

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERRI de Tizi-Ouzou**



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

**PROJET DE FIN D'ETUDES**

En vue de l'obtention du diplôme

**D'INGENIEUR D'ETAT EN AUTOMATIQUE**

**Thème**

Etude et automatisation d'une cisaille à guillotine  
à l'aide de l'API S7-300

**Proposé par :**

M<sup>r</sup> M .CHALLAL

**Dirigé par :**

M<sup>lle</sup> O. CHILALI

**Présenté par :**

M<sup>r</sup> AMAOUZ    Rahim

M<sup>r</sup> KECHAD    Hachimi

**Soutenu le :** 09 / 07 /2009    Devant le jury d'examen composé de :

**M<sup>r</sup>. M .HASNAOUI**

**M<sup>r</sup>. A. MAIDI**

**M<sup>lle</sup>. O.ADJEMOUT**

**Promotion**

**AU NOM DE DIEU LE CLEMENT ET LE  
MISERECORDIEUX**

**REMERCIEMENTS**

Nous remercions en premier lieu le Dieu qui nous a bénéficié d'une volonté suffisante pour accomplir ce modeste travail et tous ceux qui nous ont aidé de près ou de loin.

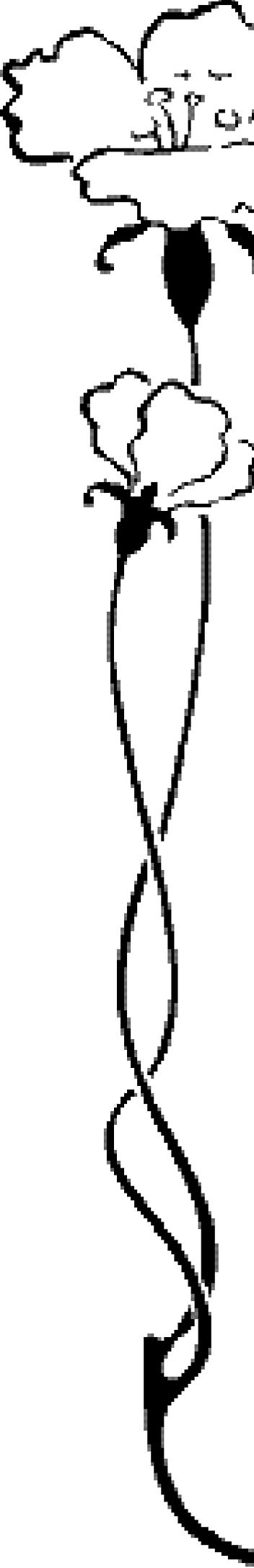
Nos remerciements vont, en premier lieu, à notre promotrice Mlle O.CHILALI qui a accepté de nous encadrer, pour ses conseils et sa présence à tout moment.

Un grand merci à Mr M.CHALLAL qui nous a proposé le sujet. Ainsi pour avoir mis à notre disposition la documentation.

Nous remercions le personnel de L'ENIEM.

Mille mercis à Mr S.AMAOUZ et Mr A.BELAHCEN qui nous ont permis d'avoir ce stage.

Nous exprimons aussi nos remerciements aux membres de jury qui ont accepté de juger notre travail.



## Dédicaces

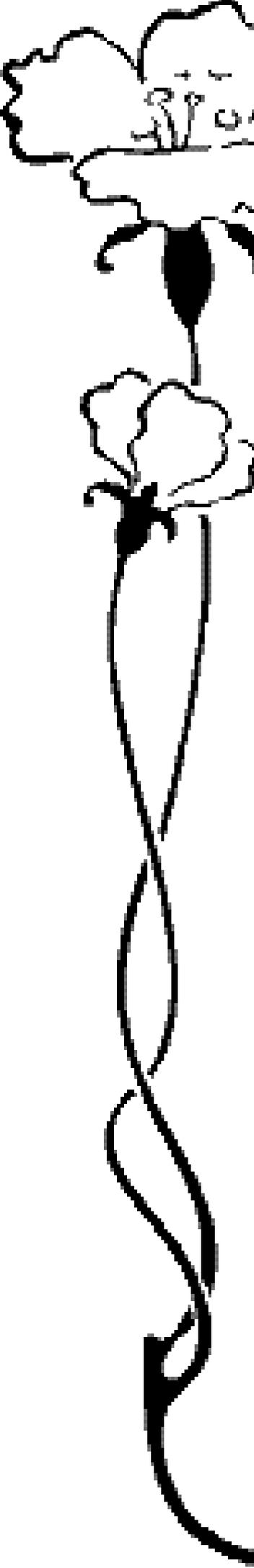
Je dédie ce modeste travail :  
A mes très chers parents qui  
ont su être toujours présent à  
mes cotés, tout au long de ma  
vie ;

A mes très chers frères, et à  
ma belle-sœur.

A « Inès », qui vient de voir le jour  
A toute ma famille (cousins et  
cousines) en particulier ma  
chère tante « dadi » et sa  
fille « kenza ».

A mes amis, tous ceux qui se  
reconnaîtront en ce mot  
« AMI ».

A mon binôme et sa famille.



Je dédie ce modeste travail :  
A mes très chers parents qui  
ont su être toujours présent à  
mes cotés, tout au long de ma  
vie pour me soutenir.  
A mes très chers frères.  
Et à mes deux très chères  
sœurs Farida, Assia que j'aime  
beaucoup.

A mes amis que j'admire  
beaucoup.  
A mon binôme ainsi toute sa  
famille.

Et à tous ceux qui se  
reconnaîtront en ce mot  
« AMI »...

**Hachimi.**

## SOMMAIRE

<b><u>Préambule</u></b> .....	1
-------------------------------	---

<b><u>Introduction générale</u></b> .....	5
---	---

### **Chapitre I : étude et description de l'installation**

1. Introduction .....	7
2. Description de l'installation.....	8
3. La partie opérative .....	9
3.1 Structure mécanique .....	10
3.2 Les prés actionneurs .....	11
3.3 Les actionneurs .....	13
3.3.1 Actionneurs utilisant l'énergie pneumatique ou hydraulique .....	13
3.3.2 Actionneurs utilisant l'énergie électrique. ....	14
3.4 Les capteurs .....	15
3.4.1 Capteurs de fin de course (fc) mécaniques .....	16
3.4.2 Les capteurs optiques .....	16
3.4.3 Capteurs de pression (pressostat) .....	16
3.4.4 Capteur rotatif .....	16
3.4.5 Temporisateurs (minuteurs) .....	17
4. Partie commande .....	17
5. Circuit électrique.....	20
6. Modes de fonctionnement .....	21
6.1 Cycle manuel .....	21
6.2 Cycle semi automatique .....	21
6.3 Cycle automatique .....	21
7. Fonctionnement de l'installation.....	21
8. Conclusion .....	22

### **Chapitre II : Modélisation de l'installation à l'aide du grafcet**

1. Introduction.....	23
2. Généralité sur le GRAFCET .....	23
2.1 Les règles d'évolution du GRAFCET .....	24
2.2 Divergence et convergence en ET (séquences simultanées) .....	25
2.3 Divergence et convergence en OU (aiguillage).....	25
2.4 Saut en avant (saut de phase) .....	26
2.5 Saut en arrière (reprise de phase) .....	27
2.6 Niveau d'un GRAFCET .....	27

2.7 Mise en équation d'un GRAFCET .....	28
3. Améliorations apportées à l'installation .....	29
3.1 Temporisation .....	29
3.2 Capteur.....	30
3.2.1 <i>Choix d'un capteur</i> .....	30
4. Modélisation de l'installation.....	31
4.1 Grafcet de niveau 1 .....	31
4.1.1Grafcet de gestion d'arrêt d'urgence .....	31
4.1.2Grafcet de la pompe .....	31
4.1.3 Grafcet de l'installation .....	32
4.2 Grafcet de niveau 2 .....	39
4.2.1Grafcet de gestion d'arrêt d'urgence .....	39
4.2.2Grafcet de la pompe .....	39
4.2.3 Grafcet de l'installation .....	40
5.Conclusion.....	46

### Chapitre III : Programmation du nouveau système automatisé à l'aide du S7-300

1. Introduction.....	47
2. Automate programmable .....	47
2.1. Définition.....	47
2.2. Architecture et fonctionnement d'un automate .....	47
2.3. Choix d'un automate .....	49
3. l'automate S7-300 .....	49
3-1- constitution du S7-300.....	49
3.1.1. Le module d'alimentation .....	50
3.1.2. Description de la CPU .....	50
3.1.3. Modules de coupleurs (IM) .....	51
3.1.4. Modules de signaux (SM) .....	51
3.1.5. Module de fonction (FM) .....	51
3.1.6. Modules de communication .....	52
3.1.7. Châssis d'extension (UR) .....	52
3.2 Fonctionnement de l'automate programmable S7-300.....	52
3.3 Programmation de l'automate S7-300 .....	53
3.3.1. Langage de programmation .....	54
3.3.2. Création d'un programme utilisateur .....	55
3.3.3. Configuration du matériel de l'automate .....	59
3.3.4. Création de la table des mnémoniques .....	60
3.3.5 Adressage des signaux d'entrées/sorties .....	60
3.3.6 Mémentos.....	61
3.3.7 Traitement du programme par l'automate .....	61
3.4 Principe de conception d'une structure de programme .....	61
3.4.1 Système d'exploitation .....	61
3.4.2 Programme utilisateur .....	62
3.5 Blocs dans le programme utilisateur.....	62
3.5.1. Bloc utilisateur .....	62
3.5.2 Blocs système pour fonctions standard et fonction système .....	63

<b>3.6 Traitement du programme par la CPU .....</b>	<b>64</b>
<b>3.6.1 Traitement linéaire du programme .....</b>	<b>65</b>
<b>3.6.2 Traitement structuré du programme .....</b>	<b>65</b>
<b>3.7 Différents blocs d'organisation des S7-300 .....</b>	<b>65</b>
<b>3.8 Implantation du GRAFCET dans le S7-300 .....</b>	<b>67</b>
<b>4 Conclusion.....</b>	<b>72</b>

**Chapitre IV : Simulation avec S7-PLCSIM**

<b>1. Introduction .....</b>	<b>73</b>
<b>2. Logiciel de simulation S7-PLCSIM .....</b>	<b>73</b>
<b>2.1 Mise en route du logiciel S7-PLCSIM .....</b>	<b>73</b>
<b>3. Visualisation de l'état du programme .....</b>	<b>77</b>
<b>4. Simulation du programme de la machine réadaptée .....</b>	<b>78</b>
<b>5. Conclusion.....</b>	<b>82</b>

<b><u>Conclusion générale</u> .....</b>	<b>84</b>
---	-----------

<b><u>Annexe</u> .....</b>	
----------------------------	--

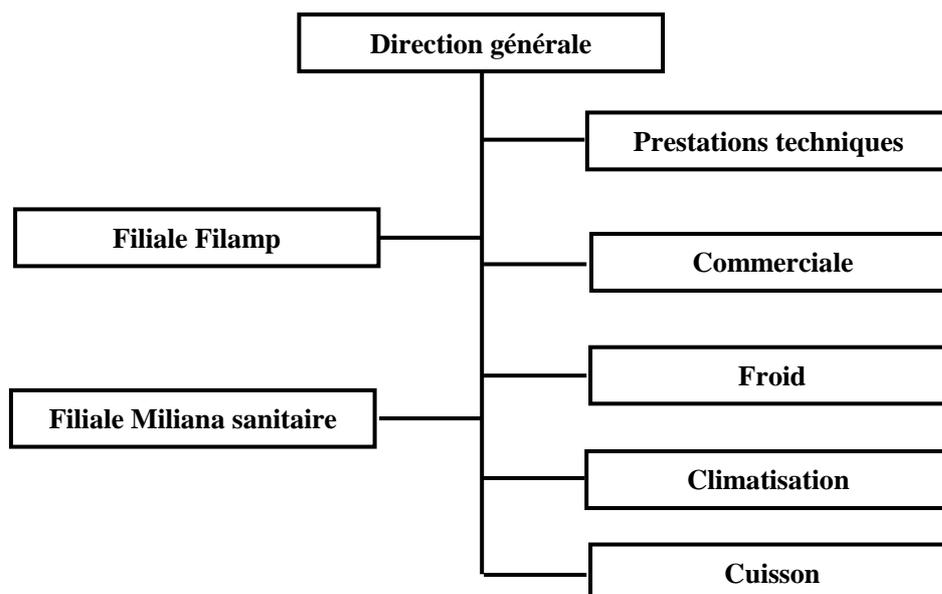
<b><u>Bibliographie</u>.....</b>	
----------------------------------	--

## 1. Historique et présentation

Depuis son implantation l'ENIEM n'a cessé de chercher à automatiser ses différentes installations pour concurrencer les leaders de l'électroménager. Dès ses débuts dans l'industrie, elle a signé des contrats de travail avec des firmes internationales en vue d'améliorer la qualité de ses produits et de former une classe travailleuse qualifiée. C'est en 1987, qu'elle a débutée l'expérience des installations automatisées, grâce à TOSHIBA, après la mise en oeuvre de la chaîne R1 (chaîne de production des réfrigérateurs à l'unité Froid) entièrement automatique. C'est dans la même année qu'elle s'est équipée d'une nouvelle installation automatique pilotée par automate SIEMENS.

L'entreprise Nationale des Industries de l'Electroménagers (ENIEM) est une entreprise publique issue de la restructuration de la SONELEC. Elle est structurée en (**Figure P1**) :

- ✓ une direction générale : sise au chef lieu de la wilaya de Tizi-ouzou. Elle est responsable de la stratégie et du développement ;
- ✓ deux filiales :-Miliana (sanitaire) fabrication de matériels sanitaires ;  
-Filamp Mohamadia fabrication des lampes d'éclairage ;
- ✓ cinq unités : implantées à la zone industrielle Aissat Idir (Oued Aissi) : prestations techniques, commerciale et trois de productions : froid, climatisation et cuisson.



**Figure P1** : organisation de l'ENIEM.

## 2. Présentation et organisation de l'unité cuisson

L'unité cuisson est spécialisée dans la production des différents types de cuisinières. Elle est organisée (**Figure P 2**) en une direction comportant : un secrétaire, contrôleur de gestion, assistant qualité environnement, un assistant sécurité industrielle, trois départements et trois services en staff.

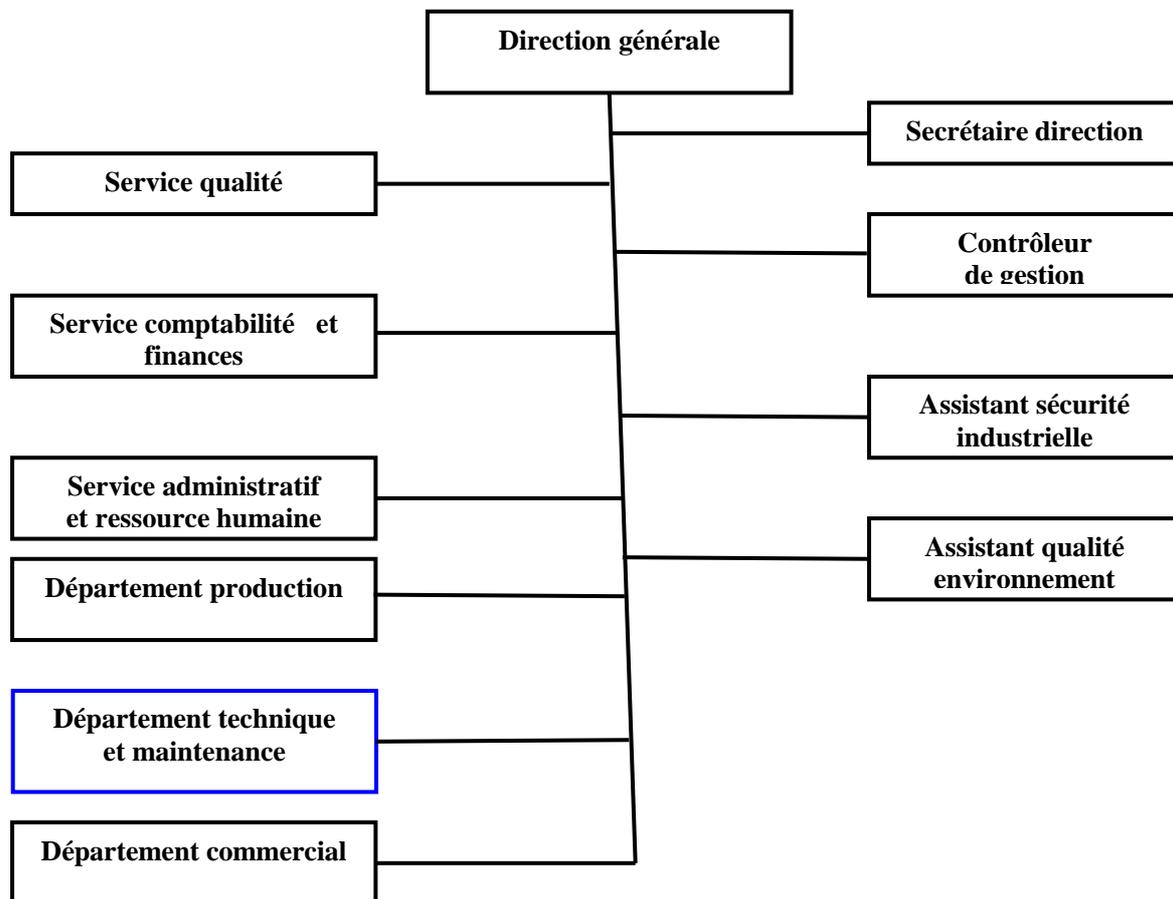


Figure P2 : organigramme de l'unité cuisson

### 2.1 Les départements de l'unité cuisson

#### 2.1.1 Département technique et maintenance

Il soutient tous les départements et services dans l'accomplissement de leurs tâches. Il est structuré comme suit :

- Secrétariat de département.
- Service développement du produit et méthode de fabrication.
- Service maintenance industrielle (où nous avons mené notre stage).

### **2.1.2 Département commercial**

Son rôle est l'approvisionnement en matières premières, composants et matières auxiliaires des différentes structures de production. Il est structuré comme suit :

- Service transit et douanes.
- Service achat.
- Service gestion des stocks.

### **2.1.2 Département production**

Sa mission est de transformer les matières premières en produits finis. Il est constitué de :

- Service ordonnancement production.
- Service fabrication tôlerie.
- Service fabrication mécanique.
- Service traitement et revêtement surface.
- Service montage final.

## **2.2 Les services de l'unité cuisson**

### **2.2.2 Service qualité**

Il a trois tâches essentielles :

- Inspection matières premières en prélevant des échantillons.
- Inspection des produits.
- Elaboration des gammes de productions.

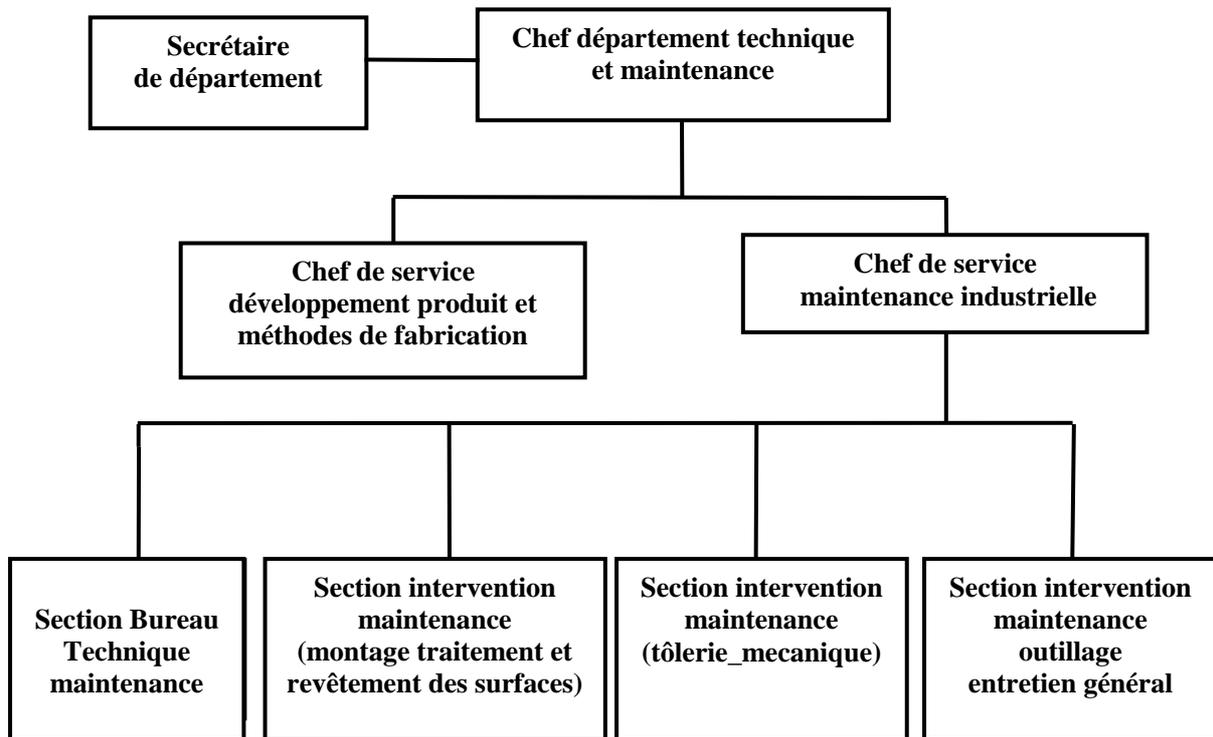
### **2.2.2 Service finances et comptabilités**

Ce service a deux tâches principales :

- Gestion et suivi des ressources financières de l'unité.
- La comptabilité de l'unité.

### **2.2.2 Service administratif et ressources humaines**

Il fait le suivi des ressources humaines de l'unité et applique toutes les procédures de gestion relatives à ces tâches : recrutement, mutation, promotion, etc.



**Figure P.3** organigramme du service maintenance.



# Introduction générale

Aujourd'hui, au XXI<sup>e</sup> siècle, les automatismes sont légion autour de nous, rien que dans nos logements : les machines à laver le linge, la vaisselle, le réfrigérateur à dégivrage automatique, etc. Comportent au moins un automatisme. Dans l'industrie, ils sont indispensables : ils effectuent quotidiennement les tâches les plus pénible, répétitives et dangereuses. Parfois, ces automatismes sont d'une telle rapidité et d'une telle précision, qu'ils réalisent des actions impossibles pour un être humain. L'automatisme est donc aussi synonyme de productivité et de sécurité.

L'automatique, en tant que technique, a évolué au cours des dernières décennies vers l'étude et la maîtrise de systèmes, de plus en plus complexes, permettant l'exécution et le contrôle de tâches techniques par des machines fonctionnant sans intervention humaine, ou au moins d'une intervention réduite.

Grâce à l'avènement des API (Automates Programmables Industriels), des machines à commande numérique sont apparues. Les API ont été peu courants lors de leur introduction, ces dernières décennies, ce sont des millions d'API qui ont vu le jour. Isolés ou en réseau, ils peuvent accomplir des tâches complexes : non seulement de contrôle, mais aussi de pilotage de machines, de traitement de données, de circulation de l'information et de simulation. Ils sont utilisés à tous les stades du processus productif (conception, production, contrôle de la qualité des produit, etc.).

Aujourd'hui, en vu de se mettre au diapason des avancées technologiques enregistrées dans l'industrie automatique et afin d'être conquérante, l'ENIEM axe sa politique de qualité sur l'amélioration de ses produits et ses services afin d'obtenir des produits à qualité uniforme pour faire face à une concurrence rude des firmes internationales d'électroménager.

La compétitivité des entreprises exige une automatisation de plus en plus flexible et évolutive des équipements de production ; ce qui explique la rude concurrence entre les firmes internationales (Schneider, Siemens, Bosch, etc.) .

Actuellement, l'ENIEM se voit confronter au manque des cartes d'entrées/sorties (électroniques) des séquenceurs qu'elle utilise. Ces cartes tombent souvent en pannes et de plus elles ne sont plus disponibles sur le marché. Malgré les efforts des ingénieurs de les réparer, cela reste insuffisant vu le rythme de la production. Ce qui contraint l'entreprise à réfléchir d'hors et déjà à remplacer ces cartes électroniques.

Dans ce contexte le sujet qui nous a été proposé par le Département Technique et Maintenance de l'unité cuisson, consiste à faire l'étude et la modélisation d'une cisaille à guillotine en utilisant comme outil de modélisation le GRAFCET en vue de proposer une solution à l'automatisation moyennant l'automate programmable Siemens.

Ainsi, notre mémoire est organisé comme suit :

- Le premier chapitre est consacré à l'étude et la description de l'installation.
- Dans le deuxième chapitre, nous nous sommes intéressés à la modélisation de l'installation à l'aide du GRAFCET.
- Le troisième chapitre est consacré à la présentation de l'automate programmable S7-300 et son langage de programmation en vue de l'automatisation de l'installation.
- Le dernier chapitre est réservé à la simulation du programme avec le logiciel S7-PLCSIM.
- Enfin, Nous terminons notre travail par une conclusion générale.

# Chapitre I

***Etude  
et description  
de l'installation***

1. Introduction

L'installation étudiée est semi automatique. Elle sert à découper différentes longueurs de tôles. Elle occupe une place très importante dans la chaîne de production des cuisinières.

Les figures I.1 et I.2 illustrent les différents parties de l'installation.

Dans ce chapitre, nous décrivons son fonctionnement et les éléments qui la composent.

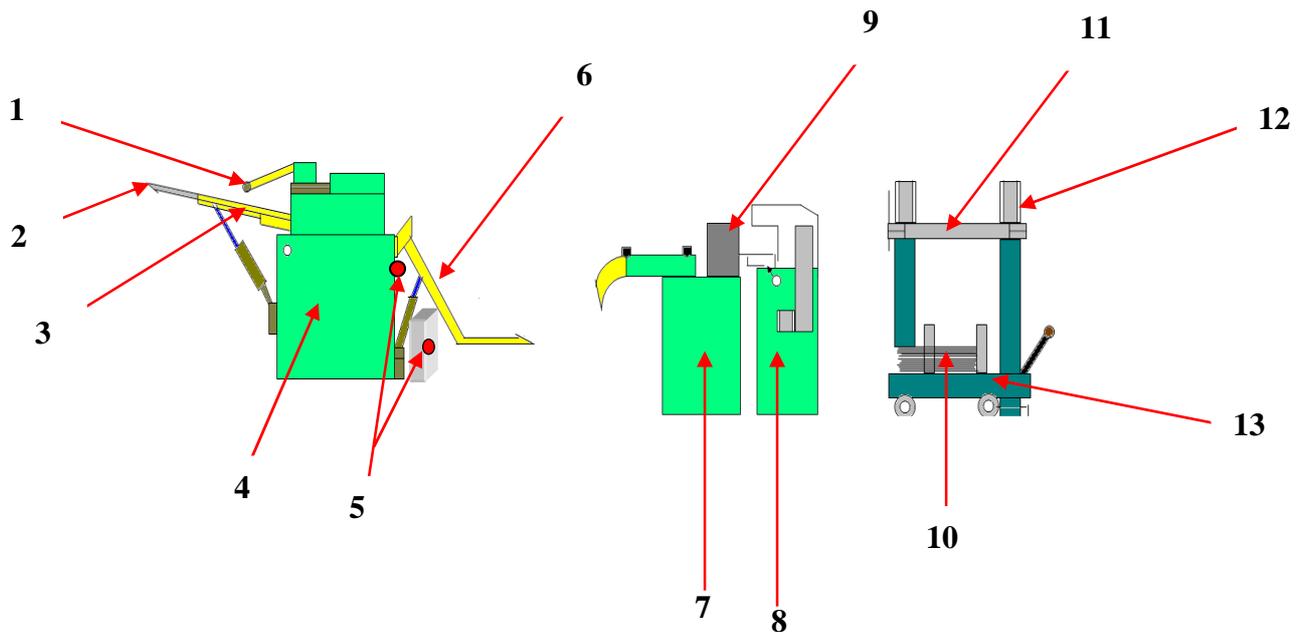


Figure I. 1 : illustrant le redresseur, l'aménage et la cisaille

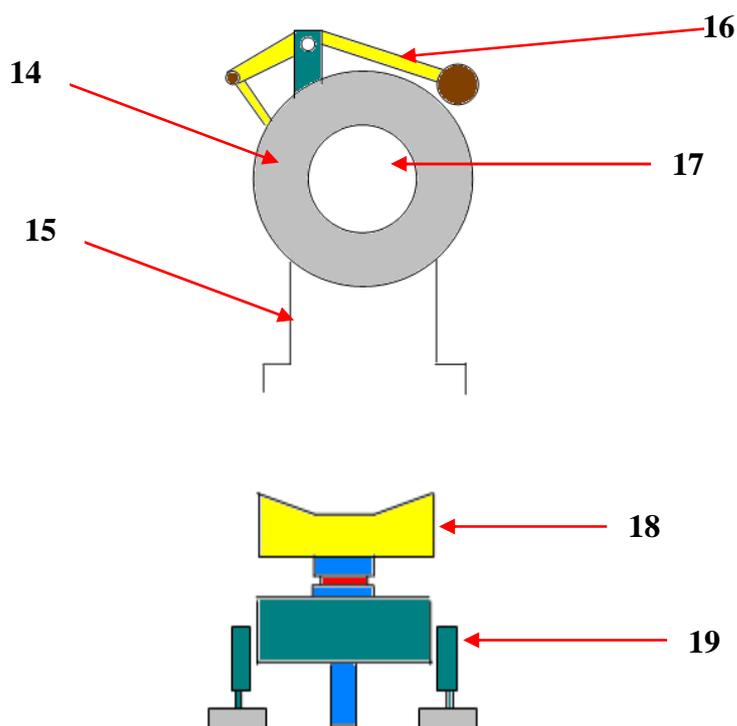


Figure I.2 : illustrant le dévidoir et le breceau

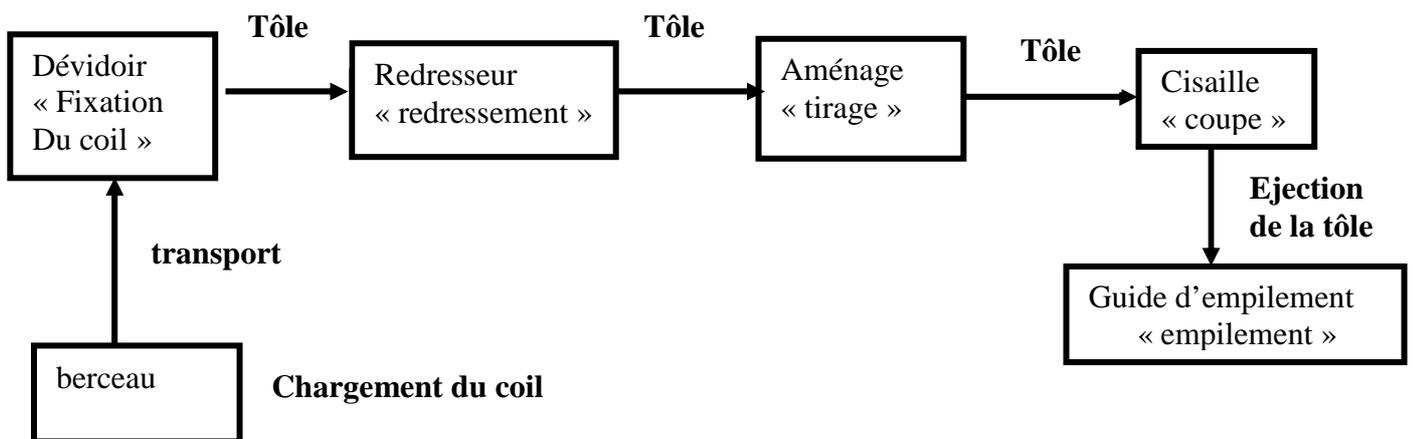
**Légendes des deux figures**

- |   |                                    |
|---|------------------------------------|
| 1. Bras maintien.                       | 2. Système d'introduction (lame) . |
| 3. Table d'entrée.                      | 4. Redresseur aménage.             |
| 5. Photocellules.                       | 6. Table de sortie.                |
| 7. Aménage motorisé.                    | 8. Cisaille .                      |
| 9. Compteur de révolution électronique. | 10. Tôles empilés.                 |
| 11. Trappes d'éjection.                 | 12. Guide d'empilement.            |
| 13. Chariot d'empilement.               | 14. Coil (bobine de tôle).         |
| 15. Dévidoir.                           | 16. Bras presseur.                 |
| 17. Mandrin.                            | 18. Table berceau.                 |
| 19. Berceau (chariot).                  |                                    |

**2. Description de l'installation**

Elle est composée de quatre parties différentes, chacune assure une fonction bien définie. En effet, la première partie est dite « berceau et dévidoir », qui servent, respectivement, à réduire au minimum la manipulation des coils, et à empêcher le déroulement imprévu des premières spires du coil et de fixer ce dernier. Quand à la deuxième partie « redresseur », elle sert à redresser la tôle. La troisième « aménage » sert au tirage de la tôle du redresseur vers la quatrième partie qui est la cisaille. Cette dernière sert à couper la tôle.

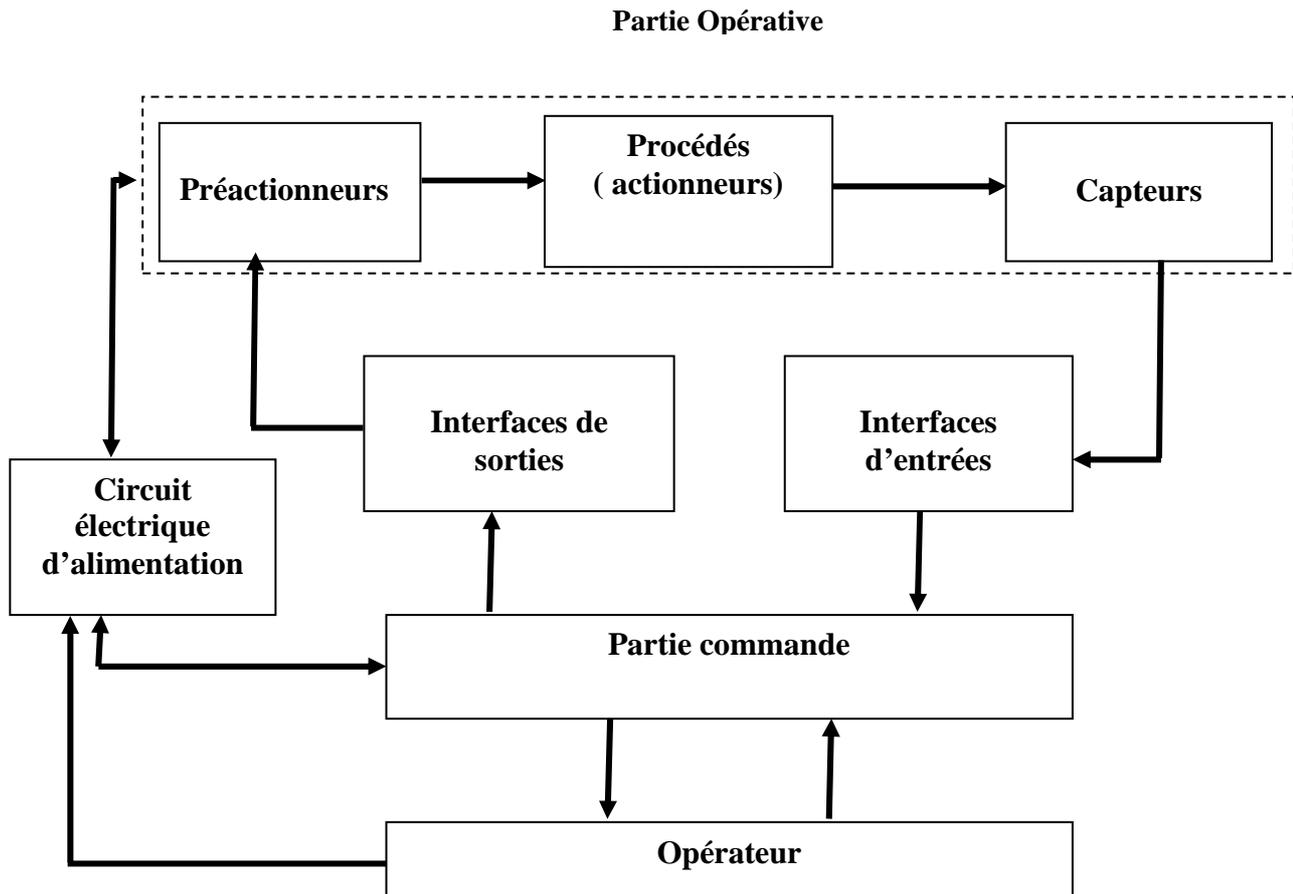
Le cycle du fonctionnement des différentes parties est résumé dans la **figure I.3** .



**Figure I.3** : schéma synoptique du cycle général de fonctionnement

Chaque partie de l'installation est composée (**Figure I. 4**) de :

- la partie opérative composée : de la structure mécanique, des pré-actionneurs, des actionneurs, des capteurs ;
- la partie commande constituée de l'armoire de commande ;
- les interfaces d'entrées/sorties ;
- le circuit électrique d'alimentation ;



**Figure I. 4** : composition des parties de l'installaton.

### 3. La partie opérative

C'est la partie qui assure les transformations des matières d'œuvres permettant d'élaborer la valeur ajoutée recherchée (produit fini). Elle est constituée de:

- La structure mécanique .
- Prés-actionneurs (électrovannes, distributeur,etc) .

- Actionneurs (vérins, moteurs, ventouse, etc) .
- Capteurs (de position, de niveau, etc) .

### 3.1. Structure mécanique : composants essentiels de chaque partie :

#### a) Berceau et dévidoir

- **Table berceau** : sous forme de V, elle sert de soutien au coil et permet de le soulever à la hauteur voulue.
- **Bras presseur** : utilisé pour empêcher le déroulement imprévu des premières spires du coil en exerçant une pression constante, grâce à deux rouleaux presseurs, sur le diamètre extérieur du coil quand on enlève les feuillards d'emballages.
- **Mandrin** : c'est sur ce dispositif qu'on fixe les coils.

#### b) Redresseur

- **Bras maitien** : supprime la courbure de la bobine au début de la bande et aide ainsi à introduire la bande dans le tirage de bande du redresseur.
- **Table d'introduction** : sert de support à la bande rentrante ; possède un système d'introduction qui sert à couper l'emballage.
- **Guide d'entrée** : - deux rouleaux parallèle à la bande pour son soutien ;
  - deux rouleaux, réglables en largeur, pour le guide latéral de la bande à redresser .
- **Onze rouleaux redresseurs** : - quatre inférieurs motorisés ;
  - trois supérieurs non motorisés ;
  - deux d'entraînement, en entrée, motorisés ;
  - deux d'entraînement, en sortie, motorisés ;
- **Table de sortie** : sert de support à la bande sortante vers l'Aménage motorisé.
- **Guide de sortie** : deux rouleaux parallèles à la bande soutiennent la tôle à la sortie du redresseur et permettent de faire la boucle de réserve de matière afin d'éviter la casse de la bande et la détérioration du redressage.

#### c) Aménage motorisé

- **Tête d'aménage** : l'aménage se fait au moyen de deux rouleaux.
- **Guide d'entrée** : sert à centrer et à aligner la bande dans l'aménage.

le guidage se fait par deux jeux de galets à gorges réglables manuellement et possède des rouleaux de soutien de la tôle, qui évitent les risques de déformation.

**d) Cisaille et guide d'empilement**

- **Lame de coupe** : outil qui coupe la tôle.
- **Trappes d'éjection** : éjection de la tôle coupée.
- Deux montants avec pieds ;
- Deux bras de soutien qui supportent des barres longitudinales sur lesquelles sont fixées des guides d'appui des pièces.
- **Tuyauteries** : la jonction des différents appareils oléo-hydrauliques et pneumatiques se fait au moyen de tuyauteries rigides ou flexibles.

**3.2 Les prés actionneurs**

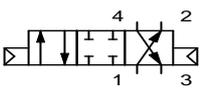
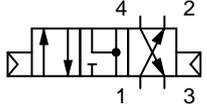
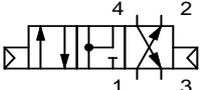
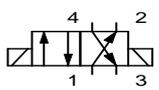
Un pré actionneur est un composant de gestion de l'énergie de commande d'actionneur. A tout actionneur est associe un pré-actionneur indispensable pour son fonctionnement.

- **Electrovanne:** pré-actionneur Tout Ou Rien (TOR), constituée principalement d'un corps de vanne où circule un fluide. 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 du fluide. Le champ de pression dépend directement de la force d'attraction de la bobine. Le bobinage doit être alimenté de façon continue pour maintenir le noyau attiré. Les électrovannes présentent dans notre installation sont résumées dans le tableau I.1

Electrovanne	Rôle	Electrovanne	Rôle
EV1	Ouverture mandrin	EV8	Rotation dévidoir avant
EV2	Fermeture mandrin	EV9	Rotation dévidoir arrière
EV3	Sortie vérin table berceau	EVC	Embrayage coupe
EV4	retour vérin table berceau	EVE	Vérin des trappes
EV6	Sortie vérin bras presseur	EVF	Frein pneumatique
EV7	Retour vérin bras presseur		

**Tableau I.1** : les différentes électrovannes dans l'installation

- **Les distributeurs :** C'est un organe dont le rôle est d'établir ou d'interrompre la communication entre le réservoir du gaz ou du fluide comprimé et l'organe commandé, ainsi, il est inséré entre la source et les organes moteurs .Notre installation est dotée de plusieurs distributeurs résumés dans le tableau .2

Type distributeur	Commande	Type distributeur	Commande
électro-magnétique 4/3 	Moteur dévidoir	électro-magnétique 4/3 	Bras presseur mandrin
électromagnétique 4/3 	Table en V	électromagnétique 4/2 	-Frein pneumatique -Embrayage coupe

**Tableau I.2 :** des distributeurs

- **Les relais :** les relais d'automatismes sont utilisés pour commander de petites charges électriques. Ils sont aussi utilisés pour servir de mémoire. Un relais peut posséder plusieurs contacts de commutation qui sont activés simultanément. Les relais qui existent sont résumés au tableau I.3

Relais	Rôle	Relais	Rôle
RP1A, RP2A RP3A	Sélection mode manuel	RP7A, RP8A RP7B, RP8B	Sélection mode Automatique
RP4a	Départ cycle	RP6B	Ejection
RP4B	Frein pneumatique	RP6A	coupe
RP5A	Remise à zéro compteur pièces	RP5B	Moteur petite vitesse

**Tableau I.3 :** les différents distributeurs de l'installation

- **Les contacteurs :** C'est un relais de haute puissance modulaire comportant des contacts à double rupture pour s'assurer de pouvoir couper des tensions et des courants élevés. Les contacteurs sont utilisés pour commuter de moyennes ou grosses charges électriques. Dès que l'on envisage de commander un moteur, quelle que soit sa puissance, on devrait utiliser un contacteur. Dans l'installation, nous avons les contacteurs qui correspondent aux relais sus-cités.

**3.3 Les actionneurs**

Un actionneur est un composant qui transforme une énergie prélevée sur une source extérieure, en une action physique sur la matière d'œuvre.

Différents critères interviennent pour le choix d'un actionneur :

- Source d'énergie disponible .
- Type d'action recherché .
- Puissance nécessaire .
- Environnement (milieu ambiant, nuisances, etc).

**3.3.1 Actionneurs utilisant l'énergie pneumatique ou hydraulique**

Les actionneurs pneumatiques convertissent l'énergie de puissance pneumatique ou hydraulique en énergie mécanique de translation, de rotation ou d'aspiration.

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

- **Les vérins :** ils sont des actionneurs linéaires dans lesquels l'énergie du fluide comprimé est transformée en travail mécanique. Du fait de la simplicité de mise en œuvre, ils sont très nombreux dans les systèmes automatisés industriels. Ceux utilisés dans notre installation sont récapitulés sans le tableau I.4

Nature du vérin	Commande	Nature du vérin	commande
Double effet hydraulique	Table berceau	Double effet hydraulique	Bras presseur
Double effet pneumatique	Table d'introduction Table de sortie	Double effet pneumatique	Trappes d'éjection

**Tableau I.4** les vérins de l'installation

- **Moteur hydraulique** : Transforme l'énergie hydraulique en énergie mécanique.
- **Pompe** : Les pompes sont des appareils qui convertissent l'énergie mécanique en énergie hydraulique.

Type du moteur	Relais thermiques	puissance	Rôle
<b>Moteur hydraulique</b>	Rtd	1.1Kw	Rotation dévidoir
<b>Pompe</b>	–	1500tr/mn (3litre/mn)	Circulation de l'huile

**Figure I.** caractéristiques du *moteur et pompe hydraulique*

➤ **Cas d'utilisatioin du vérin pneumatique**

- ✓ P → 10bar ;
- ✓ cadences élevées ;
- ✓ petits efforts..

➤ **Cas d'utilisatioin du vérin hydraulique**

- ✓ p → 300 bar ;
- ✓ dans le cas de charges variables, la vitesse est irrégulière, pour obtenir de gros efforts et/ou une vitesse constante.

**3.3.2 Actionneurs utilisant l'énergie électrique**

Il existe plusieurs types d'actionneurs électriques.

- **Moteur triphasé** : le plus utilisé dans les équipements automatiques est le moteur asynchrone, triphasé à cage (ou le moteur asynchrone à rotor en court-circuit), vu les avantages qu'il présente, dont nous citons :

- ✓ Coût de reviens bas .
- ✓ Il est robuste .
- ✓ De construction simple.
- ✓ Peu d'entretien et de surveillance .
- ✓ Encombrement réduit et simple à commander .
- ✓ Il est performant en mode de démarrage direct .
- ✓ Moteur de petites puissances, faible par apport au réseau .

- ✓ Machine ne nécessitant pas une mise en vitesse progressive .
- ✓ Le couple de démarrage est faible.

Il fonctionne avec une tension alternative triphasée, il peut être branché en étoile ou en triangle, il peut avoir deux sens de rotation en permutant deux fils de phase L<sub>1</sub> et L<sub>3</sub>, son circuit de commande comporte une protection contre les courts-circuits par fusible et un relais thermique contre la surchauffe, ainsi qu'un contacteur pour la commande.

➤ **Protection des moteurs**

✓ **Protection contre les surcharges**

Le moteurs est protégé par un relais thermique. Si le moteur se voit sollicité l'intensité du courant qui l'alimente augmente (surcharge). Sous l'effet de la chaleur due à cette surcharge une bilame fermant un contact, ouvre le circuit d'alimentation de la bobine du contacteur dont les contacts retombent.

Nous avons les moteurs et leur relais thermiques suivant :

Type du moteur	Relais thermiques	puissance	Rôle
<b>Triphasé alternatif :</b>			
M1 :aménagement GV	Rtgv	3Kw	Tirage de la tôle à grande vitesse
M2 :aménagement PV	Rtpv	3Kw	Tirage de la tôle à petite vitesse
M3 :cisaille	Rtc	3Kw	Rrtourne les lames en position haute
M :redresseur	Rtr	3KW	Tirage de la tôle

**Tableau I.5 : actionneurs électriques**

**3.4 Les capteurs**

Un capteur délivre un signale relatif à la grandeur qu'il mesure. Il prélève une information sur le comportement de la partie opérative et la transforme en une information exploitable par la partie commande. Les capteurs existants dans l'installation sont énoncés dans le tableau I.7

**3.4.1 Capteurs de fin de course (fc) mécaniques**

Ces capteurs sont à contacts et peuvent être équipés d'un galet, d'une tige souple ou d'une bille. L'information donnée est de type TOR.

**3.4.2 Les capteurs optiques**

Pour la détection des objets (tôles), la solution la plus courante utilise les capteurs photoélectriques. Ces derniers permettent de signaler la présence ou le passage d'un objet à travers ou devant un faisceau lumineux. Ils se composent essentiellement d'un émetteur de lumière associé à un récepteur photosensible.

Les trois solutions possibles sont : le système barrage, le système réflexe et le système proximité.

➤ **Le système réflexe**

Un seul boîtier remplit à la fois le rôle d'émetteur et de récepteur. Le faisceau lumineux issu de l'émetteur est renvoyé sur le récepteur par l'intermédiaire d'un réflecteur multi prismes (catadioptré).

**3.4.3 Capteurs de pression (pressostat)**

Les détecteurs de pression (ou pressostat) utilisent un organe mécanique pour provoquer la commutation si la pression est suffisante.

**3.4.4 Capteur rotatif**

Les codeurs rotatifs sont des capteurs de position angulaire. Le disque du codeur est solidaire de l'arbre tournant du système à contrôler. Il existe deux types de codeurs rotatifs, les codeurs incrémentaux et les codeurs absolus.

Type et liste capteurs	commande
<p><b>Fins de course mécaniques :</b></p> <ul style="list-style-type: none"> <li>▪ Fcbr :berceau</li> <li>▪ Fcbp :bras presseur</li> <li>▪ Fcbm :bras maintien</li> <li>▪ Fchc : cisaille haute</li> <li>▪ Fchb : cisaille basse</li> </ul>	<ul style="list-style-type: none"> <li>▪ Arrêt du berceau</li> <li>▪ Arrêt de la sortie bras presseur</li> <li>▪ Arrêt de la sortie bras maintien</li> <li>▪ Coupe</li> <li>▪ Retour de la cisaille</li> </ul>

Type et liste capteurs	Commande
<p><b>Capteurs optiques</b></p> <p>Photocellule :- Cl 1 (en bas)</p> <p style="text-align: right;">- Cl 2 (en haut)</p>	<p>-détection de la tôle arrête le moteur redresseur.</p> <p>-non détection de la tôle active le moteur redresseur</p>
<p><b>Capteur de pression</b></p> <p>-Pressostat PS</p> <p>-Pressostat Pr</p>	<p>-Arrêt de l'expansion du vérin mandrin</p> <p>-détection du seuil pneumatique de travail</p>
<p><b>Compteurs</b></p> <p>Compteur pièces</p> <p>Compteur de révolution électronique CRE</p>	<p>-Décomptage de pièces</p> <p>-débrayage de l'avance rapide et embrayage de l'approche lente.</p> <p>-débrayage de la marche lente et commande du frein pneumatique et la coupe</p>

**Tableau I.7** : les capteurs de l'installation

### 3.4.5 Temporisateurs (*minuteurs*)

Les minuteurs sont réglés par l'opérateur selon la durée voulue de chaque action. Dès que l'action est activée, la minuterie se déclenche. Après écoulement du temps de la minuterie, l'action associée s'arrête.

Nous distinguons dans le système les minuteurs suivantes:

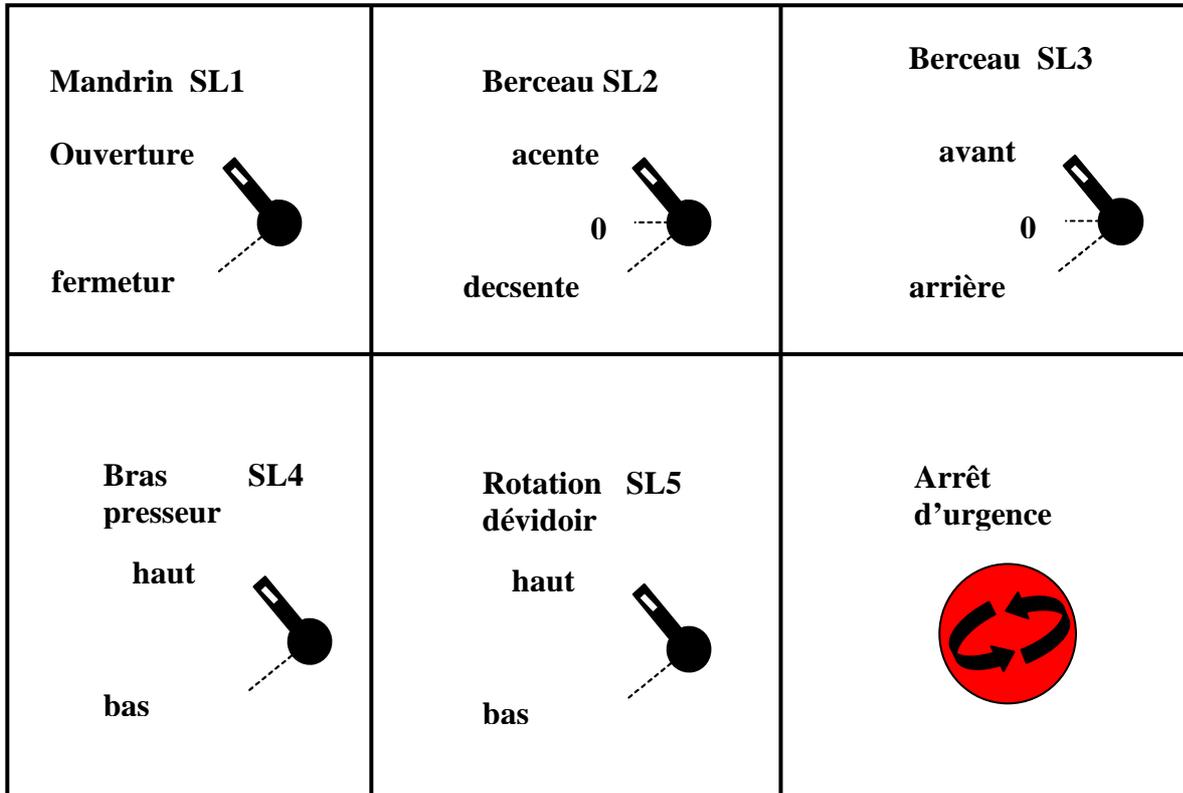
- T1 :Temporisation qui permet de régler le coup de frein du passage de la grande vitesse à la petite vitesse .
- T2 : Temporisation qui permet de régler le temps d'ouverture des trappes d'éjection.

## 4 Partie commande

Tout processus nécessite un système de contrôle et de commande permettant à l'homme d'intervenir dans le processus.

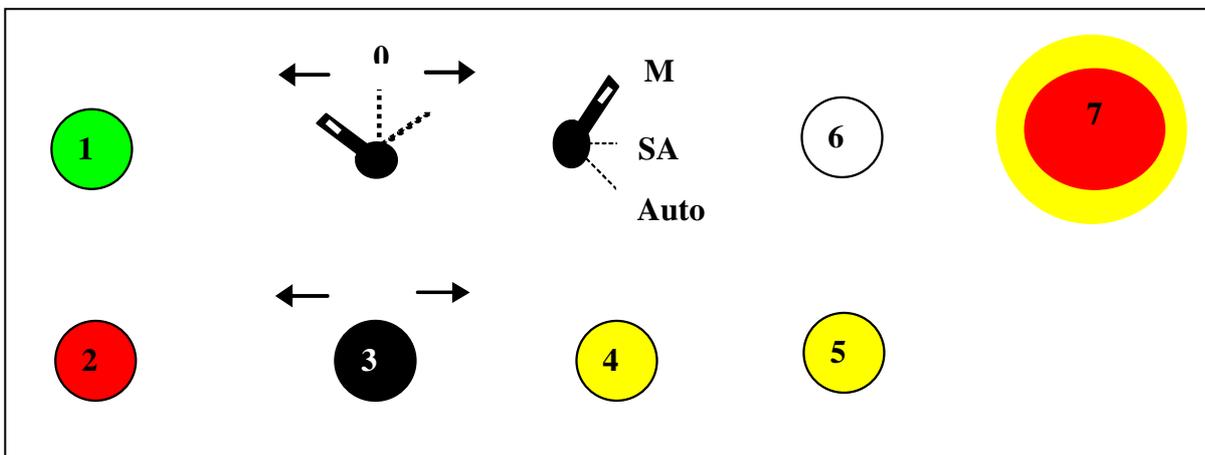
Dans notre cas, et selon le fonctionnement, il existe trois pupitres de commande placés dans différents endroits :

➤ **pupitre de commande partie hydraulique** : situé sur le bati du dévidoir. (figure I.5)



FigureI.5 : pupitre de commande partie hydraulique

➤ **Pupitre de commande redresseur** : situé sur le redresseur (FigureI.6)

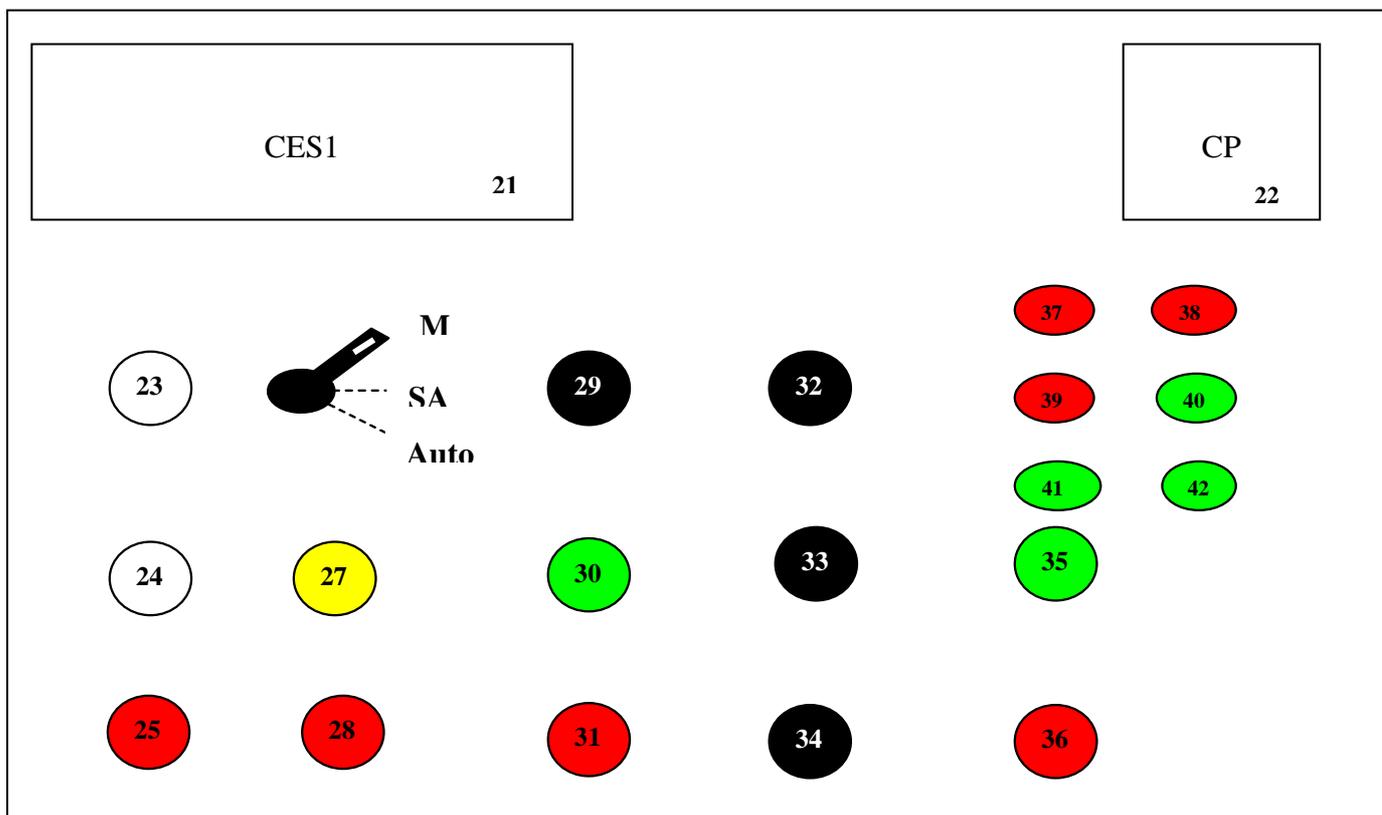


FigureI.6 :pupitre de commande redresseur

**Legende de la figure I.6**

- 1- mise en marche moteur distributeur.
- 2- arrêt moteur distributeur.
- sélecteur cycle.
- sélecteur sens avant-arrière.
- 7-arrêt d'urgence redresseur.
- 3-bouton poussoir de mise en marche avant-arrière.
- 4-départ cycle.
- 5-voyant départ cycle.
- 6-voyant sous tension.

- **Poste de commande(figure I .7):** contient aussi un coffret électrique renferment la logique de la commande :- bac à cartes (cartes relais ,carte temporisation et carte alimentation) ;
  - relais de puissance pour la commande des moteur et des embrayages ;
  - fusible ,transformateur, etc.



FigureI.7 : Poste de commande

**Legende de la figure I.7**

21 compteur CES1.	34 A-coups dévidoir (option).
22 compteur pièce.	35 Sous tension.
23 avant.	36 manque d'air.
24 arrière.	37 fin de série.
25 arrêt.	38 remise à zero.
26 selecteur cycle.	39 coupe.
27 depart cycle.	40 positionnement.
28 arrêt cycle.	41 petite vitesse .
29 A-coups coupe.	42 grande vitesse.
30 marche cisaille.	31 Arrêt cisaille.
32 A-coups petite vitesse.	33 A-coups grande vitesse.

**5. Circuit électrique**

Le schéma électrique ainsi que la signification des différents éléments le composant sont donnés en annexe B.

**6. Modes de fonctionnement**

Un sélecteur à trois positions (26) permet le choix du cycle de fonctionnement.

Manuel, semi automatique et automatique.

**6.1 Cycle manuel**

Permet le fonctionnement des boutons "à-coups".

- 32 ) à-coups grande vitesse
  - 33 ) à-coups petit vitesse
  - 34 ) à-coups dévidoir (option).
  - 29 ) à-coups coupe.
- } sens avant ou arrière sélectionné respectivement par les boutons lumineux (23) et (24).

**6.2 Cycle semi automatique**

Le redresseur doit être en marche avant, et le moteur cisaille doit être commandé (valable également en automatique). Ce cycle permet l'exécution d'une pièce qui n'est pas comptabilisée par le compteur de pièces.

La valeur du déplacement reste affichée sur le compteur, ce qui permet de calculer une éventuelle correction des réglages de seuils, si un dépassement est constaté. Pour obtenir la valeur du dépassement, il faut tenir compte du fait que le compteur travaille en décomptant :

$DEPASSEMENT = 99999.9 - VALEUR\ AFFICHEE.$

### **6.3 Cycle automatique**

Un appui sur le bouton jaune (27) du poste de commande effectue une remise à zéro du compteur. Au relâchement, le compteur se repositionne à la valeur sélectionnée, et le cycle démarre. Un appui sur le bouton (28) permet de terminer la pièce en cours, et arrête le cycle automatique. Lorsque le compteur de pièces arrive à la valeur présélectionnée, le cycle automatique s'arrête.

## **7. Fonctionnement de l'installation**

### **➤ Préparatifs**

- ✓ Choisir le mode semi automatique.
- ✓ Mettre l'installation sous tension .
- ✓ vérifier la pression de service et niveau l'huile.
- ✓ Allumage de la pompe.
- ✓ Première coupe de la tôle : L'opérateur met le coil (bobine) sur le berceau , il l'achemine ensuite au dévidoir (où il sera monté sur le mondrin) . ensuite le fil d'emballage sera coupé par la lame, ce qui va permettre le passage du feuillard ( feuille de tôle) au redresseur aménagé , ensuite à l'aménagé motorisé et fini son parcours à la cisaille (où il sera coupé).

### **➤ Principe général du fonctionnement du système en mode automatique**

- ✓ Réglage des compteurs
- ✓ On sélectionne le mode automatique
- ✓ Appui sur le bouton départ cycle
- ✓ Mise en marche du moteur cisaille
- ✓ Mise en marche des moteurs en parallèle:
  - Redresseur : - C12 activation du moteur  
- C11 arrêt du moteur
  - Aménagé : - Activation moteur grande vitesse(GV) et petite vitesse(PV).

- debrayage du moteur GV, Frein pneumatique, embrayage du moteur PV.
- debrayage du moteur PV.
- Frein pneumatique.
- ✓ Embrayage cisaille effectué par électrovalve (coupage de la tôle)
- ✓ Decomptage compteur pièces
- ✓ Répétition du cycle jusqu'à la dernière pièce.

## **8. Conclusion**

vu l'importance de l'installation au sein de l'unité cuisson (les tôles découpées partent pour l'usinage par d'autres machines) et parce que son arrêt est susceptible d'interrompre la production ; l'entreprise afin d'être conquérante veut apporter des améliorations à l'installation et ceci parce qu'elle est souvent sujette à des pannes répétitives ( l'installation où les techniciens interviennent le plus) et parce qu'elle utilise des cartes relais rares sur le marché et à coût élevé. Ceux-ci sont les deux inconvénients majeurs qui poussent l'entreprise à revoir la technologie câblée qu'elle utilise. Ajoutant à ceux là les limitations dont souffre cette technologie :

- leur encombrement ( volume) ;
- leur manque de souplesse vis-à-vis de la mise au point des commandes et de l'évolution de celles-ci (amélioration, nouvelles fonctions) ;
- la difficulté de maîtriser des problèmes complexes ;
- la complexité de recherche des pannes et donc de dépannage ;
- une rentabilité financière limitée aux fonctions simples en raison de l'apparition de technologies programmables.

Ainsi , une orientation vers l'automatisation de l'installation par les API est plus qu'indispensable. Pour cela, dans le chapitre suivant nous allons aborder les améliorations apportées à l'installation et la modélisation du nouveau système.

# Chapitre II

***Modélisation de  
l'installation à l'aide  
du GRAFCET***

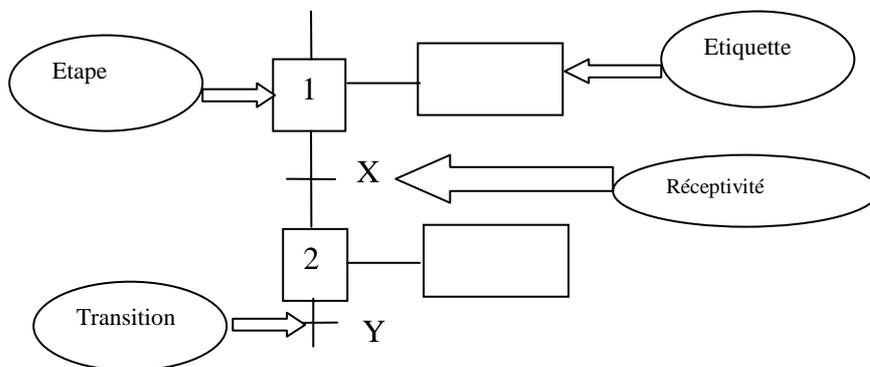
### 1. Introduction

La création d'une machine automatisée nécessite un dialogue entre le client qui définit le cahier des charges (qui contient les besoins et les conditions de fonctionnement de la machine), et le constructeur qui propose des solutions. Ce dialogue n'est pas toujours facile : le client ne possède peut-être pas la technique lui permettant de définir correctement son problème. D'autre part, le langage courant ne permet pas de lever toutes les ambiguïtés dues au fonctionnement de la machine (surtout si des actions doivent se dérouler simultanément). C'est pourquoi l'ADEPA (Agence pour le Développement de la Productique Appliquée à l'industrie) a créé le **GRAFCET**.

### 2. Généralité sur le GRAFCET

Le GRAFCET (Graphes Fonctionnel de Commande des étapes et Transitions) est l'outil de représentation graphique d'un cahier des charges. Il a été proposé par l'ADEPA en 1977 et normalisé en 1982 par la NF C03-190.

Le GRAFCET est une représentation alternée d'**étapes** et de **transitions**. Une seule transition doit séparer deux étapes (**figure II.1**).



**Figure II.1** : représentation étapes et transitions

Une **étape** correspond à une situation dans laquelle les variables de sorties conservent leur état. Les **actions** associées aux étapes sont inscrites dans les **étiquettes**.

Une **transition** indique la possibilité d'évolution entre deux étapes successives. A chaque transition est associée une condition logique appelée **réceptivité**.

**2.1 Les règles d'évolution du GRAFCET**

**Règle 1 : situation initiale**

Cette représentation indique que l'étape est initialement activée (à la mise sous tension de la partie commande). La situation initiale, choisie par le concepteur, est la **situation à l'instant initial**. (Figure II.2)



**Figure II.2 : étape initiale**

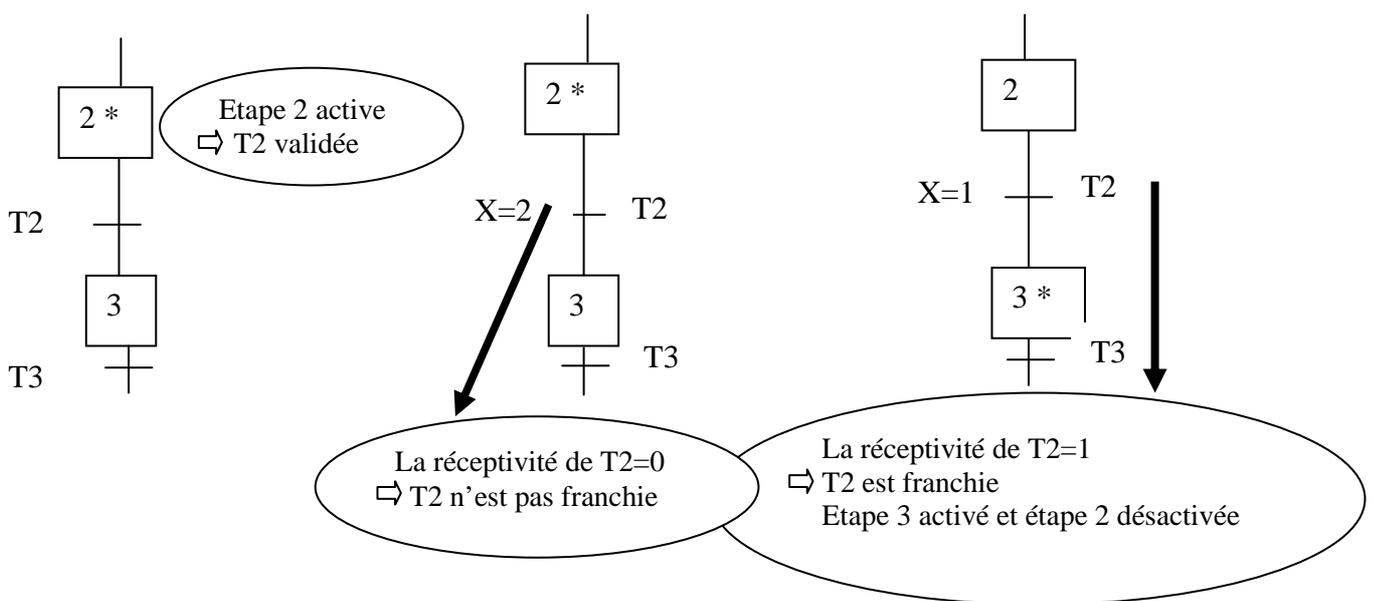
**Règle 2 : franchissement d'une transition**

Une transition est **franchie** lorsque l'étape associée est **active** et la **réceptivité** associée à cette transition est **vraie**.

**Règle 3 : évolution des étapes active**

Le franchissement d'une transition provoque simultanément :

- la **désactivation** de toutes les étapes immédiatement précédentes reliées à cette transition ;
- l'**activation** de toutes les étapes immédiatement suivantes reliées à cette transition.



**Figure II.3 : évolution des étapes actives**

### **Règle 4 : transitions simultanées**

Plusieurs transitions simultanément franchissables sont simultanément franchies.

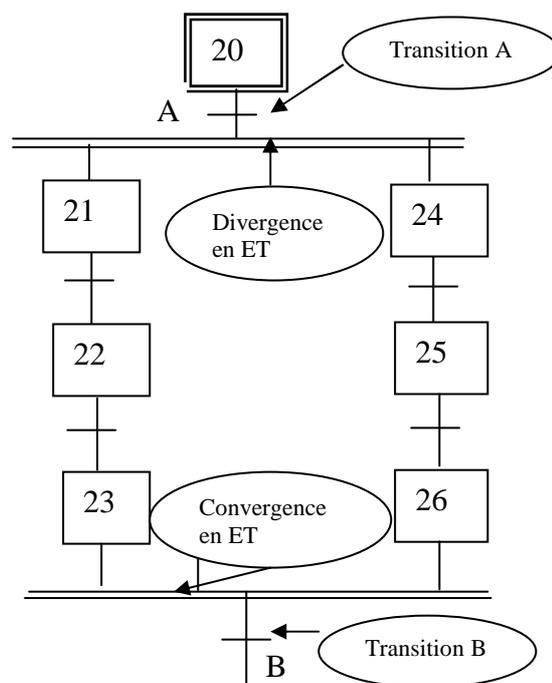
### **Règle 5 : activation et désactivation simultanées**

Une étape à la fois activée et désactivée reste active.

## **2.2 Divergence et convergence en ET (séquences simultanées)**

**Divergence en ET** : lorsque la transition A est franchie, les étapes 21 et 24 sont actives (**figure II.4**).

**Convergence en ET** : la transition B sera validée lorsque les étapes 23 et 26 seront actives. Si la réceptivité associée à cette transition est vraie, alors celle-ci est franchie (**figure II.4**).

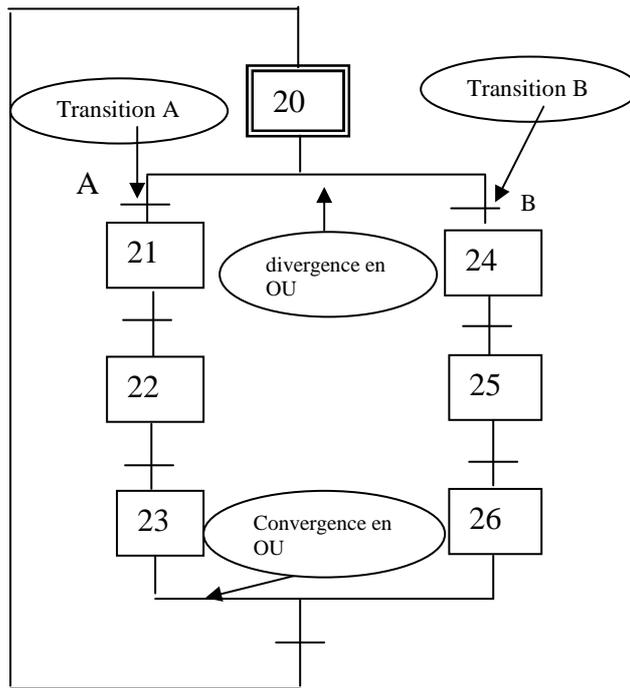


**Figure II.4** : représentation d'une séquence simultanée

## **2.3 Divergence et convergence en OU (aiguillage)**

**Divergence en OU** : l'évolution du système vers une branche dépend des réceptivités A et B associées aux transitions (**figure II.5**).

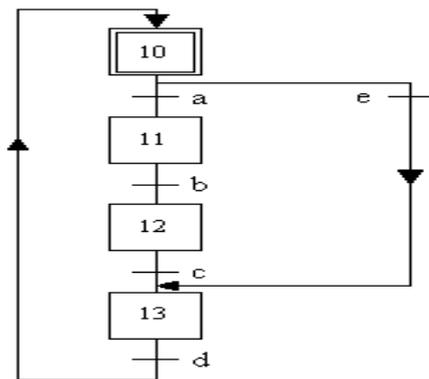
**Convergence en OU** : après l'évolution dans une branche, il y a convergence vers une étape commune (**figure II.5**).



**Figure II.5 :** représentation d'une séquence en aiguillage.

### 2.4 Saut en avant (saut de phase)

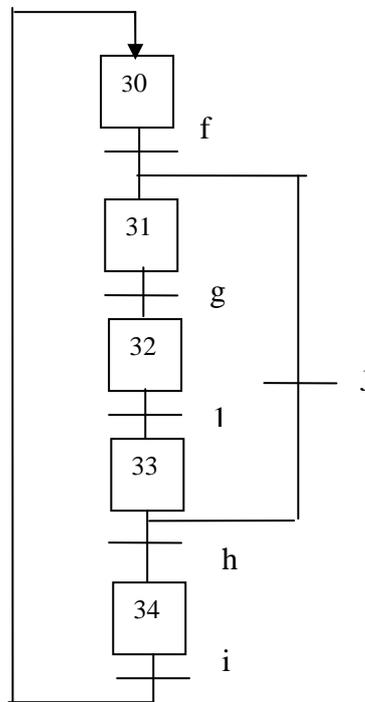
Le saut en avant permet de sauter une ou plusieurs étapes lorsque les actions à réaliser deviennent inutiles (**figure II.6**).



**Figure II.6 :** représentation du saut en avant

### 2.5 Saut en arrière (reprise de phase)

Le saut en arrière permet de reprendre une séquence lorsque les actions à réaliser sont répétitives (**FIGURE II.7**).



**Figure II.7:** représentation du saut en arrière.

### 2.6 Niveau d'un GRAFCET

#### ➤ Niveau 1

Appelé aussi le niveau de la partie commande, il décrit l'aspect fonctionnel du système et des actions à faire par la partie commande en réaction aux informations provenant de la partie opérative, indépendamment, de la technologie utilisée. Les réceptivités sont décrites en mots et non en abréviations, nous associons un verbe à l'infinitif pour exprimer les actions.

#### ➤ Niveau 2

Appelé aussi le niveau de la partie opérative, il tient compte de plus de détail de la technologie des actionneurs, des pré-actionneurs et des capteurs. La représentation des actions et des réceptivités est écrite en abréviation. Nous associons une lettre majuscule à l'action et une lettre minuscule à la réceptivité.

### 2.7 Mise en équation d'un GRAFCET

Pour passer de l'étape de modélisation du procédé par le GRAFCET à l'étape de programmation par l'un des langages accepté par l'automate utilisé, nous devons traduire notre grafcet (figureII.8 )sous forme d'équations logiques combinatoires , en précisant les conditions d'activation, de désactivation, ainsi l'initialisation et les arrêts d'urgence d'une étape et de l'action associée.

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

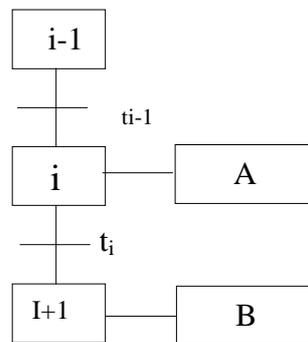
$X_n=1$  : si l'étape est active ;

$X_n=0$  : si l'étape est inactive ;

De plus, la réceptivité  $t_n$  qu'est une variable binaire à pour valeur :

$t_n=1$  : si la réceptivité est active ;

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



**FigureII.8** : *exemple d'un grafcet*

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

➤ condition d'activation de l'étape  $i$  :

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

➤ condition de désactivation de l'étape  $i$  :

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

L'action associée à l'étape  $i$  :  $A = X_i$ .

Si les conditions d'activation et désactivation se passent en même temps pour l'étape  $i$ , elle reste dans son état mémoire.

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

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

- Si en tenant compte de mode marche/ arrêt et arrêt d'urgence:

Init=1 : initialisation du grafcet (mode arrêt).

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

Audoux=1 : de toutes les actions en maintenant les étapes actives.

AUDur=1 : désactiver les étapes du grafcet.

Les équations deviennent :

- Pour les étapes initiales :

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

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

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

- Pour les étapes non initiales :

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

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

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

$$\text{Pour les étapes : } A = X_i \cdot \overline{AUDoux}$$

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

Après l'étude du système à automatiser et avoir identifié les différentes réceptivités ainsi que les différentes actions, le cahier des charges décrit au premier chapitre est modélisé sous forme de grafcet niveaux 1 et 2.

### **3. Améliorations apportées à l'installation**

Afin de tirer profil des pannes qui surviennent lors du fonctionnement automatique et pour un rendement meilleur de l'installation nous proposons : deux temporisations et un capteur

#### **3.1 Temporisation**

- Deux temporisations : qui substituerons le rôle des photocellules

- ✓ la première pour la photocellule c11 du redresseur qui servira à arrêter le moteur en cas de défaillance de la c11. Ainsi, il n'y aura pas une boucle plus importante qui risquerait de déformer la tôle sortie du redresseur. Une led s'allumera pour indiquer cette défaillance et l'opérateur aura le choix d'intervenir quand il voudra pour la réparation de la c11.

- ✓ la deuxième afin d'éviter le cisaillement et la coupure de la tôle .Ceci préservera aussi le moteurs grande vitesse et redresseur (le premier tire la tôle et le second et arrêter) et provoquera un arrêt du cycle et la réparation doit être faite avant la reprise du cycle.

Elles seront déterminées de façon à ne pas augmenter le temps du cycle.

### 3.2 Capteur

Un capteur qui détectera l'éjection des tôles afin qu'aucune tôle ne gêne l'autre (éviter le risque de déformation).

#### 3.2.1 Choix d'un capteur

Le capteur a deux parties distinctes. Une première partie qui a pour rôle de détecter un événement et une deuxième partie qui a pour rôle de traduire événement en un signal compréhensible, d'une manière ou d'une autre, par une partie PC. Pour choisir correctement un capteur, il faudra définir tout d'abord :

- le type de l'événement à détecter :
- la nature de l'événement ;
- La grandeur de l'événement ;
- l'environnement de l'événement.

En fonction de ces paramètres on pourra effectuer un ou plusieurs choix pour un type de détection. D'autres éléments peuvent permettre de cibler précisément le capteur à utiliser :

- ses performances ;
- son encombrement ;
- sa fiabilité (MTBF) ;
- la nature du signal délivré par le capteur (électrique, pneumatique) ;
- son prix, etc.

- ✓ Le choix du capteur est une photocellule de type barrage (**Figure II.9**) puisque la tôle tombe en chute libre est tombe dans l'empileur et il sera fixé aux pieds du guide d'empilement (peu encombrant) .

L'émetteur et le récepteur doivent être rigoureusement positionnés l'un en face de l'autre et leurs axes optiques confondus, de plus les supports des appareils doivent être robustes et indéformables.

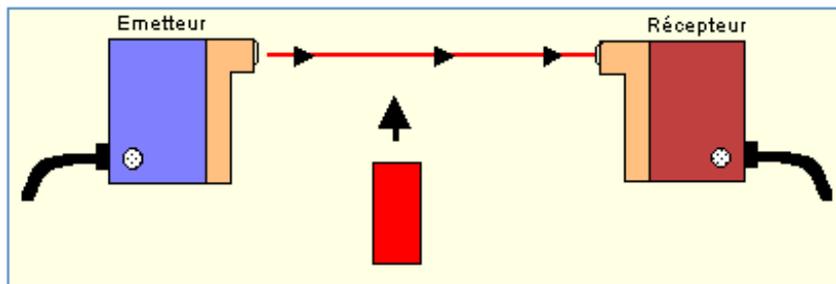


Figure II.9 : représentation d'une paire de photocellule « système barrage »

#### 4. Modélisation de l'installation

##### 4.1 Grafcet de niveau 1

##### 4.1.1 Grafcet de gestion d'arrêt d'urgence (Figure II.10)

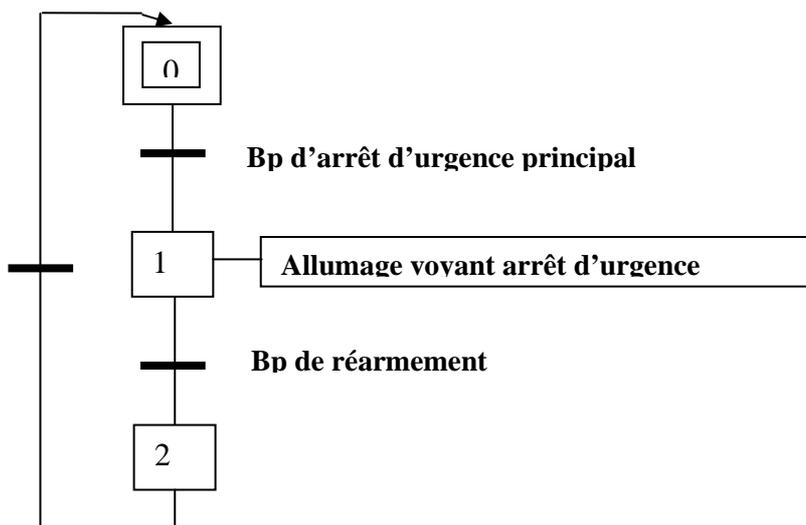


Figure II.10 : grafcet niveau 1 de gestion d'arrêt d'urgence

##### 2.1.2 Grafcet de la pompe (Figure II.11)

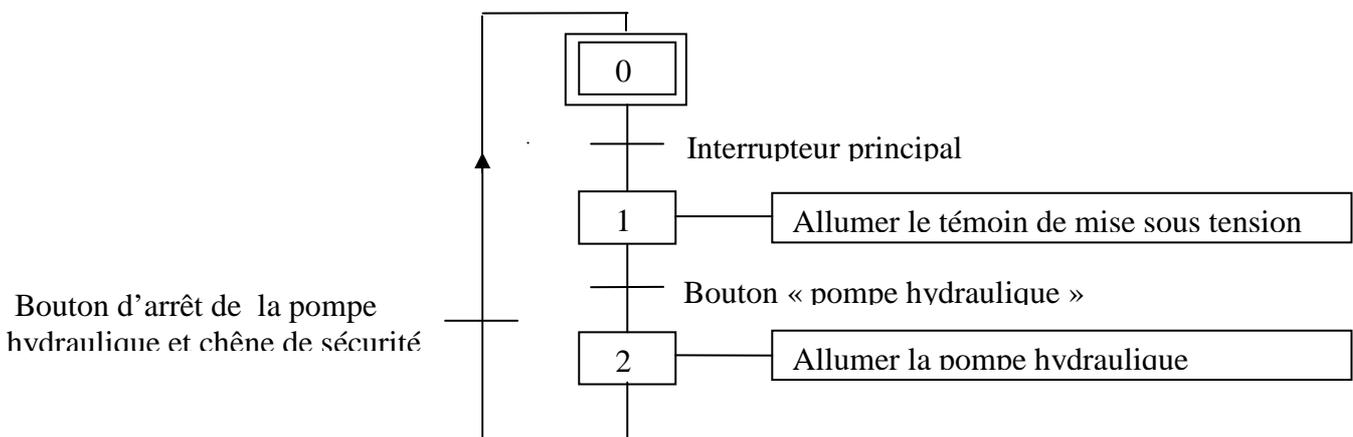
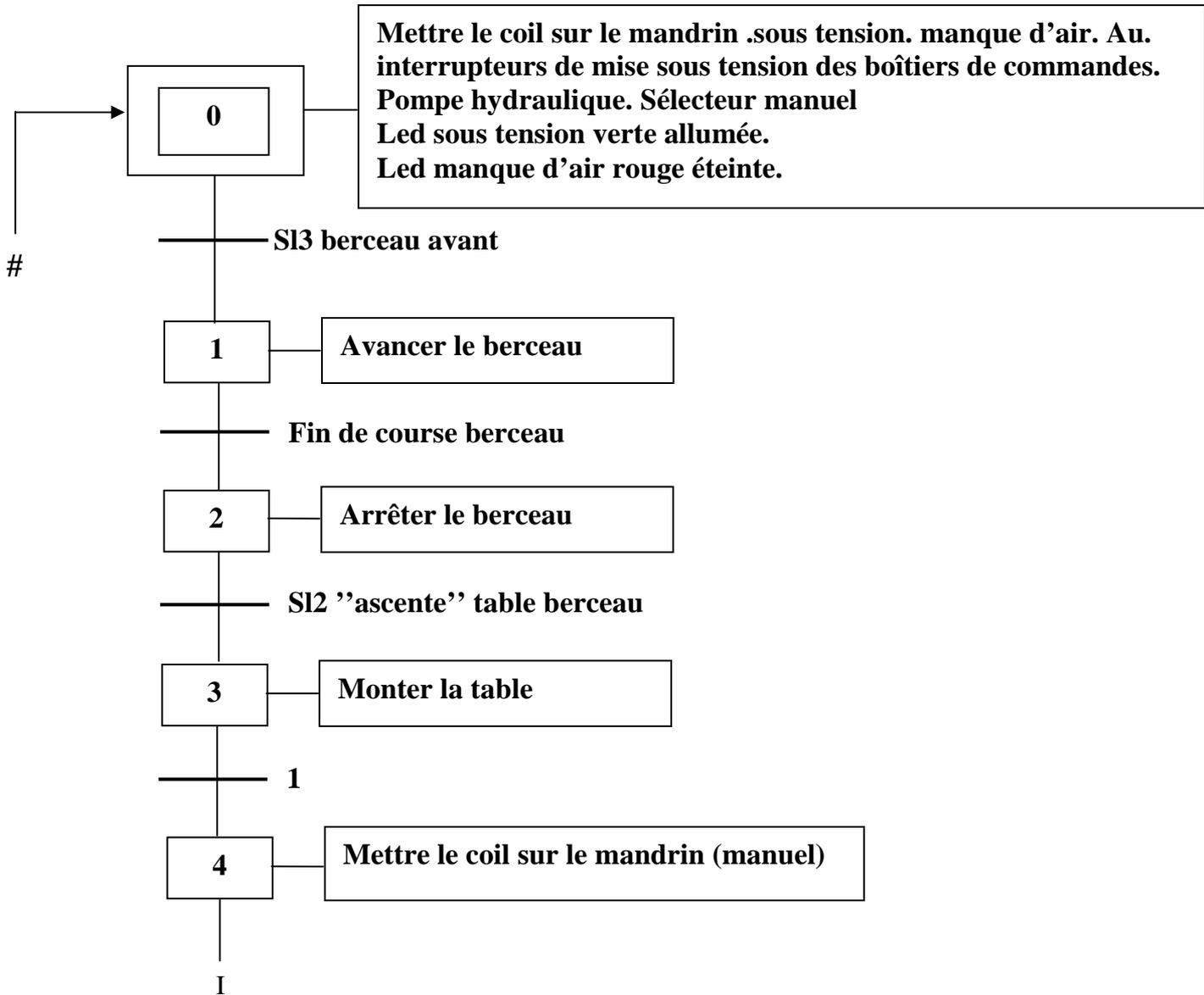
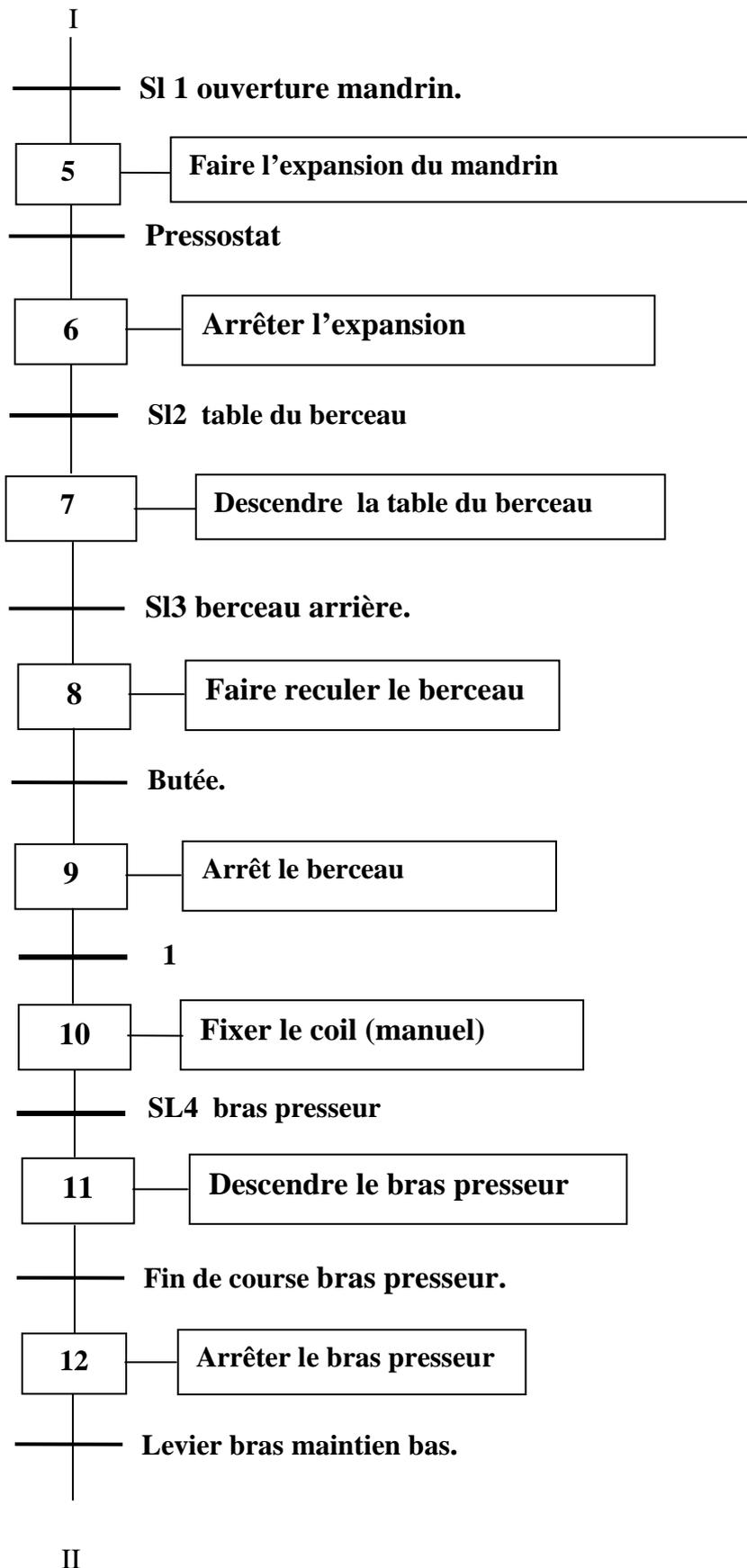
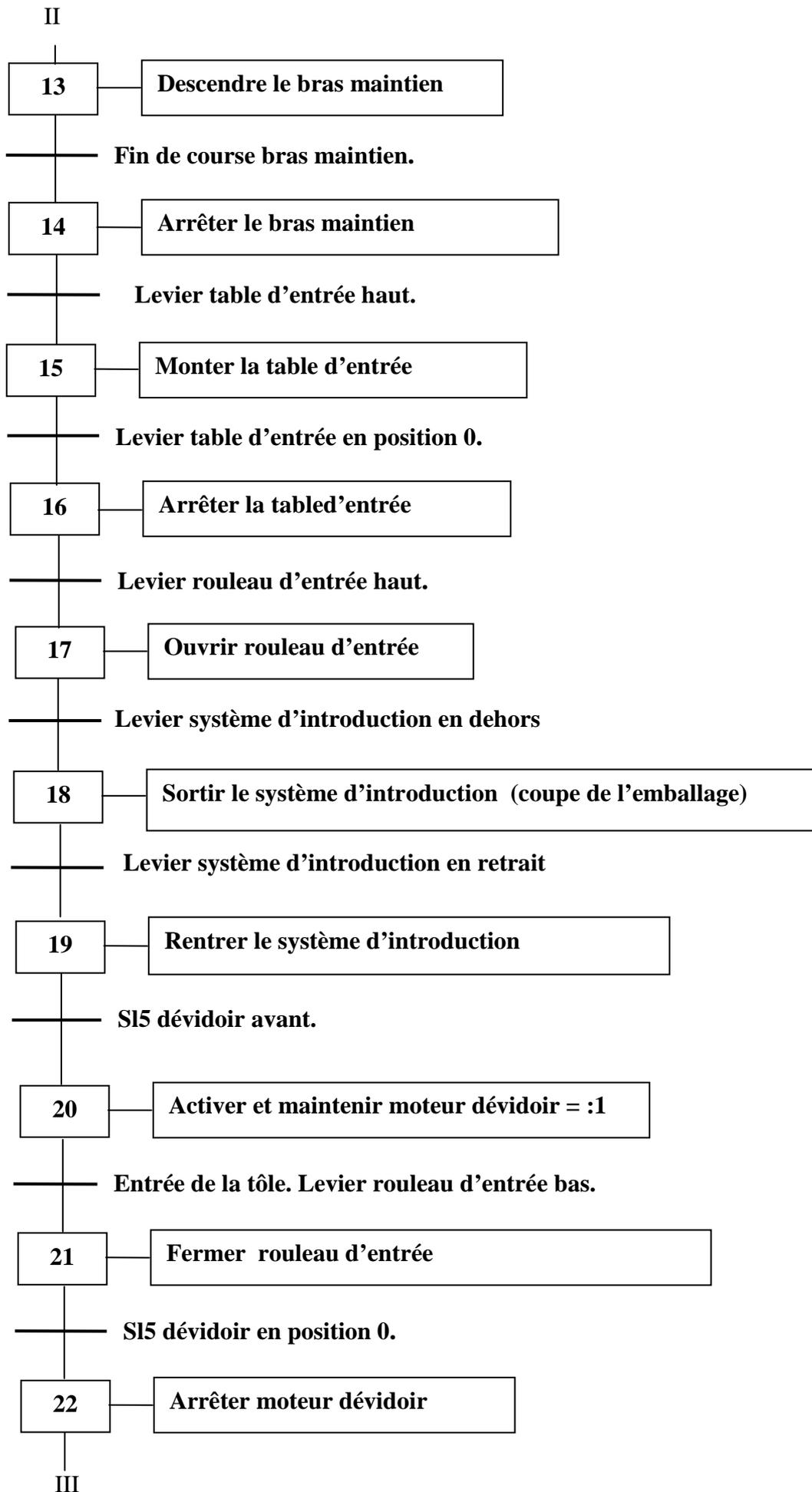


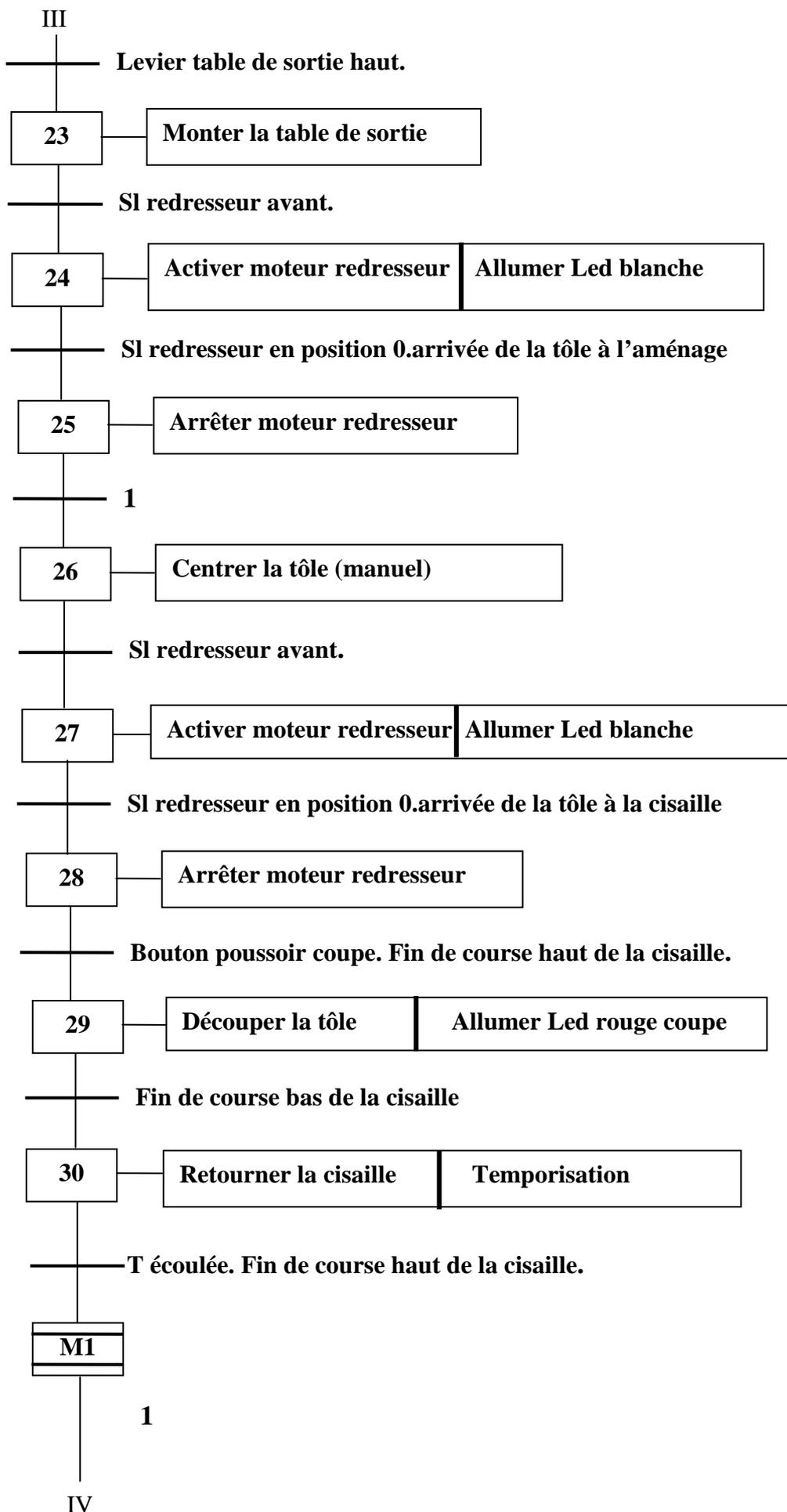
Figure II.11 : grafcet niveau 1 de la pompe

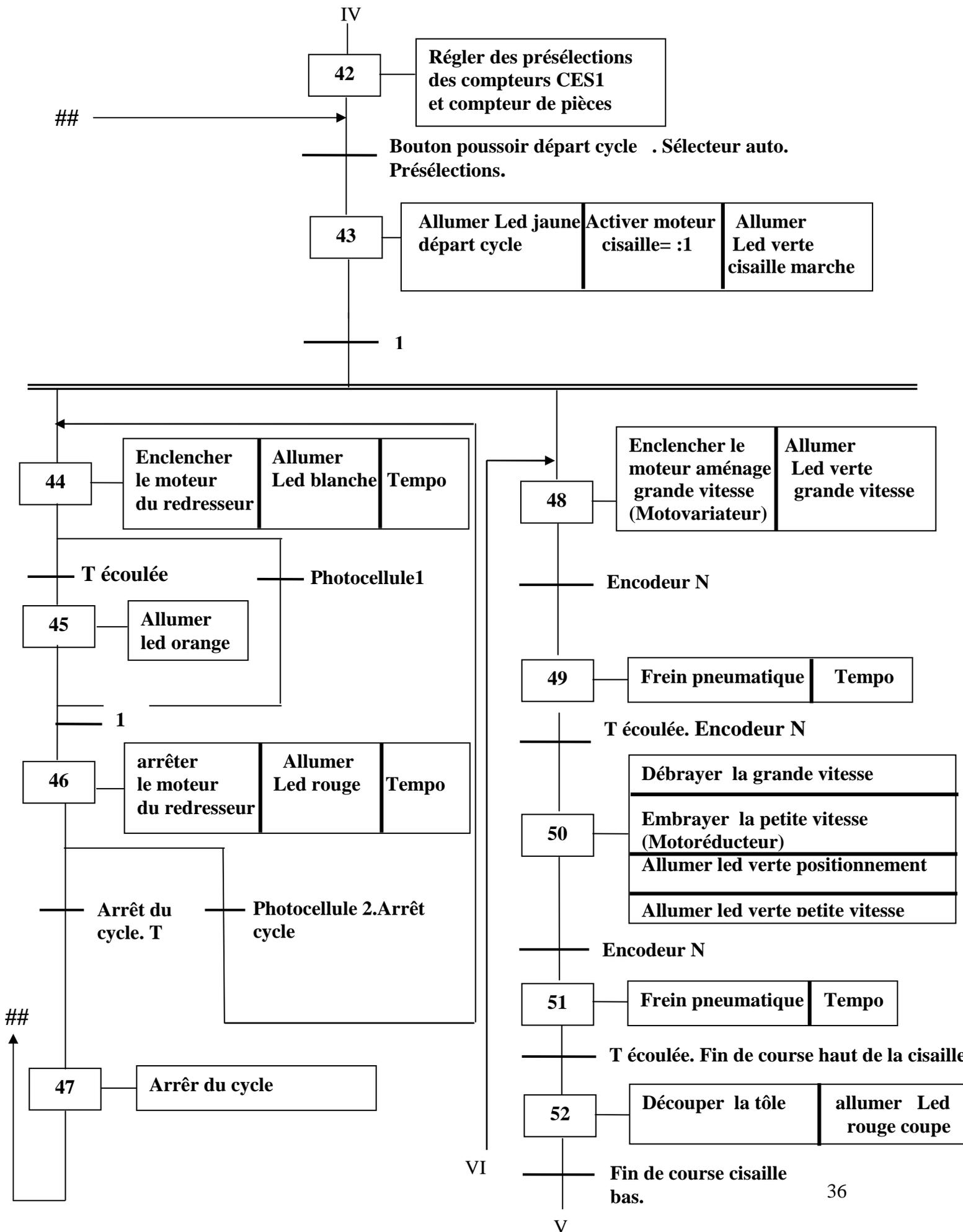
**4.1.3 Grafcet de l'installation (Figure II.12)**











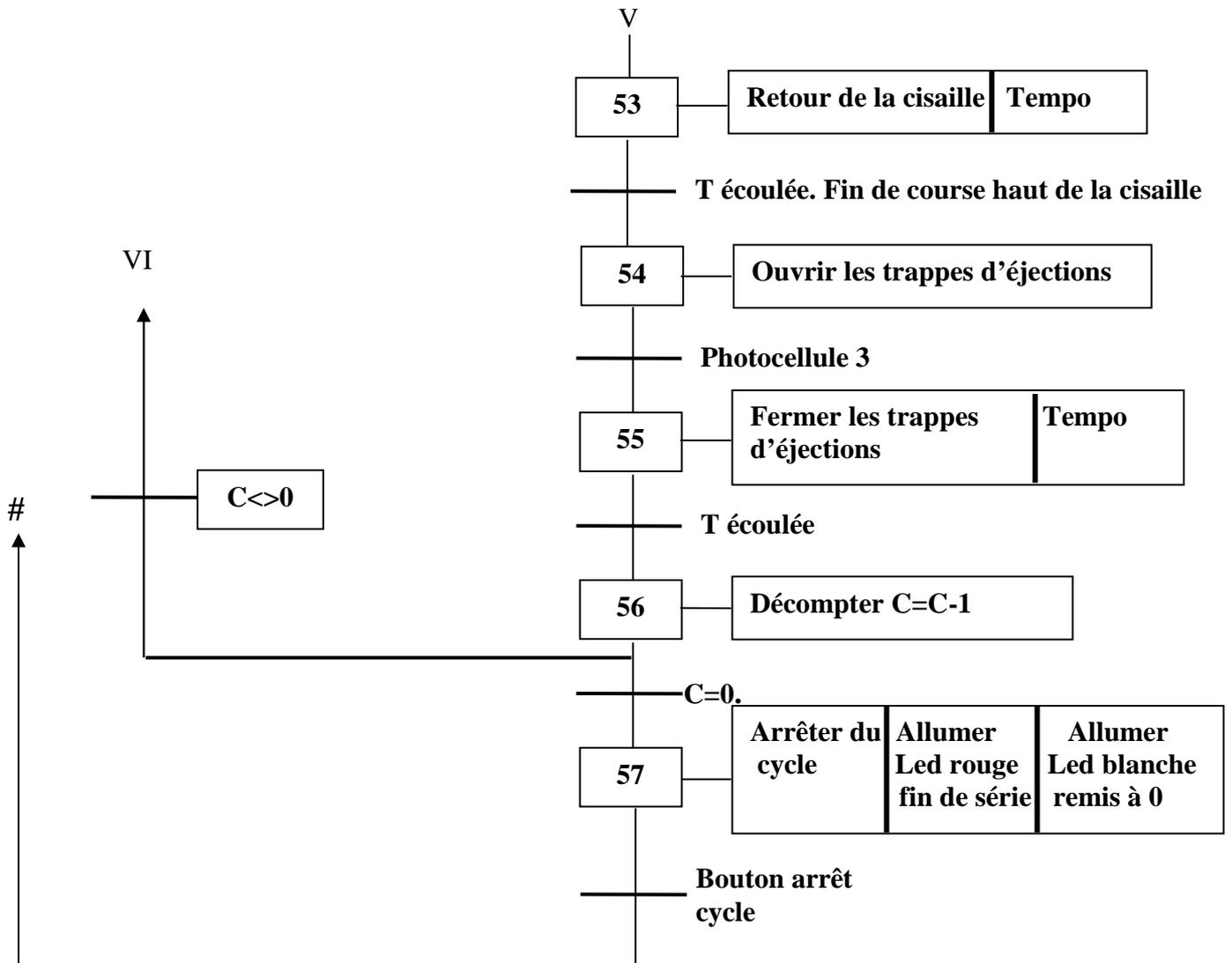
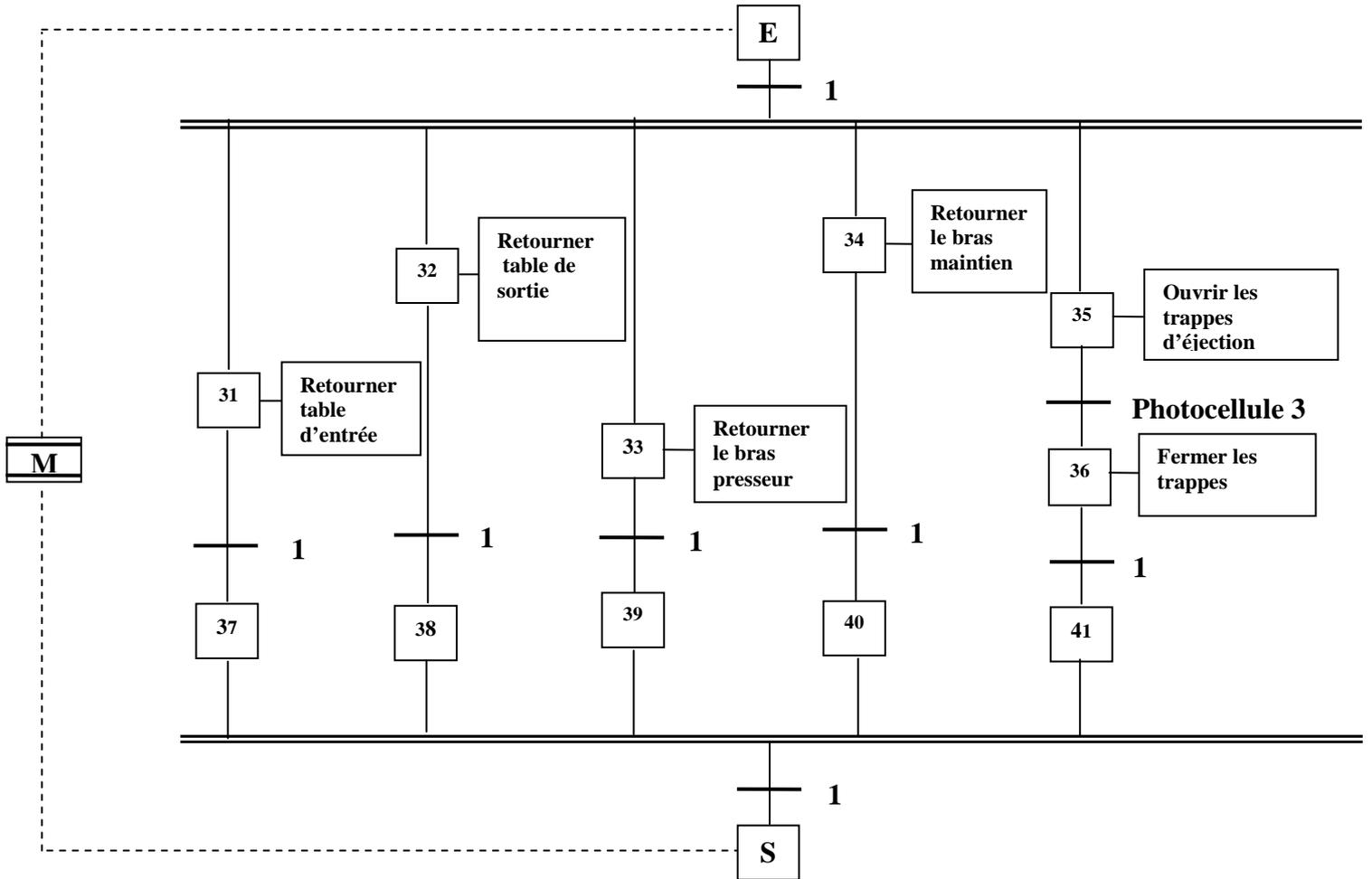


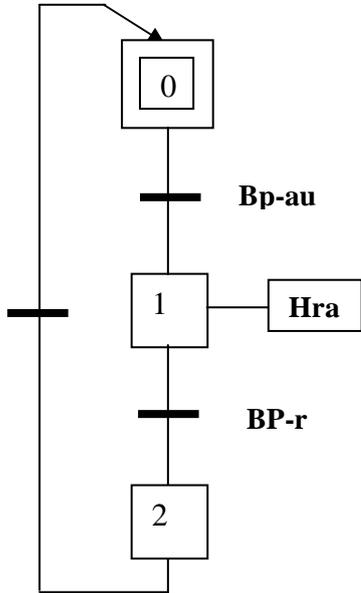
Figure II.12: grafcet niveau 1 de l'installation



**Figure II.13 : grafcet niveau 1 de la macro étape M1**

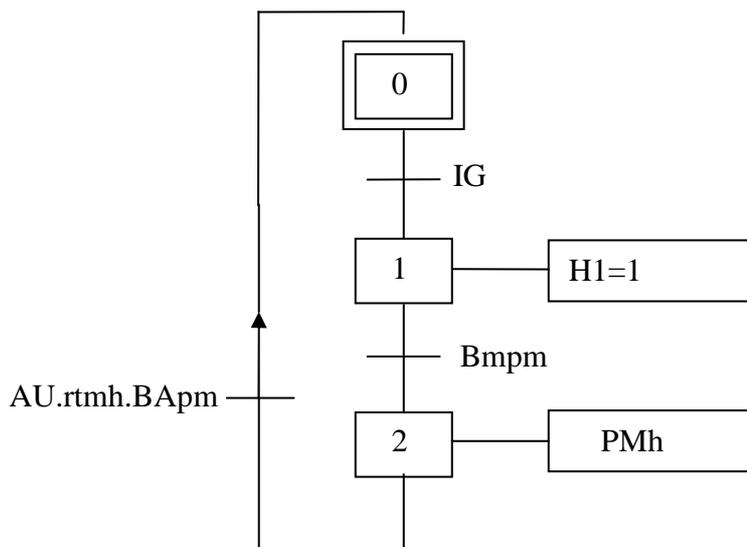
**4.2 Grafcet de niveau 2**

**4.2.1 Grafcet de gestion d'arrêt d'urgence (Figure II.14)**



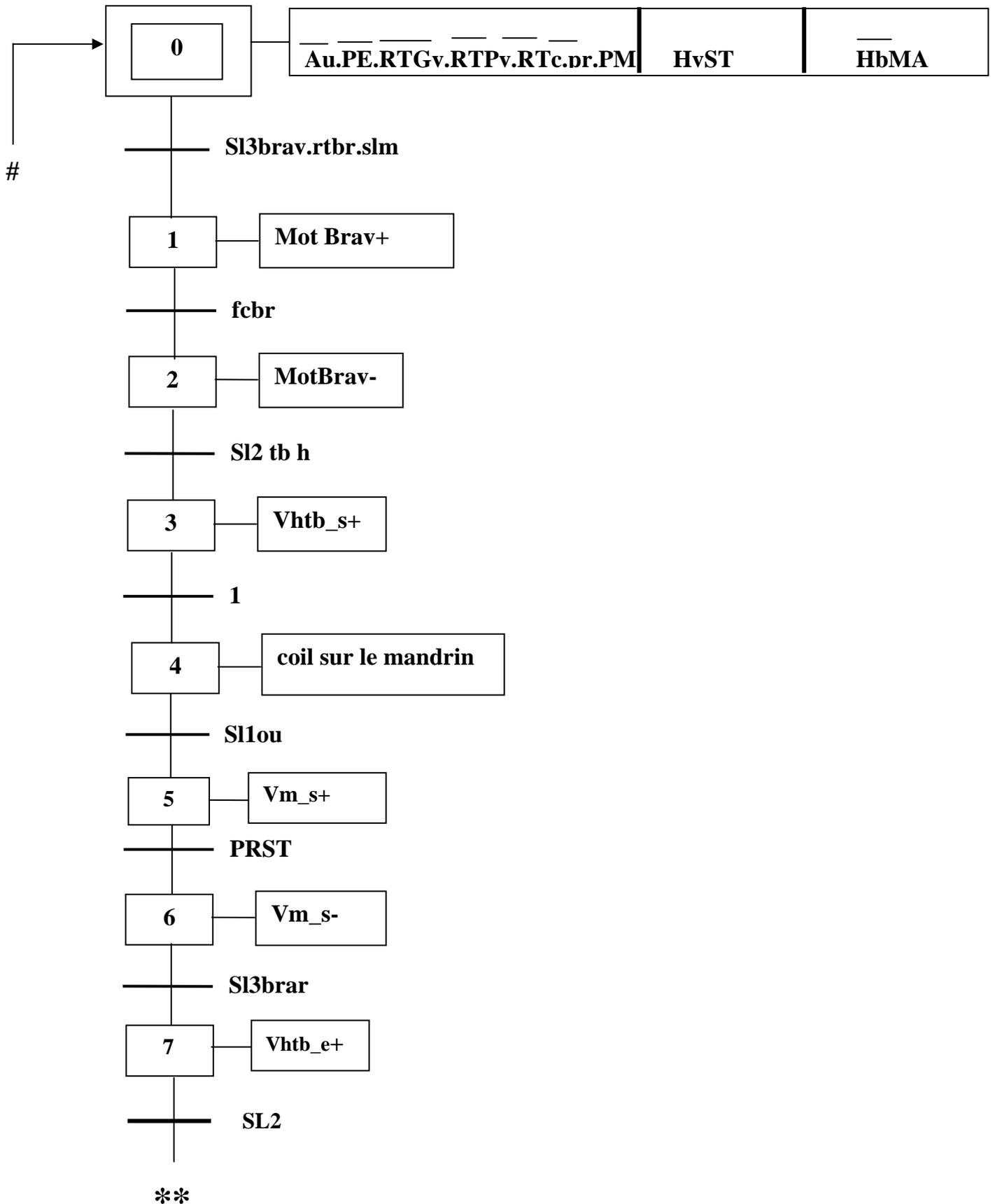
**Figure II.14 :** *grafcet niveau 2 de gestion d'arrêt d'urgence*

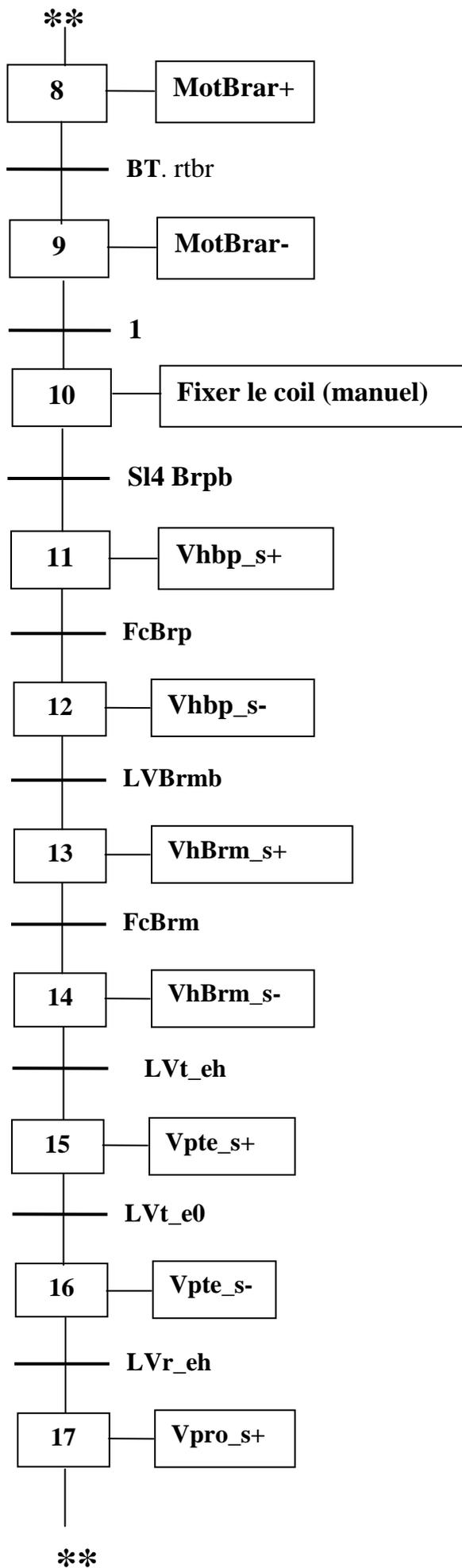
**2.2.2 Grafcet de la pompe (Figure II.15)**

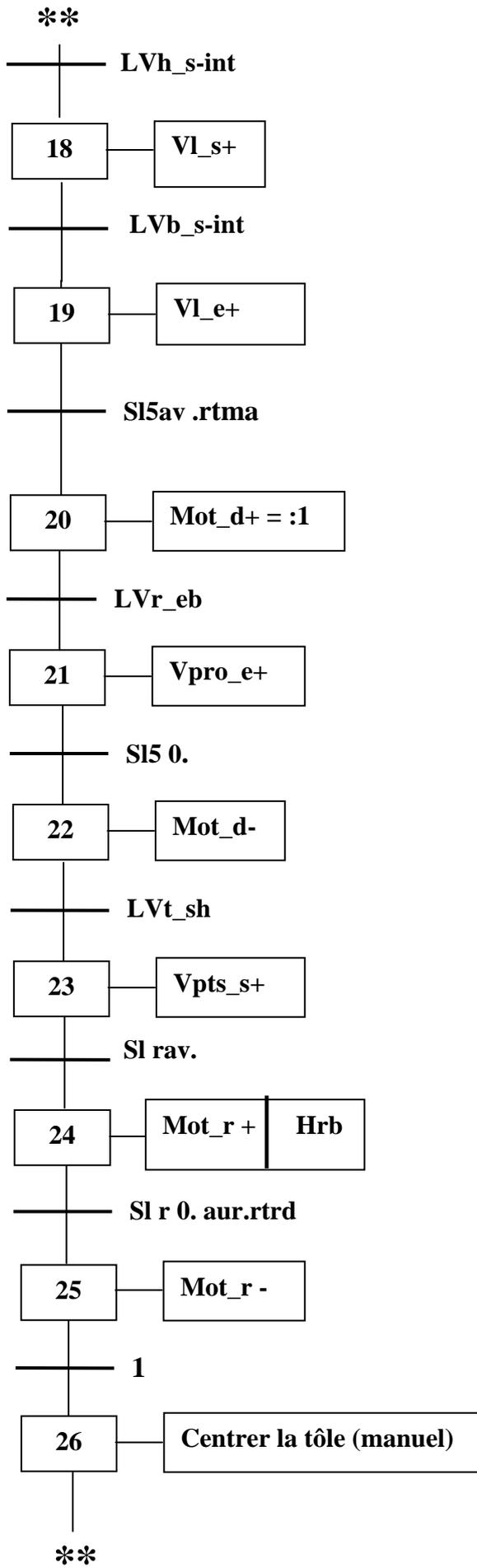


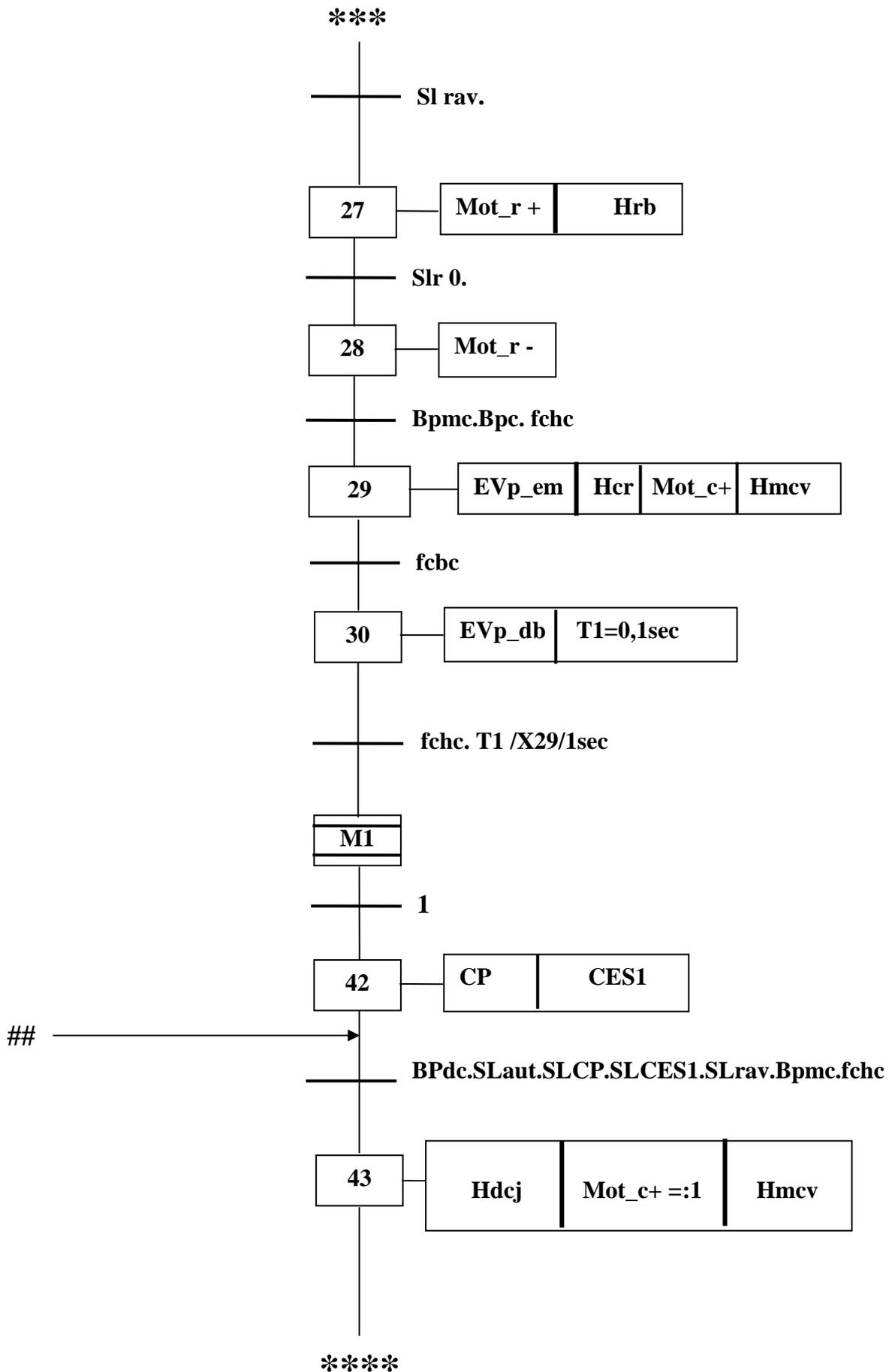
**Figure II.15 :** *grafcet niveau 2 de la pompe*

**4.2.3 Grafcet de l'installation (Figure 16)**

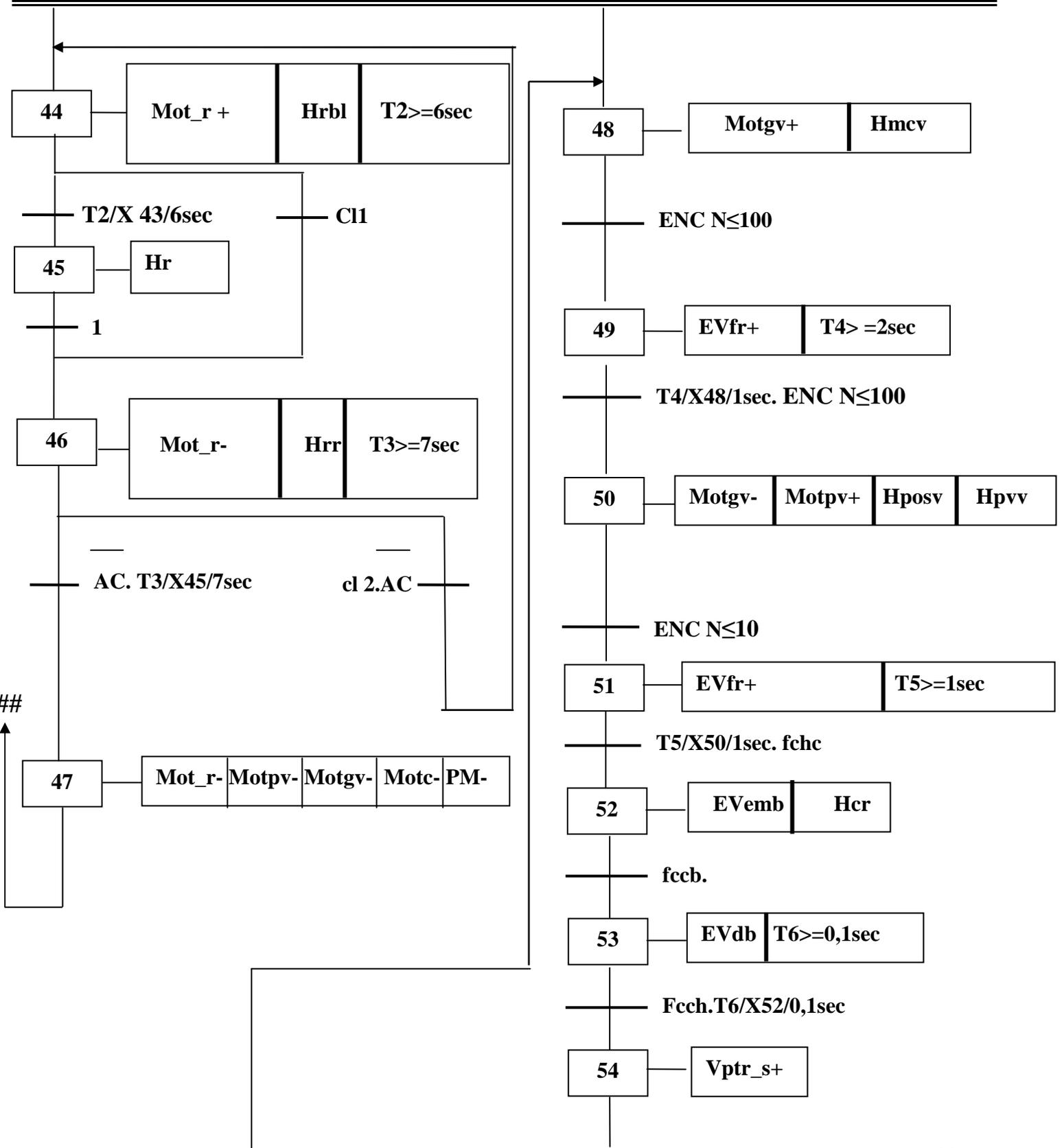
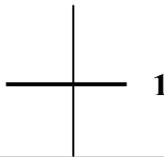








\*\*\*\*



\*\*\*\*\*

\*\*\*\*\*

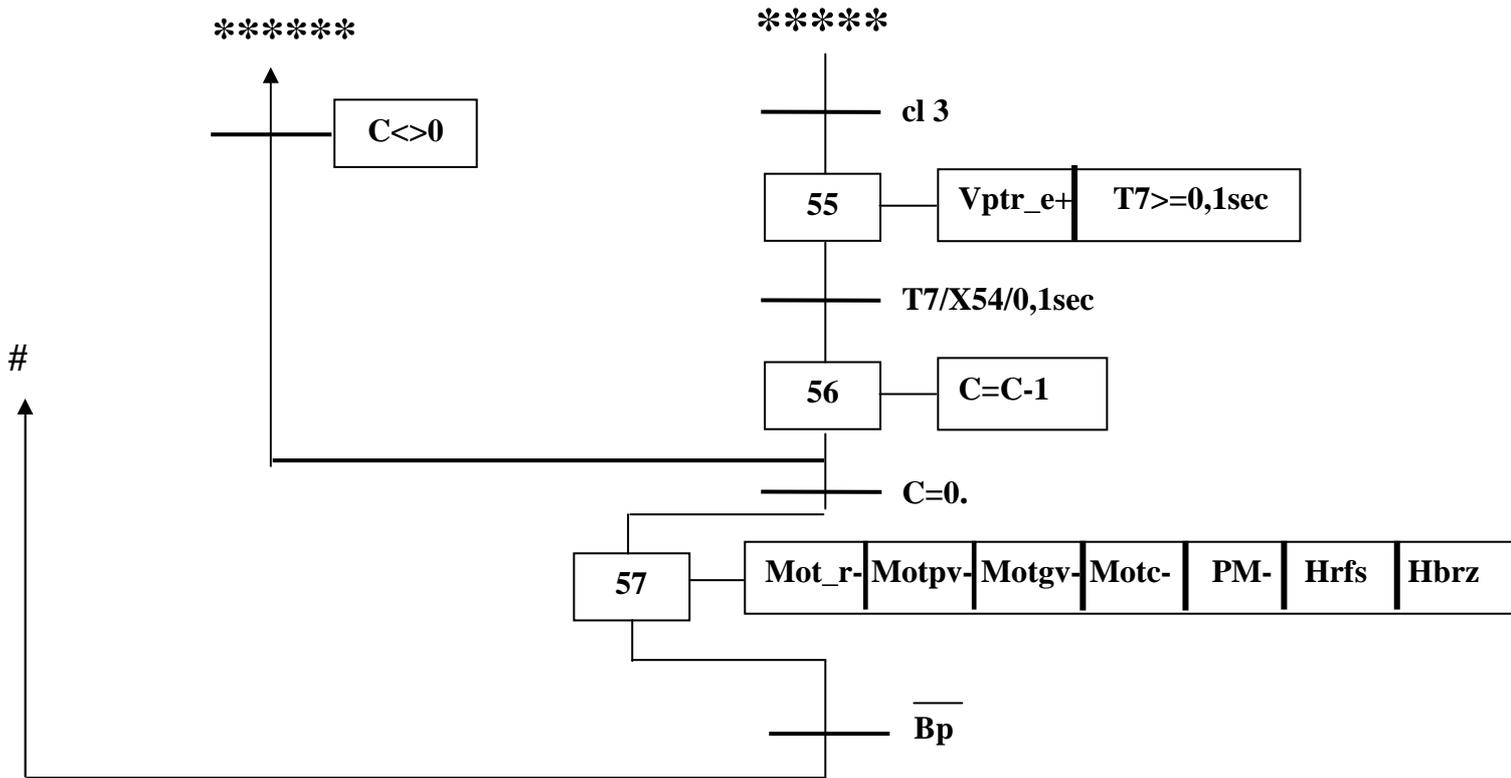


Figure II.16: grafcet niveau 2 de l'installation

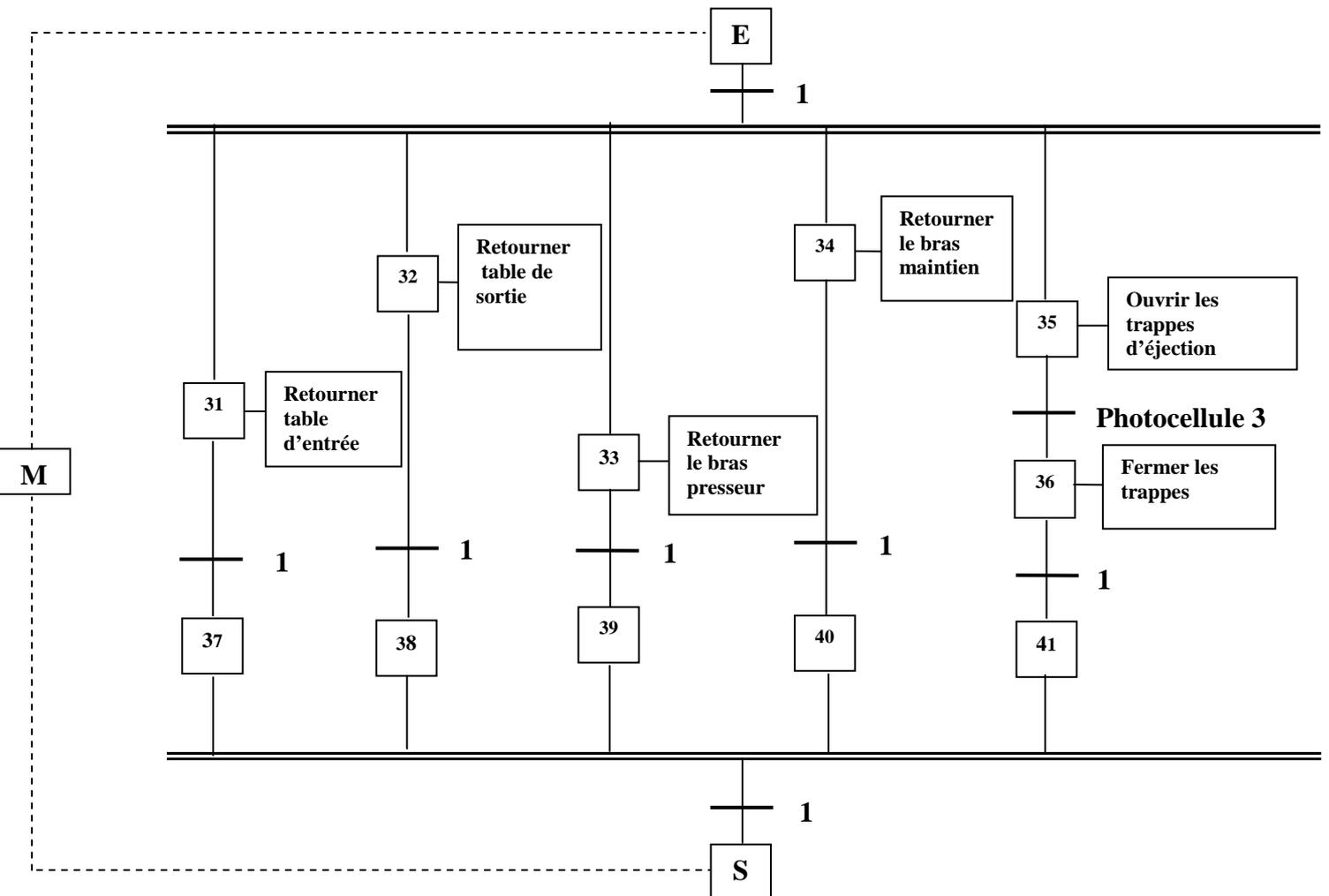


Figure II.17 : grafcet niveau 1 de la macro étape M1

### **5. Conclusion**

En tenant compte de la complexité du processus ainsi que des contraintes imposées par l'entreprise, nous avons modélisé le procédé de commande à l'aide du GRAFCET. Nous avons élaboré en premier lieu un grafcet de niveau 1 pour expliquer le système, puis le grafcet niveaux 2 qui met en œuvre et décrit la partie opérative. Ce dernier est utilisé pour la réalisation ou le dépannage des systèmes automatisés.

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

Ainsi, le GRAFCET a facilité considérablement le passage de la description à la modélisation, et nous permettra au chapitre suivant d'aborder la programmation de la partie opérative qui pilotera le procédé et ce à l'aide du STEP7.

# Chapitre III

***Programmation de  
l'installation à l'aide  
de l'API S7-300***

## **1. Introduction**

L'automate programmable industriel (API) est le constituant de base des équipements automatisés. Il est apparu vers les années soixante-dix aux Etats Unis. Initialement, il est destiné aux traitements des signaux logiques tout ou rien (TOR), et la réalisation des fonctions logiques combinatoires et séquentielles, en remplaçant des armoires à relais trop volumineuses.

L'automate programmable est un micro-ordinateur spécifique, qui se distingue des autres micro-ordinateurs par plusieurs caractéristiques :

- Il est conçu pour fonctionner dans les ambiances industrielles qui peuvent être sévères.
- Il peut gérer un grand nombre des signaux d'entrées/sorties.

## **2. Automate programmable**

### **2.1. Définition**

L'automate programmable est un appareil électrique de traitement logique d'informations, dont le programme de fonctionnement est effectué à partir d'instructions établies en fonction du processus à réaliser. Et de cette dernière nous distinguerons que l'automate joue un rôle de commande et de communication.

### **2.2. Architecture et fonctionnement d'un automate**

La structure de base d'un automate programmable comprend plusieurs ensembles fonctionnels (figure III.1).



**Figure III.1:** *photographie d'un automate programmable.*

L'API se compose de trois grandes parties (figure III.2)

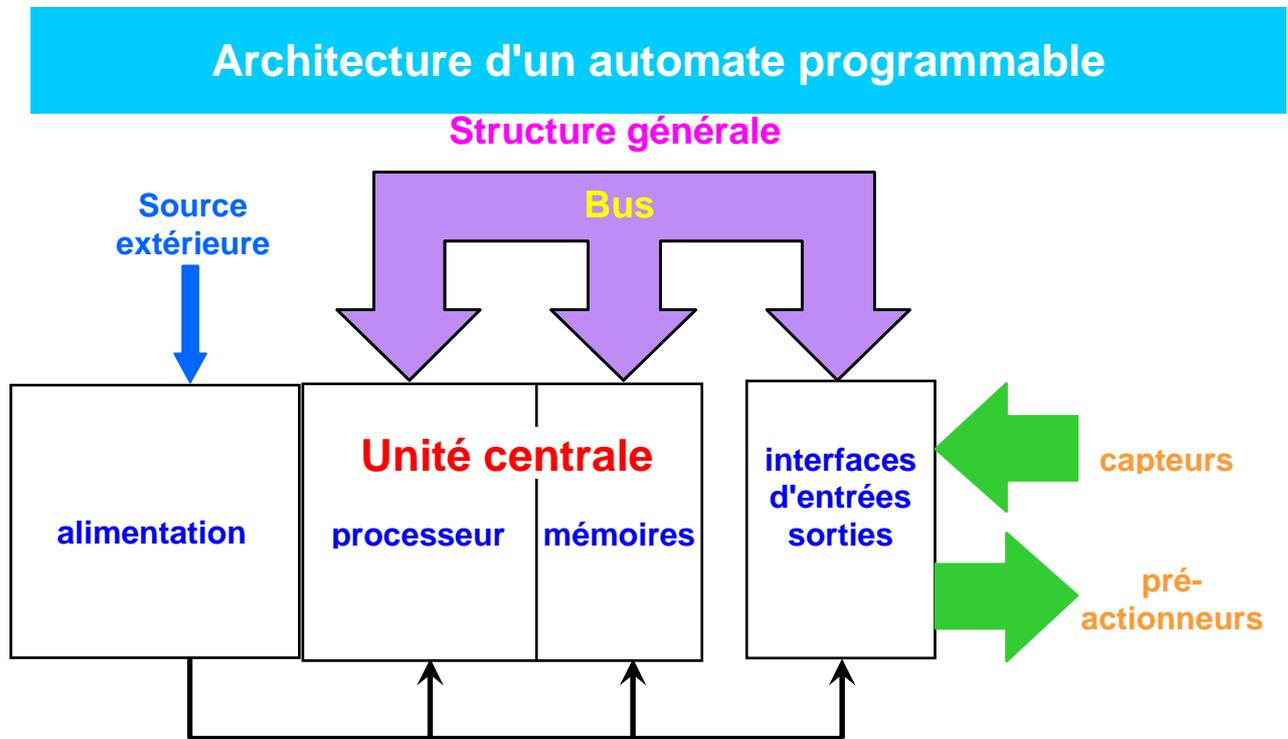


Figure III .2 : architecture d'un automate programmable.

### a) Le microprocesseur

Le microprocesseur réalise toutes les fonctions logiques (ET, OU), les fonctions de temporisation, de comptage, de calcul, etc. Il est connecté aux autres éléments (mémoire et interface d'entrée /sortie E/S) par BUS.

### b) La zone mémoire

Elle permet de :

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

### **c) Les interfaces d'entrées/sorties**

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

### **2.3. Choix d'un automate**

Le choix d'un automate n'est pas arbitraire, mais il se fait après que l'utilisateur a 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.

Vue le nombre des entrées /sorties de notre système nous avons choisit l'automate S7-300, car ces caractéristique 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.

## **3. L'automate S7-300**

### **3.1. Constitution du S7-300**

L'automate programmable S7-300 (**figure III.4**) 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.

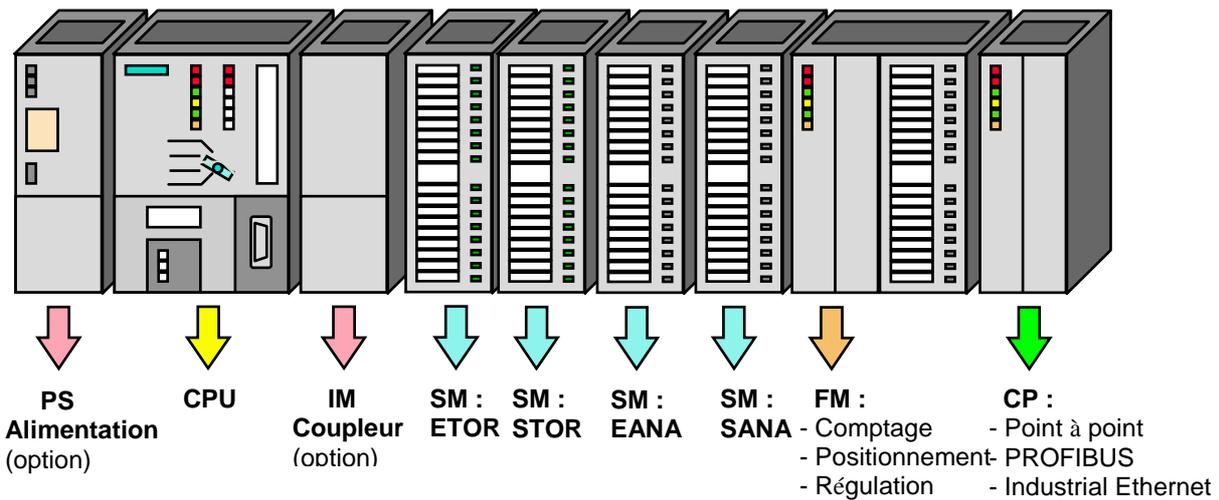


Figure III.4: les différents composants de l'automate programmable S7-300

### 3.1.1. 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.

### 3.1.2. 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 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 tel 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, allumé : les 5v sont présentes, clignote : surcharge 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.
- 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.

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

### **3.1.3. 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.

#### **3.1.4. 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 tel que les boutons poussoirs, les fins de course. Ils conviennent au raccordement d'appareils à contacts et de détecteurs de proximité en montage.
- **Modules de sorties TOR :** les modules de sorties TOR transportent le niveau de signal interne du S7-300 au niveau de 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.

#### **3.1.5. Module de fonction (FM)**

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

#### **3.1.6. 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 :

- Point à point.
- PROFIBUS.
- Industriel Ethernet

### **3.1.7. Châssis d'extension (UR)**

Il est constitué d'un profilé support en aluminium et 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.

## **3.2 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 fini 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 exécuter le système d'exploitation, la CPU passe à l'exécution du programme utilisateur, qui contient la liste d'instruction à exécuter pour faire fonctionner le procédé.

### **c) Commande du processus**

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

## **3.3 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.5**). STEP7 offre les fonctions suivantes pour l'automatisation d'une installation :

- Configuration et paramétrage du matériel.
- La création des programmes.

- Test, mise en service et maintenance de l'installation d'automatisation.
- Fonctions de diagnostic et d'exploitation lors de perturbation dans l'installation.

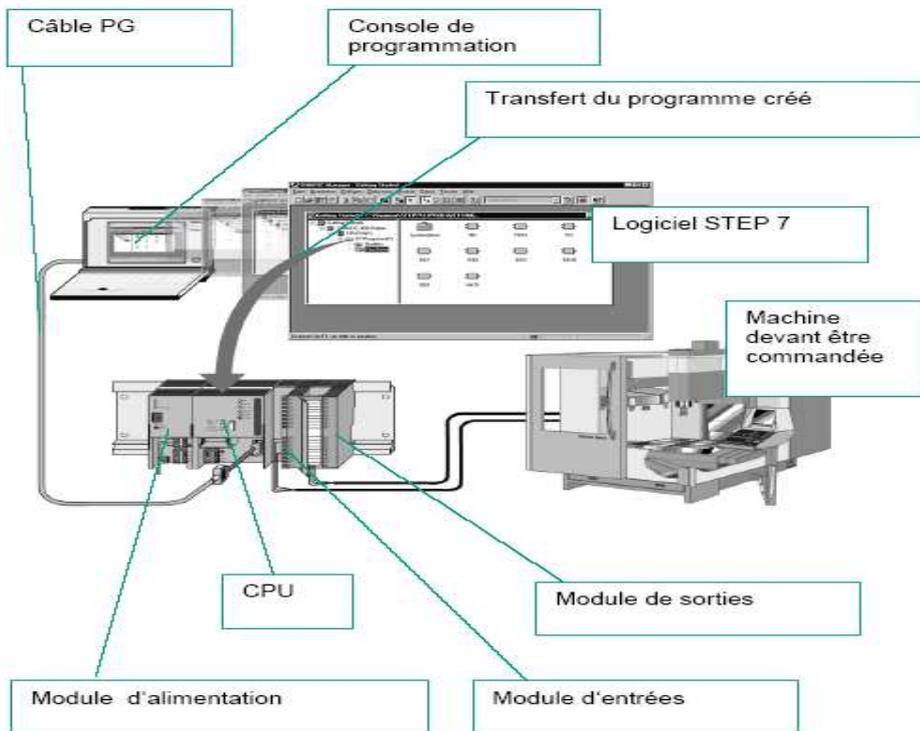


Figure III.5: vue d'ensemble de l'automatisme.

### 3.3.1. Langage de programmation

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

- schéma à contacts « CONT » ;
- logigramme « LOG » ;
- liste d'instructions « LIST ».

❖ **Schéma à contacts (CONT) :** Le schéma à contacts est une représentation graphique (figure III.6) de la tâche d'automatisation ayant recours aux symboles. C'est un langage des habitués des schémas électriques.

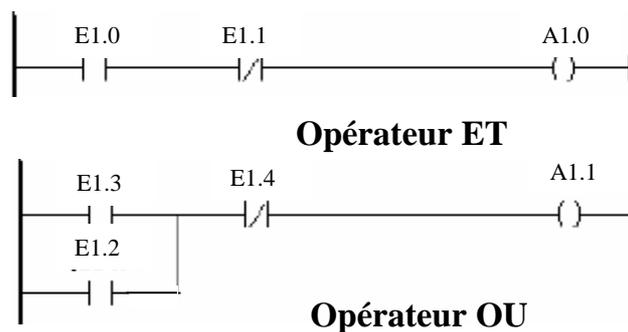


Figure III.6: représentation des opérations ET, OU par CONT

❖ **Logigramme (LOG)**

Le logigramme est une représentation graphique ayant recours aux symboles de la logique (Figure III.7). Les différentes fonctions y sont représentées par un symbole avec indicateur de fonction. Les entrées sont disposées à gauche du symbole, les sorties à droite de ce dernier.



**Figure III.7:** représentation des opérations ET, OU par LOG

❖ **Liste d'instructions (LIST)**

La tâche d'automatisation est écrite dans la liste d'instruction à l'aide des différentes instructions (**figure III.8**). C'est un langage qui s'apparente au langage machine.

<b>U E 1.0</b>	<b>O E 1.2</b>
<b>U E 1.1</b>	<b>O E1.3</b>
<b>= A 1.0</b>	<b>ON E1.4</b>
	<b>= A 1.1</b>

**Figure III.8:** représentation des opérations ET, OU par LIST.

**Remarque**

- Les programmes d'automatisation programmés en CONT ou LOG sont en principe toujours traduisibles en LIST.
- Dans la mémoire de programme de l'automate, le programme est toujours stocké en LIST car il est plus exact en langage machine.

**3.3.2. 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) ;
- fonctions (FC) ;
- blocs de données (DB) ;
- des blocs souvent utilisés en relevant du système tel que les blocs fonctionnels système (SFB) et les fonctions systèmes (SFC) sont intégrées 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 clique 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.9** )



Figure III.9 : assistant de STEP 7 'nouveau projet'

c).le choix de la CPU

En cliquant sur l'icône suivant, la fenêtre suivante apparaît, (figureIII.10) elle nous permet de choisir la CPU .Dans notre cas nous choisi la CPU 314.

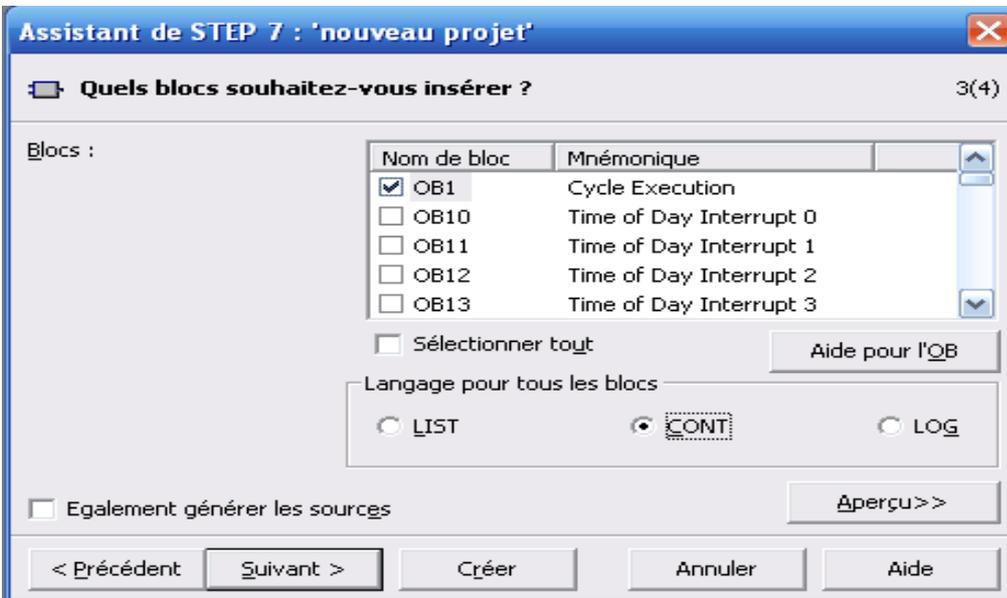


Figure III .10 : fenêtre de choix de la CPU.

**d). Choix des blocs**

Après validation de la CPU, une autre fenêtre apparaît, (**FigureIII.11**) permet de choisir les blocs à insérer, et choisir le langage de programmation (LIST, CONT, LOG).

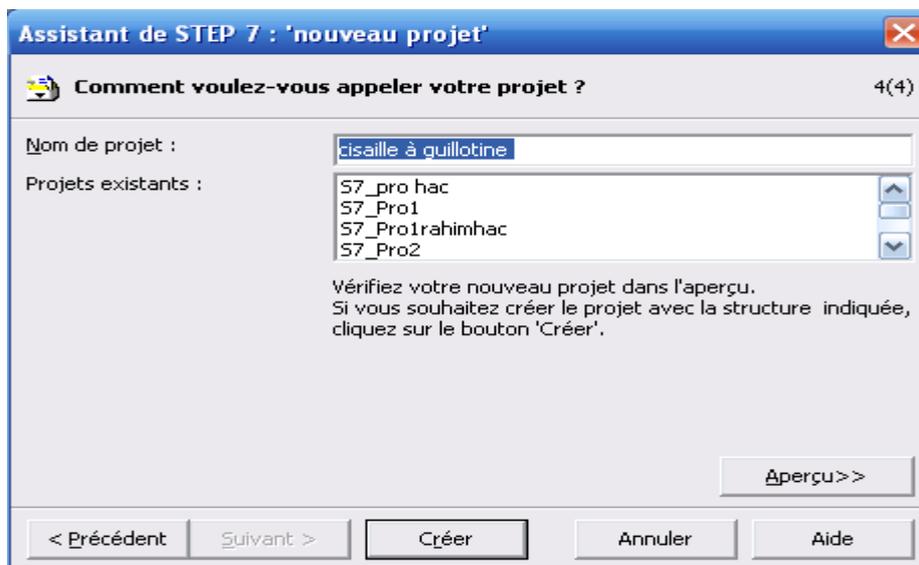
Pour notre projet nous avons choisi l'OB1 (cycle d'exécution) et le langage à contact



**Figure III.11 :** choix des blocs à utilisés et de langage.

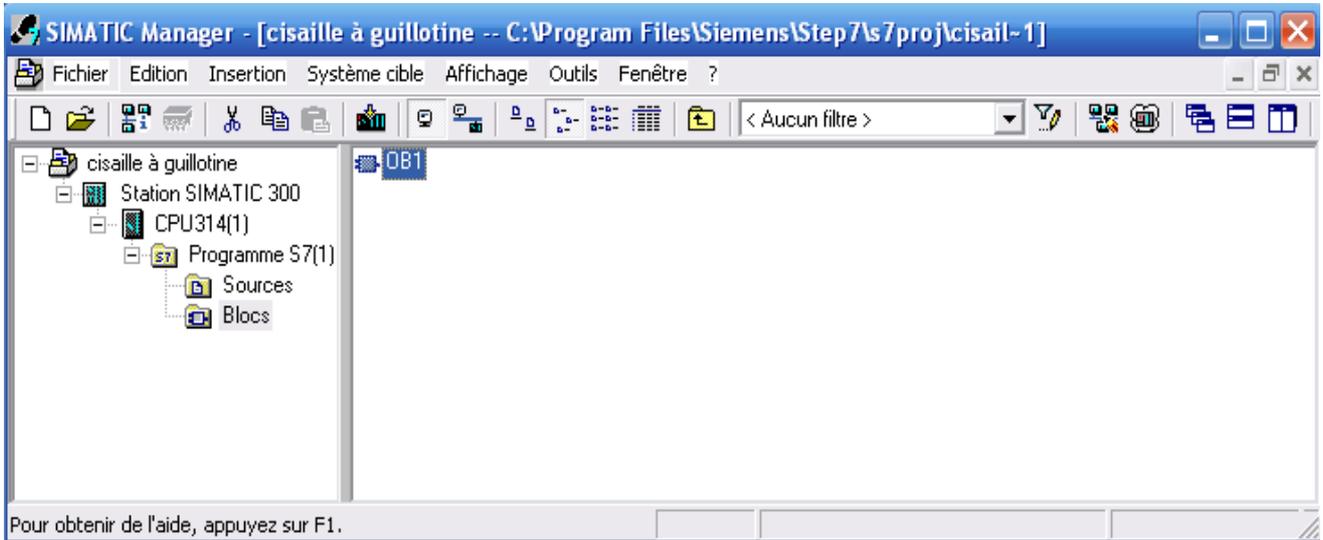
**e) nomination du projet crée**

En cliquant sur suivant, la fenêtre de (**figureIII.12**) apparaît pour permettre de nommer le projet.



**Figure III.12 :** nomination du projet.

En cliquant sur créer, la fenêtre suivante apparaît (Figure III.13) :



**Figure III.13** : *vue des composants de notre projet S7.*

### **3.3.3. Configuration du matériel 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.14) :

- Un rack.
- Le module d'alimentation PS 307 5A .
- La CPU 314 .
- Deux (02) modules d'entrées logiques de 32 bits .
- Deux (02) modules de sortie logique de 32 bits.

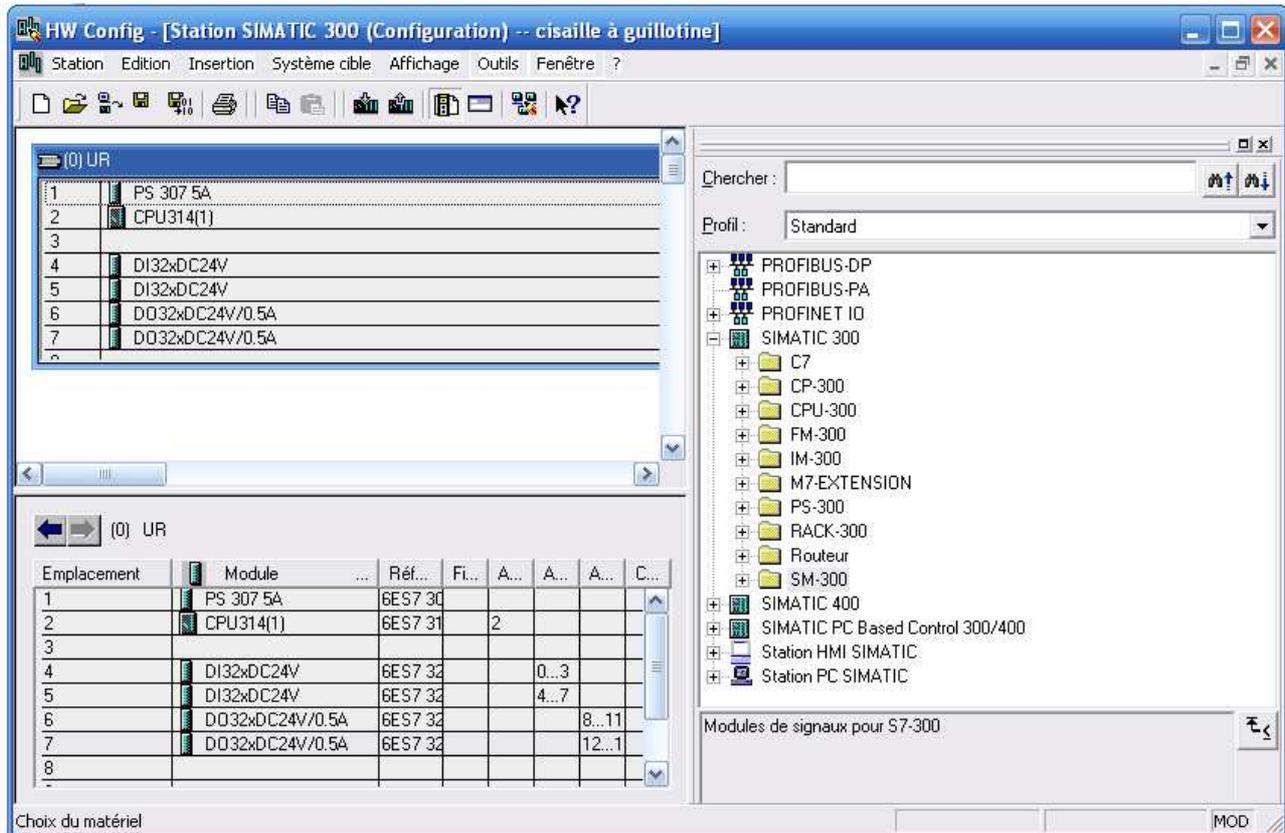


Figure III.14 : configuration matérielle de notre automate.

Sauvegardons cette configuration en cliquant sur l’item Enregistrer du menu déroulant Fichier.

### 3.3.4. Création de la table des mnémoniques

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

L’objet “mnémoniques“ (table des mnémoniques) est automatiquement crée 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 :

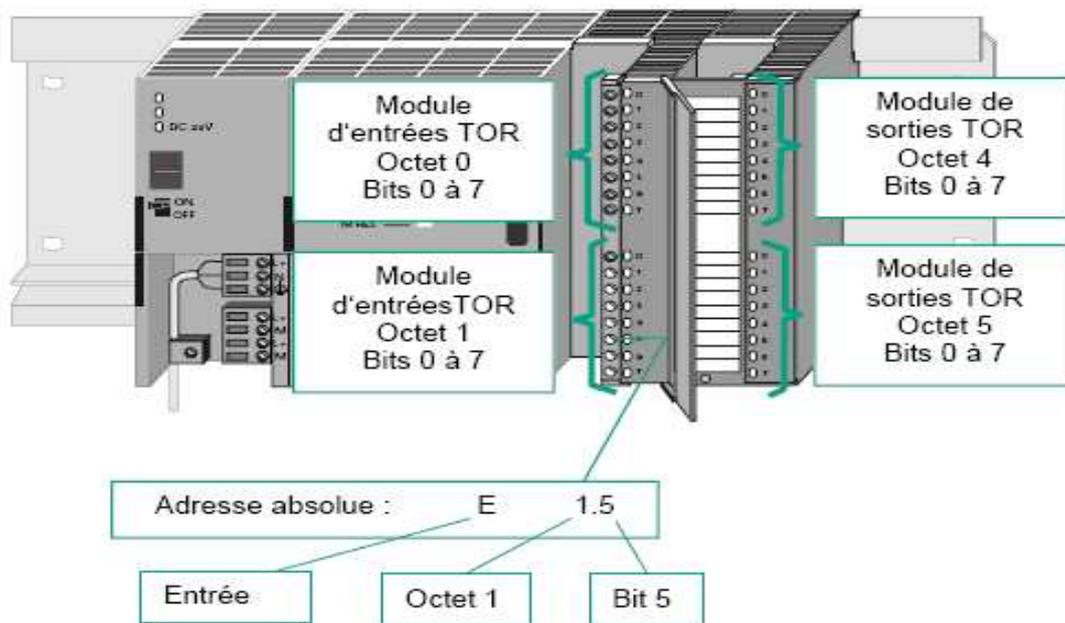
1. Nous cliquons deux fois sur le programme S7 dans la fenêtre du projet afin que l’objet “mnémoniques“ s’affiche dans la partie droite de la fenêtre.
2. Si la table des mnémoniques a été effacée ou doit être écrasée, nous pouvons insérer une nouvelle en choisissant la commande **insertion>Table des 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.

### 3.3.5 Adressage des signaux d'entrées/sorties

La déclaration d'une entrée ou sortie donnée à l'intérieur du programme s'appelle l'adressage. Les entrées/sorties des automates sont regroupés 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.(figureIII.14)



FigureIII.14: adressage d'entrées/sorties (TOR).

### 3.3.6 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 bites de mémentos).

### 3.3.7 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 de 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, tel 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.

### 3.4 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.

#### 3.4.1 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 mémoires.
- La communication avec les consoles de programmation et autres périphériques de communication.
- L'enregistrement des alarmes et l'appel des blocs d'alarmes.

#### 3.4.2 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 est essentiellement constitué de blocs.

### 3.5 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.
- Facilite de la mise en service.

#### 3.5.1. 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'autre 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 affectées à un FB donné.

### **3.5.2 Blocs système pour fonctions standard et fonction système**

Les blocs systèmes sont des fonctions prêtes à l'emploi stockés dans la CPU. Ces blocs peuvent être appelés par l'utilisateur et utiliser dans le programme.

- **Bloc fonctionnel système (SFB) :** bloc fonctionnel stocké dans le système d'exploitation de la CPU et pouvant être appelé par l'utilisateur.
- **Fonction système (SFC) :** fonction stockée dans le système d'exploitation de la CPU et pouvant être appelée par l'utilisateur.
- **Donnée système (SDB) :** zone de mémoire dans le programme configuré par différentes applications de STEP 7, pour le stockage des données dans le système d'automatisation.

## **3.6 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.15**).

### **3.6.1 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.

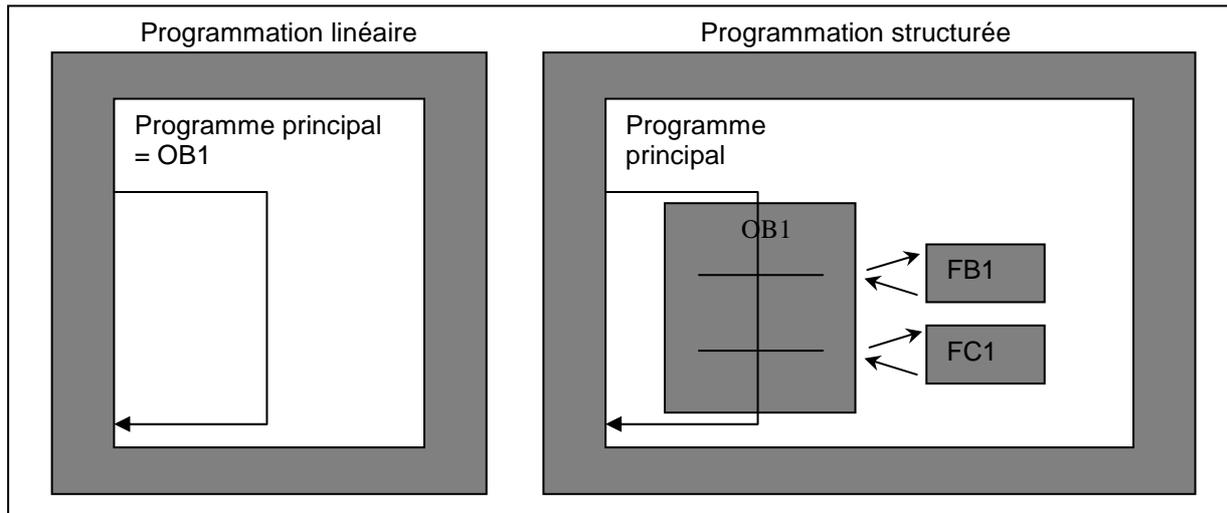
Le traitement linéaire du programme est généralement utilisé pour des commandes simples, de volume peu important et peut être réalisé au moyen d'un seul bloc d'organisation (OB).

### **3.6.2 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.15 : schéma illustrant le traitement du programme par la CPU**

### **3.7 Différents blocs d'organisation des S7-300**

Les blocs d'organisations (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 de programme telle que la mise en route de la CPU ou bien quand une erreur intervient.

Les différents blocs de l'organisation sont:

#### **a) 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) Bloc d'organisation pour traitement des erreurs**

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

**c) Bloc d'organisation de programme déclenché par alarme**

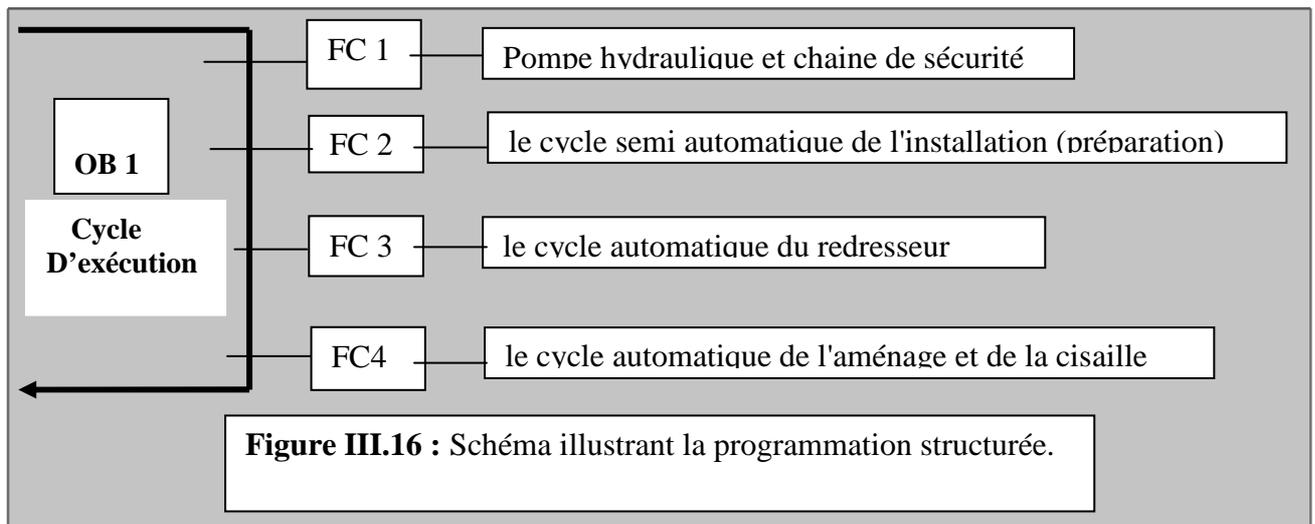
Le STEP 7 met à la disposition du programmeur différents types d'OB qui interrompt le traitement de l'OB1, à des intervalles de temps précis ou à des apparitions d'événement donnés.

**d) 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 :

- OB de démarrage OB100.
- OB de redémarrage OB101.

**✚ Notre structure de programmation**



**3.8 Implantation du GRAFCET dans le S7-300**

Les automates programme disposent pas tous d'un langage GRAFCET ce qui est le cas avec l'automate S7- 300.

Cependant il existe toujours un moyen d'implanter le GRAFCET quel que soit le langage disponible, pour cela on doit suivre certaines étapes :

- Analyser et valider le grafcet de notre machine.
- Détermination des conditions d'activation et de désactivation des étapes du GRAFCET

(Tableau III.1) :

**Condition d'activation et de désactivation des étapes :**

étapes	Conditions d'activations (CA)	Conditions de désactivations (CD)
X0	$X57.\overline{BPmm}.\overline{Init.AUD}$	$X1.\overline{Init}+AUD$
X1	$X0.sl3br.\overline{rtbr}.\overline{slm}.\overline{Init.AUD}$	$X2+Init+AUD$
X2	$X1.\overline{fcbr}.\overline{Init.AUD}$	$X3+Init+AUD$
X3	$X2.sl2tbh.\overline{Init.AUD}$	$X4+Init+AUD$
X4	$X3.\overline{Init.AUD}$	$X5+Init+AUD$
X5	$X4.sl1.\overline{rtma}.\overline{Init.AUD}$	$X6+Init+AUD$
X6	$X5.\overline{prst}.\overline{Init.AUD}$	$X7+Init+AUD$
X7	$X6.sl2.\overline{Init.AUD}$	$X8+Init+AUD$
X8	$X7.sl3br.\overline{Init.AUD}$	$X9+Init+AUD$
X9	$X8.\overline{bt}.\overline{rtbr}.\overline{Init.AUD}$	$X10+Init+AUD$
X10	$X9.\overline{Init.AUD}$	$X11+Init+AUD$
X11	$X10.sl4brp.\overline{Init.AUD}$	$X12+Init+AUD$
X12	$X11.\overline{fcbrp}.\overline{Init.AUD}$	$X13+Init+AUD$
X13	$X12.\overline{lvbrm}.\overline{Init.AUD}$	$X14+Init+AUD$
X14	$X13.\overline{fcbrm}.\overline{Init.AUD}$	$X15+Init+AUD$
X15	$X14.\overline{lvteh}.\overline{Init.AUD}$	$X16+Init+AUD$
X16	$X15.\overline{lvte-0}.\overline{Init.AUD}$	$X17+Init+AUD$
X17	$X16lh-sit.\overline{Init.AUD}$	$X18+Init+AUD$

X18	X17.lb-sit. $\overline{\text{Init.AUD}}$	X19+ Init+AUD
X19	X18.lvrh. $\overline{\text{Init.AUD}}$	X20+ Init+AUD
X20	X19.sl5.rtma $\overline{\text{Init.AUD}}$	X21+ Init+AUD
X21	X20.lvrh. $\overline{\text{Init.AUD}}$	X22+ Init+AUD
X22	X21.sl5-0. $\overline{\text{Init.AUD}}$	X23+ Init+AUD
X23	X22.lvtsh. $\overline{\text{Init.AUD}}$	X24+ Init+AUD
X24	X23.slavrd. $\overline{\text{Init.AUD}}$	X25+ Init+AUD
X25	X24.slrld-0. $\overline{\text{Init.AUD}}$	X26+ Init+AUD
X26	X25. $\overline{\text{Init.AUD}}$	X27+ Init+AUD
X27	X26.slavrd. $\overline{\text{Init.AUD}}$	X28+ Init+AUD
X28	X27.slrld-0. $\overline{\text{Init.AUD}}$	X29+ Init+AUD
X29	X28.bpc.fcch. $\overline{\text{Init.AUD}}$	X30+ Init+AUD
X30	X29.fccb. $\overline{\text{Init.AUD}}$	X31+X32+X33+X34+X35+ Init+AUD
X31	X30.fcch.T1/X30/2sec. $\overline{\text{Init.AUD}}$	X37+ Init+AUD
X32	X30.fcch.T1/X30/2sec. $\overline{\text{Init.AUD}}$	X38+ Init+AUD
X33	X30.fcch.T1/X30/2sec. $\overline{\text{Init.AUD}}$	X40 +Init+AUD
X34	. X30.fcch.T1/X30/2sec $\overline{\text{Init.AUD}}$	X39+Init+AUD
X35	X30.fcch.T1/X30/2sec $\overline{\text{Init.AUD}}$	X36+ Init+AUD
X36	X35.cl3. $\overline{\text{Init.AUD}}$	X41+ Init+AUD
X37	X31+ $\overline{\text{Init.AUD}}$	X42+ Init+AUD

X38	$X32 + \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X42+ Init+AUD
X39	$X34 + \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X42+ Init+AUD
X40	$X33 + \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X42+ Init+AUD
X41	$X36 + \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X42+ Init+AUD
X42	$\frac{(X37+X38+X39+X40+X41) + \text{Init+AUD}}$	X43+ Init+AUD
X43	$\frac{((X42.\text{bdc}.\text{stau}.\text{stpc}.\text{stces1}) + X47) \cdot \overline{\text{Init}} \cdot \overline{\text{AUD}}}{\overline{\text{Init}} \cdot \overline{\text{AUD}}}$	(X44+X48)+ Init+AUD
X44	$(X43+X46.\text{cl2}.\overline{\text{ac}}) \cdot \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X45+ Init+AUD
X45	$X44. (\text{cl1}+T2 \geq 6\text{sec}) \cdot \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X46+ Init+AUD
X46	$X45. \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X47+ Init+AUD
X47	$\frac{X46.\overline{\text{ac}}.T3/X40/7\text{sec}}{\overline{\text{Init}} \cdot \overline{\text{AUD}}}$	X43+ Init+AUD
X48	$X43+X56.\text{c} < 0. \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X49+ Init+AUD
X49	$X48.\text{ENC } N \leq 400 \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X50+ Init+AUD
X50	$X49.T4/X48/1\text{sec} \text{ENC} \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X51+ Init+AUD
X51	$X50.\text{ENC} \geq 495. \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X52+ Init+AUD
X52	$X51.T5/X50/1\text{sec}.\text{fchc} \cdot \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X53+ Init+AUD
X53	$X52.\text{fccb} \cdot \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X54+ Init+AUD
X54	$X53.\text{fch}.T6/X52/0,1\text{sec} \overline{\text{Init}} \cdot \overline{\text{AUD}}$	X55+ Init+AUD

X55	X54.c13. $\overline{\text{Init.AUD}}$	X56+ Init+AUD
X56	X55.T7/X54/0,1sec. $\overline{\text{Init.AUD}}$	X57+ Init+AUD
X57	X56.C=0 $\overline{\text{Init.AUD}}$	X0+ Init+AUD

**Conditions d'activations des actions**

<b>Actions</b>	<b>Conditions d'activations des actions</b>
<b>Mot Brav+</b>	(X1). $\overline{\text{AUDdoux}}$
<b>Mot Brar+</b>	X7. $\overline{\text{AUDdoux}}$
<b>Mot_d+</b>	X19. $\overline{\text{AUDdoux}}$
<b>Mot_r +</b>	( X26+X43). $\overline{\text{AUDdoux}}$
<b>Mot_c+</b>	(X28+X42). $\overline{\text{AUDdoux}}$
<b>Motgv+</b>	X47. $\overline{\text{AUDdoux}}$
<b>Motpv+</b>	X49. $\overline{\text{AUDdoux}}$
<b>Vpte_e+</b>	X30. $\overline{\text{AUDdoux}}$
<b>Vpte_s+</b>	(X3+X14). $\overline{\text{AUDdoux}}$
<b>Vm+</b>	X5. $\overline{\text{AUDdoux}}$
<b>Vhbp_e+</b>	X33. $\overline{\text{AUDdoux}}$
<b>Vhbp_s+</b>	(X10). $\overline{\text{AUDdoux}}$
<b>VhBrm_e+</b>	X32. $\overline{\text{AUDdoux}}$
<b>Vpro_s+</b>	X16. $\overline{\text{AUDdoux}}$

<b>Vpro_e+</b>	(X20). $\overline{\text{AUDdoux}}$
<b>VI_e+</b>	X18. $\overline{\text{AUDdoux}}$
<b>VI_s+</b>	(X17). $\overline{\text{AUDdoux}}$
<b>Vpts_e+</b>	X31. $\overline{\text{AUDdoux}}$
<b>Vpts_s+</b>	(X22). $\overline{\text{AUDdoux}}$
<b>Vptr_e+</b>	(X35+X54). $\overline{\text{AUDdoux}}$
<b>Vptr_s+</b>	(X34+X53). $\overline{\text{AUDdoux}}$
<b>EVp_em</b>	( X28+X51). $\overline{\text{AUDdoux}}$
<b>EVp_db</b>	(X29+X52). $\overline{\text{AUDdoux}}$
<b>EVfr+</b>	(X48+X50). $\overline{\text{AUDdoux}}$
<b>Cmp</b>	X55. $\overline{\text{AUDdoux}}$

**Tableau III. 1 :** *les conditions d'activation et de désactivations.*

- Définition des exigences en matière de sécurité (les alarmes, AUD, aud,...etc.)
- Affectation des mementos aux variables intermédiaires, et des adresses effectives et formelles aux entrées et sorties.
- Ecriture des équations des étapes.
- Traduction de ces équations en un programme séquentiel, en utilisant l'un des langages offert par le STEP 7 (LIST, CONT, LOG).
- Raccorder toutes les entrées, sorties et la PG à l'automate.
- Après avoir choisie la configuration matérielle et simuler le programme, nous procédons au chargement du projet dans la CPU physique.
- Après validation du projet, nous pouvons commencer la production.

## **4. Conclusion**

Dans ce chapitre nous avons présenté les différents modules de l'automate et ses langages de programmation.

Sa constitution modulaire, la facilité de réalisation d'architectures décentralisées et sa capacité de gestion des extensions importantes d'entrées/sorties font de lui une solution bénéfique pour les différentes tâches.

C'est grâce au STEP7 que le lien entre l'utilisateur et l'automate programmable S7-300 existe, car cet automate ne peut gérer ses fonctionnalités sans un programme approprié. Ainsi après avoir élaboré le programme nous allons passer à sa simulation et visualisation au chapitre suivant.

# Chapitre IV

***Simulation du  
programme avec  
S7-PLCSIM***

## 1. Introduction

Après l'élaboration du programme de commande de notre système à automatiser, 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. Qui 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. Ceci peut avoir 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 disposerons pas d'un automate.

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

Le mode de simulation est disponible à partir du gestionnaire de projet 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.

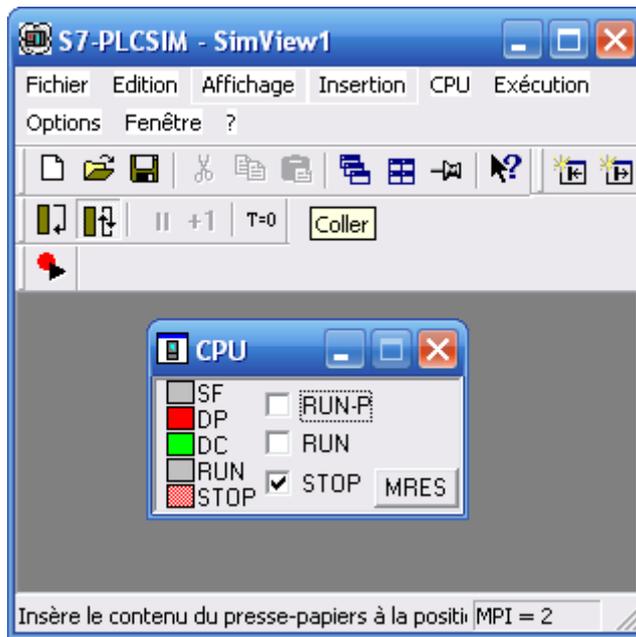
- 🗑 On ouvre le 'SIMATIC Manager' par un double clic sur son icône.



SIMATIC Manager.lnk

- ☛ Cliquez sur  ou sélectionnez la commande Outils-simulation de modules,

Cela lance l'application S7-PLCSIM et ouvre une fenêtre CPU (Figure IV):



**Figure IV .1 : fenêtre du S7-PLCSIM**

- ☛ Dans le gestionnaire de projets SIMATIC, chercher le projet-exemple projet presse Müller.
- ☛ Dans le projet exemple « projet Presse Müller », chercher le dossier blocs.
- ☛ Dans le gestionnaire de projet SIMATIC, cliquez sur  ou choisir la commande Système-cyber-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 :
- ☛ Cliquez sur  ou choisissez la commande Insertion-Entrée pour créer une fenêtre dans laquelle vous pouvez 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 peut modifier l'adresse (EB1, EB2...).
- ☛ Cliquez sur  ou choisissez la commande Insertion-Sortie pour créer une fenêtre dans laquelle vous pouvez 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...).

Cliquez sur  ou choisissez la commande Insertion-Temporisation pour créer une fenêtre dans laquelle vous pouvez 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).

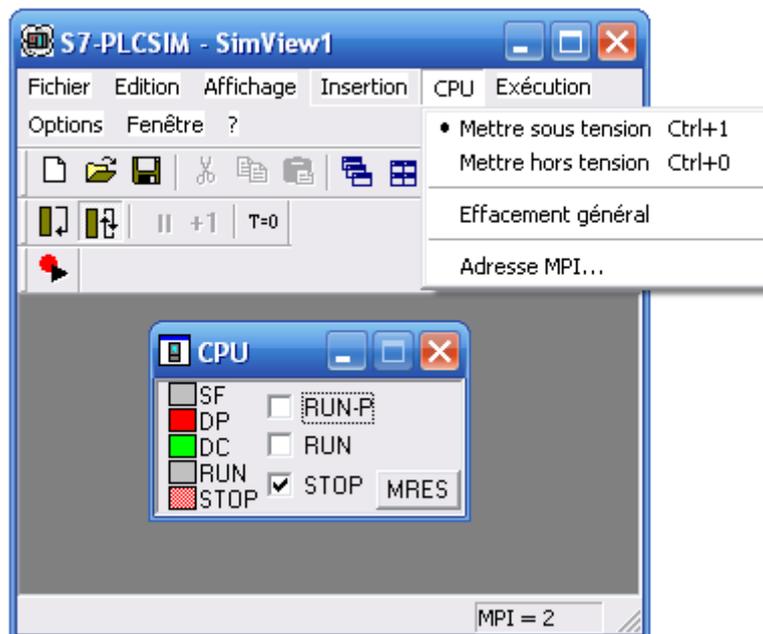
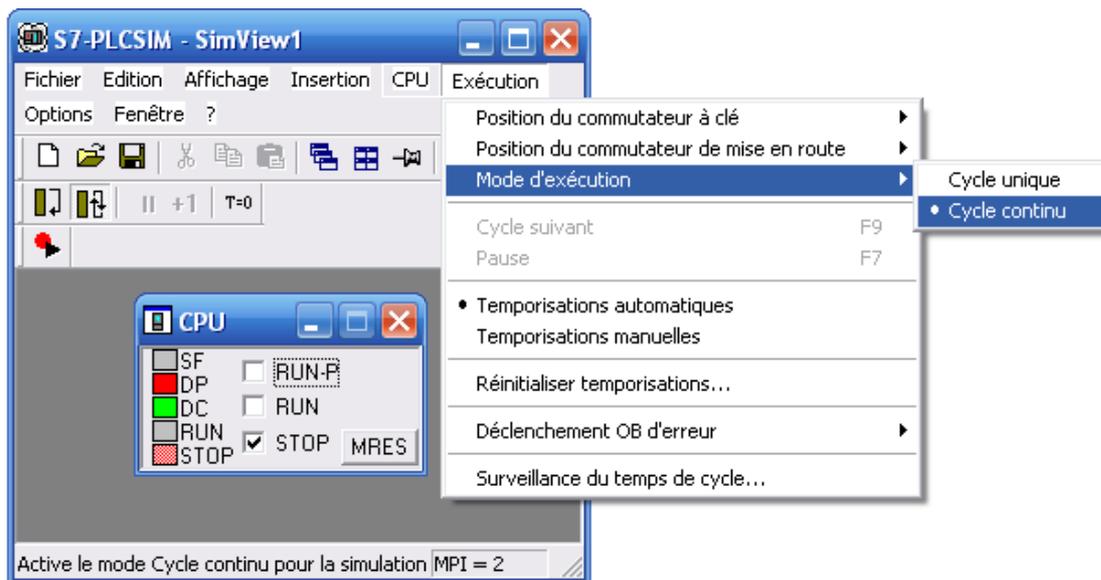


Figure IV.2 : Mise sous tension de la CPU

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



**Figure IV.2 :** *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 (Figure IV.5).

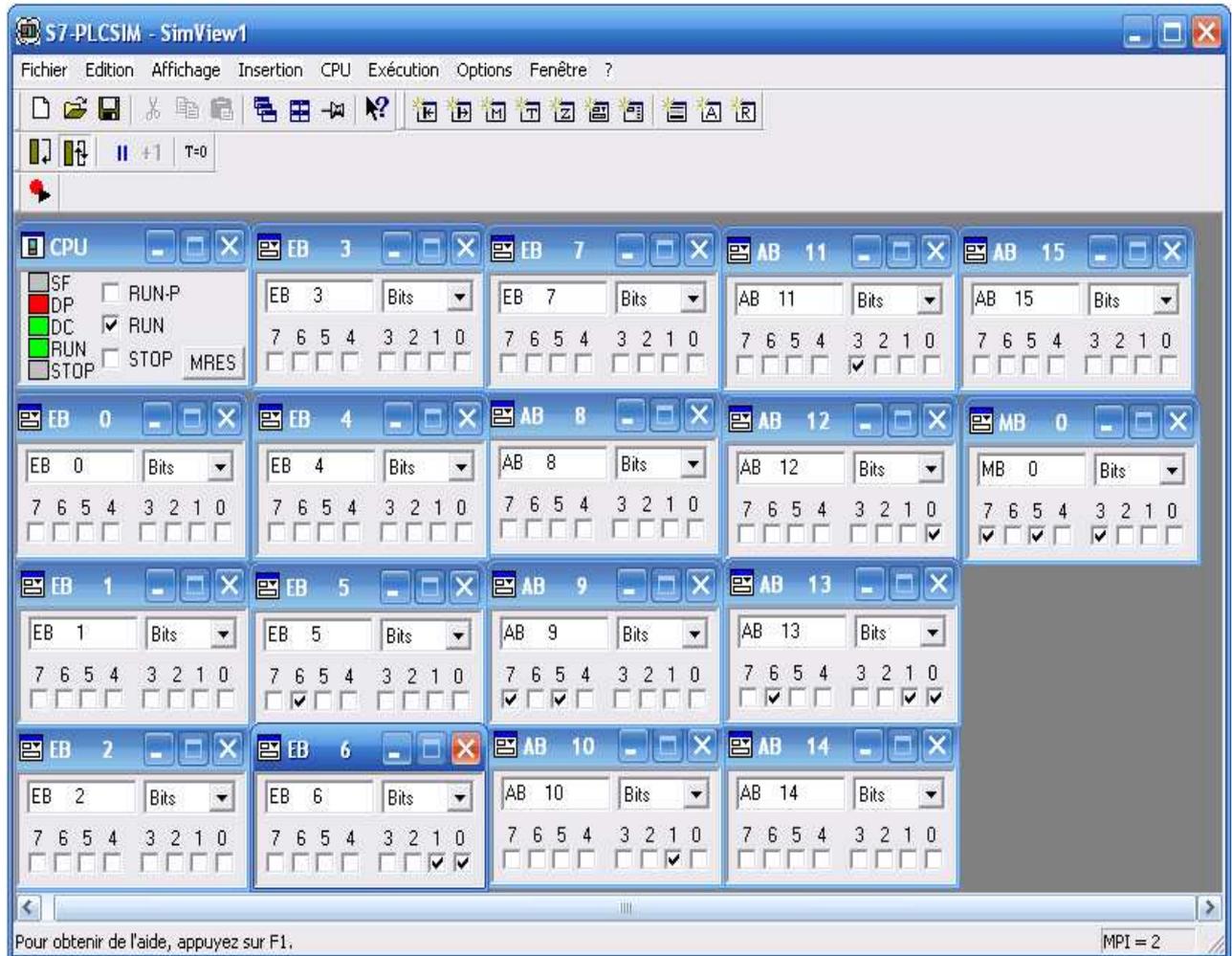


Figure IV.5 : simulateur S7-PLCSIM

### 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 sélectionnant la commande Test-Visualiser

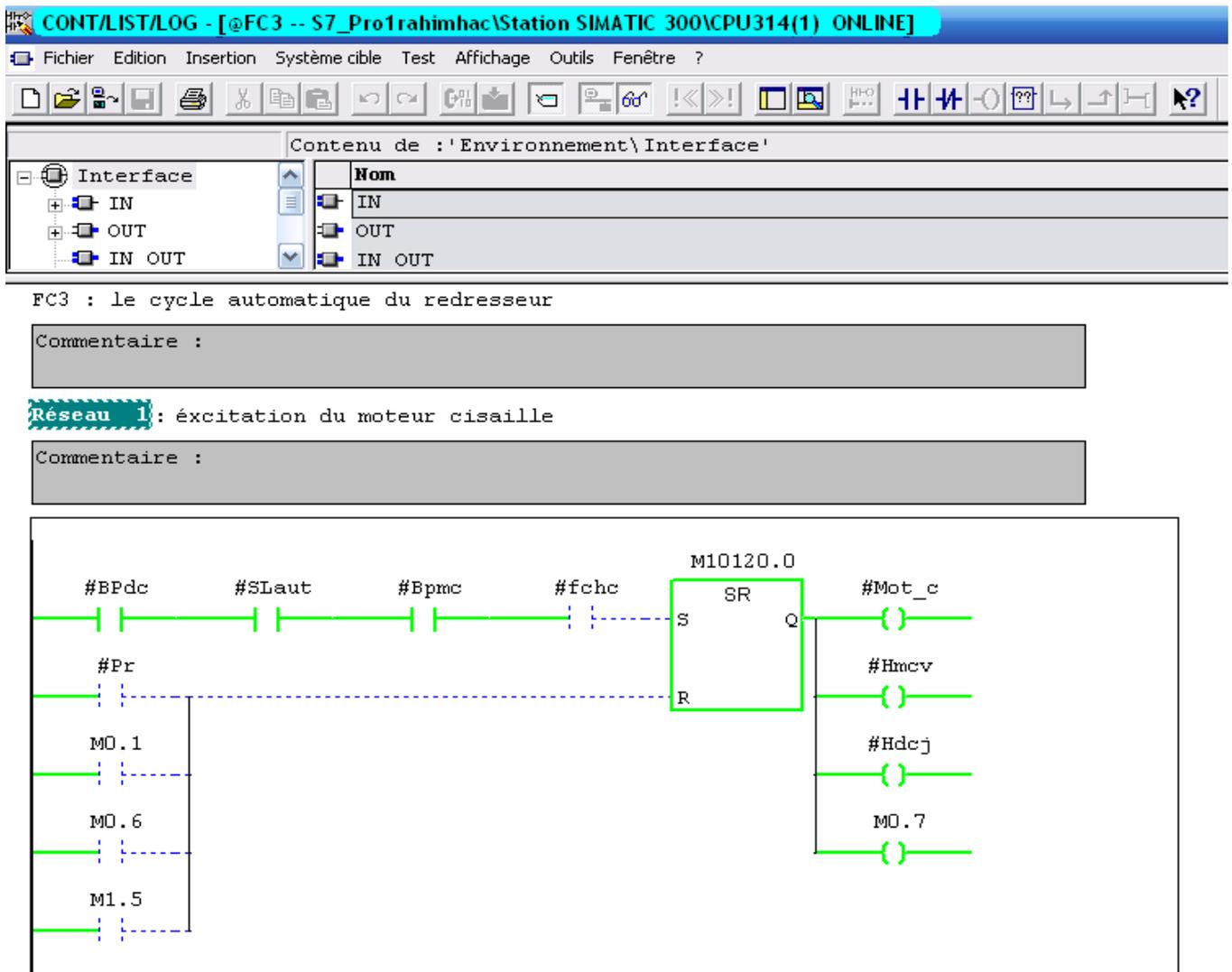


Figure IV.6 : Visualisation de l'état du programme

#### 4. Simulation du programme de l'installation réadaptée

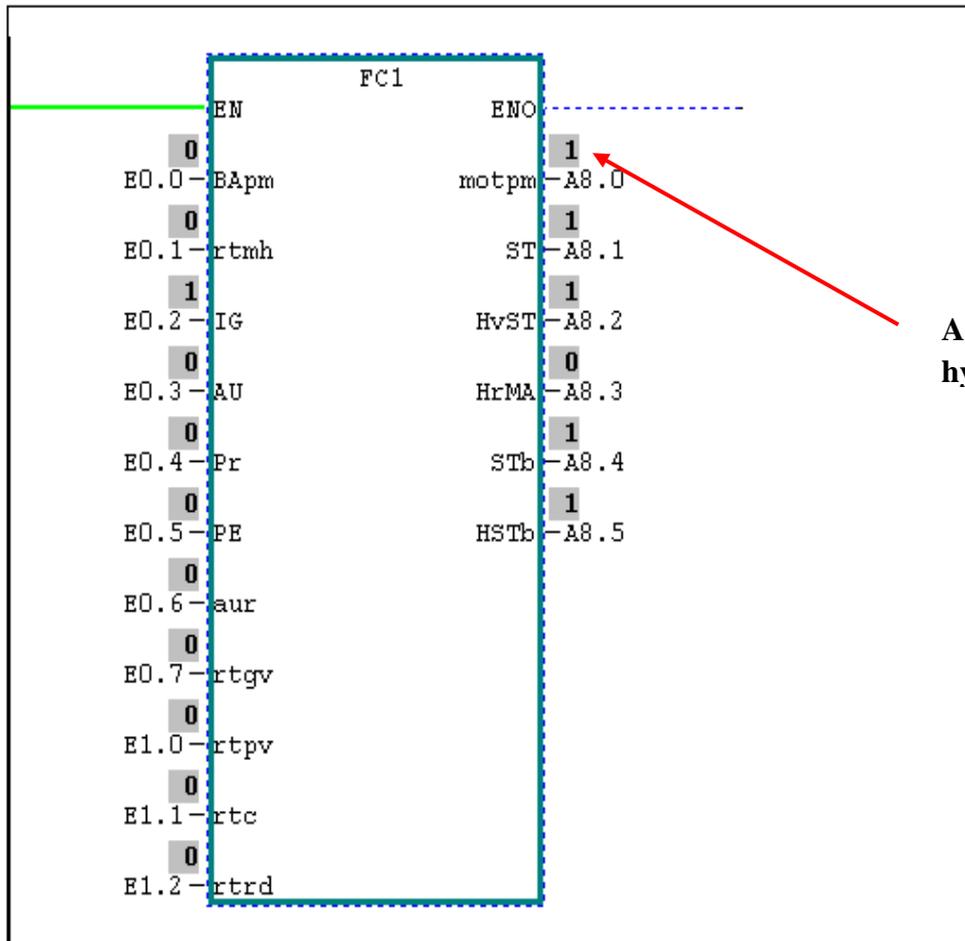
La simulation du programme de l'installation est faite en deux étapes :

- 🔧 Etape 1 : simulation du programme par bloc, c'est-à-dire charger chaque FC tout seul puis effectuer la simulation.
- 🔧 Etape 2 : simulation du cycle complet, c'est-à-dire charger tous les blocs FC puis effectuer la simulation du cycle.

**Exemple :** L'exemple suivant (Figure : IV.7) montre la simulation du programme qui commande la machine avec le logiciel de simulation S7-PLCSIM.

Réseau 1 : pompe hydrolique et chene de securité

Commentaire :



Activation de la pompe hydraulique

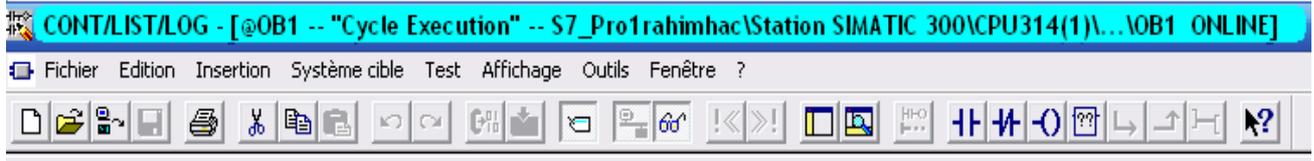


Réseau 2: le cycle semi automatique de l'installation (préparation)

Commentaire :

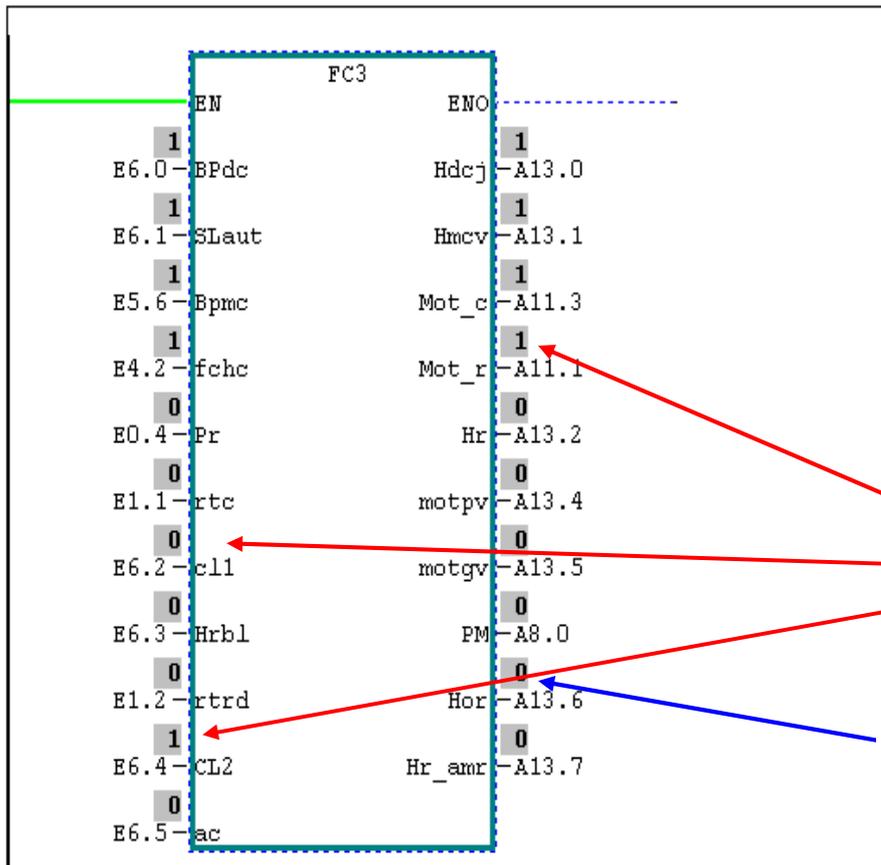
FC2		EN		ENO	
		1		0	
E1.3	s13brav		Motbrav	A8.6	
0			1		
E1.4	rtbr		Motbrar	A8.7	
0			0		
E1.5	s13br0		Vhtb_s	A9.0	
0			0		
E1.6	fcbr		Vm_s	A9.1	
0			0		
E0.3	AU		Vmte_e	A9.2	
1			1		
E1.7	s13brar		Vm_e	A9.3	
0			1		
E2.0	BT		Vhbp_s	A9.4	
1			1		
E2.1	S12tbh		Vhbp_e	A9.5	
1			1		
E2.2	S11ou		VhBrm_s	A9.6	
0			1		
E2.3	PRST		Vhbrm_e	A9.7	
0			1		
E2.4	FcBrp		Vpte_s	A10.0	
1			1		
E2.5	VBrmh		Vpte_e	A10.1	
1			1		

Mise en marche semi automatique par l'opérateur des actionneurs



**Réseau 3**: le cycle automatique du redresseur

Commentaire :



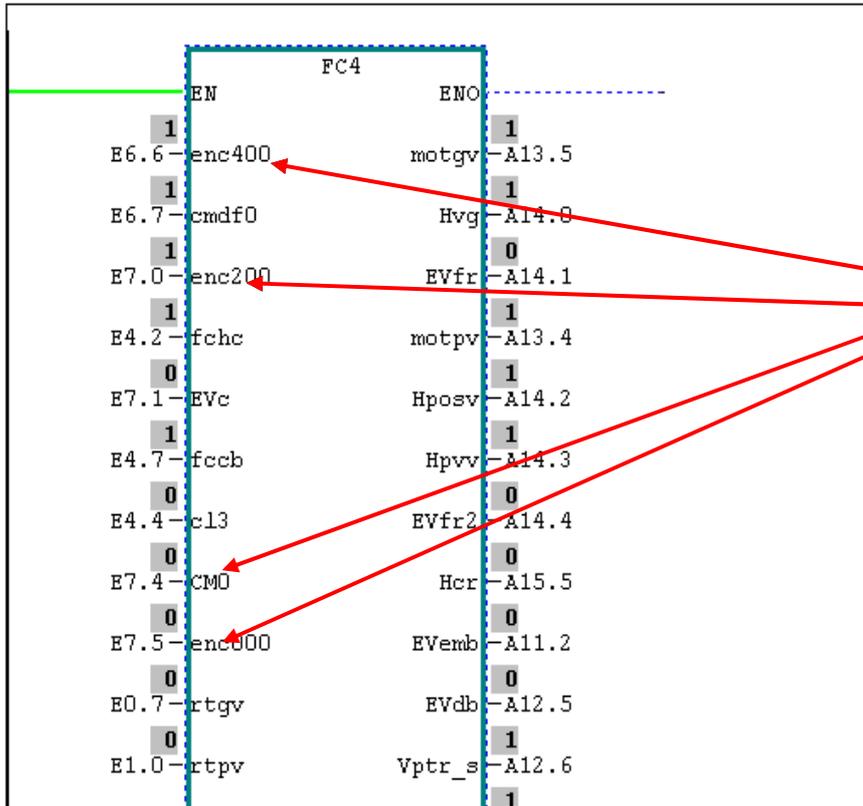
Activation et désactivation du moteur redresseur par les deux photos cellule

Voyant de l'enclenchement de la temporisation si la cellule 1 est défectueuse



**Réseau 4** : le cycle automatique de l'amenage et de la cisaille

Commentaire :



Les différents signaux délivrer par l'encodeur et exploiter par l'automate

Figure IV.7 : Visualisation du programme de la machine au niveau de l'OB1.

**Remarque :** Pour le programme de la cisaille voir annexe A.

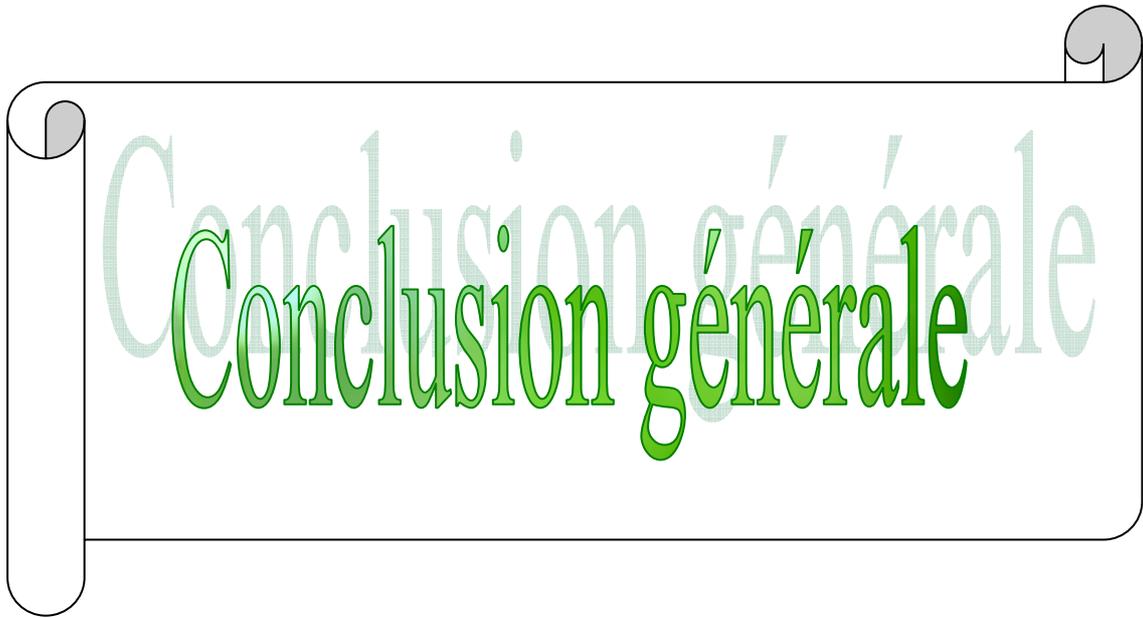
### 5. Conclusion

L'automatisation de la machine cisaille à guillotine avec l'API S7-300 nous procure plusieurs avantages tels que la flexibilité, la facilité d'extension de ses modules et la possibilité de visualisation et validation du programme établi avant son implantation sur l'automate grâce à son logiciel de simulation S7-PLCSIM.

Si nous ne disposerons pas d'un automate réel, le logiciel S7-PLCSIM devient l'outil indispensable pour la simulation des programmes et des concepts de commande automatisés avant leur implantation sur un système réels.

L'utilisation des bascules SR (S : condition d'activation, R : remise à zéro) nous permet d'activer et de désactiver chaque action et nous permettra de figer l'automate en cas de défaillance puis continuer l'exécution du programme après maintenance.

Le logiciel S7-PLCSIM nous a permet de valider et de visualiser le comportement des sorties et simuler le programme de la cisaille à guillotine.



Conclusion générale

Notre projet de fin d'étude a été réalisé en grande partie au sein du complexe électroménager de l'ENIEM dans le cadre d'un stage pratique de mise en situation professionnelle de trois mois.

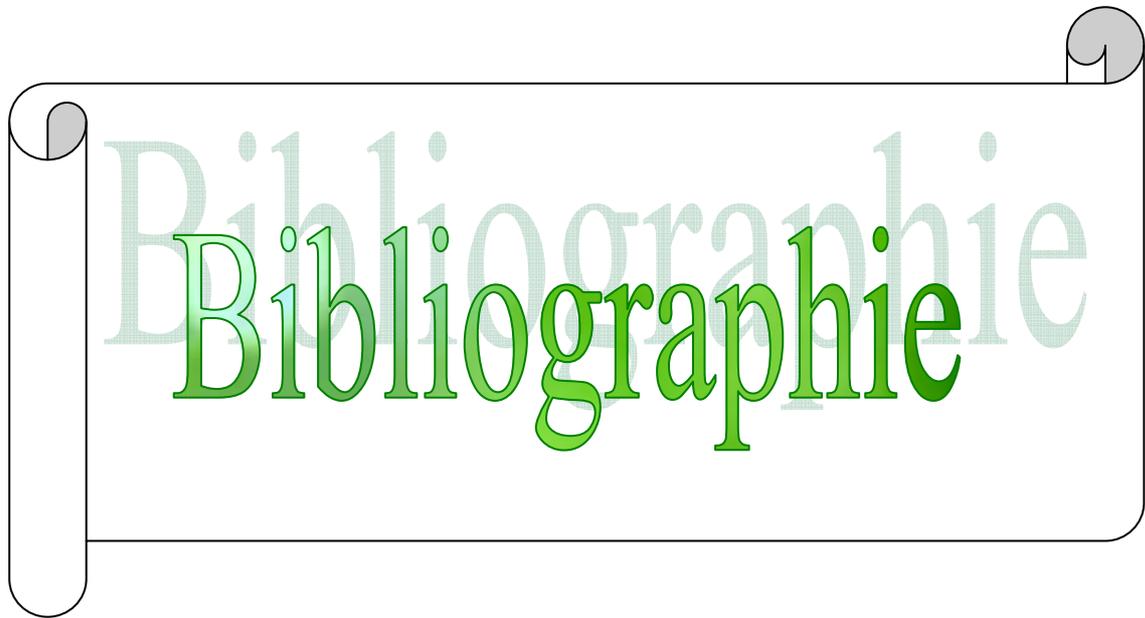
Dans ce mémoire, nous avons automatisé une cisaille à guillotine à l'unité cuisson de l'ENIEM. Nous avons procédé pour cela à l'étude de la conception matérielle et logicielle de la solution proposée.

Le GRAFCET est un outil de modélisation qui permet facilement le passage du langage de spécification au langage d'implantation. Il nous a permis de développer une solution fiable, simple à comprendre et à mettre en œuvre.

La commande des processus par des automates programmables industriels (API) est la solution recherchée de plus en plus dans l'industrie vu la justesse des traitements numériques qu'ils effectuent pour générer la commande adéquate à tout moment et dans toutes les conditions.

L'évolution des API ne cesse de continuer, notamment leurs logiciels de programmation qui présentent des interfaces très faciles à exploiter par l'utilisateur ce que nous avons constaté aussi lors de l'élaboration du programme pilote de notre système avec le STEP7.

En plus de l'étude que nous avons menée dans le cadre de notre projet, ce stage nous a permis de découvrir le monde industriel, d'enrichir nos connaissances dans le plan pratique et le domaine d'automatique (technologie) et les autres domaines et compléter ainsi notre formation théorique universitaire.



# Bibliographie

## ***BIBLIOGRAPHIE***

- ✓ [1] *Notices NORDA (documentation ENIEM)*
- ✓ [2] *API du Grafcet au Ladder v3.pdf* <http://iutcachanupsud.free.fr>
- ✓ [3] *Les dossiers pédagogiques de Festo téléchargeables à l'adresse suivante :*  
<http://www.festo.com/INetDomino/be/fr/73b8a0579e24b2e5c1256db7005425e9.htm>
- ✓ [4] <http://www.siemens.com/automation/service&support>
- ✓ [5] *Mémoire de fin étude KAHIL intitulé : étude de l'automatisation par API S7-300 de la machine de découpage et poinçonnage des contre portes frigorifique à L'ENIEM (Promotion 2007-2008).*



**OB1 - <offline>**

"Cycle Execution"

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 02/07/2009 01:10:02  
**Interface :** 15/02/1996 16:51:12  
**Longueur (bloc/code /données locales) :** 01204 01086 00026

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)"**

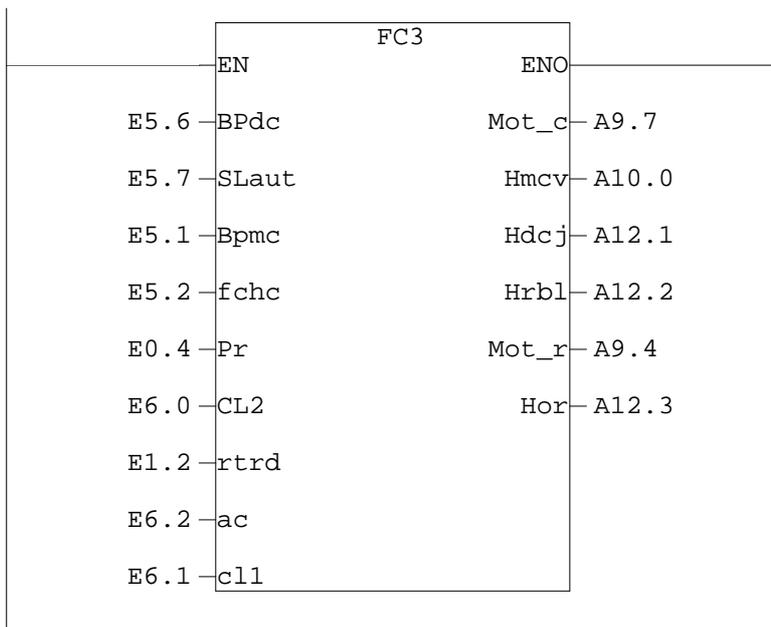
**Réseau : 1 POMPE HYDRAULIQUE**

	EN	FC1	ENO
E0.0	IG		motpm A8.0
E0.1	BApm		ST A8.1
E0.2	rtmh		HvST A8.2
E0.3	AU		STb A8.3
E0.4	Pr		HSTb A8.4
E0.5	PE		HrMA A8.5
E0.6	aur		
E0.7	rtgv		
E1.0	rtpv		
E1.1	rtc		
E1.2	rtrd		

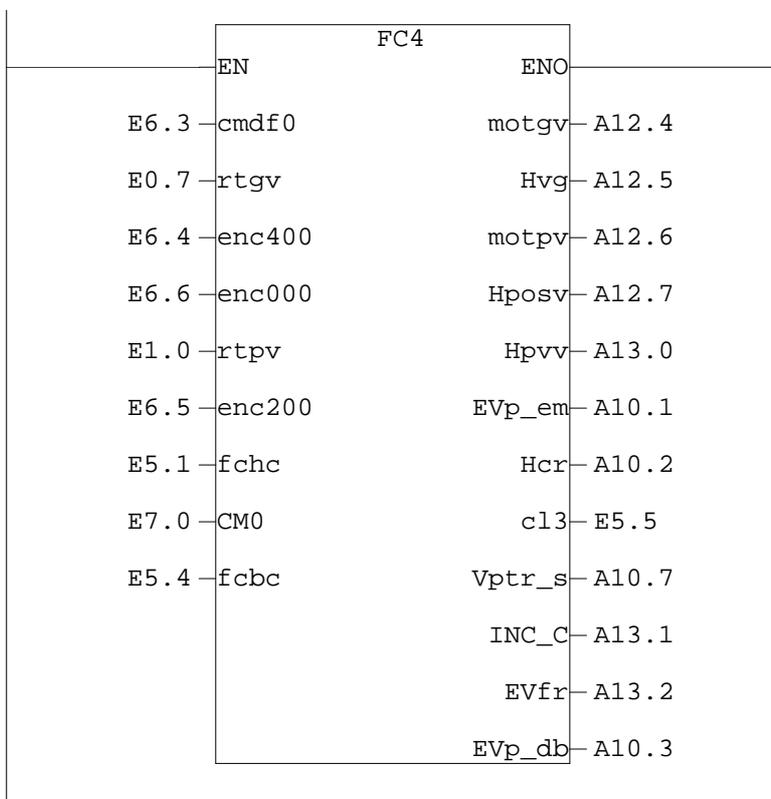
Réseau : 2      préparation de l'installation pour le cycle automatique

	EN	FC2	ENO	
E2.0	-S12tbh		Motbrav	A8.6
E0.5	-PE		Vhtb_s	A8.7
E2.1	-s12tbb		Vm_s	A9.0
E2.2	-S11ou		Vm_e	A9.1
E2.3	-PRST		vhtb_e	A9.2
E2.4	-s11f		Motbrar	A9.3
E2.5	-s13brar		mot_r	A9.4
E1.3	-s13brav		Hravb	A9.5
E1.4	-rtbr		Hrarb	A9.6
E0.3	-AU		Mot_c	A9.7
E1.5	-fcbr		Hmcy	A10.0
E1.6	-s13br0		EVp_em	A10.1
E2.6	-BT		Hcr	A10.2
E2.7	-S14Brpb		EVp_db	A10.3
E3.0	-FcBrp		Vpte_e	A10.4
E3.1	-LVBrmb		Vhbrm_e	A10.5
E3.2	-FcBrm		Vhbp_e	A10.6
E3.3	-Lvt_eh		Vptr_s	A10.7
E3.4	-LVt_e0		Vptr_e	A11.0
E0.4	-Pr		Vhbp_s	A11.1
E3.6	-LVr_eb		VhBrm_s	A11.2
E3.5	-LVr_eh		Vpte_s	A11.3
E3.7	-LVh_sint		Vpro_s	A11.4
E4.0	-Lvb_sint		Vl_s	A11.5
E4.1	-S15av		Mot_d	A11.7
E4.2	-s15ar		Mot_dar	A12.0
E4.3	-S150		Vl_e	A11.6
E4.4	-Lvt_sh			
E4.5	-Slrav			
E4.6	-S1m			
E1.2	-rtrd			
E4.7	-Slr0			
E5.0	-Slrar			
E5.1	-Bpmc			
E5.2	-fchc			
E1.1	-rtc			
E5.3	-Bpc			
E5.4	-fcbc			
E5.5	-C13			

Réseau : 3 le cycle automatique du redresseur



Réseau : 4



**FC1 - <offline>**

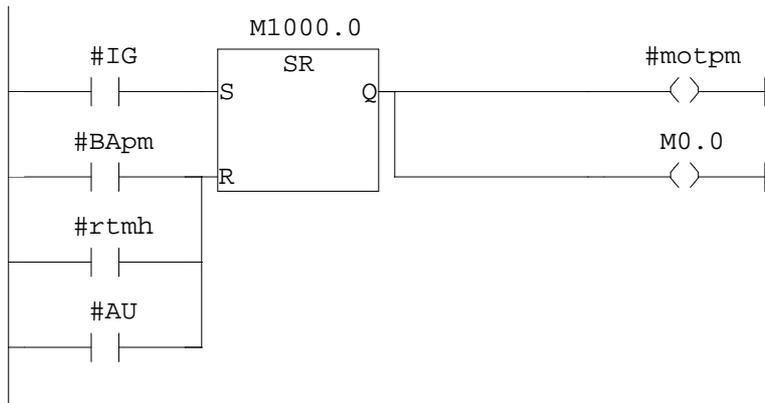
" "

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 01/07/2009 22:07:41  
**Interface :** 01/07/2009 22:07:41  
**Longueur (bloc/code /données locales) :** 00250 00118 00000

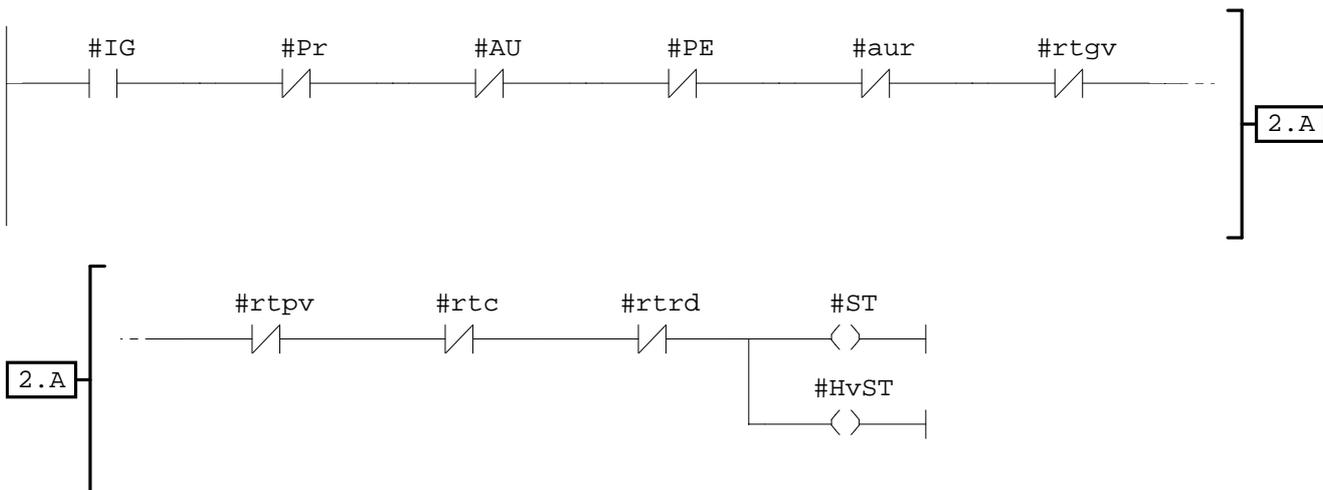
Nom	Type de données	Adresse	Commentaire
IN		0.0	
IG	Bool	0.0	
BApM	Bool	0.1	
rtmh	Bool	0.2	
AU	Bool	0.3	
Pr	Bool	0.4	
PE	Bool	0.5	
aur	Bool	0.6	
rtgv	Bool	0.7	
rtpv	Bool	1.0	
rtc	Bool	1.1	
rtrd	Bool	1.2	
OUT		0.0	
motpm	Bool	2.0	
ST	Bool	2.1	
HvST	Bool	2.2	
STb	Bool	2.3	
HSTb	Bool	2.4	
HrMA	Bool	2.5	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

**Bloc : FC1**

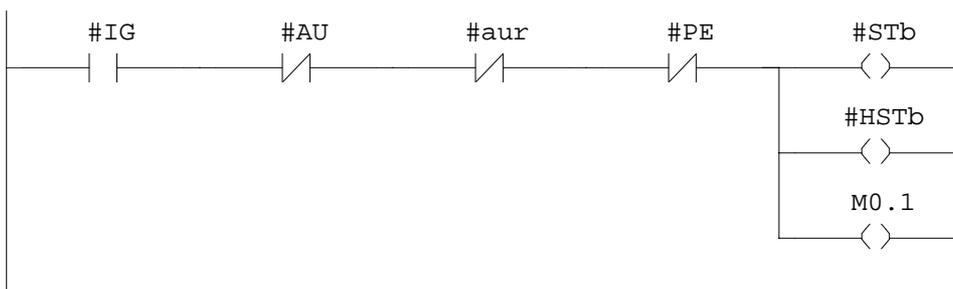
Réseau : 1      activation de la pompe



Réseau : 2      sous tension(securite du systeme)



Réseau : 3      allumage voyant sous tension blanc



Réseau : 4      manque d'air



**FC2 - <offline>**

" "

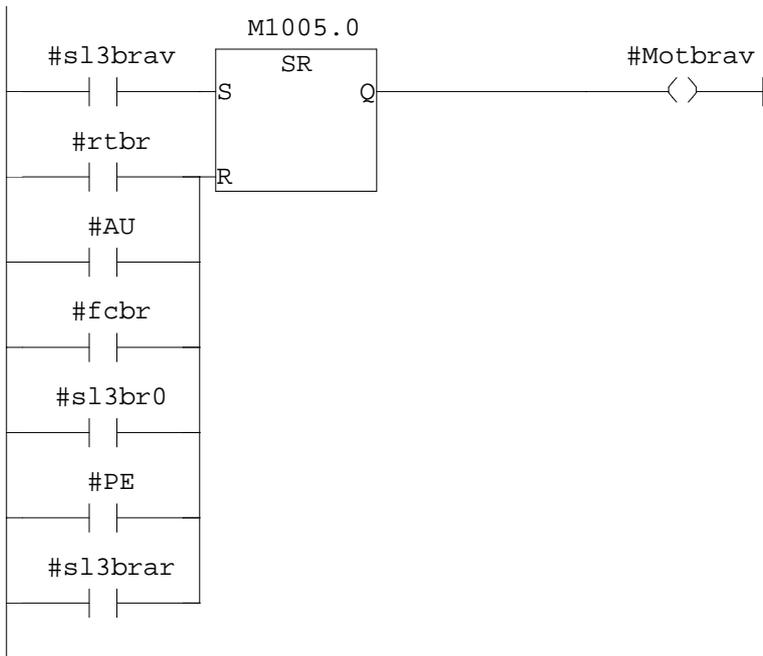
**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 01/07/2009 22:56:39  
**Interface :** 01/07/2009 22:56:39  
**Longueur (bloc/code /données locales) :** 01032 00766 00000

Nom	Type de données	Adresse	Commentaire
IN		0.0	
S12tbh	Bool	0.0	
PE	Bool	0.1	
s12tbb	Bool	0.2	
S11ou	Bool	0.3	
PRST	Bool	0.4	
s11f	Bool	0.5	
s13brar	Bool	0.6	
s13brav	Bool	0.7	
rtbr	Bool	1.0	
AU	Bool	1.1	
fcbr	Bool	1.2	
s13br0	Bool	1.3	
BT	Bool	1.4	
S14Brpb	Bool	1.5	
FcBrp	Bool	1.6	
LVBrmB	Bool	1.7	
FcBrm	Bool	2.0	
Lvt_eh	Bool	2.1	
LVt_e0	Bool	2.2	
Pr	Bool	2.3	
LVr_eb	Bool	2.4	
LVr_eh	Bool	2.5	
Lvh_sint	Bool	2.6	
Lvb_sint	Bool	2.7	
S15av	Bool	3.0	
s15ar	Bool	3.1	
S150	Bool	3.2	
Lvt_sh	Bool	3.3	
Slrav	Bool	3.4	
S1m	Bool	3.5	
rtrd	Bool	3.6	
S1r0	Bool	3.7	
Slrar	Bool	4.0	
Bpmc	Bool	4.1	
fchc	Bool	4.2	
rtc	Bool	4.3	
Bpc	Bool	4.4	

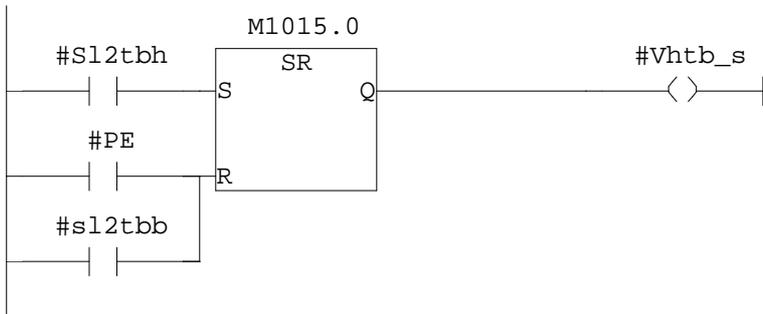
Nom	Type de données	Adresse	Commentaire
fcbc	Bool	4.5	
C13	Bool	4.6	
OUT		0.0	
Motbrav	Bool	6.0	
Vhtb_s	Bool	6.1	
Vm_s	Bool	6.2	
Vm_e	Bool	6.3	
vhtb_e	Bool	6.4	
Motbrar	Bool	6.5	
mot_r	Bool	6.6	
Hravb	Bool	6.7	
Hrarb	Bool	7.0	
Mot_c	Bool	7.1	
Hmcv	Bool	7.2	
EVp_em	Bool	7.3	
Hcr	Bool	7.4	
EVp_db	Bool	7.5	
Vpte_e	Bool	7.6	
Vhbrm_e	Bool	7.7	
Vhbp_e	Bool	8.0	
Vptr_s	Bool	8.1	
Vptr_e	Bool	8.2	
Vhbp_s	Bool	8.3	
VhBrm_s	Bool	8.4	
Vpte_s	Bool	8.5	
Vpro_s	Bool	8.6	
Vl_s	Bool	8.7	
Mot_d	Bool	9.0	
Mot_dar	Bool	9.1	
Vl_e	Bool	9.2	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

**Bloc : FC2**

Réseau : 1      avancement du berceau

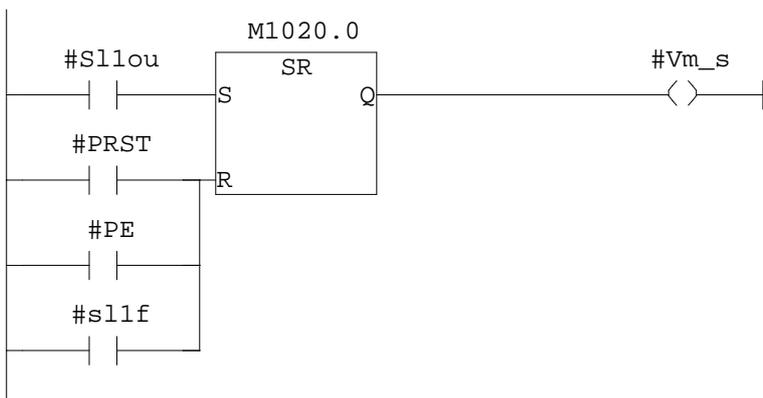


Réseau : 2      sortie du verin table berceau



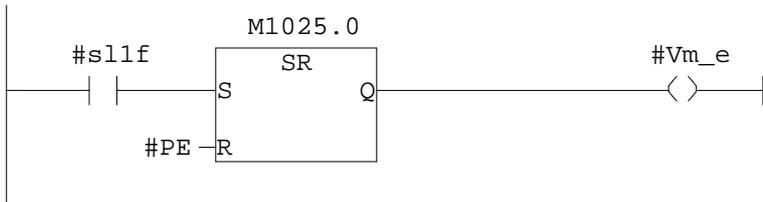
Réseau : 3      sortie du verin mandrin

après mise du coil sur le mandrin

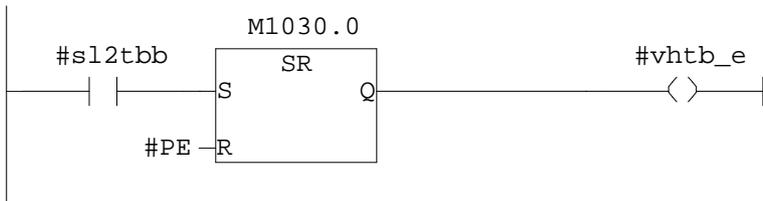


Réseau : 4      retour du verin mandrin

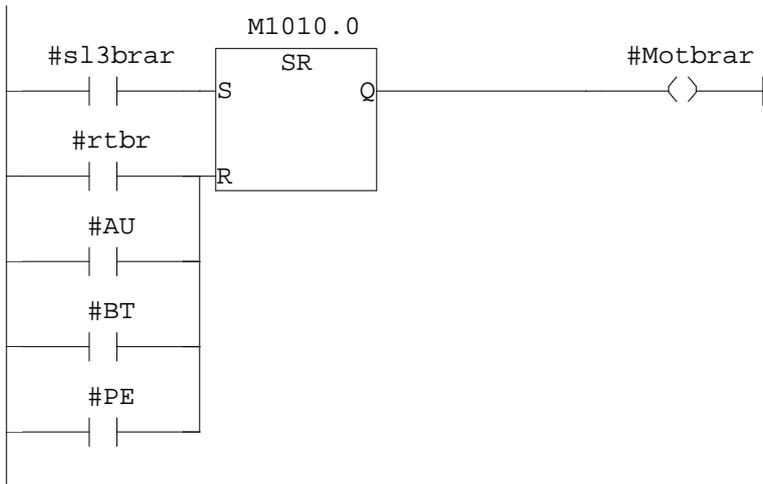
lors du changement du coil pour une nouvelle production avec des démenssion défferentes



Réseau : 5      retour du verin table berceau

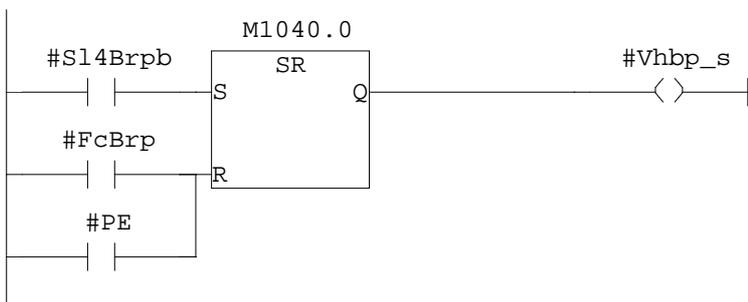


Réseau : 6      le retour du berceau

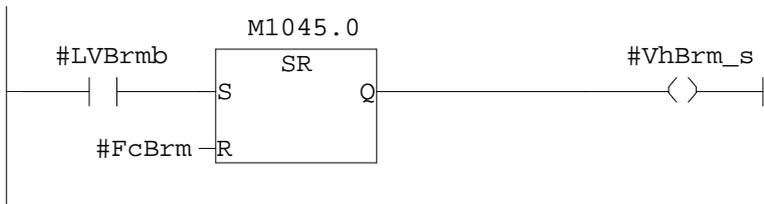


Réseau : 7      descente du bras presseur

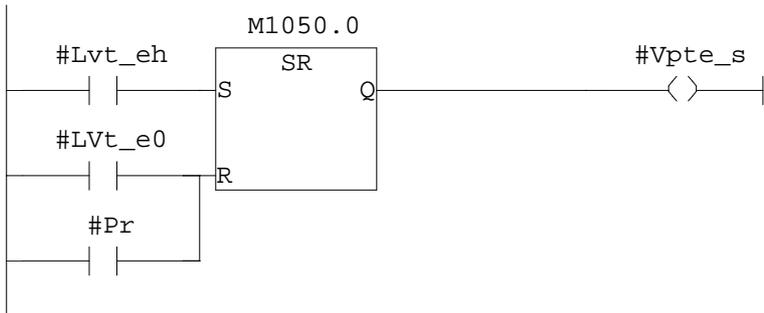
apres fixation du coil par des plaques



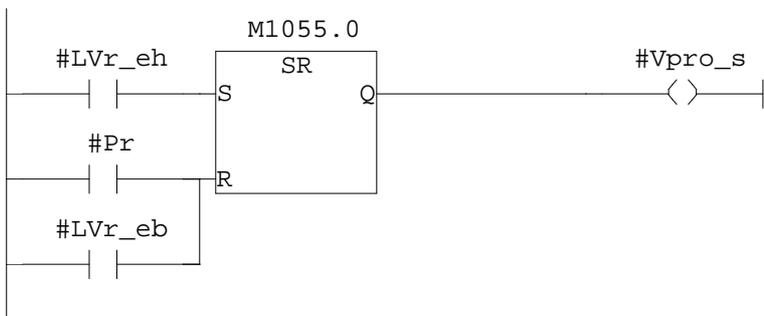
Réseau : 8      descente du bras maintien



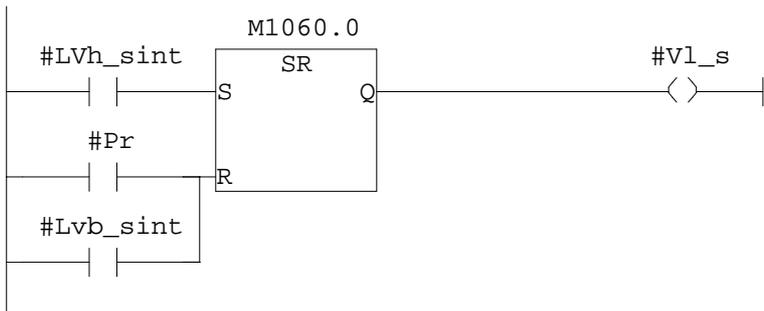
Réseau : 9      sortie du verin de la table d'entrée



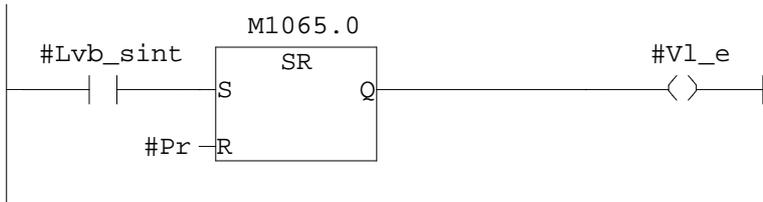
Réseau : 10      sortie du verin rouleau d'entrée



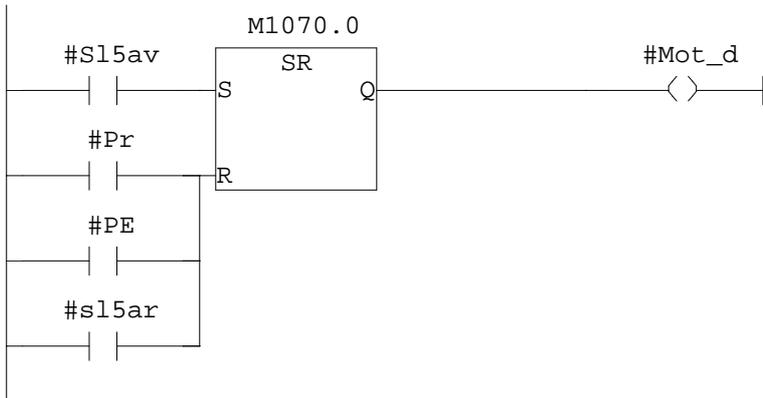
Réseau : 11      coupage de l'emballage



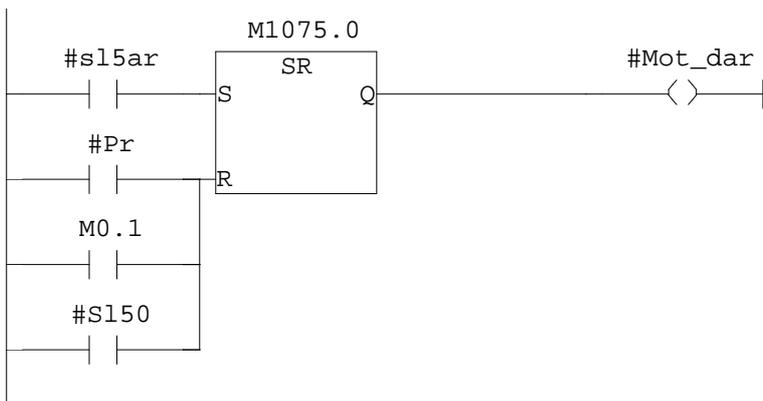
Réseau : 12      retour de lame d'introduction



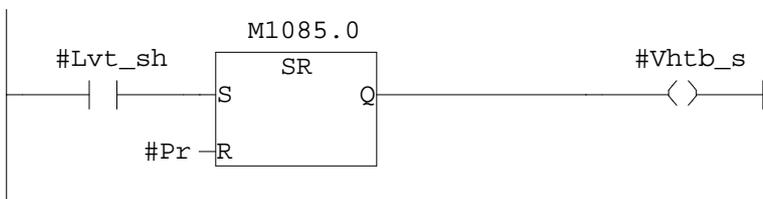
Réseau : 13      démarrage du moteur devoir en marche avant



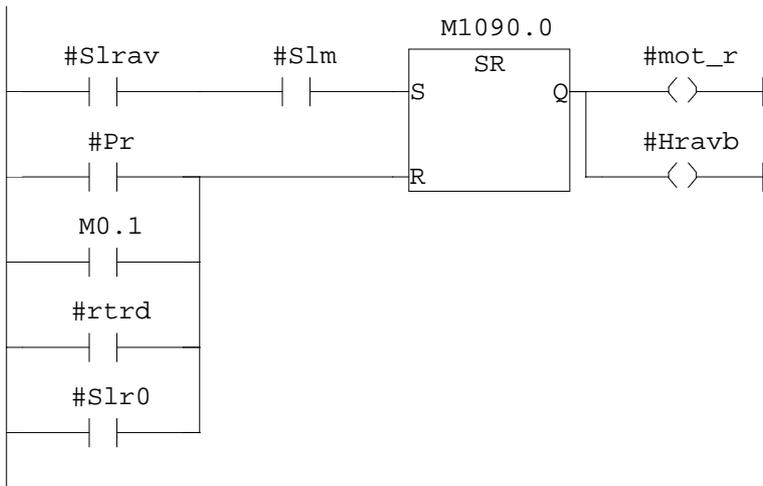
Réseau : 14      démarrage et arrêt du moteur devoir en marche arrière



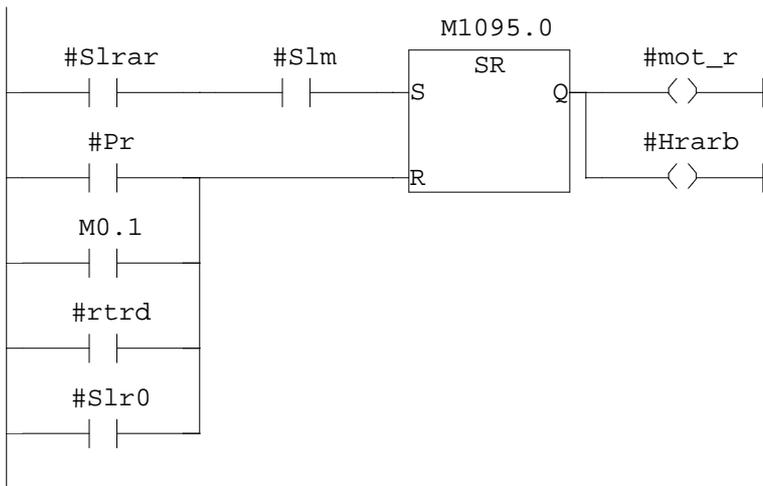
Réseau : 15      sortie de la table de sortie



Réseau : 16      demmarage du moteur redresseur en marche avant et arret

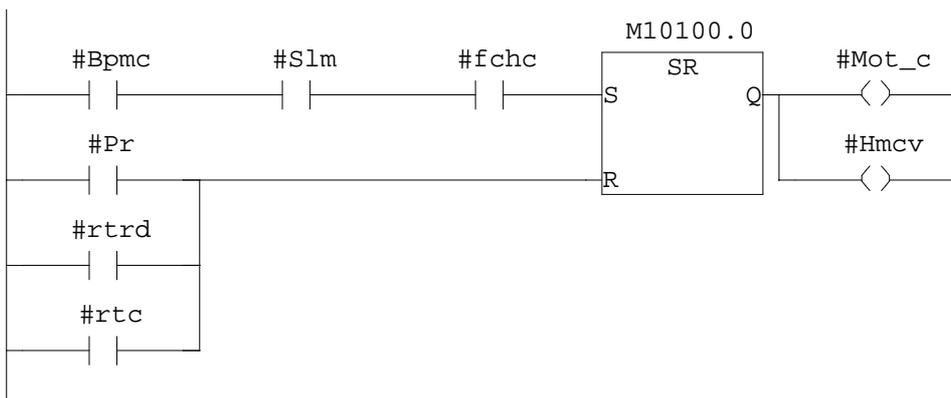


Réseau : 17      demmarage du moteur redresseur en marche arrière et arrêt

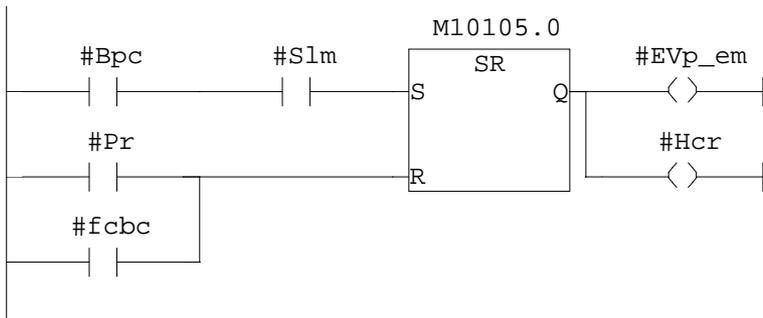


Réseau : 18      démarrage du moteur cisaille

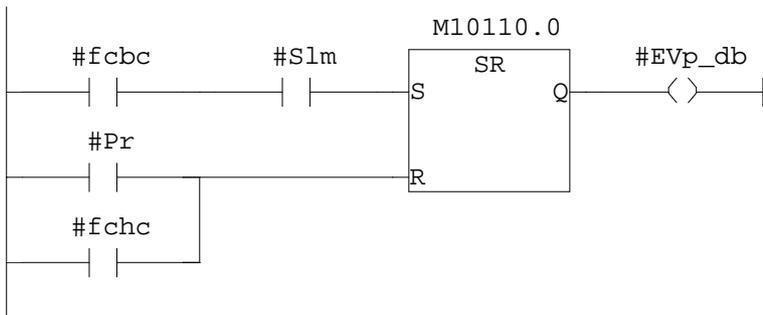
après centrage de la tôle



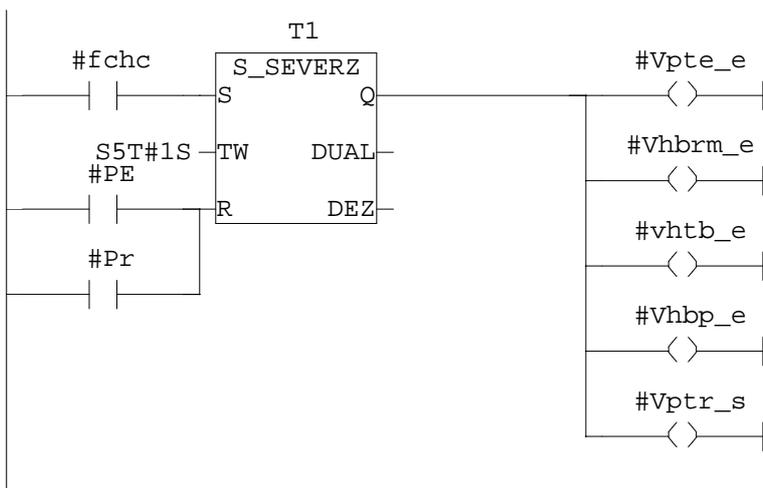
Réseau : 19      coupage de latôle



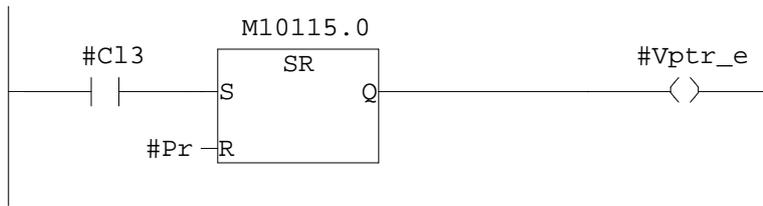
Réseau : 20      retour de la cisaille



Réseau : 21      temporisation et retour des bras et des tables et éjection



Réseau : 22      retour des trapes d'ejection



**FC3 - <offline>**

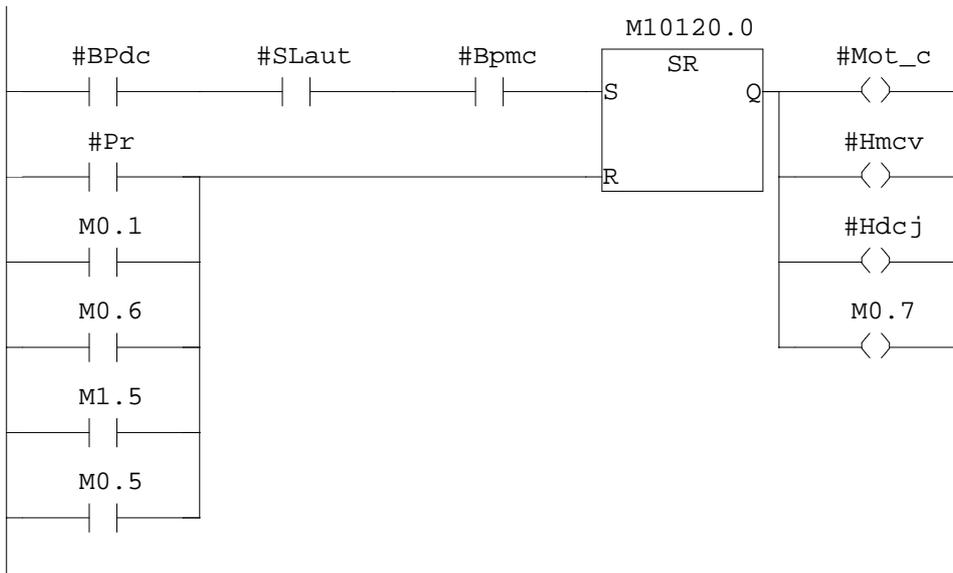
" "

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 02/07/2009 02:24:40  
**Interface :** 01/07/2009 23:38:07  
**Longueur (bloc/code /données locales) :** 00328 00194 00000

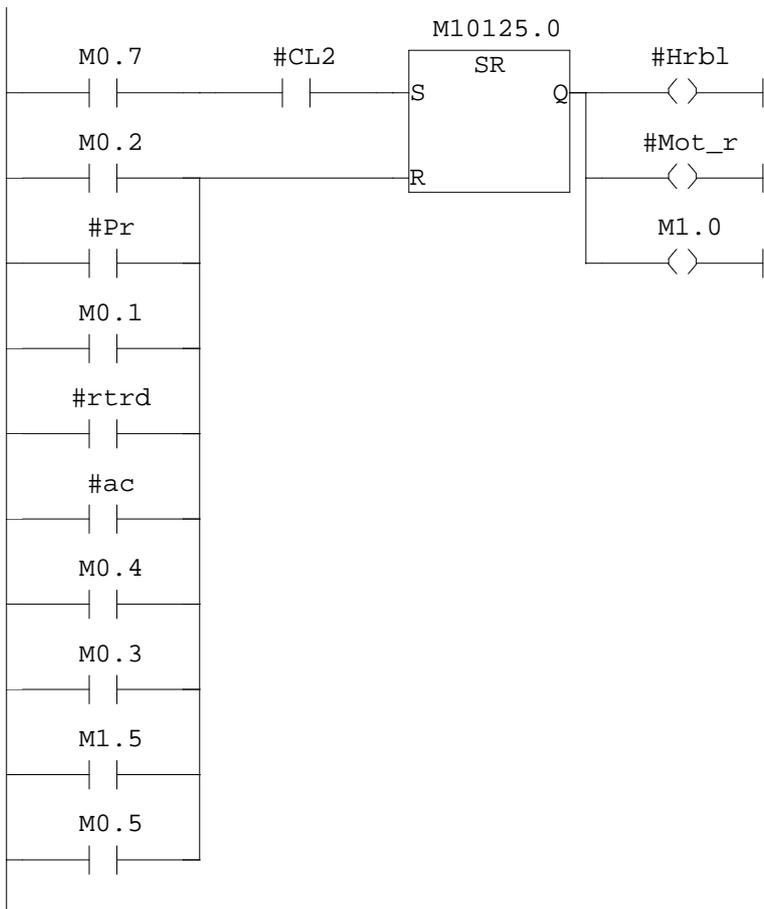
Nom	Type de données	Adresse	Commentaire
IN		0.0	
BPdc	Bool	0.0	
SLaut	Bool	0.1	
Bpmc	Bool	0.2	
fchc	Bool	0.3	
Pr	Bool	0.4	
CL2	Bool	0.5	
rtrd	Bool	0.6	
ac	Bool	0.7	
cl1	Bool	1.0	
OUT		0.0	
Mot_c	Bool	2.0	
Hmcv	Bool	2.1	
Hdcj	Bool	2.2	
Hrbl	Bool	2.3	
Mot_r	Bool	2.4	
Hor	Bool	2.5	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

**Bloc : FC3**

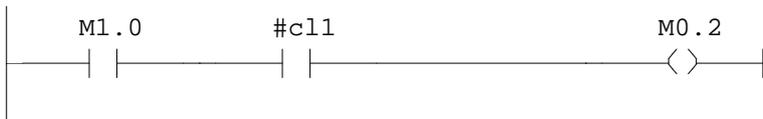
## Réseau : 1      excitation du moteur cisaille



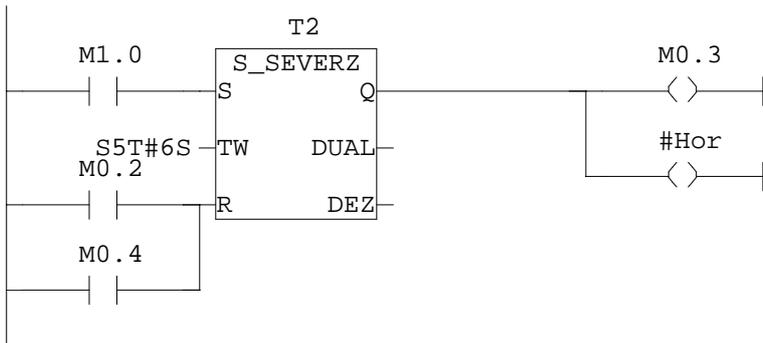
## Réseau : 2      enclenchement ET ARRÊT du moteur redresseur



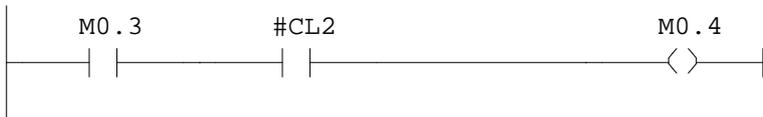
Réseau : 3      activation de la cellule 1 et arrêt du moteur redresseur



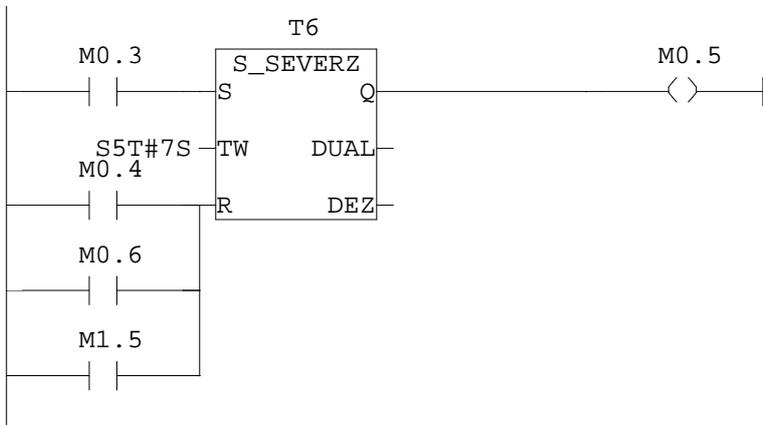
Réseau : 4      enclenchement de la temporisation lors du désfonctionnement de c



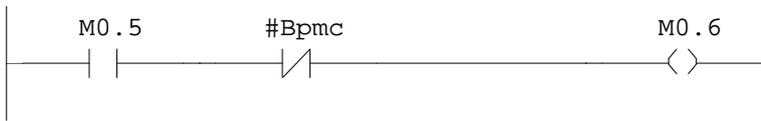
Réseau : 5      activation du redresseur provoqué par la cellule 2



Réseau : 6      temporisation lors désfonctionnement de c12



Réseau : 7      arrêt du système



**FC4 - <offline>**

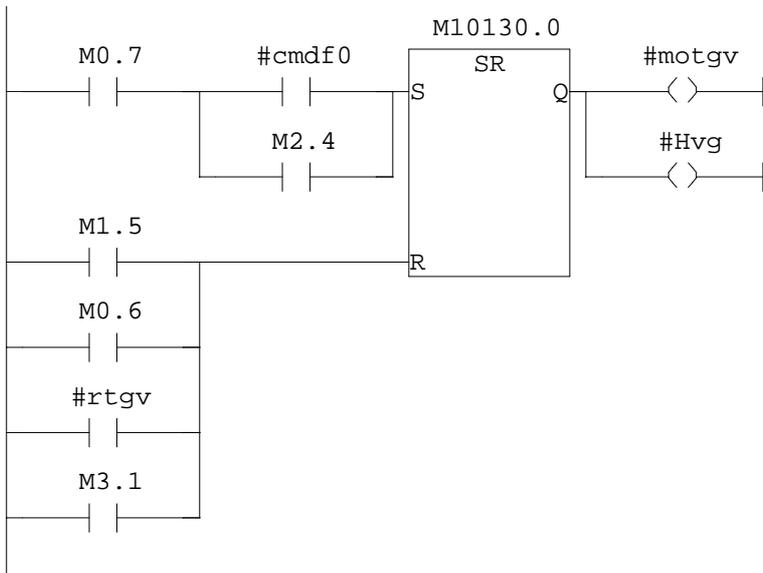
" "

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 02/07/2009 02:22:19  
**Interface :** 02/07/2009 01:04:06  
**Longueur (bloc/code /données locales) :** 00526 00368 00000

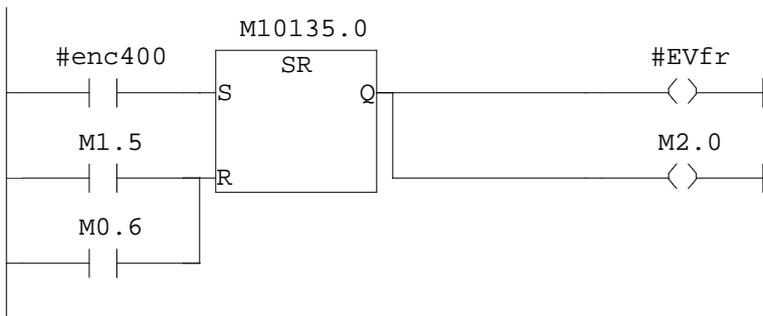
Nom	Type de données	Adresse	Commentaire
IN		0.0	
cmdf0	Bool	0.0	
rtgv	Bool	0.1	
enc400	Bool	0.2	
enc000	Bool	0.3	
rtpv	Bool	0.4	
enc200	Bool	0.5	
fchc	Bool	0.6	
CM0	Bool	0.7	
fcbc	Bool	1.0	
OUT		0.0	
motgv	Bool	2.0	
Hvg	Bool	2.1	
motpv	Bool	2.2	
Hposv	Bool	2.3	
Hpvv	Bool	2.4	
EVp_em	Bool	2.5	
Hcr	Bool	2.6	
cl3	Bool	2.7	
Vptr_s	Bool	3.0	
INC_C	Bool	3.1	
EVfr	Bool	3.2	
EVp_db	Bool	3.3	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

<b>Bloc : FC4</b>
-------------------

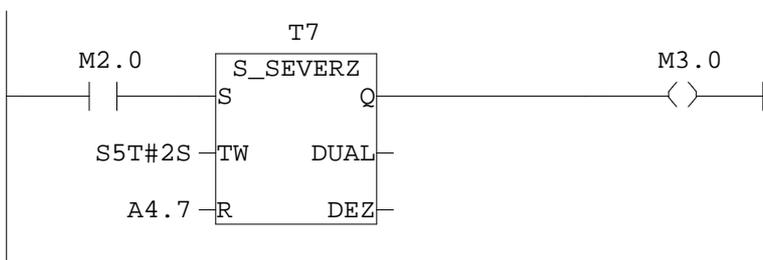
## Réseau : 1      enclenchement du motovariateur



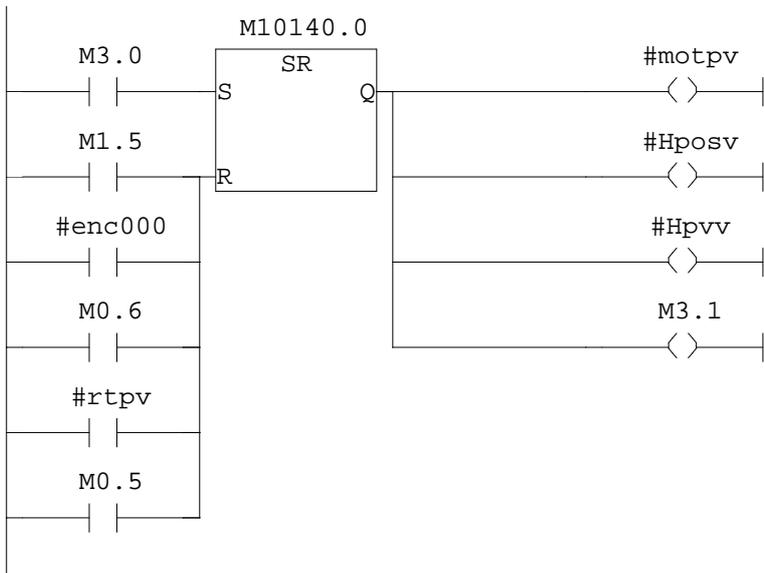
## Réseau : 2      enclenchement du frein



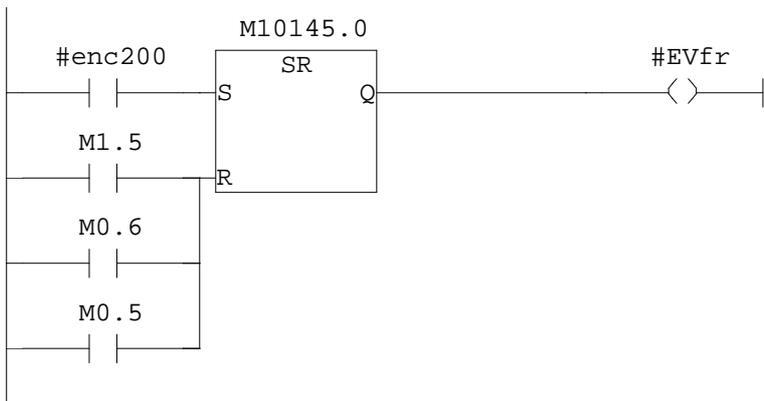
## Réseau : 3      temporisation du frein



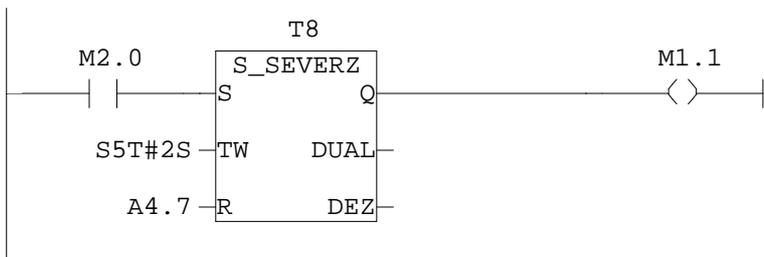
## Réseau : 4      enclenchement du motoréducteur



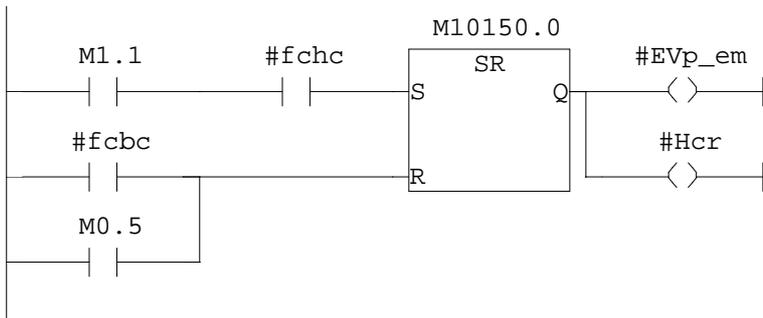
## Réseau : 5      enclenchement de l'encodeur et freinage



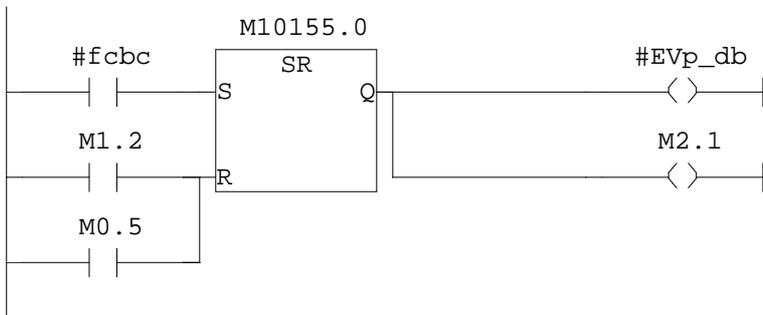
## Réseau : 6      enclenchement de la temporisation pour le 2eme frein



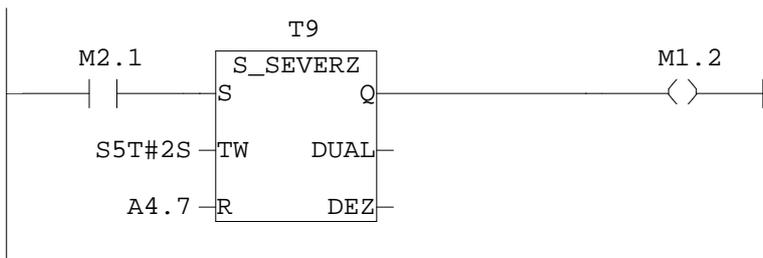
Réseau : 7      embrayage du moteur cisaille



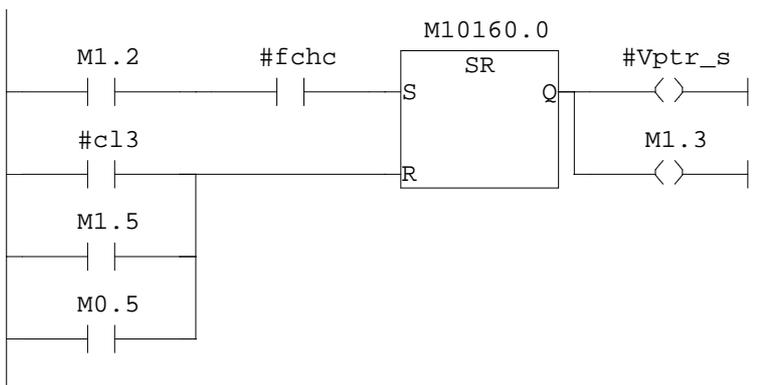
Réseau : 8      debreyage du moteur cisaille



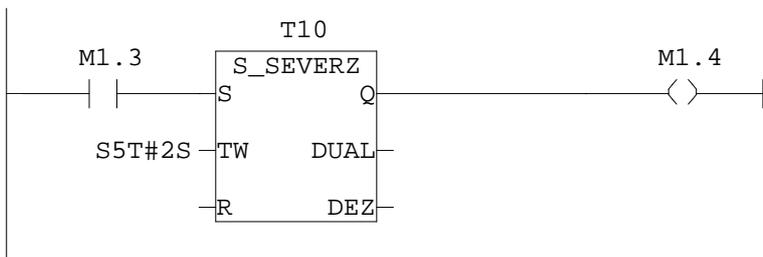
Réseau : 9      temporisation du debreyage



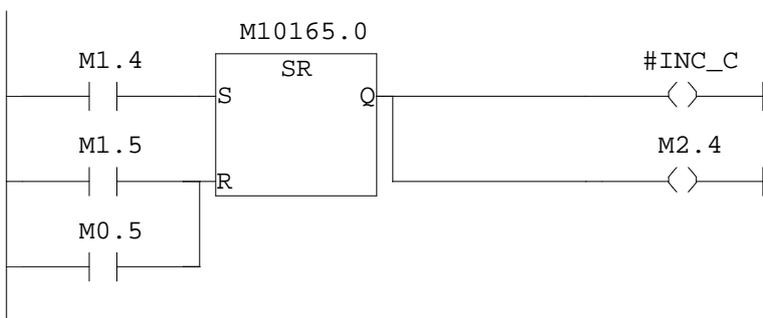
Réseau : 10      la sortie des deux verin des trapes



Réseau : 11



Réseau : 12      enclenchement du compteur soustracteur



Réseau : 13      verification de comparaison

