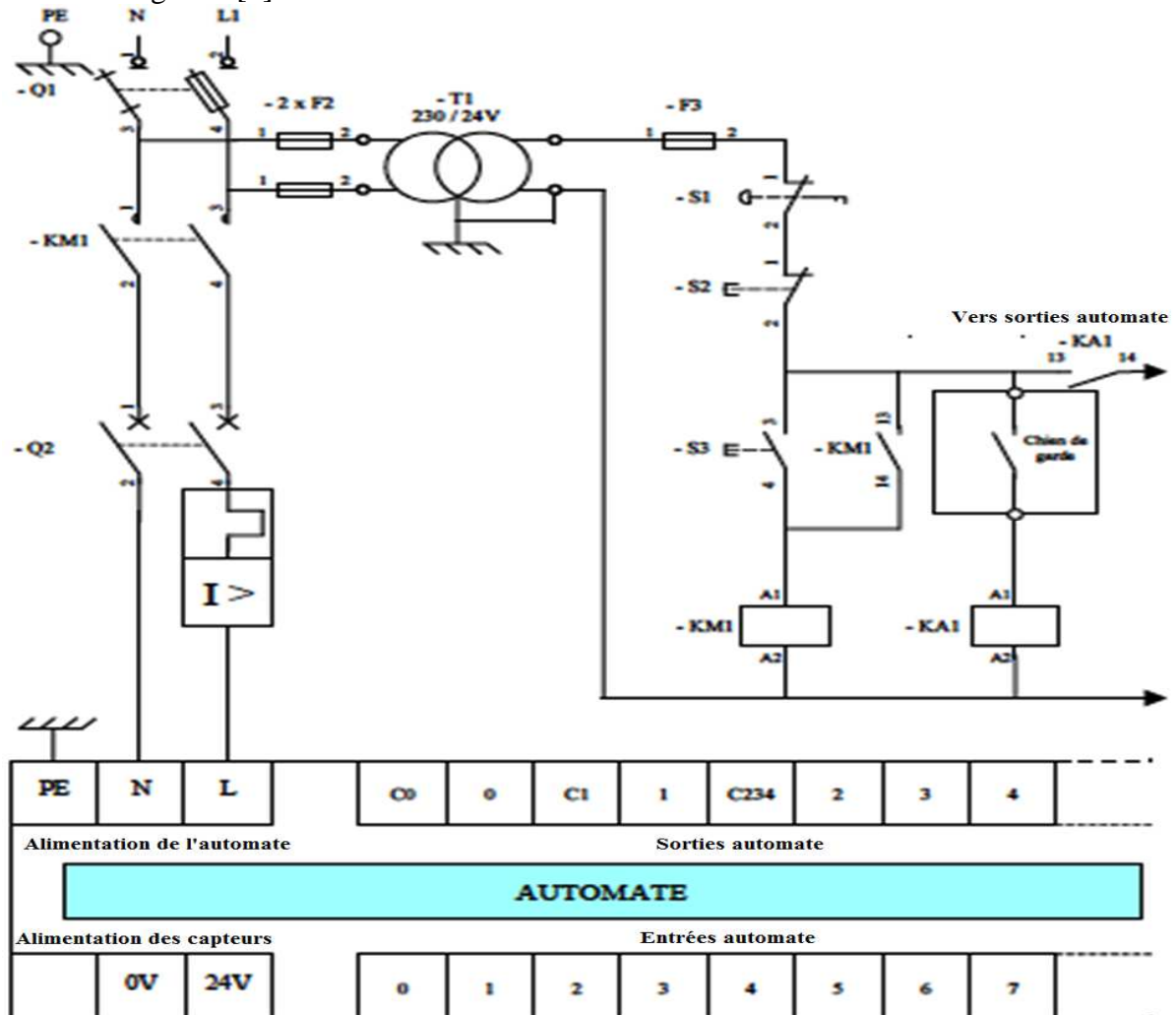


Figure III.3: Schéma de l'alimentation de l'automate.

### 5. Programmation des API :

#### 5.1. Introduction :

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

- **Le GRAFCET ou SFC** : Ce langage de programmation de haut niveau permet la programmation aisée de tous les procédés séquentiels.
- **Schéma à relais ou LD** : Ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (true/false).

- **Schéma par blocs ou FBD** : Ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables.
- **Texte structuré ou ST** : Ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithmes plus ou moins complexes.
- **Liste d'instructions ou IL** : Ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur. [11]

### 5.2. Le GRAFCET:

#### 5.2.1. Définition :

Le GRAFCET (GRAPhe Fonctionnel de Commande des Etapes et Transitions) a été proposé par ADEPA (agence pour le développement de la Productique Appliquée à l'industrie) en 1977 et normalisé en 1982 par la NF C03-190.

Le GRAFCET est un langage fonctionnel graphique destiné à décrire les différents comportements d'un automatisme séquentiel. Il aide à la réalisation, il apporte une aide appréciable lors de l'exploitation de la machine pour les dépannages et les modifications.

Le GRAFCET représente l'évolution d'un cycle comprenant des étapes et des transitions.

#### 5.2.2. Les différentes représentations du GRAFCET :

Il existe deux types de représentations :

**La représentation fonctionnelle (ou de niveau 1)** : Elle donne une interprétation de la solution retenue pour un problème posé, en précisant la coordination des tâches opératives. Elle permet une compréhension globale du système.

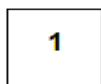
**La représentation technologique (ou de niveau 2)** : Elle donne une interprétation en tenant compte des choix technologiques relatifs à la partie commande de l'automatisme ; le type et la désignation des appareillages (S1, KM, Ka...). [10]

**La représentation de niveau 3** : C'est une représentation semblable à celle du niveau 2, la différence réside dans l'affectation des adresses correspondantes (à la sortie d'un vérin ou à l'excitation d'une électrovanne par exemple) sur l'automate programmable.

#### 5.2.3. Eléments graphiques de base d'un GRAFCET :



**Etape initiale** : L'étape initiale caractérise l'état du système au début du fonctionnement.



**Etape** : Une étape correspond à un comportement stable du système. Les étapes sont numérotées dans l'ordre croissant. A chaque étape, on peut associer une ou plusieurs actions.

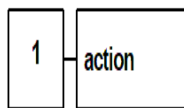
**+ Transitions** : Les transitions indiquent les possibilités d'évolutions du cycle, à chaque transition est associée une réceptivité.

## → réceptivité

**Réceptivité :** La réceptivité est la condition logique pour l'évolution du GRAFCET. Si la réceptivité est vrai (=1), le cycle peut évoluer. Les réceptivités proviennent du pupitre de commande, des fins de courses ou d'informations provenant de la partie opérative.



**Liaisons orientées :** Le GRAFCET se lit de haut en bas, autrement il est nécessaire d'indiquer son évolution avec des liaisons orientées constituées de flèches indiquant le sens.



**Action :** L'action est associée à une étape, elle est active lorsque le cycle est arrivé à cette étape. Il est possible de définir les actions conditionnelles, temporisé . . . (électrovanne, enclenchement d'un contacteur. . .).

- **Etape active :** le point indique que l'étape est active. [10]

### 5.2.4. Les règles d'évolution du GRAFCET :

**Règle N°1 – Condition initiale :** A l' instant initial, seules les étapes initiales sont actives.

**Règle N°2 – Franchissement d'une transition :** Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, et seulement si la réceptivité associée est vraie.

**Règle N°3 – Evolution des étapes actives :** Le franchissement d'une transition entraîne obligatoirement dans cet ordre la désactivation de toutes ses étapes amont et l'activation de ses étapes aval.

**Règle N°4 – Franchissement simultané :** Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.

**Règle N°5 – Conflit d'activation :** Si une étape doit être simultanément désactivée par le franchissement d'une transition aval, et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes au procédé) non désirées. [12]

### 5.2.5. Structures de base d'un GRAFCET :

Nous pouvons avoir dans un cycle machine complet, des séquences simultanées, ou des choix de séquence.

5.2.5.1. Divergence et convergence en ET :

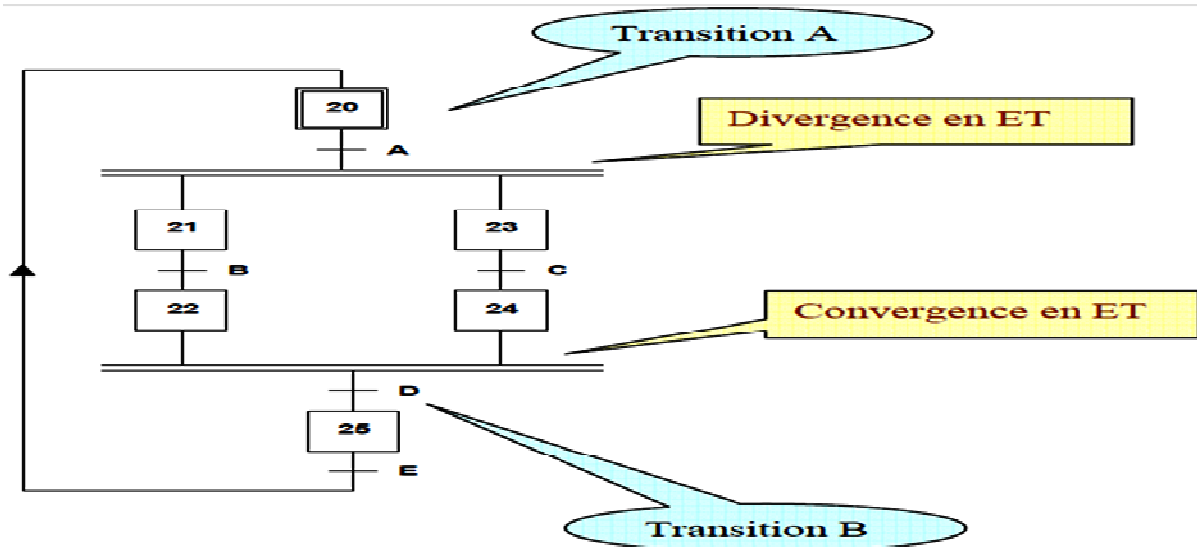


Figure III.4 : Représentation d'un exemple de divergence et convergence en ET.

**Divergence en ET :** Elle est représentée par 2 traits identiques et parallèles ; lorsque la transition A est franchie les étapes 21 et 23 sont actives.

**Convergence en ET :** Après une divergence en ET on trouve une convergence en ET. La transition D sera active lorsque les étapes 22 et 24 seront actives, si la réceptivité associée à la transition D est vraie, alors elle est franchie et l'étape 25 devient active et désactive les étapes 22 et 24.

**Remarque :** Le nombre de branches peut être supérieur à 2.

5.2.5.2. Divergence et convergence en OU (aiguillage) :

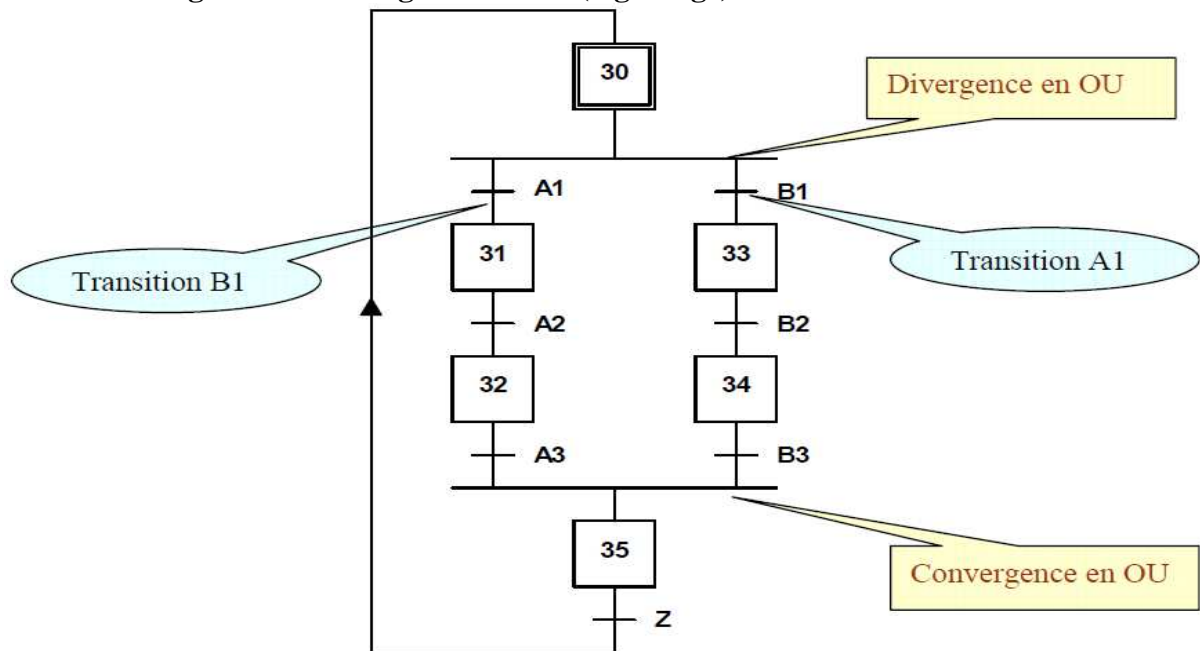


Figure III.5 : Représentation d'un exemple de divergence et convergence en OU.

**Divergence en OU :** L'évolution du système se dirige vers une des branches en fonction des réceptivités A1, B1 et de leurs transitions associées.

**Convergence en OU :** Après une divergence en OU on trouve une convergence en OU vers une étape commune, dans l'exemple c'est l'étape 35.

**Remarque :** Le nombre de branche peut être supérieur à 2, A1 et B1 ne peuvent pas être vrais simultanément.

### 5.2.5.3. Saut d'étape :

Le saut d'étape permet de sauter une ou plusieurs étapes en fonction de la progression d'un cycle.

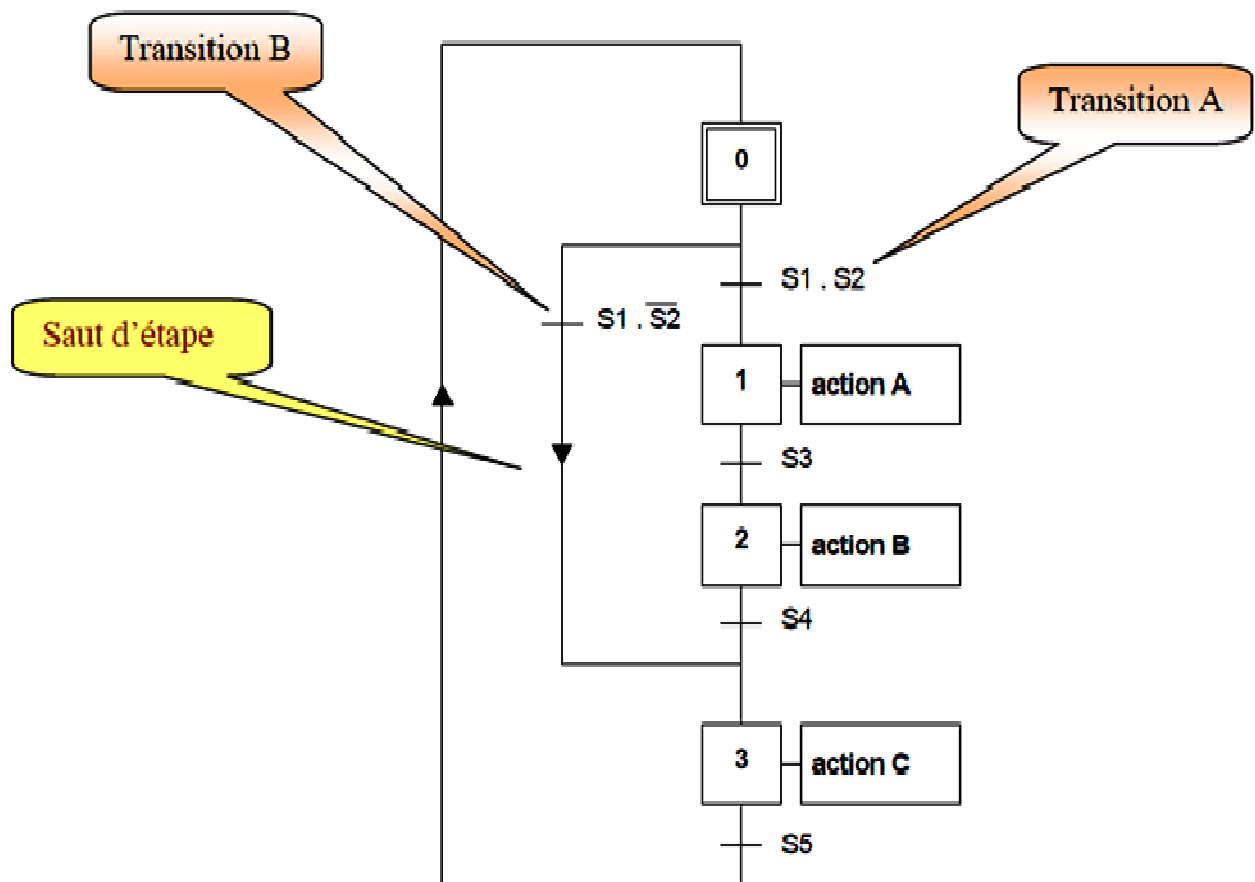


Figure III.6 : Représentation d'un exemple de saut d'étape.

Sur le GRAFCET ci-dessus après l'étape initiale 0 un choix entre 2 transitions A et B s'effectue.

La transition A associée à sa réceptivité nous permet de continuer le cycle sur l'étape 1.

La transition B associée à sa réceptivité nous permet de passer à l'étape 3, les étapes 1 et 2 sont ignorées lors du cycle.

### 5.2.5.4. Reprise d'étape :

La reprise d'étape permet de ne pas continuer le cycle mais de reprendre une séquence précédente lorsque les actions à réaliser sont répétitives.

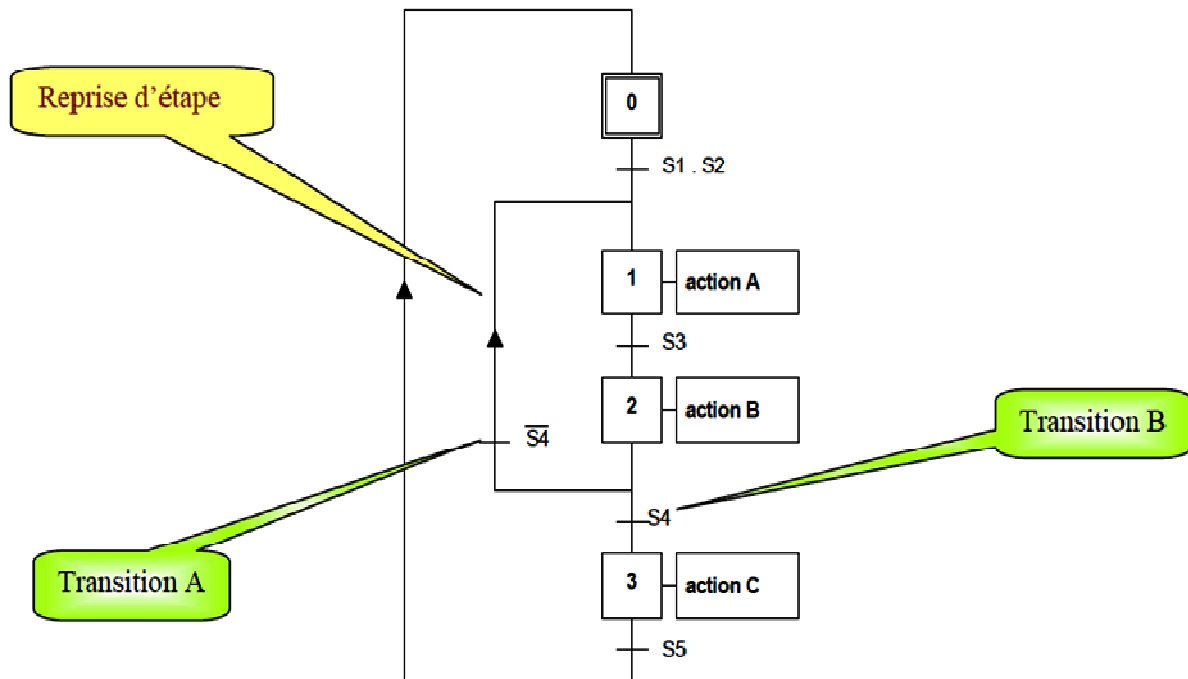


Figure III.7 : Représentation d'un exemple de reprise d'étape.

Sur le GRAFCET ci-dessus après l'étape 2 un choix entre 2 transitions A et B s'effectue. La transition A associée à sa réceptivité nous permet de reprendre le cycle sur l'étape 1. La transition B associée à sa réceptivité nous permet de passer à l'étape 3. [10]

### 5.3. Le langage à contacts ou Ladder :

#### 5.3.1. Définition :

Le langage à contacts est adapté à la programmation de traitements logiques, il utilise le schéma développé.

Nous retrouvons : La fonction ET en utilisant des contacts en série.

La fonction OU en utilisant des contacts en parallèle.

#### 5.3.2. Représentation de quelques éléments du langage Ladder :

Graphes	Désignation	Fonction	Schéma à contact
	Contact à fermeture	contact passant quand il est actionné	
	Contact à ouverture	contact passant quand il n'est pas actionné	
	connexion horizontale	permet de relier les éléments action série	
	connexion verticale	permet de relier les éléments action en parallèle	
	bobine directe	la sortie prend la valeur du résultat logique	

Figure III.8: Représentation de quelques éléments du langage Ladder.

5.4. Schéma en blocs fonctionnels (FBD) – Function Block Diagram :

5.4.1. Introduction :

Le schéma en blocs fonctionnels (FBD) est un langage graphique de programmation très courant dans l'industrie des processus. Il schématise le système de commande en termes de flux des signaux entre différents éléments de traitements. Associé au diagramme Ladder, il permet d'utiliser toutes les fonctions et tous les blocs fonctionnels des différentes bibliothèques.

Chaque fonction, chaque bloc fonctionnel est représenté par un rectangle avec, à gauche l'ensemble des paramètres d'entrée et, à droite, la ou l'ensemble des sorties.

Les entrées et les sorties booléennes sont reliées à des symboles Ladder (contacts) ou bien à des sorties et des entrées d'autres fonctions ou d'autres blocs fonctionnels.

Les entrées et les sorties analogiques sont affectées à des variables (entrées / sorties physiques, variables internes) ou bien à des sorties et des entrées d'autres fonctions ou d'autres blocs fonctionnels.

5.4.2. Connexion d'un bloc fonctionnel :

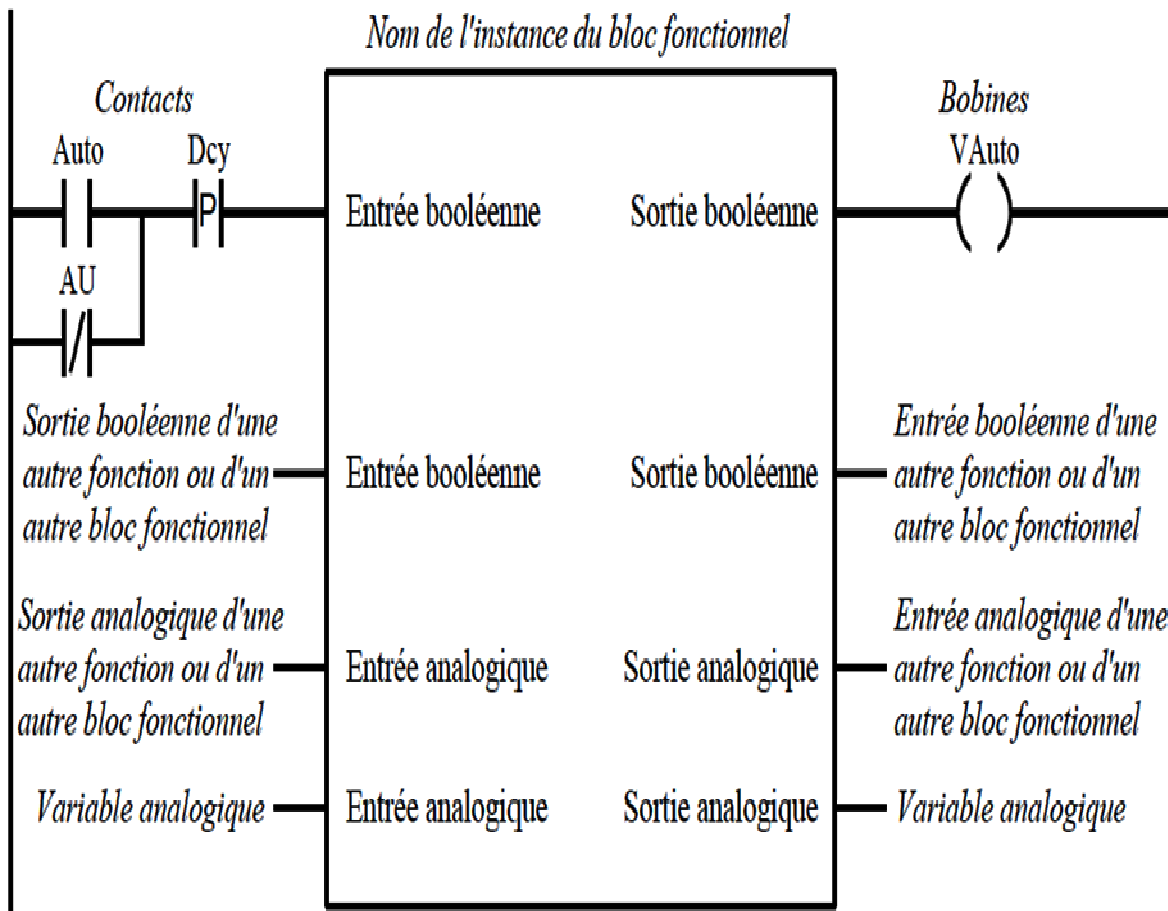


Figure III.9: Connexion d'un bloc fonctionnel.

### 5.5. Le langage « Texte structuré ou ST » :

#### 5.5.1. Introduction :

Le « Structured Text » (texte structuré) est, comme le « C » ou, le Pascal, un langage textuel de programmation structurée. C'est un langage performant et adapté aux systèmes d'automatisation.

Il est parfaitement adapté pour la définition des blocs fonctionnels complexes ou non qui seront ensuite employés dans les autres langages. C'est le langage textuel à privilégier.

#### 5.5.2. Structure générale du ST :

La structure générale d'une instruction est la suivante :

**<Variable> := <opérande> <opérateur> <opérande> .... ;**

Chaque instruction se termine par le caractère « ; »

#### 5.5.3. Quelques opérateurs du langage ST:

<i>Opérateur</i>	<i>Description</i>	<i>Opérateur</i>	<i>Description</i>
<b><i>Opérateur d'affectation</i></b>			
:=	Opérateur d'affectation		
<b><i>Opérateurs logiques</i></b>			
NOT	Négation logique	AND	ET logique
OR	OU logique inclusif	XOR	OU logique exclusif
<b><i>Opérateurs arithmétiques</i></b>			
+	Addition	-	Soustraction
*	Multiplication	/	Division
mod	Reste de la division entière		
<b><i>Opérateurs de comparaison</i></b>			
=	Égal à	◇	Différent de
>	Supérieur à	>=	Supérieur ou égal à
<	Inférieur à	<=	Inférieur ou égal à

**Figure III.10** : Représentation des opérateurs du ST.

### 5.6. Le langage liste d'instructions:

#### 5.6.1. Définition :

Le langage liste d'instructions permet de transcrire sous forme de liste :

- un schéma à contacts.
- un logigramme, des équations booléennes.
- un GRAFCET.

Il réalise aussi des fonctions d'automatisme telles que la temporisation, le comptage, pas à pas...etc. [10]

**5.6.2. Instructions de base en langage liste :**

Instructions de test	
Désignation	Fonctions
LD	Le résultat est égal à l'opérande (load : lire la valeur).
LDN	Le résultat est égal à l'inverse de l'opérande (contact ouverture).
AND	ET logique entre le résultat et précédent et l'état de l'opérande.
ANDN	ET logique entre le résultat et précédent et l'état inverse de l'opérande.
OR	OU logique entre le résultat et précédent et l'état de l'opérande.
ORN	OU logique entre le résultat et précédent et l'état inverse de l'opérande.
XOR, XORN	OU exclusif.
Instructions d'action	
ST	L'opérande associé prend la valeur de la zone de test.
STN	L'opérande associé prend la valeur inverse de la zone de test.
S	L'opérande associé est mis à 1 lorsque le résultat de la zone de test est à 1.
R	L'opérande associé est mis à 1 lorsque le résultat de la zone de test est à 1.

**Figure III.11 :** Instructions de base en langage liste. [10]

**Remarque :** L'adresse ou le code opérande est précédé de %.

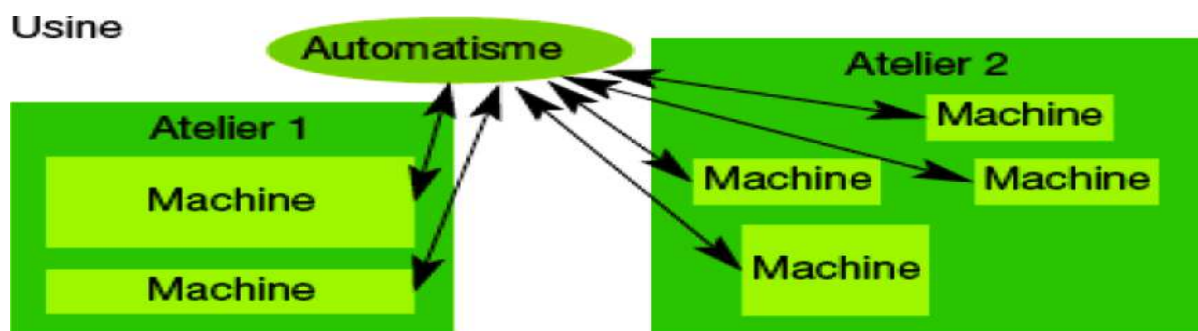
**6. Les réseaux dans l'industrie :**

**6.1. Introduction :**

Un réseau local industriel est utilisé dans une usine ou tout système de production pour connecter diverses machines afin d'assurer la commande, la surveillance, la supervision, la conduite, la maintenance, le suivi du produit, la gestion, en un mot, l'exploitation de l'installation de production. [3]

**6.2. Les automatismes centralisés :**

Jusqu' aux années 80, les automatismes, s'appuyant sur des automates programmables industriels (API) traitaient essentiellement des fonctions séquentielles.



**Figure III.12 :** Automatisme centralisé.

### 6.3. Les automatismes décentralisés :

Du fait des contraintes imposées par les systèmes centralisés, les utilisateurs se sont orientés vers une segmentation de l'architecture. Celle-ci a été faite en découpant l'automatisme en entités fonctionnelles.

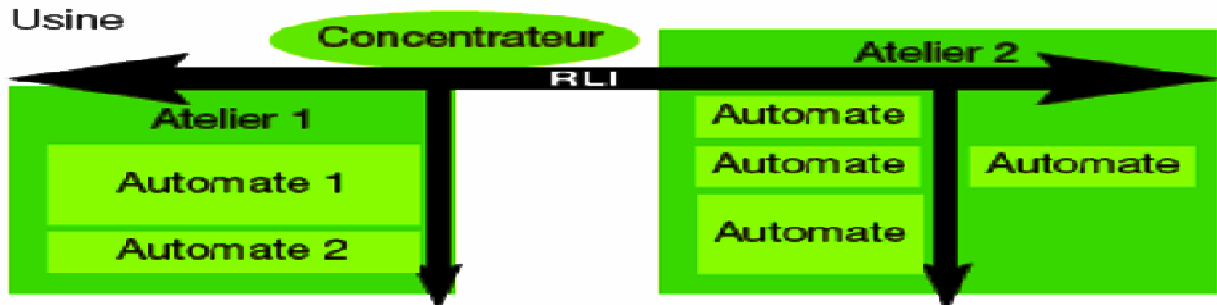


Figure III.13 : Automatisme décentralisé.

#### 6.3.1. Rôle de l'automatisme décentralisé :

- Simplifier les automatismes en réduisant le nombre d'E/S gérées.
- Faciliter la mise en service et la maintenance.

Cette segmentation a généré le besoin de communication entre les entités fonctionnelles. La fonction de communication est devenue la clef de voûte de la conception des architectures d'automatismes. [13]

### 6.4. Les réseaux de terrain :

#### 6.4.1. Définition :

Un réseau de terrain est un réseau bidirectionnel, sériel, multi branche (multi drop), reliant différents types d'équipements:

- E/S déportées, Capteurs / Actionneurs.
- Automate programmable (API), CNC.
- Calculateur, PC industriel. [12]

#### 6.4.2. Objectifs des réseaux de terrain :

Les réseaux de terrain ou bus de terrain permettent :

- La connexion de plusieurs entités d'un même système sur un même support de communication dans une zone géographique limitée (usine, atelier, automobile, électronique embarquée...).
- Le transport fiable de données sous une forme numérique d'un équipement vers un autre.
- L'ajout ou la suppression d'éléments au sein d'un même système (réduction ou extension du réseau).
- Le respect de contraintes (temps réel).

#### 6.4.3. Caractéristiques principales des réseaux de terrain :

- Sûreté de fonctionnement: perte ou détérioration d'informations (détection des erreurs), pannes d'équipements (détection et recouvrement des pannes).
- Disponibilité et prix des équipements.
- Capacité d'interconnexion à des équipements hétérogènes.

- Réseaux de faible taille (faible répartition géographique des équipements).
- Quantité relativement faible de données (notion d'évènement, variables processus, ...).
- Contraintes de temps (notion de temps réel, déterminisme, ...).
- Nombre d'équipements connectables.
- Contraintes liées à l'environnement (température, vibrations, ...etc).

**6.4.4. Classification des réseaux de terrain :**

Généralement, on regroupe sous le terme «bus de terrain» tous les bus de communication industriels. On distingue néanmoins par complexité décroissante :

- **Le bus d'usine («data bus»)** : C'est un réseau local industriel (RLI), il permet la communication entre l'automatisme et le monde informatique, il est basé sur Ethernet de type MAP (Manufacturing Automation Protocol) ou TOP (Technical and Office Protocol).
- **Le bus de terrain («field bus» et «device bus»)** : Il représente l'interconnexion des unités de traitement et des périphériques.
- **Le bus de bas niveau («sensor bus»)** : C'est un bus capteur/actionneur.

**6.5. La pyramide du CIM (Computer Integrated Manufacturing) :**

C'est une méthode largement généralisée qui représente 4 niveaux auxquels correspondent des niveaux de décision. Plus on s'élève dans la pyramide du CIM, plus le niveau de décision est important et plus la visibilité est globale. Un niveau supérieur décide ce qu'un niveau inférieur exécute.

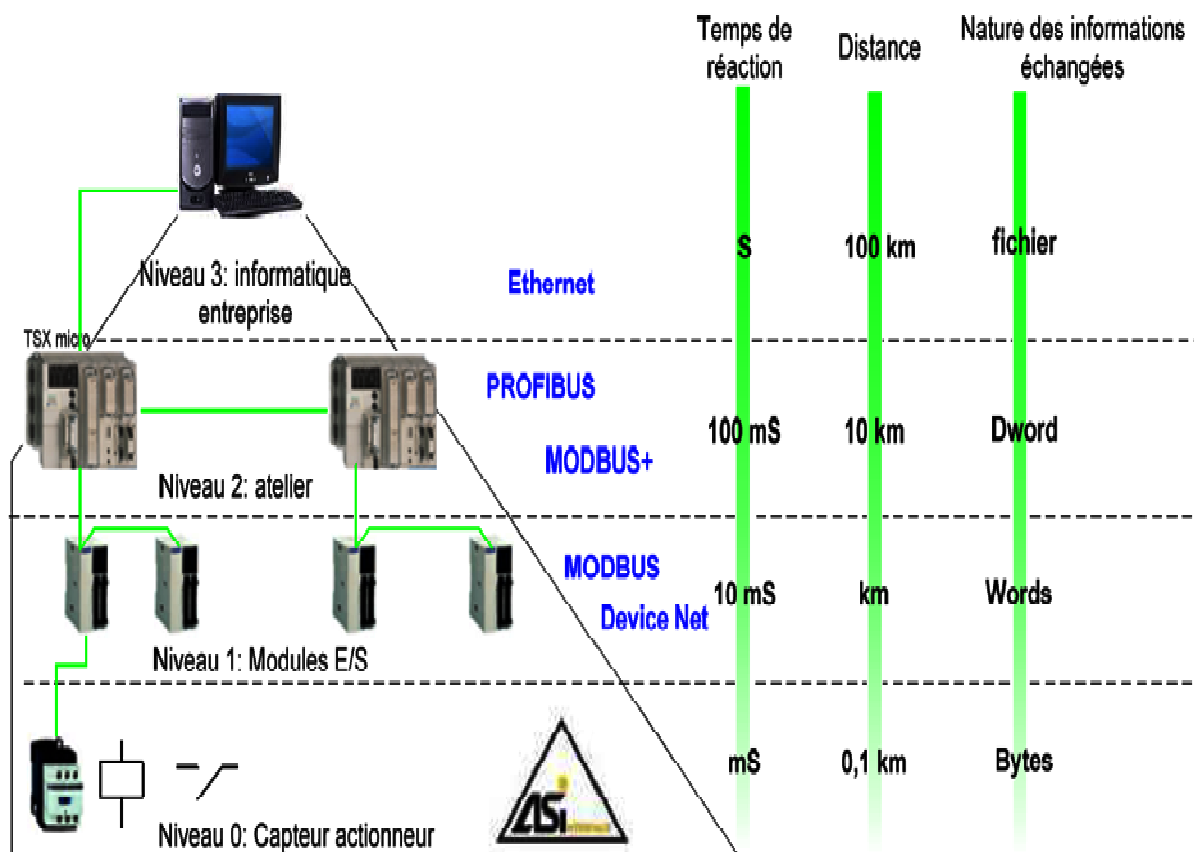


Figure III.14 : Pyramide du CIM. [21]

On distingue généralement les niveaux suivants :

- **Au niveau 3** : la gestion des produits et des stocks, la gestion des approvisionnements, la gestion des clients, des commandes et de la facturation (gérés par les ERP (entreprise ressource planning)).
- **Au niveau 2** : la localisation des produits en stocks, les mouvements physiques et la gestion des lots (géré par le système de gestion d'entrepôt).
- **Au niveau 1** : les automatismes.
- **Au niveau 0** : les capteurs et actionneurs.

Ainsi à chaque niveau, correspond un bus ou un réseau adapté aux besoins. [12]

### **7. Choix d'un automate :**

Le choix d'un automate n'est pas arbitraire, mais il se fait après que l'utilisateur ait établi le cahier des charges du système à automatiser, et cela en se basant sur un certain nombre de critères importants :

- Le nombre d'entrées/sorties.
- La nature des entrées et des sorties (numériques, analogiques).
- La nature du traitement (temporisation, comptage, etc.).
- La communication avec d'autres systèmes.
- Les moyens de sauvegarde du programme (disquettes, cassettes, etc. .)
- La fiabilité et la robustesse.
- L'immunité aux parasites et aux bruits.

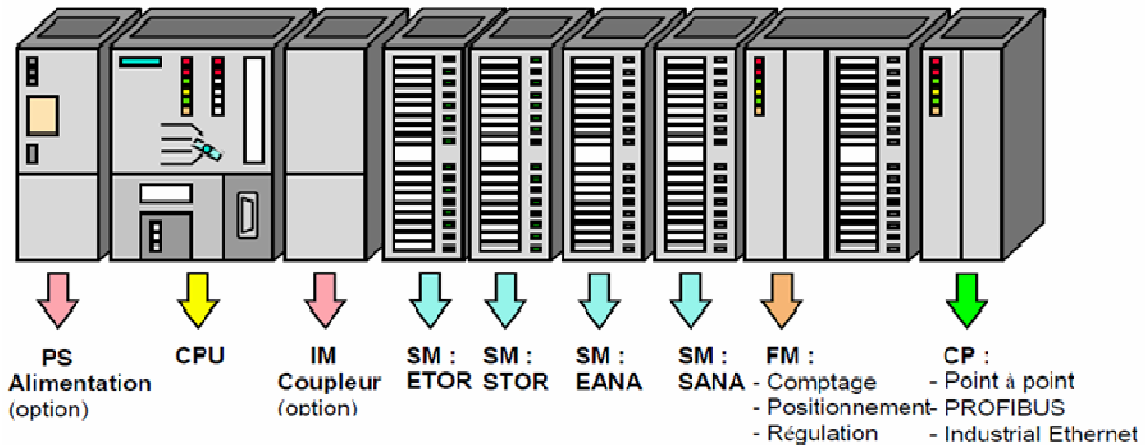
Pour notre système, nous avons choisi l'automate S7-300, car ses caractéristiques conviennent aux exigences de l'installation. De plus, il peut gérer, sans extension, 256 entrées/sorties. Avec extension jusqu'à 1024 entrées/sorties : numériques, logiques et analogiques.

### **8. L'automate S7-300 :**

#### **8.1. Constitution du S7-300 :**

L'automate programmable S7-300 (**figure III.15**) est un système d'automatisation modulaire offrant la gamme de modules suivante :

- Modules d'alimentation (PS) 2A, 5A, 10A.
- Unité centrale CPU314 travaillant avec une mémoire de 48Ko, sa vitesse d'exécution est 0,3ms/1K instructions.
- Modules de signaux (SM) pour entrées et sorties TOR et analogiques.
- Modules d'extension (IM) pour configuration multi rangée du S7-300.
- Module de fonction (FM) pour fonctions spéciales (par exemple activation d'un moteur asynchrone).
- Processus de communication (CP) pour la connexion au réseau.



**Figure III.15:** Les différents composants de l'automate programmable S7-300.

### 8.1.1. Le module d'alimentation :

Le module d'alimentation transforme la tension secteur en tension d'alimentation pour les modules électriques de l'automate programmable. Cette tension s'élève à 24V. Les tensions pour les capteurs, actionneurs et voyants lumineux qui dépassent 24V, sont fournis par des blocs d'alimentation ou transformateurs supplémentaires.

### 8.1.2. Description de la CPU :

La CPU est le cerveau de l'automate, elle lit les états des entrées, puis elle exécute le programme utilisateur en mémoire et commande les sorties.

La CPU est constituée de :

**Interface MPI :** toute CPU est équipée d'une interface MPI pour la connexion de la console de programmation (PG) ou d'un autre appareil (par exemple adaptateur PC).

**Commutateur de mode de fonctionnement :** le commutateur de mode permet de changer le mode de fonctionnement. Chaque position du commutateur de mode autorise certaines fonctions à la console de programmation. Les modes de fonctionnement suivants sont possibles :

- RUN-P : exécution du programme, accès en écriture et en lecture avec la PG.
- RUN : exécution du programme, accès en lecture seule avec la PG.
- STOP: le programme n'est pas exécuté, toutes les fonctions avec la PG sont autorisées.
- MRES : position dans laquelle un effacement général de la CPU peut être effectué.

**Signalisation des états :** certains états de l'automate sont signalés par LEDs sur la face avant de la CPU tels que :

- SF : signalisation groupée de défauts, défaut interne de la CPU ou d'un module avec fonction diagnostique.
- BATF : défaut de pile, pile à plat ou absente.
- DC5v : signalisation de tension d'alimentation 5v, allumée: les 5v sont présentes, clignotante : il y a surcharge de courant.
- FRCE : forçage, signalisation qu'au moins une entrée ou une sortie est forcée de manière permanente.
- RUN : clignotement à la mise en route de la CPU, allumage continu en mode RUN.

## **Chapitre III : Les automates programmables industriels (API)**

- STOP : allumage continu en mode STOP, clignotement lent lorsqu'un effacement
- général est requis, clignotement rapide lorsqu'un effacement général est en cours.

**Carte mémoire :** une carte mémoire peut être montée à la CPU, elle conserve le contenu du programme en cas de coupure de courant, même en l'absence de la pile.

**Pile :** elle permet de sauvegarder le contenu de la RAM en cas de coupure de courant.

### **8.1.3. Modules de coupleurs (IM) :**

Les coupleurs IM 360, IM 361 et IM 365 permettent de configurer le S7-300 sur plusieurs rangées et assurent la liaison entre les châssis et le couplage entre les différentes unités, ainsi la communication entre les entrées/sorties et d'autres périphériques et l'unité centrale.

### **8.1.4. Modules de signaux (SM) :**

- **Modules d'entrées TOR :** un module d'entrée doit permettre à l'unité centrale de l'automate, d'effectuer une lecture de l'état logique des capteurs qui lui sont associés (modules 4, 8, 16 ou 32 entrées). A chaque entrée correspond une voie qui traite le signal électrique pour élaborer une information binaire. Le processeur de l'automate vient questionner le module ; le contenu du mot d'entrées du module est alors recopié dans la mémoire de données de l'automate programmable. Les modules d'entrées «Tout Ou Rien» permettent de raccorder à l'automate différents capteurs logiques tels que les boutons poussoirs, les fins de course. Ils conviennent au raccordement d'appareils à contacts et des détecteurs de proximité en montage.
- **Modules de sorties TOR :** les modules de sorties TOR transportent le niveau du signal interne du S7-300 au niveau du signal requis par le processus. Ils conviennent par exemple au raccordement d'électrovannes, de contacteurs, de micromoteurs, de lampes et de démarreurs de moteurs, donc il permet à l'automate d'agir sur les actionneurs.
- **Modules d'entrées et de sorties analogiques :** les modules d'entrées/sorties analogiques réalisent la conversion des signaux analogiques issus du processus en signaux numériques pour le traitement interne dans le S7-300 et des signaux numériques du S7-300 en signaux analogiques destinés au processus.

### **8.1.5. Module de fonction (FM) :**

C'est un module programmable. Il nécessite une importante capacité mémoire comme : Comptage, Positionnement et Régulation.

### **8.1.6. Modules de communication :**

Les modules chargés de communication permettent d'établir des liaisons homme/machine et machine/homme. Ces liaisons sont effectuées à l'aide des interfaces de communication suivantes :

- Point à point.
- PROFIBUS.
- Industriel Ethernet.

### **8.1.7. Châssis d'extension (UR) :**

Il est constitué d'un profilé support en aluminium et d'un bus de fond de panier avec connecteur. Il est généralement installé à l'intérieur d'une armoire électrique. Il permet le raccordement électrique et le montage de divers modules.

### **8.2. Fonctionnement de l'automate programmable S7-300 :**

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

#### **a)- Réception des informations sur les états du processus :**

L'automate reçoit des informations, sur l'état du processus, via les capteurs des signaux reliés aux entrées. Le S7-300 met à jour la mémoire image des entrées au début de chaque programme, en transférant le nouvel état des signaux d'entrées des modules vers la mémoire image des entrées, ce qui permet à la CPU de savoir l'état du processus.

#### **b)- Exécution du programme utilisateur :**

Après avoir acquis les informations d'entrées et exécuté le système d'exploitation, la CPU passe à l'exécution du programme utilisateur, qui contient la liste d'instructions à exécuter pour faire fonctionner le procédé.

#### **c) - Commande du processus :**

L'automate commande le processus en appliquant une tension de 24V aux pré-actionneurs via les points de connexion de l'automate appelés sorties. Ceci permet d'activer ou de désactiver des moteurs, de faire monter ou descendre des vérins ou d'allumer ou d'éteindre des lampes.

### **8.3. Programmation de l'automate S7-300 :**

L'évolution rapide de la technique d'automatisation a donné naissance à une multitude de langages de programmation. Le logiciel STEP7 est l'outil de base pour la configuration et la programmation des systèmes d'automatisation SIMATIC (**figure III.16**). STEP7, il offre les fonctions suivantes pour l'automatisation d'une installation :

- Configuration et paramétrage du matériel.
- Création des programmes.
- Test, mise en service et maintenance de l'installation d'automatisation.
- Fonctions de diagnostic et d'exploitation lors de perturbations dans l'installation.

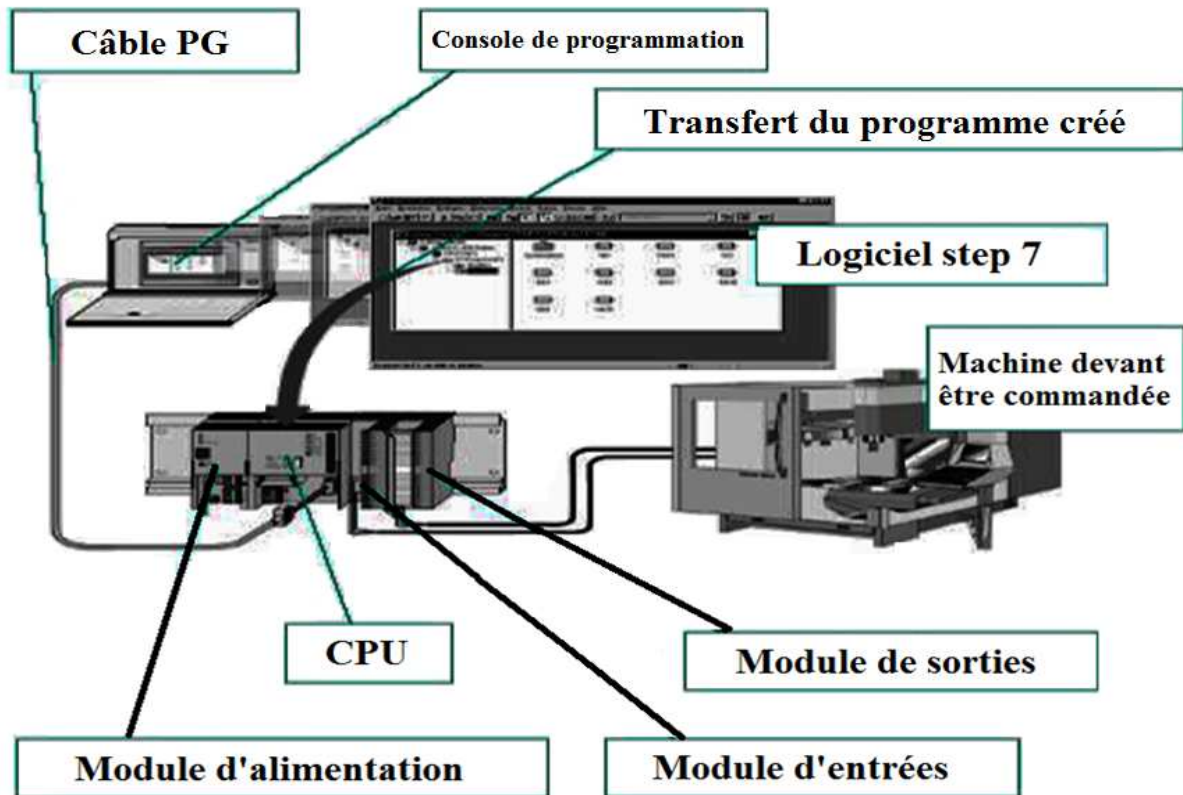


Figure III.16: Vue d'ensemble de l'automatisme.

### 8.3.1. Langage de programmation Step 7 :

Avec le STEP7, le programme peut être représenté et programmé dans trois modes différents :

- Schéma à contacts « CONT ».
- Logigramme « LOG ».
- Liste d'instructions « LIST ».

### 8.3.2. Création d'un programme utilisateur :

Un programme utilisateur comprend toutes les instructions et déclarations, ainsi que les données nécessaires au traitement de signaux de commande d'une installation ou d'un processus. Il est affecté à un module programmable (CPU par exemple) et peut être structuré en entités plus petites appelées blocs.

Un programme utilisateur devant être exécuté dans une CPU S7, est essentiellement, constitué de blocs. Il contient, en outre, des informations supplémentaires, telles que les données destinées à la configuration ou à la mise en réseau du système. En fonction de l'application, nous pouvons créer, dans le programme utilisateur, les blocs de types suivants :

- Blocs d'organisations (OB).
- Blocs fonctionnels (FB).
- Blocs fonctions (FC).
- Blocs de données (DB).
- Blocs fonctionnels système(SFB) et fonctions systèmes (SFC) (intégrés au logiciel).

Mais avant de commencer la programmation, il est nécessaire de créer un projet dans lequel les données et le programme utilisateur à créer seront structurés.

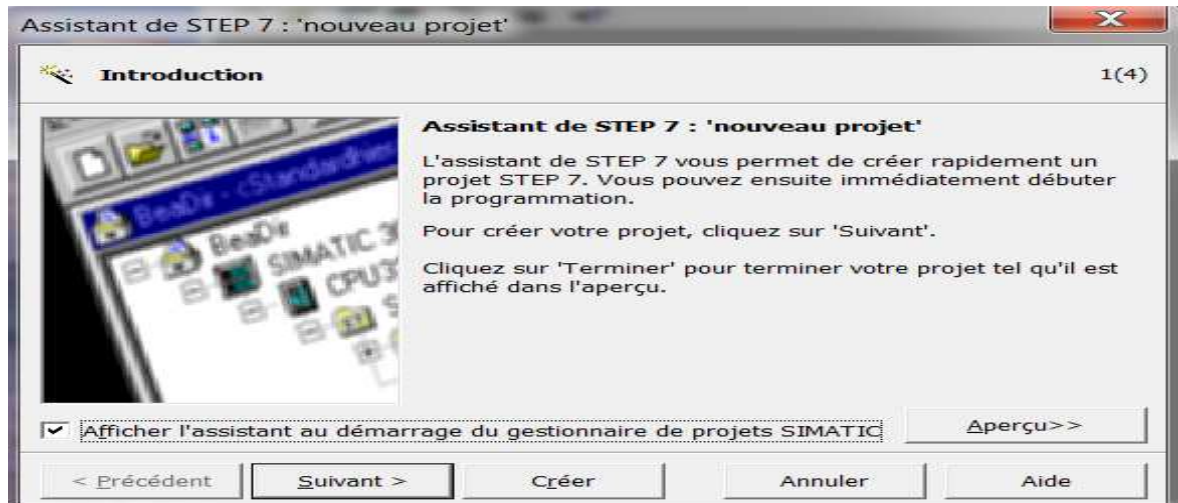
La création du projet se fait comme suit :

**a)- Lancement du logiciel :**

Double cliquer sur l'icône SIMATIC Manager sur le bureau Windows ; ceci lance l'assistant de STEP7.

**b)- Création du projet :**

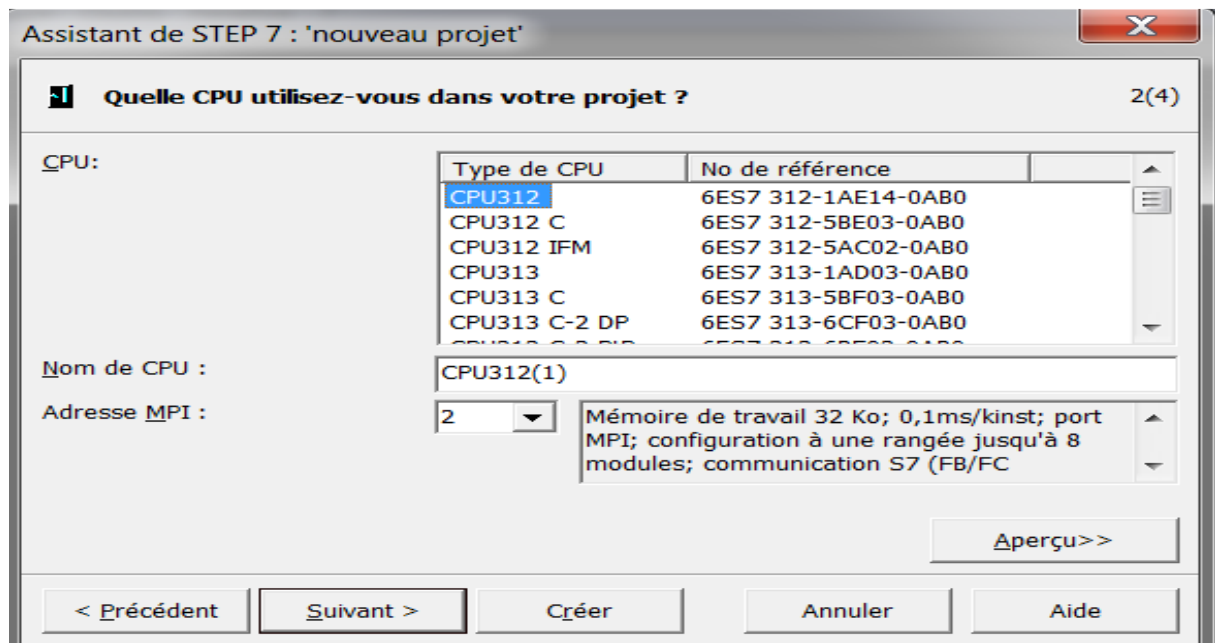
L'assistant étant lancé, une première fenêtre s'ouvre à l'écran. C'est la fenêtre d'introduction de l'assistant. (**Figure III.17**)



**Figure III.17 :** Assistant de STEP 7 « nouveau projet ».

**c)- Choix de la CPU :**

En cliquant sur l'icône « suivant », la fenêtre suivante apparaît (**figure III.18**), elle nous permet de choisir la CPU .Dans notre cas nous avons choisi la CPU 312.



**Figure III.18 :** Fenêtre de choix de la CPU.

### d)- Choix des blocs :

Après validation de la CPU, une autre fenêtre apparaît (**Figure III.19**), elle permet de choisir les blocs à insérer, et choisir le langage de programmation (LIST, CONT, LOG). Pour notre projet nous avons choisi l'OB1 (cycle d'exécution) et le langage à contact.

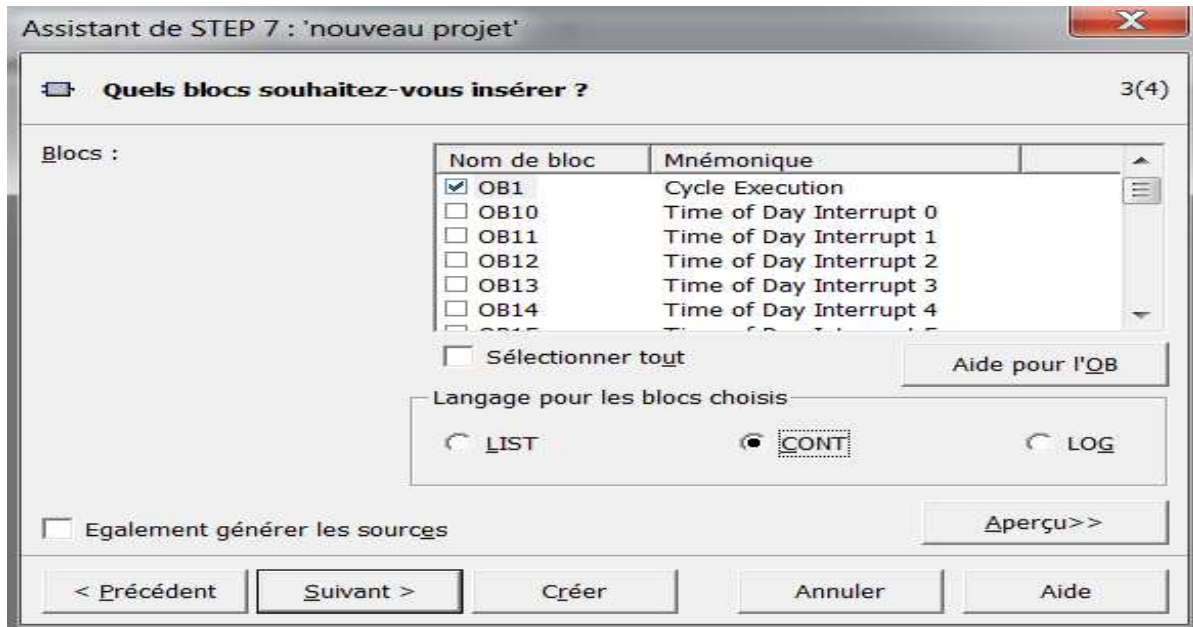


Figure III.19: Choix des blocs à utiliser et de langage.

### e)- Nomination du projet créé :

En cliquant sur suivant, la fenêtre (**figure III.20**) apparaît pour permettre de nommer le projet.

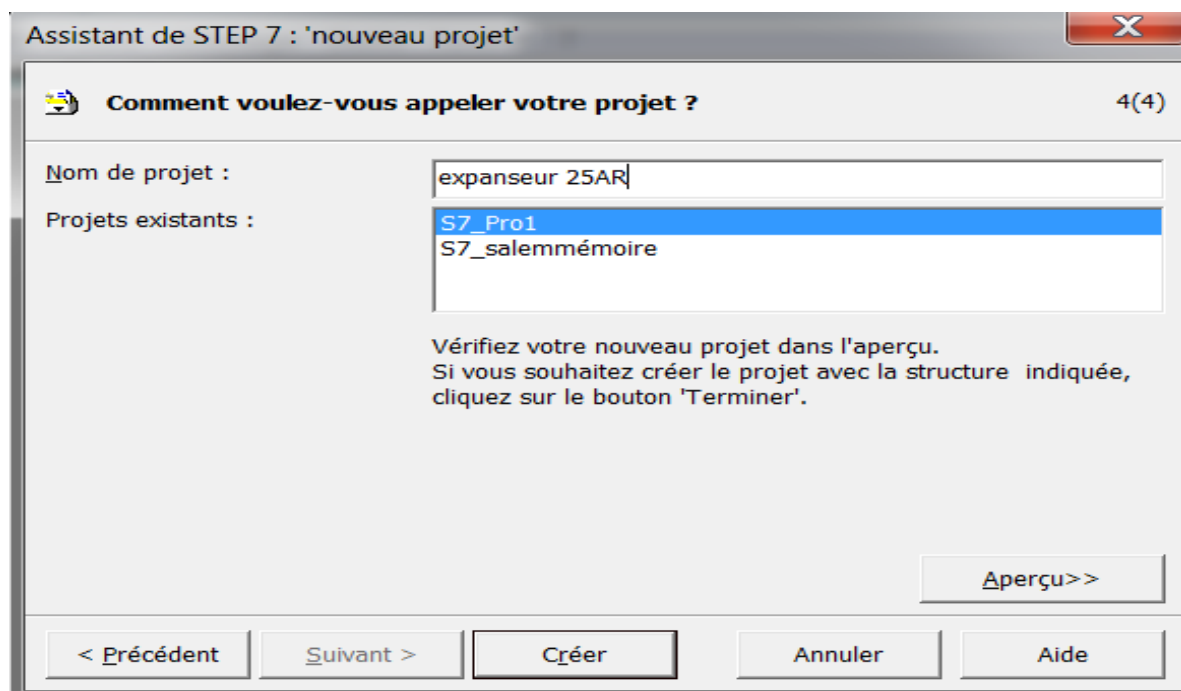
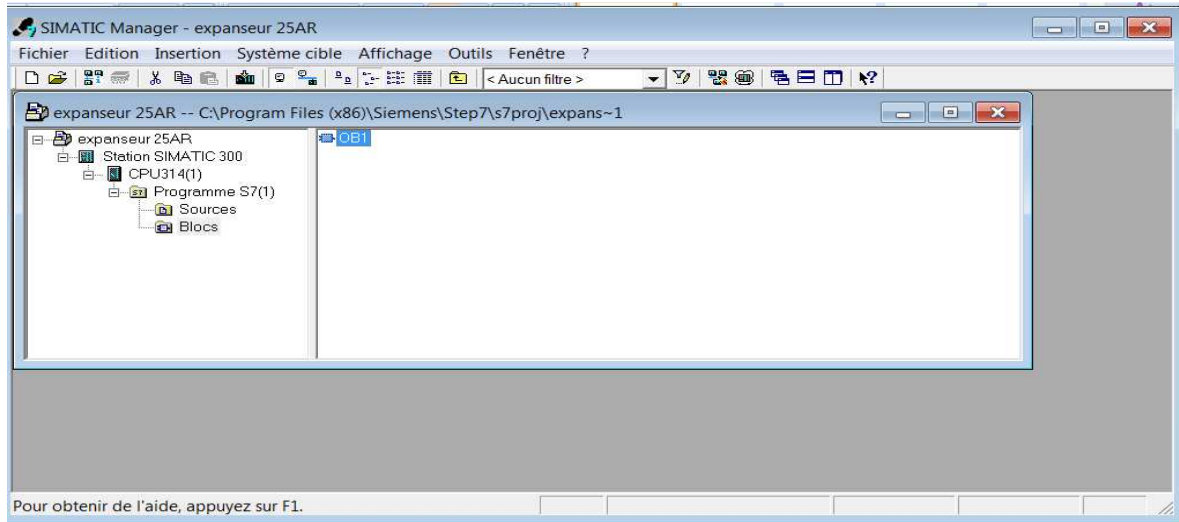


Figure III.20: Nomination du projet.

**f)- Finalisation de la création du projet :**

En cliquant sur créer, la fenêtre suivante apparaît (Figure III.21) :



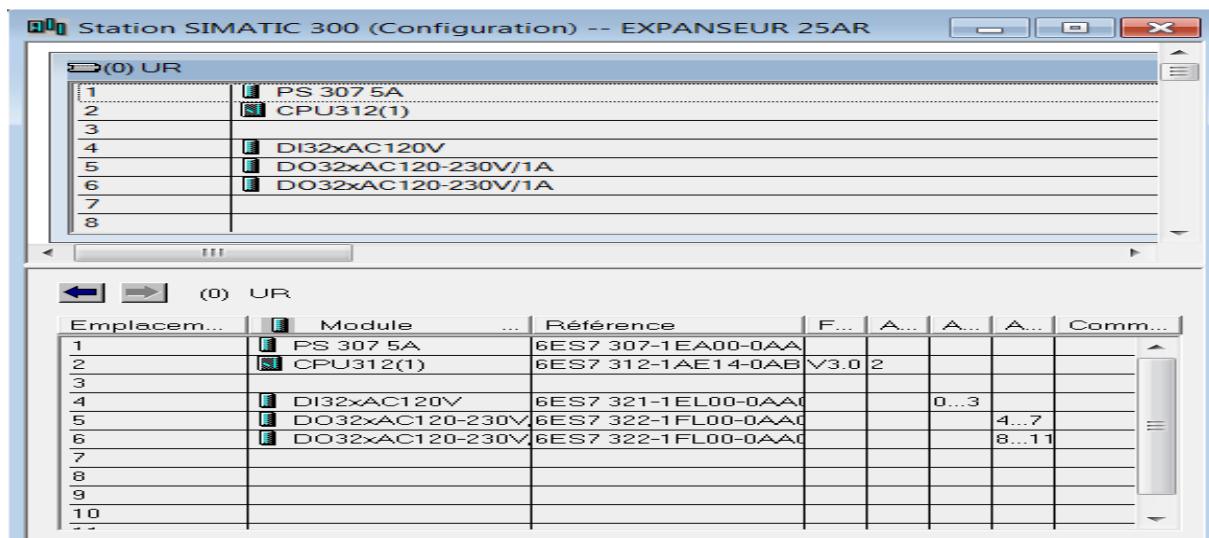
**Figure III.21:** Vue des composants de notre projet S7.

**8.3.3. Configuration matérielle de l'automate :**

La configuration matérielle consiste en la disposition de profilés support ou châssis (racks), de modules et d'appareils de la périphérie centralisée. Les châssis sont représentés par une table de configuration dans laquelle on peut enficher un nombre défini de module, comme dans les châssis réels.

STEP 7 affecte automatiquement une adresse à chaque module dans la table de configuration. Pour notre système, nous avons choisi une configuration dans laquelle nous avons (**Figure III.22**) :

- Un rack.
- Un module d'alimentation PS 307 5A.
- Une CPU 312.
- Un (01) modules d'entrées logiques de 32 bits (DI32xAC 120V).
- Deux (02) modules de sortie logique de 32 bits (DO32xAC 120-230V/1A).



**Figure III.22 :** Configuration matérielle de notre automate.

On sauvegarde cette configuration en cliquant sur l'item Enregistrer du menu déroulant Fichier.

### 8.3.4. Création de la table des mnémoniques :

Les mnémoniques permettent de rendre le programme utilisateur très lisible en déclarant les différentes entrées/sorties de la machine ainsi que les mementos utilisés afin de mieux les distinguer et faciliter la simulation du programme.

L'objet « mnémonique » (table des mnémoniques) est automatiquement créé sous un programme S7.

Pour pouvoir remplacer les données globales par des mnémoniques dans un programme, nous devons les affecter dans la table des mnémoniques.

On procède de la manière suivante :

1. Nous cliquons deux fois sur le programme S7 dans la fenêtre du projet afin que l'objet « **mnémoniques** » s'affiche dans la partie droite de la fenêtre.
2. Si la table des mnémoniques a été effacée ou doit être écrasée, nous pouvons insérer une nouvelle en choisissant la commande **insertion>Table des mnémoniques**.
3. Nous ouvrons l'objet « **mnémoniques** », par exemple nous cliquons deux fois dessus. Dans la fenêtre qui s'ouvre, nous pouvons éditer la table des mnémoniques.

### 8.3.5. Adressage des signaux d'entrées/sorties :

La déclaration d'une entrée ou sortie donnée à l'intérieur du programme s'appelle « **adressage** ».

Les entrées/sorties des automates sont regroupées en groupe de huit entrées ou sorties TOR. Cette unité de huit entrées ou sorties est appelée un octet. Chaque groupe reçoit un numéro que l'on appelle l'adresse octet.

Afin de permettre l'adressage d'une entrée ou sortie à l'intérieur d'un octet, il est divisé en huit bits. Ces derniers sont numérotés de 0 à 7. Nous obtenons ainsi l'adresse du bit. (**figure III.23**)

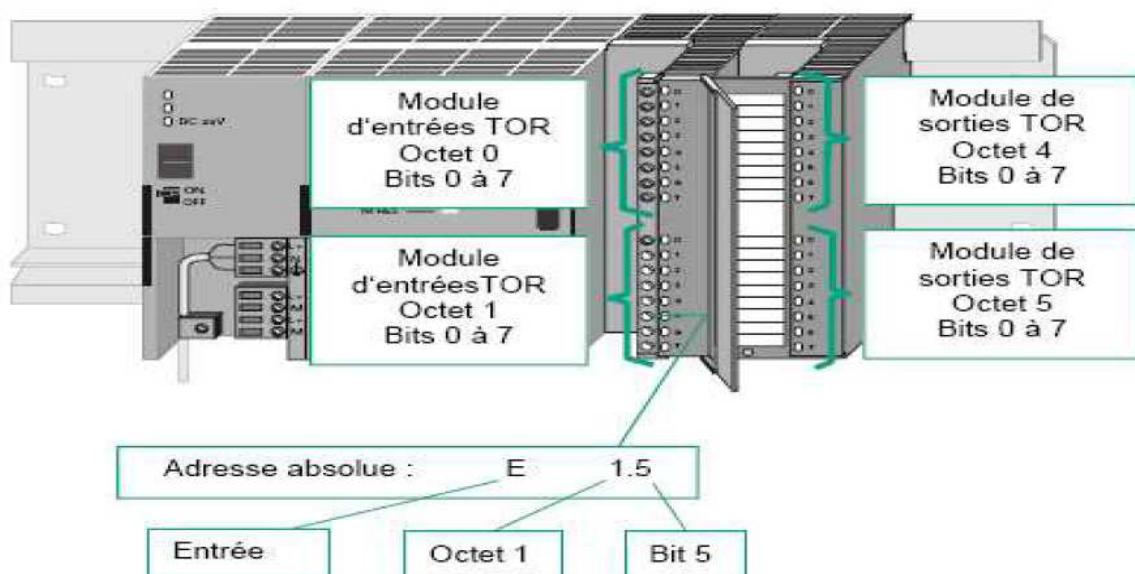


Figure III.23 : Adressage d'entrées/sorties (TOR).

### **8.3.6. Les mémentos :**

Les mémentos sont utilisés pour les opérations internes à l'automate pour lesquelles l'émission d'un signal n'est pas nécessaire. Chaque automate programmable dispose d'une grande quantité de mémentos (S7-315 dispose de 2048 bits de mémentos).

### **8.3.7. Traitement du programme par l'automate :**

Le traitement du programme dans l'automate est cyclique, il se déroule comme suit :

- **Phase 1 :** le système d'exploitation démarre la surveillance de temps de cycle.
- **Phase 2 :** la CPU lit l'état des entrées dans les modules d'entrées et met à jour la mémoire image des entrées.
- **Phase 3 :** la CPU exécute les instructions du programme utilisateur et écrit les résultats dans la mémoire image des sorties, ou elle les transfère vers les modules de sorties.
- **Phase 4 :** à la fin du cycle, le système d'exploitation exécute les travaux en attente, tels que le chargement et l'effacement des blocs ou la réception et l'émission des données globales.
- **Phase 5 :** la CPU revient alors au début du cycle et démarre à nouveau la surveillance du temps du cycle.

#### **8.3.7.1. Principe de conception d'une structure de programme :**

Au cours de l'exécution du programme dans la CPU, deux programmes différents s'exécutent, le système d'exploitation et le programme utilisateur.

##### **a)- Système d'exploitation :**

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

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

##### **b)- Programme utilisateur :**

Après avoir créé le programme utilisateur (contient toutes les fonctions nécessaires au traitement de la tâche d'automatisation spécifique) et le charger dans la CPU. Il doit :

- Déterminer les conditions pour le démarrage et le redémarrage de la CPU (par exemple, initialiser des signaux).

- Traiter des données de processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques, définir des signaux binaires pour la sortie, écrire des valeurs analogiques).

- Traiter les perturbations dans le déroulement normal du programme.

Un programme utilisateur devra être exécuté dans une CPU S7 et est essentiellement constitué de blocs.

### **8.3.7.2. Blocs dans le programme utilisateur :**

Le logiciel de programmation STEP 7 nous permet de subdiviser le procédé à automatiser en différentes parties autonomes. Il en résulte les avantages suivants :

- Ecriture de programmes importants et clairs.
- Simplification de l'organisation du programme.
- Modification facile du programme.
- Simplification du test du programme, car nous pouvons l'exécuter section par section.
- Facilité de la mise en service.

#### **a)- Bloc utilisateur :**

Le STEP 7 offre les blocs utilisateur suivants pour la programmation structurée :

- **Bloc d'organisation (OB) :** Un OB est appelé cycliquement par le système d'exploitation et constitue donc l'interface entre le programme utilisateur et le système d'exploitation. L'OB contient des instructions d'appels de bloc indiquant à l'unité de commande de l'automate l'ordre dans lequel il doit traiter les blocs.
- **Bloc fonctionnel (FB) :** Le FB dispose d'une zone de mémoire qui lui est affectée en propre. Il est possible d'affecter un bloc de données (DB) au FB à l'appel du bloc. Il est possible d'accéder aux données du DB d'instance via les appels contenus dans le FB. Nous pouvons affecter plusieurs DB à un FB. Il est possible d'appeler d'autres FB et FC dans un bloc fonctionnel via des instructions d'appels de bloc.
- **Fonction (FC) :** Une FC ne possède pas une zone de mémoire propre. Les données locales d'une fonction sont perdues après l'exécution de la fonction. Il est également possible d'appeler d'autres FB et FC dans une fonction via des instructions d'appels de blocs.
- **Blocs de données (DB) :** les DB sont utilisés pour la mise à disposition d'espace mémoire pour les variables types de données. Il existe deux types de blocs de données, les DB globaux dans lesquels tous les OB, FB et FC peuvent lire les données enregistrées ou écrire des données, et les DB d'instance qui sont affectés à un FB donné.

#### **b)- Blocs système pour fonctions standard et fonctions système :**

Les blocs système sont des fonctions prêtes à l'emploi stockées dans la CPU. Ces blocs peuvent être appelés par l'utilisateur et utilisés dans le programme.

**b.1)- Bloc fonctionnel système (SFB) :** c'est un bloc fonctionnel stocké dans le système d'exploitation de la CPU et pouvant être appelé par l'utilisateur.

**b.2)- Fonction système (SFC) :** c'est une fonction stockée dans le système d'exploitation de la CPU et pouvant être appelée par l'utilisateur.

**b.3)- Données système (SDB) :** c'est une zone de mémoire dans le programme configurée par différentes applications du STEP 7 pour le stockage des données dans le système d'automatisation.

### 8.3.7.3. Traitement du programme par la CPU :

Deux traitements du programme sont possibles en fonction de l'unité de commande et de la programmation. Le traitement linéaire et le traitement structuré (**figure III.24**).

#### a)- Traitement linéaire du programme :

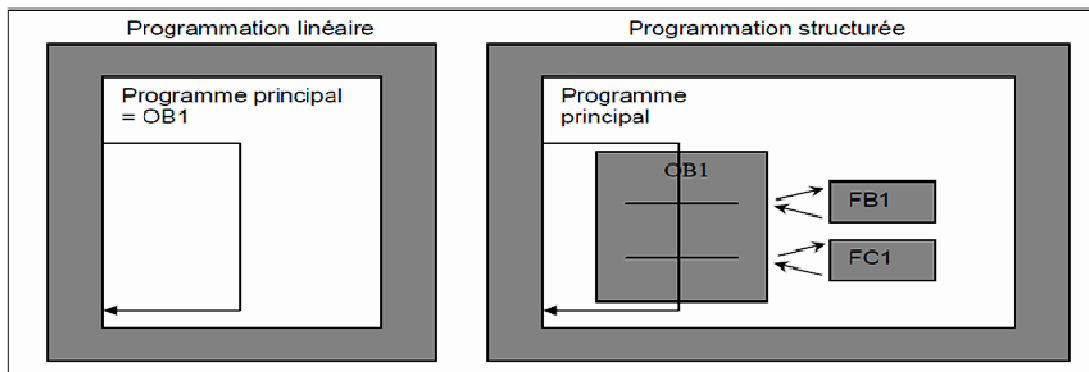
Les instructions sont traitées par l'unité de commande dans l'ordre dans lequel elles sont stockées dans la mémoire du programme. Lorsque la fin du programme (BE) est atteinte, le programme reprend depuis le début. On parle de traitement cyclique.

Le traitement linéaire du programme est généralement utilisé pour des commandes simples, de volume peu important et peut être réalisé au moyen d'un seul bloc d'organisation (OB).

#### b)- Traitement structuré du programme :

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

Dans ce cas la CPU exécute le programme principal dans OB1 et fait appel là, et quand il le faut aux autres blocs (FC, FB). A la fin de l'exécution du bloc appelé, la CPU revient pour poursuivre le programme appelant.



**Figure III.24 :** Schéma illustrant le traitement du programme par la CPU.

### 8.3.7.4. Les différents blocs d'organisation des S7-300 :

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation de la CPU et le programme utilisateur. Ils nous permettent de déclencher l'exécution de certaines parties du programme telles que la mise en route de la CPU ou bien quand une erreur intervient.

Les différents blocs d'organisation sont:

#### a)- Le bloc d'organisation (OB1) :

Le système d'exploitation de la CPU S7 exécute l'OB1 de manière cyclique quand la mise en route de la CPU est terminée.

### b)- Le bloc d'organisation pour le traitement des erreurs :

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

- **Bloc d'organisation de programme déclenché par alarme :**

Le STEP 7 met à la disposition du programmeur différents types d'OB qui interrompent le traitement de l'OB1, à des intervalles de temps précis ou à des apparitions d'événement donnés.

- **Blocs d'organisation pour le programme de mise en route :**

Lors de la mise en route, le système d'exploitation appelle l'OB de mise en route correspondant, à savoir :

- L'OB de démarrage (OB100).
- L'OB de redémarrage (OB101). [14]

### 9. Sécurité :

Les systèmes automatisés sont, par nature, source de nombreux dangers (tensions utilisées, déplacements mécaniques, jets de matière sous pression ...).

Placé au cœur du système automatisé, l'automate se doit d'être un élément fiable car :

- un dysfonctionnement de celui-ci pourrait avoir de graves répercussions sur la sécurité des personnes.
- les coûts de réparation de l'outil de production sont généralement très élevés,
- un arrêt de la production peut avoir de lourdes conséquences sur le plan financier.

Aussi, l'automate fait l'objet de nombreuses dispositions pour assurer la sécurité :

- **Contraintes extérieures :** l'automate est conçu pour supporter les différentes contraintes du monde industriel et à fait l'objet de nombreux tests normalisés (tenue aux vibrations, CEM ...).
- **Coupures d'alimentation :** l'automate est conçu pour supporter les coupures d'alimentation et permet, par programme, d'assurer un fonctionnement correct lors de la réalimentation (reprises à froid ou à chaud).
- **Mode RUN/STOP :** Seul un technicien peut mettre en marche ou arrêter un automate et la remise en marche se fait par une procédure d'initialisation (programmée).
- **Contrôles cycliques :**
  - Procédures d'autocontrôle des mémoires, de l'horloge, de la batterie, de la tension. d'alimentation et des entrées / sorties.
  - Vérification du temps de scrutation à chaque cycle appelée **Watchdog** (chien de garde), et enclenchement d'une procédure d'alarme en cas de dépassement de celui-ci (réglé par l'utilisateur).
- **Visualisation :** Les automates offrent un écran de visualisation où l'on peut voir l'évolution des entrées / sorties.

## **Chapitre III : Les automates programmables industriels (API)**

La défaillance d'un automate programmable pouvant avoir de graves répercussions en matière de sécurité, les normes interdisent la gestion des arrêts d'urgence par l'automate ; celle-ci doit être réalisée en technologie câblée.

On peut également ajouter des modules de sécurité à l'automate (sécurité des machines).

Il existe enfin des automates dits de sécurité (APIsS) qui intègrent des fonctions de surveillance et de redondance accrues et garantissent la sécurité des matériels. [9]

### **10. Conclusion :**

Dans ce chapitre nous avons défini l'API de manière générale et ses différents composants ainsi que ses langages de programmation. Puis nous nous sommes focalisés sur l'API S7-300 et nous avons indiqué les étapes à suivre pour créer un projet sous STEP 7 (qui est le logiciel de programmation de notre automate) et faire une configuration matérielle de notre automate. Nous verrons par la suite l'insertion de notre programme sous STEP 7 ainsi que sa simulation sous PLC SIM.

# *Chapitre IV.*

## **Insertion du programme de commande et Simulation sous S7-PLCSIM**

## 1. Introduction :

Après la copie du programme de commande de notre système vers le nouvel automate (S7-300), sous STEP-7, nous arrivons à l'étape décisive du travail effectué. Cette étape est la validation du programme par simulation et vérification de son bon fonctionnement.

Pour cela nous avons utilisé le logiciel S7-PLCSIM qui est un logiciel optionnel de STEP7. En effet, il permet d'exécuter et de tester le programme utilisateur dans un système d'automatisation simulé sur un ordinateur ou une console de programmation. Cette application dispose d'une interface permettant de surveiller et de modifier le programme.

## 2. Logiciel de simulation S7-PLCSIM :

S7-PLCSIM est un logiciel optionnel de STEP 7. Son utilisation suppose que la version de base de STEP 7 est déjà installée.

Le domaine d'application du logiciel S7-PLCSIM est le test des programmes STEP 7 pour les automates S7-300 et S7-400 que l'on ne peut pas, tester immédiatement sur le matériel, et ceci pour différentes raisons :

- Petits blocs de programme qui ne peuvent pas encore être testés dans une séquence unique sur la machine.
- L'application est critique, car elle peut occasionner des dommages matériels ou blessures corporelles en cas d'erreurs de programmations. La simulation permet de supprimer ces erreurs dès la phase de test, et dans le cas où nous ne disposons pas d'un automate.

### 2.1. Mise en route du logiciel S7-PLCSIM :

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

- On ouvre le 'SIMATIC Manager' par un double clic sur son icône.



SIMATIC Manager






- On clique sur l'icône , ou on sélectionne la commande Outils-simulation de modules. Cela lance l'application S7-PLCSIM et ouvre une fenêtre CPU (Figure IV.1):



Figure IV .1 : fenêtre du S7-PLCSIM.

- Dans le gestionnaire de projets SIMATIC, chercher le sur le quel on travaille. Dans ce dernier, chercher le dossier blocs.
- Dans le gestionnaire de projet SIMATIC, on clique sur l'icone  ou, on choisit la commande **Système cyble-Charger** pour charger le dossier blocs dans l'API de simulation.
- Dans l'application S7-PLCSIM, on crée de nouvelles fenêtres pour visualiser les informations provenant de l'API de simulation :
- On clique sur l'icone  ou, on choisit la commande **Insertion-Entrée** pour créer une fenêtre dans l'laquelle on peut visualiser et forcer des variables dans la zone de mémoire des entrées (zone E). Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut EB0, mais on peut modifier l'adresse (EB1, EB2...).
- On clique sur l'icone  ou, on choisit la commande **Insertion-Sortie** pour créer une fenêtre dans laquelle on peut visualiser et forcer des variables dans la zone de mémoire des sorties (zone A). Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut Ab0, Mais on peut modifier l'adresse (AB1, AB2...).
- On clique sur l'icone  ou, on choisit la commande **Insertion-Temporisation** pour créer une fenêtre dans laquelle on peut visualiser et forcer les temporisations utilisées par le programme. Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut T0.
- Choisir le menu **CPU** dans la fenêtre du S7\_PLCSIM et vérifier que la commande **mettre sous tension** est activée (figure IV.2).

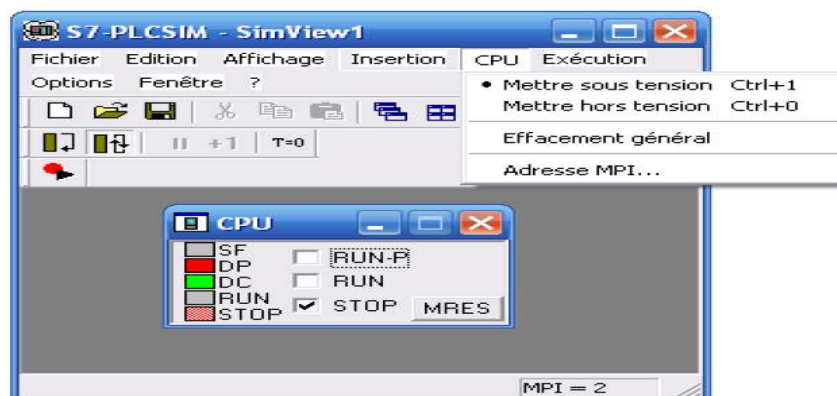


Figure IV.2 : Mise sous tension de la CPU.

- Choisir la commande **Exécution-Mode d'exécution** et vérifier que la commande **cycle continu** est activée (figure IV.3).

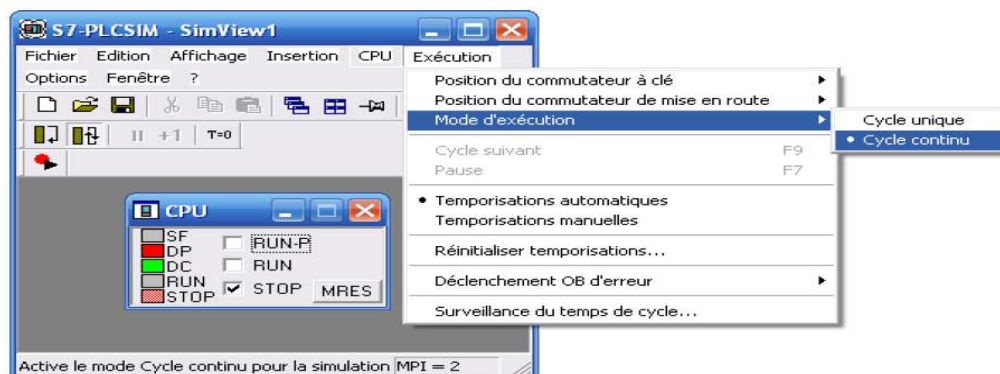


Figure IV.3 : Choix du cycle continu.

- Mettre la CPU de simulation en marche en cliquant sur l'une des cases à cocher **RUN** ou **RUN-P** (figure IV.4).




Figure IV.4 : Mise en marche de la CPU.

- Une fois toutes les fenêtres d'entrées et de sorties sont prêtes, nous activons les entrées voulues pour lire l'état des sorties (Dans notre cas le programme est réalisé auparavant, on utilise juste des mementos pour simuler le petit programme créé permettant de voir le fonctionnement de la machine sous winCC qu'on verra au chapitre suivant ), (Figure IV.5).



Figure IV.5 : simulateur S7-PLCSIM.

### 3. Visualisation de l'état du programme :

Après le chargement du programme dans la CPU du simulateur et la mise de cette dernière en mode « RUN », le STEP7 nous permet de visualiser l'état du programme soit en cliquant sur l'icône , ou en sélectionnant la commande **Test-Visualiser**. [14]

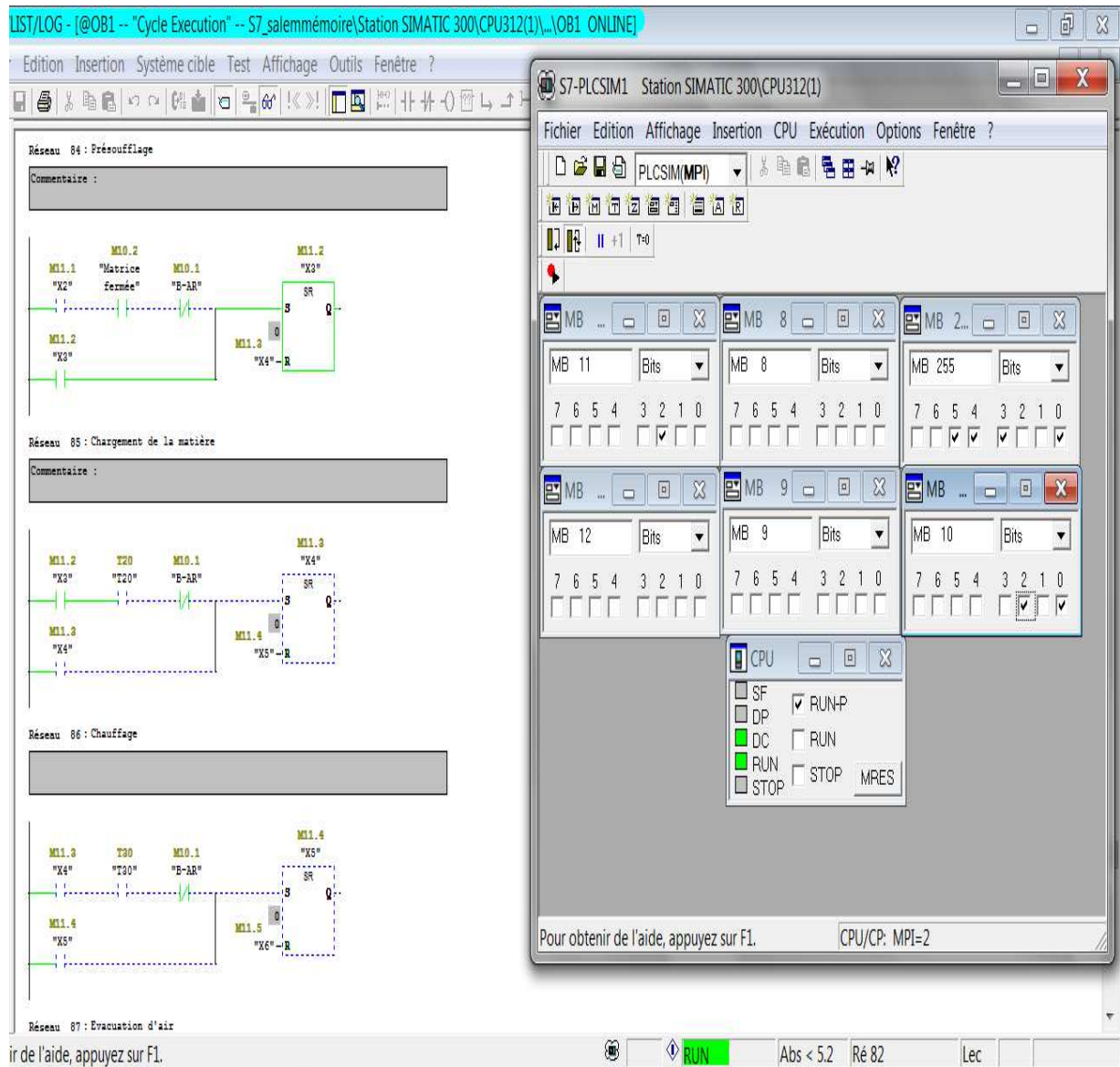


Figure IV.6 : Visualisation d'une partie du programme créé.

#### 4. Conclusion :

Le logiciel S7-PLC SIM est très précieux à l'automaticien, il permet de tester n'importe quel programme mis en œuvre sous step 7 avant son essai en pratique et éviter les risques qui peuvent être courus par un essai direct. Il nous permet d'être averti des erreurs commises et qui peuvent endommager le matériel ou mettre à risque la vie des opérateurs en nous indiquant qu'il y a une erreur dans notre travail.

Un programme simulé sans erreur sous S7-PLC SIM nous assure un bon fonctionnement à l'état réel.

# Chapitre V :

*Supervision de l'EXPANSEUR ACE 25 AR sous WinCC Flexible 2008*

## **1. Introduction :**

Avec le développement de l'informatique, il est devenu possible de traiter des données dans le domaine industriel, grâce à des vues préalablement créées et configurées, et à l'aide d'un logiciel adéquat. Le logiciel WinCC est une entité capable de présenter à l'opérateur des informations utiles, afin qu'il prenne à temps les bonnes décisions pour la conduite du procédé. Il a, essentiellement, pour mission de collecter les données (acquisition et stockage) et les mettre en forme (traitement), afin de les présenter à l'opérateur (supervision). [16]

## **2. Généralités sur la supervision :**

### **2.1. Définition de la supervision :**

La supervision est une forme évoluée du dialogue Homme-Machine, elle présente beaucoup d'avantages pour les processus industriels de production. Elle facilite à l'opérateur la surveillance de l'état du fonctionnement d'un procédé ainsi que son contrôle-commande. Elle permet grâce à des vues préalablement créées et configurées 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.

Lorsque la complexité des processus augmente et que les machines et installations doivent répondre à des spécifications de fonctionnalités toujours plus sévères, l'opérateur a besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'Interface Homme-Machine (IHM). [17]

### **2.2. Constitution d'un système de supervision :**

La majorité des systèmes de supervision se compose, généralement, d'un moteur central (logique) 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 (Figure V.1).

#### **2.2.1. Module de visualisation :**

Il permet d'obtenir et de mettre à la disposition des opérateurs des éléments d'évaluation du procédé par ses volumes de données instantanées.

#### **2.2.2. Module d'archivage :**

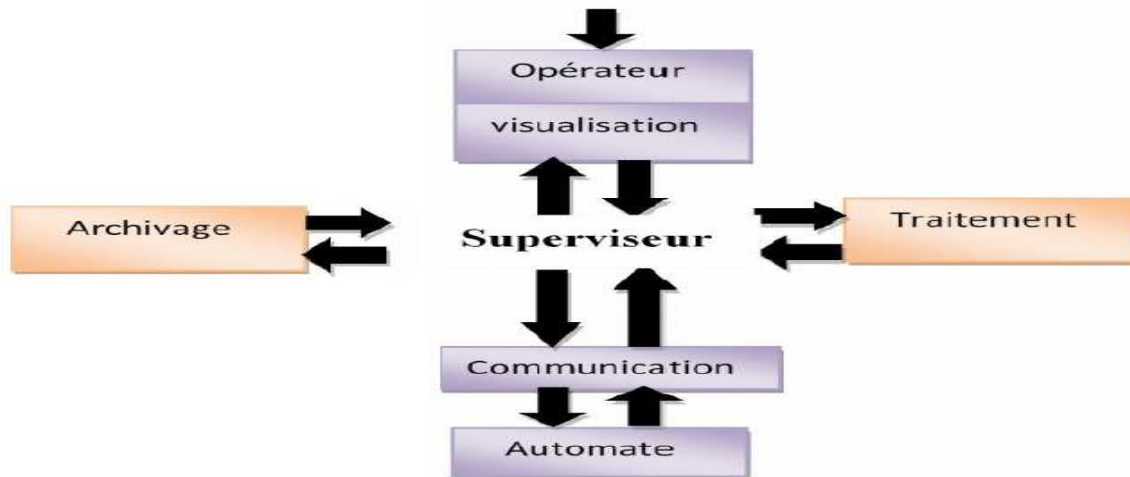
Il mémorise des données (alarmes et événements) pendant une longue période, et permet l'exploitation des données pour des applications spécifiques à des fins de maintenance ou de gestion de production.

#### **2.2.3. 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.

#### **2.2.4. Module de communication :**

Le module de communication assure l'acquisition et le transfert des données et gère la communication avec les automates programmables industriels et autres périphériques.



**Figure V.1** : schéma synoptique d'un système de supervision. [16]

### **2.3. Avantage de la supervision :**

Un système de supervision donne de l'aide à l'opérateur dans la conduite des procédés industriels, son but est de présenter à l'opérateur des résultats expliqués et interprétés et son avantage principal est :

- Surveiller les procédés industriels à distance.
- La détection des défauts.
- Le diagnostic et le traitement des alarmes.

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. Il existe par conséquent une interface entre l'opérateur et WinCC flexible (sur le pupitre opérateur) et une interface entre WinCC flexible et le système d'automatisation

Un système IHM se charge des tâches suivantes :

#### **a) Représentation du processus :**

Le processus est représenté sur le pupitre opérateur. Lorsqu'un état du processus évolue par exemple, l'affichage du pupitre opérateur est mis à jour.

#### **b) Commande du processus :**

L'opérateur peut commander le processus via l'interface utilisateur graphique. Il peut par exemple définir une valeur de consignes pour un automate ou démarrer un moteur.

#### **c) Vue des alarmes :**

Lorsque surviennent des états critiques dans le processus, une alarme est immédiatement déclenchée, par exemple, lorsqu'une valeur limite est franchie.

#### **d) Archivage des valeurs processus et d'alarmes :**

Les alarmes et les valeurs processus peuvent être archivées par le système IHM. On peut ainsi documenter la marche du processus et accéder ultérieurement aux données de la production écoulée.

**e) Documentation des valeurs processus et des alarmes :**

Les alarmes et les valeurs processus peuvent être éditées par le système IHM sous forme de journal. On peut ainsi consulter les données de production à la fin du travail d'une équipe.

**f) Gestion des paramètres processus et des machines :**

Les paramètres du processus et des machines peuvent être enregistrés au sein du système IHM dans des recettes. Ces paramètres sont alors transférables en une seule opération sur l'automate pour démarrer la production d'une variante du produit par exemple.

**3. Présentation du logiciel de supervision WinCC FLEXIBLE 2008:**

WinCC (**Windows Control Center**) flexible 2008 est un système IHM (**I**nterface-**H**omme-**M**achine) très performant développé par SIEMENS. C'est un outil flexible qui s'intègre parfaitement dans les solutions d'automatisation et de techniques de l'information et qui est destiné à la configuration des systèmes de supervision.

WinCC flexible permet la saisie, l'affichage et l'archivage des données tout en facilitant les tâches de conduite et de surveillance aux exploitants. Il est compatible avec Windows et comporte des objets graphiques prédéfinis tels que : L'affichage numérique, la bibliothèque complète de symboles IHM, l'affichage de textes et courbes, les champs d'édition de valeurs du processus,...etc.

**3.1. Avantages de WinCC flexible 2008 :**

- WinCC permet de visualiser le process et de concevoir l'interface utilisateur graphique destinée à l'opérateur.
- WinCC permet à l'opérateur de surveiller le processus. Pour ce faire, le process est visualisé par un graphisme à l'écran. Dès qu'un état du process évolue, l'affichage est mis à jour.
- WinCC permet à l'opérateur de commander le process. A partir de l'interface utilisateur graphique, il peut par exemple entrer une valeur de consigne ou ouvrir une vanne.
- Lorsqu'un état du process devient critique, une alarme est déclenchée automatiquement. L'écran affiche une alarme en cas de franchissement d'un seuil défini.
- Les alarmes et les valeurs du process peuvent être imprimées et archivées sur support électronique par WinCC. Ceci permet de documenter la marche du process et d'avoir accès ultérieurement aux données de production du passé.
- Les interfaces de programmation ouvertes de WinCC permettent d'intégrer différents programmes pour piloter le process ou exploiter des données.
- On peut adapter WinCC de façon optimale aux exigences de notre process. Le système supporte de nombreuses configurations. La gamme des configurations s'étend du système monoposte aux systèmes répartis à plusieurs serveurs en passant par les systèmes client-serveur.
- La configuration WinCC peut être modifiée à tout moment même après mise en service, les projets existants n'en sont pas affectés.

- WinCC est un système IHM compatible avec le réseau Internet qui permet de réaliser des solutions basées sur le web (contrôle-commande à distance).

### **3.2. WinCC et SIMATIC STEP 7:**

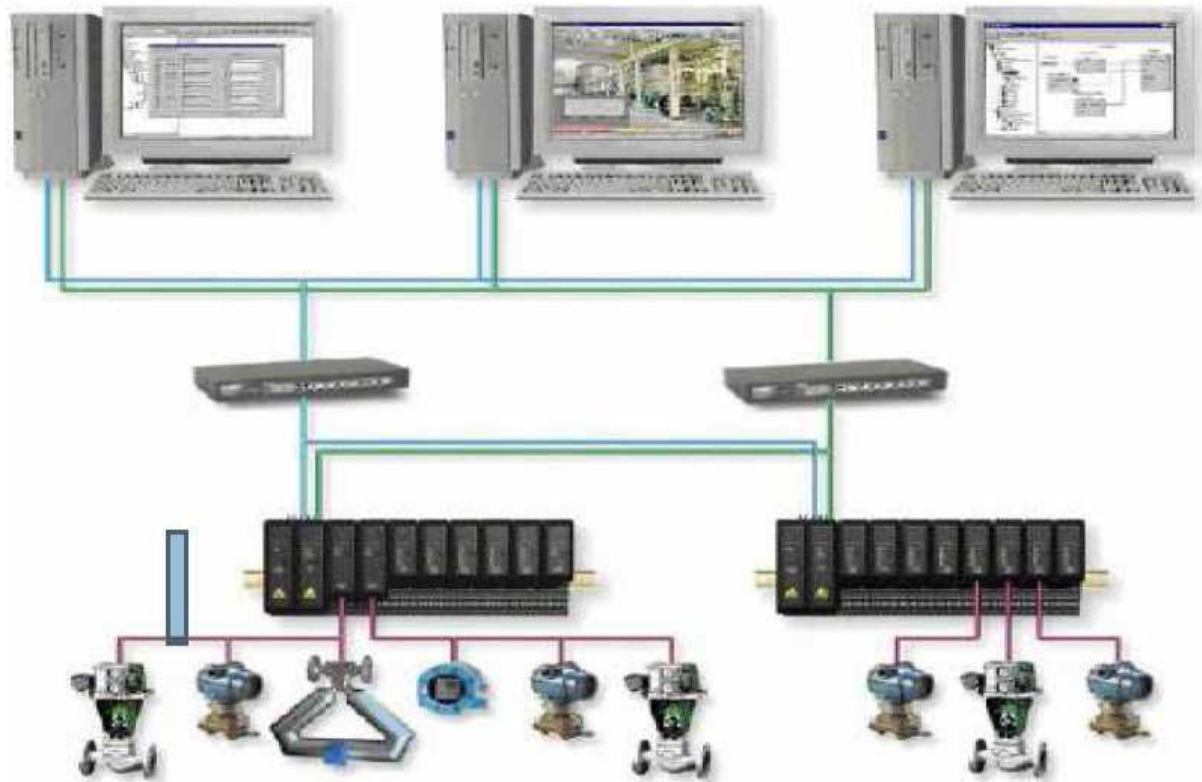
Faisant partie du concept TIA de Siemens (Totally Integrated Automation), WinCC s'avère particulièrement efficace dans le cadre d'une mise en œuvre avec des automates programmables de la famille du produit SIMATIC. Les automates programmables d'autres marques sont bien entendus également pris en charge.

WinCC s'intègre parfaitement au logiciel SIMATIC STEP7. Cela nous permet de choisir des mnémoniques et des blocs de données de SIMATIC STEP7 comme variable dans WinCC. On économise ainsi en temps et on évite aussi des sources d'erreurs dues à la répétition de la saisie.

### **3.3. Communication entre le PC de supervision et l'automate:**

La communication entre le PC de supervision et la machine ou le processus est réalisée par l'intermédiaire de l'automate, au moyen de « variables ». La valeur d'une variable est écrite dans une zone mémoire (adresse) de l'automate où elle est lue par le PC de supervision.

La structure générale est illustrée dans la figure suivante :



**Figure V.2 :** Structure générale de communication entre le PC de supervision et l'API.[17]

## **4. Développement d'un système de supervision sous WinCC Flexible 2008:**

### **4.1. Réalisation des vues de supervision de l'EXPANSEUR ACE 25 AR:**

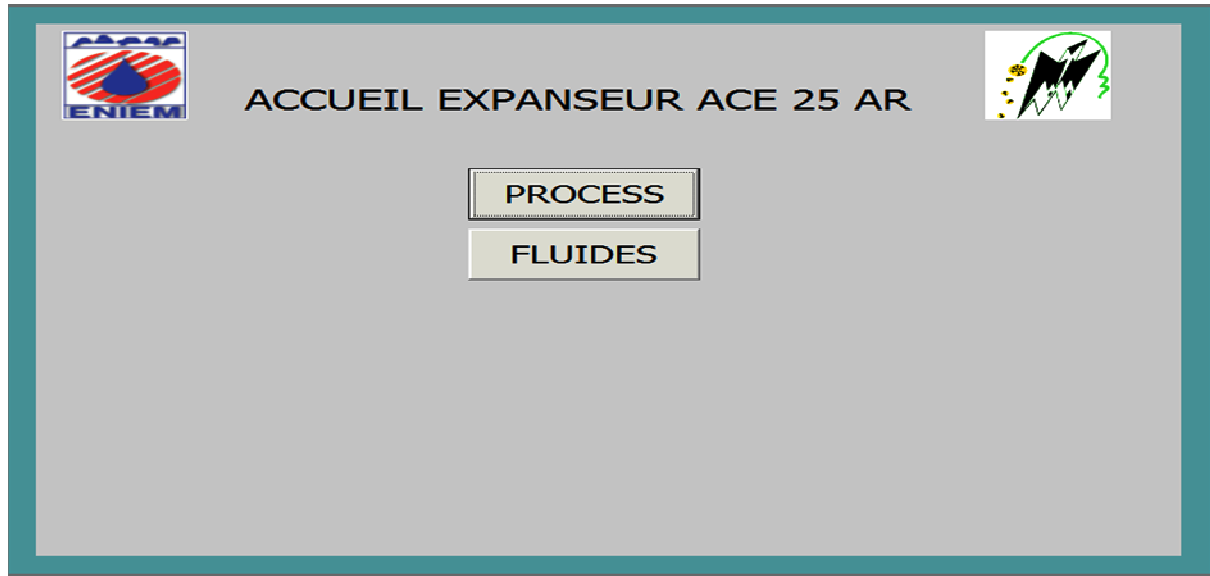
L'EXPANSEUR ACE 25 AR se compose d'une partie mécanique et d'une partie hydraulique.

On a développé 3 vues pour cette machine :

- Vue d'accueil.
- Vue du process.
- Vue du circuit des fluides.

- **Vue d'accueil :**

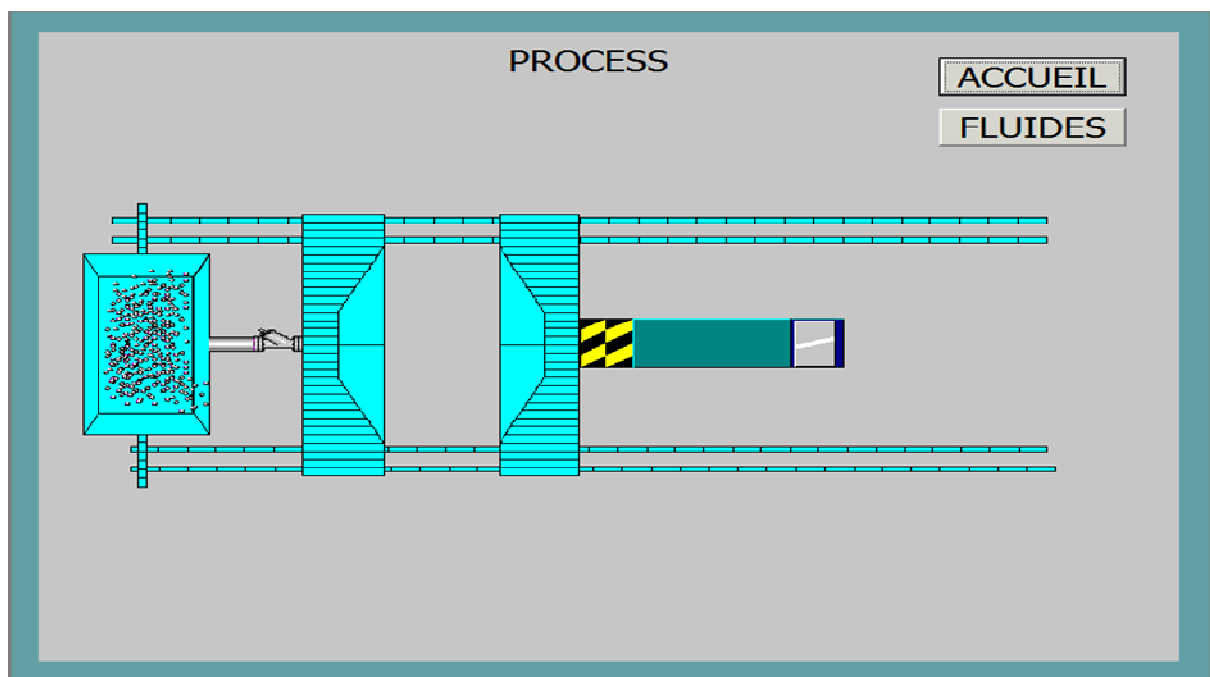
La vue d'accueil comporte les boutons de navigation qui permettent de basculer vers les autres vues.



**Figure V.3 :** Vue d'accueil de l'Expanseur ACE 25 AR.

- **Vue du Process:**

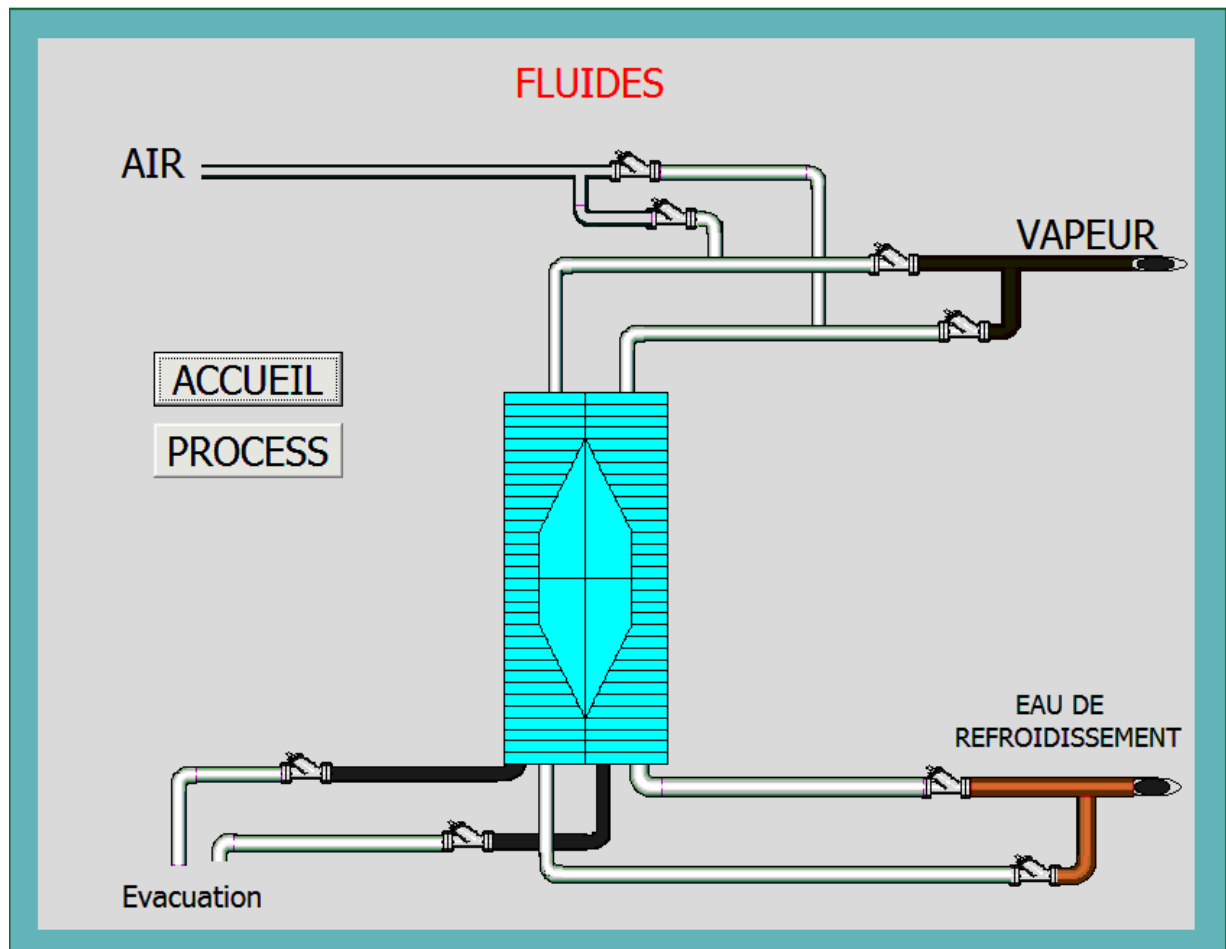
Sur cette vue, on pourra voir les étapes du moulage du polystyrène, en effet après fermeture complète de la matrice, il y aura chargement de la matière (grains de polystyrène) dans le moule. Après chauffage de ces derniers ils se fusionnent et prennent la forme du moule ou ils sont chargé. Après refroidissement à eau du produit puis par radiation pour que pression d'expansion baisse, on procède au démoulage du produit en injectant de l'air dans la matrice et en ouvrant cette dernière.



**Figure V.4 :** Vue du process de l'expanseur ACE 25 AR.

- **Vue du circuit des fluides :**

Sur cette vue se trouvent les trois fluides que sont l'air, l'eau de refroidissement et la vapeur qui seront sollicités à tour de rôle dans les phases de moulage et de démoulage.



**Figure V.4 :** Vue du circuit des fluides de l'expandeur ACE 25 AR.

### **5. Conclusion :**

Dans ce chapitre on a découvert l'outil de supervision winCC, qui est d'une grande utilité à l'automaticien. Avec ses riches outils, il permet d'avoir toute une usine dans un petit écran. Il permet de contrôler à distance de la commande.

J'ai essayé à travers les vues des fluides et du processus de montrer le fonctionnement de l'EXPANSEUR ACE 25 AR en grosso modo sans entrer dans les détails.

# Conclusion générale

## **Conclusion générale :**

La problématique abordée dans ce mémoire entre dans le cadre de l'automatisation des outils de production que connaissent actuellement les entreprises nationales. C'est donc une problématique d'actualité et d'importance économique majeure pour notre nation.

Dans ce cadre, j'ai effectué le stage de fin d'étude à l'ENIEM qui est l'une des plus importantes entreprises économiques nationales et qui est d'une importance particulière pour notre région puisqu'elle reste jusqu'à présent la plus importante entreprise en terme d'emploi et en terme de dynamique économique.

Comme toutes les entreprises nationales, l'ENIEM est confrontée au défi de rénover les équipements de production qui sont devenus vétuste afin de sauvegarder son outil de production.

Le problème qui m'a été proposé consiste à remplacer le séquenceur Toshiba qui pilote le fonctionnement de la machine de moulage par un automate Siemens S7-300. L'intérêt de ce remplacement est d'éviter les arrêts successifs causés par les anciens équipements qui arrivent à leur fin de vie. D'autre part, le remplacement du séquenceur Toshiba est une nécessité car il y a risque d'arrêt total en raison d'absence sur le marché de pièces de rechange en cas de pannes. La firme Toshiba ne fabrique plus ce type d'équipement. L'objectif qui m'a été assigné est de proposer donc une solution de rechange.

Avec l'aide du personnel technique de l'entreprise, j'ai pu mener à bien mon projet. Pendant le temps passé dans l'entreprise, en plus du travail spécifique lié à mon projet, j'ai aussi acquis quelques connaissances sur le plan technique et aussi sur le plan social et relationnel à la vie d'une entreprise industrielle et le monde du travail.

La conception proposée comporte la solution d'automatisation sous l'API S7-300 et le module de supervision réalisé sous WinCC. Le bon fonctionnement de cette conception a été illustré en simulation avec le programme que j'ai développé.

J'espère que le travail que j'ai réalisé sera concrétisé dans un futur proche par l'entreprise avec néanmoins des améliorations nécessaires à apporter. J'espère aussi que ce modeste travail servira de support sur le plan méthodologique et de suivi de projet aux futures promotions. Ce sont là quelques perspectives que je souhaite être développées.

# *Bibliographie*

### Bibliographie :

[1]: J. AKAICHI : Systèmes automatisés de production à intelligence distribuée. Thèse de Doctorat, Université des sciences et technologies de Lille, 1996.

[2]: Cours Contrôle et Procédés université de Marseille (POLYTECH MARSEILLE, AIX, Marseille Université) (téléchargé en 2016).

[3]: [www.bannaladi.fr/cours/structure1\\_SA.pdf](http://www.bannaladi.fr/cours/structure1_SA.pdf) (télécharger en 2016).

[4]: [http://philippe.berger2.free.fr/automatique/cours/cpt/les\\_capteurs.htm](http://philippe.berger2.free.fr/automatique/cours/cpt/les_capteurs.htm) (consulté en 2016).

[5]: L. Medjoudj, I. Hettak, D. Hamouma, Implémentation d'une commande numérique des vérins sur un API en vue d'une automatisation de la chaîne de transfert des réfrigérateurs, projet réalisé à l'ENIEM, Mémoire de Master, Département d'Automatique, Faculté de Génie Electrique et d'Informatique, UMMTO, 2009.

[6] : <http://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm> (télécharger en 2016).

[7] : ROIZOT Sébastien 1 LPO ASTIER (AUBENAS 07), <http://www.courselec.free.fr> (consulté en 2016).

[8] : [http://bernarderic4926.perso.sfr.fr/dr/automatisme/api/dr\\_api.pdf](http://bernarderic4926.perso.sfr.fr/dr/automatisme/api/dr_api.pdf) (télécharger en 2016).

[9] : Alain GONZAGA,  
[sitelec.org/download.php?filename=coursautomates\\_programmables\\_industriels.pdf](http://sitelec.org/download.php?filename=coursautomates_programmables_industriels.pdf) (téléchargé en 2016).

[10] : [http://pamelard.electro.pagesperso-orange.fr/fichier%20pdf/automatisme\\_programmation%20des%20API%20cours.pdf](http://pamelard.electro.pagesperso-orange.fr/fichier%20pdf/automatisme_programmation%20des%20API%20cours.pdf) (téléchargé en 2016).

[11] : L. BERGOUGNOUX, POLYTECH - Marseille, Département de Mécanique Energétique 2eAnnée Option S.I.I.C, 2004–2005 (téléchargé en 2016).

[12] : <http://tvaira.free.fr/reseaux/rli.pdf> (téléchargé en 2016).

[13] : [http://www.acgrenoble.fr/ecole.entreprise/CRGE/cteresources/Reseaux\\_Terrain/Reseaux\\_de\\_terrain.pdf](http://www.acgrenoble.fr/ecole.entreprise/CRGE/cteresources/Reseaux_Terrain/Reseaux_de_terrain.pdf) (téléchargé en 2016).

[14] : R. AMAOUZ, H. KECHAD, Etude et automatisation d'une cisaille à guillotine à l'aide de l'API S7-300, projet réalisé à l'ENIEM, Mémoire de Master, Département d'Automatique, Faculté de Génie Electrique et d'Informatique, UMMTO, 2008-2009.

[15] : Documentation technique, mise à disposition par le service de maintenance plastique de l'unité « froid » à l'ENIEM.

[16] : S. Akrou, N. Arabe, Automatisation et supervision d'une station de transport de sucre, projet réalisé à l'usine Coca-Cola Rouiba, Mémoire de Master, Département d'Automatique, Faculté de Génie Electrique et d'Informatique, UMMTO, 2009.

[17] : M. Djellout, Y. Guenfoud, Développement de commande décentralisée et supervision de l'unité traitement brut (Séparateurs et Manifold) du centre de production de Gassi Toil SONATRACH, projet réalisé au centre de production de Gassi Toil SONATRACH, Mémoire de Master, Département d'Automatique, Faculté de Génie Electrique et d'Informatique, UMMTO, 2009.

[18] : [www.eniem.com.dz](http://www.eniem.com.dz) (consulté en 2016).

[19] : [httpsti-beziers.fr/ftsipmspip\\_tsipmhtmljgbplastiquesobtention%20plastique.htm#pro7](http://sti-beziers.fr/ftsipmspip_tsipmhtmljgbplastiquesobtention%20plastique.htm#pro7) (consulté en 2016).

[20] : [https://fr.wikipedia.org/wiki/Machine\\_%C3%A0\\_courant\\_continu](https://fr.wikipedia.org/wiki/Machine_%C3%A0_courant_continu) (consulté en 2016).

# Annexes

# *Annexe A.*

Programme permettant le  
fonctionnement de la machine  
Expandeur ACE 25 AR

**OB1 - <offline>**

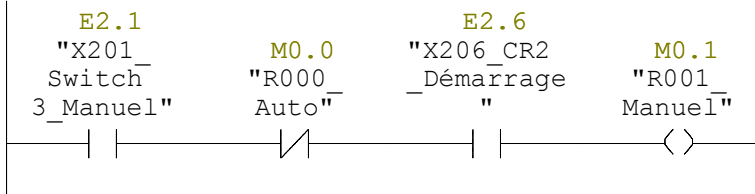
"Cycle Execution"

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 08/09/2016 10:55:04  
**Interface :** 15/02/1996 16:51:12  
**Longueur (bloc/code /données locales) :** 02438 02118 00022

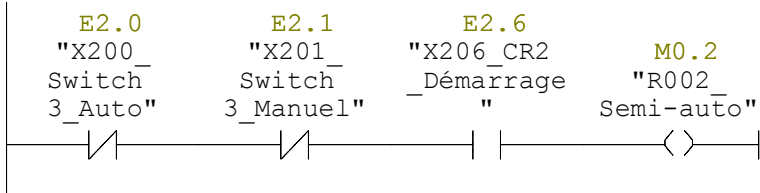
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

**Bloc : OB1 "Main Program Sweep (Cycle)"**

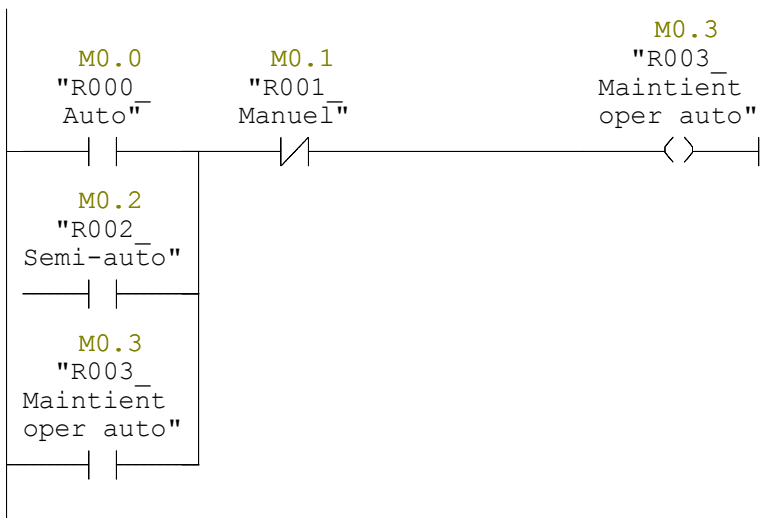
Réseau : 1



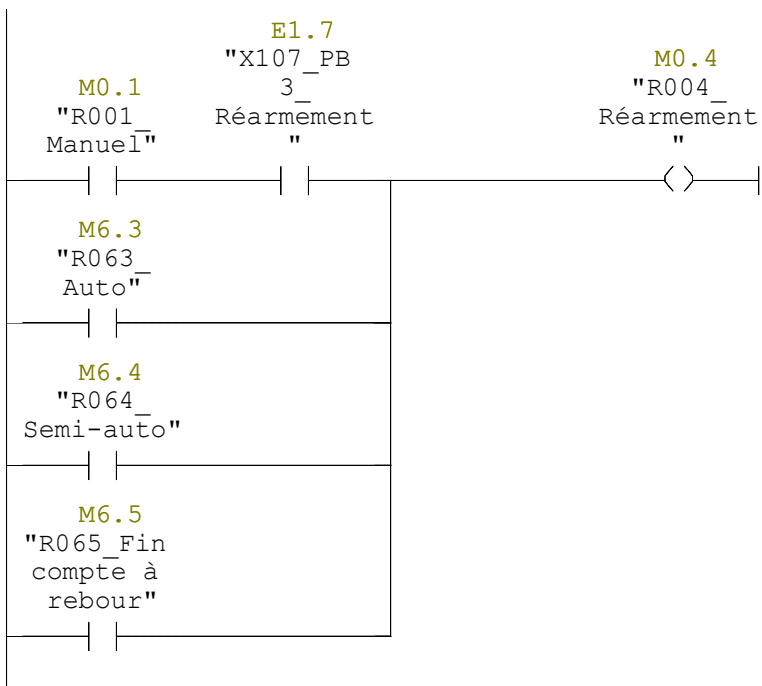
Réseau : 2



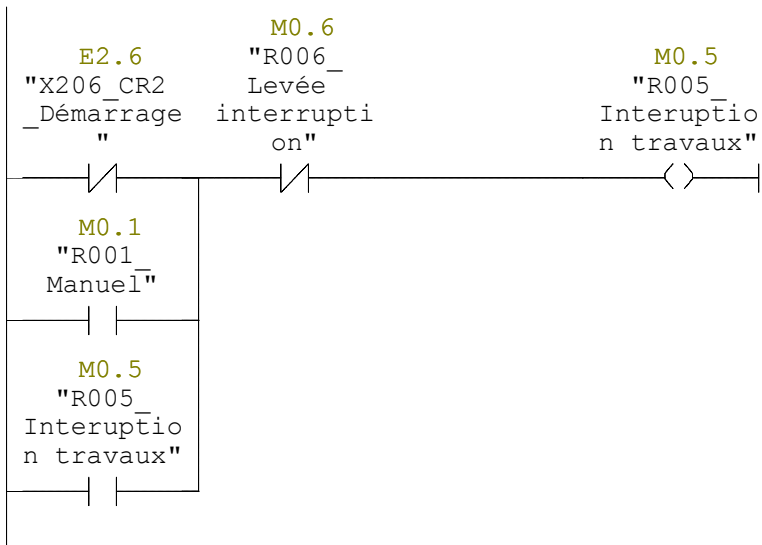
Réseau : 3



Réseau : 4



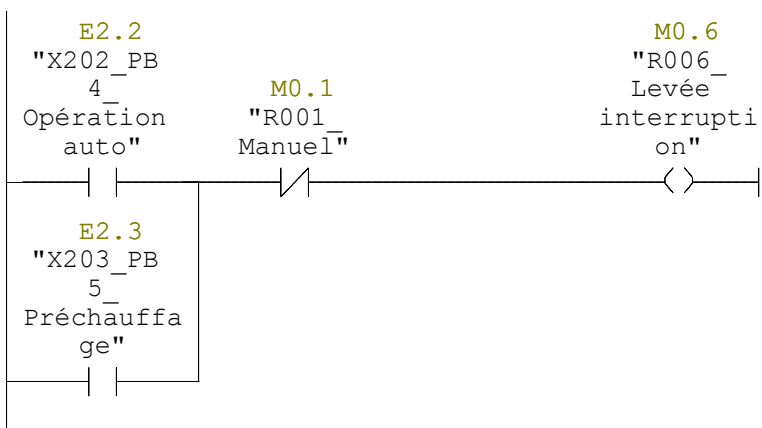
Réseau : 5



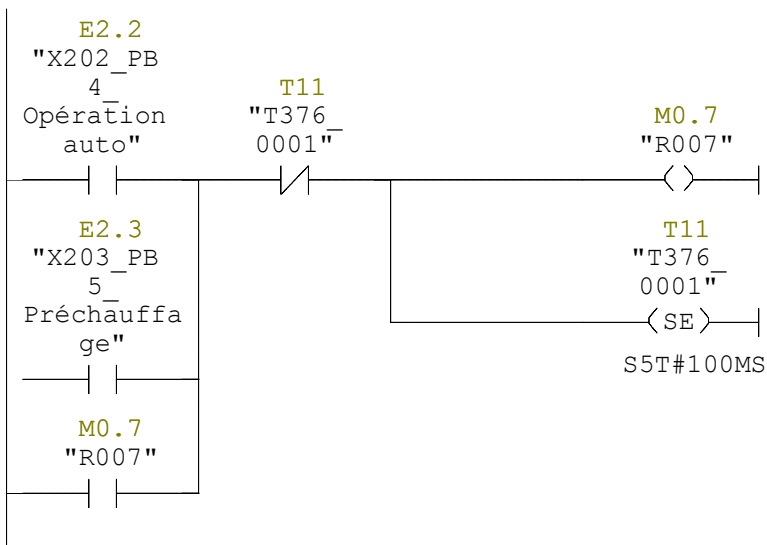
Réseau : 6



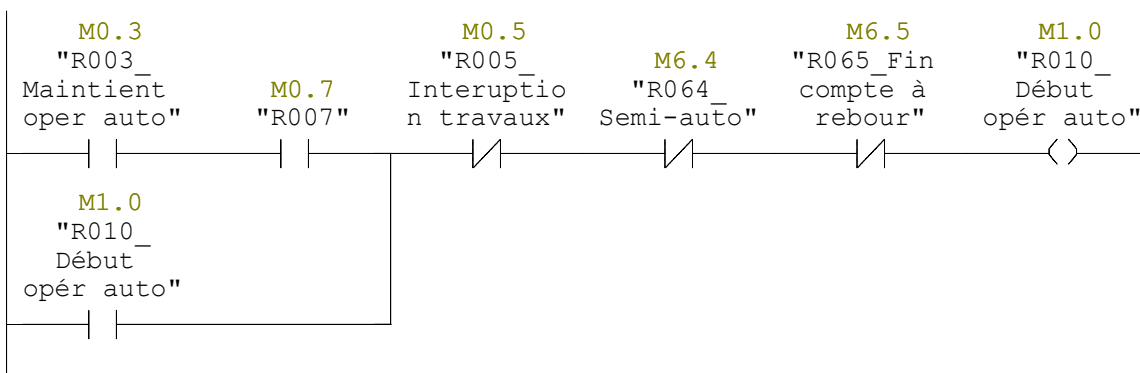
Réseau : 7



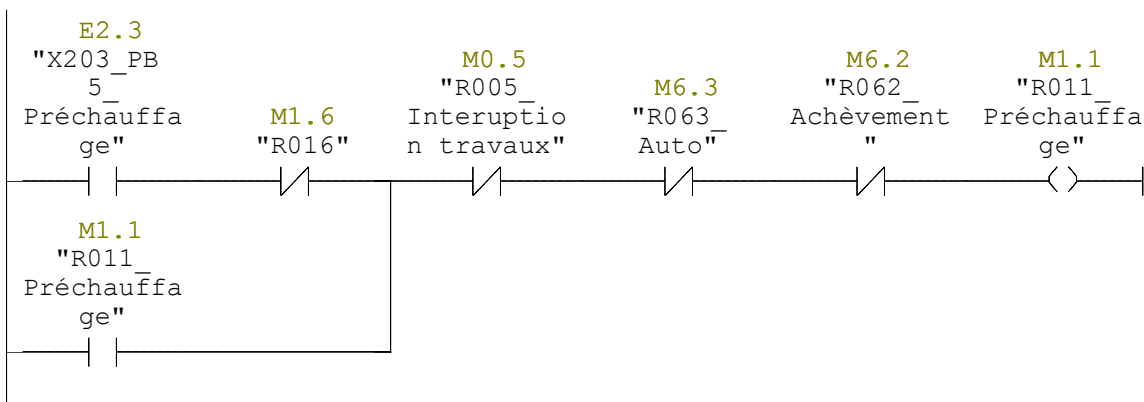
Réseau : 8



Réseau : 9



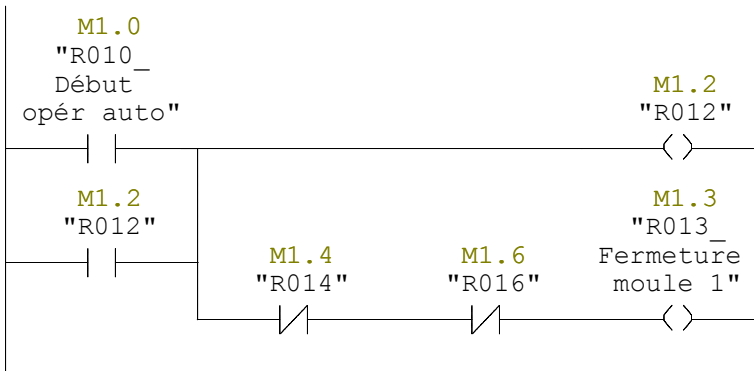
Réseau : 10



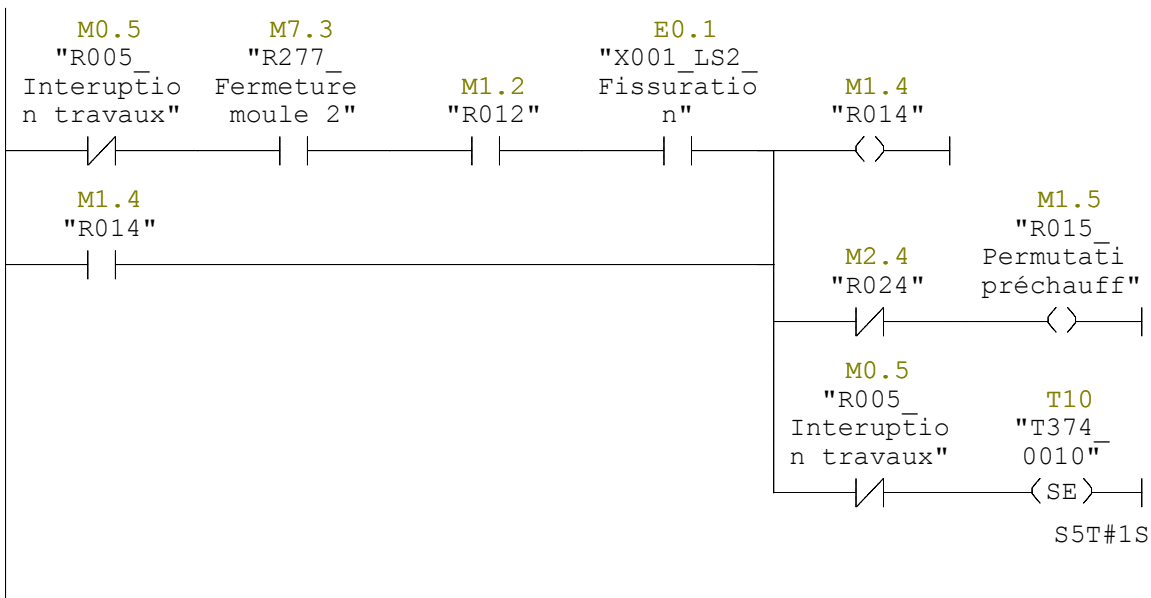
Réseau : 11



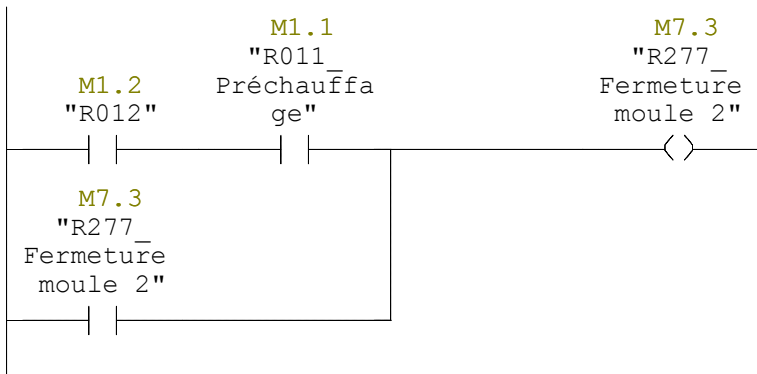
Réseau : 12



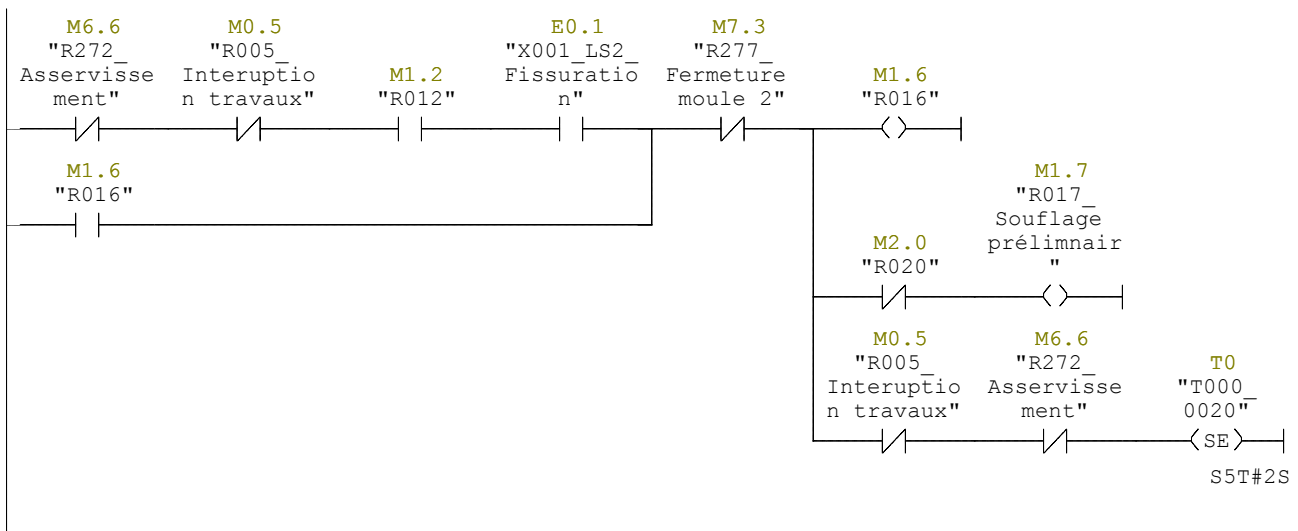
Réseau : 13



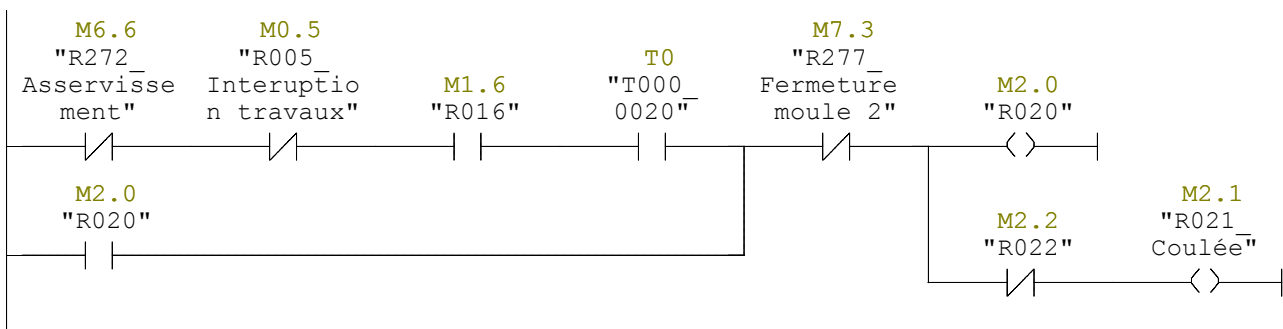
Réseau : 14



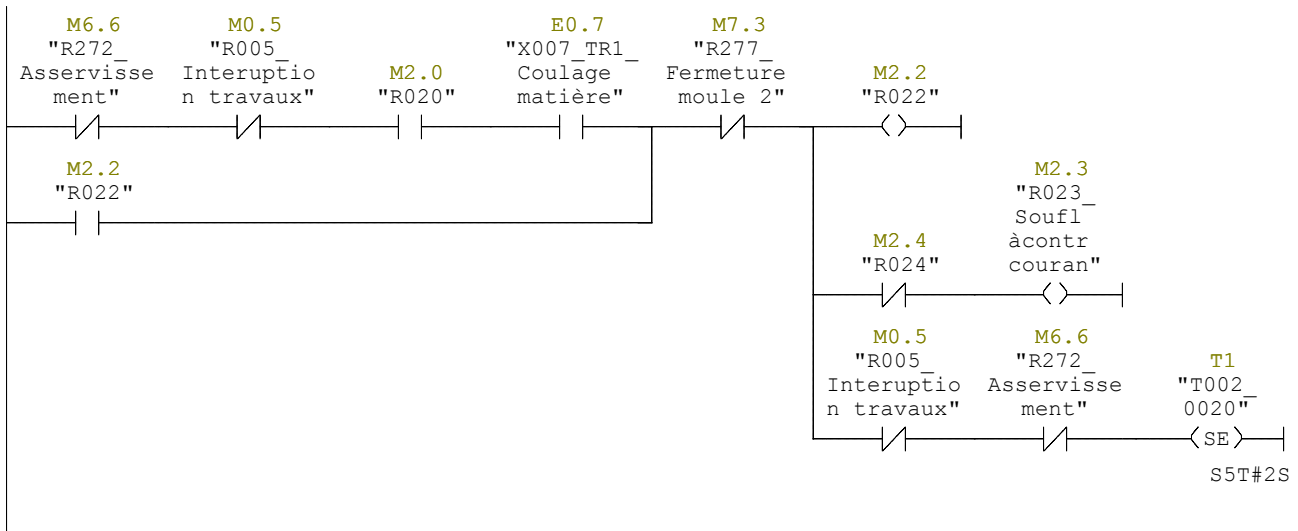
Réseau : 15



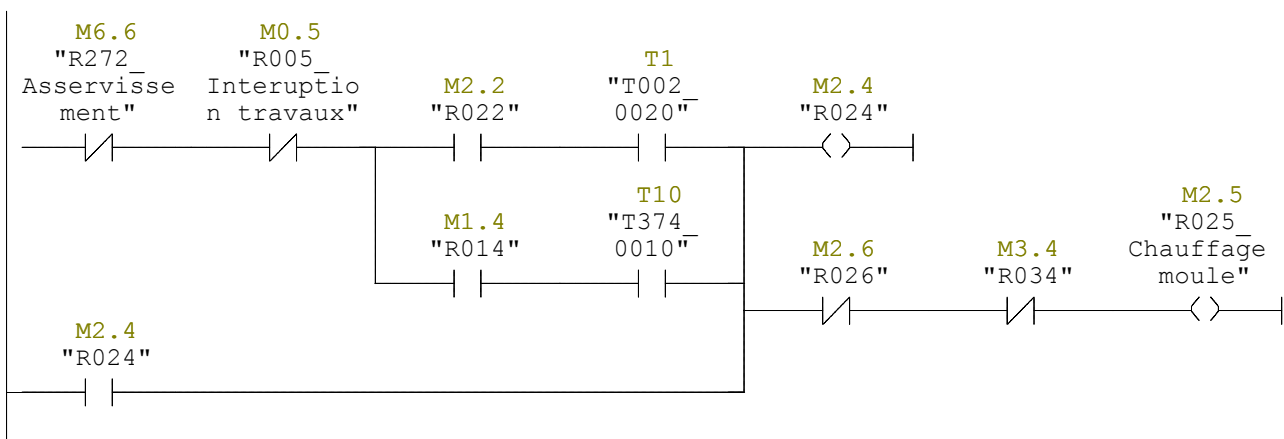
Réseau : 16



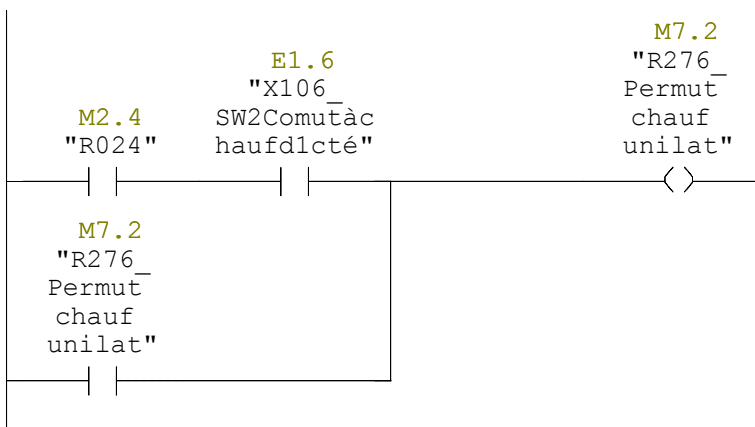
## Réseau : 17



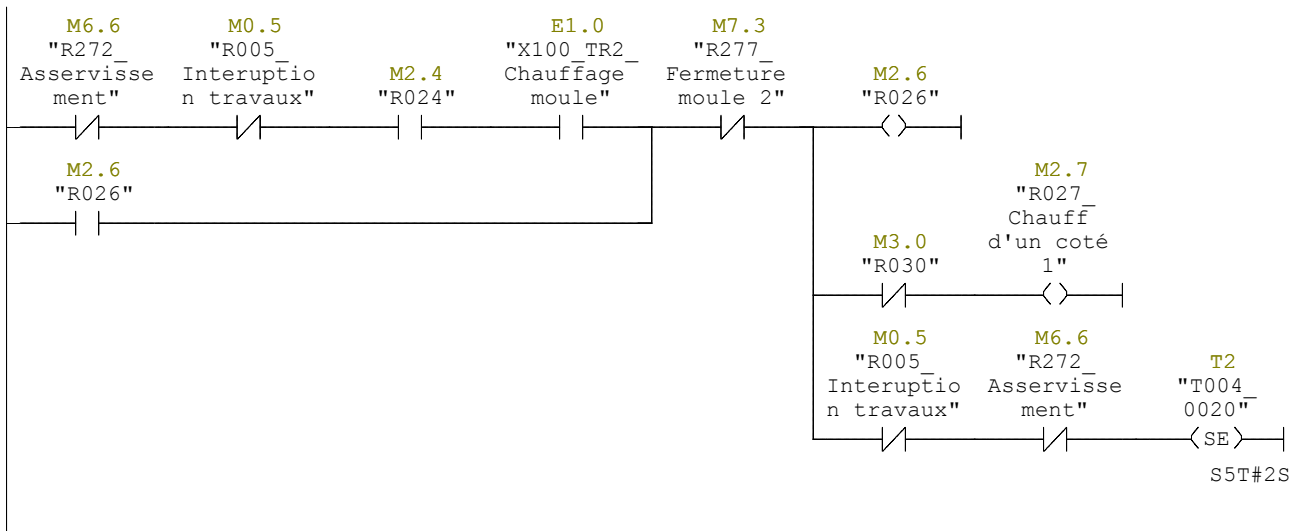
## Réseau : 18



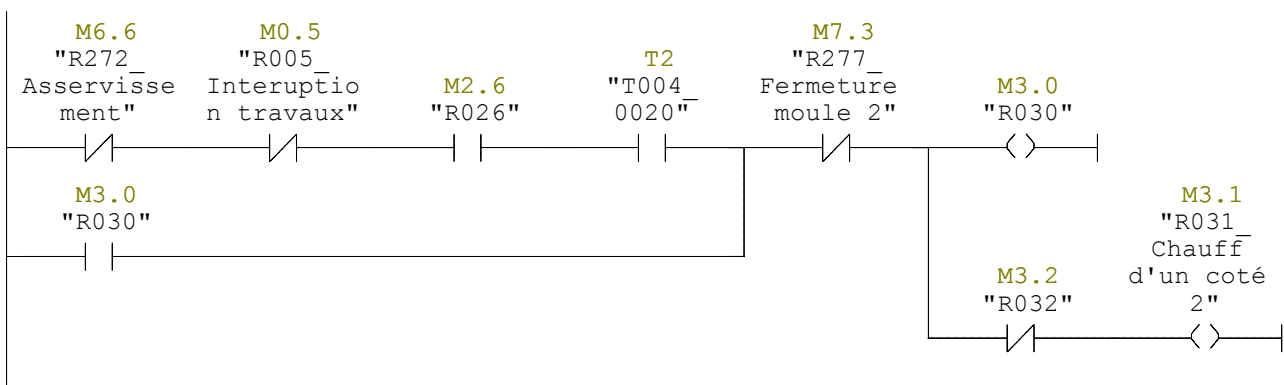
## Réseau : 19



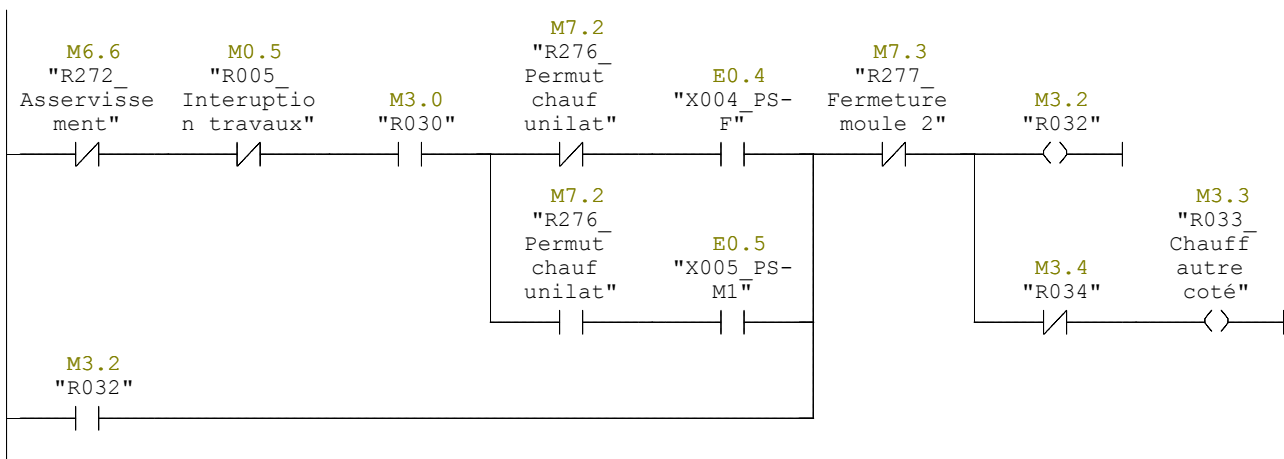
Réseau : 20



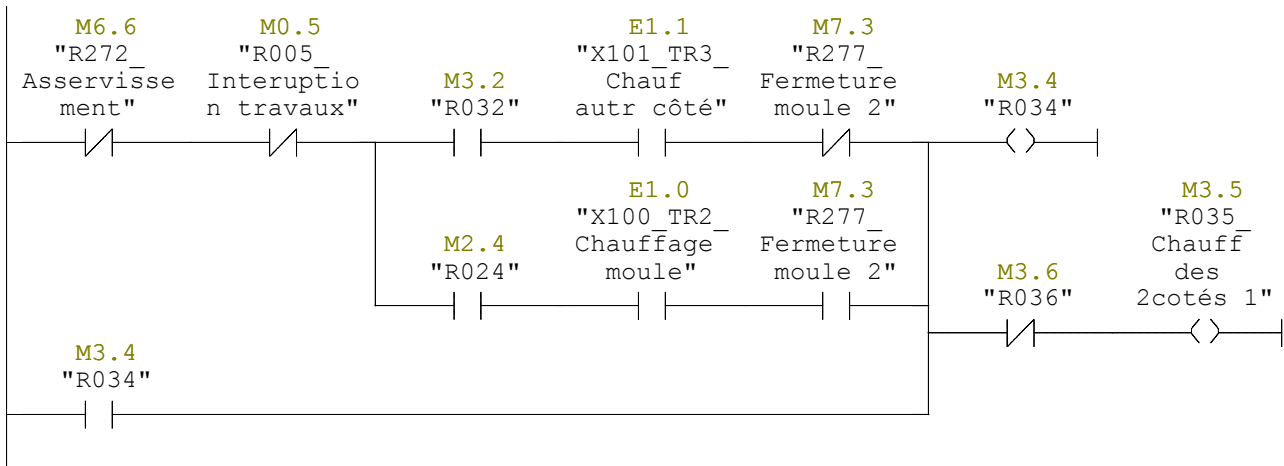
Réseau : 21



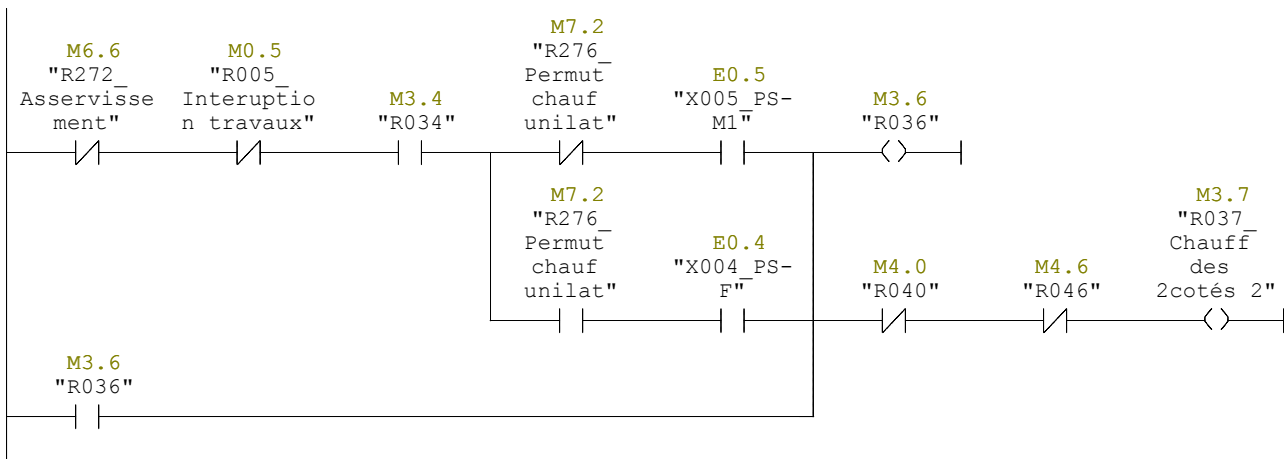
Réseau : 22



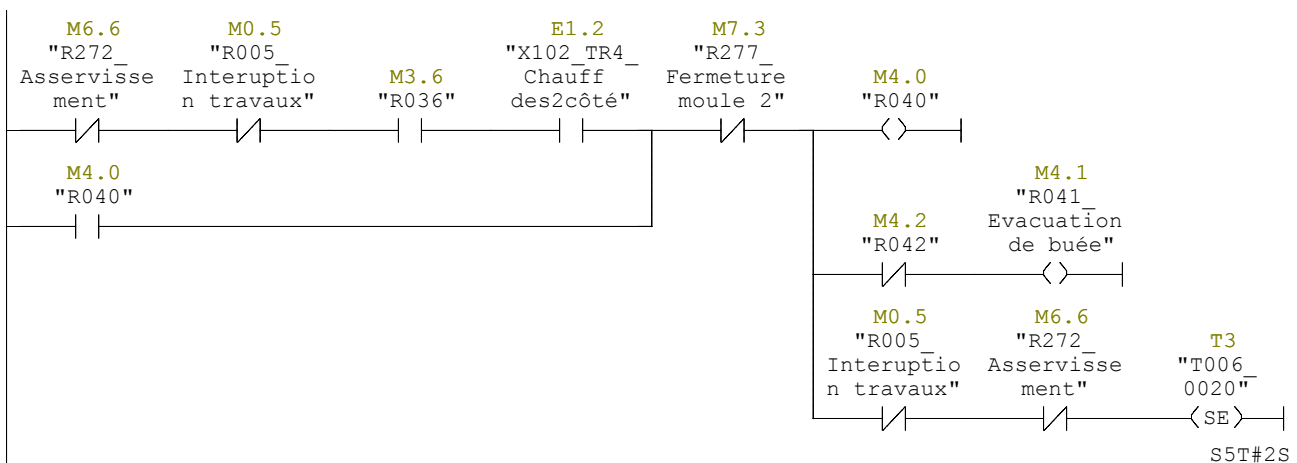
## Réseau : 23



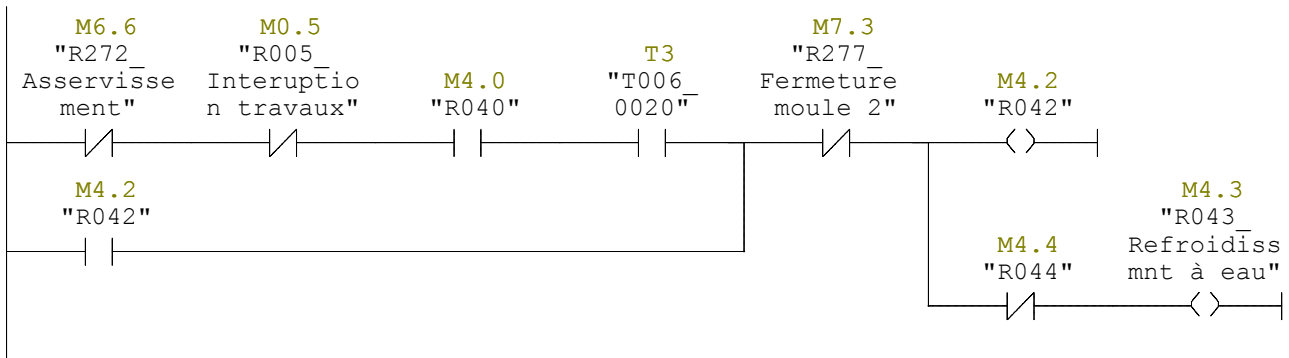
## Réseau : 24



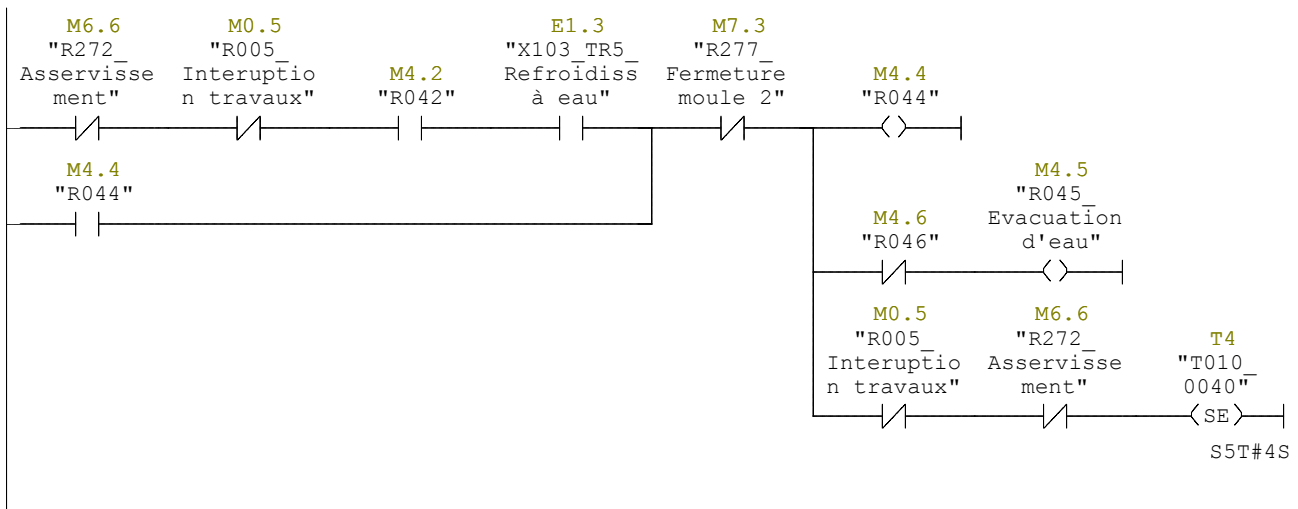
## Réseau : 25



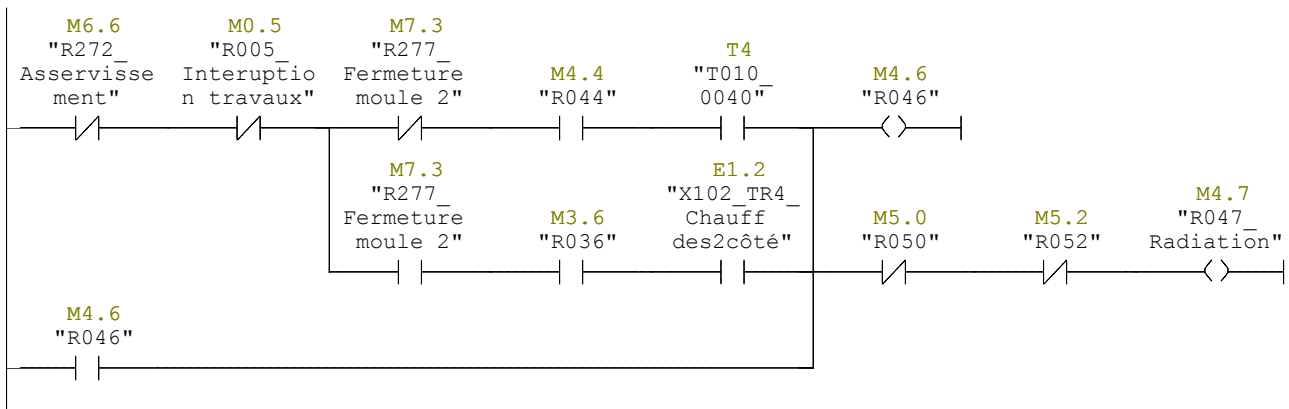
Réseau : 26



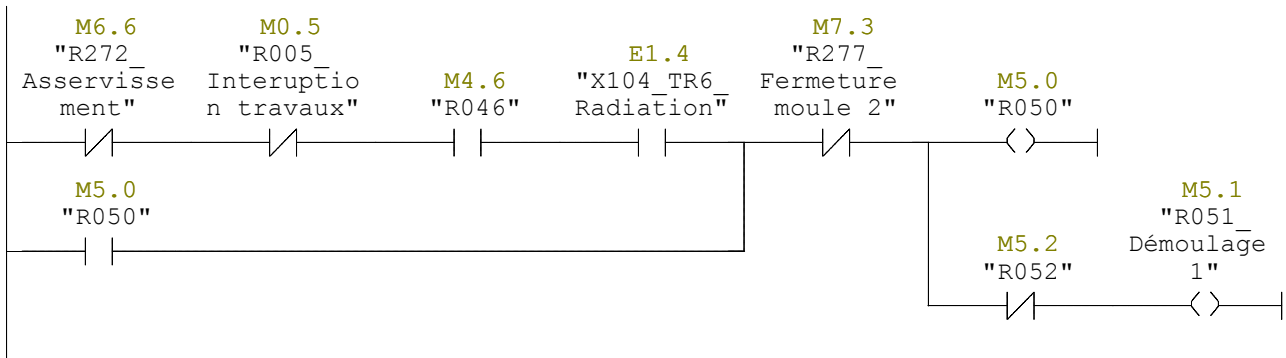
Réseau : 27



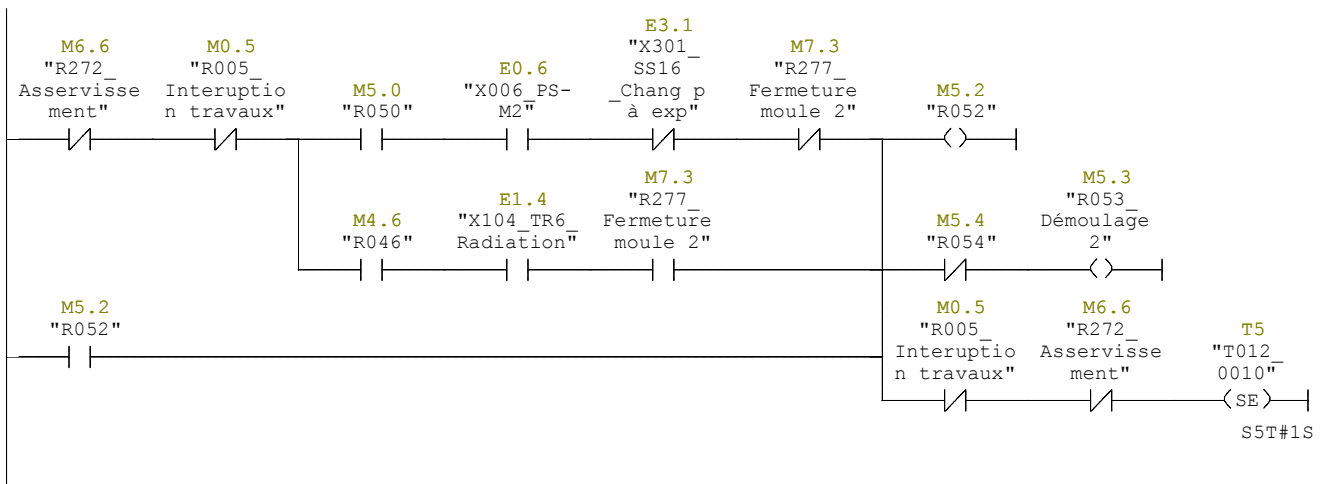
Réseau : 28



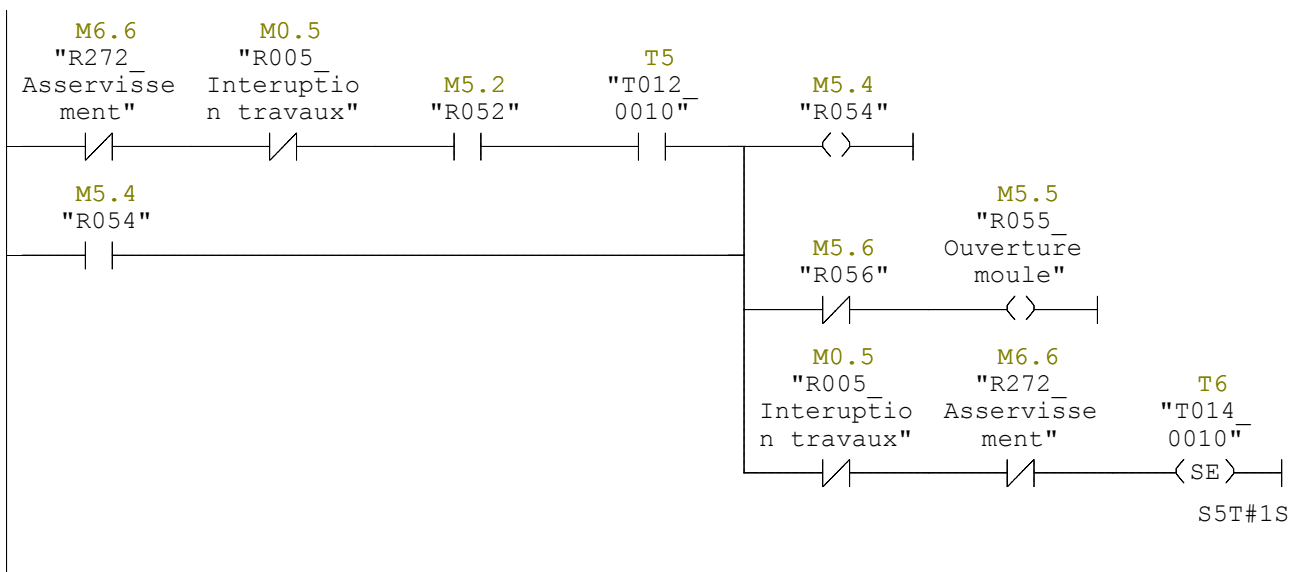
Réseau : 29



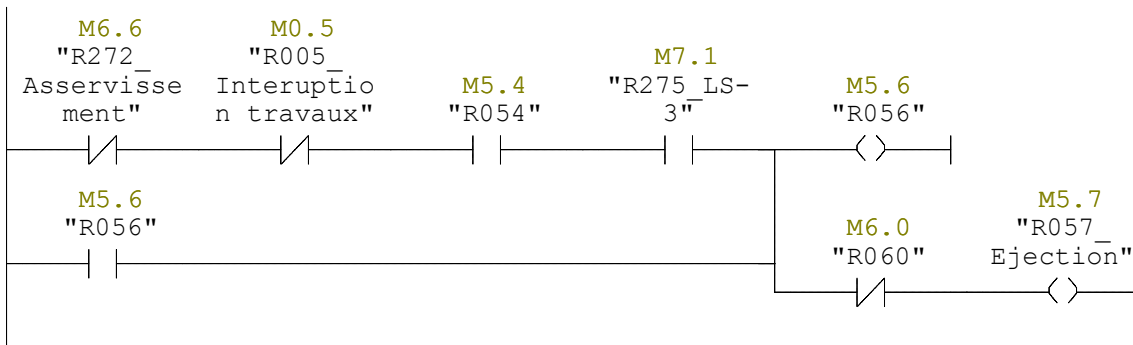
Réseau : 30



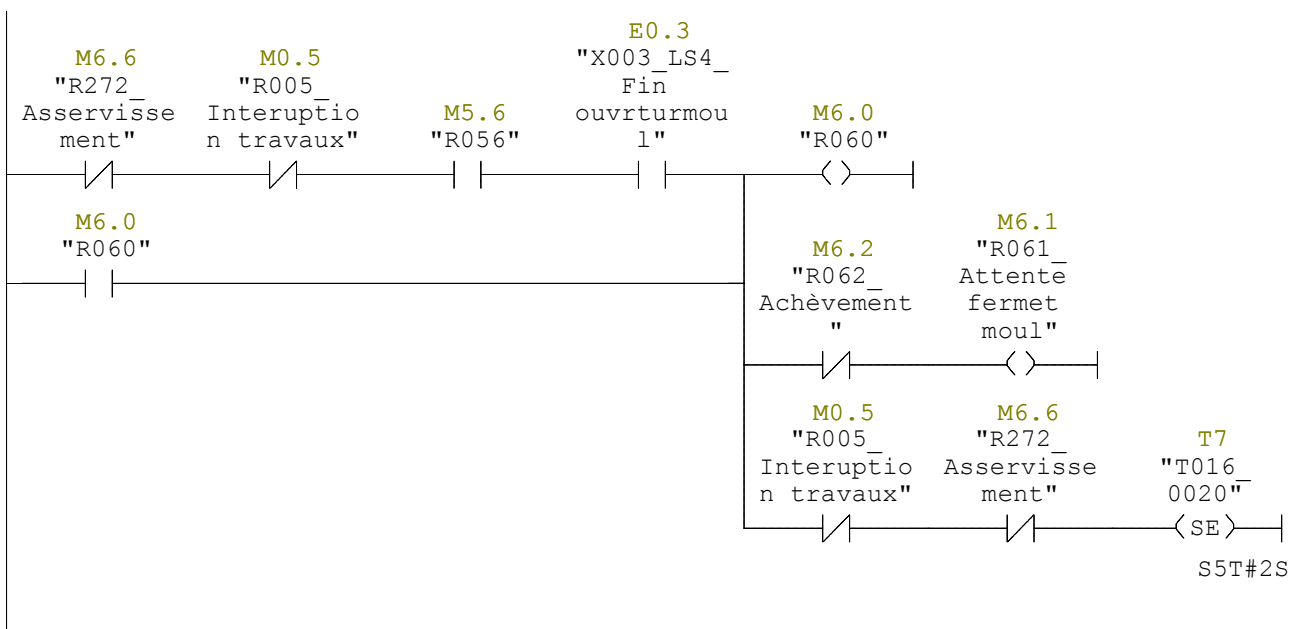
Réseau : 31



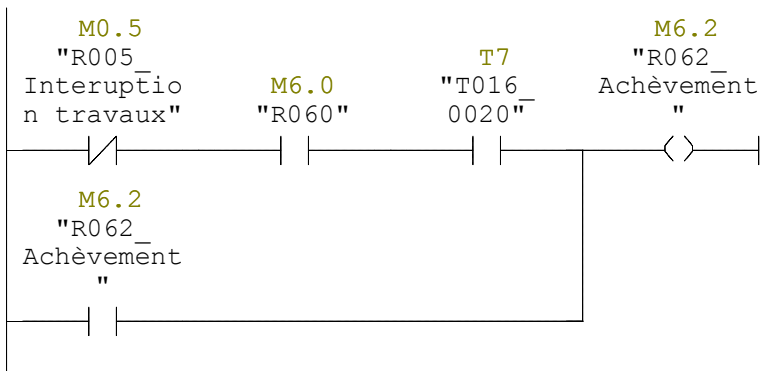
Réseau : 32



Réseau : 33



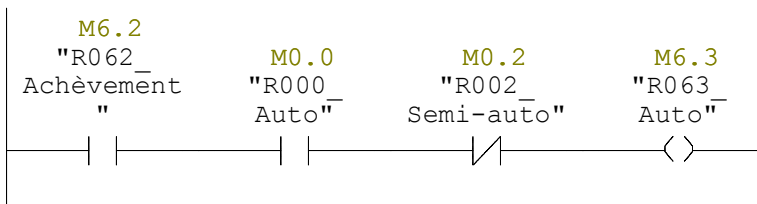
Réseau : 34



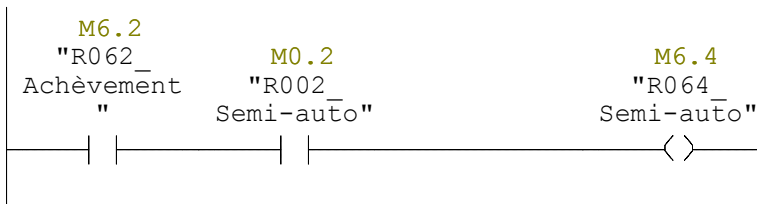
Réseau : 35



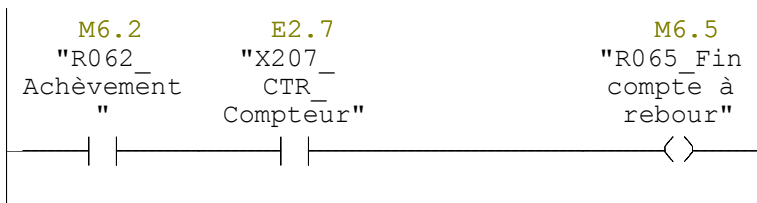
Réseau : 36



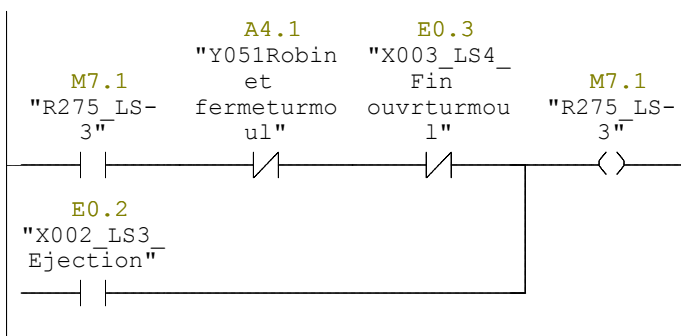
Réseau : 37



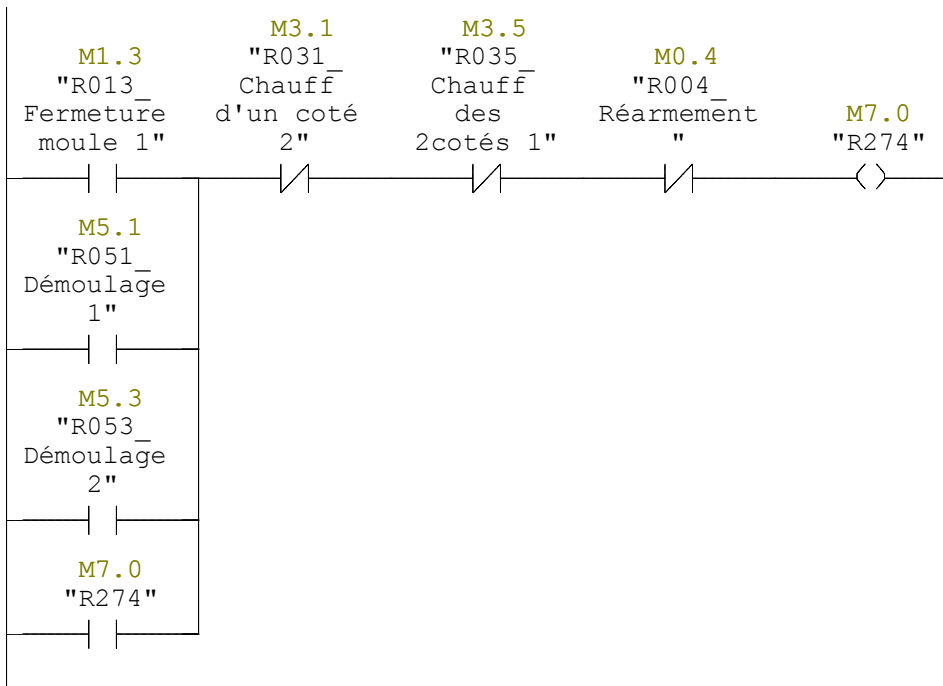
Réseau : 38



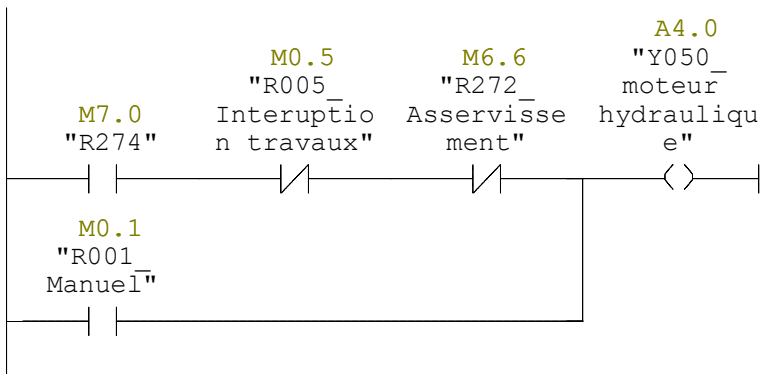
Réseau : 39



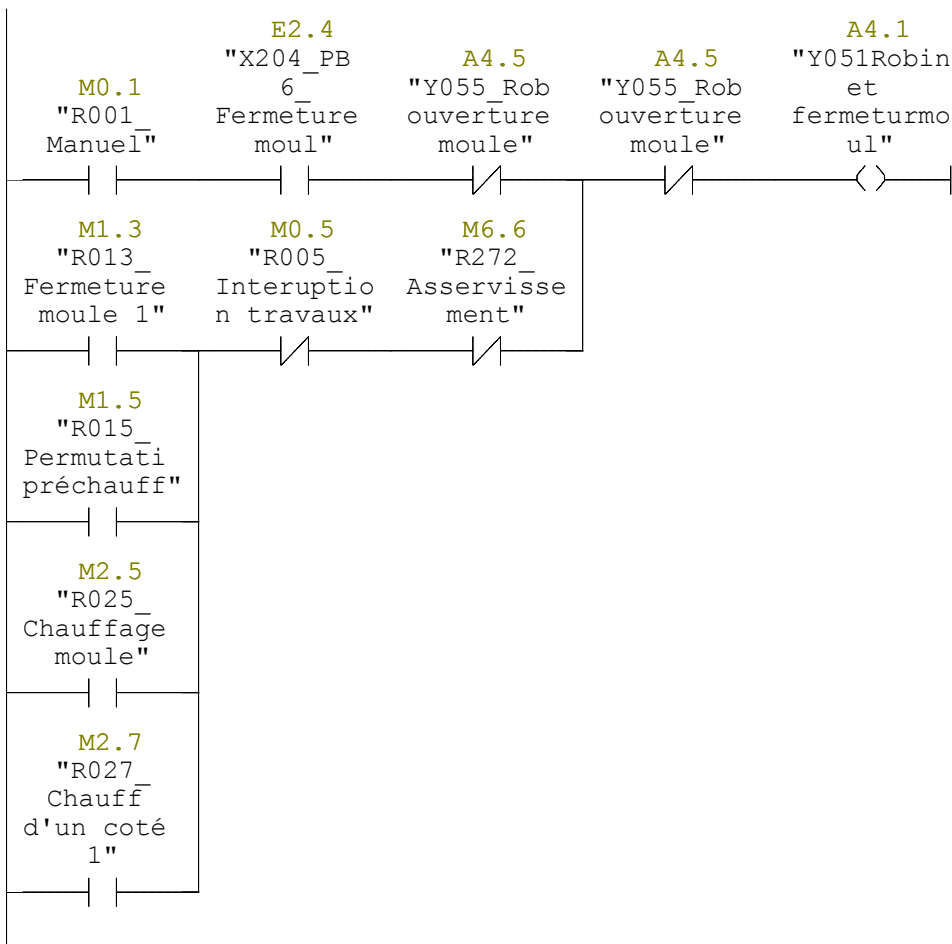
Réseau : 40



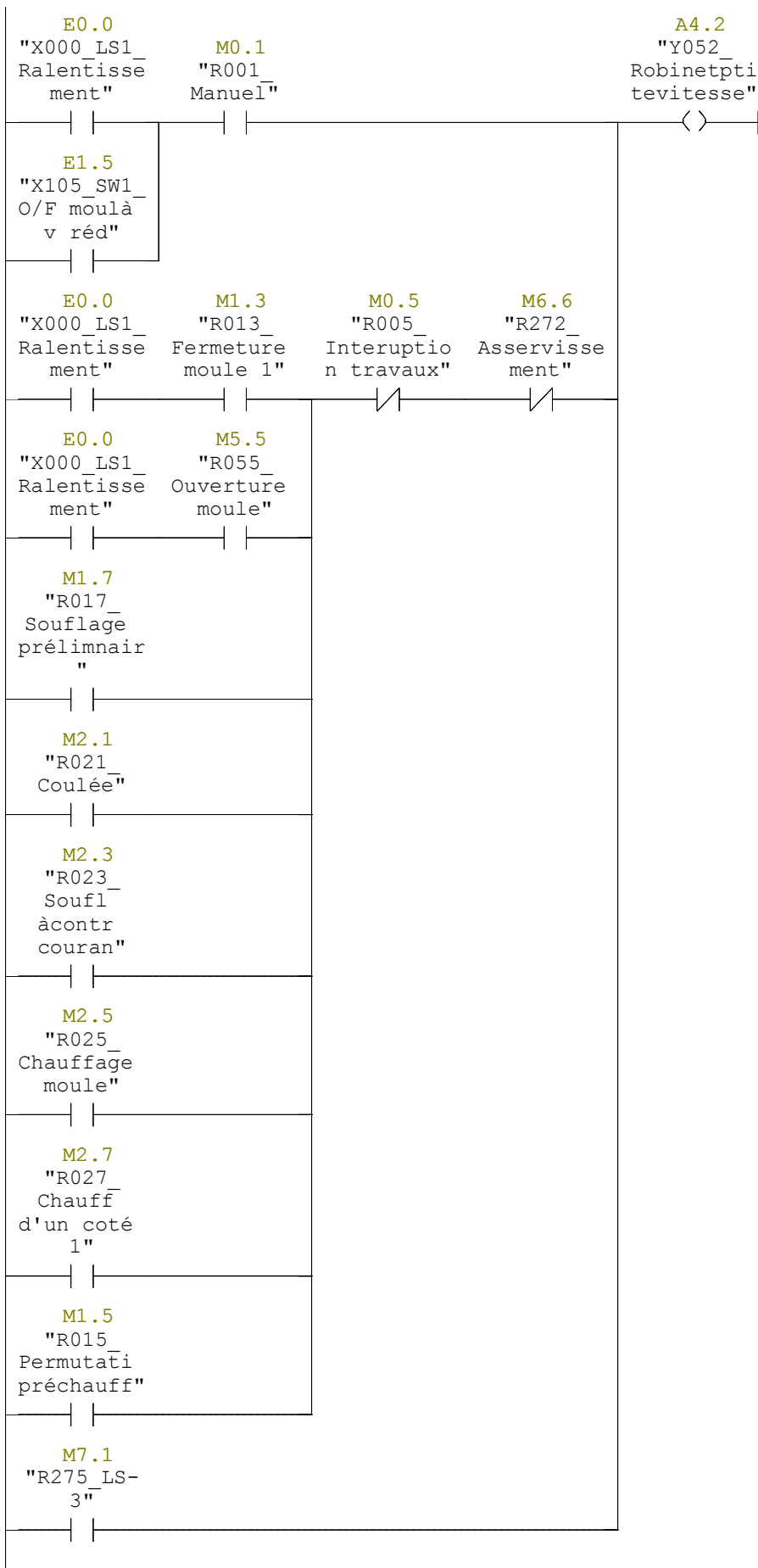
Réseau : 41



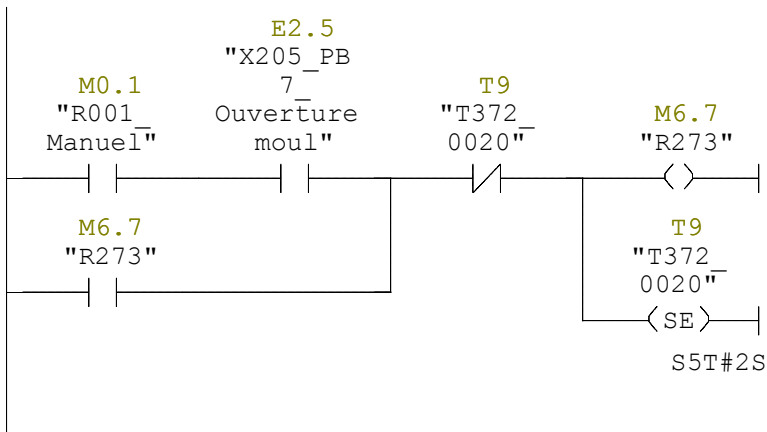
Réseau : 42



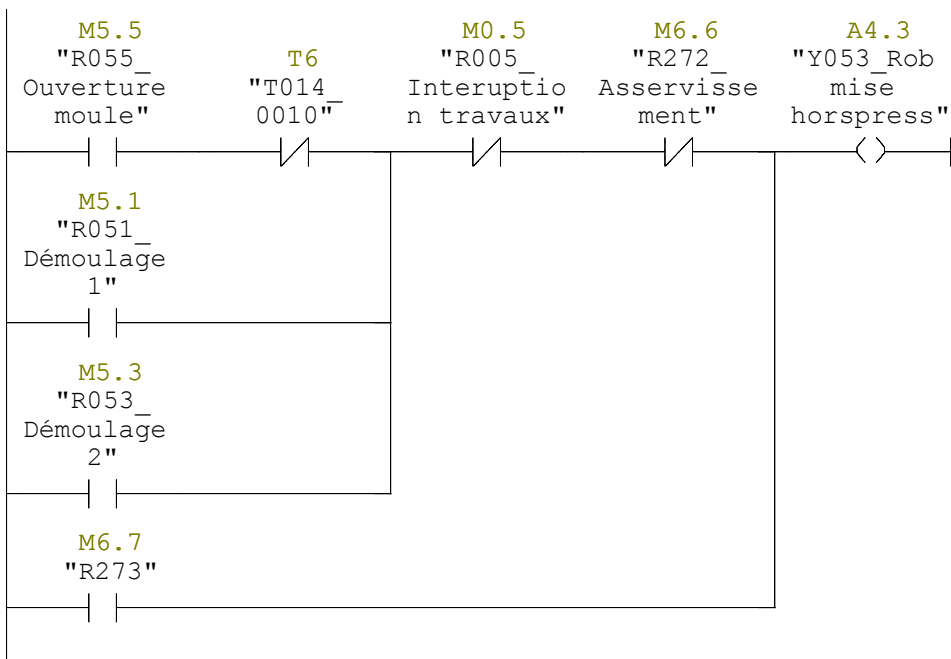
Réseau : 43



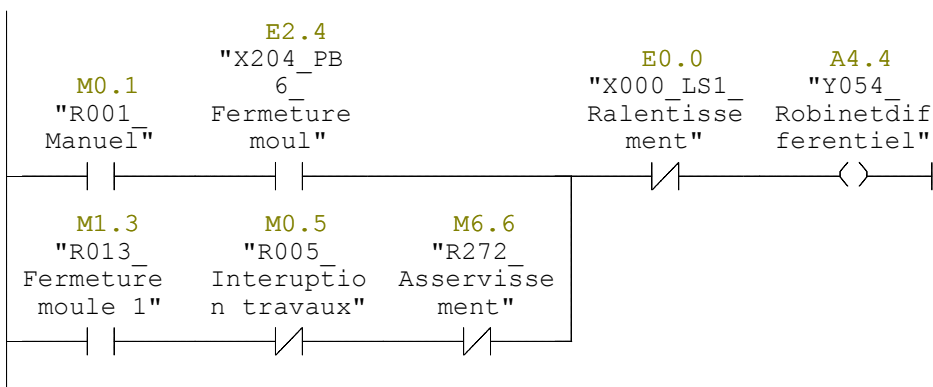
Réseau : 44



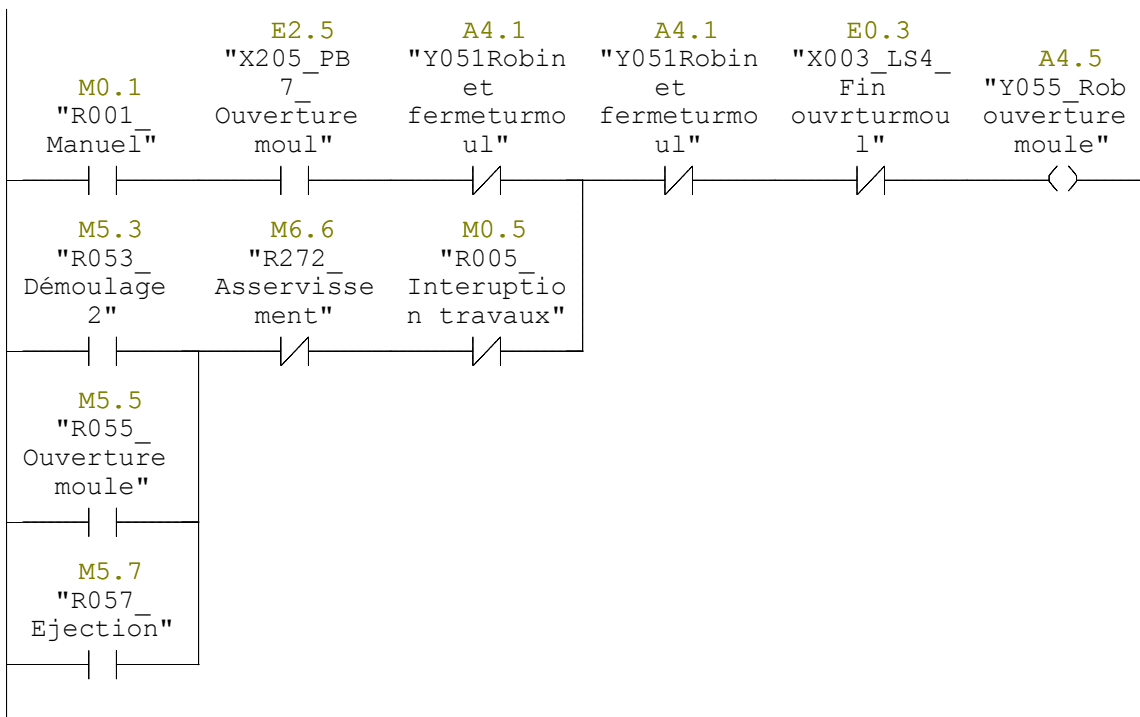
Réseau : 45



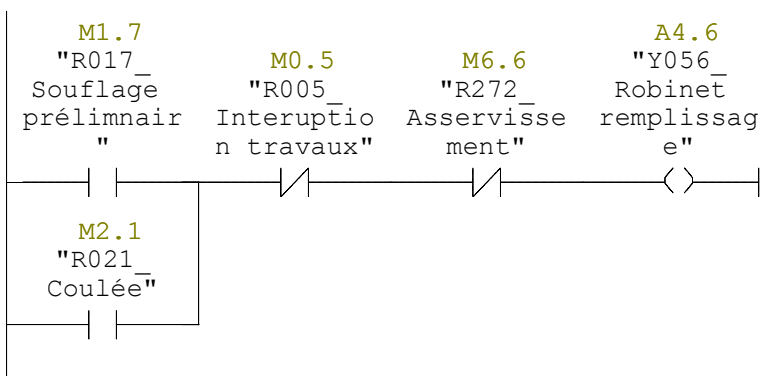
Réseau : 46



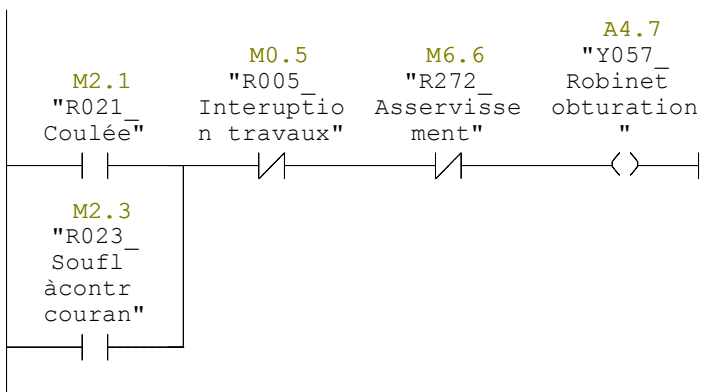
Réseau : 47



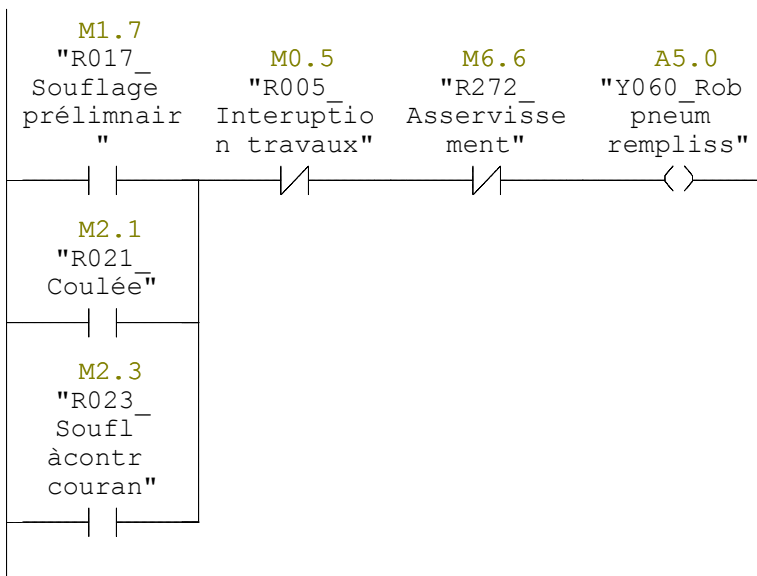
Réseau : 48



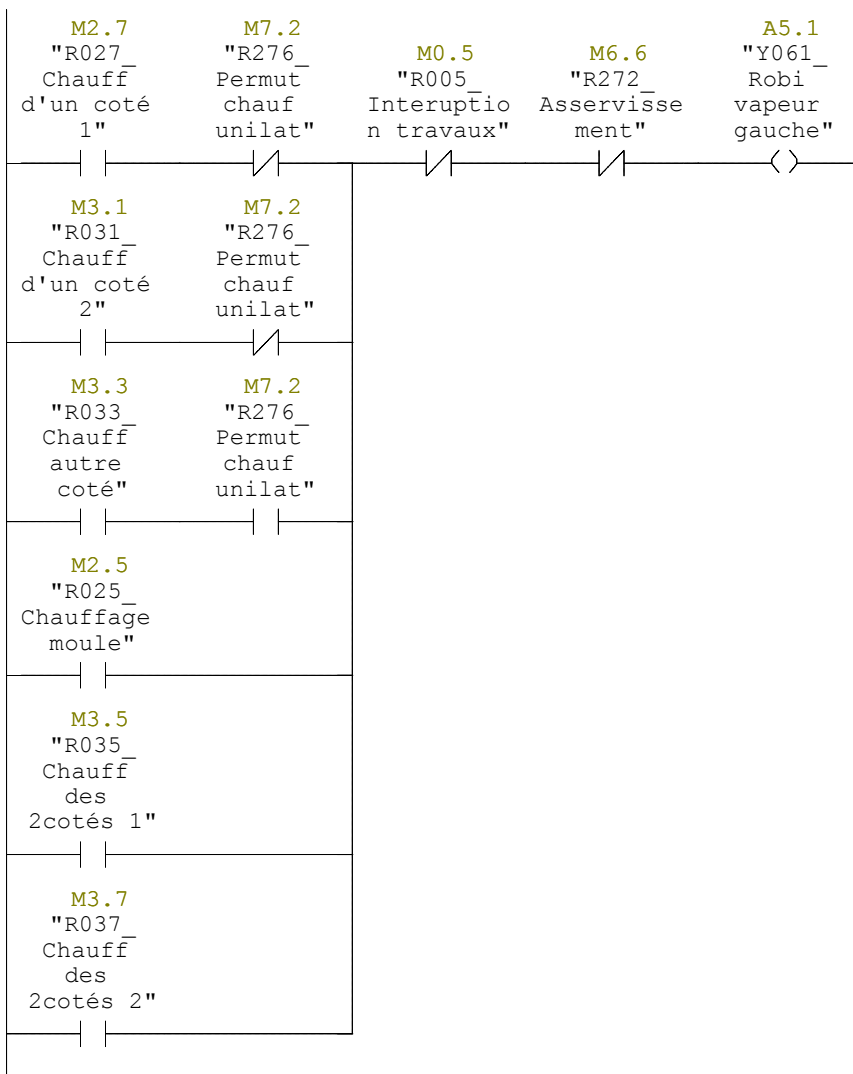
Réseau : 49



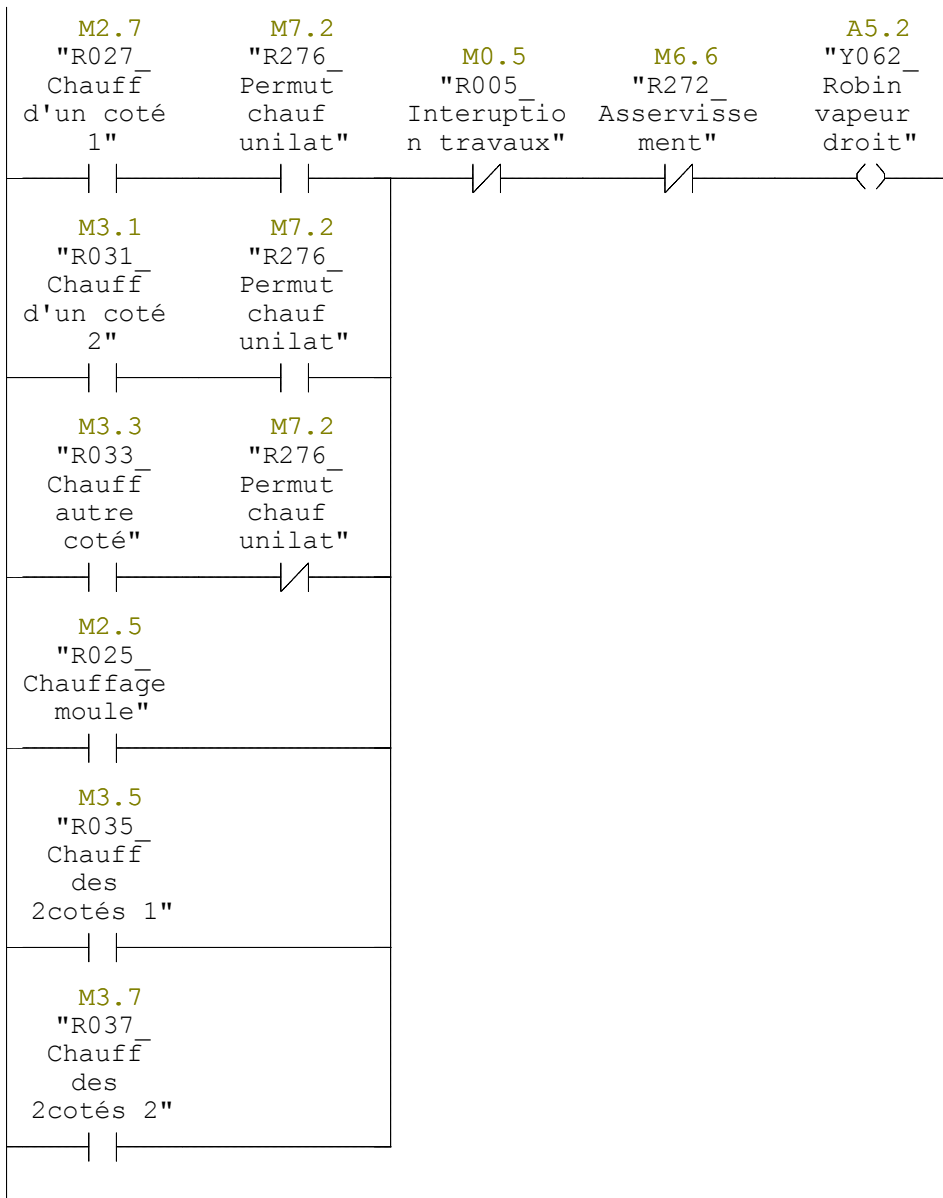
Réseau : 50



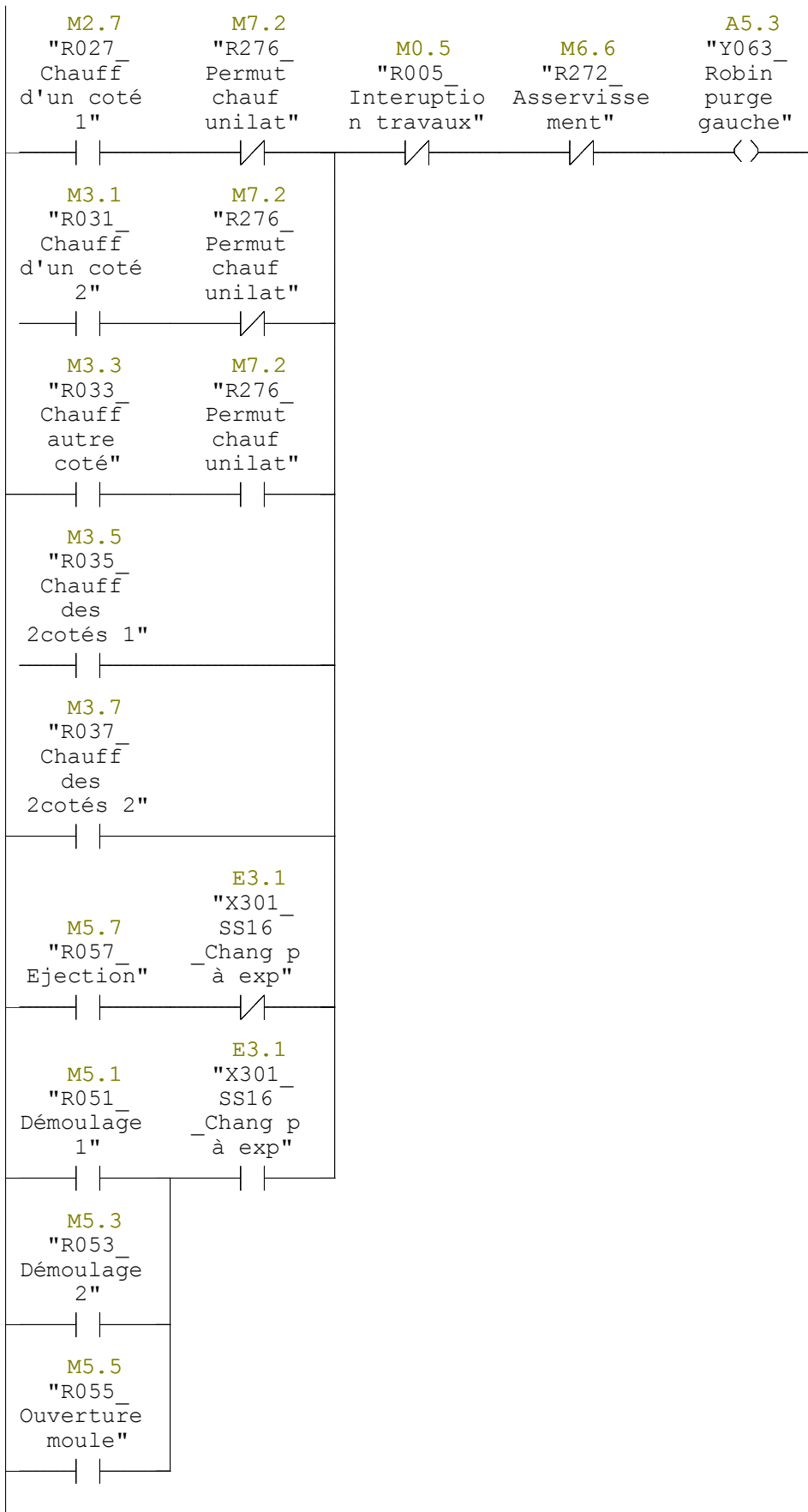
Réseau : 51



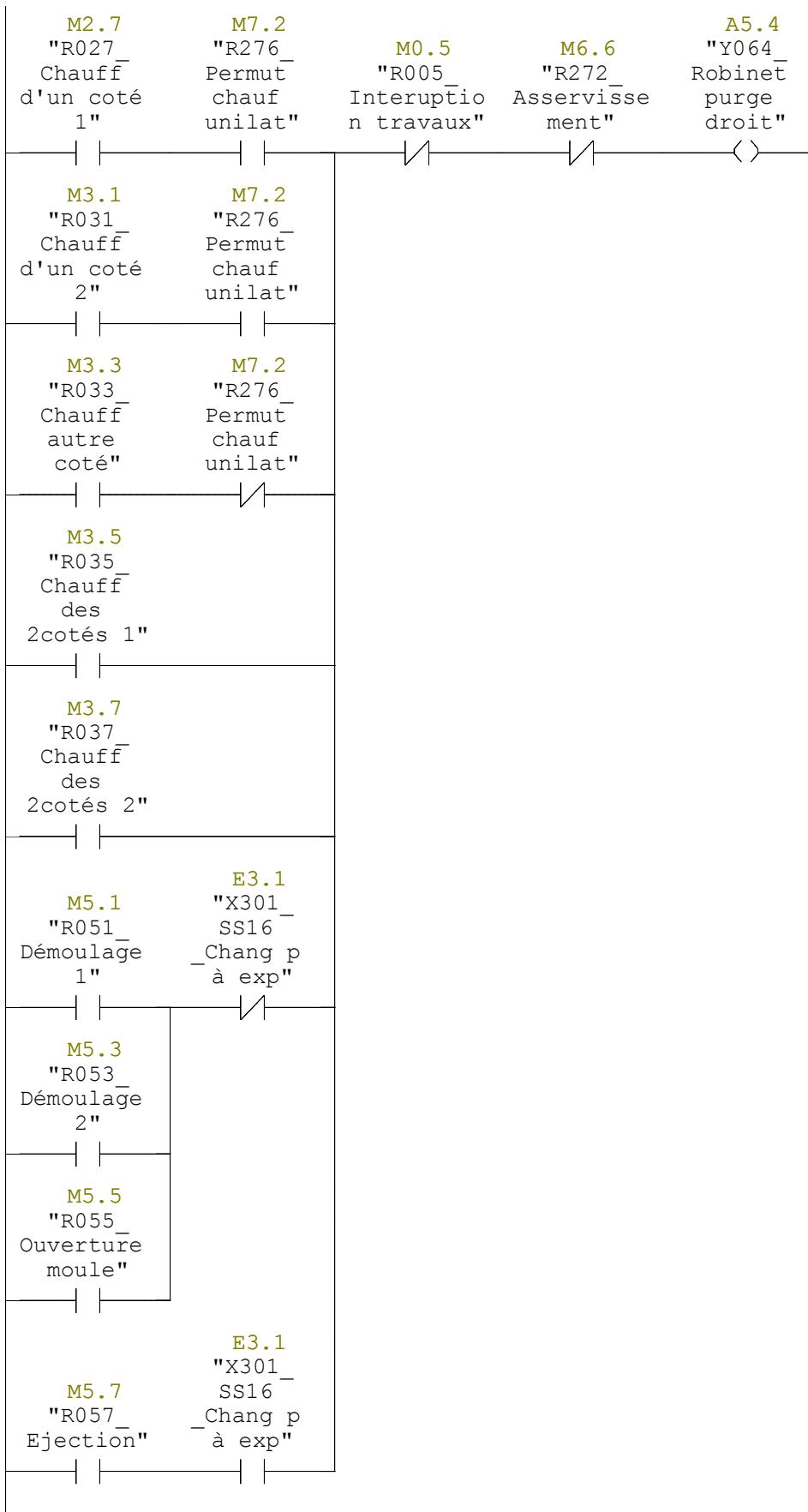
Réseau : 52



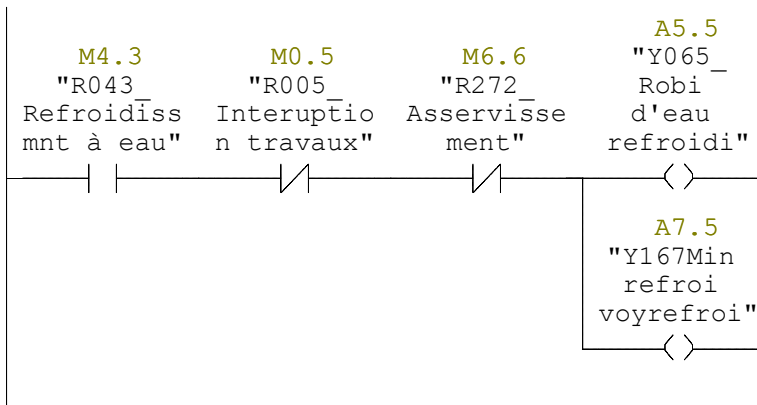
Réseau : 53



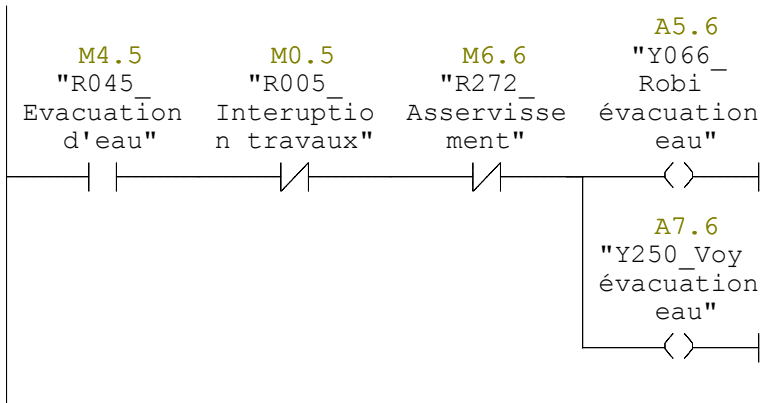
Réseau : 54



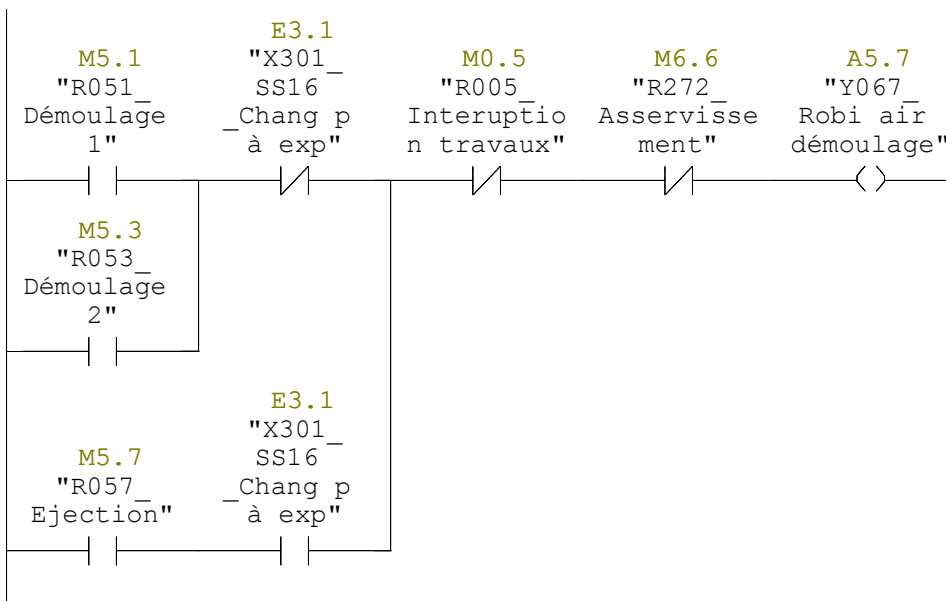
Réseau : 55



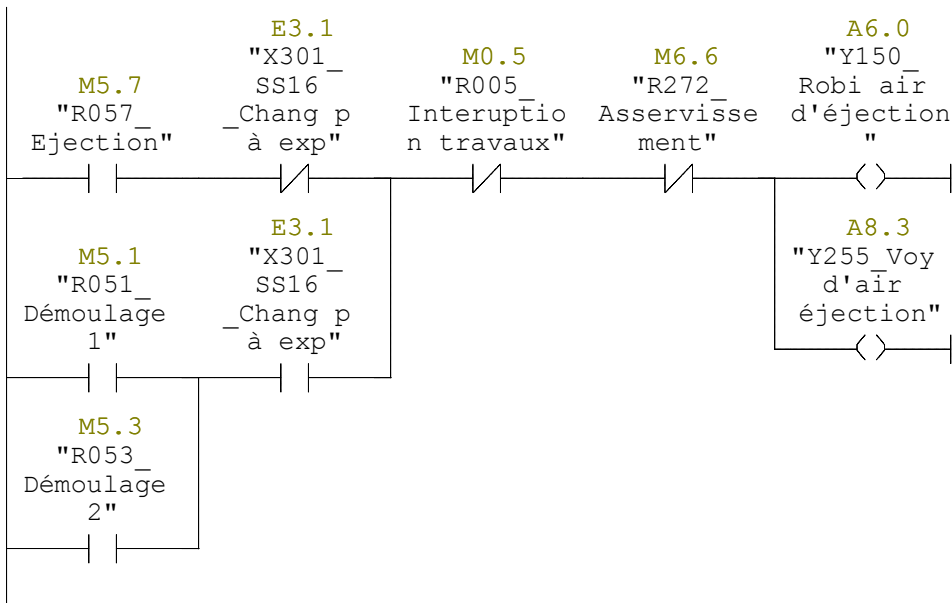
Réseau : 56



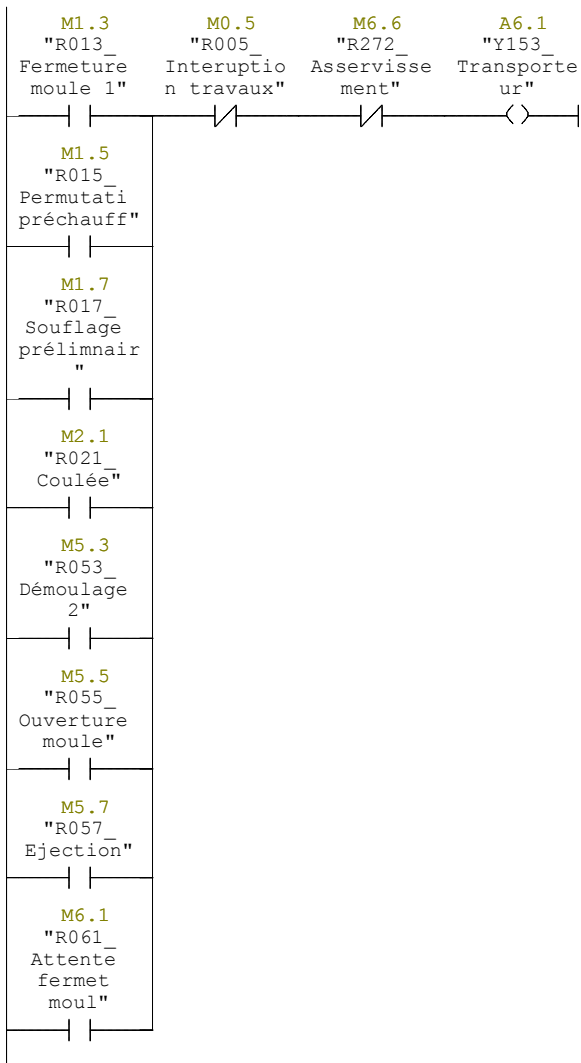
Réseau : 57



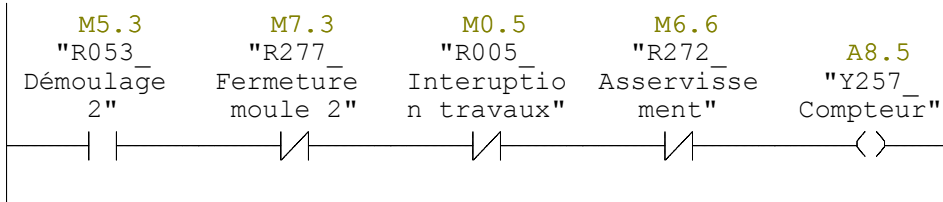
Réseau : 58



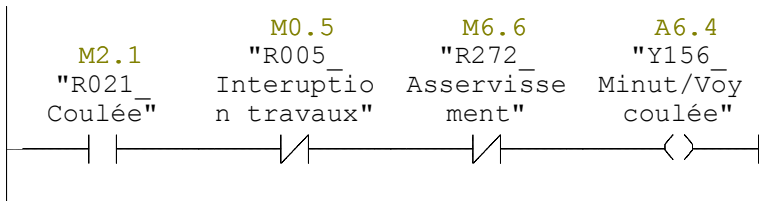
Réseau : 59



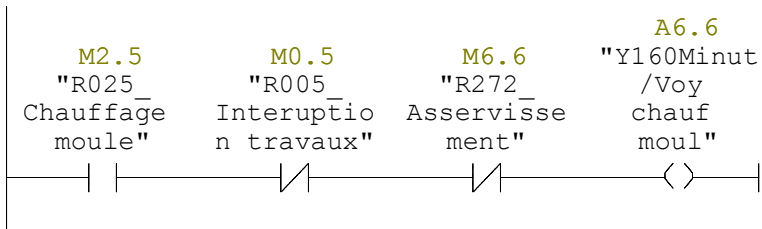
Réseau : 60



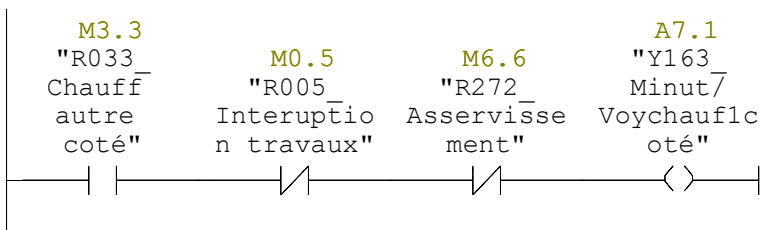
Réseau : 61



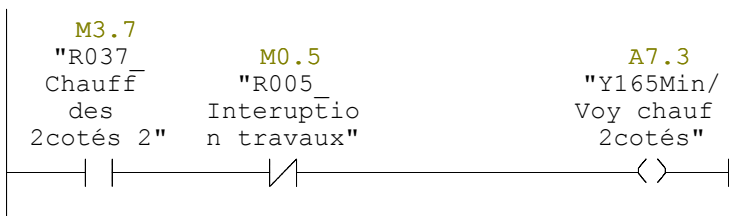
Réseau : 62



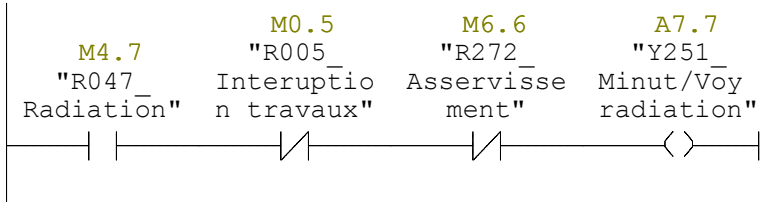
Réseau : 63



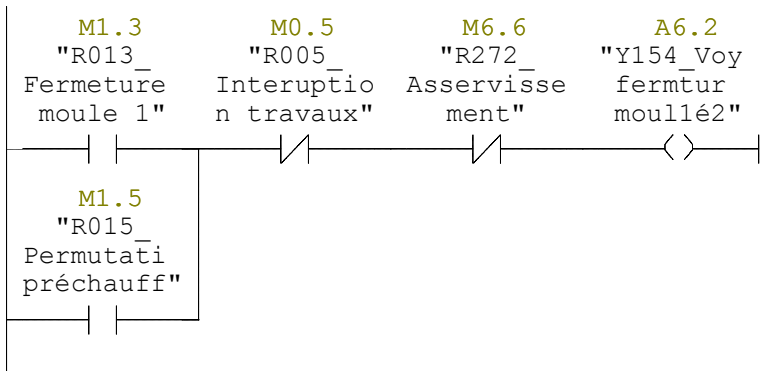
Réseau : 64



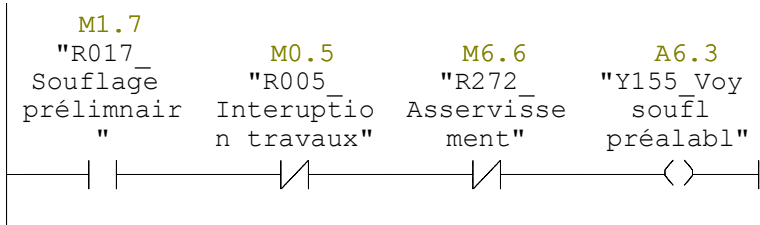
Réseau : 65



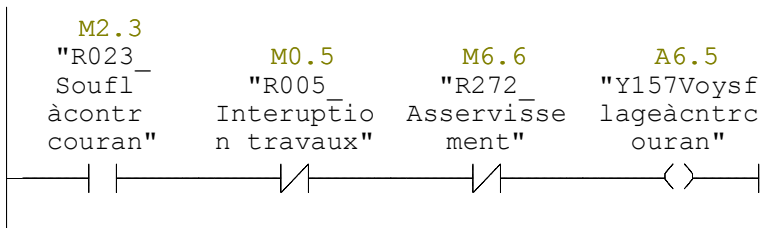
Réseau : 66



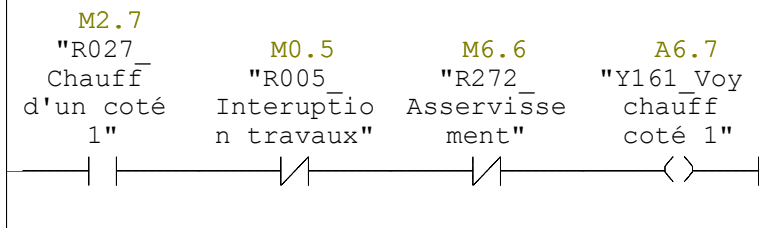
Réseau : 67



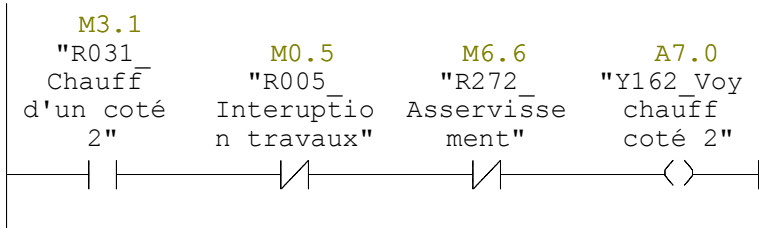
Réseau : 68



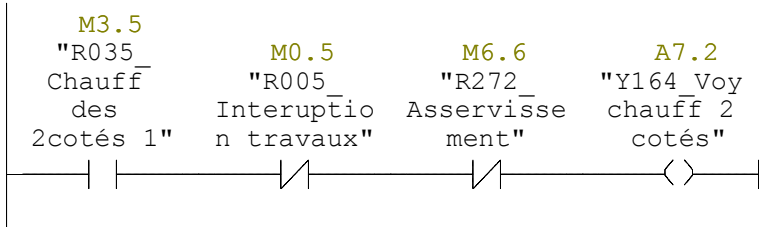
Réseau : 69



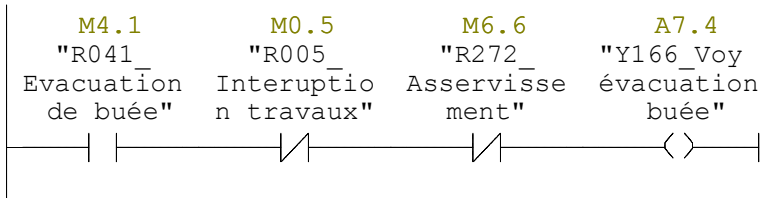
Réseau : 70



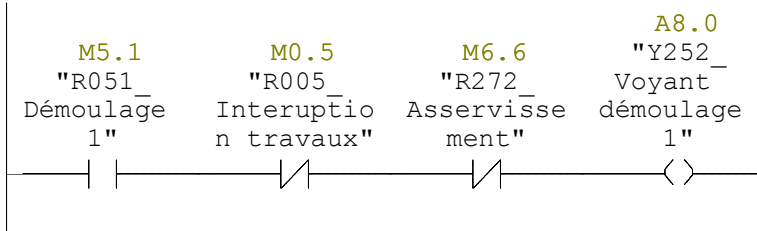
Réseau : 71



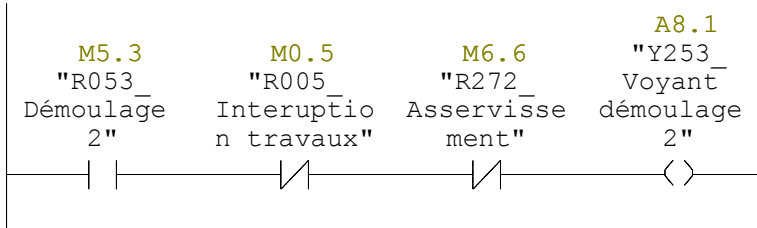
Réseau : 72



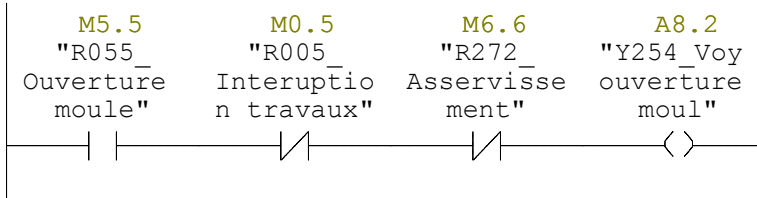
Réseau : 73



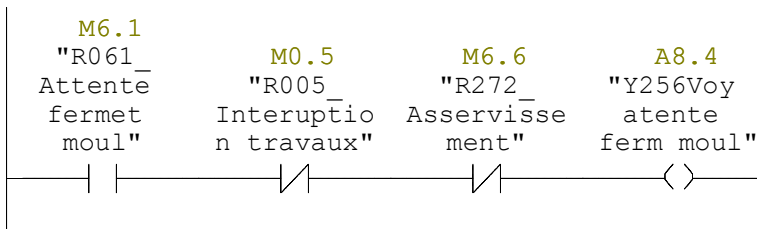
Réseau : 74



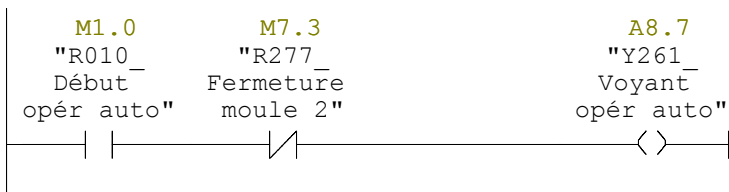
Réseau : 75



Réseau : 76



Réseau : 77



Réseau : 78



Réseau : 79



# *A n n e x e B . .*

Programme permettant la  
simulation du système de  
supervision winCC avec son  
Grafcet

**OB1 - <offline>**

"Cycle Execution"

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 14/09/2016 17:48:17  
**Interface :** 15/02/1996 16:51:12  
**Longueur (bloc/code /données locales) :** 02434 02114 00022

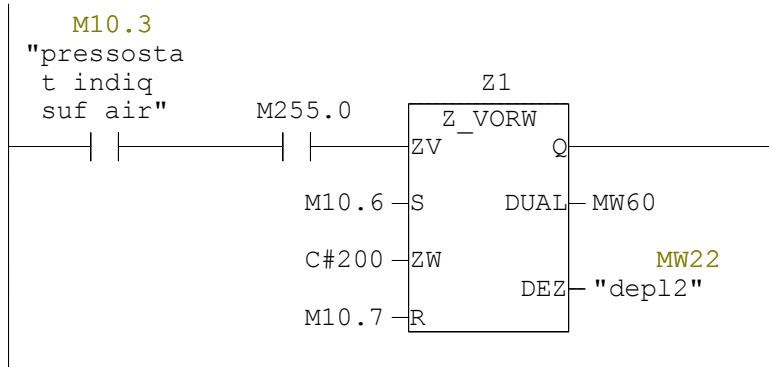
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

**Bloc : OB1 "Main Program Sweep (Cycle)"**

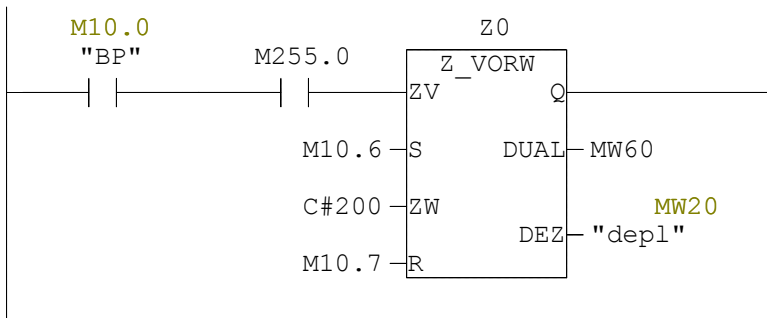
Annexe A: Programme permettant le fonctionnement de la machine ACE 25 AR.

Réseau : 80

Annexe B: Programme permettant la simulation du système de supervision winCC

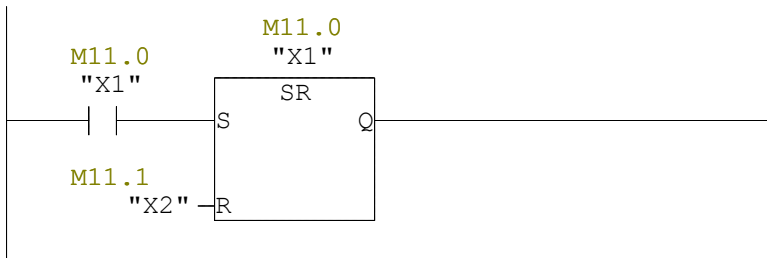


Réseau : 81

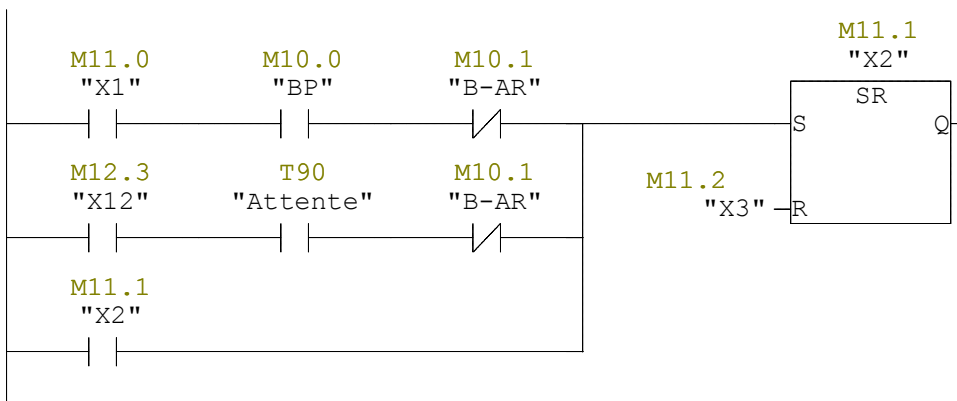


Réseau : 82

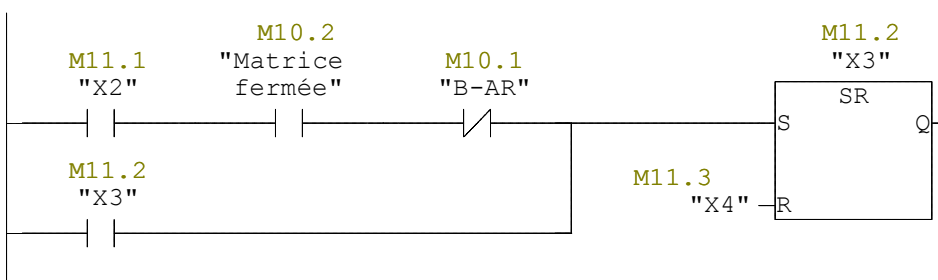
Etape initiale(sans action)



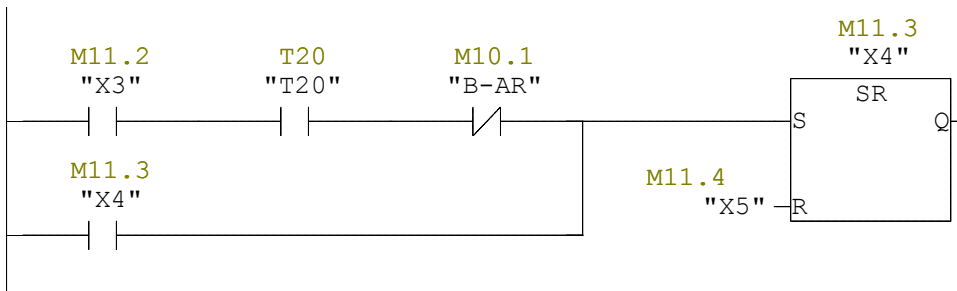
Réseau : 83 Fermeture de la matrice



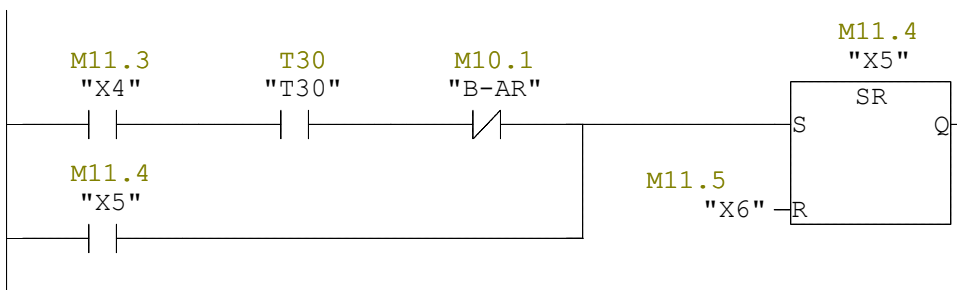
Réseau : 84 Présoufflage



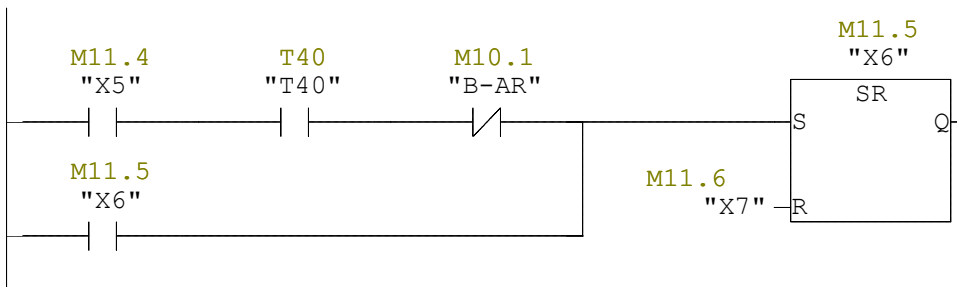
Réseau : 85      Chargement de la matière



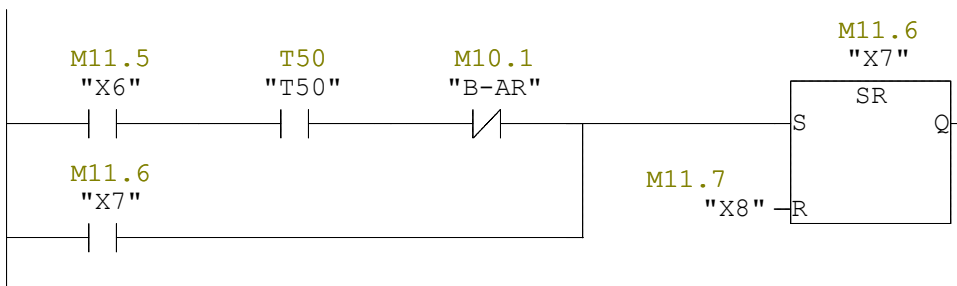
Réseau : 86      Chauffage



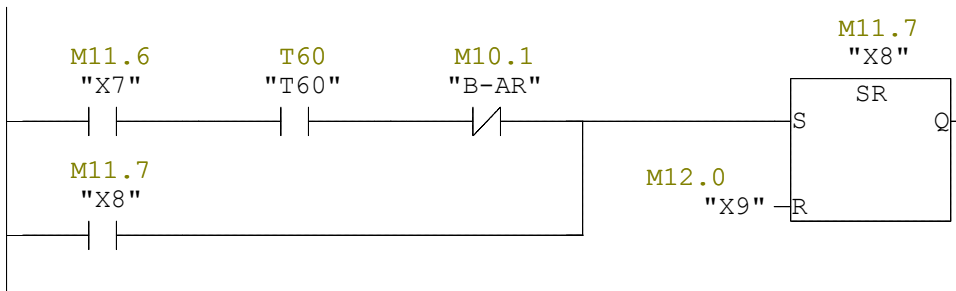
Réseau : 87      Evacuation d'air



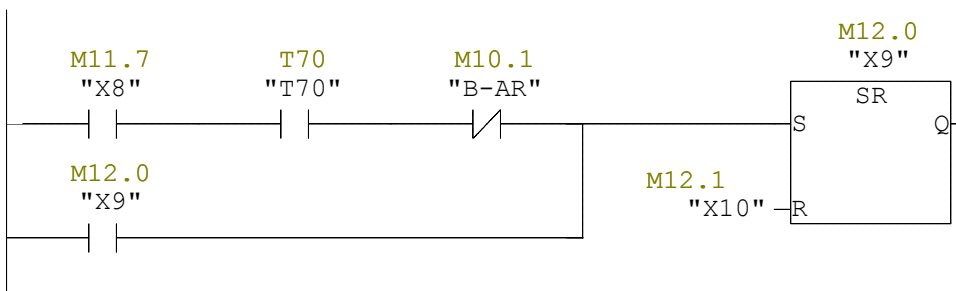
Réseau : 88      Refroidissement à eau



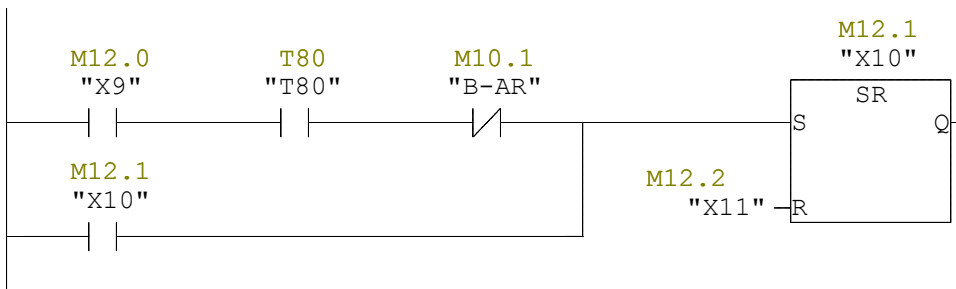
Réseau : 89 Evacuation d'eau et injection d'air d'évacuation commandées par



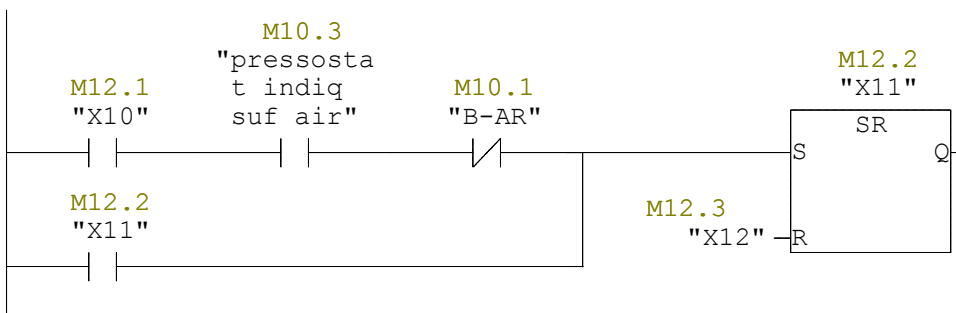
Réseau : 90 Refroidissement par radiation



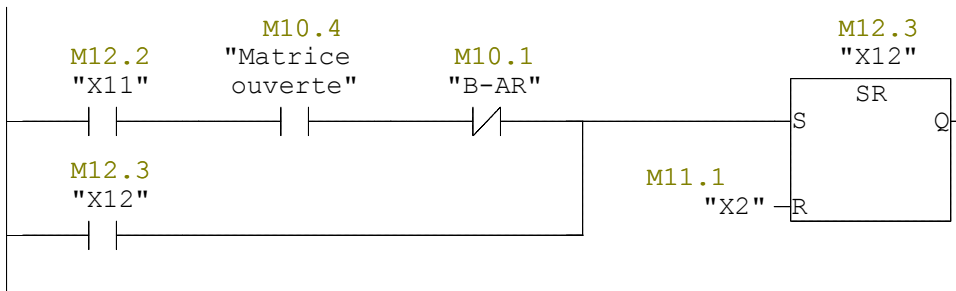
Réseau : 91 Injection d'air



Réseau : 92 Démoulage (ouverture de la matrice) et injection d'air



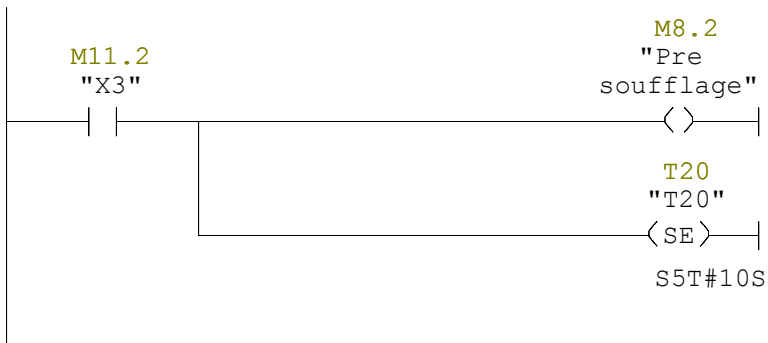
Réseau : 93 Attente



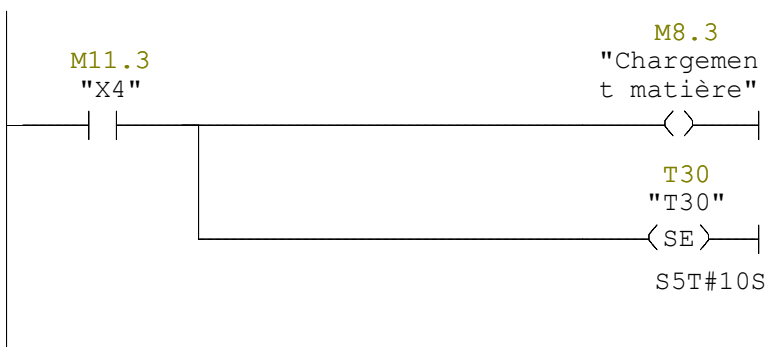
Réseau : 94



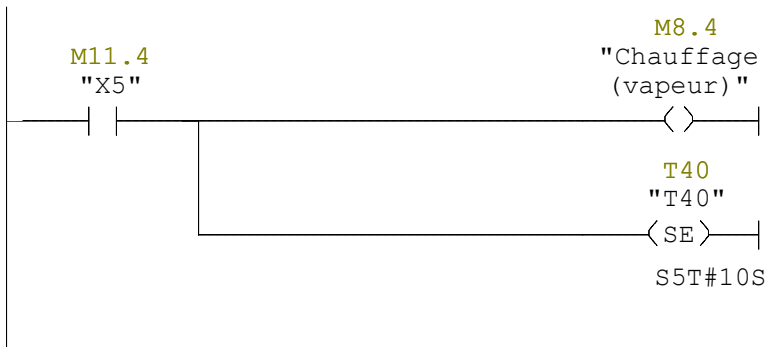
Réseau : 95



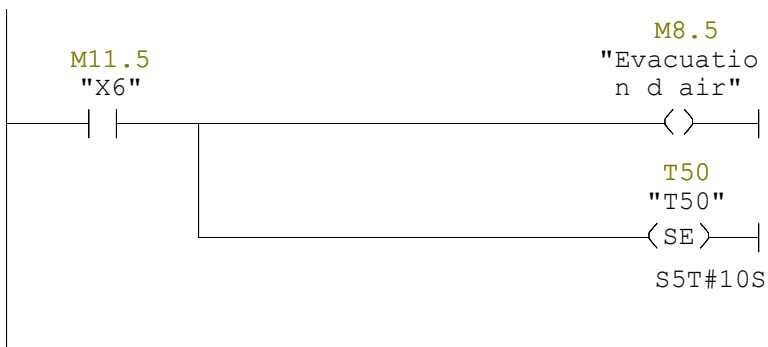
Réseau : 96



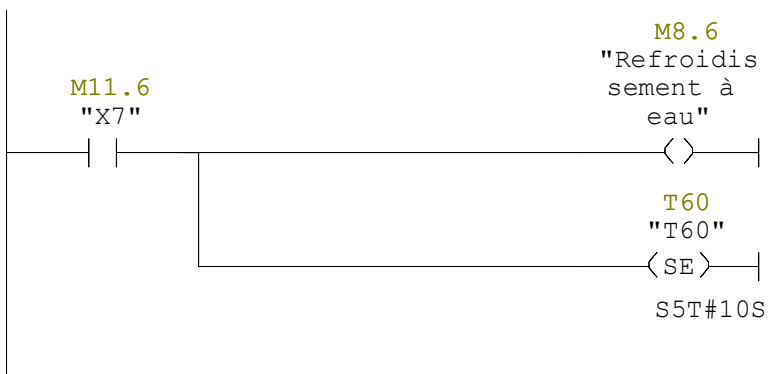
Réseau : 97



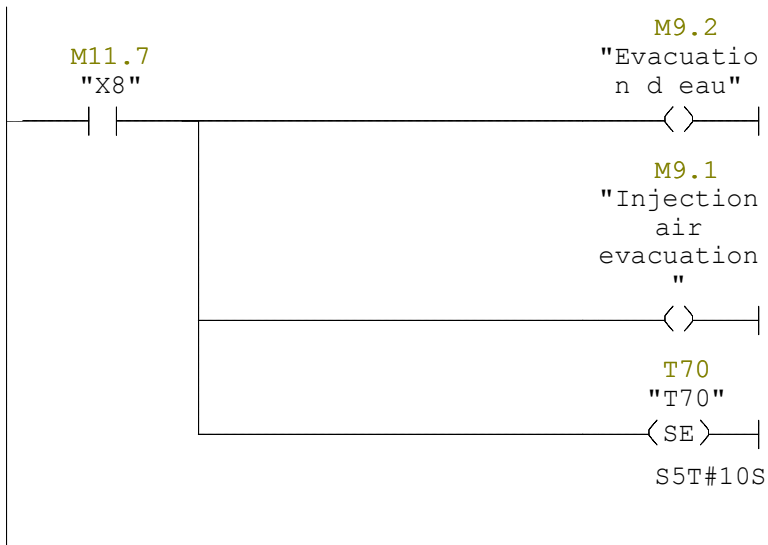
Réseau : 98



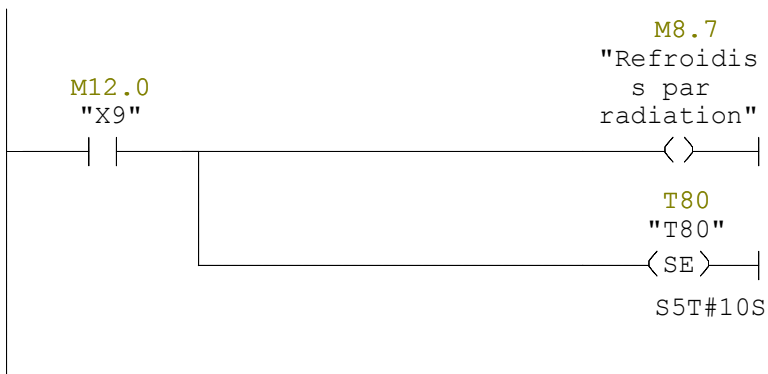
Réseau : 99



Réseau : 100



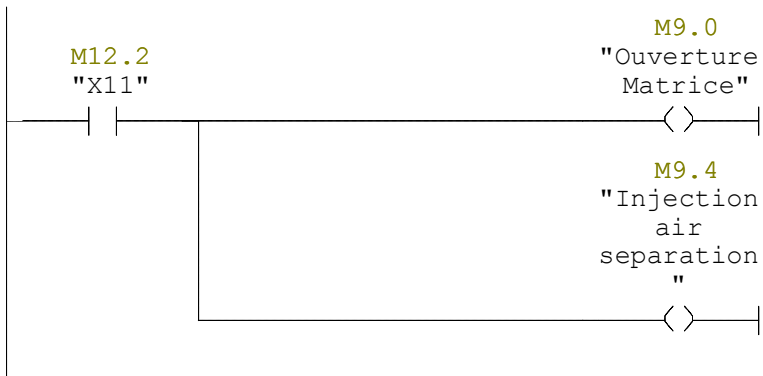
Réseau : 101



Réseau : 102



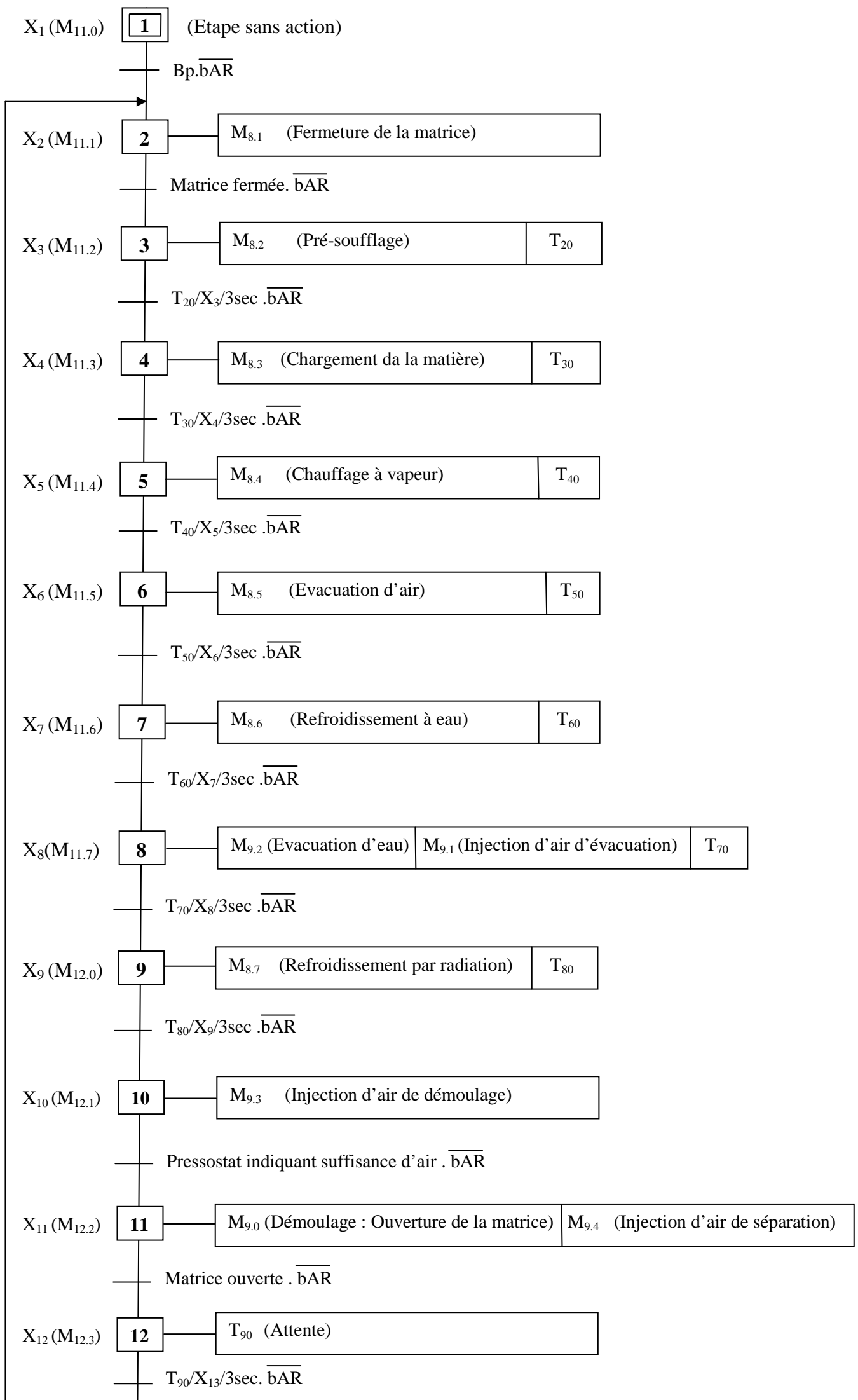
Réseau : 103



Réseau : 104



**Grafcet :**



# **Annexe C.**

## **Table des mnémoniques**

**Propriétés de la table des mnémoniques**

Nom : Mnémoniques  
Auteur :  
Commentaire :  
Date de création : 30/06/2016 18:52:31  
Dernière modification : 14/09/2016 17:44:48  
Dernier filtre sélectionné : Tous les mnémoniques  
Nombre de mnémoniques : 182/182  
Dernier tri : Mnémonique ordre croissant

Etat	Mnémonique	Opérande	Type de données	Commentaire
×				
	Attente	T 90	TIMER	
	B-AR	M 10.1	BOOL	
	BP	M 10.0	BOOL	
	Chargement matière	M 8.3	BOOL	
	Chauffage(vapeur)	M 8.4	BOOL	
	Cycle Execution	OB 1	OB 1	
	depl	MW 20	WORD	
	depl2	MW 22	WORD	
	Evacuation d air	M 8.5	BOOL	
	Evacuation d eau	M 9.2	BOOL	
	Fermeture Matrice	M 8.1	BOOL	
	Injection air demoulage	M 9.3	BOOL	
	Injection air evacuation	M 9.1	BOOL	
	Injection air separation	M 9.4	BOOL	
	Matrice fermée	M 10.2	BOOL	
	Matrice ouverte	M 10.4	BOOL	
	MR	A 9.2	BOOL	
	MS	A 9.1	BOOL	
	Ouverture Matrice	M 9.0	BOOL	
	Pre soufflage	M 8.2	BOOL	
	pressostat indiq suf air	M 10.3	BOOL	
	R000_Auto	M 0.0	BOOL	
	R001_Manuel	M 0.1	BOOL	
	R002_Semi-auto	M 0.2	BOOL	
	R003_Maintient oper auto	M 0.3	BOOL	
	R004_Réarmement	M 0.4	BOOL	
	R005_Interruption travaux	M 0.5	BOOL	
	R006_Levée interruption	M 0.6	BOOL	
	R007	M 0.7	BOOL	
	R010_Début opér auto	M 1.0	BOOL	
	R011_Préchauffage	M 1.1	BOOL	
	R012	M 1.2	BOOL	
	R013_Fermeture moule 1	M 1.3	BOOL	
	R014	M 1.4	BOOL	
	R015_Permutati préchauff	M 1.5	BOOL	
	R016	M 1.6	BOOL	
	R017_Soufflage prélimnair	M 1.7	BOOL	
	R020	M 2.0	BOOL	
	R021_Coulée	M 2.1	BOOL	
	R022	M 2.2	BOOL	
	R023_Soufl àcontr couran	M 2.3	BOOL	
	R024	M 2.4	BOOL	
	R025_Chauffage moule	M 2.5	BOOL	
	R026	M 2.6	BOOL	
	R027_Chauff d'un coté 1	M 2.7	BOOL	
	R030	M 3.0	BOOL	
	R031_Chauff d'un coté 2	M 3.1	BOOL	
	R032	M 3.2	BOOL	
	R033_Chauff autre coté	M 3.3	BOOL	
	R034	M 3.4	BOOL	
	R035_Chauff des 2cotés 1	M 3.5	BOOL	
	R036	M 3.6	BOOL	
	R037_Chauff des 2cotés 2	M 3.7	BOOL	
	R040	M 4.0	BOOL	
	R041_Evacuation de buée	M 4.1	BOOL	
	R042	M 4.2	BOOL	
	R043_Refroidissmnt à eau	M 4.3	BOOL	
	R044	M 4.4	BOOL	
	R045_Evacuation d'eau	M 4.5	BOOL	

Etat	Mnémonique	Opérande	Type de données	Commentaire
	R046	M 4.6	BOOL	
	R047_Radiation	M 4.7	BOOL	
	R050	M 5.0	BOOL	
	R051_Démoulage 1	M 5.1	BOOL	
	R052	M 5.2	BOOL	
	R053_Démoulage 2	M 5.3	BOOL	
	R054	M 5.4	BOOL	
	R055_Ouverture moule	M 5.5	BOOL	
	R056	M 5.6	BOOL	
	R057_Ejection	M 5.7	BOOL	
	R060	M 6.0	BOOL	
	R061_Attente fermet moul	M 6.1	BOOL	
	R062_Achèvement	M 6.2	BOOL	
	R063_Auto	M 6.3	BOOL	
	R064_Semi-auto	M 6.4	BOOL	
	R065_Fin compte à rebour	M 6.5	BOOL	
	R272_Asservissement	M 6.6	BOOL	
	R273	M 6.7	BOOL	
	R274	M 7.0	BOOL	
	R275_LS-3	M 7.1	BOOL	
	R276_Permut chauff unilat	M 7.2	BOOL	
	R277_Fermeture moule 2	M 7.3	BOOL	
	Refroidiss par radiation	M 8.7	BOOL	
	Refroidissement à eau	M 8.6	BOOL	
	T000_0020	T 0	TIMER	
	T002_0020	T 1	TIMER	
	T004_0020	T 2	TIMER	
	T006_0020	T 3	TIMER	
	T010_0040	T 4	TIMER	
	T012_0010	T 5	TIMER	
	T014_0010	T 6	TIMER	
	T016_0020	T 7	TIMER	
	T020_0010	T 8	TIMER	
	T20	T 20	TIMER	
	T30	T 30	TIMER	
	T372_0020	T 9	TIMER	
	T374_0010	T 10	TIMER	
	T376_0001	T 11	TIMER	
	T40	T 40	TIMER	
	T50	T 50	TIMER	
	T60	T 60	TIMER	
	T70	T 70	TIMER	
	T80	T 80	TIMER	
	X000_LS1_Ralentissement	E 0.0	BOOL	
	X001_LS2_Fissuration	E 0.1	BOOL	
	X002_LS3_Ejection	E 0.2	BOOL	
	X003_LS4_Fin ouvrturmoul	E 0.3	BOOL	
	X004_PS-F	E 0.4	BOOL	
	X005_PS-M1	E 0.5	BOOL	
	X006_PS-M2	E 0.6	BOOL	
	X007_TR1_Coulage matière	E 0.7	BOOL	
	X1	M 11.0	BOOL	
	X10	M 12.1	BOOL	
	X100_TR2_Chauffage moule	E 1.0	BOOL	
	X101_TR3_Chautr côté	E 1.1	BOOL	
	X102_TR4_Chautr des2côté	E 1.2	BOOL	
	X103_TR5_Refroidissà eau	E 1.3	BOOL	
	X104_TR6_Radiation	E 1.4	BOOL	
	X105_SW1_O/F moulà v réd	E 1.5	BOOL	
	X106_SW2Comutàchaufd1cté	E 1.6	BOOL	
	X107_PB 3_Réarmement	E 1.7	BOOL	
	X11	M 12.2	BOOL	
	X12	M 12.3	BOOL	
	X2	M 11.1	BOOL	
	X200_Switch 3_Auto	E 2.0	BOOL	
	X201_Switch 3_Manuel	E 2.1	BOOL	
	X202_PB 4_Opération auto	E 2.2	BOOL	
	X203_PB 5_Préchauffage	E 2.3	BOOL	
	X204_PB 6_Fermeture moul	E 2.4	BOOL	

Etat	Mnémonique	Opérande	Type de données	Commentaire
	X205_PB 7_Ouverture mou	E 2.5	BOOL	
	X206_CR2_Démarrage	E 2.6	BOOL	
	X207_CTR_Compteur	E 2.7	BOOL	
	X3	M 11.2	BOOL	
	X300_PS1_Verrouillage	E 3.0	BOOL	
	X301_SS16_Chang p à exp	E 3.1	BOOL	
	X4	M 11.3	BOOL	
	X5	M 11.4	BOOL	
	X6	M 11.5	BOOL	
	X7	M 11.6	BOOL	
	X8	M 11.7	BOOL	
	X9	M 12.0	BOOL	
	Y050_moteur hydraulique	A 4.0	BOOL	
	Y051Robinet fermeturmoul	A 4.1	BOOL	
	Y052_Robinetptitevitesse	A 4.2	BOOL	
	Y053_Rob mise horspress	A 4.3	BOOL	
	Y054_Robinetdifferentiel	A 4.4	BOOL	
	Y055_Rob ouverture moule	A 4.5	BOOL	
	Y056_Robinet remplissage	A 4.6	BOOL	
	Y057_Robinet obturation	A 4.7	BOOL	
	Y060_Rob pneum rempliss	A 5.0	BOOL	
	Y061_Robi vapeur gauche	A 5.1	BOOL	
	Y062_Robin vapeur droit	A 5.2	BOOL	
	Y063_Robin purge gauche	A 5.3	BOOL	
	Y064_Robinet purge droit	A 5.4	BOOL	
	Y065_Robi d'eau refroidi	A 5.5	BOOL	
	Y066_Robi évacuation eau	A 5.6	BOOL	
	Y067_Robi air démoulage	A 5.7	BOOL	
	Y150_Robi air d'éjection	A 6.0	BOOL	
	Y153_Transporteur	A 6.1	BOOL	
	Y154_Voy fermtur moulé2	A 6.2	BOOL	
	Y155_Voy soufl préalable	A 6.3	BOOL	
	Y156_Minut/Voy coulée	A 6.4	BOOL	
	Y157Voysflageàcntrcouran	A 6.5	BOOL	
	Y160Minut/Voy chauff mou	A 6.6	BOOL	
	Y161_Voy chauff coté 1	A 6.7	BOOL	
	Y162_Voy chauff coté 2	A 7.0	BOOL	
	Y163_Minut/Voychauf1coté	A 7.1	BOOL	
	Y164_Voy chauff 2 cotés	A 7.2	BOOL	
	Y165Min/Voy chauff 2cotés	A 7.3	BOOL	
	Y166_Voy évacuation buée	A 7.4	BOOL	
	Y167Min refroi voyrefroi	A 7.5	BOOL	
	Y250_Voy évacuation eau	A 7.6	BOOL	
	Y251_Minut/Voy radiation	A 7.7	BOOL	
	Y252_Voyant démoulage 1	A 8.0	BOOL	
	Y253_Voyant démoulage 2	A 8.1	BOOL	
	Y254_Voy ouverture mou	A 8.2	BOOL	
	Y255_Voy d'air éjection	A 8.3	BOOL	
	Y256Voy atente ferm mou	A 8.4	BOOL	
	Y257_Compteur	A 8.5	BOOL	
	Y260_Relais mode manuel	A 8.6	BOOL	
	Y261_Voyant opér auto	A 8.7	BOOL	
	Y262_Voyant préchauffage	A 9.0	BOOL	