

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



Faculté de Génie Electrique et d'Informatique
Département d'Electronique

PROJET DE FIN D'ETUDES

En vue de l'obtention du diplôme

De master 2 en Electronique

Thème

Etude et automatisation des séquences d'entrées sorties de
Presse de transfert d'ENIEM par l'API S7-300

Proposé par :

Mr: B.BOUGDOUR

Présenté par :

M^{elle}: AKROUF Djamilia

M^{elle} : BELASSEL Meriem

M^{elle} : TAIBI Houria

Dirigé par :

Mr : ALLAD

Promotion 2013

La liste des figures

Figure 1.1 : Schéma synoptique de la machine	1
Figure 1.2 : Group dérouleur	3
Figure. 1.3 : chargeur (chariot mobile)	3
Figure. 1.4 : Bloc redresseur	4
Figure. 1.5 : Principe de fonctionnement d'un variateur de vitesse d'un MCC	6
Figure. 1.6 : la fosse.....	6
Figure 1.7 : Groupe aménage	7
Figure. 1.8 : Presse	8
Figure. 1.9 : Partie supérieur de la presse	8
Figure. 1.10 : moteur principale et volant d'inertie	9
Figure . 1.11 : Pupitre ligne.....	10
Figure . 1.12 : Le transfert des outilles.....	10
Figure. 1.13 :Table d'évacuation de pièces	11
Figure. 1.14 : pupitre de commande	11
Figure 1.15 : Système de proximité.....	13
Figure. 1.16 : Capteur fin de cours.....	13
Figure 1.17 : Vue en coupe d'un vérin pneumatique	14

Figure. 1.18 : Vérin simple effet	15
Figure .1.19 : un vérin double effet classique	15
Figure .1.20 : Vérins rotatifs classiques	16
Figure. 1.21 : Les distributeurs	17
Figure .1.22 : Schéma de puissance et de commande d'un moteur à un seul sens de rotation.....	19
Figure .1.23 démarrage direct d'un moteur triphasé à deux sens de rotations	20
Figure. 1.24 : le moteur à courant continu	21
Figure. 1.25 : Moteur hydraulique	21
Figure 2.1 : Représentation de deux étapes différentes.....	26
Figure II.2 : Etape inactive et étape active	27
Figure 2.3 : représentation d'une transition réceptive	27
Figure 2.4 : exemples de liaisons orientées	28
Figure 2.5 : conditions de franchissement	29
Figure 2.6 : activations ou désactivations simultanées d'une transition	30
Figure 2.7 : Action Conditionnelle « type C »	31
Figure 2.8 : Action Retardée	31
Figure 2.9 : Action Limitée	32
Figure 2.10 : Actions mémorisées	32
Figure 2.11 : Opérateurs Fronts montants et descendants	33
Figure 2.12 : Macro- étape	34
Figure 2.13 : Situation initiale	34

Figure 2.14: Transition non validée	35
Figure 2. 15 : Transition validée puis franchissable à t_1	35
Figure 2.16 : Franchissement d'une transition	36
Figure 2.17 : Franchissements simultanés	36
Figure 2.18: Activation et désactivation simultanées d'étape (étape 12)	36
Figure 2.19: Règle de franchissement d'une transition	37
Figure 2.20: représentation générale d'un GRAFCET	38
Figure 2.21 : Exemple d'un GRAFCET qui sera converti en partie commande	40
Figure II.22 : Nommage symbolique des entrées/sorties	40
Figure 3.1: structure d'un système automatisé industriel	68
Figure 3.2 : architecture interne d'un automate programmable	69
Figure 3.3 : L'interface d'entrée d'un API	70
Figure 3.4 :L'interface de sortie d'un API	70
Figure 3.5 : <i>Alimentation de l'automate</i>	71
Figure 3.6 : Alimentation des entrées	72
Figure 3.7: Alimentation des sorties	72
Figure 3.8 : Constitution externe de l'automate.....	76
Figure 3.9 :module d'alimentation	76
Figure 3.10:alimentation d'un API	77
Figure 3.11: Vue d'ensemble des éléments entrant dans l'automatisation d'un système	82
Figure 3.12 : structure du programme	84

Figure 3.13:fenêtre de création d'un objet	85
Figure 3.15: Sélection des blocs et le langage de programmation (CONT)	86
Figure 3.16 : Configuration matériels	87
Figure 3.17 : Vue des composants d'un projet S7.....	87
Figure 3.18 : Fenêtre du S7-PLCSIM	89
Figure 3.19 : Mise sous tension de la CPU	89
Figure 3.20 : résultat de la simulation du programme de la presse.....	90
Figure 4.1 : Constitution d'un système de supervision	91
Figure 4.2 : Vue d'ensemble du progiciel Win CC flexible.....	93
Figure 4.3 : la fenêtre de dialogue	95
Figure 4.4 : liaison entre la station S7-300 et la station de supervision HMI	96
Figure 4.5 : la vue d'accueil	97
Figure 4.6 : Vue pour redresseur	98
Figure 4.7 : vue aménage	99
Figure 4.8 : vue pour le fonctionnement de la cisaille	100
Figure 4.9 : Vue pour chargements de la presse	101

Sommaire

introduction générale 1

Chapitre 1 : Description de la machine « Presse de transfert »	3
1.1. Introduction	3
1.2. Description de la machine.....	4
1.2.1. Groupe dérouleur	5
1.2.2. Le groupe redresseur	6
1.2.3. Fosse	8
1.2.4. Groupe aménage	9
1.2.5. Groupe cisaille	9
1.2.6. Table d'aménage	10
1.2.7. Presse	10
1.2.8. Table d'évacuation de pièces	13
1.2.9. Pupitre de commande	13
1.2.10. Tapis d'évacuation de déchets	14
1.3. Capteurs	14
1.3.1. Principales caractéristiques des capteurs	14
1.3.2. Différents types de capteurs	14
1.3.3. Choix d'un capteur	15
1.4. Actionneurs	16
1.4.1. Vérins	16
1.4.2. Types de vérins	17
1.5. Pré-Actionneurs	19
1.6. Moteurs	20
1.7. Fonctionnement de la machine.....	25
1.8. Conclusion	27
Chapitre 2 : modélisation de la machine par GRAFCET.....	28
2.1. Introduction	28
2.2 Généralités sur GRAFCET	28
2.2.1. Historique.....	28
2.2.2Définition	28
2.2.3. Composition	29
2.2.4. Macro-étapes	36
2.2.5. Règles d'évolution du GRAFCET	37
2.2.6. Niveaux du GRAFCET	40
2.2.7. Mise en équation d'un GRAFCET	41
2.3. Notation des entrées/sorties.....	42

2.4.Conclusion	43
2.5. GRAFCET de la machine	44
Chapitre 3 :programmation de l'automate S7-300	67
3.1. Introduction.....	67
3.2. Automates Programmables Industriels(API).....	67
3.2.1. Définition d'un API.....	67
3.2.2. Structure d'un système automatisé industriel	67
3.2.3. Rôle d'un automate programmable dans les systèmes automatisés industriels.....	68
3.2.4. Architecture interne d'un automate programmable.....	69
3.2.5. Câblage d'un automate.....	70
3.2.6. Langages de programmation des API	72
3.2.7. Choix d'un automate programmable.....	73
3.3 .Automate programmable S7-300	74
3.3.1. Critères de choix de l'automate.....	74
3.3.2.présentation générale.....	74
3.3.3. Caractéristiques	74
3.3.5. Fonctionnement de l'automate programmable	79
3.3.6. Nature des informations traitées par l'automate	80
3.3.7. Ces avantages	80
3.3.8.passage de S5 à S7	81
3.3.9.programmation de l'automate S7-300.....	81
3.4. Simulation et la validation du programme	87
3.4.1. Mise en route du logiciel S7-PLCSIM	87
3.4.2. Simulation du programme de la presse transfert	90
3.4. Conclusion	90
Chapitre 4 : Plate forme de supervision sous WinCC Flexible 2008	91
4.1. Introduction.....	91
4.2. Définition de la supervision industrielle	92
4.3. Constitution d'un système de supervision	94
4.4 .Présentation du logiciel WinCC flexible.....	93
4.4.1. Définition de logiciel de supervision Win CC	93
4.4.2. Application disponibles sous Win CC	96
4.4.3. Création d'un projet sous Win CC	96
4.4.4. Intégration dans SIMATIC STEP 7	97
4.5. Plateforme de supervision de la station (ENIEM).....	98
4.6. Conclusion	103

Introduction Générale

Durant ces dernières années nous assistons à un développement énorme dans le monde industriel ; ce qui a provoqué le phénomène de concurrence au quel les entreprises sont confrontés dans des domaines très importants comme la mécanique, l'informatique et l'automatisation...les industries doivent garder leurs outils de production performant et fiable car cela reflète parfaitement leurs aptitudes à affronter le marché mondiale. D'où l'avènement des Automates Programmables Industriels qui jouent un rôle très important dans l'amélioration des procédés de productions et leur sécurités.

Le SIMATIC constitue une vaste plate forme d'automatisation offrant des solutions à des problèmes complexes pour tous les secteurs d'activités. Le logiciel STEP7 a été conçu pour offrir des fonctions conviviales de conduite et de simulation du processus, ce qui simplifie la mise en œuvre de nombreuses caractéristiques du système de commande, notamment la gestion de base de données commune.

Pour notre projet de fin d'étude, les responsables du département de maintenance de l'unité cuisson, nous ont confié la modernisation de la commande « presse de transfert » en remplaçant l'automate S5-115U existant par l'automate S7-300 de la firme SIEMENS ; cette amélioration va permettre à cette unité de :

- Réduire les frais de mains d'œuvre.
- Eviter les travaux dangereux et pénibles.
- Réaliser des opérations impossibles à contrôler manuellement.
- Commander à distance.
- Augmenter les performances du système de production.
- Améliorer la sécurité de l'installation industrielle et du personnel.

C'est dans ce contexte qu'a germé cette idée de contribuer à la conception de cette unité qui sera l'objet de notre projet de fin d'étude.

Pour ce faire nous avons décomposé notre travail en quatre principaux chapitres :

- Le 1^{er} chapitre comporte la description et le fonctionnement de la « presse transfert ».
- Le second chapitre sera consacré à la modélisation de la machine par l'outil graphique « GRAFCET ».
- Le 3^{ème} chapitre portera sur une description des automates programmables industriels ensuite la programmation à l'aide de l'automate S7-300 qui traitera la simulation du programme proposé par le logiciel S7-PLCSIM.
- Le 4^{ème} et dernier chapitre sera consacré à la supervision de tous les blocs qui constitue la machine à l'aide de WinCC flexible 2008.

Nous terminons notre travail une conclusion générale et perspective.

Chapitre 1

Description de la machine « Presse de transfert »

1.1. Introduction

La machine étudiée, presse transfert, est de type (630 2MR-TR3). Sert à l'emboutissage de tôle des cuisinières. Elle est complètement automatique et fonctionne à l'aide d'un séquenceur S5 de la firme SIEMENS. Elle est fabriquée sur commande de l'ENIEM à l'entreprise MANZONI en Italie. Elle occupe une place très importante dans l'unité cuisson. Elle utilise comme matière première des bandes de tôle à très faible taux de carbone de largeur 1 m et de 2mm d'épaisseur max, enroulée sous forme de bobines. A la sortie de la presse les pièces usinées sont finies et prêtes à être montées.

1.2. Description de la machine

Les parties essentielles de la presse de transfert sont représentées sur le schéma bloc de la figure. I.1.

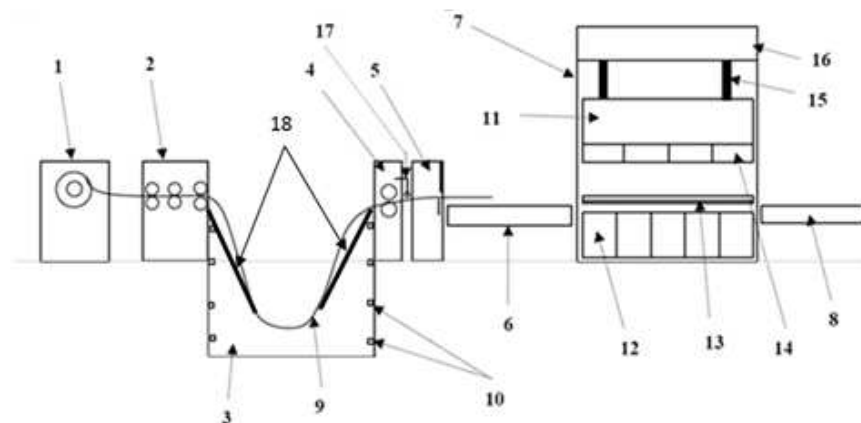


Figure 1.1 : Schéma synoptique de la machine

- 1- Dérouleur
- 2- Redresseur
- 3- Fosse
- 4- Système d'aménagement
- 5- Cisaille
- 6- Table d'aménagement
- 7- Presse
- 8- Tapis d'évacuation
- 9- Bobine
- 10- Capteurs photocellules de la fosse
- 11- Coulisseau
- 12- Partie inférieure des outils (les 4 tables portes outils)
- 13- Deux barres portes pinces
- 14- Partie supérieure des outils
- 15- Système bielle et l'excentrique
- 16- Ensemble moteur principal, volant d'inertie et réducteur
- 17- Encodeur
- 18-Dévidoir (deux plan basculant)

1.2.1. Groupe dérouleur

Le dérouleur est composé de sept distributeurs qui alimentent les vérins et les moteurs hydrauliques suivants :

- Moteur hydraulique qui assure la translation de la benne
- Deux vérins assurant le mouvement vertical de la benne
- Moteur hydraulique qui assure la rotation des mandrins
- Vérin double effet du blocage de la rotation des mandrins
- Vérins de chaque expansion (mâchoire)
- Vérin du rouleau presseur

Quatre capteurs de fin de course mécanique sont installés, deux pour la détection de la position des expansions et les deux autres pour détecter la position du vérin de blocage.

Le bloc dérouleur est représenté sur la figure I.2

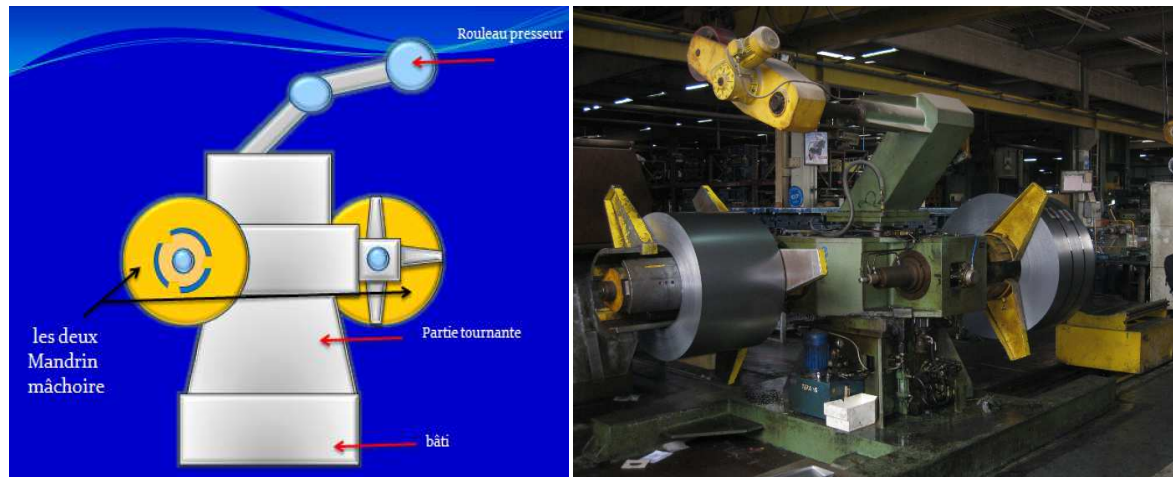


Figure 1.2 : Group dérouleur

Ce dispositif sert à dérouler la tôle pendant le cycle de travail, il est constitué essentiellement d'un chargeur, d'un bâti sur lequel est montée la partie tournante qui porte les deux mandrins mâchoires. La rotation de la partie tournante est assurée par un moteur hydraulique. Après une rotation de 180° , cette dernière sera bloquée par un vérin double effet.

1.2.1.1. Chargeurs (chariot mobile)

Il est en acier, il comporte une benne, sur laquelle l'opérateur dépose le rouleau de tôle. Elle est soutenue par un vérin hydraulique qui lui permet de se déplacer verticalement. Le chariot peut se déplacer longitudinalement sur des rails à l'aide d'un moteur hydraulique.

Un capteur de fin de course mécanique est installé au début des rails, il indique que le chariot est prêt à être chargé par les griffes.

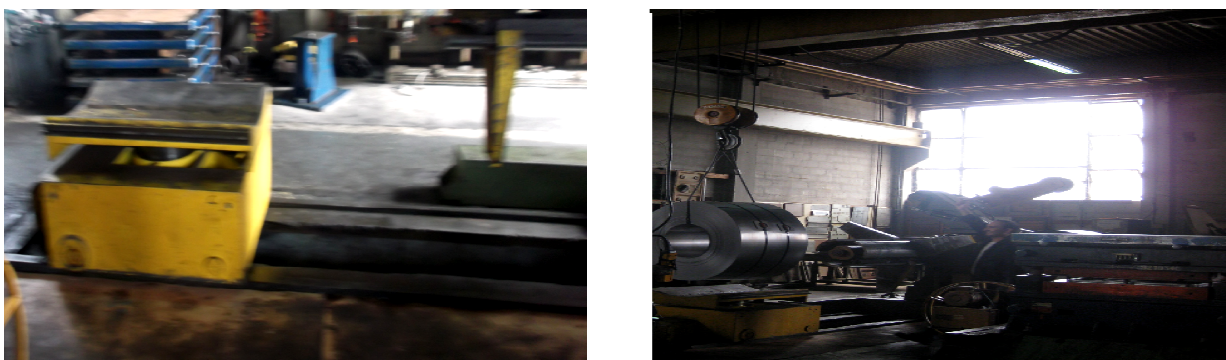


Figure. 1.3 : chargeur (chariot mobile)

1.2.1.2. Mandrins (mâchoires)

Ils sont formés de trois dents qui s'étirent à l'aide d'un vérin hydraulique pour entretenir la bobine. Le dérouleur contient deux mandrins mâchoires. Sur le deuxième est chargée une bobine en stand-by.

1.2.1.3. Rouleau presseur

Il est muni d'une roue entraînée par un moteur triphasé assurant la rotation de la bobine, il donne ainsi la possibilité de faire entrer la tôle dans le redresseur à la phase de préparation du cycle. Ensuite le rouleau presseur regagne sa position initiale à l'aide d'un VDE installé sur son bras.

1.2.1.4. Groupe frein

Il est de type pneumatique mono disque, refroidi, à bas moment d'inertie. Il sert à réduire la vitesse de roulement de la bobine pendant la phase de chargement.

1.2.1.5. Centrale hydraulique

Elle est composée d'un moteur électrique qui entraîne une pompe, d'un réservoir d'huile, un régulateur de pression et d'un manomètre qui mesure sa valeur

1.2.2. Le groupe redresseur

Il sert à redresser les petites déformations de la tôle ; constitué essentiellement d'un introducteur et d'un redresseur (figure.1.4).

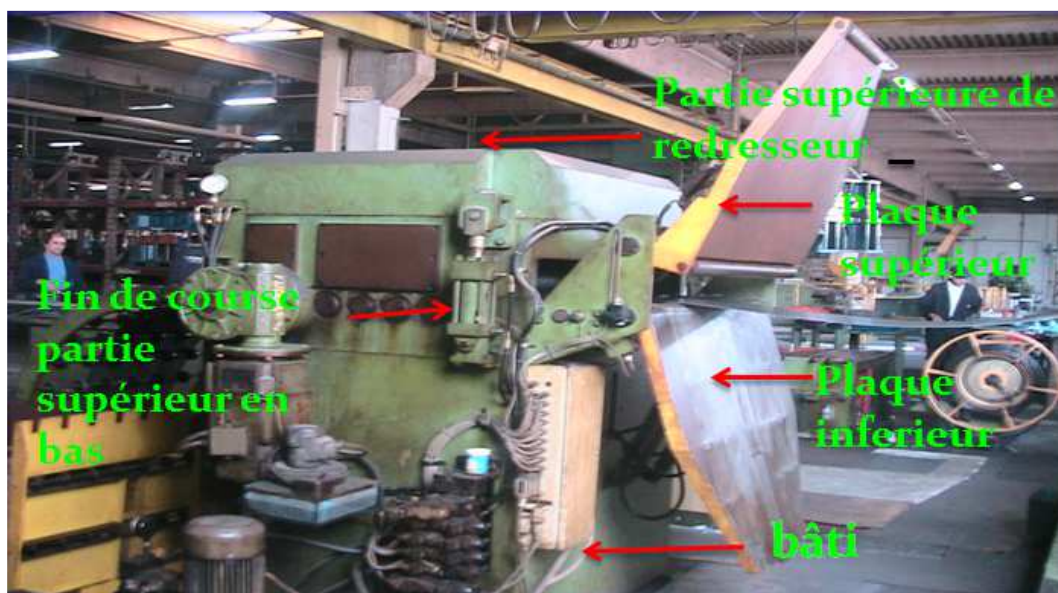


Figure. 1.4 : Bloc redresseur

1.2.2.1. Introducteur

Il est constitué d'une glissière qui possède deux plaques.

- **Plaque inférieure**

Elle est entraînée par un VDE, elle se positionne pour faciliter la réception de la tôle. Elle se termine par une lame qui se translate, par effet d'un VDE, pour dessaisir la tôle.

- **Plaque supérieure**

Elle contient deux roues, elle se positionne sur la plaque inférieure pour faciliter l'introduction de la tôle dans le redresseur. La glissière est munie aussi de deux capteurs de fin de course mécanique détectant la position des plaques.

1.2.2.2. Redresseur

Il est composé de deux rouleaux entraînés par un moteur triphasé, et sept rouleaux de redressage contre-roulé à leur tour. La commande des rouleaux tendeurs et redresseurs est obtenu par un groupe moteur à courant continu et un variateur de vitesse.

Un capteur photocellule est installé pour indiquer la présence de la tôle à l'entrée du redresseur.

1.2.2.3. Variateur de vitesse

Les variateurs de vitesse sont constitués principalement d'un convertisseur et d'une électronique de commande. En général, le convertisseur est un hacheur ou un onduleur.

Un variateur de vitesse pour moteur à courant continu basé sur le principe de la modulation de largeur d'impulsion MLI dite aussi PWM (pulse width modulation) et commandé par un microcontrôleur.

Dans notre machine le variateur de vitesse utilisé est de type EFG EF 1005, qui fonctionne avec une puissance de 7,5kw est une tension d'alimentation 400V.

- **Principe de fonctionnement d'un variateur de vitesse pour un MCC**

Pour faire varier la vitesse d'un moteur CC on peut faire varier la tension d'alimentation à ses bornes mais dans ce cas une partie importante de l'énergie est consommée par le dispositif d'alimentation, on préfère l'alimenter de façon discontinue avec un hacheur et faire ainsi varier la tension moyenne à ses bornes. On parle alors de Modulation par Largeur d'Impulsions (MLI).

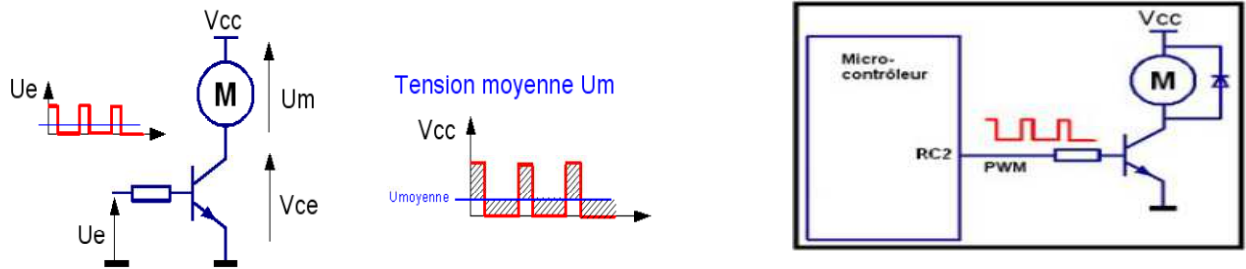


Figure. I.5 : Principe de fonctionnement d'un variateur de vitesse d'un MCC

1.2.3. Fosse

La machine est conçue pour un fonctionnement automatique fiable et pratique à grande vitesse de production, pour cela une fosse se trouve entre le redresseur et le groupe aménage, elle contient de la tôle sous forme d'un arc (figure. I.6).

Aux extrémités de cette dernière, il y a deux plans basculants, qui seront positionnées horizontalement à chaque début d'armement de la chaîne à l'aide de deux DVE.

Quand la tôle arrive à la table d'aménage les deux plans basculants reprennent leur position initiale, la tôle va se tendre vers le bas formant un arc. Cette technique permet d'éviter la déformation et le cisaillement de la tôle pendant le fonctionnement normal de la machine, en plus elle va assurer une disponibilité continue de la tôle pour alimenter la cisaille.

Pour obtenir la forme désirée de cette boucle des capteurs (photocellules), ont été installés, comme indiqué dans le schéma de la figure I.1.



Figure. I.6: la fosse

1.2.4. Groupe aménage

Il est situé en amont de la cisaille (figure-1-7), il comporte deux rouleaux superposés entraînés par un moteur à courant continu, ainsi qu'un dispositif de calcul de pas. Ce dispositif est constitué d'un encodeur, d'une roue d'appuis, et d'un capteur de fin de course mécanique pour indiquer la position de la roue. L'arrêt des rouleaux s'effectue par un système de freinage pneumatique.

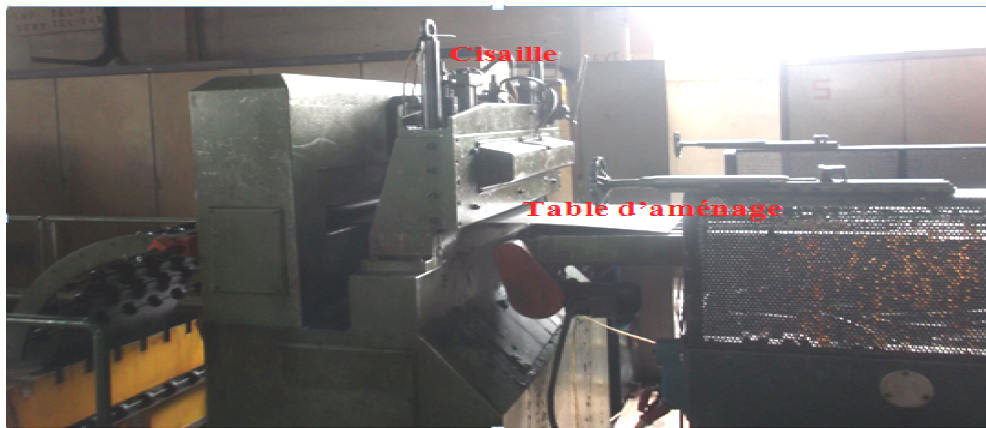


Figure 1.7: Groupe aménage

1.2.5. Groupe cisaille

Il contient deux vérins hydrauliques, qui sont alimentés par un même distributeur, trois amortisseurs pour éviter un retour brusque de la lame, et de deux capteurs de fins de course magnétiques, pour indiquer la position haute et la position basse de la lame. Le circuit hydraulique du groupe contient une pompe entraînée par un moteur électrique triphasé qui comprime l'huile dans le réservoir.

Les caractéristiques du groupe cisaille sont :

- Largeur bande maxi 1.000 mm
- Epaisseur bande maxi 2 mm
- Effort de cisaille maxi 7400 kg environ
- Pression d'exercice max 120 bar

1.2.6. Table d'aménage

Elle est constituée de six courroies placées longitudinalement, entraînées par un moteur triphasé à un seul sens de rotation, elle est située à la sortie de la cisaille pour transporter les

pièces brutes coupées. A son extrémité, un capteur de fin de course mécanique et deux capteurs photoélectriques sont installés pour détecter la présence de la pièce voir (figure-I-6).

1.2.7. Presse

C'est la partie la plus essentielle de la machine, elle contient une partie supérieure et une partie inférieure voire (figure.1.8)

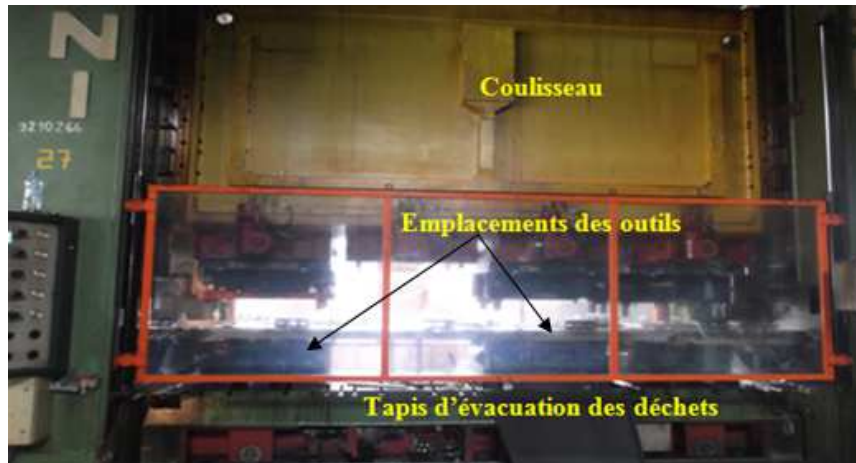


Figure.1.8 : Presse

1.2.7.1. Partie supérieure

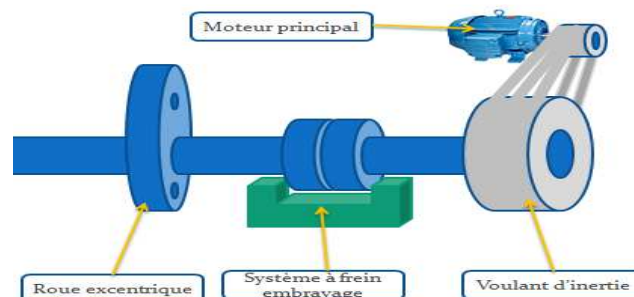


Figure. 1.9: Partie supérieur de la

1.2.7.1.1. Moteur principal

C'est un moteur à courant continu à puissance variable à couple constant alimenté par une petite génératrice, entraîne à l'aide d'une courroie et un volant d'inertie (figure. I.10). Il possède un système de contrôle qui permet de sélectionner la vitesse lente pour la mise en point des outils (2coups par minute) ou bien la vitesse de production, les réglages s'obtiennent de façon continue par potentiomètre. Son démarrage est toujours à la vitesse minimal en suite il atteint automatiquement la vitesse établie sur le potentiomètre.

1.2.7.1.2. Volant d'inertie

Il est conçu de manière à pouvoir développer la puissance nominale de la presse 15 coup par minute avec un degré de ralentissement de 0.15. Par l'intermédiaire d'un arbre et d'un réducteur il entraîne deux bielles excentriques qui sont liées à un coulisseau.

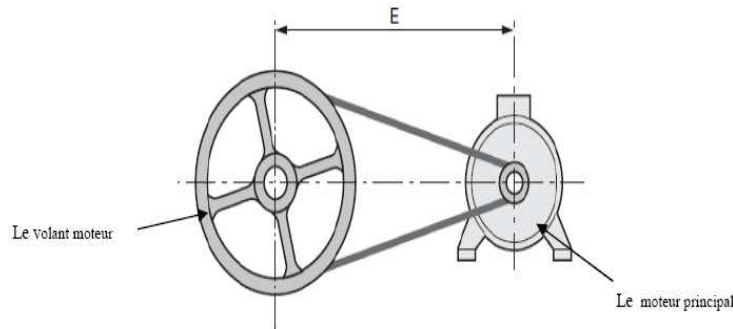


Figure. 1.10 : moteur principale et volant d'inertie.

1.2.7.1.3. Groupe excentrique

Il est composé d'un excentrique emboîté directement avec l'engrenage lent, et supporté par un pivot central fixe.

1.2.7.1.4. Groupe frein embrayage

C'est un disque refroidi, à bas moment d'inertie. Le dimensionnement est effectué selon les tours et l'énergie de la presse.

1.2.7.1.5. Coulisseau

Il est composé de quatre structures assemblées avec des anneaux de centrage pour garantir un alignement parfait

Le coulisseau comporte quatre emplacements d'outils et chaque emplacement est doté d'un moteur asynchrone triphasé à deux sens de rotation qui permet de régler individuellement les outils. La fixation des outils est effectuée par des vérins hydrauliques rotatifs.

La pression d'équilibrage est réglable en fonction de la vitesse par le pupitre ligne (figure.1.11) automatique connecté à un grand réservoir de compensation qui maintient la pression.

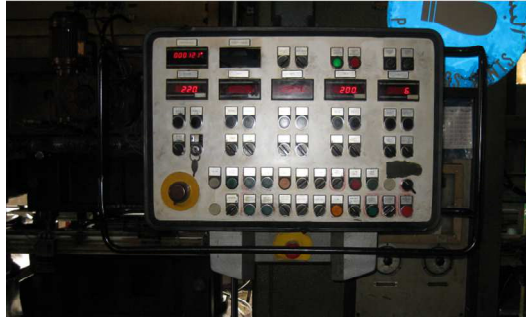


Figure .1.11 : Pupitre ligne

1.2.7.2. Transfert

Le transfert est complètement mécanique avec mouvement sur les trois axes .deux barres à porte pincettes sont munies d'un capteur électromagnétique indiquant la tenue de la pièce, fournissant ainsi un signal pour enclencher la sortie d'un petit vérin pneumatique servant à empoigner la pièce pendant son déplacement.

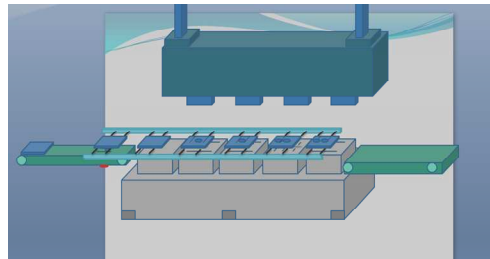


Figure .1.12: Le transfert des outils

1.2.7.3. La partie inférieure

Elle comporte une table en acier. Dans sa partie supérieure, il ya des rainures pour le fixage des outils et à l'intérieur de la table se trouve des logements pour les plaques, un poste de préparation et quatre postes d'usinages ces plaques sont fixées au bâti de la presse par six (6) vérins hydrauliques rotatifs.

Une inclinaison de 45°est prévue aux glissières d'évacuation de déchets pour faciliter l'introduction des déchets dans le dispositif d'évacuation.

1.2.8. Table d'évacuation de pièces

Elle est munie d'un moteur triphasé entrainant la rotation d'un tapis roulant, pour évacuer les pièces usinées. Sa commande est basée sur un compteur calculant le nombre de cycle complet des barres. Ce nombre est programmable sur le pupitre de commande, une fois atteint,

l'enclenchement du tapis est conditionné à un signal fourni par un capteur photoélectrique installé à la fin du tapis, qui indique l'évacuation des pièces par l'opérateur (figure. I.13)



Figure. I.13: Table d'évacuation des pièces

1.2.9. Pupitre de commande

Le pupitre est un des éléments permettant le dialogue entre l'opérateur et la partie commande. L'opérateur envoie des consignes à l'opérateur et reçoit des informations principalement visuelles. Il dispose pour cela des différentes formes de boutons, sélecteurs et voyants. (figure-I-14)



Figure.1.14 : pupitre de commande

1.2.10. Tapis d'évacuation de déchets

Il est composé de deux tapis, un frontal et l'autre à l'arrière de la presse. Ils sont positionnés au-dessous de niveau du sol et raccordés avec des glissières en tôle mises sur outil par soins de l'outilleur. Les tapis sont de longueur de la presse avec partie terminale qui remonte jusqu'au niveau d'un mètre du niveau du sol pour les récolter dans des caissons.

1.3. Capteurs

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

L'information délivrée par un capteur peut être logique (2 états), numérique (valeur discrète) ou analogique (amplitude, forme).

1.3.1. Principales caractéristiques des capteurs

- **l'étendue de la mesure**

C'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.

- **La sensibilité**

C'est la plus petite variation d'une grandeur physique que le capteur peut détecter.

- **La rapidité**

C'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information prise en compte par la partie commande.

- **La précision**

C'est l'aptitude d'un capteur à répéter une information sur une mesurande (position, vitesse,...etc.) Quand les mêmes conditions sont réunies.

1.3.2. Différents types de capteurs

- **Capteur photoélectrique**

Un capteur photoélectrique (photocellule) est un capteur de proximité. Il se compose d'un émetteur de lumière associé à un récepteur. La détection d'un objet se fait par la réflexion du faisceau lumineux qui produit un signal électrique. Ce signal est traité pour être exploité par la partie commande (voir figure **.1.15**)

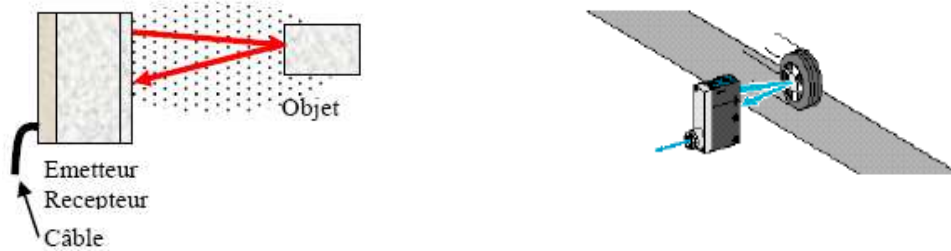


Figure 1.15 : Système de proximité

• Capteur de position

Les capteurs de position, nommés aussi « interrupteurs de fin de course » permettent d'informer sur la présence d'un objet par contact. Le contact est généralement maintenu à sa position de repos par un ressort (figure 1.16).

L'information donnée par ce capteur est de type tout ou rien et peut être électrique ou pneumatique. [4]

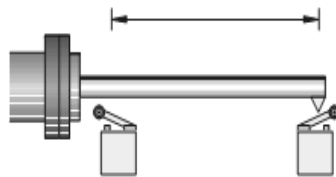


Figure.1.16: Capteur fin de cours

1.3.3. Choix d'un capteur

Le choix d'un capteur est conditionné par une multitude de facteurs dont on peut citer quelques-uns :

- L'environnement : température, humidité, poussière, projections diverses.
- La place disponible pour loger, fixer et régler l'appareil.
- La nature du circuit électrique.
- Le nombre et la nature des contacts.
- L'effort nécessaire pour actionner le contact.
- Les conditions d'exploitations, caractérisées par la fréquence de manœuvre, la nature, la masse et la vitesse du mobile à contrôler, la précision et la fidélité exigée.

- Le nombre de cycles de manœuvres.
- Le niveau de protection recherché entre les chocs, les projections des liquides.
- La source d'alimentation alternative ou continue.
- Le signal de sortie statique, électromécanique.
- Le type de raccordement : connecteur, câble.

1.4. Actionneurs

Les actionneurs sont des composants capables de produire un phénomène physique à partir de l'énergie qu'ils reçoivent, parmi ces actionneurs on trouve les vérins.

1.4.1. Vérins

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

Le vérin est constitué d'un piston muni d'une tige qui se déplace librement à l'intérieur d'un tube. Pour faire sortir la tige on applique une pression sur la face avant du piston, et sur la face arrière pour faire entrer la tige.

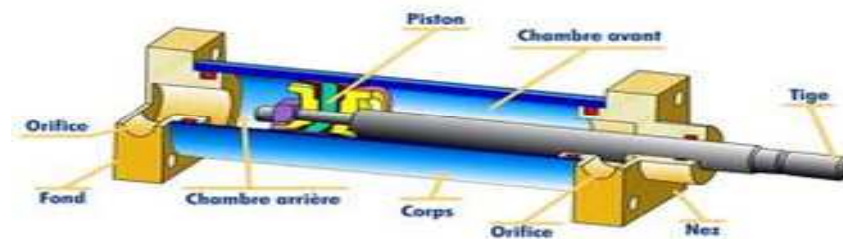


Figure1.17: Vue en coupe d'un vérin pneumatique

Selon la manière d'admission de l'air comprimé (ou l'huile), on distingue deux types de vérins : le vérin simple effet et le vérin à double effet.

Dans la presse transfert les vérins utilisés sont : les vérins pneumatiques simples effet, les vérins pneumatiques doubles effet, les vérins hydrauliques doubles effet et les vérins hydrauliques rotatifs.

1.4.2. Types de vérins

1.4.3.1. Vérins simple effet (VSE)

L'ensemble tige-piston se déplace dans un seul sens sous l'action du fluide sous pression. Le retour est effectué par un autre moyen que l'air comprimé : ressort, charge, Pendant le retour, l'orifice d'admission de l'air comprimé est mis à l'échappement (figure. **1.18**).

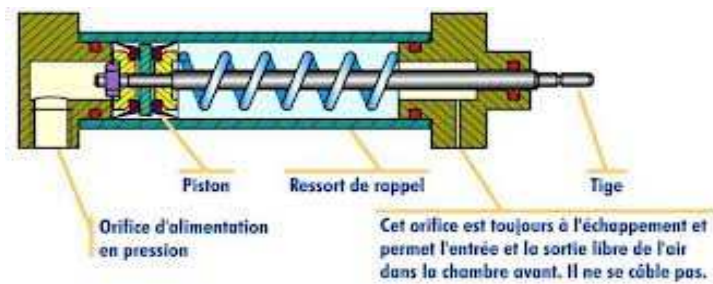


Figure. **1.18**: Vérin simple effet

Les vérins simple effet sont économiques, et leur consommation en fluide est réduite, ils sont plus longs que les vérins double effet, la vitesse de la tige est difficile à régler en pneumatique. En trouvent leur utilisation dans des travaux simples (serrage, éjection, ...).

1.4.3.2. Vérin double effet

Ce type vérin comporte deux orifices d'alimentation, développe une force disponible à l'aller comme au retour pour produire un travail. L'ensemble tige-piston peut se déplacer dans les deux sens sous l'action du fluide sous pression.

L'effort en poussant (sortie de la tige) est légèrement plus grand que l'effort en tirant (entrée de la tige) car la pression n'agit pas sur la partie de surface occupée par la tige.

tige-piston peut se déplacer dans les sous pression (figure. **1.19**).

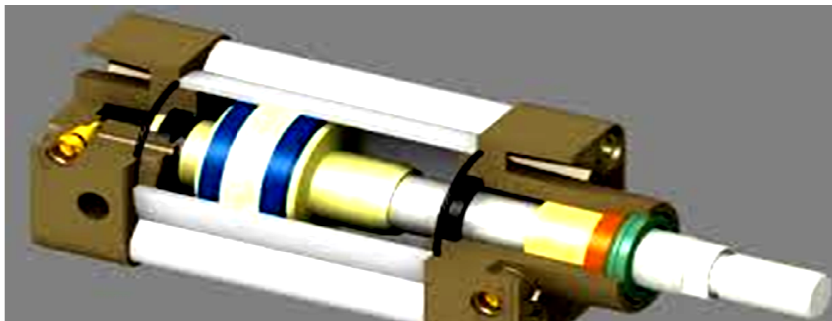


Figure **.1.19**: Vérin double effet classique

Les vérins double effet sont d'une par contrôle du débit à l'échappement, et à la présence des amortissements de fin de course, mais ils sont plus coûteux. Grande souplesse d'utilisation, grâce au réglage de la vitesse;

1.4.3.3. Vérins rotatifs

L'énergie du fluide est transformée en mouvement de rotation par exemple, vérin double effet entraînant un système pignon-crémaillère. L'angle de rotation peut varier entre 90 et 360°. (Figure. I.20).

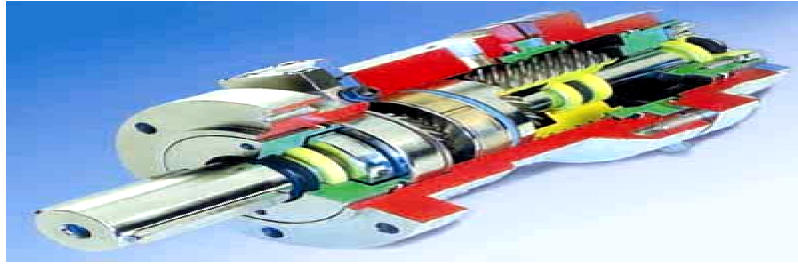


Figure .I.20: Vérins rotatifs classiques

La machine étudiée possède 16 vérins rotatifs, qui sont utilisé pour la fixation de la partie supérieur des outils et 4 autres pour la fixation la table sortante.

1.5. Pré-Actionneurs

Un pré-actionneur est un organe qui assure la distribution de l'énergie disponible aux actionneurs sur ordre de la partie commande. Son choix dépend de l'énergie distribuée. On distingue deux types de pré-actionneurs :

1.5.1. Distributeurs

Ils sont utilisés pour commuter et contrôler le débit du fluide sous pression, suite à la réception d'un signal de commande qui peut être mécanique, électrique ou pneumatique. Ils permettent de :

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

• Constitution d'un distributeur hydraulique

Le distributeur comporte un coulisseau, ou tiroir qui se déplace dans le corps du distributeur. Il permet de fermer ou d'ouvrir les orifices à travers les quels circule le fluide sous pression.

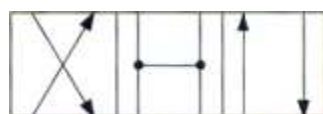
• Représentation schématique d'un distributeur

La représentation d'un distributeur s'effectue à l'aide de cases. Il ya autant de cases que de position possibles. A l'intérieur des cases, on représente les voies des passages de l'air ou l'huile pour chacune des positions.

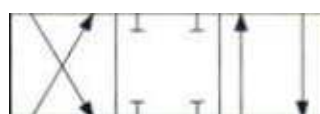
Pour caractériser un distributeur, il faut définir le nombre de voies ou d'orifices ainsi que le nombre de positions (exemple distributeur 3/2 : ce distributeur comprend 3 orifices et deux positions) (voir figure .I.21).

Le raccordement des tuyauteries se représentent sur la case symbolisant l'état de repos du distributeur. Pour en comprendre le fonctionnement, il faut imaginer que les canalisations son fixes et que ce sont les cases qui se déplacent devant les canalisations, et non l'inverse.

La presse transfert est munie de distributeurs de type : 4/3, 4/2, 5/2.



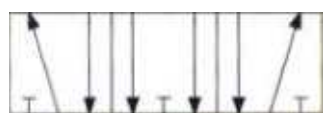
Distributeur 4/3 à centre partiellement ouvert.



Distributeur 4/3 à centre ferme



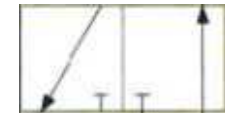
Distributeur 4/3 à centre ouvert en H



Distributeur 5/3 à centre ouvert



Distributeur 4/2 N.O



Distributeur 3/2 N.O

Figure 1.21 : Les distributeurs

1.5.2. Contacteur

Un contacteur est un relais de haute puissance. Il est utilisé pour commuter de moyennes ou grosses charges électriques. Il se compose d'une bobine qui est l'organe de commande

1.6. Moteurs

1.6.1. Moteurs électriques

Un moteur électrique est une machine servant à transformé l'énergie électrique en énergie mécanique rotationnelle. Il est basé sur le principe de l'action d'un champ tournant sur un enroulement en court-circuit.

Les moteurs sont des actionneurs électriques forts utilisés en milieu industriel, ils varient selon la tâche à accomplir.

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

La machine étudiée est équipée de (23) moteurs asynchrone à un seul sens de rotation, de (09) moteurs asynchrone à deux sens de rotation et de (02) moteurs à courant continu dont un est muni d'un variateur de vitesse.

1.6.2. Moteur Asynchrone triphasé

Le moteur asynchrone triphasé est largement utilisé dans l'industrie, sa simplicité de construction en fait un matériel très fiable qui demande peu d'entretien .Il est constitué d'une partie fixe, le stator et d'une partie rotative qui est le rotor.

Le moteur est relié au réseau par deux dispositifs de sécurité: relai thermique, sectionneur. Son circuit de puissance est alimenté en triphasé, par contre son circuit de commande est alimenté par l'intermédiaire d'un transformateur de sécurité.

• Relais thermique

Le relai thermique permet de protéger un récepteur contre les surcharges. Chaque relai comprend trois bilames constituées chacune de deux métaux dont le coefficient de dilatation est différent .Lors d'un incident, le courant absorbé par le récepteur augmente, les bilames se déforment et actionnent un mécanisme qui provoque la mise hors tension du récepteur.

• Sectionneur

Le sectionneur est un appareil électromécanique permettant de séparer de façon mécanique, un circuit électrique et son alimentation. Son objectif est d'assurer la sécurité des personnes travaillant sur la partie isolée du réseau électrique.

On alimente le circuit de puissance du moteur (figure **I.22**) par un sectionneur Q. On appuyant sur le bouton poussoir BP la bobine s'alimente .Elle ferme les contacts Km (le maintien) permettant l'alimentation du moteur dans un seul sens de rotation.

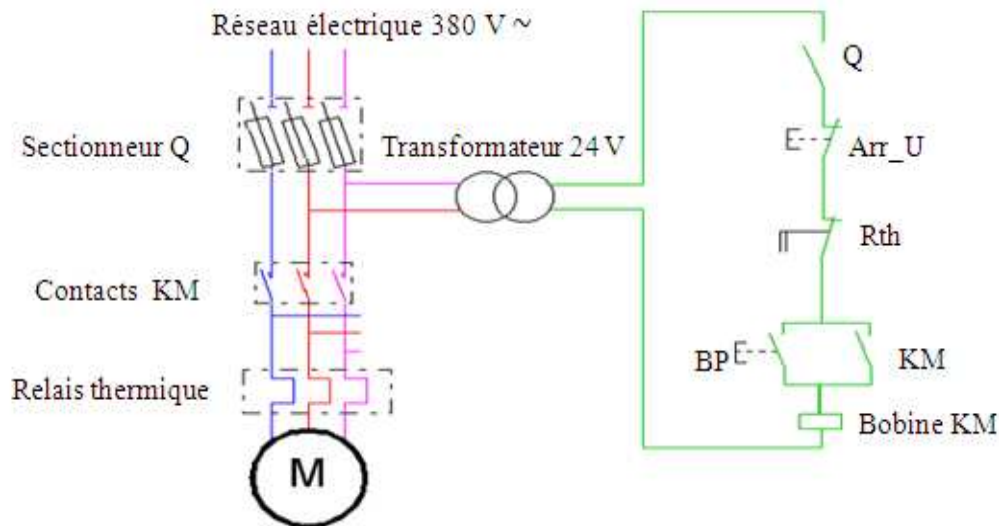


Figure .1.22: Schéma de puissance et de commande d'un moteur à un seul sens de rotation

La permutation de deux lignes permet d'obtenir un deuxième sens de rotation pour le moteur (un verrouillage pour la protection du moteur est obligatoire). Le moteur est contrôlé par le frein et le variateur de vitesse.

- **Le contacteur (KM)**

Permet d'alimenter le moteur avec une commande manuelle ou automatique avec un automate programmable.

L'action sur B1 excite la bobine du contacteur KM1 qui s'autoalimente, Après l'arrêt du premier sens l'action sur B2 excite la bobine de KM2 donc le deuxième sens de rotation, un verrouillage entre les deux sens de rotation assure la sécurité (figure I.22).

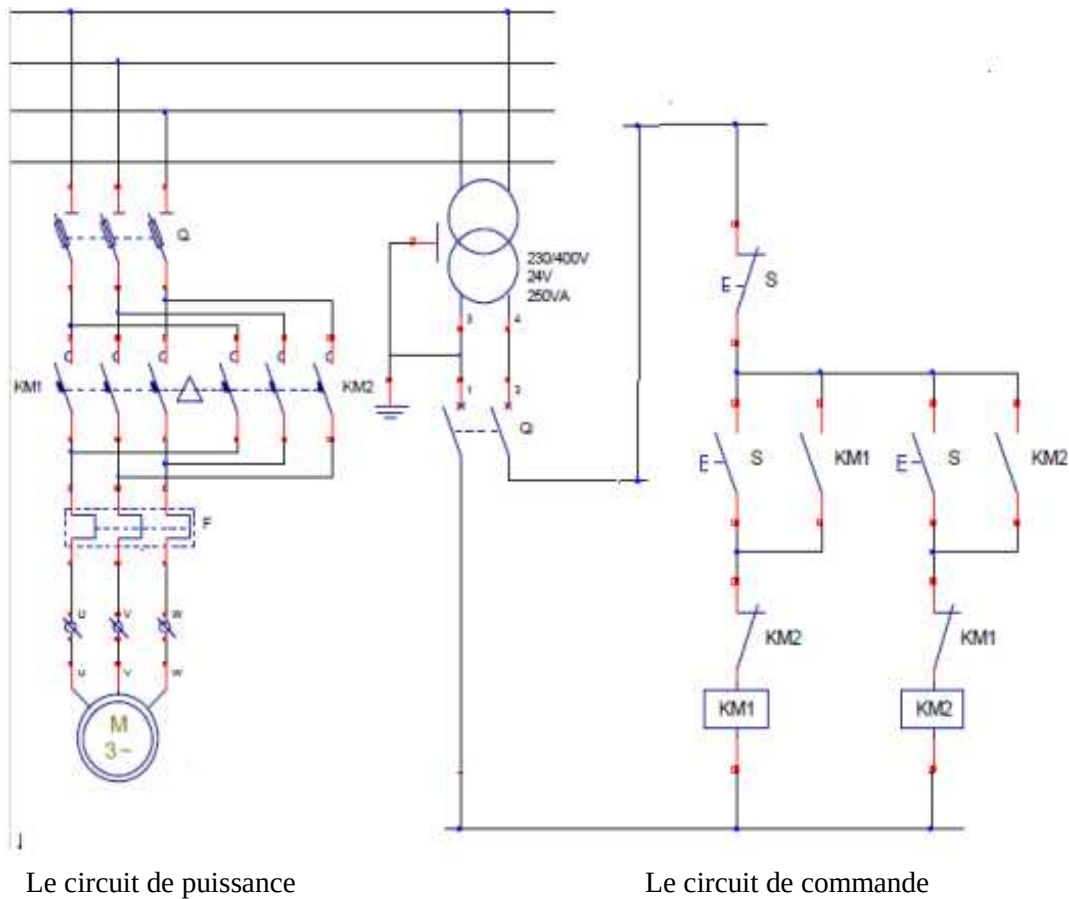


Figure .1.23. Démarrage direct d'un moteur triphasé à deux sens de rotations.

V.3.Moteur à courant continu

Il comprend une partie tournante (le rotor) qui est constitué d'un noyau métallique avec un bobinage en cuivre, et une partie fixe (le stator) qui comporte des aimants permanents qui engendrent un champ magnétique dont le flux traverse le rotor. L'espace étroit entre le rotor et le stator est nommé entrefer (Figure. 1.24).

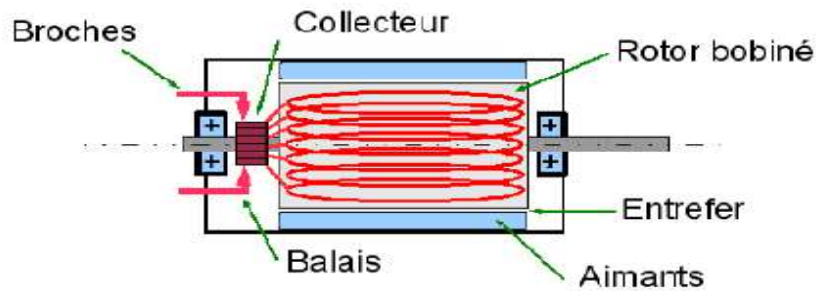


Figure. 1.24 : le moteur à courant continu

1.6.4. Moteur hydraulique

Dans ce type d'actionneur, l'énergie hydraulique fournie par un fluide sous pression est transformée en une énergie mécanique (figure. I.25). Il en résulte un mouvement de rotation sur l'arbre de sortie. Le moteur hydraulique présente deux caractéristiques :

- le couple moteur
- la vitesse de rotation

Ces moteurs entraînent les systèmes mécaniques. Si le couple résistant devient trop important, la pression monte. Quand elle atteint la valeur du réglage du limiteur de pression, le débit retourne au réservoir.

Leur avantage c'est qu'ils développent une grande puissance pour un encombrement réduit mais ils sont coûteux.



Figure. 1.25 : Moteur hydraulique

1.7. Fonctionnement de la machine

Avant la mise en marche en mode automatique, on doit procéder à la phase de préparation, qui consiste à mettre sous-alimentation les différents organes électriques, pneumatiques et

hydrauliques de la machine. Une fois la préparation accomplie, il faut vérifier toutes les positions initiales de chaque bloc de la machine.

1.7.1. Chargement de la tôle à la presse transfert

Pour charger la presse à chaque épuisement de tôle, les étapes suivantes devront être réalisées :

- Poser la bobine centrée au-dessus du berceau (Ve) de charge, et porter le chargeur près du dérouleur.
- Lever le plan du berceau pour mettre la bobine sur le mandrin du dérouleur
- Elargir les mandrins jusqu'à ce que le centrage soit obtenu
- Faire descendre le berceau, puis éloigner le du dérouleur
- Tourner le dérouleur après avoir enlevé le goujon d'arrêt, puis le bloquer après avoir effectué une rotation de 180°.
- Faire descendre le rouleau presseur pour caler la bobine.
- positionner la plaque inférieure de la glissière (tangente à la bobine), et faire sortir la lame.
- Enlever la tête du redresseur et tourner le rouleau presseur, qui permettra à la tôle d'avancer, puis faire descendre la plaque supérieure.
- Serrer la tête du redresseur sur la bande, en abaissant le rouleau trainant.
- Ouvrir les deux plans d'introduction jusqu'à la position "tout ouvert"
- Faire monter le rouleau presseur.
- Régler les guides tôle à l'entrée et à la sortie de la machine à redresser.
- Élever les plans basculants entre le redresseur et l'aménage.
- Faire avancer la bande sur les plans en actionnant les rouleaux entrainants du redresseur jusqu'aux rouleaux de l'aménage.
- Ouvrir les rouleaux de l'aménage.
- Elever la roue du l'encodeur.
- Continuer à faire avancer la bande au-delà de l'aménage, et après la cisaille.
- De même qu'au redresseur, régler les guides-tôle de l'aménage.
- Serre le rouleau de l'aménage.
- Faire descendre la roue du codeur.
- Faire descendre les plans basculants de la fosse.
- Couper le début de la bande, et enlever la partie coupée.

Après avoir effectué toutes ces opérations, l'opérateur procède au réglage des pincettes. Il usine quelques pièces au mode manuelle, ensuite il lance le mode automatique.

1.7.2. Les conditions initiales pour passer à la phase automatique

- Le capteur fin de course du chargeur est actionné (le chargeur est en arrière).
- Le capteur photocellule du redresseur détecte la présence de la tôle.
- Les plans basculant ont repris leur position basse (capteurs de fin de course des plans).
- L'encodeur est en position basse (capteur de fin de course mécanique).
- La cisaille est en position « haut » (capteur de fin de course magnétique).
- La première pièce est détectée par le capteur de fin de course photoélectrique de la table d'aménage.
- L'équilibrage des barres est indiqué par quatre capteurs électromagnétiques.

1.7.3. Le cycle automatique de la machine

Après avoir assuré le chargement de la tôle à la machine, ainsi que la tenue des conditions initiales, on tourne le sélecteur de modes pour choisir le fonctionnement de la machine en mode automatique qui est synchronisé comme suit :

- La vitesse du moteur, faisant tourner les rouleaux du redresseur, est conditionnée par les quatre capteurs photoélectriques de la fosse
- Le moteur du bloc aménage est actionné une fois que la cisaille a regagné sa position haute, et s'arrête une fois la longueur de la tôle correspond au pas réglé au niveau de l'encodeur.
- La cisaille coupe la tôle dès que la longueur de la tôle est égale au pas réglé à l'encodeur, au même temps, le capteur photocellule de la table d'aménage indique que la pièce précédente coupée par la cisaille est dégagée.
- Le déplacement de la pièce de la table d'aménage vers les différentes table porte outils de la presse s'effectue à l'aide des barres porte pincettes et leurs mouvement selon les trois axes est synchronisé avec le mouvement de coulisseau :
 - L'ouverture et la fermeture des barres se fait à l'aide de cames soutenu par des vérins double effet, leurs fin course positive et négative sont munies de deux capteur magnétiques
 - Le mouvement de translation des barres se fait à l'aide d'un dispositif transportant le mouvement de coulisseau composé des cames et d'un système frein/embrayage pneumatique.

1.8. Conclusion

Dans ce chapitre, nous avons étudié la machine « presse de transfert » et ses différents blocs qui la constituent ainsi son principe de fonctionnement.

On a aussi parlé de quelques capteurs, pré-actionneurs, actionneurs et ses différents moteurs utilisés; à fin de faciliter la modélisation qui sera présentée dans le chapitre suivant

Chapitre 2

Modélisation de la machine par GRAFCET

2.1. Introduction

L'automatisation d'un système nécessite la satisfaction d'un cahier des charge élaboré ; en fonction des relations existantes entre la partie commande et la partie opérative ; des conditions de fonctionnement ; ainsi que le dialogue avec l'opérateur. D'après les automaticiens, la modélisation d'un automatisme graphique s'effectue sur l'un des outils graphiques suivant: soit les Réseaux de Pétri(RDP) qui sont utilisés par des chercheurs ou bien le GRAFCET qui est mis en œuvre en industrie.

2.2 Généralités sur GRAFCET

2.2.1. Historique

Le **GRAFCET** (**GRA**phe **F**onctionnel de **C**ommande **E**tape **T**ransition) est né en 1977 des travaux de l'**AFCET** (**A**ssociation **F**rançaise pour la **C**ybernétique **E**conomique et **T**echnique), en tant que synthèse théorique des différents outils existant à cette époque (organigramme, organ phase, diagramme de Girard, réseaux de Pétri, etc....) Mis sous forme actuelle par l'**ADEPA** (**A**gence nationale pour le **D**eveloppement de la **P**roduction **A**utomatisées) en 1979, normalisé sur le plan français en 1990 (norme NF C03-190), sur le plan européen (EN61131) et sur le plan international en 1992 (norme CEI 1131) .[4]

2.2.2. Définition

Le GRAFCET est un modèle de représentation graphique des comportements successifs d'un système logique, préalablement défini par ses entrées et ses sorties. C'est un langage clair, strict mais sans ambiguïté, permettant par exemple au réalisateur de montrer au donneur d'ordre

comment il a compris le cahier des charges. Langage universel, indépendant (dans un premier temps) de la réalisation pratique (peut se "câbler" par séquenceurs, être programmé sur automate voire sur ordinateur).[4]

2.2.3. Composition

Le GRAFCET est défini par un ensemble constitué d'éléments graphiques de bases suivant :

- Etapes
- Transitions
- Liaisons orientées
- Actions associées aux étapes
- Réceptivités associées aux transitions.

2.2.3.1.Étapes

Une étape est une période de temps permettant de réaliser complètement une ou plusieurs actions. Chaque étape est représentée par un carré numéroté. La numérotation est réalisée par des chiffres ou des nombres entiers positifs dans un ordre croissant. Deux étapes différentes ne doivent jamais porter le même numéro. Ils existent deux types d'étapes: [9]

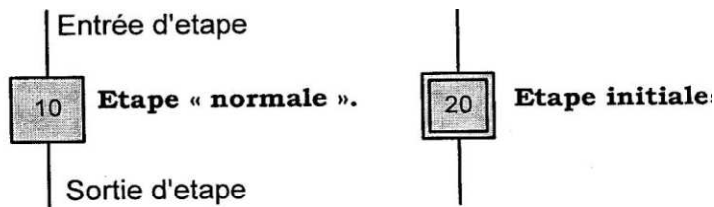


Figure 2.1 : Représentation de deux étapes différentes

Une étape initiale est active au début du cycle, c'est une étape activée sans condition au démarrage.

Par convention, une étape peut être active ou inactive. Si une étape est active, toutes les actions associées à celle-ci seront exécutées. Si une étape est inactive, les actions qui lui sont associées ne peuvent en aucun cas être lancées. On peut associer à chaque étape une variable binaire exprimant son activité :

$X_i = 1$ si i est active Remarquez, que l'on remplace le numéro d'étape par un

$X_i = 0$ si i est inactive i dans le cas présent. Le X étant le symbole d'une étape.



Figure 2.2 : Etape inactive et étape active

Une action est toujours associée à une étape. Elle n'est commandée que si l'étape est activée.

Les actions sont décrites de façon littérale ou symbolique à l'intérieur d'un ou plusieurs rectangles reliés au symbole de l'étape à laquelle elles sont associées.

2.2.3.2. Transitions

Une transition est une condition de passage d'une étape à une autre. Elle n'est que logique (dans son sens vrai ou faux), sans notion de durée. La condition est définie par une réceptivité qui est généralement une expression booléenne (c.à.d. avec des « ET » et des « OU ») de l'état des capteurs.

On représente une transition par un petit trait horizontal sur une liaison verticale. On note à droite la réceptivité. Dans le cas de plusieurs liaisons arrivant sur une transition, on les fait converger sur une grande double barre horizontale, qui n'est qu'une représentation du dessus de la transition. De même pour plusieurs liaisons partant sous une transition.

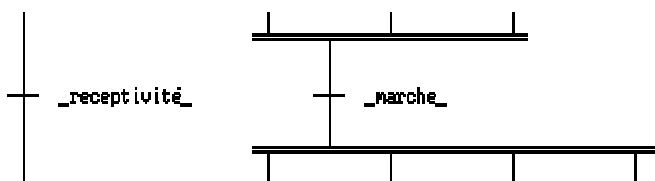


Figure 2.3 : Représentation d'une transition réceptive

2.2.3.3. Liaisons orientées

On a deux types de liaisons orientées soit :

- Les liaisons relient les étapes aux transitions
- Les transitions aux étapes

Par convention, le sens naturel d'évolution est du haut vers le bas. Dans un cas différent, il faut montrer le sens d'évolution par une flèche (Figure 2.4).

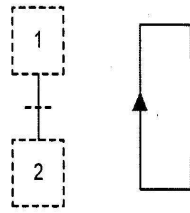


Figure 2.4 : Exemples de liaisons orientées

Pour que ces liaisons puissent être valides il faut suivre ces trois règles :

• **Règle n°1**

Les étapes initiales sont activées inconditionnellement à l'initialisation de l'automatisme, au début du cycle.

• **Règle n°2**

Le franchissement d'une transition ne peut se produire que :

- Si la transition est validée (étapes immédiatement précédentes actives)
- Si la réceptivité associée est vrai (équation logique associée égale à 1)
- Si les deux conditions sont réunies, la transition devient franchissable et est alors obligatoirement franchie.

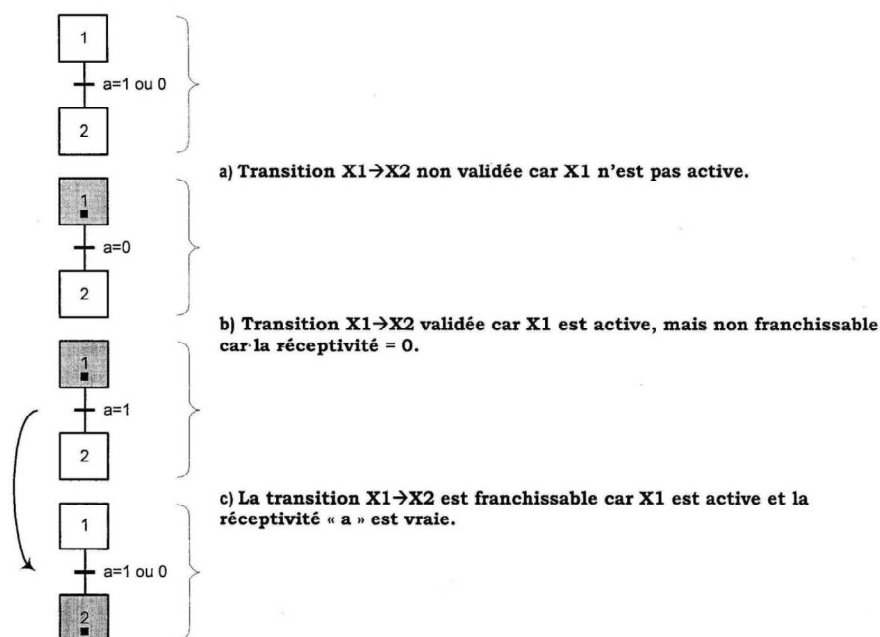


Figure 2.5: Conditions de franchissement

• Règle n°3

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
- Lorsqu'une même action est appelée par plusieurs étapes actives simultanément, alors la valeur de la sortie associée est la disjonction (ou logique) des différentes valeurs données par les différentes étapes. En particulier si cette sortie est émise à vrai par une seule étape alors sa valeur sera vraie.

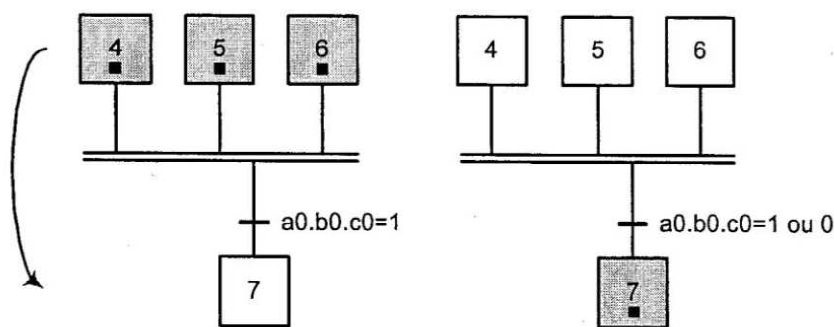


Figure 2.6: Activations ou désactivations simultanées d'une transition

2.2.3.4. Actions

Une action pouvait être associée à une étape. Les actions servent à émettre des ordres vers la partie opérative. Une action est une sortie du système logique que nous modélisons, elle correspond donc à une composante S_m du vecteur de sortie. Ces actions peuvent être de trois types :

- **Actions continues.**
- **Actions conditionnelles** : qui peuvent être classées en :
 - ✓ action conditionnelle simple,
 - ✓ action conditionnelle retardée,
 - ✓ action conditionnelle limitée dans le temps.

• Actions mémorisées

2.2.3.4.1. Actions continues

La sortie S correspondante est émise à vrai tant que l'étape associée est active. Lorsque l'étape devient inactive la sortie est émise à faux.

2.2.3.4.2. Actions conditionnelles

Une action conditionnelle n'est exécutée que si l'étape n associée est active et si la condition associée est vraie. Cette condition notée c_n est une expression booléenne portant sur l'ensemble des entrées, des variables d'étapes et exprimée à l'aide des opérateurs logiques et, ou, non. On fait apparaître dans le cadre de l'action attaché à l'étape, le caractère conditionnel qui peut être de trois types :

• Type C (Condition)

Une action conditionnelle simple est l'exécution de l'action est soumis à une condition, (représentation normalisée à la figure 2.7).

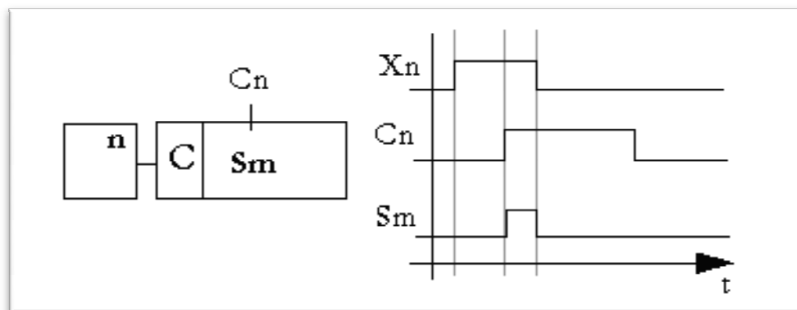


Figure 2.7: Action Conditionnelle « type C »

• Type D (Delay)

Une action conditionnelle retardée sur l'étape n est une action conditionnelle où la condition C_n s'écrit : " $t_1/x_n/d$ " avec d , délai associé au retard (représentation normalisée à la figure 2.8).

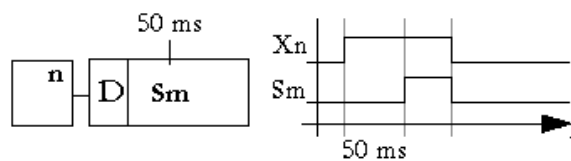


Figure 2.8: Action Retardée

• Type L (Limited)

Une action conditionnelle limitée dans le temps sur l'étape n est une action conditionnelle où la condition c_n s'écrit non (L/x_n) avec L , durée associée à la limitation temporelle (représentation normalisée à la figure 2.9).

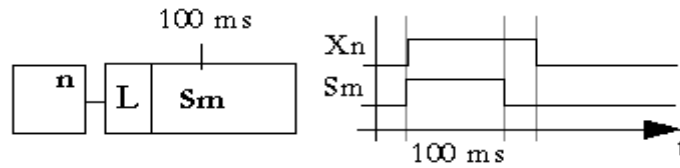


Figure 2.9 : Action Limitée

• Remarque

Toutes ces actions conditionnelles peuvent se regrouper sous la forme des actions où la condition c_n s'écrit comme une expression booléenne portant sur l'ensemble des entrées, des variables d'étapes et exprimée à l'aide des opérateurs et, ou, non ainsi que l'opérateur à retard " $t_1/x_n/t_2$ ".

2.2.3.4.2. Actions mémorisées

Une étape à action mémorisée permet de mettre la sortie correspondante dans un état spécifié lors de son activation. Sa désactivation ne remet pas la sortie associée à son état d'origine : le passage dans un autre état de cette sortie devra être décrit explicitement par une autre étape. Ainsi la mémorisation à l'état vrai d'une sortie se symbolise par la lettre S (set) et la mémorisation à l'état faux par la lettre R (reset) dans le cadre de l'action attachée à l'étape (Figure 2.10).

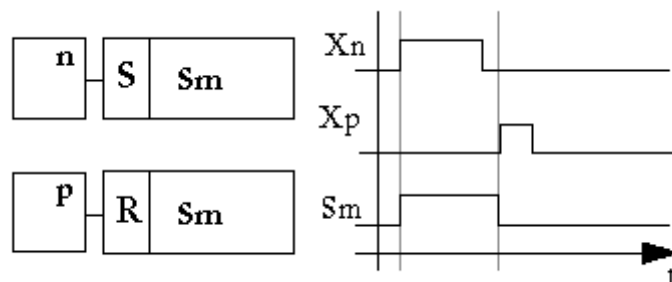


Figure 2.10 : Actions mémorisées

2.2.3.5. Réceptivités

Une réceptivité est associée à chaque transition (l'absence de réceptivité est en fait la réceptivité toujours vrai). C'est une condition qui détermine la possibilité ou non d'évolution du système par cette transition.

Une réceptivité s'exprime comme étant une expression booléenne écrite à l'aide de variables d'entrées E_i , des variables d'étapes E_i , des opérateurs logiques « ET », « OU », non ainsi que de l'opérateur à retard " $t_1/x_n/t_2$ ", auquel on peut rajouter les opérateurs front montant et front descendant. Ces opérateurs permettent d'introduire le concept d'événement (Figure 2.11). Ils expriment le changement d'état d'une variable booléenne.

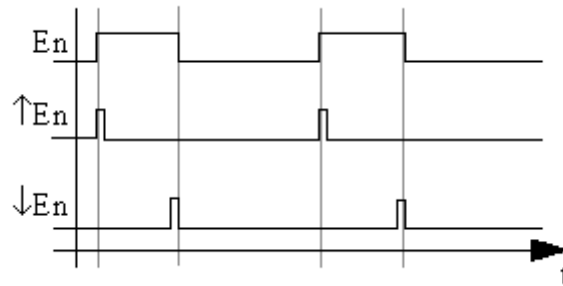


Figure 2.11 : Opérateurs Fronts montants et descendants

2.2.4. Macro-étapes

Le concept de la macro-étape permet des descriptions par raffinement successifs. Ainsi plusieurs niveaux de représentation peuvent être mis en œuvre. Le premier niveau exprimant globalement la fonction à remplir sans se soucier de tous les détails superflus qui seront décrits dans les niveaux suivants, correspondant à une analyse plus fine. Finalement le dernier niveau pourra être celui correspondant à l'implémentation de la partie commande dont on spécifie le comportement.

Une macro-étape M_i est l'unique représentation d'un ensemble unique d'étapes et de transitions nommé "expansion de « M_i »".

Une macro-étape obéit aux règles suivantes:

- l'expansion de ME comporte une étape particulière (et une seule) dite étape d'entrée et une étape particulière (et une seule) dite étape de sortie.
- l'étape d'entrée a la propriété suivante: tout franchissement d'une transition amont de la macro-étape, active l'étape d'entrée de son expansion.
- l'étape de sortie a la propriété suivante: elle participe à la validation des transitions aval de la macro-étape.
- en dehors des transitions amont et aval de ME, il n'existe aucune liaison structurale entre, d'une part une étape ou une transition de l'expansion ME, et d'autre part, une étape ou une transition n'appartenant pas à ME.

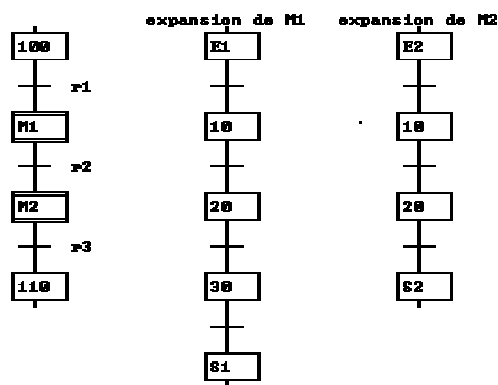


Figure 2.12: Macro- étape

2.2.5. Règles d'évolution du GRAFCET

Un GRAFCET possède un comportement dynamique dirigé par cinq règles, elles précisent les causes et les effets du franchissement des transitions.

- Règle N°1
- Situation initiale

La situation initiale d'un GRAFCET caractérise le comportement initial de la partie commande vis-à-vis de la partie opérative, de l'opérateur et/ou des éléments extérieurs. Elle correspond aux étapes actives au début du fonctionnement : ces étapes sont les étapes initiales (Figure 2.13).

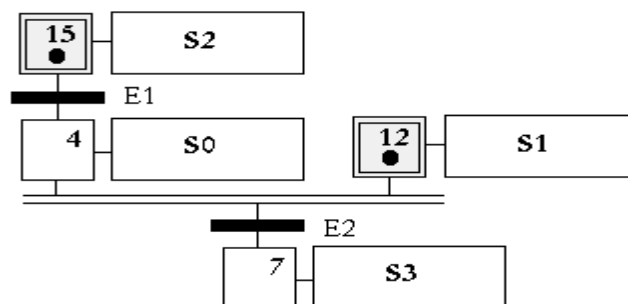


Figure 2.13: Situation initiale

- Règle 2
- Franchissement d'une transition

Une transition est dite validée lorsque toutes les étapes amont (immédiatement précédentes reliées à cette transition) sont actives.

Le franchissement d'une transition se produit :

- lorsque la transition est validée
- la réceptivité associée à cette transition est vraie
- L'exemple de (Figure 2.14) montre une transition non validée car l'étape 4 n'est pas active
- L'exemple de (Figure 2.15) montre une transition validée mais non-franchissable avant t_1 , elle devient franchissable à t_1 car E_2 devient vraie

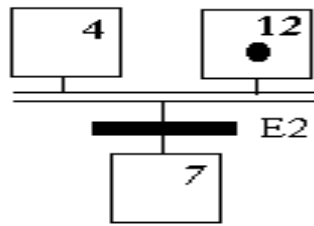


Figure 2.14: Transition non validée

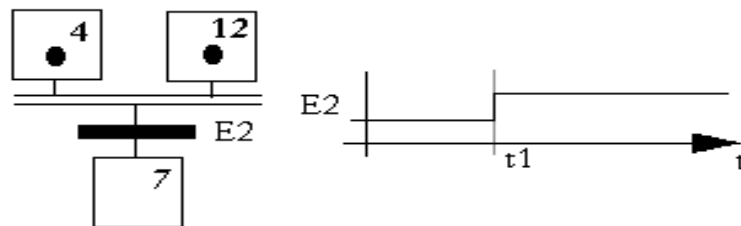


Figure 2.15 : Transition validée puis franchissable à t_1

• Règle 3

• Evolution des étapes actives

Le franchissement d'une transition entraîne simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes (Figure 2.16).

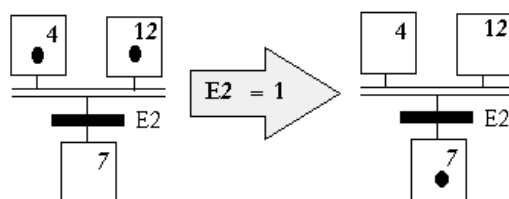


Figure 2.16 : Franchissement d'une transition

- Règle 4

- Evolution simultanée

Plusieurs transitions simultanément franchissables sont simultanément franchies (figure 2.16).

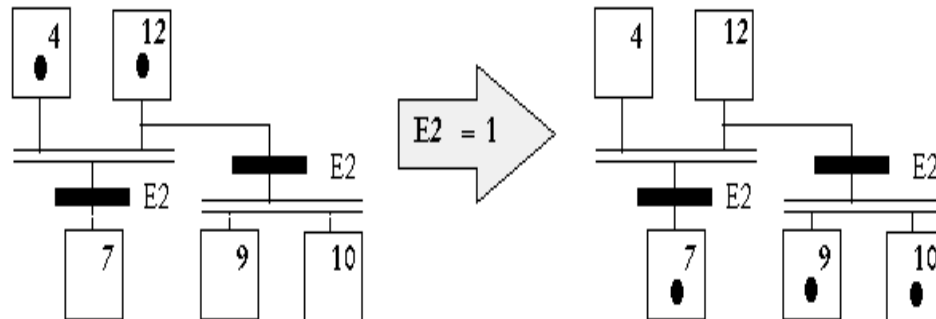


Figure 2.17 : Franchissements simultanés

- Règle 5

- Activation et désactivation simultanée d'une étape

Si au cours du fonctionnement la même étape est simultanément activée et désactivée elle reste active (Figure 2.18).

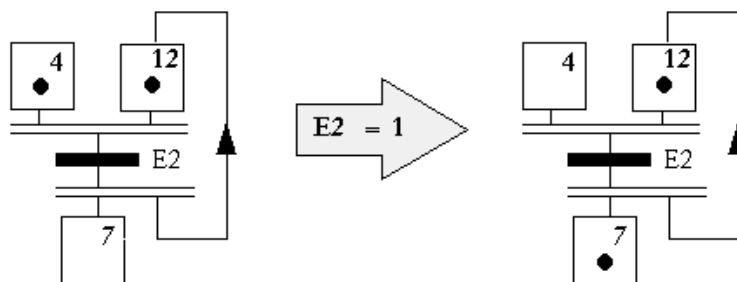


Figure 2.18: Activation et désactivation simultanées d'étape (étape 12)

Remarque

Il est important de noter que :

- la durée d'un franchissement de transition est de durée très petite (aussi petite que l'on veut) mais non nulle. Ainsi si deux transitions successives (séparée par une étape) ont pour réceptivité le

même front d'une variable, alors il faudra deux fronts de cette variable pour franchir les deux transitions (Figure 2.19). Ce temps est en relation directe avec la durée d'un front.

- Le GRAFCET fait l'hypothèse d'un monde asynchrone : deux événements non corrélés ne peuvent survenir simultanément. Soit formellement, l'expression logique E_i et E_j est toujours évaluée à faux.

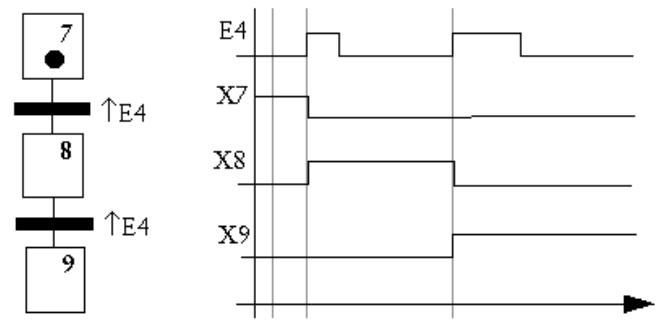


Figure 2.19: Règle de franchissement d'une transition

2.2.6. Niveaux du GRAFCET

On ne peut concevoir et réaliser un automate que si le cahier de charge établi auparavant est clair, précis, sans ambiguïté ni omission du rôle et des performances de l'équipement à réaliser. Afin d'aboutir à ce résultat, il est souhaitable de diviser la description en deux niveaux successifs et complémentaires.

- **Niveau 1**
- **Spécifications fonctionnelles**

Ce niveau décrit le comportement de la partie commande vis-à-vis de la partie opérative. Cette description est établie par des spécifications fonctionnelles permettant de comprendre ce que l'automatisme doit faire, face aux différentes situations pouvant se présenter.

- **Niveau 2**

Spécifications technologiques

Pour décrire précisément comment l'automatisme devra physiquement s'insérer dans l'ensemble qu'il constitue avec son environnement, des spécifications technologiques ont été apportées en complément des spécifications fonctionnelles. Cela permettra un automate pilotant réellement la partie opérative.

• Niveau 3

Dans ce cas on reprend le GRAFCET de niveau 2, en affectant les informations aux étiquettes d'entrée de l'automate et les ordres aux étiquettes de sortie de l'automate. Il s'adapte aux caractéristiques de traitement d'un API.

2.2.7. Mise en équation d'un GRAFCET

Pour passer de l'étape de modélisation du procédé par GRAFCET comme le montre la (Figure 2.20) à l'étape de programmation par l'un des langages acceptés par l'automate, on traduit notre GRAFCET de niveau 2 par des équations combinatoires. Considérant une étape x_n notée comme suit :

$x_n = 1$ si l'étape n est active

$x_n = 0$ si l'étape n est inactif

La réceptivité t_n , étant une variable binaire, a pour valeur :

$t_n = 1$ si la réceptivité est vrai

$t_n = 0$ si la réceptivité est faux

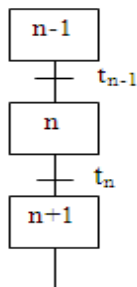


Figure 2.20: représentation générale d'un GRAFCET

- Une transition est soit validée, soit non validée. Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives. Elle ne peut être franchie que :
 - Lorsqu'elle est validée
 - La réceptivité associée à la transition est vraie

La traduction de cette règle donne la Condition d'Activation de l'étape n :

$$C_A x_n = x_{n-1} t_{n-1} \dots \dots \dots (2.1)$$

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

La traduction de cette règle donne la Condition de Désactivation de l'étape n :

$$x_n = x_n t_n = x_{n+1} \dots \dots \dots (2.2)$$

Si la c_A et la c_D de l'étape n sont faux, l'étape n reste dans son état. C'est à dire que l'état de x_n à l'instant $t + \delta t$ dépend de l'état précédent de x_n à l'instant t .

- En combinant les deux équations (1) et (2), on obtient l'équation suivante :

$$x_n = c_A x_n + \overline{c_A} \overline{x_n} \cdot x_n \dots \dots (2.3)$$

2.3. Notation des entrées/sorties

Lors de l'établissement du GRAFCET, l'utilisation de noms explicites pour les entrées du système modélisé ; ainsi que pour les sorties (ces actions). Lors du passage au GRAFCET de réalisation, on utilise plutôt des noms logiques pour les entrées/sorties du système.

Par convention, les entrées seront rangées dans un vecteur (vecteur d'entrée) et nommées E_i où i est le rang de l'entrée considérée dans le vecteur d'entrée. De même les sorties seront rangées dans un vecteur (vecteur de sortie) et nommées S_i où i est le rang de la sortie considéré dans le vecteur de sortie.

Par la suite on utilisera ces notations E_i et S_i afin de désigner les entrées et les sorties de la partie commande pour des commodités d'écriture.

La (Figure 2.21) montre un exemple de nommage symbolique des entrées/sorties pour le GRAFCET de (Figure 2.20). [10]

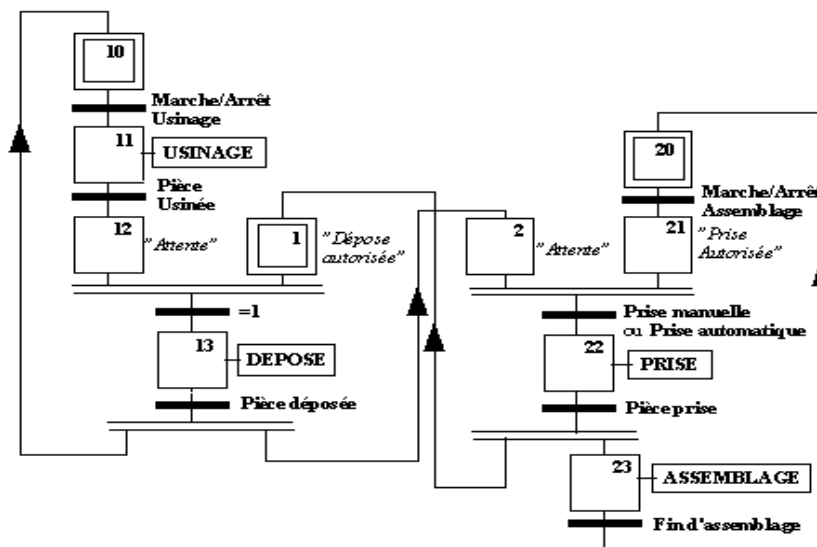


Figure 2.21 : Exemple d'un GRAFCET qui sera converti en partie commande

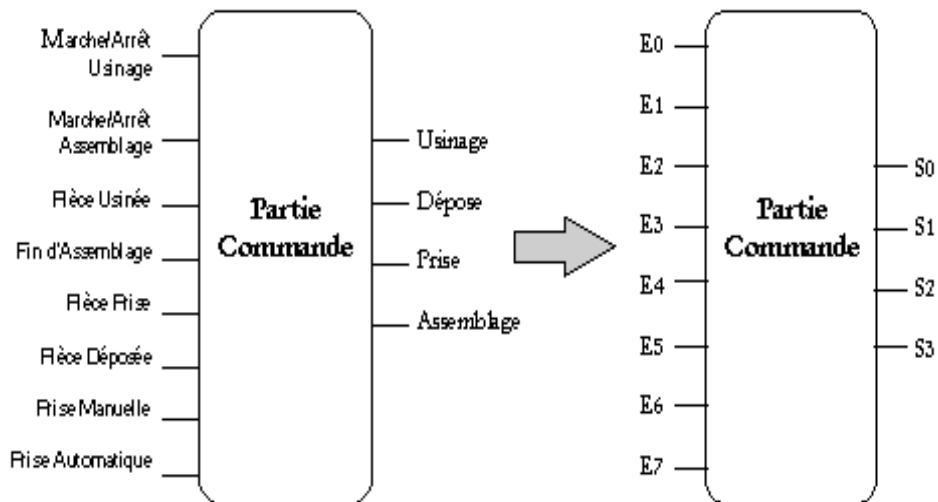


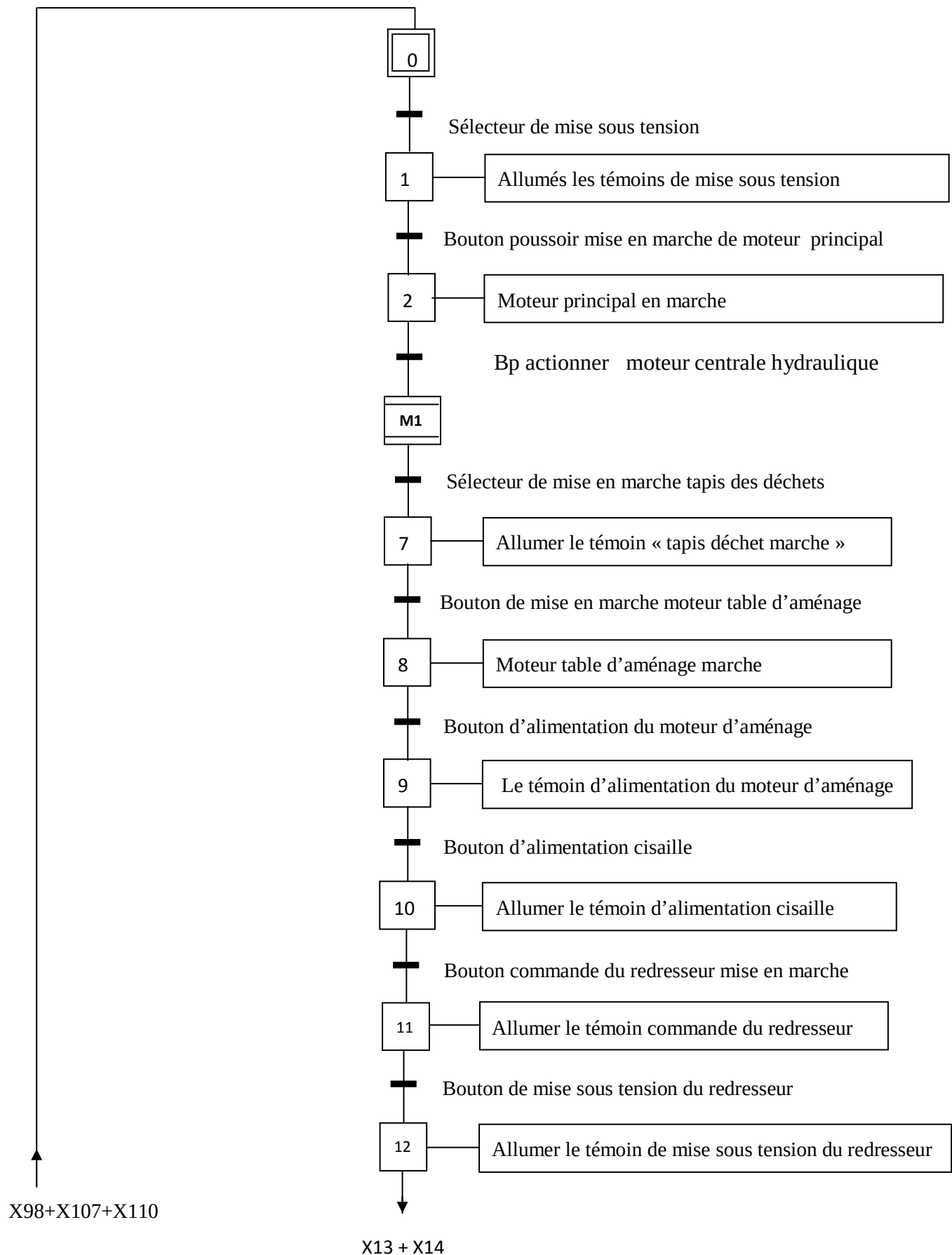
Figure 2.22 : Nommage symbolique des entrées/sorties

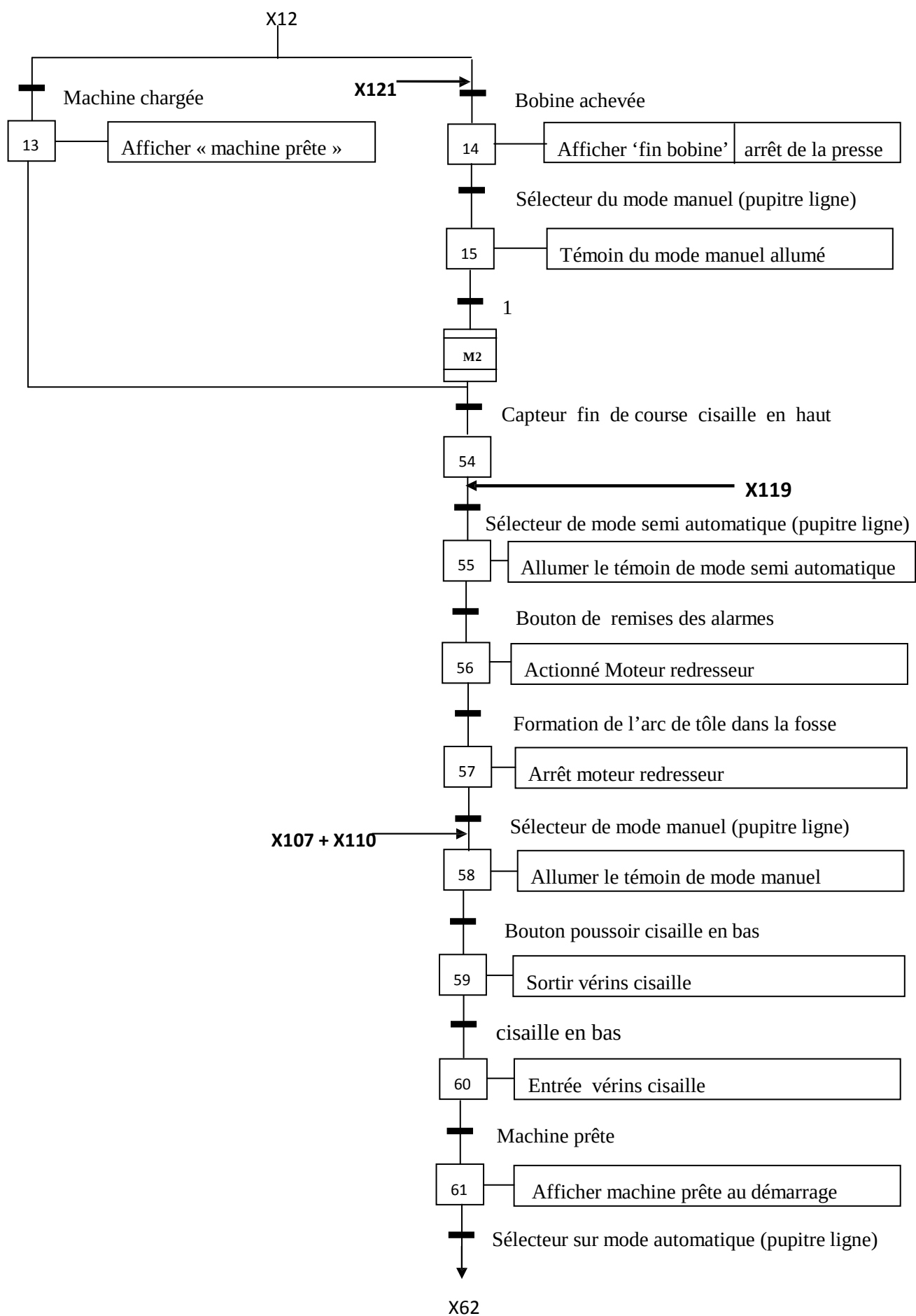
2.5.Conclusion

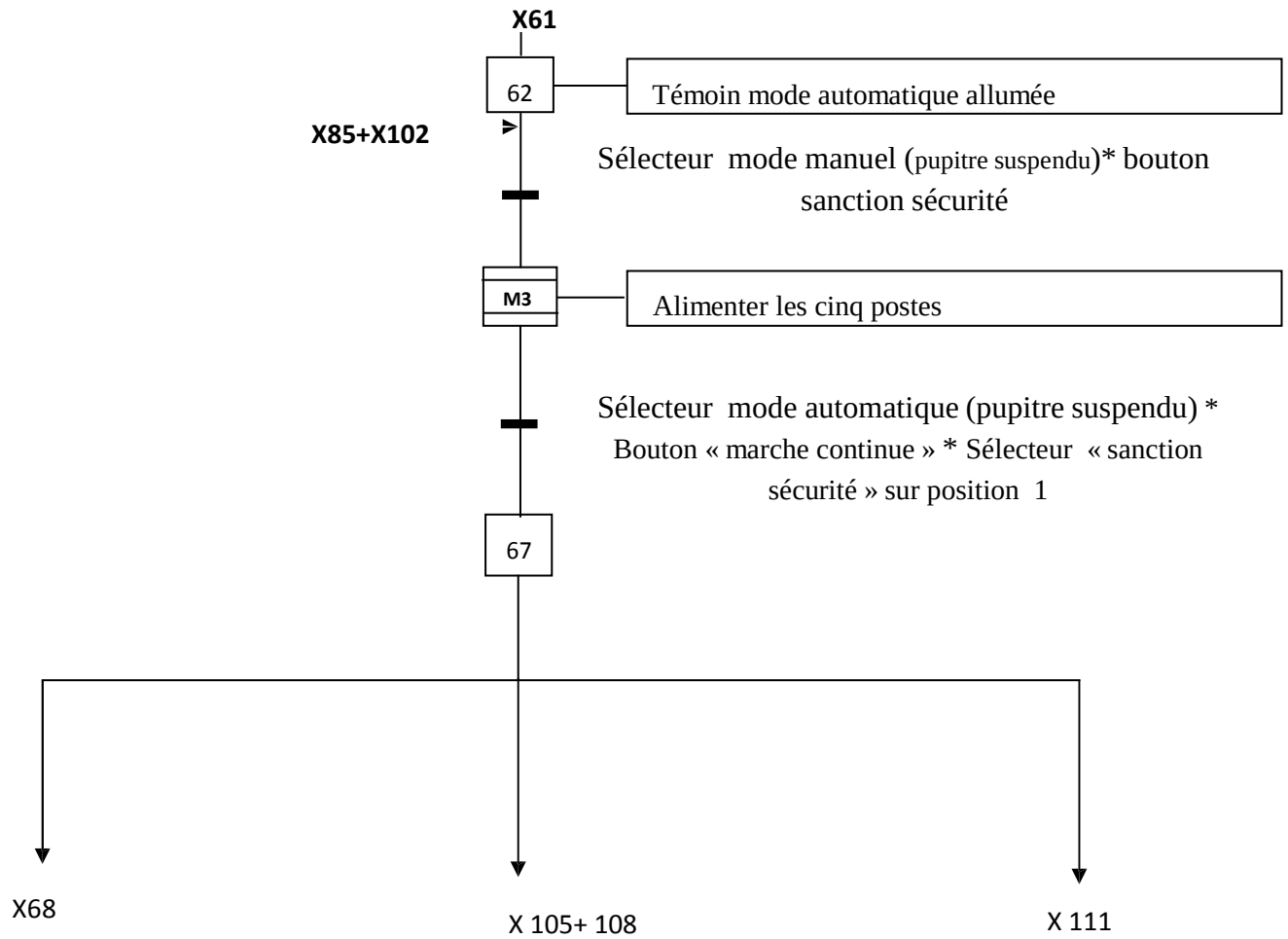
Dans notre 2^{ème} chapitre nous avons modélisé le procédé de commande à l'aide du GRAFCET; d'ailleurs nous avons d'abord élaboré le niveau 1, afin de bien expliquer le fonctionnement du système ; ensuite nous avons présenté le GRAFCET de niveau 2 qui est moins encombrant un avec lequel la modélisation est effectuée.

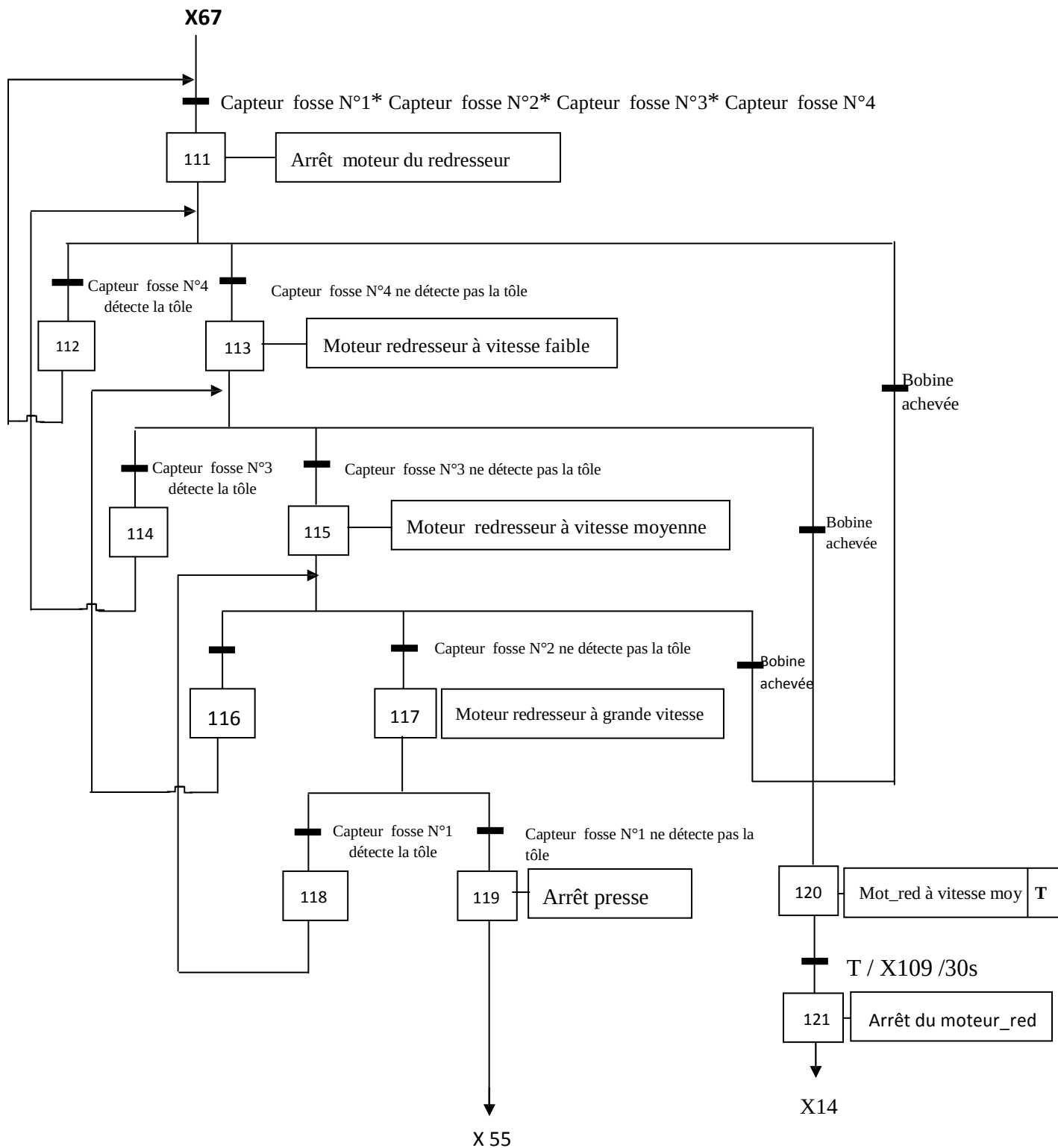
Nous constatons que le GRAFCET outil de modélisation intéressant parce qu'il facilite la transition entre le cahier des charges fonctionnel à l'un des langages d'implémentation ; ce qui va nous permettre de transcrire la programmation de la partie commande qui pilotera le procédé à l'aide du logiciel STEP7, plus précisément c'est l'objet de notre prochain chapitre.

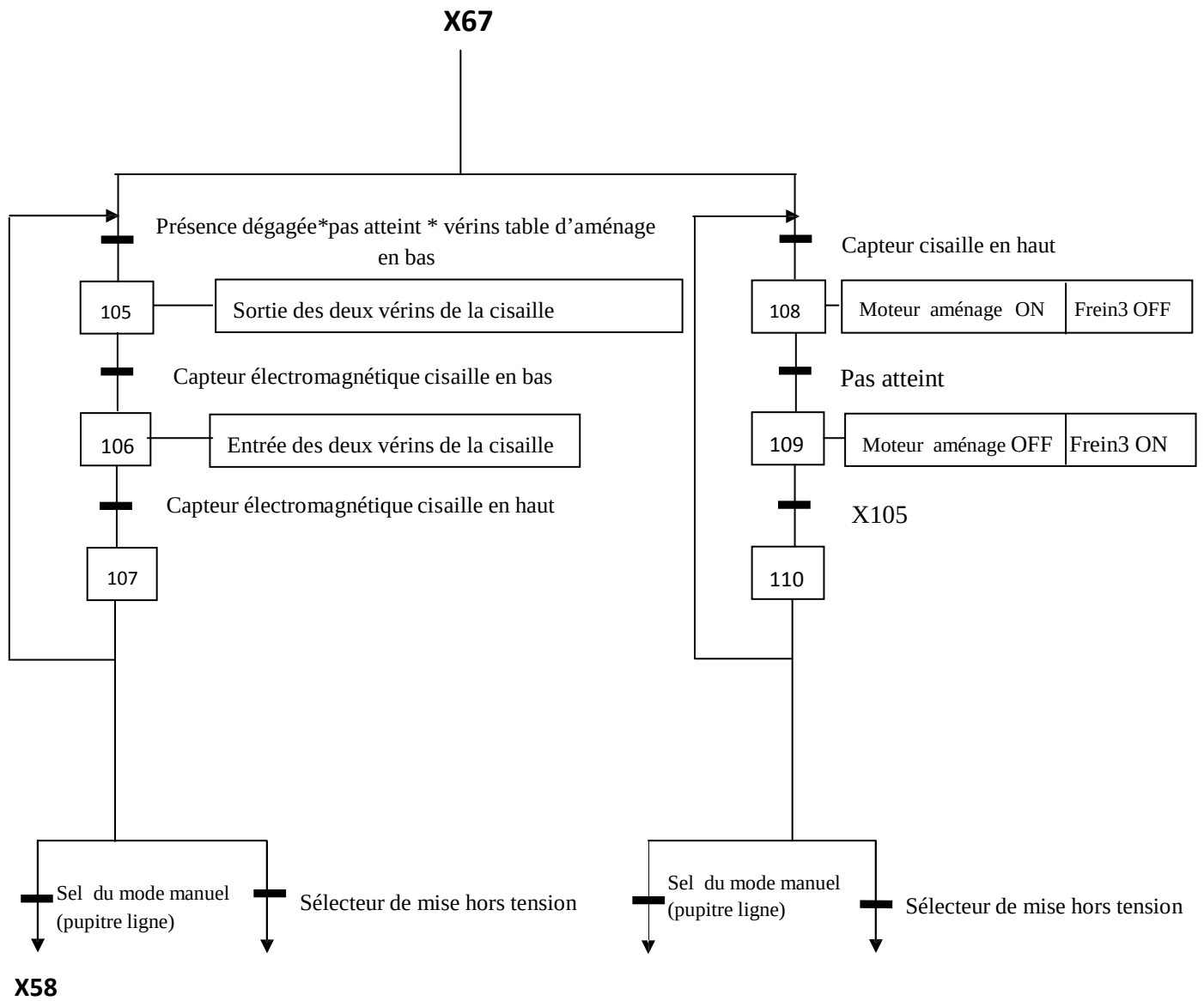
1- Le Grafcet niveau 1 de la machine:











ERROR: syntaxerror
OFFENDING COMMAND: --nostringval--

STACK:

1004
10406
10

Chapitre 3

Programmation de l'automate S7-300

3 .1. Introduction

Après modélisation par GRAFCET, nous arrivons à l'étape décisive de notre travail qui consiste à valider le programme développé par simulation et vérification de son bon fonctionnement ; pour cela nous avons choisi l'automate programmable S7-300 « logiciel optionnel du STEP7 ».

L'objectif majeur de cette étape est de tester les programmes avant d'établir la liaison avec le système à commandé, car son application peut causer des dommages matériels ou des blessures corporelles en cas d'erreur de programmation.

3.2. Automates Programmables Industriels(API)

3.2.1. Définition d'un API

Automate Programmable Industriel API(en anglais **Programmable Logic Controller PLC**) est un dispositif électronique programmable destiné à la commande de processus industriels en temps réel par un traitement séquentiel. Il envoie des ordres vers les pré-actionneurs, à partir de données d'entrées (capteurs), de consignes et d'un programme informatique.

3.2.2. Structure d'un système automatisé industriel

Un système automatisé se compose de deux parties qui coopèrent:

- **Partie opérative**

Constituée du processus à commander, des actionneurs qui agissent sur ce processus et des capteurs permettant de mesurer son état.

- **Partie commande**

Élabore les ordres pour les actionneurs en fonction des informations issues des capteurs et des consignes. Cette partie commande peut être réalisée par des circuits câblés, ou par des dispositifs programmables (automates, calculateurs).

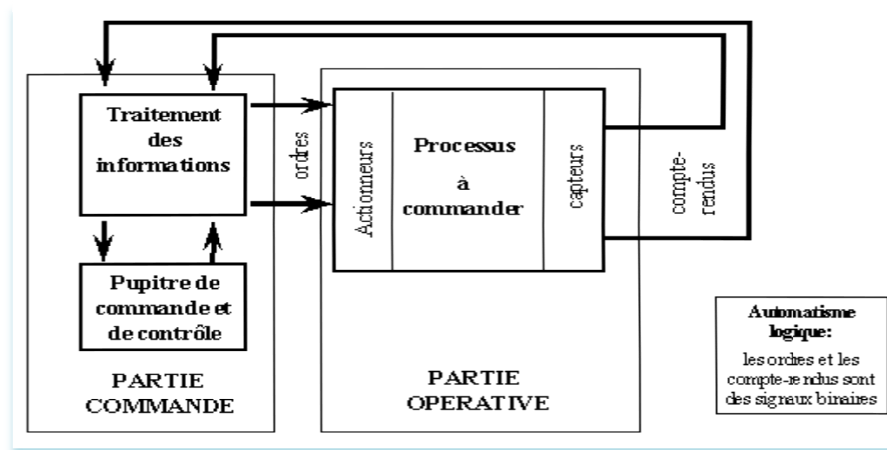


Figure 3.1: Structure d'un système automatisé industriel

3.2.3. Rôle d'un automate programmable dans les systèmes automatisés industriels

Les automates programmables industriels (API) représentent l'outil de base de l'automatisation des systèmes de production. Un API permet de piloter un système de production conformément à un programme placé dans sa mémoire. Sa flexibilité explique son large domaine d'utilisation, il est généralement placé en ambiance industrielle, où il représente le cœur de la partie commande d'un système automatisé. Il est en relation avec les autres parties du système grâce à son interface d'entrée-sortie (Figure 3.2).

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

3.2.4. Architecture interne d'un automate programmable

La structure interne d'un API peut se présentée comme indiqué dans la (figure 3.2)

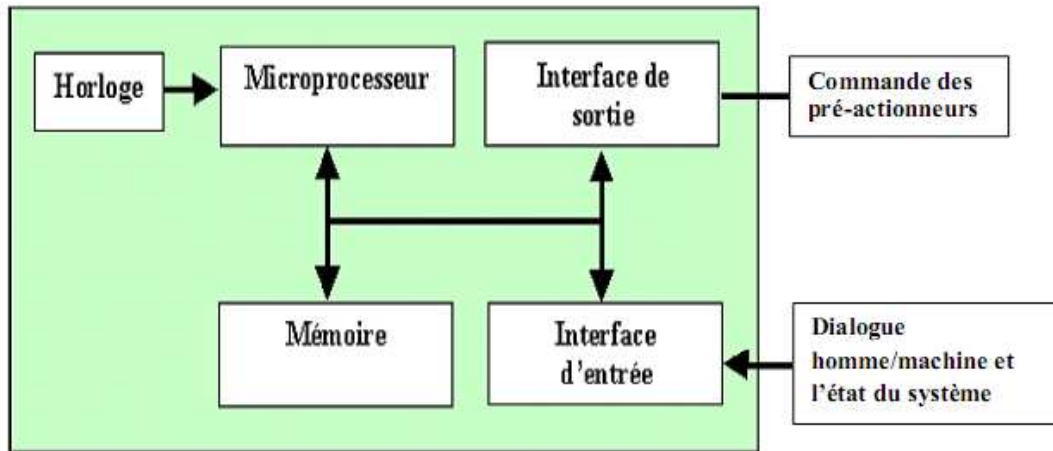


Figure 3.2 : Architecture interne d'un automate programmable

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

Un API se compose donc de trois grandes parties :

- o Le processeur.
- o La zone mémoire.
- o Les interfaces Entrées/Sorties

Les interfaces d'entrées reçoivent l'information en provenance du capteur, éliminent les parasites et isolent électriquement l'unité de commande de la partie opérative. (Figure 3.3).

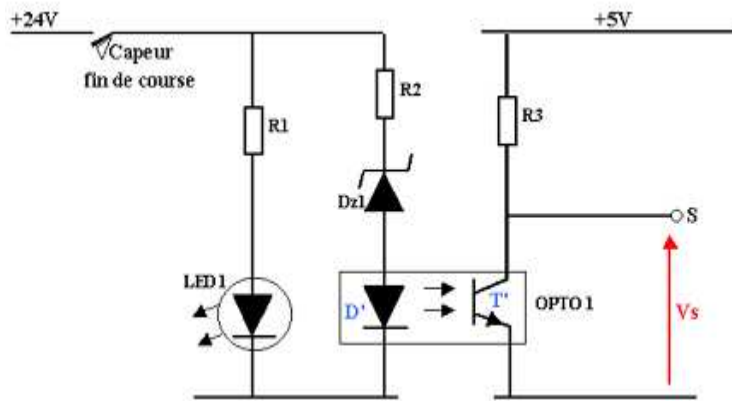


Figure 3.3 : L'interface d'entrée d'un API

Les interfaces de sortie commandent les pré-actionneurs et éléments de signalisation du système, et adaptent les niveaux de tension de l'unité de commande à celle de la partie opérative du système en garantissant une isolation galvanique entre ces derniers (Figure 3.4).

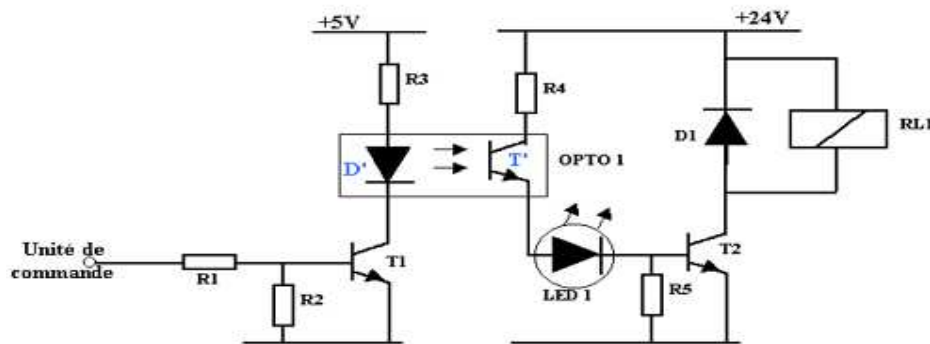


Figure 3.4 :L'interface de sortie d'un API

3.2.5. Câblage d'un automate

3.2.5.1. Alimentation de l'automate (Figure 3.5)

L'alimentation de l'automate est prise entre une ligne de réseau L1 et un neutre pour avoir 220V, une protection électrique PE serre comme prise de terre. Pour une protection contre les courts circuit, une protection thermique protège l'automate contre les intensités. Un transformateur 230/24V assure l'alimentation des sorties (la signalisation, alarme et les bobines des relais...) .Un maintien de la masse (0V) est réalisée par KM1. KA1 contrôle la mise sous tension (24V) des sorties de l'automate selon les ordres du chien de garde. [2]

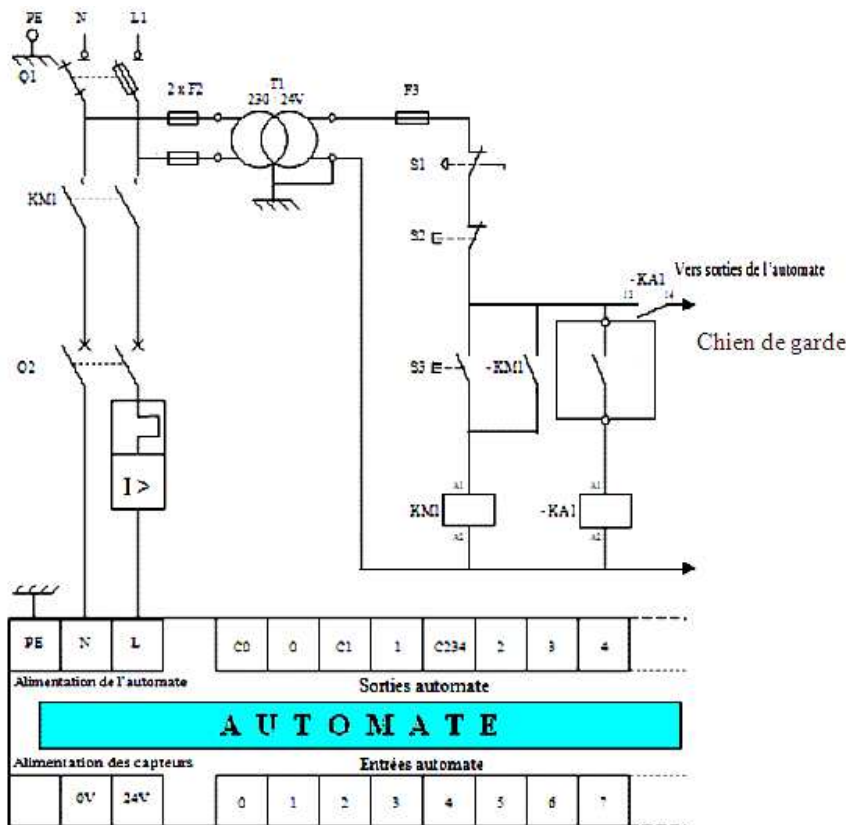


Figure 3.5 : Alimentation de l'automate

3.2.5.2. Alimentation des entrées de l'automate

L'automate est muni d'une alimentation pour les capteurs. Les entrées sont connectées au 0 V (commun) (Figure 3.6). Les informations des capteurs sont traitées par les interfaces d'entrées. La tension 24V est reliée directement aux différents capteurs et l'information passe à travers le circuit d'entrée. [2]

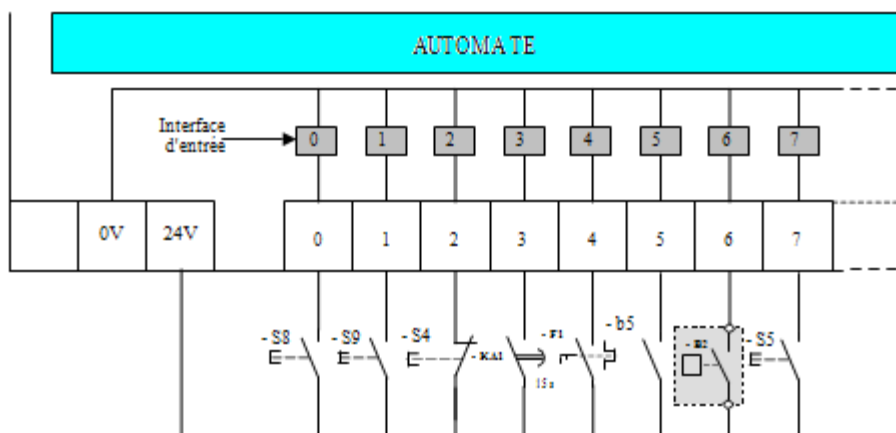


Figure 3.6 : Alimentation des entrées

3.2.5.3. Alimentation des sorties de l'automate

Les interfaces de sorties permettent d'alimenter le divers pré actionneurs (témoin, alarme, relai voir la figure 3.7). 0V est relié aux pré-actionneurs ,24V est relie au commun des sorties (C 0, C1 C234...) les sortie sont actionner par des circuits de sorties.

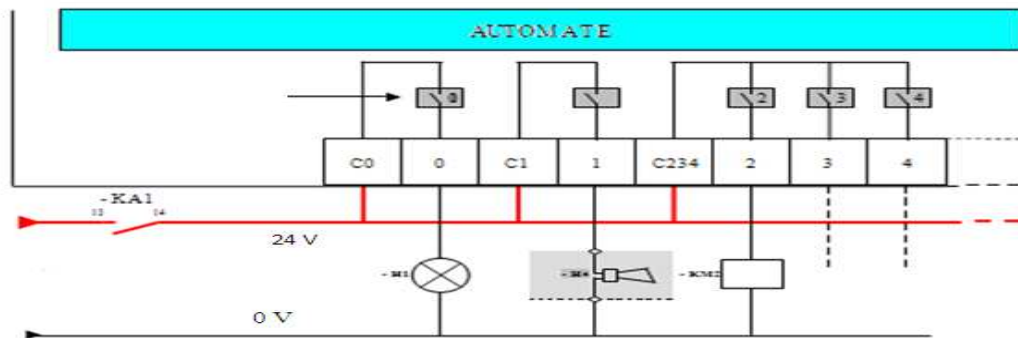


Figure 3.7: Alimentation des sorties

3.2.6. Langages de programmation des API

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

-Instruction List(IL)

C'est un langage textuel de type assembleur.

-Structured Text (ST)

C'est un langage textuel structuré similaire au pascal.

-Ladder Diagram (LD)

C'est un langage graphique, très utilisé au milieu industriel, car il s'inspire des circuits de commande basés sur la logique électrique, les équations combinatoires étant câblées à l'aide de contacts et de relais. Un programme est décrit par un diagramme sous forme d'échelle. Chaque échelon de l'échelle contient un ensemble de symboles graphiques qui peuvent être des contacts ou des bobines. Un contact permet la lecture d'une variable booléenne tandis qu'une bobine permet d'affecter une valeur à une variable booléenne.

-Sequential Function Charts (SFC)

C'est un langage graphique permettant de structurer tout comportement séquentiel pouvant être décrit dans l'un des quatre autres langages de la norme.

-Function Block Diagram (FBD)

C'est un langage graphique permettant d'exprimer le comportement des fonctions, des blocs fonctionnels ou des programmes comme un ensemble de boîtes noires interconnectées (à la manière des portes logiques en électronique).

3.2.7. Choix d'un automate programmable

Le choix d'un automate programmable est en premier lieu :

- le choix d'une société ou d'un groupe et les contacts commerciaux et expériences vécues sont déjà un point de départ.
- Le personnel de maintenance doit toutefois être formé sur ces matériels et une trop grande diversité des matériels peut avoir de graves répercussions. Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages.
- La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables

Il faut ensuite quantifier les besoins :

- **Nombre d'entrées/sorties**

Le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées/sorties nécessaires devient élevé.

- **Type de processeur**

La taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.

- **Fonctions ou modules spéciaux**

Certaines cartes permettront de "soulager" le processeur et devront offrir les caractéristiques souhaitées (résolution ...). [2], [5]

- **Fonctions de communication**

L'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ...).

3.3 Automate programmable S7-300

3.3.1. Critères de choix de l'automate

D'après le cahier des charges établi, l'automate choisi doit répondre à certains critères :

- La capacité de traitement du processeur.
- Le nombre entrées/sorties.
- La nature des entrées/sorties (numériques, analogiques, booléennes).
- La fiabilité.
- La qualité du service après vente
- La durée de garantie

3.3.2.présentation générale

L'automate S7-300 est un mini automate modulaire pour des applications d'entrées et de milieu de gamme fabriqué par la firme SIEMENS, on peut le composer en fonction de nos besoins à partir d'un vaste éventail de modules.

SIMATIC S7 désigne un produit de la société SIEMENS qui sont des appareils fabriqués en série, conçus indépendamment d'une tâche précise. Tout les éléments logiques, fonctions de mémoire, temporisations, compteurs...etc., nécessaire à l'automatisation sont prévus par le fabricant et sont intégrés à l'automate. Ils se distinguent principalement par le nombre des :

- Entrées et sorties.
- Compteurs.
- Temporisation.
- Mémentos.
- La vitesse de travail.

3.3.3. Caractéristiques

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

- Gamme diversifiée de la CPU.

- Gamme complète du module.
- Possibilité d'exécution jusqu'à 32 modules.
- Bus de fond de panier intégré en module.
- Possibilité de mise en réseaux avec MPI, PROFIBUS ou INDUSTRIAL ETHERNET.
- Raccordement central de la PG avec accès à tous les modules.
- Liberté de montage au différent emplacement.
- Configuration et paramétrage à l'aide de l'outil configuration matériels.

Plusieurs automates programmables S7-300 peuvent communiquer entre eux aux moyens d'un câble-bus PROFIBUS pour une configuration décentralisée.

3.3.4.1. Constitution interne

Toute "l'intelligence" de l'automate réside dans sa CPU (unité centrale). Elle se compose des éléments suivant:

- Un microprocesseur (traitement des informations) avec une horloge interne (fonction de temporisation, synchronisation des opérations) Plusieurs entrées (réception des données externes).
- Plusieurs sorties (émission des données vers l'extérieur, - pour relais, écran).
- Mémoires, on a deux types de mémoire :
 - Une mémoire ROM ou EPROM (contient le système d'exploitation)
 - Une mémoire RAM divisée en zone
 - o Données utiles au programme
 - o Données utiles au programmeur

3.3.4.2 Constitution externe

L'automate programmable industriel (API) S7-300 est composé de différents modules, comme le montre la (figure 3.8).

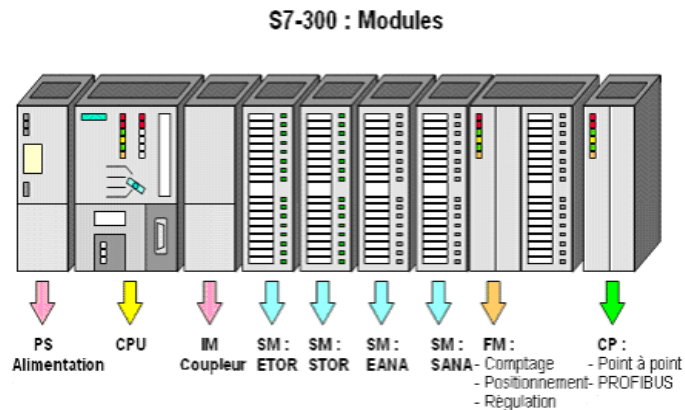


Figure 3.8 : Constitution externe de l'automate

Modules d'alimentation (PS)

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

Désignation	CS	Tension à la sortie	Tension à l'entrée
PS307	2A	DC24v	AC120v/230v
PS307	5A	DC24v	AC120v/230v
PS307	10A	DC24v	AC120v/230v

Figure 3.9 :Module d'alimentation



Figure 3.10:alimentation d'un API

▪ Description de la CPU

La CPU (Central Processing Unit) est le cerveau de l'automate, elle lit les états des entrées, ensuite, elle exécute le programme utilisateur en mémoire et enfin, elle commande les sorties (action).

Elle comporte une unité de commande et de calcul, des mémoires, un programme système et des interfaces vers les modules de signaux.

La CPU constituée de :

▪ Interface MPI

Chaque 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 fonctionnement

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

- ✓ **RUN-P** : exécution de programme, accès en écriture et en lecture avec la PG.
- ✓ **RUN** : exécution de 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 la quelles un effacement général de la CPU peut être effectué.

➤ Signalisation des états

Certaines états de l'automate sont signalés par des LEDs sur la face avant de la CPU tel que :

- ✓ **SF** : signalisation groupée des défauts, défauts 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.
- ✓ **FCRE** : forçage signalisation qu'au moins une entrée ou une sortie est forcée de manière permanente.
- ✓ **RUN** : clignotement de la mise en route de la CPU, allumage continue en mode RUN.
- ✓ **STOP** : allumage continue au mode STOP, clignotement rapide lorsqu'un effacement général est en cours.

• La carte mémoire

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

• La pile

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

• Borne pour l'alimentation et la terre fonctionnelle

Ce bloc est commun à la majorité des CPU des S7-300, on trouve les différentes bornes d'alimentation tel que :

- **Cavalier amovible pour montage sans liaison à la terre.**

Remarque

Pour l'alimentation en CC de 24 V, il est recommandé d'utiliser le module d'alimentation PS 307.

- **Modules de coupleur (IM)**

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

Les coupleurs IM 306/IM 361 ou IM365 permettent de réaliser des configurations à plusieurs châssis.

- **Modules entrées/sorties tout ou rien (ETOR/STOR)**

Ce sont des circuits chargés d'adapter en tension et en courant les signaux entre l'unité centrale et les entrées-sorties. Ils assurent en outre un isolement entre les entrées-sorties et l'unité centrale. L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses. L'interface de sortie comporte de la même façon des adresses de sortie. Chaque pré actionneur est relié à une de ces adresses.

- **Modules d'entrées/sorties analogiques (EANA/SANA)**

- **Entrées analogiques**

L'information de certains capteurs peut être analogique. Elle est transmise à l'automate via un convertisseur analogique-numérique.

- **Sorties analogiques**

Les ordres et les signalisations de l'automate vers la partie opérative sont toujours numériques. Pour leurs exploitation un convertisseur numérique-analogique est nécessaire.

- **Modules de fonctions (FM)**

Il a pour rôle l'exécution de tâches de traitement des signaux du processus à temps critique et nécessitant une importante capacité mémoire comme la carte de comptage rapide. Elle permet d'acquérir des informations de fréquences élevées incompatibles avec le temps de traitement de l'automate. Un module de comptage comporte un compteur/décompteur dont le contenu est incrémenté ou décrémenté par des impulsions en provenance de l'extérieur.

- **Modules de communication (CP)**

Ils permettent d'établir des liaisons hommes-machines, machines-machines via des interfaces de communication :

- Profibus.
- Industriel Ethernet. [6]

3.3.5. Fonctionnement de l'automate programmable

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

- **Réception des informations sur les états du système**

Le S7-300 reçoit des informations sur l'état du processus via les capteurs de signaux reliés aux entrées, et il va mettre à jour la mémoire image des entrées au début de chaque cycle de programme, en transférant le nouvel état des signaux d'entrée des modules vers la mémoire image des entrées ce qui permet à la CPU de connaître l'état du processus.

- **Système d'exploitation**

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

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

- **Exécution du programme utilisateur**

Après avoir acquis les informations d'entrée, exécuter le système d'exploitation, la CPU passe à l'exécution de programme utilisateur, qui contient la liste d'instructions à exécuter pour faire fonctionner le processus. Il est composé essentiellement de blocs de

données de code et de blocs d'organisation.

- **Commande de processus**

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

3.3.6 Nature des informations traitées par l'automate

Les informations peuvent être du type :

- **Tout ou rien (T.O.R)**

L'information ne peut prendre que deux états (vrai/faux, 0 ou 1 ...).

C'est le type d'information délivrée par un détecteur, un bouton poussoir...

- **Analogique**

L'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température...)

- **Numérique**

L'information est continue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

3.3.7. Ces avantages

Les S7-300 offrent de nombreux avantages :

- Une construction compacte et modulaire, libre de contraintes de configuration.
- Une riche gamme de modules adaptés à tous les besoins du marché et utilisables en architecture centralisée ou décentralisée, qui réduit grandement le stock de pièces de rechange.
- Une large gamme de CPU adaptées à toutes les demandes de performance pour pouvoir obtenir des temps de cycle machines courts, certaines étant dotées de fonctions technologiques intégrées comme par ex. le comptage, la régulation ou le positionnement. [7]

3.3.8. passage de S5 à S7

Le passage de S5 vers S7 nécessite la connaissance de la différence existante entre eux.

Toutes les nouveautés apportées sur la gamme des SIMATIC S7, ont permis de compléter la gamme SIMATIC S5. On trouve les types de modules suivants :

- Unités centrales (CPU), Modules d'alimentation (PS), Coupleurs (IM).
- Processeurs de communication(CP) ;(par exemple pour le raccordement à PROFIBUS)
- Modules de fonction FM ; (par exemple de comptage, régulation, positionnement etc.).
- Les modules d'entrées/sorties TOR ou analogiques sont maintenant appelés des modules de signaux (SM).
- Les modules de S7 se signalent par de nouvelles performances techniques. Il n'y a plus de cavaliers ni de commutateurs sur les modules. Les plages d'entrées des modules d'entrées analogiques se laissent facilement paramétrer dans STEP 7.
- Les appareils d'extension et la périphérie décentralisée ET 200 peuvent déclencher des alarmes.

On a par exemple la possibilité pour les modules d'entrées TOR paramétrables d'indiquer dans l'application de configuration du matériel de STEP 7 quelles voies devront déclencher une alarme de processus lors d'un changement de front. [8]

3.3.9.programmation de l'automate S7-300

A chaque famille d'automates correspond un langage spécifique de programmation. Les automates de la famille SIEMENS sont programmés grâce au le logiciel STEP 7 via une console de programmation ou PC et sous un environnement WINDOWS (Figure 3.8).

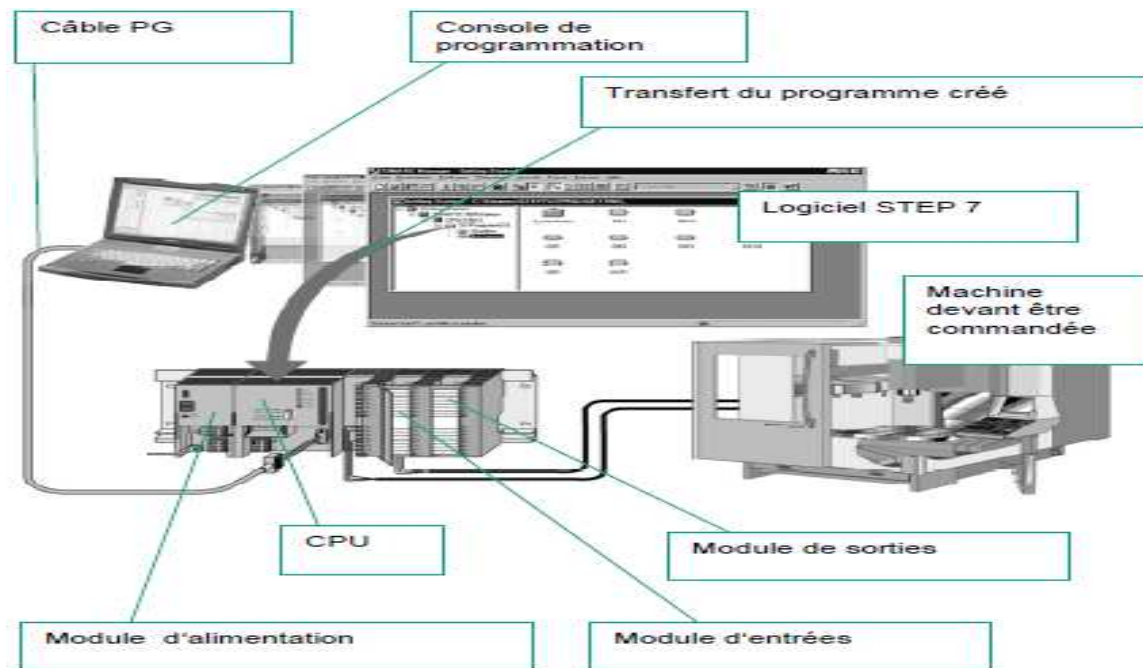


Figure 3.11: Vue d'ensemble des éléments entrant dans l'automatisation d'un système

Le logiciel STEP7 offre les possibilités suivantes :

- configuration et paramétrage du matériel et de la communication.
- création et gestion des projets
- gestion des mnémoniques
- test de l'installation d'automatisation. [8]

3.3.10. Programmation en STEP7

La programmation en STEP7 a trois modes de représentation :

- **Mode liste (LIST)**

Langage textuel proche du comportement interne de l'automate.

- **Mode logigramme (LOG)**

Langage graphique, utilisant les symboles de l'électronique numérique (portes logiques)

- **Mode contact (CONT)**

Suite de réseaux parcourus séquentiellement. Les entrées sont représentées par des interrupteurs -| |- (ou -|/- si entrée inversée), les sorties par des bobines -()-.

D'autres langages (optionnels) existent, les trois qui semblent les plus intéressants sont SCL (langage proche du Pascal, permettant des algorithmes et calculs complexes), GRAPH (proche du GRAFCET), Hi GRAPH (proche des réseaux de Pétri). [8]

3.3.11. Blocs du programme utilisateur

Il faut avoir l'habitude de subdiviser le procédé à automatiser en différentes tâches. Les parties d'un programme utilisateur structuré correspondant à ces différentes tâches, sont les blocs de programmes.

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

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

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

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

Un OB est appelé cycliquement par le système d'exploitation et constitue donc une interface entre le programme utilisateur et le système d'exploitation. L'OB contient des instructions d'appel de blocs indiquant à l'unité de commande de l'automate l'ordre dans lequel il doit traiter les blocs.

- **Bloc fonctionnel (FB)**

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

- **Fonction (FC)**

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

peuvent faire appel à des blocs de données globaux pour la sauvegarde de données. Une fonction contient un programme qui est exécuté lorsqu'elle est appelée par un autre bloc de code. Elle peut-être utilisée pour :

- Renvoyer une valeur de fonction au bloc appelant (exemple : fonction mathématique).
- Exécuter une fonction technologique.
- **Bloc de données (DB)**

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

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

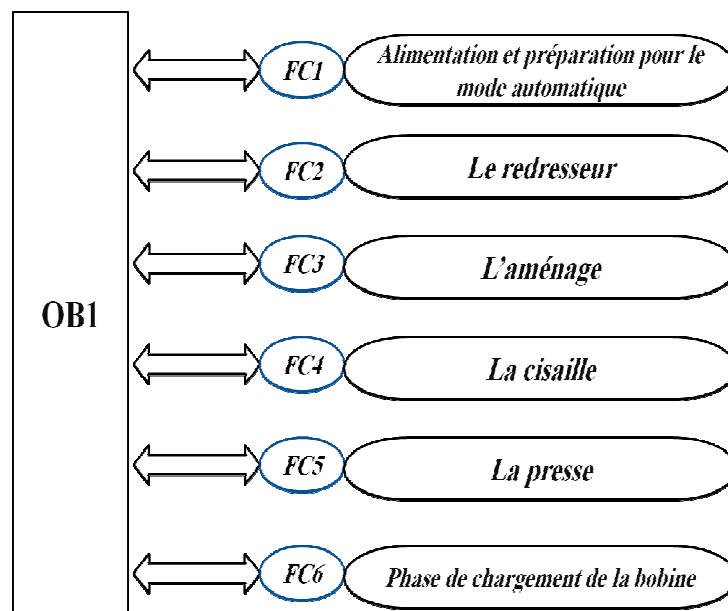


Figure 3.12 : structure du programme

3.3.12. Création d'un projet dans S7-300

Pour créer un projet STEP7, on dispose d'une certaine liberté d'action, en effet on a deux solutions possibles soit :

- 1- Commencer par la configuration matérielle
- 2- Commencer par écrire le programme.

Dans notre cas les procédures suivies pour la création du projet sous le logiciel STEP7, sont comme suit : [13]



1-Lancer SIMATIC manager par un double clique sur son icône.

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



Figure 3.13:fenêtre de création d'un objet

3-La fenêtre suivante permet la création d'un projet



Figure 3.14 : CPU 314 sélectionné

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

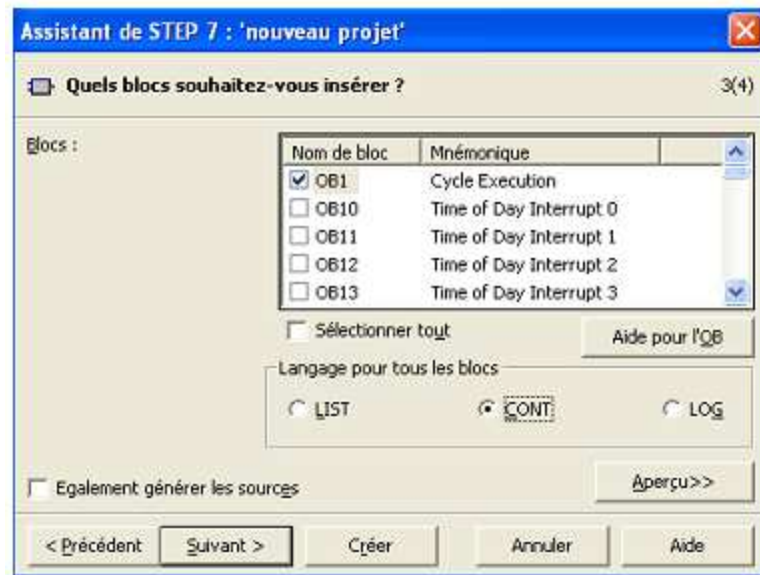


Figure 3.15: Sélection des blocs et le langage de programmation (CONT)

5- Une fois le projet crée, il est nécessaire de configurer le matériel à utiliser comme le montre la (figure 3.16).

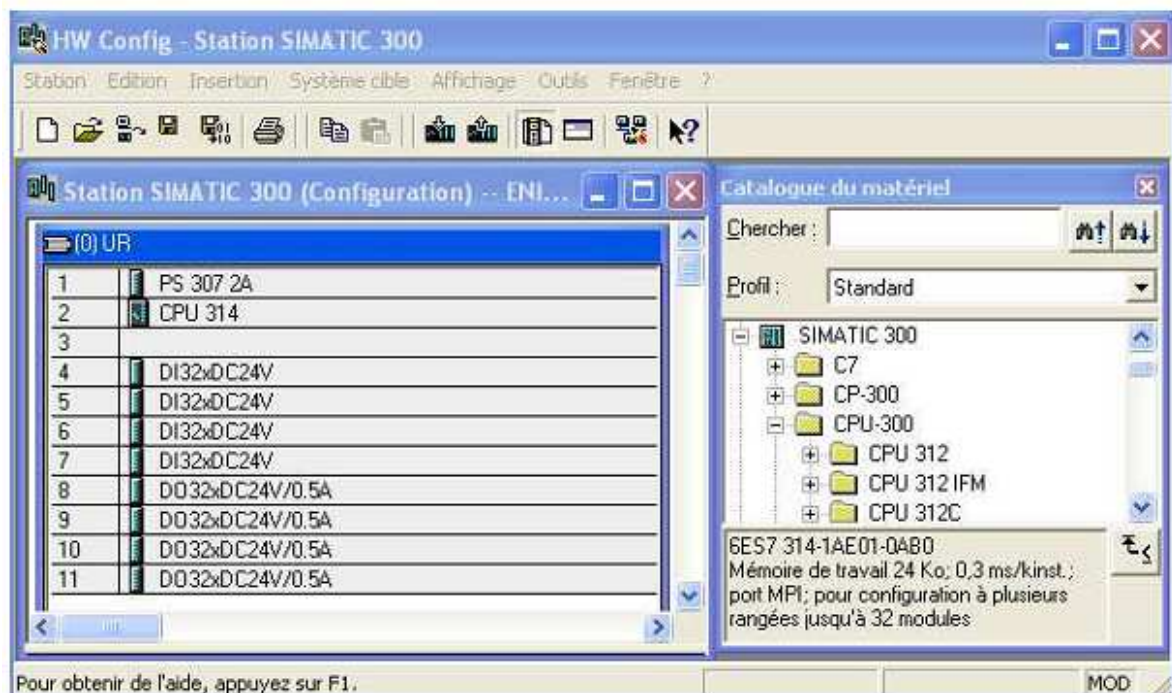


Figure 3.16 : Configuration matériels

6- Ensuite on passe au programme utilisateur que nous avons écrit pour commander la machine, ce dernier est composé d'objets définis dans l'environnement de STEP7. (Figure 3.17).

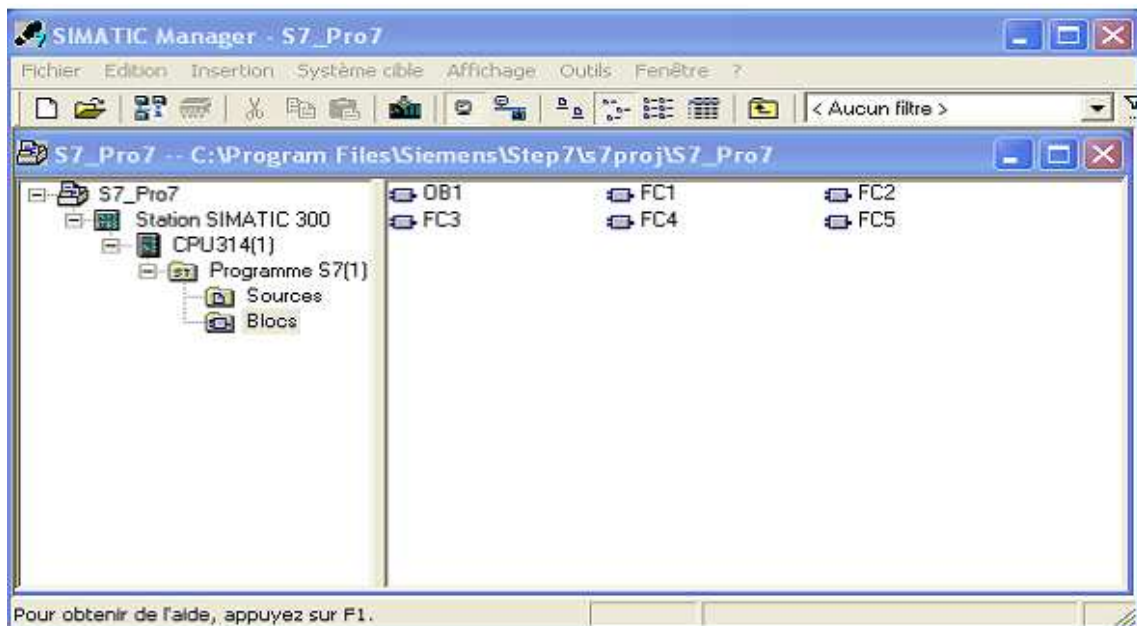


Figure 3.17 : Vue des composants d'un projet S7

3.4. Simulation et la validation du programme

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 STEP 7.

3.4.1. 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.

La procédure à suivre est :

- Ouvrir le gestionnaire de projet SIMATIC.
- sélectionnez la commande outils > simulation de modules.

Cela lance l'application S7-PLCSIM et ouvre une fenêtre CPU (Figure. 3.18)

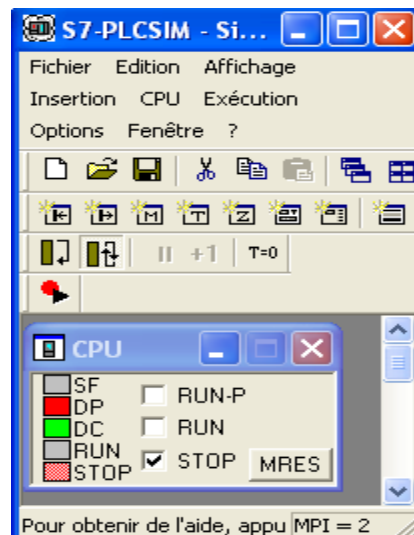


Figure 3.18 : Fenêtre du S7-PLCSIM

- Dans le gestionnaire de projet SIMATIC, chercher le projet-exemple projet presse transfert.
- Dans le projet exemple «projet presse », chercher le dossier blocs.
- Dans le gestionnaire de projet SIMATIC, choisir la commande Système cible >charger pour charger le dossier blocs dans l'API de simulation.

Dans l'application S7-PLCSIM, crier de nouvelles fenêtres pour visualiser les informations provenant de l'API de simulation.

- Choisir le menu CPU dans la fenêtre du S7-PLCSIM et vérifier que la commande Mettre sous tension est activée (Figure 3.19).

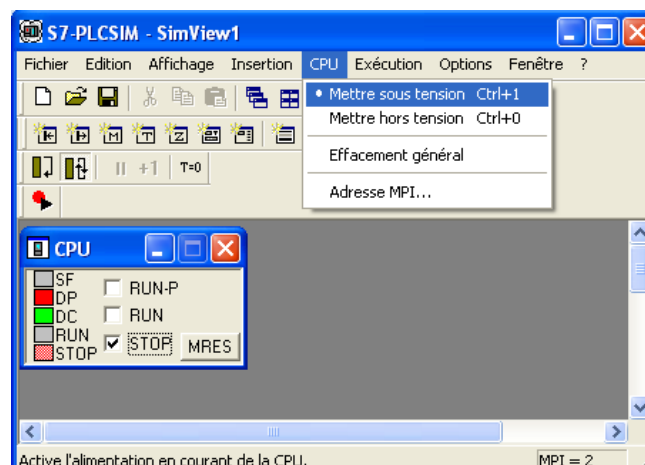


Figure 3.19 : Mise sous tension de la CPU

- Mettre la CPU de simulation en marche en cliquant sur l'une des cases à cocher RUN ou RUN-P.

Une fois toutes les fenêtres d'entrées et de sorties sont prêtes, nous activons les entrées voulues pour lire l'état des sorties.

3.4.2. Simulation du programme de la presse transfert

La simulation du programme de la presse transfert dans les blocs fonctions FC s'effectue Comme suit :

- Charger les blocs fonctions dans la PG.
- Activer la fonction de visualisation.
- Forcer les entrées nécessaires pour chaque réseau.

Les résultats de la simulation sont donnés comme dans l'exemple du réseau de la (figure 3.20)

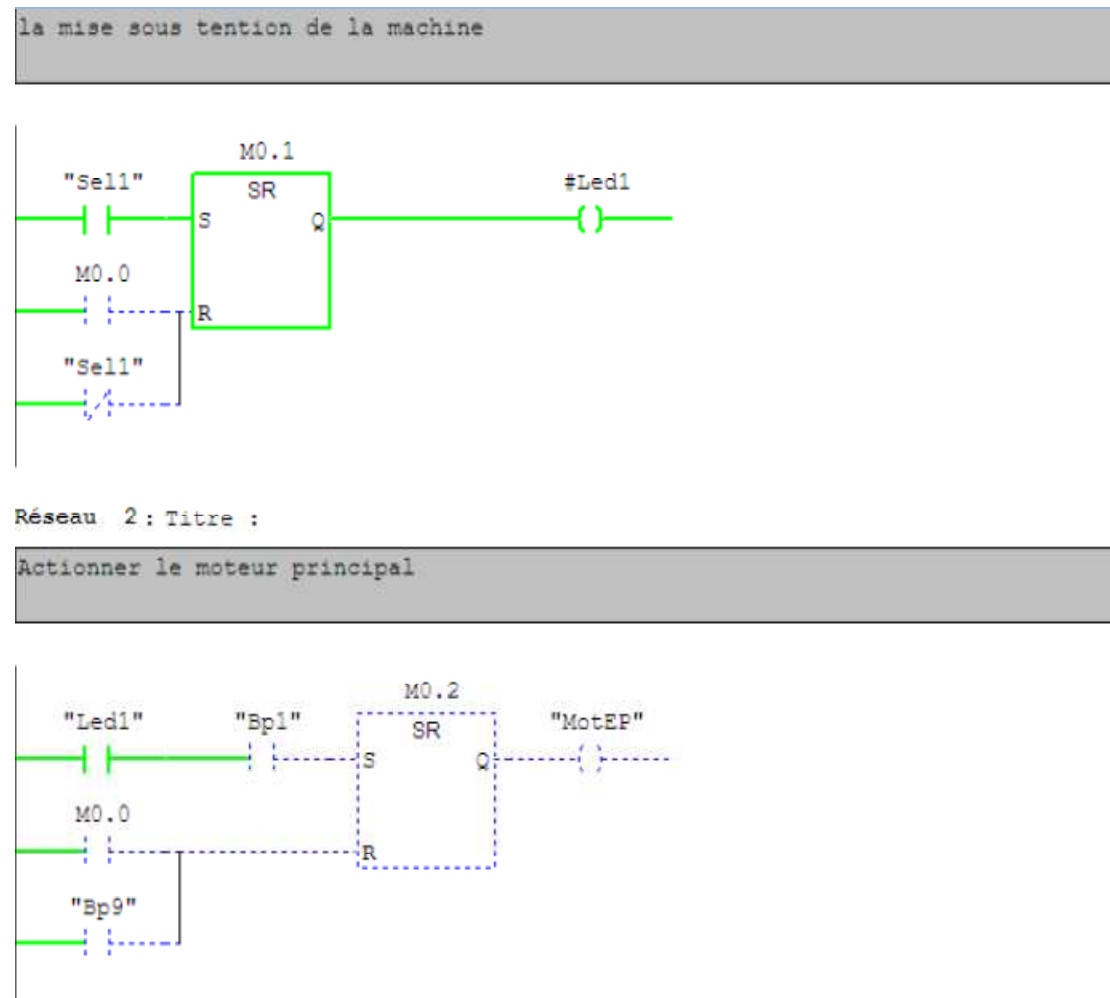


Figure 3.20 : résultat de la simulation du programme de la presse

3.5. Conclusion

Dans ce 3^{eme} chapitre nous avons tout d'abord décrit les automates programmables industrielles(API) d'une façon générale.

Par la suite, nous passons à la présentation de l'automate « S7-300 » qui a été choisi comme solution adéquat, facile à adapter aux diverses conditions industrielles ; son avantage est de passer facilement de S5vers S7.

L'utilisation de la programmation structurée est indisponible vue le degré de difficulté du fonctionnement de notre processus ; une fois tous les FC sont programmés, nous allons les insérer dans le bloc OB1 pour la phase simulation.

Chapitre 4

Plate forme de supervision sous WinCC Flexible 2008

4.1. Introduction

Suite a l'automatisation industrielle, l'opérateur humain a été contraint de conduire ou de superviser des machines automatisées, en réduisant les prises d'information et les actions directes sur le processus, ce qui conduit à l'élaboration d'interface d'interaction Homme/Machine, flexible et aussi lisible pour un simple operateur.

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

L'automate S7-300 de la firme SIEMENS (Il est disponible sur le marché, doté d'une technologie moderne, programmation simple, documentation disponible) est conçu pour des solutions dédiées aux systèmes manufacturiers. Il constitue à ce titre une plate-forme d'automatisation universelle pour les applications avec des architectures centralisées et décentralisées.

4.2. Définition de la supervision industrielle

Dans l'industrie, la supervision est une technique de suivi et de pilotage informatique de procédés de fabrication automatisés pour les amener à leur point de fonctionnement optimal

La supervision d'un système inclut des fonctions de collecte et de visualisation d'informations.

Le but c'est de disposer en temps réel d'une visualisation de l'état d'évolution des paramètres du processus, ce qui permet à l'opérateur de prendre rapidement des décisions appropriées à ses objectifs.

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

- détection des défauts ;
- Le diagnostic et le traitement des alarmes ;
- surveillance du Process à distance.

4.3. Constitution d'un système de supervision

La plus grande partie des systèmes de supervision se compose généralement d'un moteur central (logiciel) auquel se rattachent des données provenant des équipements (automate). Ce moteur central assure l'affichage, le traitement et l'archivage des données, ainsi que la communication avec d'autres périphériques (Figure 4.1). [14]

-Module de visualisation

Il permet à l'opérateur la visualisation du procédé au cours de son déroulement.

-Module d'archivage

Il mémorise les données pendant le déroulement du cycle, pour une longue période

-Module de traitement

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

-Module de communication

Assure l'acquisition et le transfert de données et gère la communication avec les APIs et d'autres périphériques.

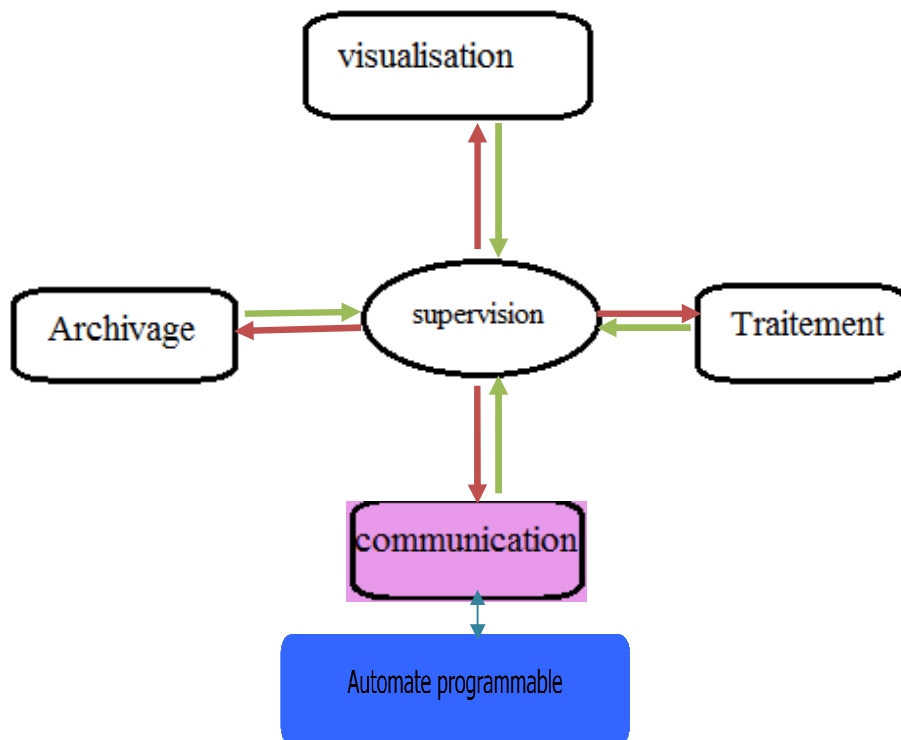


Figure 4.1 : Constitution d'un système de supervision

4.4 .Présentation du logiciel WinCC flexible**4. 4.1. Définition logiciel de supervision Win CC**

SIMATIC WinCC (Windows Control Center) est un système de supervision de processus modulaire qui offre des fonctions performantes de surveillance d'automatismes, c'est l'interface homme-machine (IHM) développé par SIEMENS. Il constitue la solution de conduite et de supervision de procédés sur ordinateur, pour des systèmes monoposte et multipostes. Il offre une bonne solution de supervision car il met à la disposition de l'opérateur des fonctionnalités adaptées aux exigences d'une installation industrielle.

WinCC permet une supervision, par un ou plusieurs PC indépendants (nommés OS, station opérateur) d'un ou plusieurs automates (AS). Il y a deux approches possibles :

- o Sur le PC, on appelle WinCC, dans lequel on définit les vues (le graphisme) que l'on prévoit,

ainsi que les variables (entrées/sorties/mémos et DB) des différents automates qu'on va superviser.

Les automates, quand à eux, sont programmés indépendamment (en Step7 par exemple), il faut simplement que le programme automate prévoie de mettre à jour les variables utilisées par la supervision.

- o Sous STEP7, on crée un projet regroupant l'ensemble : programmes des automates et supervision. Cette deuxième approche nous semble préférable, car on gère le problème globalement.

Rien que dans le cas de mises à jour des programmes, on évitera le risque d'avoir des versions différentes sur différents matériels.

Choisissons donc le second cas. On crée sous STEP7 un projet. On y insère d'abord les différents matériels (AS : automates et leurs E/S, liaisons réseau). Dans chaque AS on définit les programmes et les mnémoniques. Dans ces mnémoniques, il faut en particulier définir les variables qui seront supervisées (sélection de la ou les lignes de mnémoniques concernés, clic droit -> propriétés spécifiques -> contrôle-commande puis cocher la case contrôle-commande).

faire un copier-coller (ou glissement de souris) pour récupérer des données dans un projet antérieur.

L'environnement de travail de Win CC flexible se compose de plusieurs éléments ; certains de ces éléments sont liés à des éditeurs particuliers, visibles uniquement lorsque cet éditeur est active. Il met à disposition un éditeur spécifique pour chaque tâche de configuration.

On peut configurer par exemple l'interface utilisateur graphique d'un pupitre operateur avec l'éditeur "Vues". Pour la configuration d'alarmes, on peut utiliser par exemple l'éditeur "Alarmes TOR" ou autre.

Dans Win CC flexible, chaque projet crée contient principalement des vues que l'on crée

pour le contrôle-commande de machines et d'installations. Lors de la création des vues, vous disposez d'objets prédéfinis permettant d'afficher des procédures et de définir des valeurs du processus.

Les différents outils et barres de l'éditeur de vues sont représentés dans la figure qui suit :

- Barre des menus

La barre des menus contient toutes les commandes nécessaires à l'utilisation de Win CC Flexible. Les raccourcis disponibles sont indiqués en regard de la commande du menu.

- Barres d'outils

La barre d'outils permet d'afficher tout dont le programmeur a besoin.

- Zone de travail

La zone de travail sert à configurer des vues, de façon qu'il soit le plus compréhensible par l'utilisateur, et très facile à manipuler et consulter les résultats.

- Boîte à outils

La fenêtre des outils propose un choix d'objets simples ou complexes qu'on insère dans les vues, par exemple des objets graphiques et éléments de commande.

- Fenêtre des propriétés

Le contenu de la fenêtre des propriétés dépend de la sélection actuelle dans la zone de travail, lorsqu'un objet est sélectionné, on peut éditer les propriétés de l'objet en question dans la fenêtre des propriétés.[14]

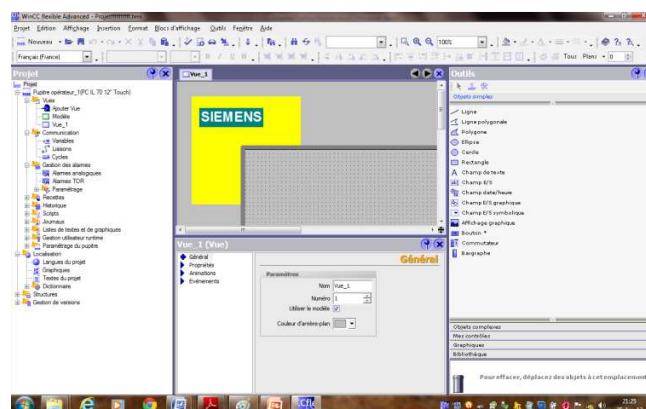


Figure 4.2 : Vue d'ensemble du progiciel Win CC flexible

4.4.2. Application disponibles sous Win CC

Win CC se compose de plusieurs applications pour accomplir la fonction de supervision. Il dispose des modules suivants :

- Graphic designer

Il offre la possibilité de créer des vues de procédés, et de les configurer en leur affectant les variables correspondantes. À cet effet, il dispose d'une bibliothèque d'objet et permet de créer des objets selon le besoin. Il assure la fonction de visualisation grâce au graphics runtime.

-Tag logging

On y définit les archives, les valeurs de process à archiver et les temps de cycle de saisie et d'archivage. En outre on y configure la mémoire tampon sur le disque dur.

-Alarm logging

Il se charge de l'acquisition, traitement, visualisation, l'archivage des alarmes et mettre à la disposition des utilisateurs les fonctions nécessaires à la reprise des alarmes issues du process.

-Global script runtime

Il dispose de deux éditeurs : l'éditeur C et l'éditeur Visual Basic Script (VBS), à l'aide desquels on crée des actions et des fonctions qui ne sont pas prévues dans le Win CC.

-Report designer

Il a des fonctions avec lesquelles on peut lancer la prévisualisation d'une impression. On y trouve aussi des modèles de mise en page de journal qu'on peut adapter en fonction du besoin.

-User administrator

C'est là que s'effectue la gestion des utilisateurs et des autorisations. On y crée des nouveaux utilisateurs, on leur attribue les mots de passes et on leur affecte la liste des autorisations.

En plus des modules suscités, Win CC dispose d'une série d'applications, comme le stock de variables et le gestionnaire d'accès à un réseau internet...etc.

4.4.3. Création d'un projet sous Win CC

Pour créer un projet dans Win CC, procédez de la manière suivante:

Les principales étapes suivies pour créer notre application sous WINCC sont:

- Créer un projet.
- Sélectionner et installer l'API.
- Définir les variables dans l'éditeur de variables.
- Créer et éditer les vues dans l'éditeur Graphic designer.
- Paramétrer les propriétés de WINCC runtime.
- Activer les vues dans le WINCC runtime.
- Utiliser le simulateur pour tester les vues du process.

Lorsque Win CC démarre pour la première fois, une fenêtre de dialogue s'affiche :

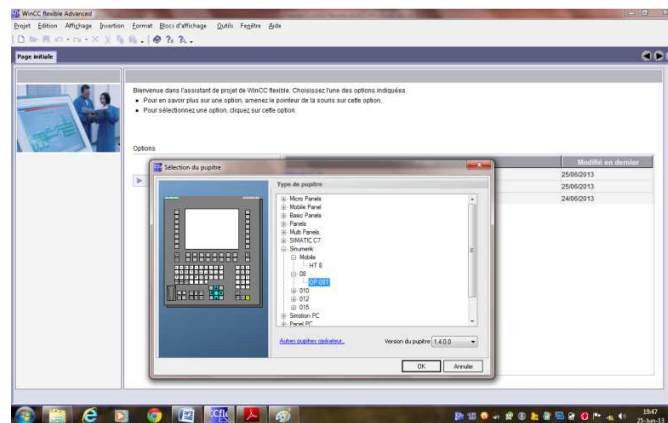


Figure 4.3 : Fenêtre de dialogue

4.4.4. Intégration dans SIMATIC STEP 7

Les variables du processus représentent la liaison pour la communication entre le système d'automatisation et le système IHM. Sans les avantages de la TIA (Totally Integrated Automation), on devra définir chaque variable à deux reprises : une fois pour le système d'automatisation et une fois pour le système IHM.

L'intégration de SIMATIC STEP 7 dans l'interface de configuration permet de diminuer la fréquence des erreurs et de réduire les tâches de configuration nécessaires. Durant la configuration, nous accédons directement à la table des mnémoniques de STEP 7 ainsi qu'aux paramètres de communication.

La table des mnémoniques de STEP 7 contient la définition des points de données (par exemple adresses ou types de données) qu'on a paramétrées lors de la création du programme de commande.

Les paramètres de communication contiennent les adresses de bus ainsi que les protocoles de commande. On définit les paramètres de communication avec NetPro, par exemple.

La figure suivante montre la liaison entre la station S7-300 et la station de supervision HMI.

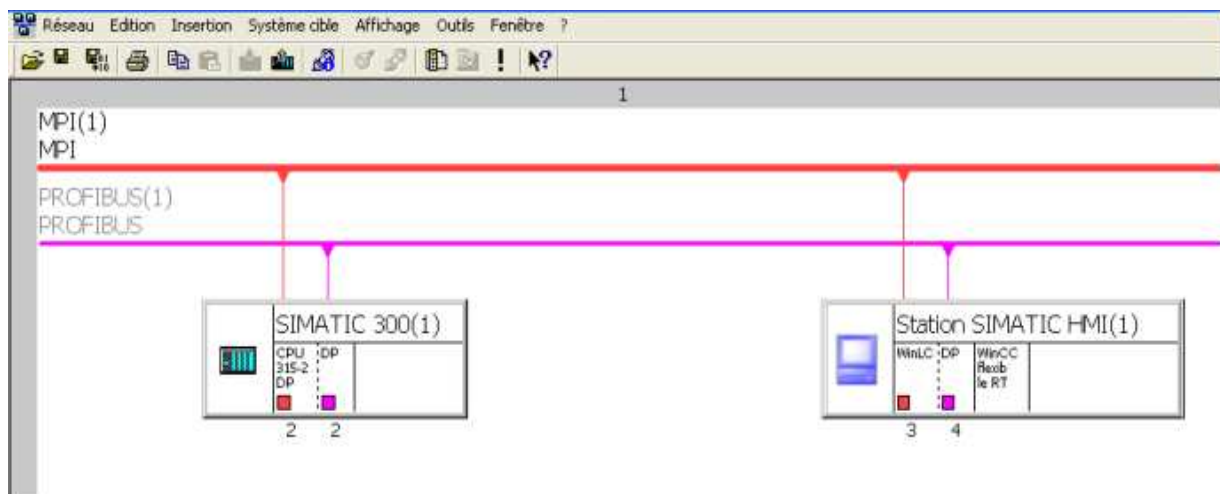


Figure 4.4 : Liaison entre la station S7-300 et la station de supervision HMI

4.5. Plateforme de supervision de la station (ENIEM)

Pour élaborer la plateforme de supervision qui permet le control commande de notre station, nous avons créé 5 vues données comme suit :

4.5.1. Vue d'accueil

La vue d'accueil est la vue d'entrée qui sera tout le temps visible sur le pupitre qui sera placé sur notre station. Elle présente essentiellement le cicle de l'unité (ENIEM) .Cette vue contient un bouton poussoir qui permet d'aller à la première vue , Ce bouton poussoir peut être verrouillé par un mot de passe interdisant ainsi l'accès à toute personne étrangère. et contient aussi les boutons de navigation à partir desquels on peut sélectionner la vue à visualiser.

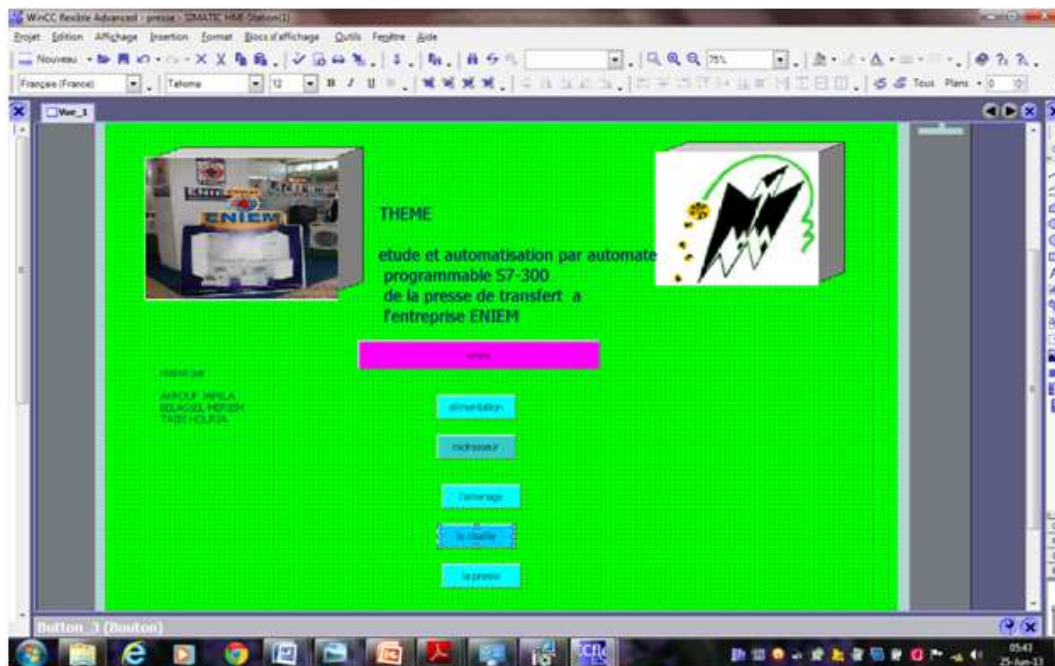


Figure 4.5 : Vue d'accueil

4.5.2. Vue pour redresseur

À partir de cette vue, nous pouvons démarrer la mis en marche de redresseur.

La vitesse du moteur, faisant tourner les rouleaux du redresseur, est conditionnée par

Les quatre capteurs photoélectriques de la fosse :

- o Le premier capteur (au seuil), indique que le moteur du redresseur doit tourner à une grande vitesse
- o Le deuxième capteur correspond à une vitesse moyenne.
- o Le troisième capteur, à une vitesse faible.
- o Le quatrième marque l'arrêt total du moteur.

Un bouton de commande pour l'arrêt d'urgence permettant de s'arrêter en cas de dysfonctionnement, et un bouton pour basculer vers la vue d'accueil

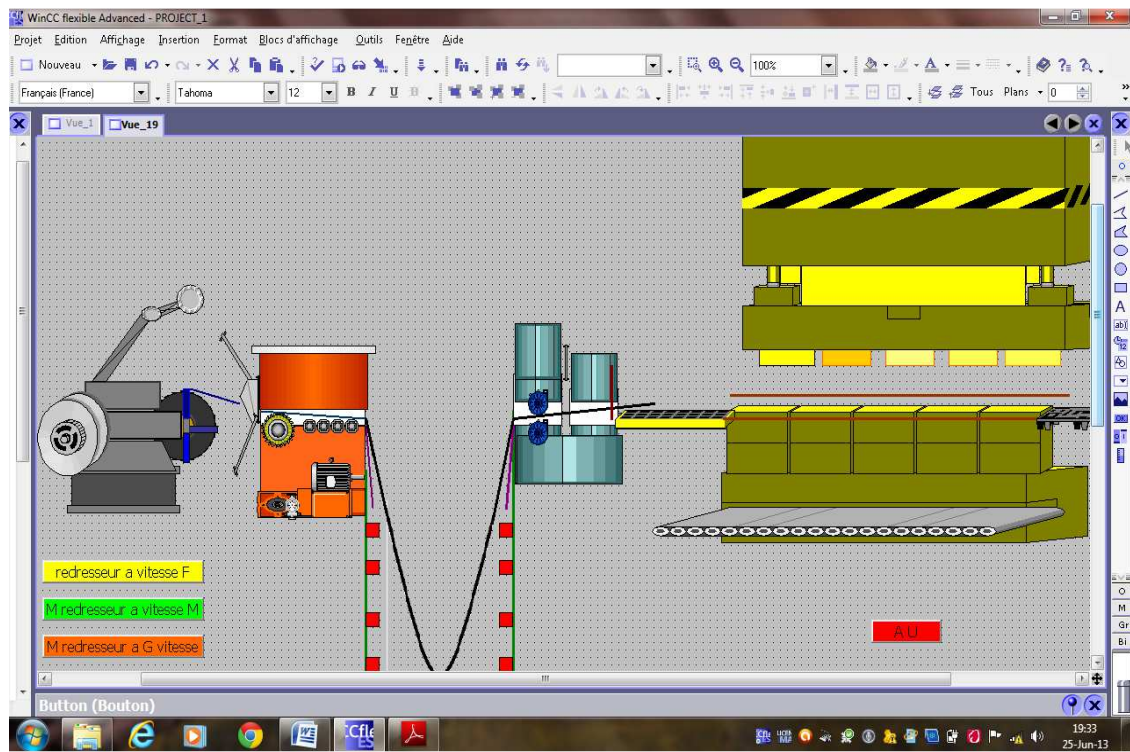


Figure 4.6 : Vue pour redresseur

4.5.3. Vue aménage

A partir de cette vue nous pouvons :

- Démarrer la mise en marche de moteur d'aménage.
- Commander le frein du moteur d'aménage.

Un bouton de commande pour l'arrêt d'urgence permettant de s'arrêter en cas de dysfonctionnement, et un bouton pour basculer vers la vue d'accueil.

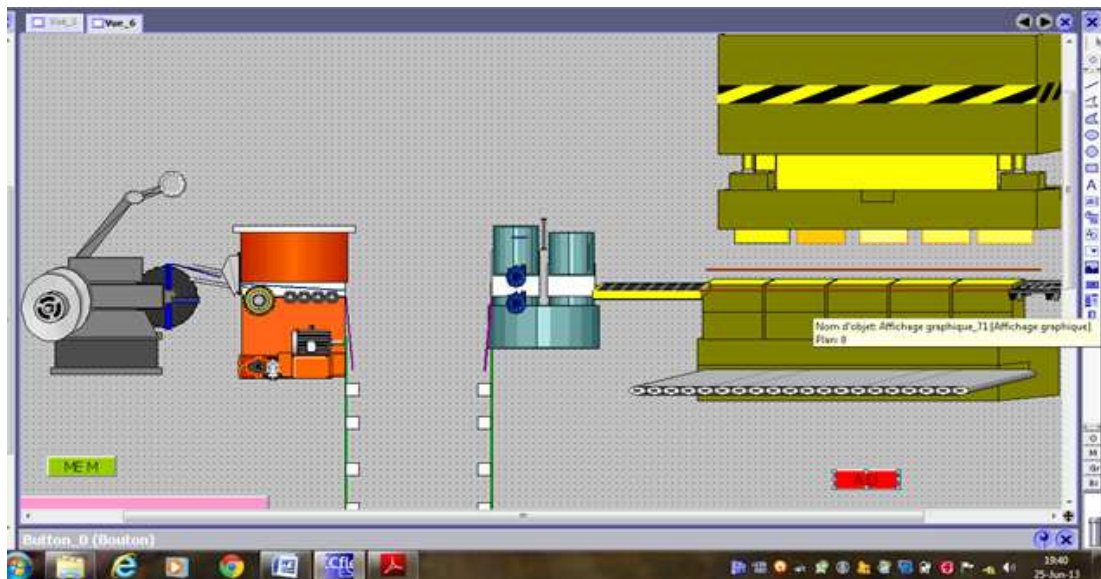


Figure 4.7 : Vue aménage

4.5.4. Vue pour le fonctionnement de la cisaille

La cisaille coupe la tôle dès que la longueur de la tôle est égale au pas réglé à l'encodeur.

A partir de cette vue (Figure 4.8) nous pouvons commander :

- La sortie de vérin cisaille
- L'entrée de vérin cisaille
- Le soulèvement de la pièce
- Fermeture et soulèvements des barres

Un bouton de commande pour l'arrêt d'urgence permettant de s'arrêter en cas de dysfonctionnement, et un bouton pour basculer vers la vue d'accueil.

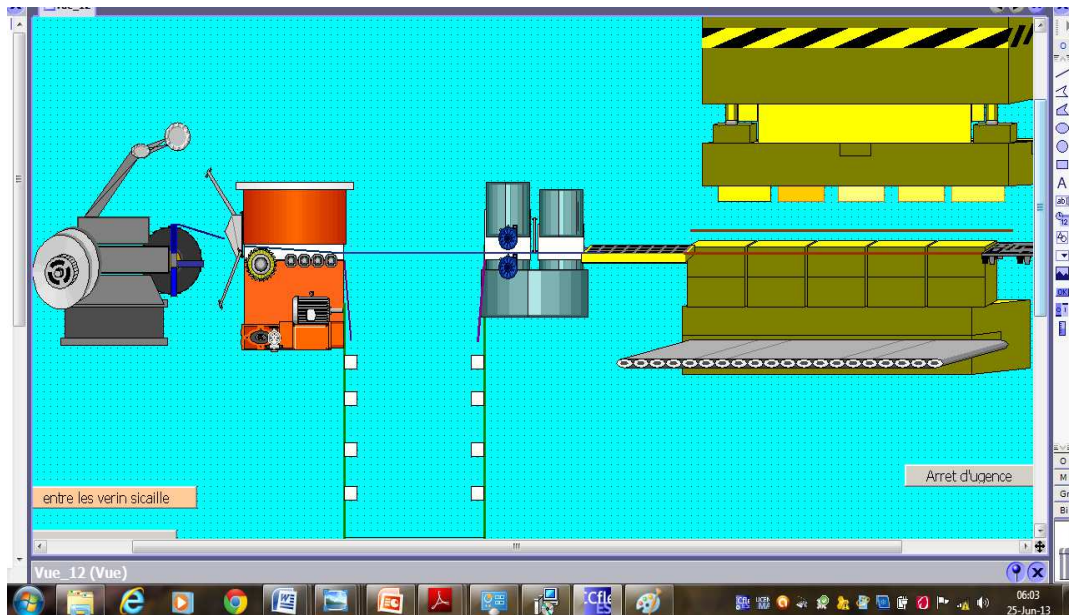


Figure 4.8 : vue pour le fonctionnement de la cisaille

4.5.5. Vue pour chargements de la presse

A partir de cette vue (Figure 4.9) en plus de la visualisation de l'évolution de la station nous pouvons passer a l'étape de l'alimentation des quatre postes de la presse qui s'effectue comme suit:

- Sélectionner le mode automatique sur le pupitre presse (pupitre principal)
- Sélectionner le mode manuel sur le pupitre suspendu.

Le déplacement de la pièce de la table d'aménage vers les différentes table porte outils de la presse s'effectue a l'aide des barres porte pincettes et leurs mouvement.

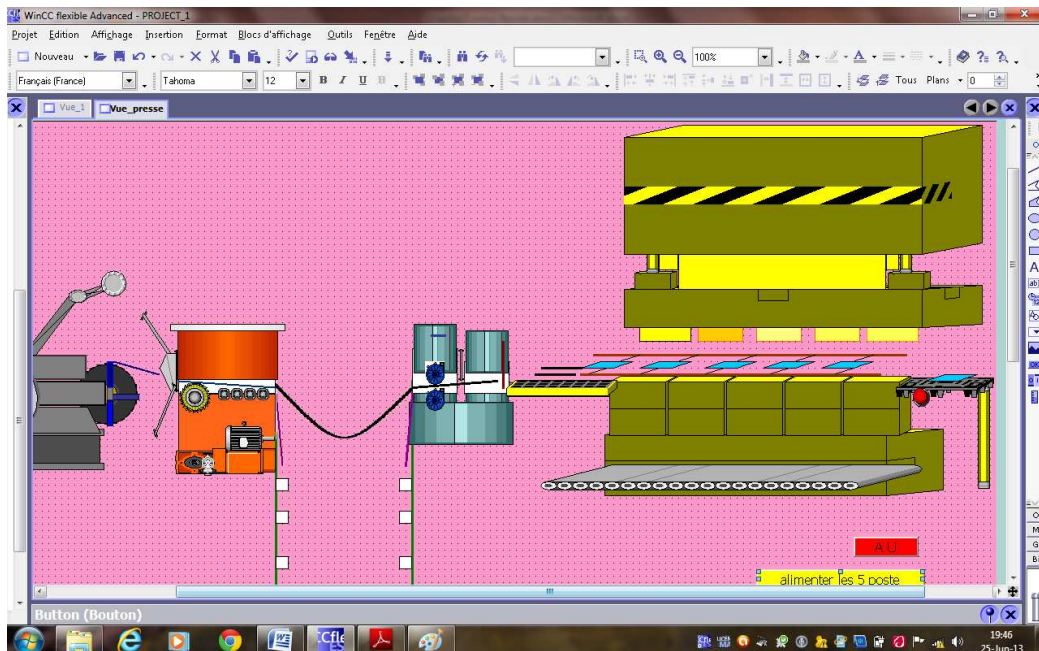


Figure 4.9 : Vue pour chargements de la presse

4.6. Conclusion

Dans la dernière étape de notre travail, nous avons supervisé notre station en développant des vues qui décrivent son fonctionnement sous WinCC flexible laquelle permettra de gérer toutes les opérations assignées à la station.

Conclusion Générale

Notre projet de fin d'étude à l'unité cuisson de l'entreprise (ENIEM) ; situé à la zone industrielle (OUED AISSI) nous a apporté en terme de connaissances pratiques reçus sur le terrain, ainsi une transition facile de la formation au domaine professionnel et de découvrir le monde du travail.

Le but de notre projet est de concevoir une solution programmable plus performante pour la presse qui servira à remplacer le séquenceur S5 par un automate programmable S7-300.

Après l'étude de la machine « presse de transfert », nous avons modélisé par un outil très puissant GRAFCET, une solution de commande automatisée par l'API S7-300 à l'aide de son langage de programmation STEP7 dont les avantages sont les suivant:

- Augmentation de la productivité.
- Bonne qualité de produits.
- Sécurité des opérateurs et des matériels.
- Dans la dernière étape de notre travail, nous avons supervisé notre machine en développant des vues qui décrivent son fonctionnement.

Nous espérons que notre travail sera consacré sur le plan pratique, que nos efforts puissent servir à quelque chose et ce mémoire sera utile pour les promotions à venir.

En fin, la solution proposé dans ce mémoire ne pourra être réellement validé qu'une fois implanté sur un automate, mais les résultats obtenus nous rendent optimiste à ce sujet.

Propriétés de la table des mnémoniques

Nom : Mnémoniques

Auteur :

Commentaire :

Date de création : 11/06/2013 17:01:26

Dernière modification : 21/06/2013 09:55:14

Dernier filtre sélectionné : Tous les mnémoniques

Nombre de mnémoniques : 190/190

Dernier tri : Mnémonique ordre croissant

Etat	Mnémonique	Opérande	Type de données	Commentaire
	A Mot H1	A 13.3	BOOL	arrêt de mot H1
	A Mot H2	A 13.4	BOOL	
	A_ME2	A 14.4	BOOL	
	A_ME3	A 14.3	BOOL	arrêt moteur e 3
	APP1	A 13.1	BOOL	afficher panne poste 1
	APP2	A 13.5	BOOL	afficher panne poste 2
	APP3	A 13.6	BOOL	afficher panne poste 3
	APP4	A 13.7	BOOL	afficher panne poste 4
	APP5	A 14.0	BOOL	afficher panne poste 5
	Ar_ME5	A 14.1	BOOL	arrêt de moteur E5
	ARur	E 1.1	BOOL	Arrêt d'urgence doux
	ARut	E 1.0	BOOL	Arrêt d'urgence total
	Bp 3	E 0.2	BOOL	Mise en marche tapis déchets
	Bp1	E 0.0	BOOL	Moteur principal mise en marche
	BP11	E 1.2	BOOL	Ve en avant
	bp12	E 1.3	BOOL	Ve en arrière
	BP13	E 1.4	BOOL	Montée de la Ve
	Bp14	E 1.5	BOOL	Descente de la Ve
	BP15	E 1.6	BOOL	Ouvrir mâchoire
	Bp16	E 1.7	BOOL	Fermer mâchoire
	BP17	E 2.0	BOOL	Rotation bobine droite
	Bp18	E 2.1	BOOL	Rotation bobine gauche
	Bp19	E 2.2	BOOL	Blocage rotation
	BP2	E 0.1	BOOL	Alimenter les éjecteurs
	BP20	E 2.3	BOOL	Déblocage rotation
	Bp21	E 2.4	BOOL	Descendre le rouleau presseur
	Bp22	E 2.5	BOOL	Montée du rouleau presseur
	Bp23	E 2.6	BOOL	Bande en avant
	Bp24	E 2.7	BOOL	Bonde en arrière
	Bp25	E 3.0	BOOL	Plaque inférieure vers le haut
	Bp26	E 3.1	BOOL	Sortir la lame
	Bp27	E 3.2	BOOL	Rouleau postérieur vers le haut
	Bp28	E 3.3	BOOL	Actionner le rouleau presseur
	Bp29	E 3.4	BOOL	Descendre la plaque inférieure
	Bp30	E 3.5	BOOL	Rouleau postérieur vers le bas
	Bp31	E 3.6	BOOL	Entrée lame
	Bp32	E 3.7	BOOL	Plaque supérieur vers le haut
	Bp33	E 4.0	BOOL	Rouleau presseur vers le haut
	Bp34	E 4.1	BOOL	Plan basculant gauche vers le haut
	Bp35	E 4.2	BOOL	Plan basculant droite vers le haut
	Bp36	E 4.3	BOOL	Rouleau supérieur d'aménage vers le haut
	Bp37	E 4.4	BOOL	Roue encodeur vers le haut
	Bp38	E 4.5	BOOL	Actionner moteur redresseur (remises des alarmes)
	Bp39	E 4.6	BOOL	Rouleau supérieur aménage vers le bas
	Bp4	E 0.3	BOOL	Mise en marche table d'aménage
	Bp40	E 4.7	BOOL	Roue encodeur vers le bas

Etat	Mnémonique	Opérande	Type de données	Commentaire
	Bp41	E 5.0	BOOL	Cisaille ver le bas
	Bp42	E 5.1	BOOL	Bouton " marche_continue " sur le pupitre suspendu
	Bp43	E 5.2	BOOL	Presse manuel
	BP44	E 5.3	BOOL	Plaque supérieur vers le bas
	Bp45	E 5.4	BOOL	Alimentation centrale hydraulique principale
	Bp46	E 5.5	BOOL	Alimentation de la centrale hydraulique d'équilibrage des barres
	Bp47	E 5.6	BOOL	Bouton d'alimentation circuit de graissage machine.
	Bp5	E 12.1	BOOL	Moteur aménage mise en marche
	Bp6	E 0.4	BOOL	Mise en marche cisaille
	Bp7	E 0.5	BOOL	Commande redresseur mise marche
	Bp8	E 0.6	BOOL	Redresseur mise en marche
	Bp9	E 0.7	BOOL	Arrêt Moteur principal
	C EM 1	E 9.4	BOOL	Capteur électromagnétique cisaille en haut
	C FC1	E 7.1	BOOL	Capteur de fin de course mécanique chargeur.
	C FC10	E 8.2	BOOL	Capteur de fin de course mécanique plan basculant droit
	C FC11	E 8.3	BOOL	Capteur de fin de course mécanique redresseur
	C FC2	E 7.2	BOOL	Capteur de fin de course mécanique rotation du dérouleur gauche
	C FC3	E 7.3	BOOL	Capteur de fin de course mécanique rotation du dérouleur droit
	C FC4	E 7.4	BOOL	Capteur de fin de course mécanique blocage rotation
	C FC5	E 7.5	BOOL	Capteur de fin de course mécanique déblocage rotation
	C FC6	E 7.6	BOOL	Capteur de fin de course mécanique plaque supérieure
	C FC7	E 7.7	BOOL	Capteur de fin de course mécanique plaque inférieure
	C FC8	E 8.0	BOOL	Capteur de fin de course mécanique encodeur en bas
	C FC9	E 8.1	BOOL	Capteur de fin de course mécanique plan basculant gauche
	C PE1	E 8.5	BOOL	Capteur photoélectrique bobine achevée
	C PE2	E 8.6	BOOL	Capteur photoélectrique présence pièce (table d'aménage)
	C PE3	E 8.7	BOOL	Capteur photoélectrique fosse N°1
	C PE4	E 9.0	BOOL	Capteur photoélectrique fosse N°2
	C PE5	E 9.1	BOOL	Capteur photoélectrique fosse N°3
	C PE6	E 9.2	BOOL	Capteur photoélectrique fosse N°4
	C PE7	E 9.3	BOOL	Capteur photoélectrique évacuation pièces par l'opérateur
	CEM 10	E 11.6	BOOL	Capteur électromagnétique barres rapprochées
	CEM 11	E 11.7	BOOL	Capteur électromagnétique barres éloignées
	CEM 2	E 9.5	BOOL	Capteur électromagnétique cisaille en bas
	CEM 3	E 9.6	BOOL	Capteur électromagnétique barres toute en avant
	CEM 4	E 9.7	BOOL	Capteur électromagnétique barres toute en arrière
	CEM 5	E 10.0	BOOL	Capteur électromagnétique barres en bas
	CEM 6	E 10.1	BOOL	Capteur électromagnétique barres en haut
	CEM 7 1 A	E 10.2	BOOL	Capteur détectant la retenue de la pièce dans le poste 1barre antérieure
	CEM 7 1 P	E 10.3	BOOL	Capteur détectant la retenue de la pièce dans le poste 1barre postérieure
	CEM 7 2 A	E 10.4	BOOL	Capteur détectant la retenue de la pièce dans le poste 2barres antérieures
	CEM 7 2 P	E 10.5	BOOL	Capteur détectant la retenue de la pièce dans le poste 2barre postérieure
	CEM 7 3 A	E 10.6	BOOL	Capteur détectant la retenue de la pièce dans le poste 3barre antérieure
	CEM 7 3 P	E 10.7	BOOL	Capteur détectant la retenue de la pièce dans le poste 3barre postérieure
	CEM 7 4 A	E 11.0	BOOL	Capteur détectant la retenue de la pièce dans le poste 4barre antérieure
	CEM 7 4 P	E 11.1	BOOL	Capteur détectant la retenue de la pièce dans le poste 4barre postérieure

Etat	Mnémonique	Opérande	Type de données	Commentaire
	CEM 7 5 A	E 11.2	BOOL	Capteur détectant la retenue de la pièce dans le poste 5barre antérieure
	CEM 7 5 P	E 11.3	BOOL	Capteur détectant la retenue de la pièce dans le poste 5barre postérieure
	CEM 8	E 11.4	BOOL	Capteur électromagnétique vérins de soulèvement pièce en haut
	CEM 9	E 11.5	BOOL	Capteur électromagnétique vérins de soulèvement pièce en bas
	CON	Z 1	COUNTER	compteur
	CONN	Z 2	COUNTER	
	CPE1	E 13.2	BOOL	
	CYCL_EXC	OB 1	OB 1	Cycle Execution
	Emb1	A 0.0	BOOL	Embrayage voulant d'inertie
	fin bobine	A 0.1	BOOL	affichage fin bobine
	Fr1	A 0.2	BOOL	Frein du voulant d'inertie
	Fr2	A 0.3	BOOL	frein translation des barres
	Fr3	A 10.6	BOOL	Frein du Moteur aménage
	Fr4	A 10.7	BOOL	Frein du Moteur dérouleur
	Led 1	A 11.0	BOOL	Témoin de mise sous tension
	Led 10	A 12.1	BOOL	Témoin de mode manuel (pupitre ligne).
	Led 11	A 12.2	BOOL	Témoin mode automatique (pupitre ligne)
	Led 12	A 12.7	BOOL	Témoin d'alimentation de la centrale hydraulique principale
	Led 13	A 12.4	BOOL	Témoin d'alimentation de la centrale hydrauliqued'équilibrage barres
	Led 14	A 12.5	BOOL	Témoin d'alimentation du circuit de graissage machine
	Led 2	A 11.1	BOOL	Témoins éjecteurs
	Led 3	A 11.2	BOOL	Témoin " tapis déchet marche "
	Led 4	A 11.3	BOOL	Témoin d'alimentation du moteur d'aménage
	Led 5	A 11.4	BOOL	Témoin d'alimentation cisaille
	Led 6	A 11.5	BOOL	Témoin " commande redresseur "
	Led 7	A 11.6	BOOL	témoin de mise sous tension du redresseur
	Led 8	A 11.7	BOOL	Témoin du mode manuel (pupitre ligne)
	Led 9	A 12.3	BOOL	Témoin de mode semi automatique (pupitre ligne)
	M E1	A 8.3	BOOL	Rotation rouleur presseur
	M E10	A 9.5	BOOL	Centrale hydraulique principale
	M E11	A 9.6	BOOL	Moteur équilibrage des barres
	M E12	A 9.7	BOOL	Centrale hydraulique graissage barres
	M E2	A 3.1	BOOL	Moteur du redresseur
	M E2 MGv	A 10.2	BOOL	Moteur redresseur marche à une grande vitesse
	M E2 MVF	A 10.0	BOOL	Moteur redresseur marche à une vitesse faible
	M E2 MVM	A 10.1	BOOL	Moteur redresseur marche à une vitesse moyenne
	M E3	A 8.5	BOOL	Moteur du groupe aménage
	M E4	A 8.6	BOOL	Moteur table d'aménage
	M E5 M	A 9.0	BOOL	Le tapis d'évacuation pièces
	M E7	A 9.2	BOOL	Stations hydrauliques.
	M E8	A 9.3	BOOL	Moteur d'éjecteur
	M E9	A 9.4	BOOL	Moteur tapis des déchets
	M EP	A 8.7	BOOL	Le moteur principal
	m H1 S1	A 13.2	BOOL	translation chargeur vers l'avant
	M H1 S2	A 8.0	BOOL	translation chargeur vers l'arriere
	M H2 D	A 8.1	BOOL	Rotation dérouleur droite
	M H2 G	A 8.2	BOOL	Rotation dérouleur gauche
	M W 5	A 12.6	BOOL	Compteur
	Ma_p	E 13.5	BOOL	Machine pret
	machine prete	A 13.0	BOOL	affichage machine prete
	ME3_M	A 14.2	BOOL	moetur me3 m

Etat	Mnémonique	Opérande	Type de données	Commentaire
	p_r	E 6.7	BOOL	presence des piece dans les 5 poste
	p_r F	E 14.4	BOOL	
	pas atteint	E 8.4	BOOL	Encodeur mesurant la longueur de la tôle sortie de la cisaille
	PNR	E 12.0	BOOL	la retenue des pièces dans tout les postes
	Sel 2_1	E 12.2	BOOL	Sélecteur des modes sur la position 1 : manuel (pupitreligne)
	Sel 2_2	E 12.3	BOOL	Sélecteur des modes sur la position 2 : semi-automatique(pupitre ligne)
	Sel 2_3	E 12.4	BOOL	Sélecteur des modes sur la position 3 : automatique (pupitreligne)
	Sel 3_1	E 12.5	BOOL	Sélecteur des modes sur la position 1 : manuel (pupitresuspendu)
	Sel 3_2	E 12.6	BOOL	Sélecteur des modes sur la position 2 : automatique (pupitre suspendu)
	Sel 4_0	E 12.7	BOOL	Sélecteur " sanction sécurité " sur position 0 -désactiver lessécurités
	Sel 4_1	E 5.7	BOOL	Sélecteur " sanction sécurité " sur position 1 -activer lessécurités
	Sel 5	E 6.1	BOOL	Sélecteurs d'alimentation des éjecteurs
	Sel 6	E 6.2	BOOL	Sélecteurs de mise en marche les tapis de déchets
	Sel1	E 6.3	BOOL	Sélecteur de mise sous tension
	V1 e	A 4.1	BOOL	
	V10 e	A 6.3	BOOL	Les deux vérins d'aménage
	V10 s	A 6.2	BOOL	sortie du Vérin roue encodeur
	V11e	A 6.5	BOOL	entrée du Vérin roue encodeur
	V11s	A 6.4	BOOL	
	V12 s	A 6.6	BOOL	
	V12e	A 6.7	BOOL	Les deux vérins de la cisaille
	V13e	A 7.1	BOOL	Vérins table d'aménage
	V13s	A 7.0	BOOL	
	V14e	A 7.3	BOOL	4 petits vérins de la retenue de la pièce
	V14s	A 7.2	BOOL	
	V15 s	A 7.4	BOOL	
	V15e	A 7.5	BOOL	Ouverture et fermeture mâchoire
	V16 e	A 7.7	BOOL	Rouleau postérieur redresseur
	V16 s	A 7.6	BOOL	
	V1s	A 4.0	BOOL	Vérin chargeur
	V2e	A 4.3	BOOL	
	V2s	A 4.2	BOOL	Vérin blocage rotation dérouleur
	V3 e	A 4.5	BOOL	
	V3s	A 4.4	BOOL	Vérin rouleau presseur
	V4 e	A 4.7	BOOL	Vérin plaque inférieure
	V4 s	A 4.6	BOOL	
	V5 e	A 5.1	BOOL	Vérin lame
	V5s	A 5.0	BOOL	
	V6 e	A 5.3	BOOL	vérin plaque supérieure
	V6s	A 5.2	BOOL	
	V7 e	A 5.5	BOOL	vérin plaque supérieure
	V7 s	A 5.4	BOOL	
	V8 e	A 5.7	BOOL	Vérin plan basculant gauche
	V8s	A 5.6	BOOL	
	V9 e	A 6.1	BOOL	Vérin plan basculant droit
	V9s	A 6.0	BOOL	
	1	E 14.5	BOOL	TRANSITION VRAIS

Bibliographie

[1] : Documentations **ENIEM**.

[2] : Documentation technique SIMATIC MANAGER, version 5.4 2006.

[3] : Documentation technique SIMATIC WIN CC flexible 2008.

[4]: « Du Grafcet aux réseaux de pétri » Deuxième édition revue et augmentée. **RENE DAVID, HASSANE ALLA**.

[5] : « GRAFCET et logique industrielle programmée» .**THELIEZ SYLVAIN**.

[6]: mémoire de fin d'étude «automatisation d'une machine à mousser les portes de réfrigérateurs based'un API SIMATIC « S7-300».Département d'Automatique.Promotion2008. **T.SIDHOUM, M.BERREAS**

[7] «comprendre, maitriser et appliquerle GRAFCET » Edition CEPADUES 1994
.M.BLANCHARD

[8] : Mémoire de fin d'études U.M.M.T.O département automatique «Automatisation de l'équipement de moussage des portes de réfrigérateurs par API S7-300 à l'ENIEM », promotion 2007/2008.

[9] :Projet de fin d'études U.M.M.T.O département automatique Automatisation de la chaîne pour mousser des armoires à périphériques décentralisées par l'API S7-300, promotion 2008/2009.

[10]Projet de fin d'études U.M.M.T.O département automatique Etude et automatisation par automates programmables S7-300 d'une presse transfert à l'entreprise ENIEM, promotion 2008/2009.

[11] : Navigation sur le moteur de recherche « Google » (Recherche par mots clé).

[12]: **Site web** <http://pagesperso-orange.fr/xcotton/electron/coursetdocs.html>.

Résumé

Dans notre projet de fin d'étude, les responsable de l'unité cuisson de ENIEM nous a propose d'étudier la presse de transfert, qui est une machine qui fabrique les moules de cuisinières ; dont le but est d'automatisation en passant de S5 vers S7 ; pour cela nous avons :

- étudier la conception de la machine et décrire ces différents blocs
- nous avons modélisé la machine par un outil très puissant GRAFCET afin de pouvoir programmer
- nous avons choisir l'automate programmable S7-300
- en fin on a terminé par la supervision des différentes blocs de la machine sous le logiciel Wincc flexible.
-

Mots clés:

ENIEM, SIEMENS, VDE, MLI, PWM, MCC, Q, KM, Rth, Arr-U, Bp, RDP, GRAFCET, ADEPA, ME, STEP7, S7-300, API, PLC, IL, ST, LD, SFC, FBD, SIMATIC S7, CPU, MPI, PROFIBUS, INDUSTRIAL, ETHERNET, PG, RAM ROM, EPROM, PS,CPU,IM, SM,FM,CP, RUN-P, RUN, STOP, MRES, SF, BATF, Dc5v, FCRE, EANA/SANA, FM, OB, T.O.R, CP, LIST, LOG, CONT, SCL, FB, Fc, DB, S7-PLCSIM,S5 ,S7, Win CC, OS, VBS, TIA, IHM, NetPro.