

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'Electrotechnique



de fin d'études en vue de l'obtention du diplôme
D'INGENIEUR D'ETAT EN ELECTROTECHNIQUE

Option : Machines Electriq

Thème

Etude de l'automatisation par automate programmable

S7-300 de la machine à garnir les encoches

De l'ENEL

Proposé par

Mr A. HARBIT (ENEL).

Dirigé par

Mr R. MANSOURI (UMMTO).

Présenté par :

Mr FAHEM NASSIM

Mr HAMMAR YAZID

Promotion : 2008

Remerciements

Au terme de notre travail nous tenons à remercier tous ceux qui ont contribué de près ou de loin à sa réalisation.

Nous tenons à exprimer nos sincères remerciements à notre co-promoteur **Mr A. HARBIT** qui nous a proposé ce sujet et pour l'aide précieuse qu'il nous a apporté.

Nos remerciements les plus spéciaux à notre promoteur **Mr R. MANSOURI** qui a accepté de nous encadrer et pour sa disponibilité et ses orientations pour mener à bien notre travail.

Nous remercions toutes les personnes qui nous ont encouragés pour aller au bout de notre travail, spécialement **Mr M. FAHEM**, chef du département maintenance à l'ENEL.

Nos grands et sincères remerciements à nos chères familles qui nous ont soutenus.

Nos derniers remerciements vont aux membres de jury qui nous feront l'honneur de juger notre travail.

SOMMAIRE

Préambule	1
Introduction générale.....	3
Chapitre I : Description de la machine	
Introduction	5
I -1 Partie opérative	7
I -1-1 La structure mécanique	7
I -1-2 pré actionneurs.....	9
I -1-3 Actionneurs	11
I -1-4 Capteurs	12
I -2 Le circuit pneumatique	13
I -3 Circuit Electrique	15
I -3-2 Module de commande	16
I -4 Fonctionnement de la machine	16
I -4-1 Fonctionnement Automatique	16
I -4-2 Fonctionnement pas à pas.....	17
I -4-3 Ajustage.....	17
I -5 Description des différentes étapes du fonctionnement.....	17
I -5-1 Avance	17
I -5-2 Formage- coup	18
I -5- 3 Enchâssement	19
I -5-4 Rotation du plateau	20
Conclusion	22
Chapitre II : Modélisation de la machine par GRAFCET	
II -1 Introduction au GRAFCET.....	23
II -2 Historique.....	23
II -3 Définition	23
II -4 Cahier de charges.....	23
II -5 Les éléments de base du GRAFCET.....	24

II -5-1	Les étapes.....	24
II -5-2	Les transitions et les réceptivités associées.....	25
II -5-3	Liaisons orientées.....	27
II -5-4	Règles d'évolution du GRAFCET.....	27
II -5-5	Erreurs de GRAFCET.....	28
II -5-6	Sélection de séquence et séquences simultanées.....	29
II -6	Niveaux d'un GRAFCET.....	31
II -6-1	GRAFCET de niveau 1.....	31
II -6-2	GRAFCET de niveau 2.....	31
II -7	Application du GRAFCET pour modéliser la machine.....	32
II -7-2	Liste des vérins et rôle attribué à chacun.....	32
II -7-3	Capteurs et bouton poussoirs.....	33
II -7-4	Abréviations utilisées dans le GRAFCET de niveau 2.....	34
II -7-5	Représentation du GRAFCET de niveau 1.....	35
II -7-6	Représentation du GRAFCET de niveau 2.....	37
Conclusion	39

Chapitre III : introduction aux automates programmable

III -1	Introduction	40
III -2	Définition	40
III -3	Place des automates programmables	40
III -4	Mode de fonctionnement	41
III -5	Architecture interne d'un automate programmable	41
III -5-1	Module d'alimentation	42
III-5-2	Unité centrale de traitement (CPU)	42
III -5-3	Modules d'entrées	43
III -5-4	Modules de sorties	44
III -5-5	Modules de communication	44
III -6	Choix d'un API	46
III -7	Les Avantages des API	46
III -8	Les Inconvénients des API	46

III -9	Programmation d'un API	46
III -9-1	Langage de Programmation	46
III -10	Mise en Œuvre d'un API	48
Conclusion	48

Chapitre VI : l'Automate S7-300 et son langage de programmation

Introduction	49
IV -1	Constitution de l'Automate S7-300	49
IV -2	Caractéristiques de l'automate S7-300	49
IV -3	Modularité du S7-300	50
IV -3-1	Unité Centrale (CPU)	50
IV -3-2	Modules d'alimentation	52
IV -3-3	Modules de signaux « SM »	52
IV -4	Logiciel STEP7	54
IV -4-1	Introduction.....	54
IV -4-2	Définition du STEP7	54
IV -4-3	Programmation de l'automate S7-300	54
IV -4-4	Adressage des modules du S7-300	55
IV -4-5	Mémentos	57
IV -4-6	Traitement du programme par la CPU	57
IV -4-7	Les différents blocs du programme utilisateur	59
IV -4-8	Création d'un projet STEP7	62
IV -4-9	Configuration et paramétrage de l'automate	68
Conclusion.....	70

Chapitre V : Simulation et validation avec PLCSIM

V-1	Introduction	71
V-2	Présentation du S7 PLCSIM.....	71
V-3	Mise en route du logiciel S7-PLCSIM	71
V-4	Visualisation de l'état du programme	75
	Conclusion	76
	Conclusion générale	77

Introduction générale

Dans la conjoncture relative à la mondialisation et l'économie du marché, les entreprises nationales doivent impérativement, améliorer leur productivité tant sur le plan de la qualité que du coût de leur produits afin d'éviter leur disparition.

Pour ce faire, il est nécessaire que ces entreprises procèdent à une rénovation totale des moyens technologiques de productions.

Actuellement, et à travers le monde, les grandes entreprises donnent beaucoup d'importance à l'automatisation des différentes tâches de production, et les raisons de cet intérêt sont dues aux différents avantages que procure ce procédé,

Avec le progrès technologique, l'automatisation des installations constitue un des facteurs essentiels contribuant à la croissance de la productivité, et un élément important dans l'amélioration de la sécurité du travail ainsi que la réduction des coûts de production.

De nos jours, grâce au développement de l'informatique et de la microélectronique et à la fabrication de microprocesseurs de plus en plus performants, l'automatisation est assurée par des Automates Programmables Industriels (API) qui intègrent ces nouvelles technologies.

Depuis l'ouverture de l'entreprise **Electro-Industries (EI)** en 1983, elle s'est équipée de plusieurs machines, les premières étaient dotées de commandes électromécaniques, les suivantes étaient commandées par des séquenceurs électriques, et les plus récentes sont commandées par des automates programmables.

Au niveau de ces deux chaînes de production (moteurs et transformateur), on trouve de nombreux postes de production, utilisant des automatismes en technologie câblée à base de relais et contacteurs, avec tout les inconvénients que présentent ces derniers, tel que la durée de vie limitée, les pannes fréquentes provoquées par l'usure mécanique des contacteurs, ...etc.

Dans ce contexte le sujet qui nous a été proposé par le Département "Maintenance" de l'unité moteurs/E.I, consiste à remplacer le système de commande d'une machine à garnir les encoches existant, par la commande à base d'un automate programmable industriel « **SIEMENS** ».

Pour ce faire on à organiser notre travail comme suite :

- Le premier chapitre est consacré à la description de la machine, et son fonctionnement dans l'état actuel (logique câblée).
- Le deuxième chapitre se base sur la modélisation de l'unité de commande par l'outil **GRAFCET**.
- Le troisième chapitre nous met en contact avec les automates programmables, en ce référent à des données générales.
- Le quatrième chapitre est consacré à l'étude de l'automate **S7-300**, avec ses deux aspects, matériel et logiciel.
- Le dernier chapitre est réservé à la simulation via le logiciel **S7-PLCSIM**

Finalement nous terminons notre travail par une conclusion générale

Introduction :

La machine électrique tournante dans son ensemble se compose de deux parties essentielles, le rotor et le stator. Si on prend l'exemple des machines à courant alternatif, de type synchrone ou asynchrone, le processus de fabrication du stator passe par plusieurs étapes . A savoir : le découpage et l'isolation par vernis des tôles magnétiques, l'emballage,...etc.

La mise en œuvre du paquet statorique étant achevée, on doit procéder au montage de l'enroulement (bobinage), qui assure une fonction très importante dans le fonctionnement de la machine, se traduisant par la création du champ tournant, lorsque ce dernier est soumis à un système de tension triphasé.

Avant de placer ce dernier dans le paquet statorique, on doit procéder à l'isolation des encoches, cette étape est très importante, elle permet d'éviter les éventuels courts circuits pouvant se produire, du fait que le vernis n'assure qu'une isolation entre conducteurs et ces derniers par frottement avec le stator peuvent perdre leur caractéristiques d'isolation..

Le procédé d'isolation des encoches statoriques se fait à l'aide d'un papier particulier, muni d'une certaine rigidité. Ce dernier est découpé aux mêmes dimensions internes de l'encoche et est inséré dans celle-ci avant qu'elle ne reçoive l'enroulement.

La figure suivante illustre une coupe transversale du stator, et à titre d'explication on a représenté une encoche isolée par une gaine de papier.

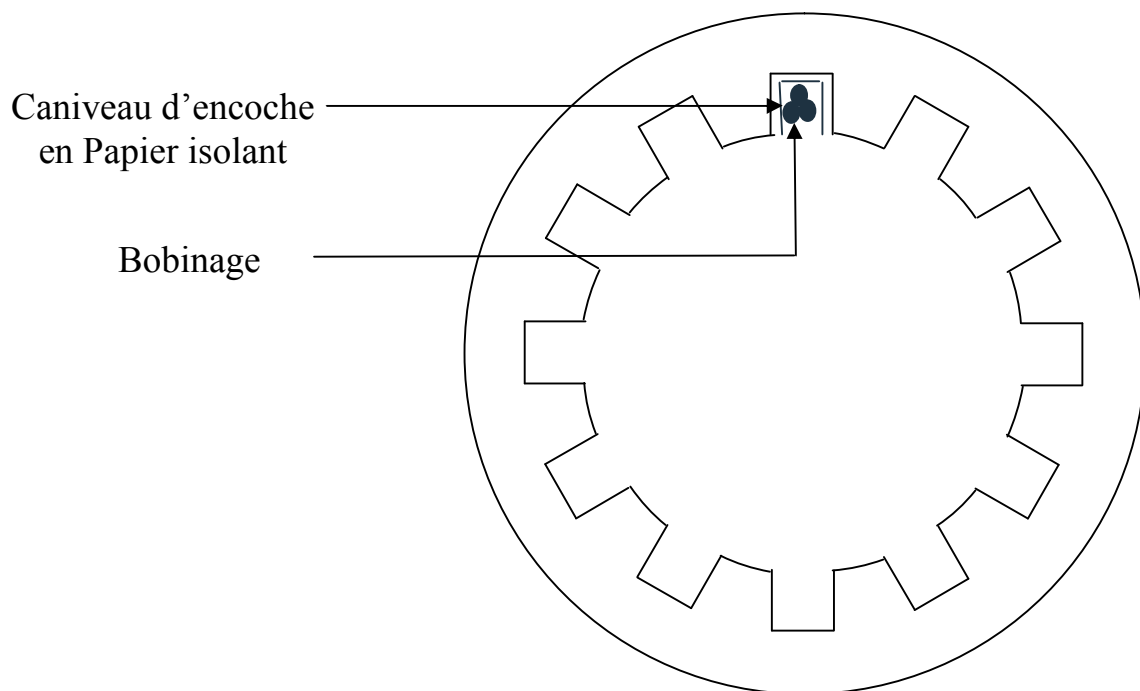


Figure I.1 : schéma illustrant l'isolation d'une encoche à l'aide d'une gaine de papier

La machine étudiée est une machine à garnir les encoches, son travail consiste à découper des gaines à partir d'un rouleau de papier isolant fixé sur un dérouleur, et à les insérer dans les encoches d'un stator placé sur un plateau horizontal muni d'un mouvement de rotation autour de son axe.

La machine à garnir les encoches est une machine à commande électropneumatique, ses différents mouvements sont assurés par des vérins à double effet, qui sont alimentés par des électro distributeurs, dont les bobines sont excitées via les organes de commandes.

La figure I.2 illustre l'allure extérieur de cette machine.



Figure I.2 : Machine à garnir les encoches : vue d'ensemble

D) Description de la machine :

La machine est essentiellement composée des modules suivants :

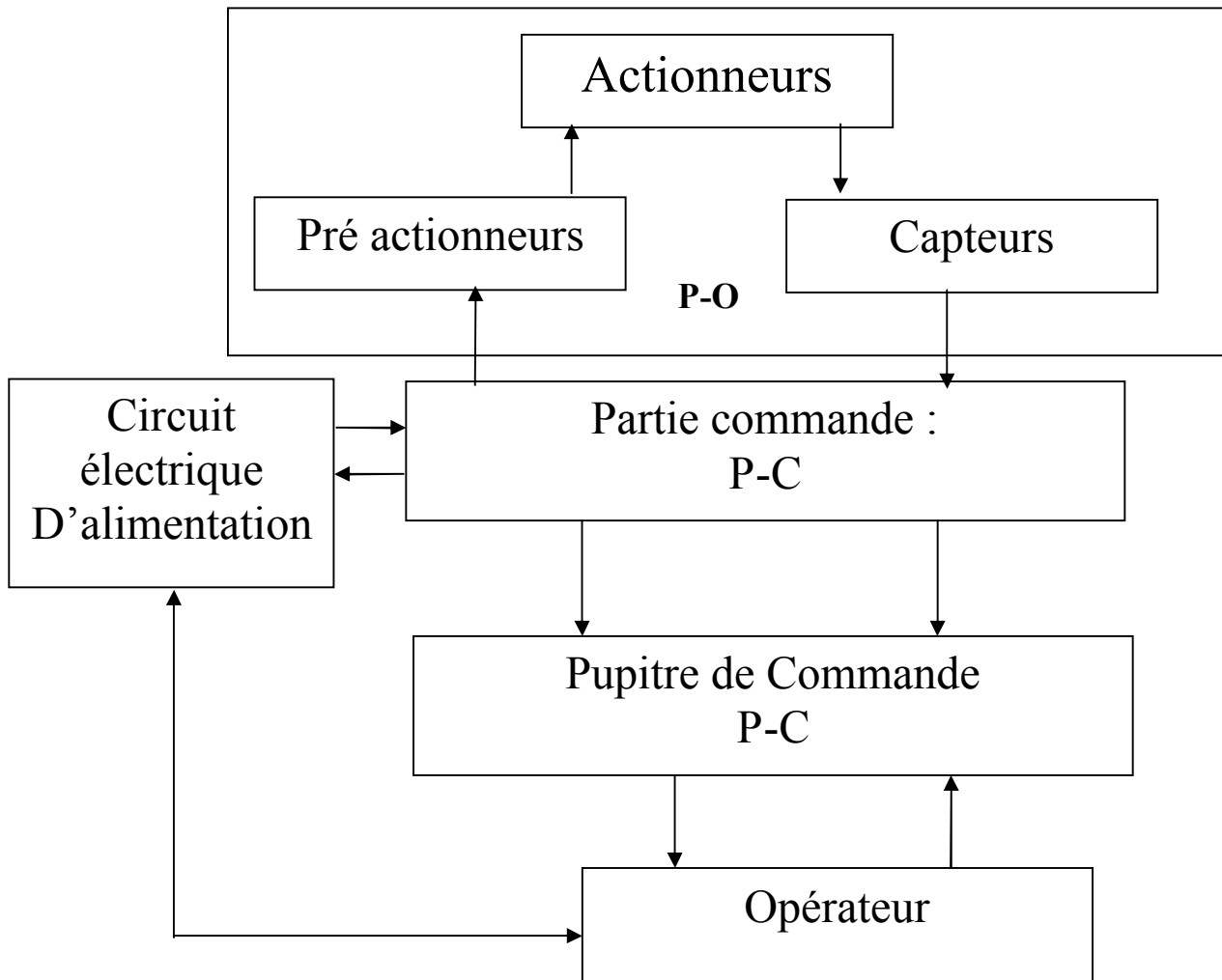


Figure I.3 : différents modules de la machine.

I-1 Partie opérative :

Elle représente la structure mécanique, composée des différents pré-actionneurs, actionneurs, capteurs et tout le circuit pneumatique.

I-1-1 Structure mécanique de la machine :

La machine comporte essentiellement deux parties

- **Une partie basse :** Elle contient les systèmes de vérins pneumatiques à déclenchement électrique, assurant les fonctions : « avance du papier », « le formage-coupe » et « enchâssement de la gaine ».

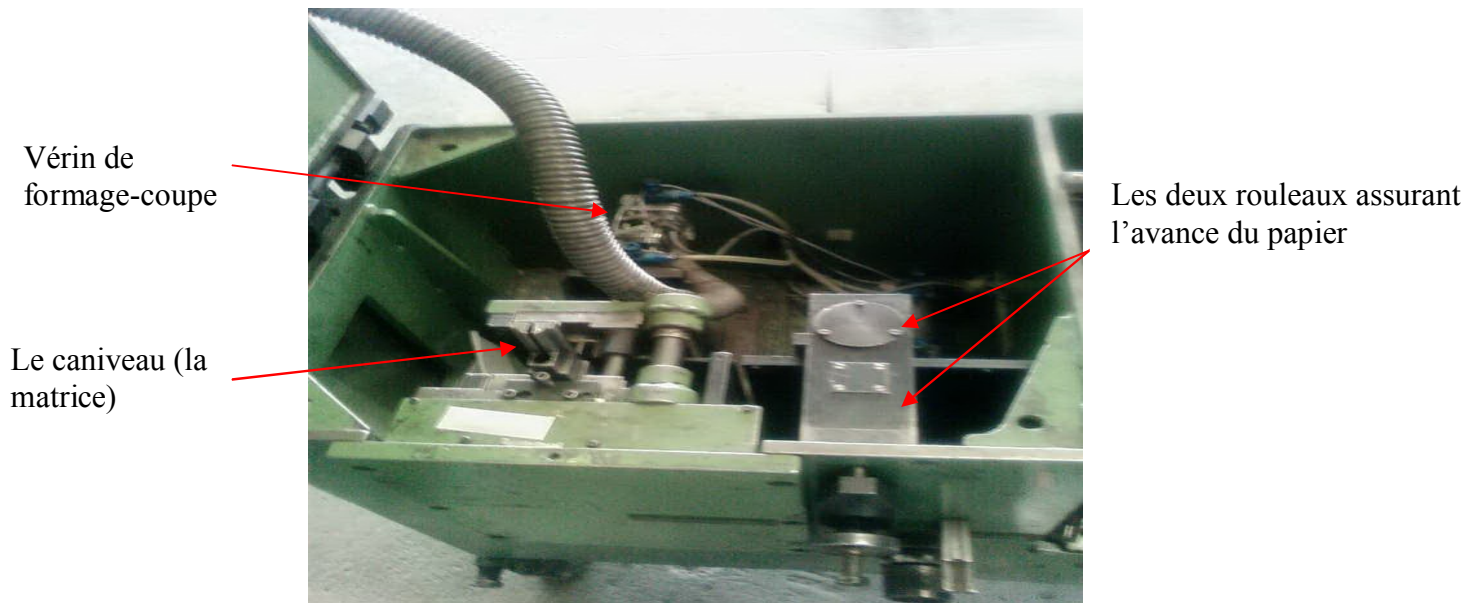


Figure I.4 : Intérieur de la machine (partie basse de la machine).

- **Une partie haute :** C'est la partie qui contient le plateau sur lequel on fixe le stator. Sur cette dernière on trouve les différents vérins, assurant les fonctions « d'indexage », « rotation du plateau » et « d'encliquetage ».

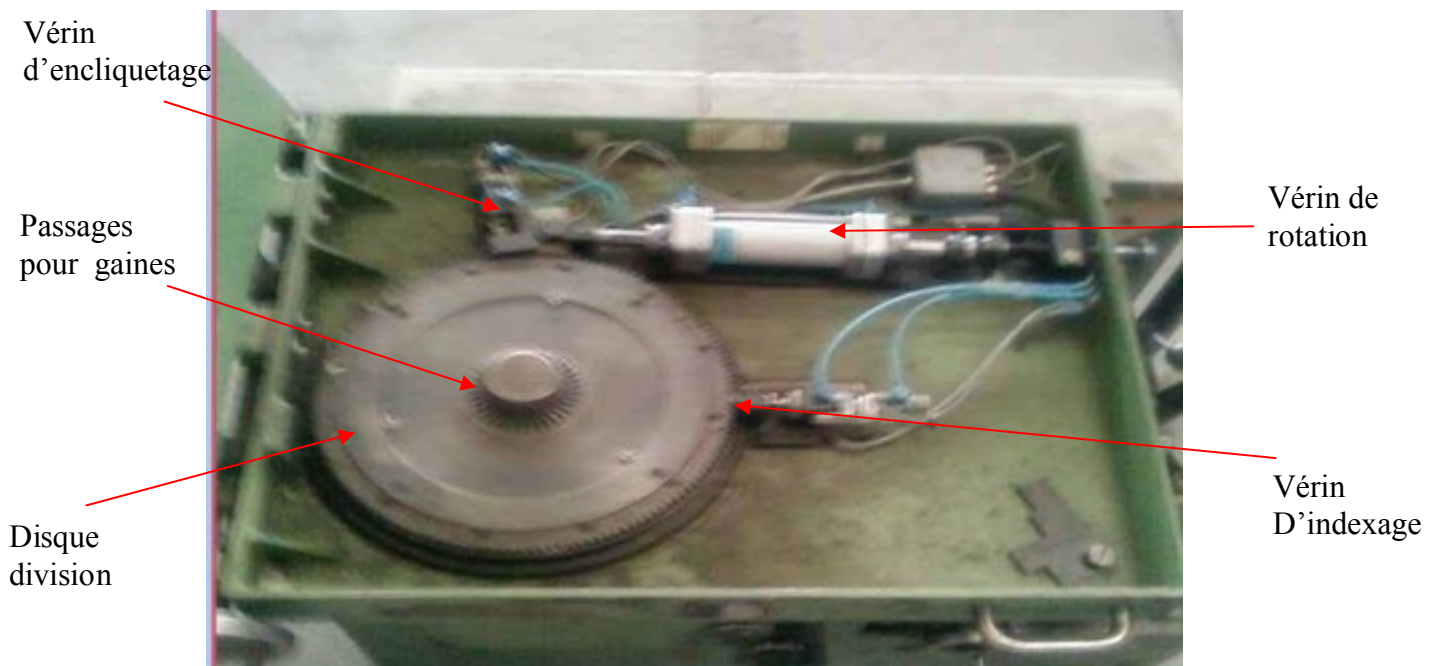


Figure I.5 : Dispositif de rotation (partie haute de la machine).

I-1-2 Pré actionneurs :

Ce sont des dispositifs, servant à une gestion de l'énergie en fonction de la commande de l'actionneur.

Ils s'interposent entre la source d'énergie de puissance et l'actionneur qu'ils alimentent.

- Distributeur :

Organe permettant d'injecter l'air comprimé aux différents actionneurs de la machine.

Un distributeur est caractérisé par :

- le nombre d'orifices : 2, 3, 4 ou 5 ;
- le nombre de mode de distribution ou position : 2 ou 3 ;
- Le type de commande du pilotage assurant le changement de position, simple pilotage avec rappel par ressort ou double pilotage ;
- la technologie de pilotage : pneumatique ou hydraulique ;

La machine est munie d'un seul type de distributeur, il s'agit du distributeur 4/2 (4 orifices, 2 position).

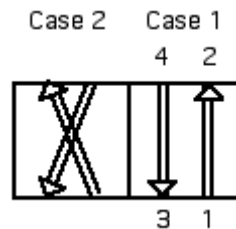


Figure I.6 : schéma d'un distributeur 4/2

La structure réelle de l'ensemble électro-distributeur (distributeur + électrovanne), est illustrée sur la figure suivante :



Figure I.7 : Distributeur Electro-pneumatique.

- Electrovanne :

C'est un composant électrique qui fonctionne en tout ou rien (deux états stables), engendrant le déclenchement du distributeur (passage ou blocage de l'air comprimé).



Figure I.8 : schéma et photo de l'électrovanne.

L'électrovanne est constituée principalement :

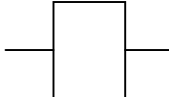
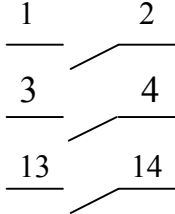
D'un noyau métallique attaché a un ressort, et d'une bobine, et lorsque celle-ci est alimentée, il y a création d'un champ magnétique qui provoque le déplacement du noyau (ressort en compression), et avec l'action du ressort le noyau reprend sa position initiale, assurant ainsi la pénétration ou le blocage de l'air comprimé.

Ce type de pré actionneur est commandé par une tension continue d'une valeur de: 24 volts

- Relais :

Leur rôle est d'ouvrir ou fermer un circuit électrique, ils sont des appareils à commande électromagnétique.

Lorsque un signal de la partie commande arrive, la bobine du relais est alimentée et elle attire l'armature provoquant ainsi la commutation des contacts.

Désignation	Symbole							
KM		<table style="border: none;"> <tr> <td style="border: none;">1</td> <td style="border: none;">2</td> </tr> <tr> <td style="border: none;">3</td> <td style="border: none;">4</td> </tr> <tr> <td style="border: none;">13</td> <td style="border: none;">14</td> </tr> </table> 	1	2	3	4	13	14
1	2							
3	4							
13	14							

- Repérage :

Circuit de puissance : repères à un chiffre.

Circuit de commande : repères à deux chiffres.

I-1-3 Actionneurs :

Ce sont les constituants qui permettent d'agir directement à l'endroit où on a besoin d'une force, en se servant de l'énergie provenant d'une source extérieure de telle manière à assurer une action physique en fonction de la tâche souhaitée.

- les vérins :

Ils sont utilisés dans la plupart des industries manufacturières, permettant de reproduire les actions manuelles d'un opérateur telles que : serrer, soulever, poinçonner, plier,...etc.

On a en général deux sortes de vérins :

• Vérin simple effet :

Ce type de vérin reçoit l'énergie dans un seul sens, avec un seul orifice d'alimentation permettant l'arrivée de cette énergie, en provoquant le déplacement du piston dans un seul sens. Son retour est assuré par l'action d'un ressort, ce type de vérin est utilisé au cas des faibles courses.

• vérin double effet :

Ce type de vérin offre la possibilité de recevoir l'énergie qu'il transforme en travail (déplacement) à travers deux orifices d'alimentation, permettant ainsi de commander le piston dans les deux directions de fonctionnement.

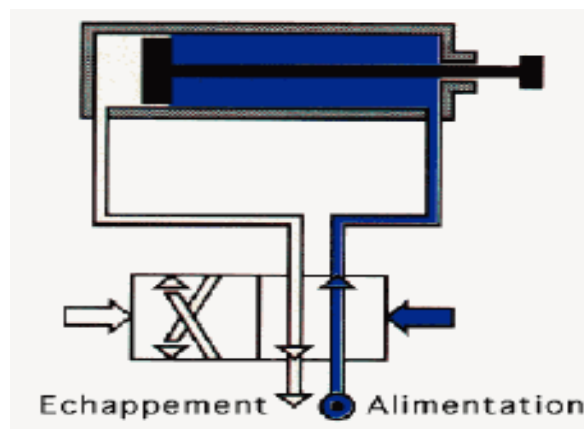


Figure I.9 : Vérin pneumatique à double effet.

I.1.4 Capteurs :

Ce sont des composants offrant des informations sur un processus donné, en faisant une transformation d'une grandeur d'entrée (grandeur mesurée), en une grandeur exploitable par un système de traitement (grandeur de sortie).

Le capteur délivre une information qui peut être logique (2 états 1 ou 0), ou bien numérique (en valeur discrète). Cette information peut également être analogique, dans ce cas, elle doit être convertie en un signal numérique.

- capteur de proximité :

Il permet la détection de l'objet sans contact mécanique, le principe de fonctionnement est basé sur l'action d'un champ magnétique. On trouve plusieurs types, détecteurs à effet capacitif et d'autres à effet inductif

• Détecteur de proximité inductif :

Il permet la détection des métaux ferreux.

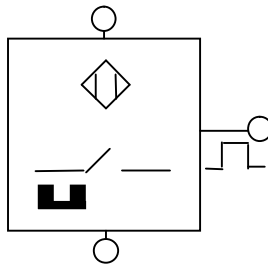


Figure I.10 : Capteur de proximité magnétique

- capteur de pression (pressostat) :

Permet la mesure de la pression, en ayant la pression de l'air comme signal d'entrée, et suivant la valeur de celle-ci, égale ou supérieur à la valeur de consigne, il donne une réponse sous forme d'un signal électrique tout ou rien.

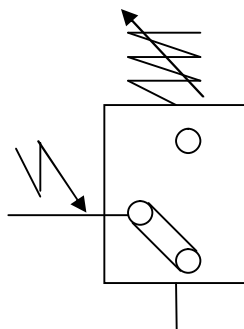


Figure I.12 : Schéma du capteur de pression

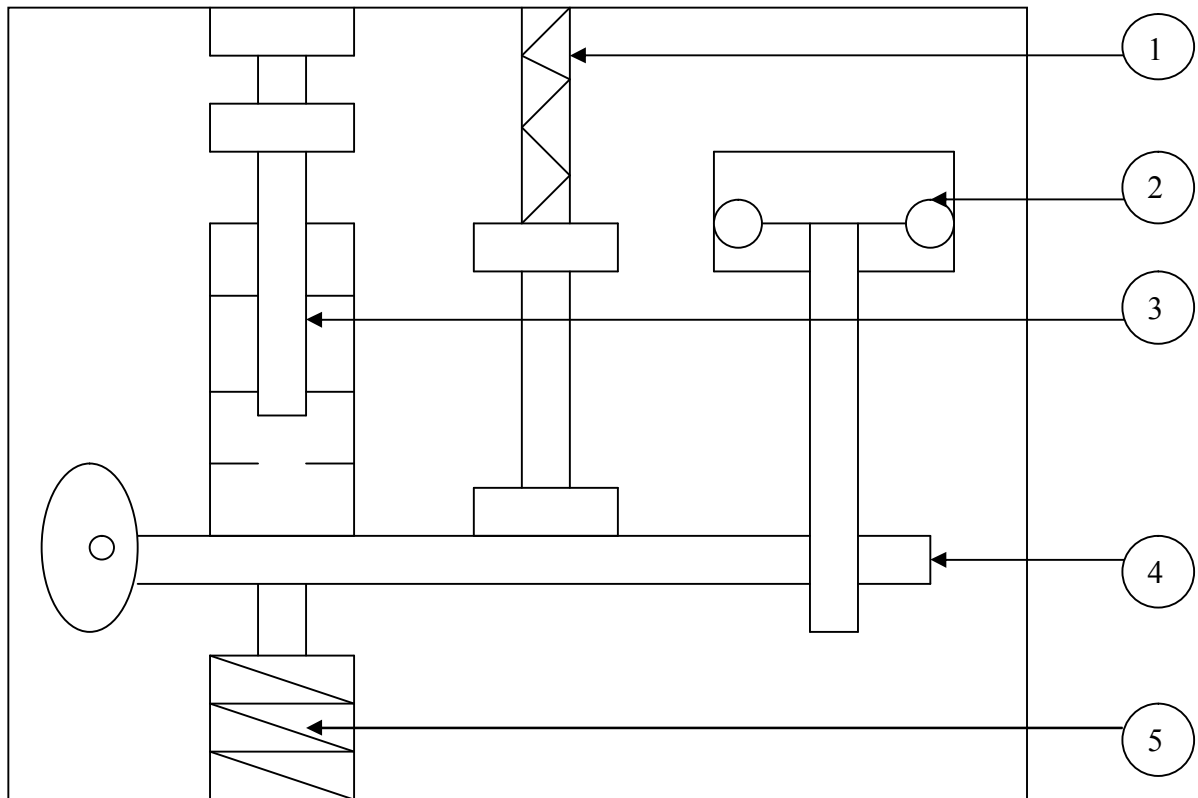


Figure I.13 : Principe du capteur de pression

- | | |
|---|--|
| 1- Ressort de réglage de point haut. | 4- Levier d'action. |
| 2- Contact électrique. | 5- ressort de réglage de l'écart. |
| 3- Élément sensible. | |

La tige retenue par le ressort, reçoit la force exercée par la membrane sensible, et se déplace pour actionner le commutateur. Et quant la pression diminue, la tige reprend sa position initiale (pressostat a l'état normal).

Ce capteur est utilisé pour la détection et le contrôle des pressions dans un circuit hydraulique ou pneumatique.

I -2 Le circuit pneumatique :

Le circuit pneumatique comporte les différents éléments qui fonctionnent à base de l'air comprimé.

- Source d'air comprimé :

C'est un réservoir (citerne) ou l'on stock de l'air sous une très haute pression. Il est relié par une conduite aux différents points d'utilisation de ce dernier.

- **Filtre :**

Il sert à éliminer les impuretés contenues dans l'air comprimé, afin d'éviter des perturbations (Encrassements) du système pneumatique: vérins, détendeur,...etc.

Les liquides ainsi que les grosses particules solides sont séparés par la force centrifuge, et la séparation se fait à l'aide d'une capsule à tourbillon qui se trouve à l'entrée supérieur de la tête du filtre, vers la fin ces impuretés glissent dans la zone de repos du filtre.

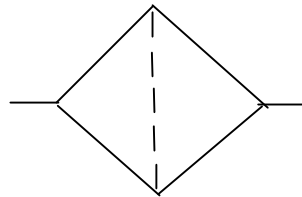


Figure I.14 : Schéma d'un filtre

- **Manomètre :**

Il indique la pression d'air de service, dans les différents éléments constituant le circuit pneumatique.



Figure I.14 : Manomètre.

- **Régulateur de Débit Unidirectionnel :**

Il combine l'action d'un clapet anti-retour, et d'un clapet d'étranglement, le clapet anti-retour bloc le passage d'air comprimé dans une direction, et ce dernier passe à travers le clapet d'étranglement

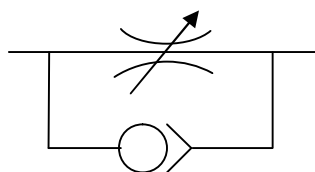


Figure 1.15 : Réducteur de débit Unidirectionnel

- Soupape de décompression rapide :

Elle sert à augmenter la vitesse d'exécution d'un vérin (vitesse de déplacement du piston) et cela en facilitant la sortie (l'échappement) de l'air contenu dans celui-ci.

Comme c'est illustré sur le circuit pneumatique, le vérin qui assure l'enclassement est équipé d'une soupape d'échappement rapide.

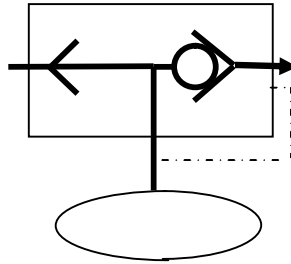


Figure I.16 : Purge rapide ou bien soupape de décompression rapide

I -3 Circuit Electrique :

Il sert à alimenter en courant électrique les différents composants, en délivrant différentes tensions (220v, 24v).

Il est constitué de deux modules essentiels :

- Module de puissance
- Module de commande

I -3-1 Module de puissance :

Il comporte les éléments suivants :

- Une source d'alimentation triphasée.
- Le sectionneur principal, servant à l'ouverture ou la fermeture du circuit électrique.
- Des fusibles pour la protection contre les surtensions.
- Un transformateur à isolation galvanique, permettant la transformation de la tension d'entrée de 380 volts, pour donner une tension de 220 volts, et assurant la protection contre les perturbations du réseau d'alimentation
- Un autre transformateur offrant la tension nécessaire pour alimenter la partie commande.

- Le pont de diode, qui permet le redressement de la tension, pour donner 24volts en continu.
- Les contacteurs à commande électromagnétique pour l'ouverture ou la fermeture d'un circuit électrique.

I -3-2 Module de commande :

Il est composé des éléments suivants :

- les voyants lumineux incorporés dans les boutons poussoirs, pour connaître l'état de fonctionnement du sujet à commander.
- Les différents pré-actionneurs servants à la commande des étapes du fonctionnement (électrovanne, contacteurs, etc)

I -4 Fonctionnement de la machine :**- Préparation de la machine :**

Au départ on doit s'assurer d'une bonne préparation de la machine, ce qui peut être vérifié en ce référant aux conditions suivantes :

- a) bonne mise en place de l'outillage.
- b) Régler le nombre d'encoche du paquet statorique.
- c) Ordonner l'avance de la gaine.
- d) Suivre les étapes nécessaires pour choisir la longueur de la gaine d'encoche.

- Mise en marche :

Pour mettre en service la machine on doit suivre les étapes suivantes :

- a) Enclencher l'interrupteur principal se trouvant sur l'armoire électronique.
- b) Ouvrir l'alimentation en air comprimé
- c) Vérifier que le bouton « Arrêt d'urgence » est déverrouillé
- d) Actionner le bouton poussoir de mise en service « EN » se trouvant sur le tableau de commande, et le voyant vert « Prés » s'allume.

Toutefois la machine nous offre la possibilité de choisir les trois modes de fonctionnement suivants :

I -4-1 Fonctionnement Automatique :

On positionne le décompte sur le nombre d'encoche statoriques, puis grâce à un sélecteur se trouvant sur le tableau de commande, on choisi le mode automatique, le bouton poussoir « marche automatique » s'allume.

Après avoir actionné ce dernier, le voyant incorporé s'éteint, et la machine se met en fonctionnement (insérer les gaines de papier dans les encoches du stator). Elle ne se met à l'arrêt qu'après avoir isolé toutes les encoches du stator voulu, (c à d : remise à l'état initial du décompte).

I -4-2 Fonctionnement pas à pas :

En choisissons de mettre le sélecteur sur la position « pas à pas », le bouton poussoir correspondant s'allume, et dès qu'on actionne ce dernier, la machine effectue un cycle de travail complet, c à d (toutes les actions nécessaires pour l'isolation d'une seule encoche).

I -4-3 Mode ajustage :

On positionne le sélecteur sur le mode ajustage, et les boutons poussoirs: «avance», «formage- coupe», «enchâssement», «division du plateau» (représentant les différentes étapes du processus) s'allument. Ces fonctions peuvent être démarrées individuellement, et cela en actionnant le bouton correspondant.

I -5 Description des différentes étapes du fonctionnement :

Afin d'assurer l'isolation des encoches, la machine suit un processus de travail repartit en six étapes. Et pour établir un fonctionnement normal et ainsi éviter tout risque d'accident, ces dernières doivent suivre l'ordre de succession suivant : « formage coupe » , « enchâssement », « avance du papier », « encliquetage », « indexage » et « rotation du plateau ».

Les différentes étapes de ce processus sont assurées par des vérins à double effet. Leur fonctionnement est décrit comme suit :

I -5-1 Formage-coupe :

Cette étape consiste à découper le rouleau de papier isolant en petites gaines de dimensions équivalentes à celles de l'encoche à isoler, et à leur donner la même forme géométrique que cette dernière.

La fonction « formage-coupe » est assurée par un vérin à double effet (Z_1) placé horizontalement sur la partie basse de la machine et commandé par une électrovanne (Y_{1a} (4/2)).

Le piston de ce vérin est muni d'un outil (poinçon) dit de formage-coupe dont la forme est illustrée dans la figure I.17.

Lorsque la bande de papier atteint le niveau de la matrice (caniveau), le piston du vérin enchâssement est en position basse. La bobine de l'électrovanne (Y_{1a}) s'excite et l'arrive d'air vers le vérin change de côté (d'orifice), ainsi le piston du vérin (Z_1) se déplace de sa position initiale vers l'avant et le poinçon enfonce le papier dans la matrice pour lui donner la forme de cette dernière, et grâce à ses arrêtes tranchantes, la bande de papier y est également découpée.

La figure I.17 illustre les différents éléments qui composent le système de la fonction formage-coupe :

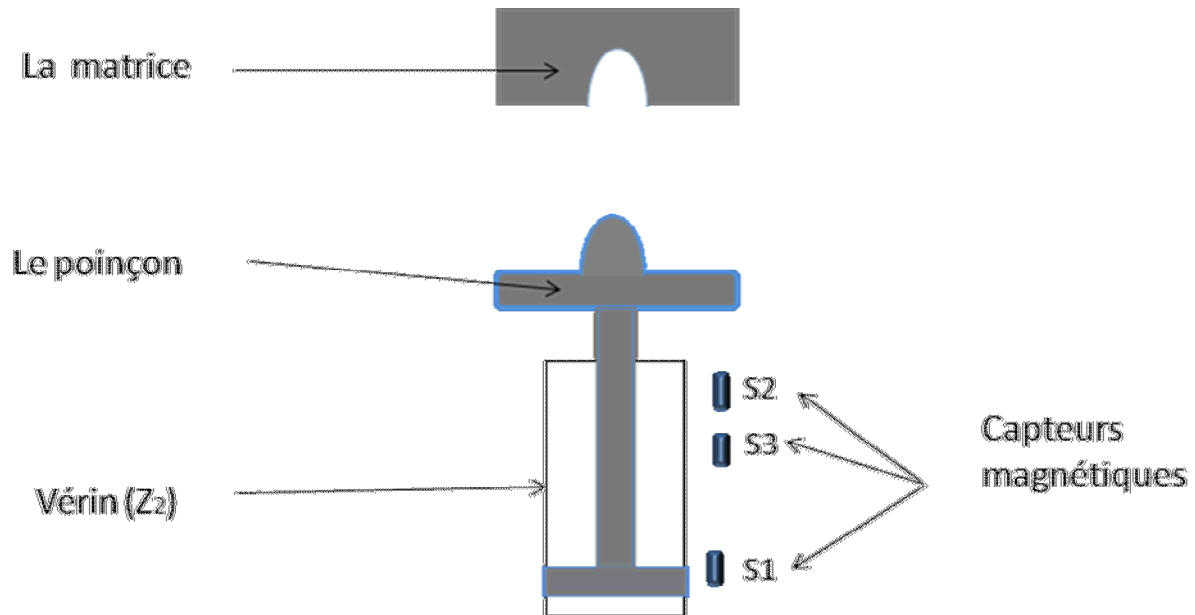


Figure I.17 : dispositif assurant la fonction formage-coupe

I -5-2 Enchâssement :

Après avoir procédé à l'étape du formage coupe, la gaine de papier reste dans le caniveau. A fin de pouvoir l'insérer dans l'encoche, un vérin à double effet (Z_3) dont le piston est muni d'un outil appelé couteau est placé verticalement au-dessous de ce même caniveau.

Le vérin (Z_3) est commandé par une électrovanne (Y_{3a} (4/2)), lorsque la bobine de cette dernière est excitée (grâce aux signaux délivrés par les différents capteurs (S_3 , S_6 , S_7 ...etc.)), l'arrivée d'air dans le vérin change de côté (d'orifice), et le piston qui était initialement en position basse se déplace vers le haut, ainsi le couteau pénètre dans le caniveau et pousse la gaine de papier afin de l'insérer dans l'encoche du stator.

La figure I.18 illustre les différents éléments qui composent le système de la fonction enchâssement :

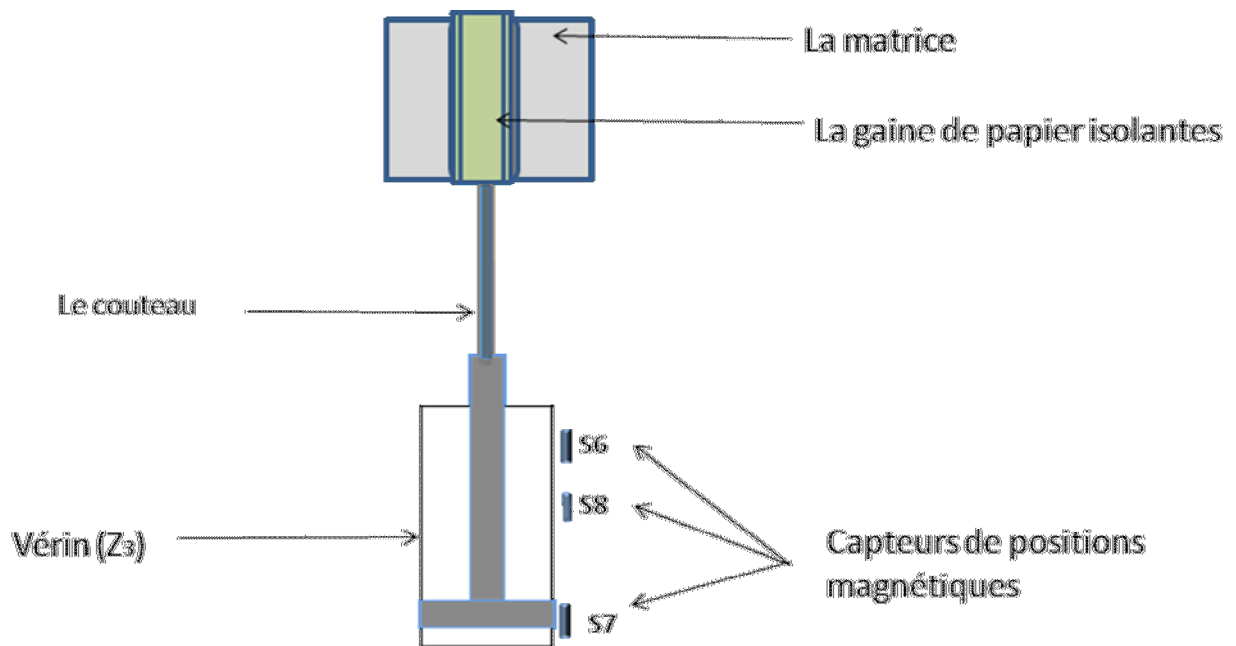


Figure I.19 dispositif assurant l'enchâssement de la gaine

I -5-3 Avance :

Dans le but de faire avancer la bande de papier jusqu'au lieu de formage et de découpage de la gaine d'encoche, celle-ci est insérée entre deux rouleaux qui se frottent, l'un est fixe, l'autre est muni d'un mouvement de rotation autour de son axe.

Ce mouvement de rotation est engendré par un système d'engrenage entraîné par un vérin à double effet (Z_2).

Le vérin (Z_2) est commandé par une électrovanne (Y_{2a} (4/2)), qui fonctionne suivant les signaux délivrés par les capteurs ($S_1, S_4, S_5 \dots$ etc.).

La figure I.19 illustre les différents éléments qui composent le système de la fonction avance du papier :

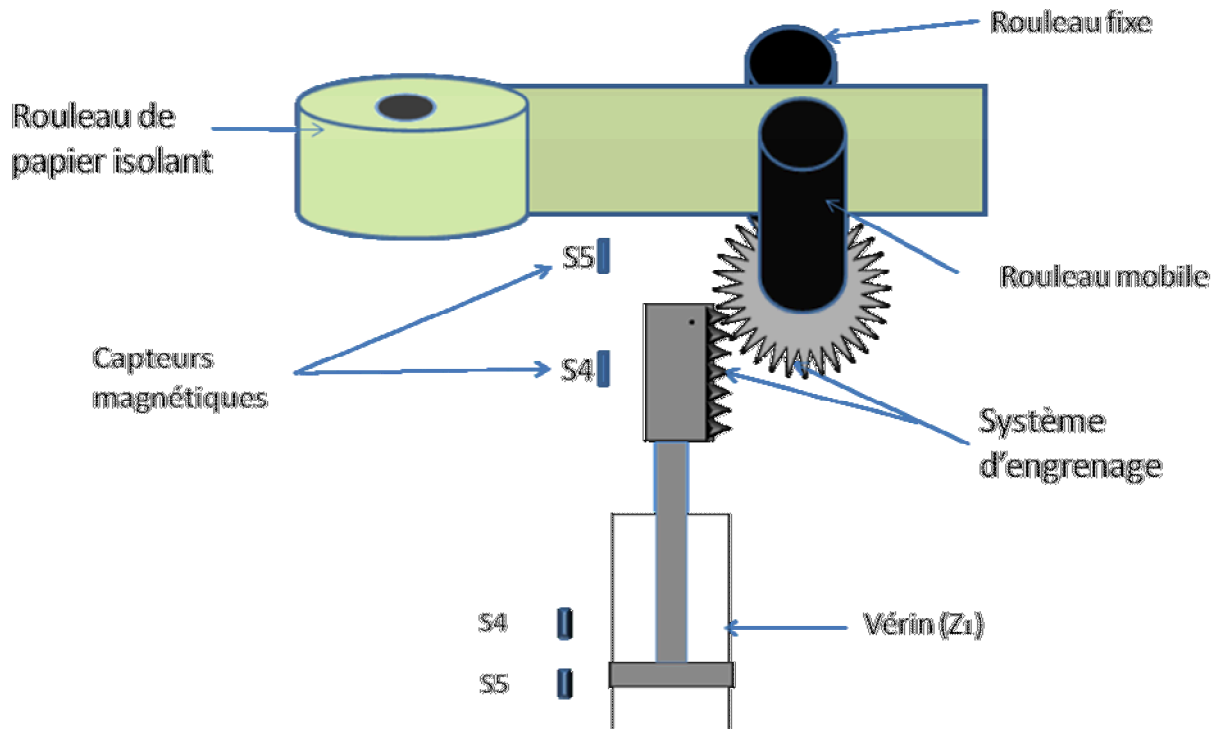


Figure I.17 : dispositif assurant la fonction avance du papier

I -5-4 Rotation du plateau :

Un fois l'une des encoches est isolée, on doit faire tourner le stator afin d'aligner l'encoche suivante sur l'axe du caniveau. Et pour se faire le stator est placé sur un plateau fixé sur une roue d'entrée, capable de tourner autour de son axe.

Ce mouvement de rotation est géré par un système de trois vérins à double effet (Z_1, Z_5, Z_6), dont les fonctions individuelles sont les suivantes :

Vérin Z_4 : c'est lui qui engendre le mouvement de rotation du plateau, la course de son piston est réglable, adaptable aux différents stators (nombre d'encoches, diamètre extérieur, ...etc.)

Vérin Z_5 : ce vérin est fixé au piston du vérin Z_4 , son piston est muni d'une cale en forme de (V). Son rôle est d'encliqueter la roue dentée du plateau afin qu'elle puisse être entraînée par le piston du vérin de rotation.

Vérin Z_6 : il a pour rôle d'indexer (de bloquer) le plateau lorsque la fonction d'enchâssement de la gaine se fait, et ainsi éviter que le couteau ne percute l'une des dents statorique si celui-ci venait à bouger.

Les vérins (Z_4, Z_5, Z_6) sont commandés respectivement par les électrovannes ($Y_{4a}, Y_{5a}, Y_{6a}, (4/2)$). Ces derniers fonctionnent suivant les signaux délivrés par les capteurs : ($S_7, S_8, S_9, S_{10}, S_{11}, S_{12}, S_{13}, S_{14}$).

La figure I.20 illustre la disposition des différents vérins de la partie supérieure de la machine, ainsi que l'allure de l'outil associé à chacun d'eux :

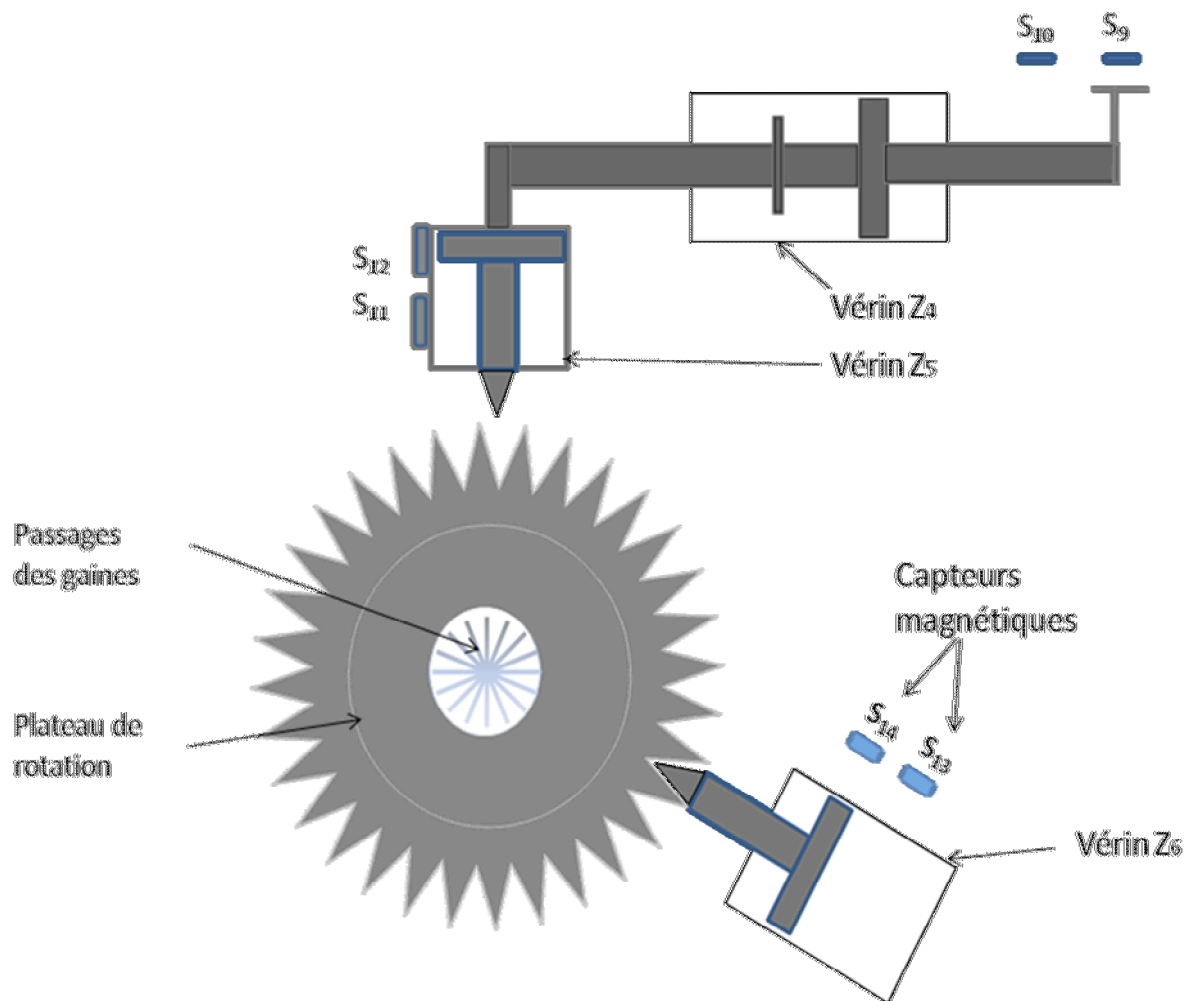


Figure I.20 dispositif assurant la rotation du plateau de division

Conclusion :

L'étude du présent chapitre nous a mis en contact avec la technologie de la machine ; son mode de fonctionnement ainsi que les différents éléments qui la constituent. Et cela dans le but de faciliter le processus de modélisation que nous allons étudier dans le chapitre suivant

II -1 Introduction au GRAFCET

Pour être analysé communiqué et compris, le fonctionnement d'un système automatisé doit être modélisé. Selon les moments de la vie du système les modèles sont différents pour répondre à des besoins précis et variés (dessins mécanique, algorithmes, etc...).

Bien que l'utilisateur, qui doit comprendre ou modifier les modèles initiaux à des fins de conduite ET/OU de maintenance, n'ait pas le même objectif que le concepteur, il ne peut en règle générale, découvrir et appréhender le fonctionnement du système automatisé qu'à partir de documents réalisés en phase de conception, il est donc essentiel que les méthodes et outils utilisés permettant de véhiculer les informations élaborées au long du cycle de la vie du système.

Le fonctionnement d'un système automatisé peut être représenté par un ensemble appelé **GRAFCET**. [13]

II -2 Historique

Le **GRAFCET** est né en 1977 des travaux de l'**AFCET** (Association Française pour la Cybernétique Economique et Technique), en tant que synthèse théorique des différents outils existant à cette époque (organigramme, diagramme de Gérard, etc...). Mais sous sa forme graphique actuelle l'**ADEPA** (Agence pour le **D**Éveloppement de la **P**roduction **A**pplique à l'industrie) en 1979 normalisé sur le plan Français en 1981 (norme NFCO3-190), il est aujourd'hui normalisé sur le plan national (norme CEI 848). [13]

II -3 Définition

Le **GRAFCET** (**G**raphe de **C**ommande **E**tape **T**ransition) est un diagramme fonctionnel dont le but est de décrire graphiquement, suivant un cahier des charges, les différents comportements et l'évolution d'un automatisme séquentiel. Il est à la fois simple à utiliser et constitue un outil de dialogue entre toutes les personnes collaborant à la conception, à l'utilisation et à la maintenance de la machine à automatiser. [13]

II -4 Cahier de charges

Le cahier des charges d'un automatisme est une description fonctionnelle faite par l'utilisateur, elle analyse le comportement de la partie commande vis-à-vis de la partie opérative permettant au concepteur de comprendre ce que l'automatisme doit faire aux différentes situations pouvant se présenter.

II -5 Les éléments de base du GRAFCET

Un GRAFCET est composé d'étapes, de transitions et de liaisons

La figure II.1 illustre les différents symboles qui caractérisent le GRAFCET : [13]

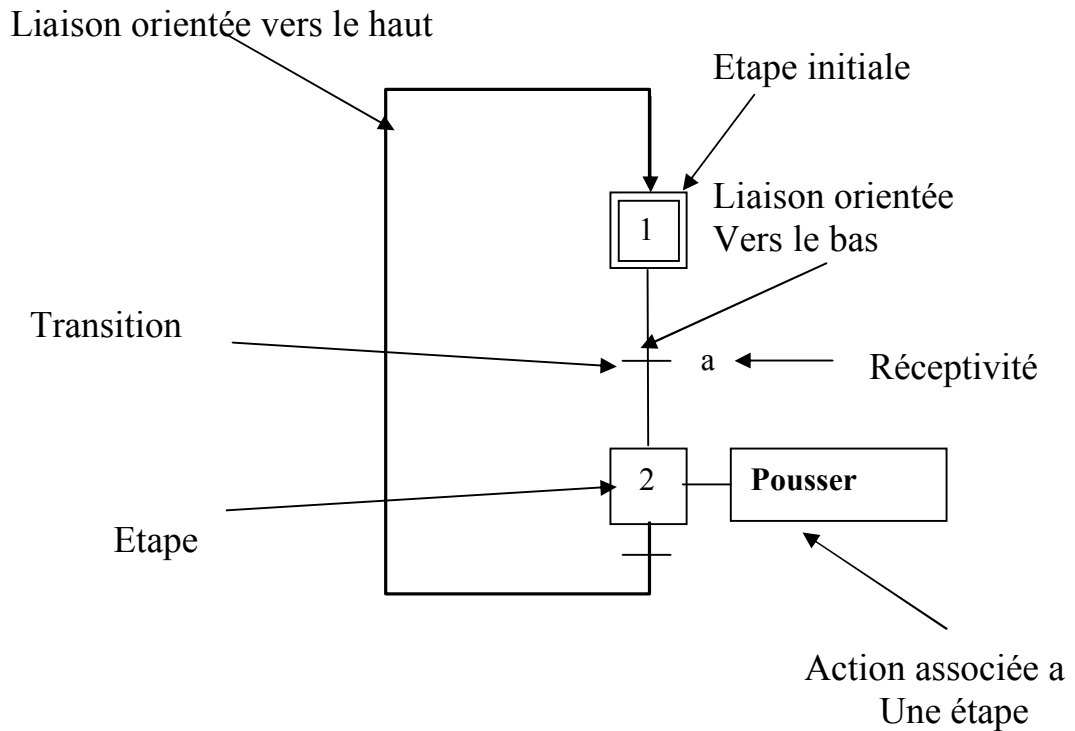


Figure II-1 la symbolisation du GRAFCET

II -5-1 Les étapes

Une ETAPE correspond à une phase durant laquelle on effectue une ACTION pendant une certaine durée. L'action doit être stable, c'est à dire que l'on fait la même chose pendant toute la durée de l'étape. Lorsque l'étape est active, on le précise en y ajoutant un point.



Figure II-2 Etape active (a) / Etape inactive (b)

II -5-1-1 Étape initiale

L'étape initiale représente l'étape active du système au début du fonctionnement. Elle se différencie en doublant les cotées du carré



Figure II.2.1 étape initial

II -5-1-2 Les actions associées à une étape

A chaque étape sont associées une ou plusieurs actions. Ces dernières réalisent ce qui doit être fait chaque fois que l'on active les étapes auxquelles elles sont associées. Ces étapes peuvent être externes (sortie de l'automate pour commander le processus) ou internes (temporisation, comptage, etc...).

Une étape peut n'avoir aucune action,

Exemple : attente d'un événement

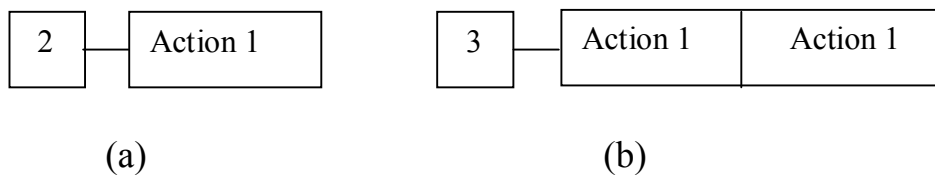


Figure II-3 Etape simple (a) / Etape a plusieurs actions (b).

Il est possible de définir plusieurs types d'action :

II -5-2 Les transitions et les réceptivités associées

II -5-2-1 Les 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) et sans notion de durée. A chaque transition est associée une réceptivité.

On représente une transition par un trait vertical.

II -5-2-2 Réceptivité

La réceptivité est une condition logique qui doit être remplie pour le franchissement de la transition. Elle est représentée à droite de la barre représentant la transition.

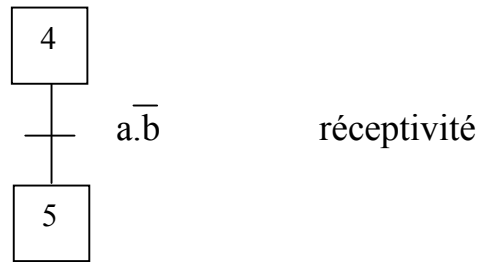


Figure II-4 réceptivité associée à l'étape

➤ **Temporisation**

La temporisation est une réceptivité qui permet une prise en compte du temps. Elle implique l'utilisation d'un temporisateur. Ce type de réceptivité est noté comme suite : $T/Xi/Q$ où i est le numéro de l'étape comportant l'action de temporisation, et Q est la durée écoulée depuis l'activation de l'étape Xi

➤ **Réceptivité toujours vraie**

La réceptivité toujours vraie est une réceptivité dont la valeur logique est toujours « =1 », le franchissement de cette transition se fera dès que les étapes précédentes sont actives

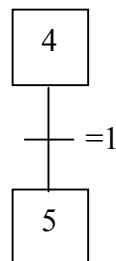


Figure II-5 réceptivité toujours vraie

➤ **Réceptivité à niveau**

La réceptivité à niveau est une réceptivité faisant intervenir une condition logique.

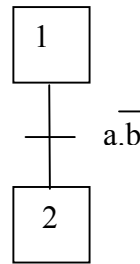


Figure II-6 réceptivité à niveau

II -5-3 Liaisons orientées

Une LIAISON est un arc orienté (ne peut être parcouru que dans un sens). A une extrémité d'une liaison il y a une seule étape, à l'autre une transition. On la représente par un trait plein et rectiligne, vertical ou horizontal.

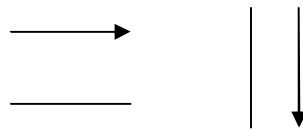


Figure II-7 : Liaisons orientées

II -5-3 Règles d'évolution du GRAFCET

La modification de l'état de l'automatisme est appelée évolution, et est régie par 5 règles : [13]

- **Règle 1 : (situation initiale)**

La situation initiale d'un GRAFCET caractérise le comportement initial de la partie commandée vis-à-vis de la partie opérative, elle correspond aux étapes actives au début du fonctionnement.

- **Règle 2 (franchissement d'une transition)**

Une transition est soit validée, soit non validée. Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives.

Elle ne peut être franchie que lorsqu'elle est validée et que sa réceptivité est vraie. Elle est alors obligatoirement franchie.

- **Règle 3 (évolution des étapes actives)**

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

- **Règle 4 (évolution simultanée)**

Plusieurs transitions simultanément franchissables sont simultanément franchies.

- **Règle 5**

Si au cours du fonctionnement une étape doit être à la fois activée et désactivée, elle reste active.

II -5-4 Erreurs de GRAFCET

➤ **Erreurs de fond**

Deux étapes ou deux transitions ne peuvent pas se suivre. Une divergence en OU est toujours fermée par une convergence en OU. Il est de même pour la divergence en ET.

➤ **Erreurs de forme**

Pour éviter ces erreurs deux règles sont à retenir :

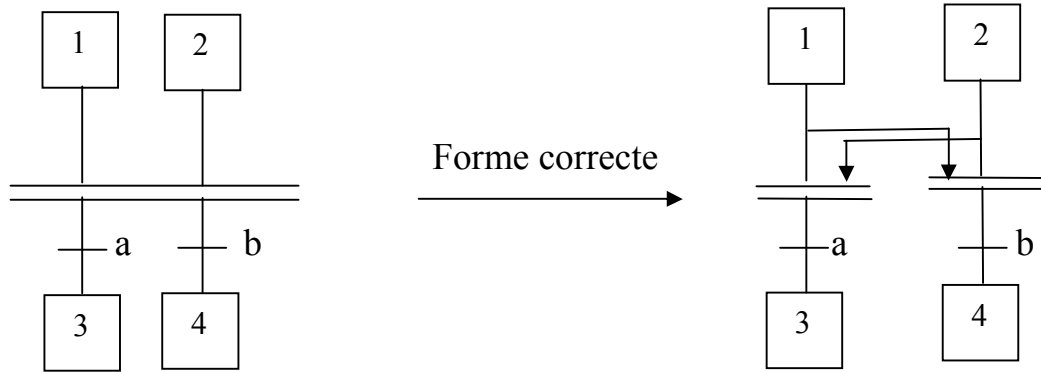
- **Règle 1 :**

Aucun accès aux symboles (étape ou transition) n'est autorisé en dehors du regroupement de liaisons prévues.

- **Règle 2 :**

Les regroupements de liaison (simple ou double trait) ne peuvent être ni isolés des symboles, ni communs à plusieurs symboles.

Exemple



II -5-5 Sélection de séquence et séquences simultanées

Le modèle GRAFCET présentes deux structures particulières, la sélection de séquence et les séquences simultanées

II -5-5-1 Sélection de séquence

La sélection de séquence dans un GRAFCET permet de choisir une suite une suite d'étapes plutôt qu'une autre. Cette structure est composée d'une seule étape en amont de plusieurs transitions en aval qui permettent le choix de la séquence. Elle se représente à l'aide d'un simple trait horizontal. La fin d'une sélection de séquences permet la reprise d'une séquence unique

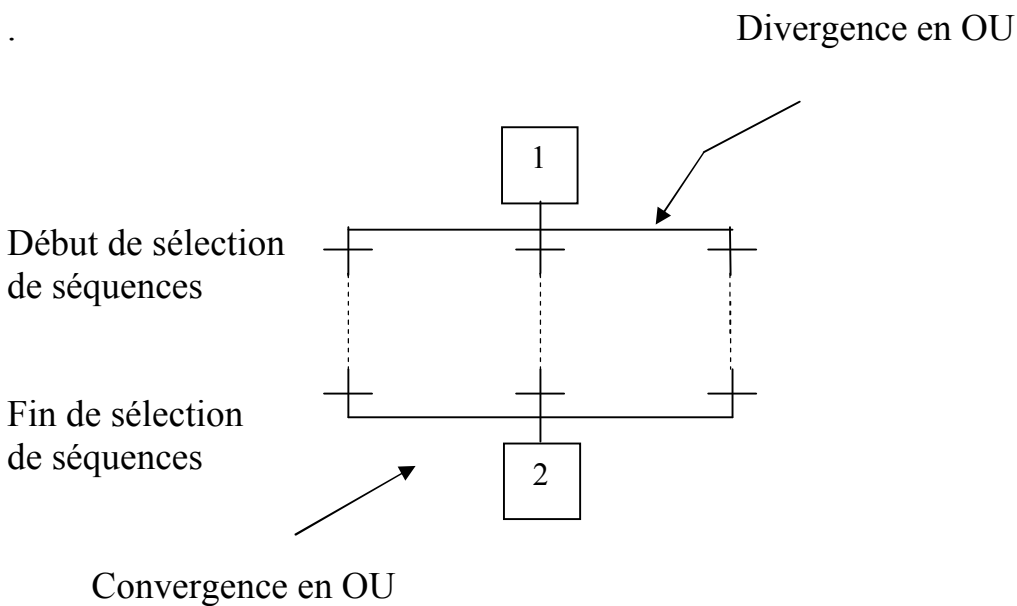


Figure II-8 représentation graphique de la sélection de séquence

II -5-5-1-1 Saut d'étape

Les sauts d'étapes permettent de sauter plusieurs étapes en fonction des conditions d'évolution. Le saut d'étape comprend au minimum le saut d'une étape. Dans l'exemple ci-dessous, si la réceptivité « a.b » est vraie alors les étapes 5 à 8 vont être sautées.

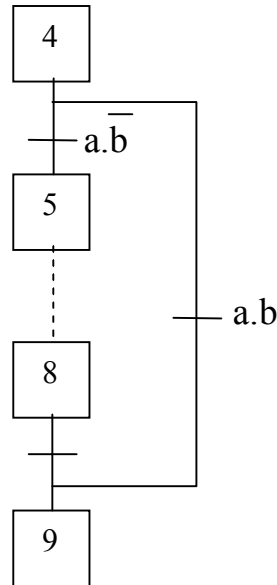


Figure II-9 saut d'étape

II -5-5-1-2 Reprise d'étape

La reprise d'étapes au contraire permet de recommencer plusieurs fois si nécessaire la même séquence. La reprise des séquences doit comporter au moins trois étapes puisque l'activation d'une étape comporte la désactivation de l'étape précédente et la validation de l'étape suivante.

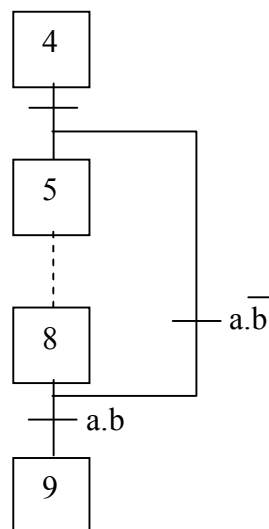


Figure II-10 Reprise d'étape

II -5-5-2 Séquences simultanées

Lorsque l'on souhaite réaliser plusieurs séquences simultanées, on utilise cette structure. Elle est composée d'une seule étape et d'une seule transition en amont qui permet de déclencher simultanément plusieurs séquences d'étape. Elle est représentée à l'aide d'un double trait horizontal.

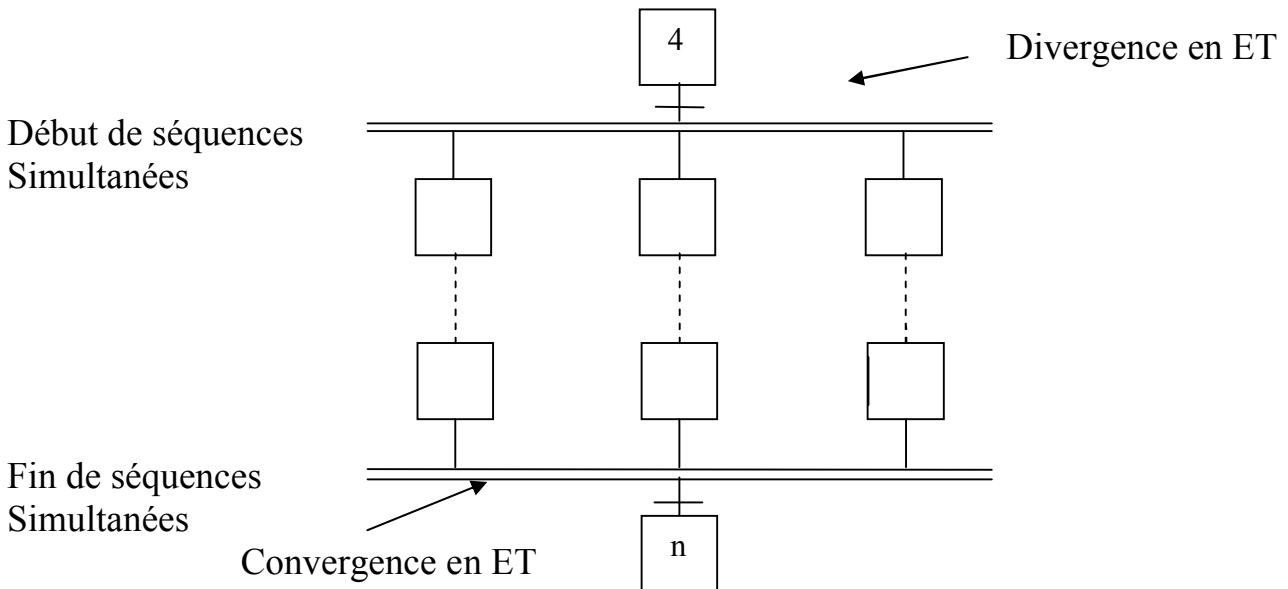


Figure II-11 représentation de séquences simultanées

II -6 Niveau d'un GRAFCET :

II -6-1 GRAFCET du niveau 1 :

Ce GRAFCET décrit sous forme d'actions fonctionnelles le comportement de la partie Opérative pour obtenir les actions désirées. Dans ce GRAFCET le système sera décrit sous forme littérale, sans tenir compte de la technologie utilisée. Il est souvent utilisé pour vendre ou décrire un système. Il est l'outil idéal pour expliquer un système à des non professionnels ou établir un cahier des charges. [6]

II -6-2 GRAFCET du niveau 2 :

A ce niveau, le concepteur s'implique dans le fonctionnement de la partie commande. Le langage est codé. Il reçoit des informations et émet des ordres, le choix technologique est retenu.

Ce GRAFCET met en œuvre et décrit la partie opérative et s'adresse à des spécialistes. Il est utilisé pour la réalisation ou le dépannage des systèmes automatisés. [6]

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

II -7 Application du GRAFCET pour modéliser la machine

II -7-1 Préparation de la machine à garnir les encoches :

- Mise en place de l'outillage
- Réglage du nombre d'encoches du paquet statorique
- Réglage de l'avance de la gaine d'encoche
- Réglage de la longueur d'encoche
- Placer la bonde de papier isolant sur le dérouleur

La machine à garnir les encoches est à commande électro-pneumatique, les différents mouvements sont réalisés par des vérins pneumatiques à double effet, alimentés par des électro-distributeurs.

II -7-2 Liste des vérins et rôle attribué à chacun d'eux :

- Vérin Z1 (formage coupe) : à l'extrémité de ce vérin est soudé un outil dont le rôle est de découper et de donner la forme de l'encoche à la gaine de papier isolant (bande de papier isolant). Et cela en l'enfonçant dans un caniveau de mêmes dimensions internes que celles de l'encoche à isoler, et situé au dessous du plateau.
- Vérin Z2 (avance du papier) : la bande de papier passe entre deux rouleaux qui se frottent, l'un est fixe et l'autre est animé d'un mouvement de rotation autour de son axe. la cause de ce mouvement de rotation est attribuée au vérin Z2, qui lors de l'éjection de son piston, entraîne avec lui une roue dentée sur laquelle est fixé ce rouleau.
- Vérin Z3 (enchâssement) : Il est placé sous le caniveau et il est doté d'un mouvement vertical, à l'extrémité de son piston on a placé un couteau dont le rôle est d'insérer (enchâsser) la gaine isolante dans l'encoche.
- Vérin Z4 (rotation du plateau) : Il a comme rôle, de faire tourner le plateau sur lequel est fixé le stator, d'un angle correspondant au pas du stator (distance entre deux encoches successives). A l'extrémité du piston de ce vérin est monté un autre vérin (Z5).

- Vérin Z5 (encliquetage) : le plateau sur lequel est fixé le stator est muni d'une roue dentée, le rôle du vérin Z5 est d'encliqueter cette dernière afin qu'elle soit entraînée par le mouvement du vérin Z4
- Vérin Z6 (indexage) : son piston est muni d'un outil en forme de cale (V), son rôle, est de fixer le plateau sur lequel est mis le stator, lorsque les encoches reçoivent les gaines de papier isolant.

II -7-3 Capteurs et bouton poussoirs

- **Capteurs de fin de course magnétique**

S1 : vérin Z₁ en arrière
 S2 : vérin Z₁ en avant
 S3 : vérin Z₁ avancé de 3/4 de sa course
 S4 : vérin Z₂ en avant
 S5 : vérin Z₂ en arrière
 S6 : vérin Z₃ en avant
 S7 : vérin Z₃ en arrière
 S8 : vérin Z₃ avancé de 3/4 de sa course
 S9 : vérin Z₄ en arrière
 S10 : vérin Z₄ en avant
 S11 : vérin Z₅ en avant
 S12 : vérin Z₅ en arrière
 S13 : vérin Z₆ en arrière
 S14 : vérin Z₆ en avant

- **Autres capteurs**

P : capteur de pression

- **Boutons poussoirs**

AUS : bouton poussoir d'arrêt d'urgence desserré
 EN : bouton poussoir de mise en service

- **Autres boutons**

AUT : sélecteur sur position automatique
 EIN : sélecteur sur position pas à pas
 C : impulsion du décompteur lorsque il est incrémenté

II -7-4 Abréviations utilisées dans le GRAFCET de niveau 2

Y1a : l'électrovanne qui commande le vérin Z1

Y2a : l'électrovanne qui commande le vérin Z2

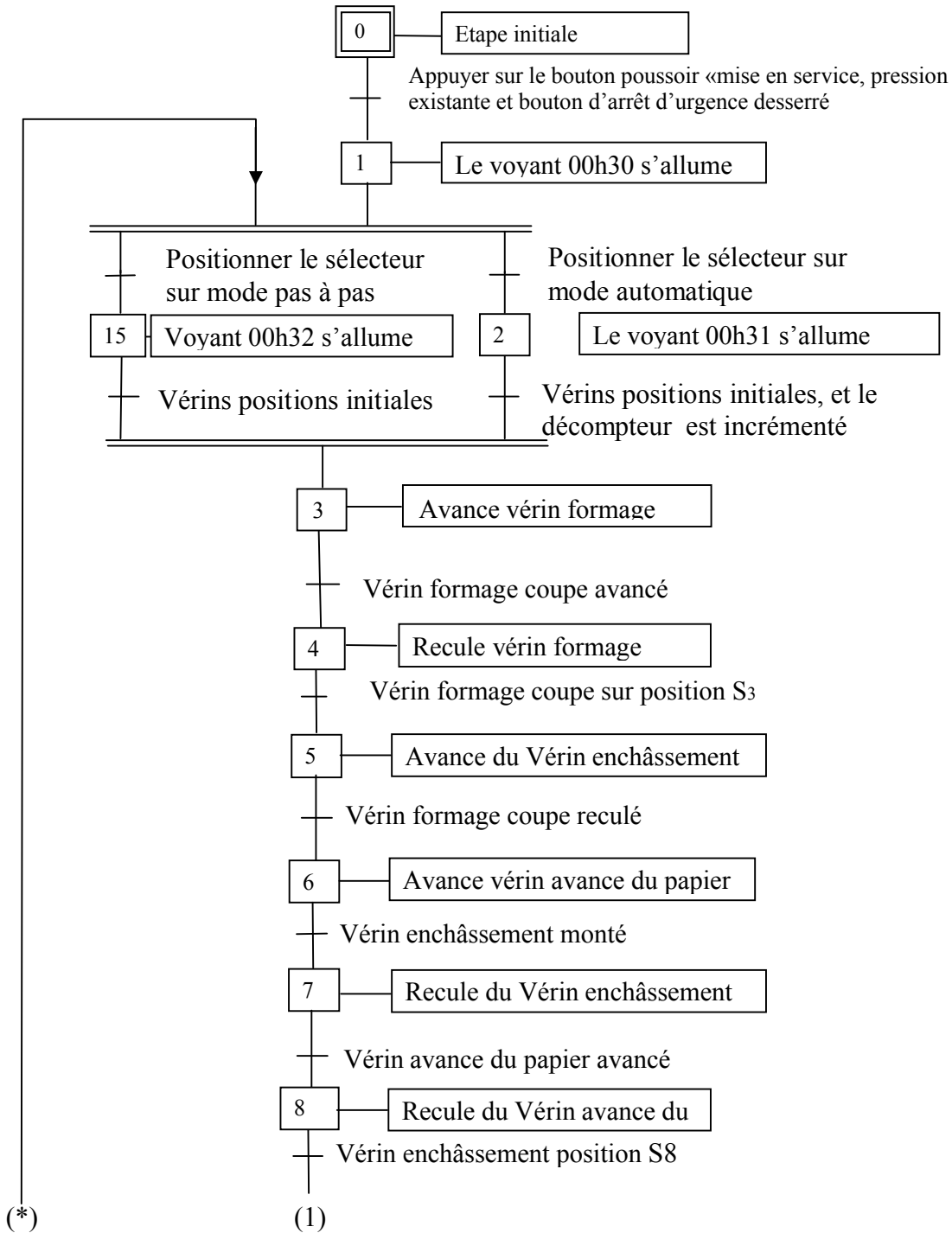
Y3a : l'électrovanne qui commande le vérin Z3

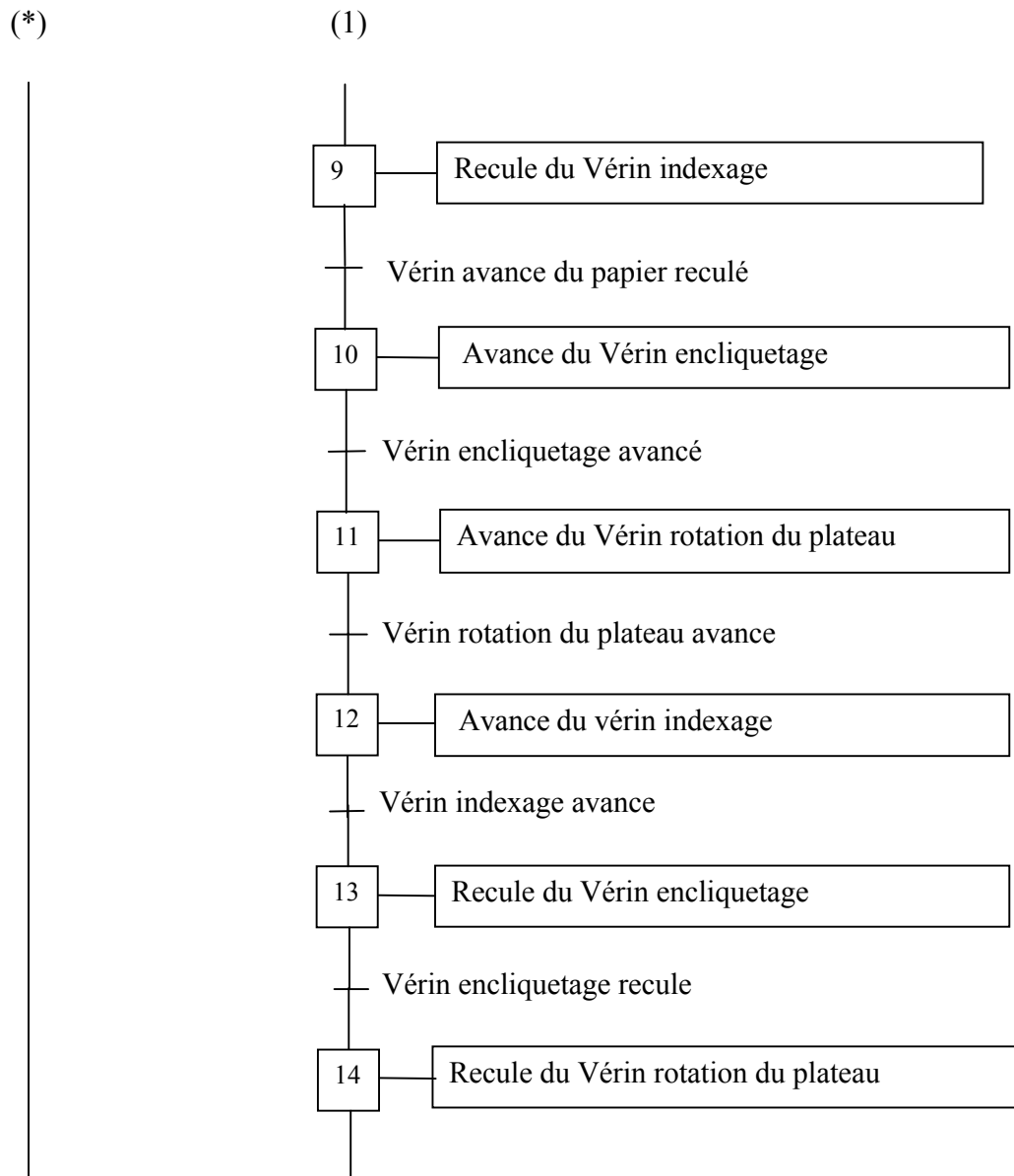
Y4a : l'électrovanne qui commande le vérin Z4

Y5a : l'électrovanne qui commande le vérin Z5

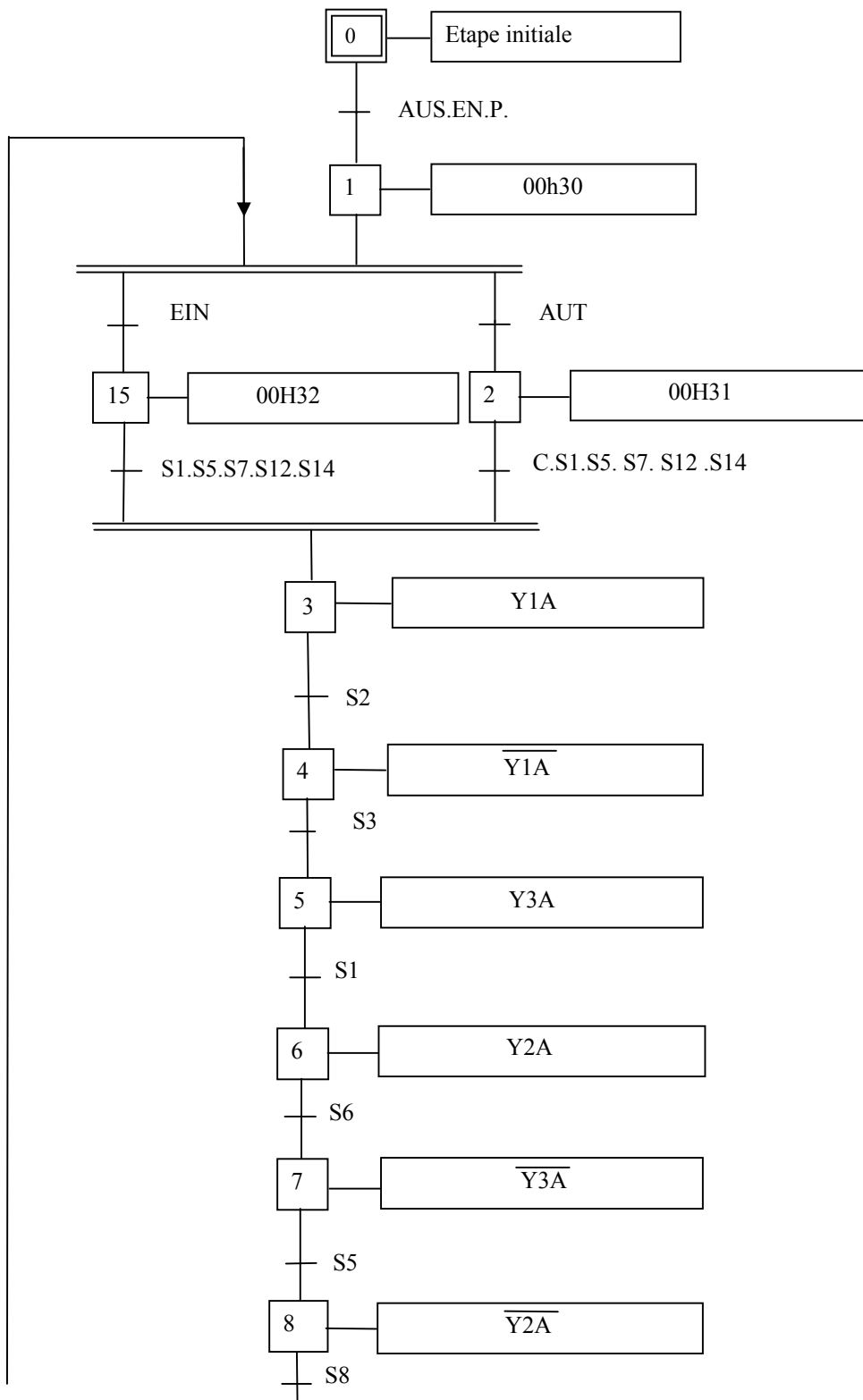
Y6a : l'électrovanne qui commande le vérin Z6

GRAFCET de niveau 1





GRAFCET de niveau 2

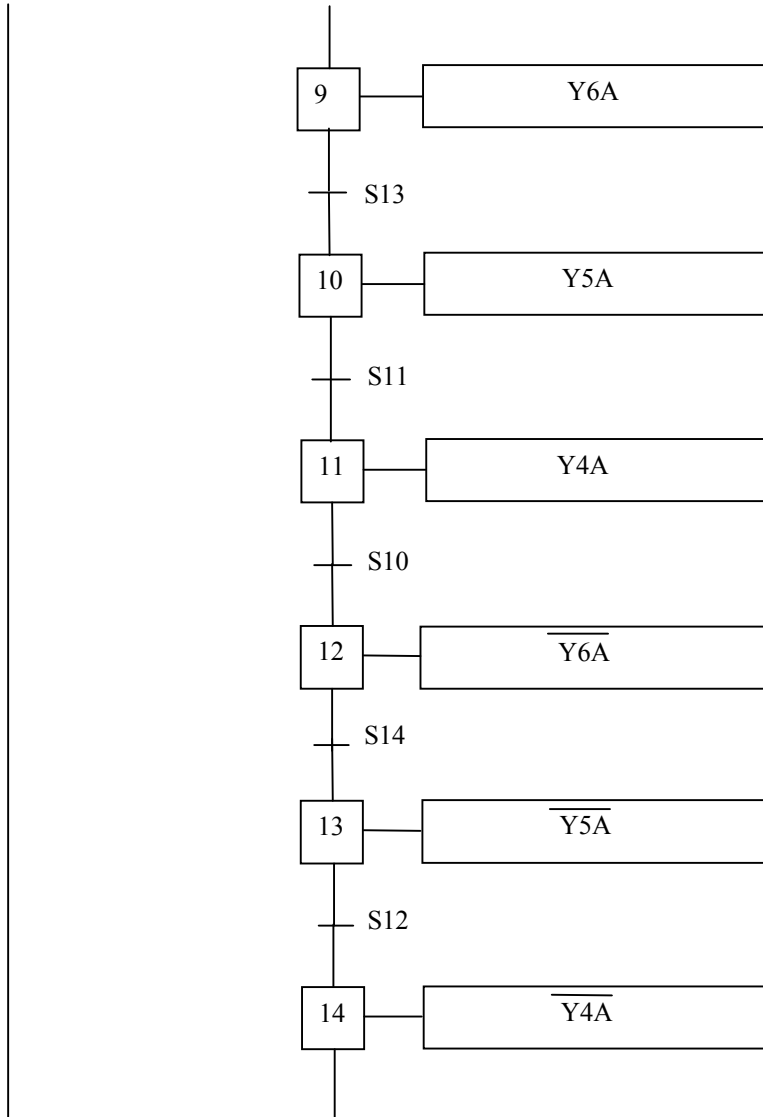


(*)

(1)

(*)

(1)



VIII Conclusion :

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

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

Ainsi, le GRAFCET a facilité considérablement le passage de la description à la modélisation et nous permettra au chapitre suivant d'aborder la programmation de la partie opérative qui pilotera le procédé et ce à l'aide du STEP 7.

III -1 Introduction :

Le recours à l'amélioration de la technique d'automatisation s'avère quelque chose d'inévitable, sachant les conséquences sur le plan industriel et recherche, c'est bien ce qui c'est produit, vers la fin des années 60, où on a assisté à un développement continu de l'industrie automobile, engendrant différents modèles de voitures, ceci a impliqué des transformations des chaînes de montage et de leurs armoires de commande, étant donné que ces dernières sont réalisées à base de relais, avec une complexité pouvant provoquer des risques élevés en cas d'erreurs de branchement.

Enfin, l'idée qui s'est imposée est de créer un appareil programmable permettant de remplacer les armoires de commande, un dispositif appelé Automate Programmable Industriel

III -2 Définition :

Un automate Programmable industriel est un dispositif programmable dessiné à piloter des procédés ou parties opératives, et adaptable à un maximum d'applications, du point de vue traitement, composants, langages.

Les API remplacent progressivement les équipements câblés, et grâce à leur structure, ils permettent de modifier simplement les séquences d'un processus, sans introduire les contraintes de câblage.

III -3 Place des automates programmables :

Grâce à ses différents avantages, voir sa souplesse d'utilisation, son adaptation à tous les milieux industriels, ainsi que son encombrement réduit, l'API est actuellement le moyen le plus utilisé dans la commande des systèmes automatique de production (SAP).

Aujourd'hui, tout les secteurs industriels font recours aux automates, citons :

- l'industrie automobile
- construction mécanique
- transformation des matières plastiques

Ce dispositif joue un rôle de commande, assurant des actions, en fonction d'un programme approprié à partir des informations recueillies (détecteurs, capteurs) et dans le processus de production, l'API joue un rôle de communication, voir un dialogue d'exploitation avec des opérateurs humains, ou encore des connexions avec d'autres automates participants dans la chaîne de production.

III -4 Mode de fonctionnement :

L'API fonctionne suivant un programme, il reçoit les informations des dispositifs d'entrée, et prend des décisions sur les actions à exécuter (activer ou désactiver des sorties).

Principalement son rôle consiste à fournir à la partie opérative des instructions afin d'exécuter un travail précis,

Exemple : la sortie ou la rentrée d'une tige de vérin.

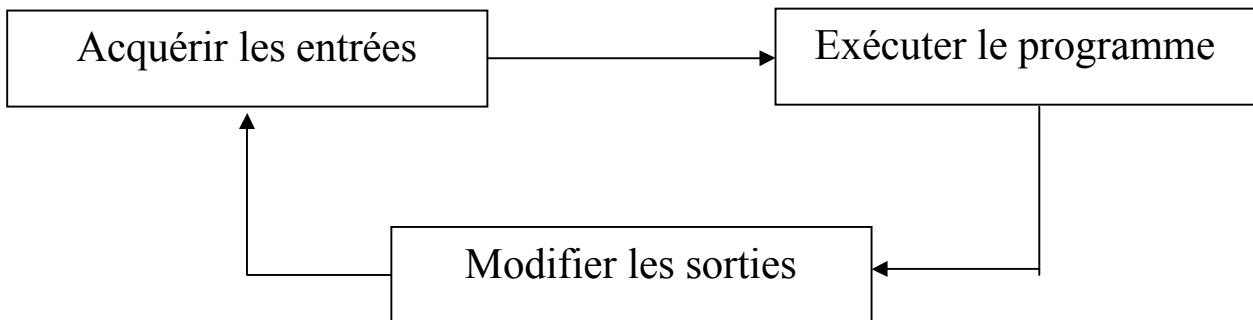


Figure III.1 : cycle de fonctionnement d'un API

III -5 Architecture interne d'un automate programmable :

Un automate est constitué principalement dans sa partie matérielle par :

- les entrées auxquelles sont accordés les capteurs, qui fournissent les informations relatives à l'équipement ou au procédé à commander ou à contrôler
- l'unité arithmétique et logique, ou unité de traitement qui réalise les instructions logiques (ET, OU...) et arithmétiques (addition, comptage).
- Une console de programmation, intégrée ou non à la machine, servant à introduire les instructions « utilisateur » en instructions exécutables par l'automate.
- Les sorties qui émettent les ordres élaborés par l'unité de traitement vers les actionneurs (contacteurs, électrovannes...etc.).
- L'alimentation ainsi que les divers circuits annexes.

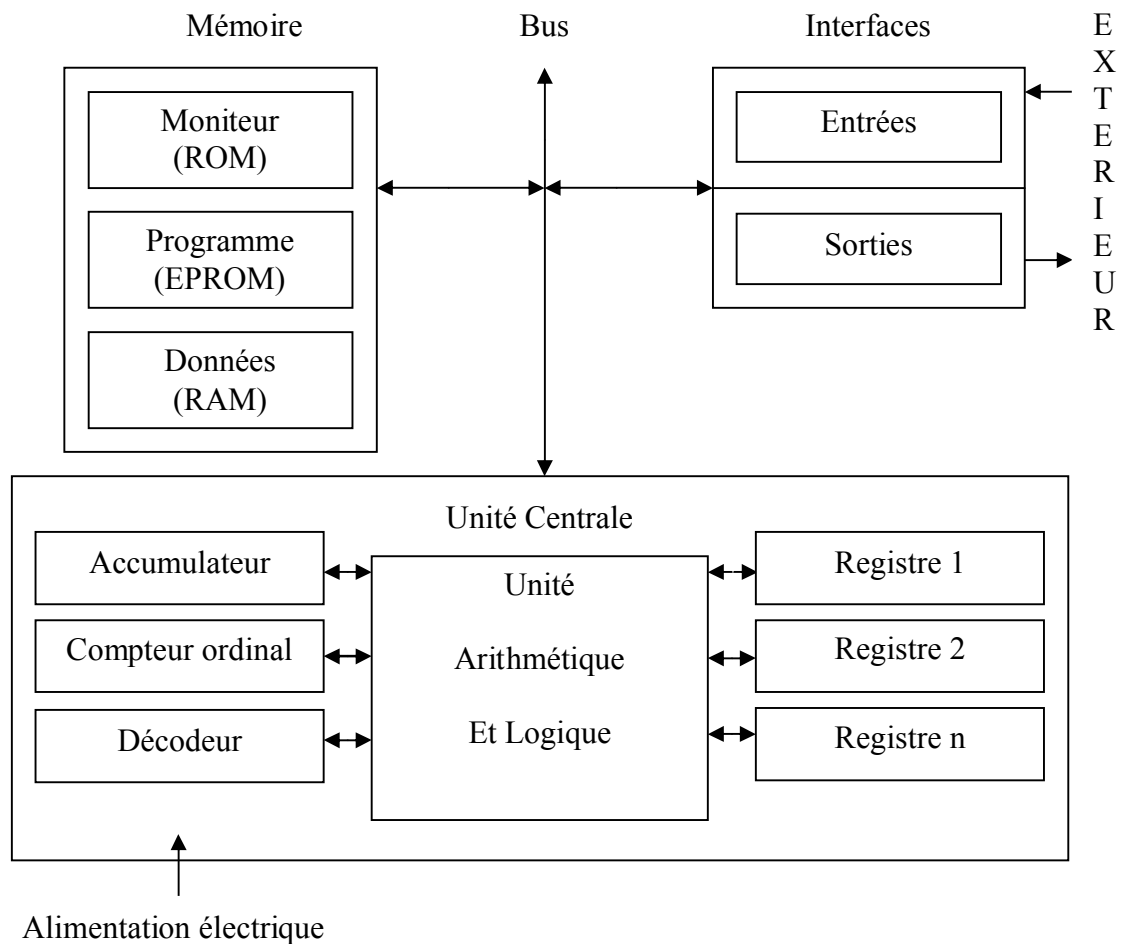


Figure III.2 : Architecture interne d'un API.

III -5-1 Module d'alimentation :

Permettant à l'automate de recevoir l'énergie électrique nécessaire à son fonctionnement, à partir d'une alimentation en 220V alternatif.

Ce module délivre des tensions dont l'automate à besoin (5v, 12v, 24v) continu, et une mise à la terre doit être prévue.

La mise sous tension de l'API est indiquée par un voyant positionné sur la façade.

III -5-2 Unité centrale de traitement (CPU) :

Elle représente le regroupement du processeur et de la mémoire centrale, commande l'interprétation et l'exécution des instructions programmes, ces dernières sont effectuées les unes après les autres, séquencées par une horloge.

Si deux actions doivent être simultanées, l'API les traite successivement.

L'unité centrale est caractérisée principalement par la vitesse de traitement, voir le temps de réponse (scrutation des entrées, vitesse de traitement et affectation des sorties).

Elle se compose de deux éléments qui sont : Le processeur et la mémoire.

III -5- 2-1 Processeur :

Il Assure une bonne commutation entre la zone mémoire et les interfaces d'entrées et de sorties, et génère les instructions du programme.

Il est composé des éléments suivants :

- Unité logique (UL), qui traite les opérations logiques ET, OU...
- Unité arithmétique et logique (UAL) traitants les opérations de temporisation de comptage et de calcul
- Un accumulateur, représentant le registre de travail, dans le quel se range une donnée ou un résultat.
- Un registre d'instructions, qui contient durant le temps de traitement, l'instruction à exécuter.
- Un décodeur d'instructions, qui décode l'instruction à exécuter en y associant le microprogramme de traitement.

III -5-2-2 Mémoire

Une mémoire est conçue sous forme d'un circuit électrique complexe, permettant de stocker des données binaires, ces dernières peuvent être lues par l'automate.

On distingue deux types de mémoires :

- Une mémoire volatile dont le contenu peut être effacé ou modifié facilement, l'exemple des mémoires programmables (EPROM).
- Une mémoire non volatile, avec un contenu qui ne peut être ni effacé, ni modifié, elle porte toutes les instructions permettant d'interroger les modules d'entrées, et avoir des données sur l'état du procédé, ainsi que l'interprétation et l'exécution des instructions provenant de la console de programmation, et la transmission des ordres aux modules de sorties, cette mémoire est appelée la mémoire morte (ROM).
- La mémoire vive (RAM), nécessitant une batterie de sauvegarde pour éviter de perdre son contenu au cas de coupure du courant.

III -5-3 Modules d'entrées :

Ils Permettent à l'automate d'acquérir les informations provenant soit du pupitre de commande, ou bien des capteurs et détecteurs.

Des diodes électroluminescentes sur le module indiquent l'état de chaque entrée, et facilitent graduellement le contrôle du bon fonctionnement des dispositifs d'entrées et leur raccordement.

III -5-4- Modules de sorties :

Offrent à l'automate la possibilité de commander les divers pré-actionneurs (contacteurs, électrovanne,...etc.), actionneurs (vérins, moteurs,...etc.), et éléments de signalisation.

Les tensions de sorties sont de 5v en contenu ou de 24, 48, 110, 220 volts en contenu ou en alternatif.

Les sorties peuvent être logiques, analogiques, ou bien numériques

III -5-5 Modules de communication :

Assurant un dialogue, d'une part entre l'opérateur et l'automate, et entre l'automate et d'autres automates, on peut les classer selon trois éléments principales.

a - Console de programmation :

Constituée d'un petit boîtier portant un clavier et un affichage simple, comme elle peut être aussi sous forme d'un ordinateur, avec écran cathodique et clavier, elle sert à programmer, visualiser et modifier l'état des entrées et sorties, ainsi que le contrôle et le diagnostic de l'API.

La console de programmation peut être débranchée, lors du fonctionnement automatique de l'API.

b - les boîtiers de tests :

Intervenant dans le procédé d'entretien, permettant de visualiser le programme, ou les valeurs de paramètres, parmi ses fonctions :

- affichage de la ligne de programme à contrôler.
- Visualisation de l'instruction (code opérande et adresse de l'opérande).
- Visualisation de l'état des entrées.
- Visualisation de l'état des sorties.

c - les Unités de dialogue en ligne :

Sous forme de boîtiers, comportant des touches de fonctions numériques, une visualisation, et un dispositif de sécurité, elles sont destinées aux personnels spécialistes du procédé et non de l'automate, et leur offre la possibilité d'agir sur certains paramètres :

- Forçage des entrées / sorties.
- Modification de constantes, temporisations et compteur.
- Exécution de partie de programme.
- Chargement de programme en mémoire à partir de disquettes ou disques.

III -6 Choix d'un API :

Afin de choisir un automate, on doit faire référence au cahier des charges et choisir un dispositif qui nous permettra de mieux suivre et gérer le procédé, et cela en considérant les éléments suivants :

Le matériel, le logiciel et les services annexes, ainsi que le nombre d'entrées, et de sorties du procédé à automatisé par l'API.

Suivant ces règles, les critères essentiels pour le choix d'un API sont :

- les compétences de traitement du processeur (vitesse, opération...etc.)
- la qualité du service après vente.
- Le type et le nombre des entrées/sorties nécessaires.

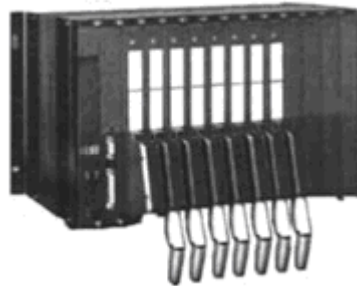


FIGURE : III.3 / FESTO FPC 405 : Architecture modulaire : carte de base, carte processeur, carte de motorisation, cartes E/S. programmation sur PC.

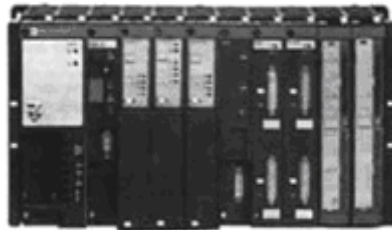


FIGURE : III.4 / TSX 47, nombre d'entrées et de sorties : 256



FIGURE : III.5 / C50 ou C100, nombre d'entrées/sorties variables

III -7 Les Avantages des API :

- L'API est flexible, du fait de sa programmation, la modification de sa tâche est facile, contrairement au cas des systèmes de commande à relais réels, dont toute modification nécessite l'ajout ou le retrait de relais, ainsi que la modification des raccordements ; et cette opération peut engendrer des erreurs de branchements qui peuvent provoquer des risques élevés
- Il est moins encombrant que l'armoire de commande à relais, une unité centrale de traitement d'environ 0,1 mètre cube peut remplacer des centaines de relais de commande, ainsi que tout le câblage qui relie leur contacts.
- Ne provoque pas de bruit (silencieux), et consomme beaucoup moins d'énergie.
- Le fait qu'il ne dispose pas de pièces mobiles, lui offre plus de fiabilité, or que la commande à relais peut voir des commandes erronées, causées par l'usage des pièces en mouvement, tel que l'oxydation et le soudage des contacts des relais.

III -8 les Inconvénients des API :

- L'API ne supprime pas tout le câblage, il reste celui de puissance.
- Sa vitesse d'exécution peut être insuffisante dans certains cas.
- L'ordre dans lequel on écrit le programme peut perturber le comportement de la commande, causé par des problèmes de type aléas de séquence.
- Une même fonction n'a pas nécessairement le même effet, d'un API à l'autre.

III -9 Programmation d'un API :

Écrire dans la mémoire, la description détaillée du travail à accomplir, l'API est conçue de manière à pouvoir être programmé facilement ; sans avoir recours à des connaissances informatiques.

III -9-1 Langage de Programmation :

Désigne essentiellement les différentes techniques et symboles utilisés, et la manière dont ils sont regroupés, afin de programmer l'API.

a - le Langage en Echelle :

Il est facile, comparé aux autres langages de programmation, son nom est inspiré d'une méthode graphique, servant à décrire les circuits à relais des armoires de commande, il est appelé aussi schéma à contact. (CONT).

Type de schéma en contact, opération en ET :

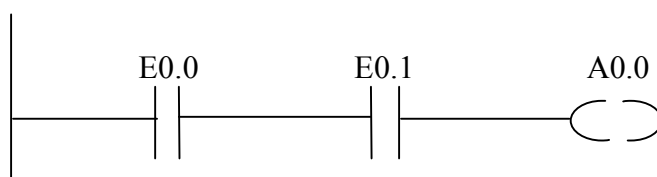


Figure III.6 : Exemples de représentation en langage contact

Type de schéma en contact, opération en OU :

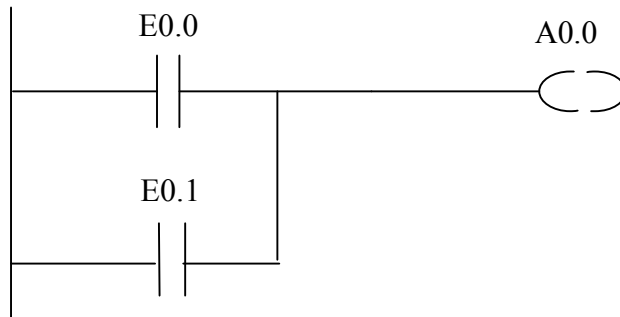


Figure III.7 : Exemples de représentation en langage contact

b - Logigramme (LOG) :

Il est inspiré de l’algèbre de boole ; il est facile de passer du diagramme en échelle au logigramme, mais l’inverse s’avère difficile.

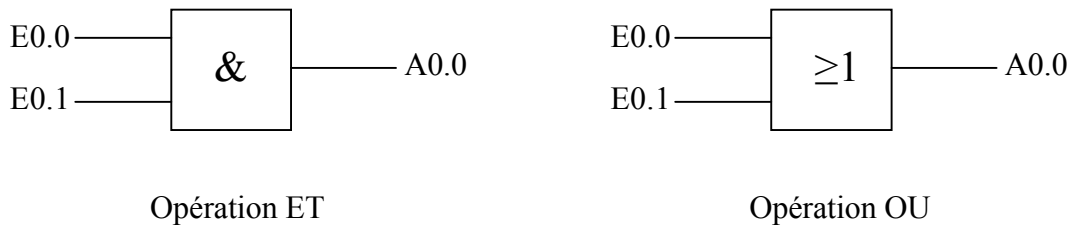


Figure III.8 : Exemples de représentation en logigramme

c - Langage Liste :

La représentation est telle que, l’instruction qui est l’unité de base du programme est constituée d’une opération et d’un opérande.

U E0.0	O E0.0
U E0.1	O E0.1
= A0.0	= A0.0

Figure III.9 : Exemples de représentation en liste d’instruction

III -10 Mise en Œuvre d'un API :

D'abord on doit bien avoir des données claires sur notre procédé d'automatisation :

A savoir :

- Les différentes commandes qui gèrent notre système d'automatisation.
- Définir le fonctionnement des capteurs, pré-actionneurs, ainsi que tous les organes de sorte

- Connaître le déroulement du processus à réaliser.

Toute fois la mise en marche de l'API, doit nous permettre :

- Une commande automatique de notre système.
- Un accès à l'exploitation, voir la possibilité d'effectuer des réglages, et différentes fonctions de maintenance.
- Une programmation, quant on envisage une première mise en œuvre.

La mise en œuvre suit les étapes suivantes :

- 1 - Etablir le grafctet correspondant au processus d'automatisation.
- 2 – Ecriture du programme, dont on peut procéder de deux manières, soit en écrivant le programme directement, ou bien en utilisant un logiciel d'assistance à la programmation

- 3 – Effectuer un transfert du programme dans l'unité centrale de l'automate.
- 4 - Teste a vide, présentant la mise au point finale du programme.
- 5 - Raccordement de l'automate à la machine.

Conclusion :

L'étude du chapitre nous à met en contact avec la technologie des automates programmables, leurs structures, fonctionnement et langages de programmation.

Au niveau du chapitre qui suit, on va s'intéresser à un exemple d'automate bien particulier, pour procéder à la programmation à base du logiciel step7.

Introduction :

Le SIMATIC S7-300 est un automate à extensibilité modulaire, utilisé particulièrement dans l'industrie manufacturière, Possédant une gamme régulière de CPU, qui offre un niveau très élevé de performances, avec une grande vitesse de traitement

Le SIMATIC S7 désigne un produit de la société SIEMENS.

IV – 1 Constitution de l'Automate S7-300 :

L'automate S7-300 possède :

- Des CPU de différents niveaux de performances
- Des Modules de signaux pour Entrées/Sorties « TOR » et analogique, ainsi que des Modules de fonction pour les différentes fonctions technologiques
- Une possibilité de mise à niveau par MPI
- Une largeur réduite des Modules, permettant un gain de place au montage
- Une structure compacte, lui permettant le placement aux milieux exigus.

IV -2 Caractéristiques de l'automate S7-300 :

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

- Gamme diversifiée de la CPU.
- Gamme complète du module.
- Possibilité d'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 S7-300 peuvent communiquer entre eux aux moyens d'un câble-bus PROFIBUS pour une configuration décentralisée.

IV –3 Modularité du S7-300 :

Parmi les caractéristiques essentielles du S7-300, le fait qu'il est disposé d'une vaste gamme de Modules.

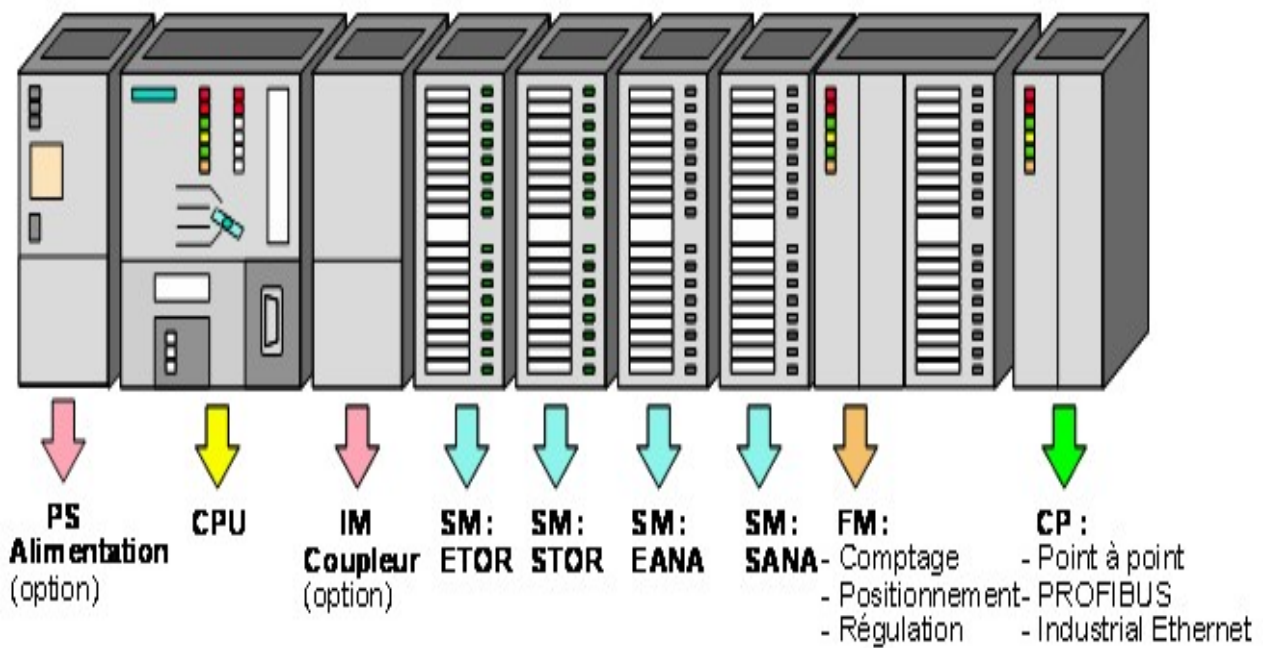


Figure IV-1 : Disposition des modules de l'API S7-300

IV –3-1 Unité Centrale (CPU) :

On a le choix entre plusieurs CPU, avec différentes performances, la CPU 314 représente des grandes performances.



Figure VI-2 : exemple de CPU S7-300

IV -3-1-1 Interface MPI :

Afin d'assurer une connexion de la console de programmation, ou un adaptateur PC par exemple, la CPU est équipée d'une Interface « MPI ».

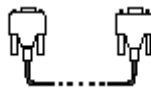


Figure IV-3 : Câble MPI

Le câble relie la PG (console de programmation) ou le PC à la CPU

IV -3-1-2 Signalisation des états :

Sur la face avant de la CPU, on trouve des LEDS permettant à l'automate de signaler certains états, tel que par exemple RUN :

- Etat de mise en fonctionnement
- Clignotement à la mise en route de la CPU

IV -3-1-3 Commutateur de mode de fonctionnement :

Il assure le changement du mode de fonctionnement, chaque position choisie autorise certaines fonctions bien particulières à la console de programmation, tel que RUN, STOP...etc.

IV -3-1-4 Carte Mémoire :

La CPU peut être équipée d'une carte mémoire, permettant la sauvegarde du programme en cas de coupure du courant, même en absence de pile.

IV -3-1-5 La pile :

Le contenu de la RAM peut être sauvegardé même en absence de courant, via la pile.

IV -3 -2 Modules d'alimentation :

Le S7-300 peut être alimenté avec une tension de 24 vcc, cette dernière est assurée via le module d'alimentation par conversion de la tension Secteur 380/220 vac

Les capteurs, actionneurs et voyants lumineux qui demandent plus de 24 v, sont alimentés par des blocs d'alimentation ou transformateurs supplémentaires.



Figure IV-4

IV -3 -3 Modules de signaux « SM » :

Assurant une liaison entre la CPU de l'automate S7-300, et le processus à commandé.

On à différent modules de signaux :

IV -3 -3-1 Modules d'Entrées/Sorties TOR :

Permettant à l'automate de recevoir des informations sur l'état des différents capteurs, boutons poussoirs....etc.

IV -3-3-2 Modules analogiques :

Si on a à faire à des capteurs et actionneurs analogiques, on doit faire appel à des modules d'entrées analogiques, assurant la conversion des signaux analogiques du système à commander en signaux numériques, qui peuvent être traités par l'API S7-300.

Toute fois la conversion des signaux numériques en signaux analogiques destinés au processus à commander est faite, via les modules de sorties.

Les modules d'Entrées/Sorties analogiques réalisent les deux fonctions, c'est-à-dire la conversion Analogique-Numérique, et Numérique-Analogique.

IV -3-3-3 Modules de simulation :

Pour valider le programme, et vérifier le bon fonctionnement de notre automatisme, on doit procéder à la simulation, ce que nous permettrons de tester le programme.

Au niveau du S7-300, le module de simulation offre différentes fonctions :

- La signalisation d'état des signaux de sorties par des LED
- La simulation des signaux des capteurs à base d'interrupteur

IV -3-3-4 Modules de communication :

Assurant une communication par transmission série, comme ils peuvent aussi établir des liaisons point à point avec d'autres automates SIMATIC S7-300, SIMATIC S5 ou bien d'autres constructeurs.

IV -3-3-5 Console de programmation :

Permet la saisie le traitement et la sauvegarde des données machines, et les données concernant le déroulement du processus.



Figure IV-5 : Console de programmation

IV - 4 Logiciel STEP7 :**IV -4-1 Introduction :**

Les automates programmables effectuent des tâches d'automatisation traduites sous forme de programme d'applications. L'utilisateur définit la manière dont l'automate doit commander l'installation par une suite d'instructions. Le programme doit être écrit dans un langage déterminé avec des règles définies pour que l'automate puisse l'exécuter, pour cela le logiciel le plus performant à l'exécution de ces tâches est le STEP7.

IV -4-2 Définition du STEP7 :

LE STEP7 est un logiciel de base pour la configuration et la programmation des systèmes d'automatisation. Il fait partie de l'industrie logicielle SIMATIC, il existe en plusieurs versions telles que : STEP micro/Dos et STEP7 micro/Win pour les applications S7-300 et S7-400

IV -4-3 Programmation de l'automate S7-300 :

La programmation des automates de la famille S7 se fait par la console de programmation ou par PC et sous un environnement WINDOWS. Via le langage de programmation STEP7.

Le logiciel STEP7 offre les fonctions suivantes pour l'automatisation :

- Configuration et paramétrage du matériel et de la communication.
- La création et la gestion des projets
- La gestion des mnémoniques.
- La création des programmes pour des systèmes cibles S7.
- Le chargement des programmes dans les systèmes cibles.
- Le test de l'installation d'automatisation
- Possibilité d'extension grâce aux applications proposées par l'industrie logicielle SIMATIC

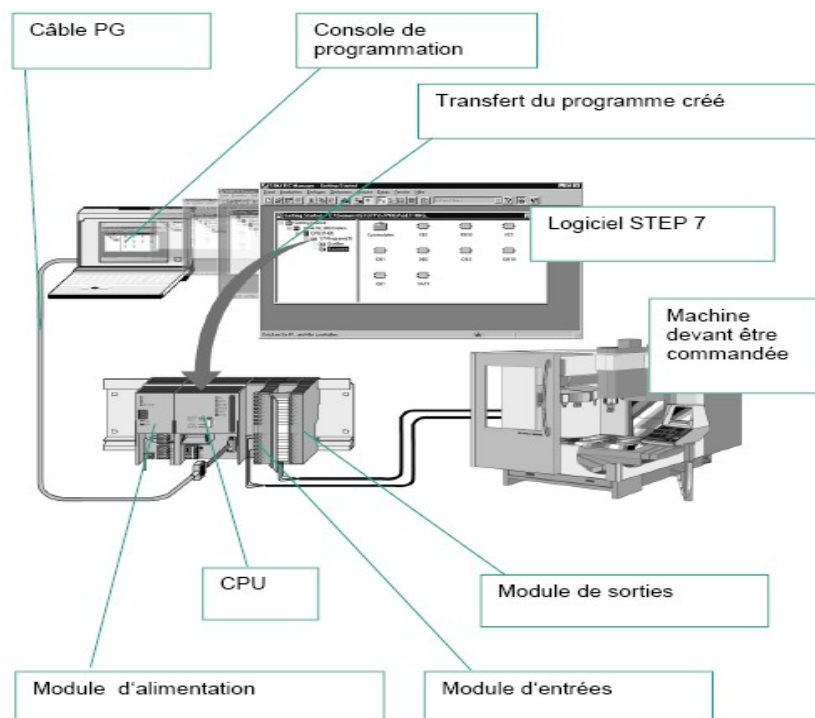


Figure IV.6: vue d'ensemble de l'automatisme.

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

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

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

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

IV -4-4 Adressage des modules du S7-300

On a deux types d'adressage :

- Adressage lié à l'emplacement : il s'agit du mode d'adressage par défaut, c'est-à-dire que le step7 effectue à chaque numéro d'emplacement une adresse de défaut de modèle fixé à l'avance.
- Adressage libre : dans ce mode d'adressage, il faut effectuer à chaque mode une adresse de votre choix, pourvu qu'elle soit continue dans la plage d'adresses possible de la CPU.

IV -4--.1 L'adressage des modules lié a l'emplacement :

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

IV -4-4-2 L'adressage absolu des modules de signaux :

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

a) Adressage des modules TOR :

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

- l'adressage d'octet dépend de l'adresse de début de module.
- l'adressage de bit est indiqué sur le module.

La figure (II-8) illustre un exemple d'adressage absolu ou direct.

b) Adressage des modules analogiques :

L'adressage d'une voie d'entrée ou sortie analogique est toujours une adresse de mot. L'adresse de la voie dépend de l'adresse de début de module. Si le premier module analogique occupe l'emplacement 4, l'adresse de début par défaut est 256. L'adresse de début de chaque module analogique suivant est incrémentée de 16 par emplacement. Les voies d'entrée et de sortie analogique d'un module d'E/S analogique ont la même adresse de début.

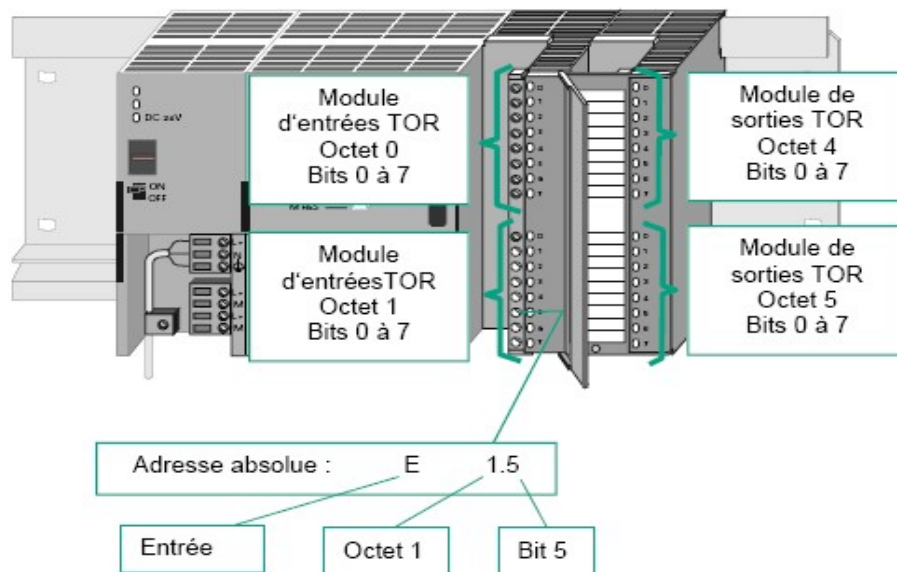


Figure IV.7 : exemple d'adressage absolu d'un module TOR.

IV -4 5 Mémentos :

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

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

IV -4 -6 Traitement du programme par la CPU :

Selon la manière de programmation choisie pour le problème d'automatisation, la CPU peut traiter le programme comme suite :

IV -4-6-1 Programme linéaire :

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

Le temps requis par l'appareil pour un traitement de toutes les instructions est appelé « **temps cycle** ». Le traitement linéaire du programme est généralement utilisé pour des commandes simples, de volume peu important et peut être réalisé au moyen d'un seul bloc d'organisation (OB).

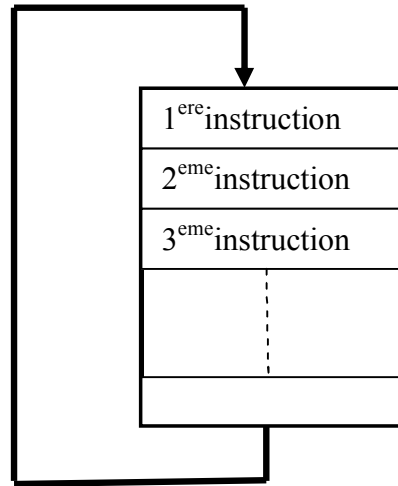


Figure IV.8: Programme linéaire

IV -4-6-2 Programme structuré

Pour les tâches d'automatisation de grande envergure, le programme peut être divisé en blocs de programme ordonnés par fonctions plus petites et faciles à manier. Ceci présente l'avantage de pouvoir tester séparément et de les regrouper en une seule fonction pour l'exécution.

Les blocs du programme doivent être appelés via les instructions d'appel de blocs (**ALLxx / UCxx / CCxx**). Si la fin d'un bloc est détectée, le traitement du programme se poursuit dans le bloc appelant.

La structure du programme est représentée dans la (figure IV.9) :

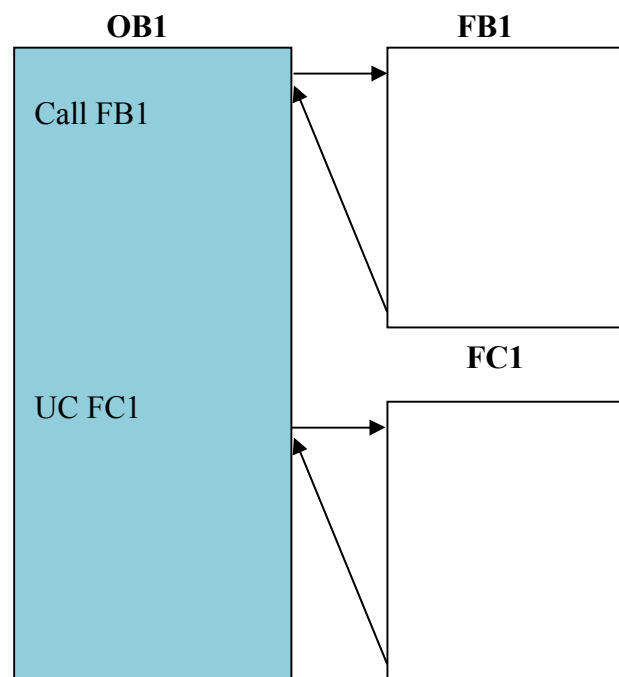


Figure IV.9: Programme structure

Les entités intervenant dans cette structuration sont des blocs de code : les blocs d'organisations (OB), les blocs fonctionnels (FB), les fonctions (FC), les blocs de données (DB), les blocs fonctionnels système (SFB) et les fonctions système (SFC).

IV -4-7 Les principaux blocs du programme utilisateur :

Le logiciel STEP7 dans ces différents langages de programmation possède un nombre important de blocs utilisateur, destiné à structurer le programme utilisateur dont on peut citer les blocs importants suivants :

❖ Bloc d'organisation (OB) :

Un OB est une structure importante d'un programme utilisateur, il est appelé cycliquement par le système d'exploitation ; il gère le traitement de programme cyclique et déclenché par alarme ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs. Vous pouvez programmer les blocs d'organisations et déterminer ainsi le comportement de la CPU en résumé l'OB constitue l'interface entre le système d'exploitation et le programme utilisateur.

Les OB définissent l'ordre dans lequel les différentes parties du programme sont traitées. L'exécution d'un OB peut être interrompue par l'appel d'un autre OB, cette interruption se fait selon la priorité, comme c'est illustré dans la figure suivante (Figure-IV.10)

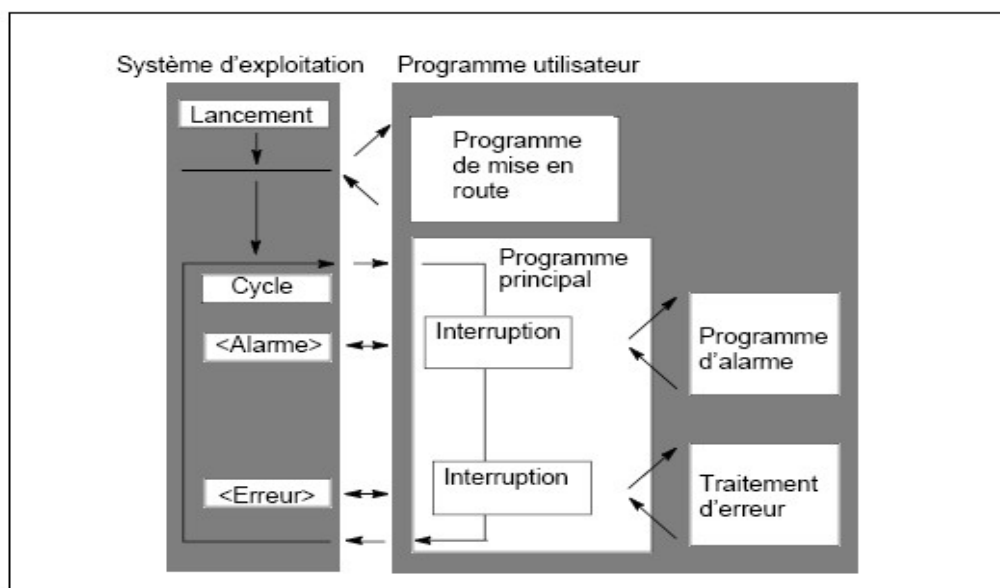


Figure IV.10 : Traitement du programme avec possibilité d'interruption.

❖ Bloc fonctionnel (FB) :

Un bloc fonctionnel contient un programme qui est exécuté, lorsque ce bloc fonctionnel est appelé par un autre bloc de code. Il facilite la programmation de fonctions complexes, comme la commande de moteur (accélération).

Un bloc de données d'instance est associé à chaque FB qui constitue la mémoire, les paramètres transmis au FB ainsi que les variables statiques sont sauvegardés dans le bloc de données locales.

❖ Fonction (FC) :

Les fonctions font partie des opérations que le concepteur du programme d'utilisateur, elle ne possède pas de mémoire. Les variables temporaires d'une fonction sont sauvegardées dans la pile des données locales, ces données sont perdues à l'achèvement 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 quand cette fonction est appelée par un autre bloc de code peut être utilisée pour :

- Renvoyer une valeur de fonction au bloc appelant (exp. Fonction mathématique).
- Exécuter une fonction technologique.

❖ Bloc de données (DB) :

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

- Bloc de données d'instance : Il est associé à chaque appel de bloc fonctionnel transmettant des paramètres, il contient des paramètres effectifs et des données statique du FB, on appelle l'instance l'appel d'un FB pour un DB.
- Bloc de données globales (DB) : Contrairement aux blocs de code (OB, FC, FB), les DB ne contiennent pas d'instructions step7. Ils servent à l'enregistrement de données utilisateur, ils contiennent des données variables que le programme utilisateur utilise.

Tout FB, FC, OB peut 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.

Le rôle des DB dans un programme utilisateur est illustré en (figure IV-11)

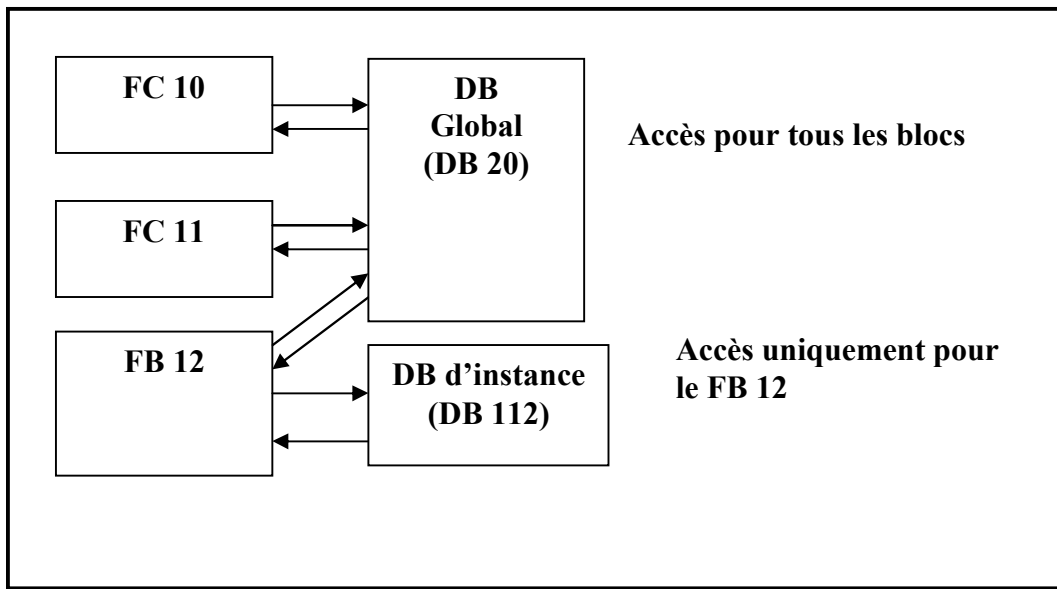


Figure IV-11 : Illustration du rôle des DB dans un programme utilisateur

❖ **Blocs fonctionnels système et fonction système :**

Parmi les atouts des CPU S7, on trouve, les fonctions déjà programmées, qui peuvent être utilisées par le programme utilisateur.

❖ **Bloc fonctionnel système (SFB) :**

Un SFB est un bloc fonctionnel intégré à la CPU S7. Comme les SFB font partie du système d'exploitation, ils ne sont pas chargés en tant que partie de programme. Comme les FB, les SFB constituent des blocs avec mémoire, vous devez donc leur associer des blocs de données d'instance que vous chargez dans la CPU en tant que partie du programme.

❖ Fonction système (SFC) :

SFC sont des fonctions préprogrammées et testées intégrées dans la CPU S7, elles peuvent être appelées par le programme utilisateur. Elles constituent dans blocs sans mémoire dans le système d'exploitation, elles sont pour rôles.

- L'adressage des modules.
- La mise à jour des MIE/ MIS et le traitement de champs binaire.
- La communication par données globales.
- Le diagnostic systèmeetc.

❖ Bloc de données système (SDB) :

C'est une zone de mémoire dans le programme, configurée par différentes applications de STEP 7 pour le stockage des données dans le système d'automatisation.

IV -4-8 Création d'un projet STEP7 :

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

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

Le schéma suivant (figure IV-12) illustre les deux solutions possibles lors de la conception d'une solution d'automatisation :

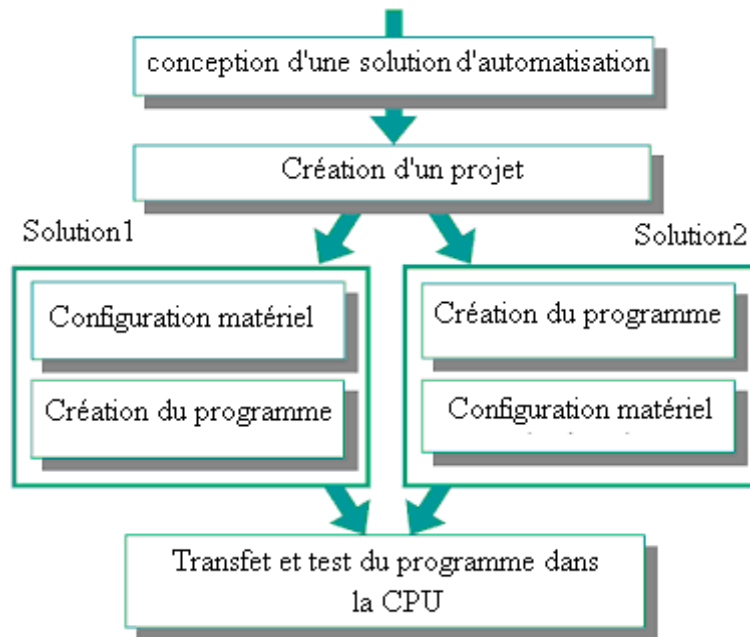


Figure IV -12 : Schéma illustrant les deux solutions possibles pour la programmation

Toutes fois il est recommandé de commencer par la configuration matérielle pour les installations qui contiennent beaucoup d'entrées et de sorties, l'application de la configuration matérielle de STEP7 présente l'avantage de la sélection automatique des adresses. Si on commence par la création de programme, il faudra rechercher les adresses en fonction des constituants choisies, dans ce cas on n'a pas bénéficié de la fonction d'adressage automatique de STEP7.

Les procédures qui vont nous permettre la création de projet sous logiciel STEP7 sont comme suit :


- 1- Double cliquer sur l'icône SIMATIC Manager  ; ceci lance l'assistant de STEP7.
- 2- La fenêtre illustrée en (figure –IV.13) apparaît, elle permet la création d'un nouveau projet.



Figure IV.13 : Assistant de STEP 7 : 'nouveau projet'.

3- En cliquant sur l'icône suivant, la fenêtre suivante apparaît, elle nous permet de choisir la CPU.

Pour notre projet nous avons choisi la CPU 314.

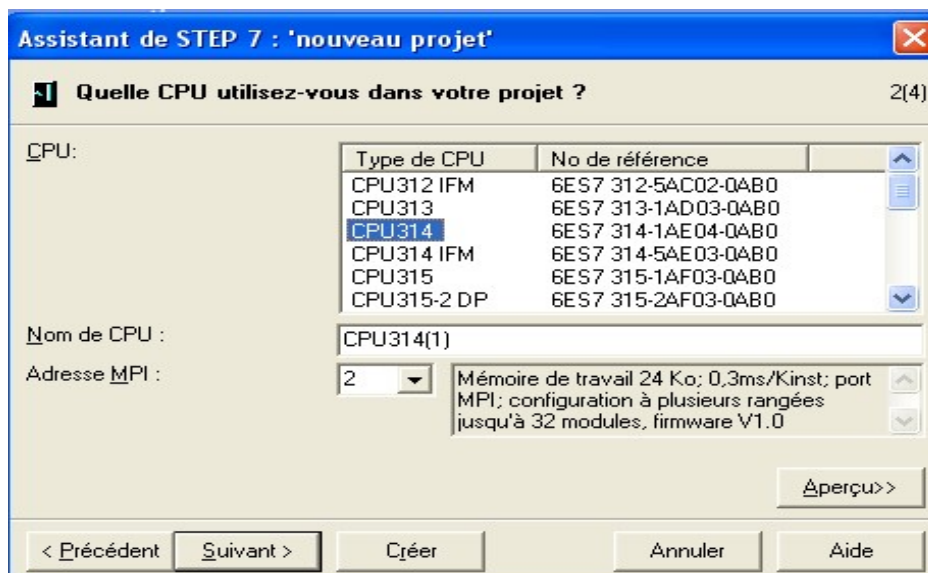


Figure-IV.14 : Fenêtre de choix de la

4- Après validation de la CPU, la fenêtre qui apparaît permet de choisir les blocs à insérer, et choisir le langage de programmation (LIST, CONT, LOG).

5-

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

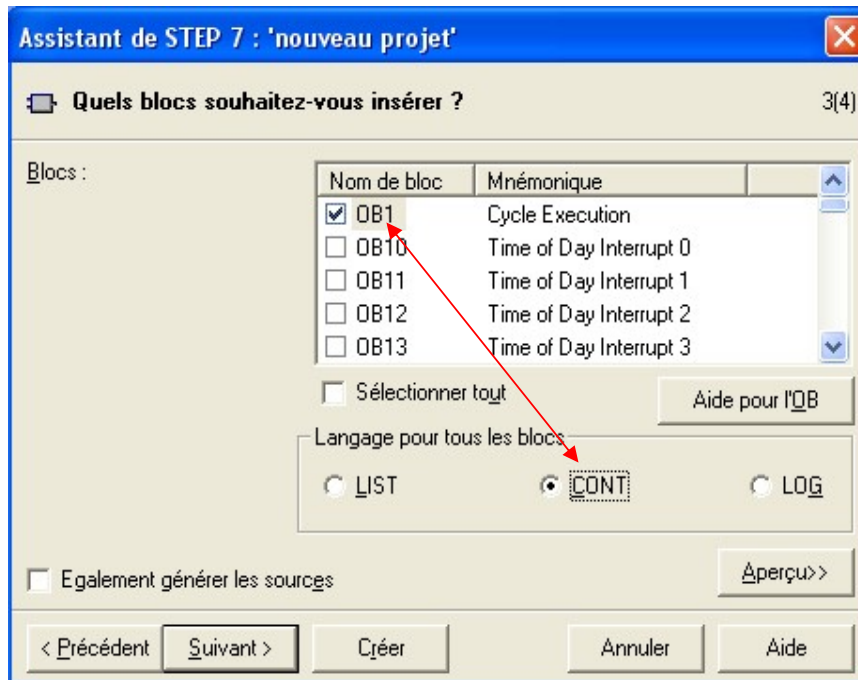


Figure IV.15 : choix des Blocs à utilisés et de langage.

6- En cliquant sur suivant, la création de projet apparaît pour le nommer.

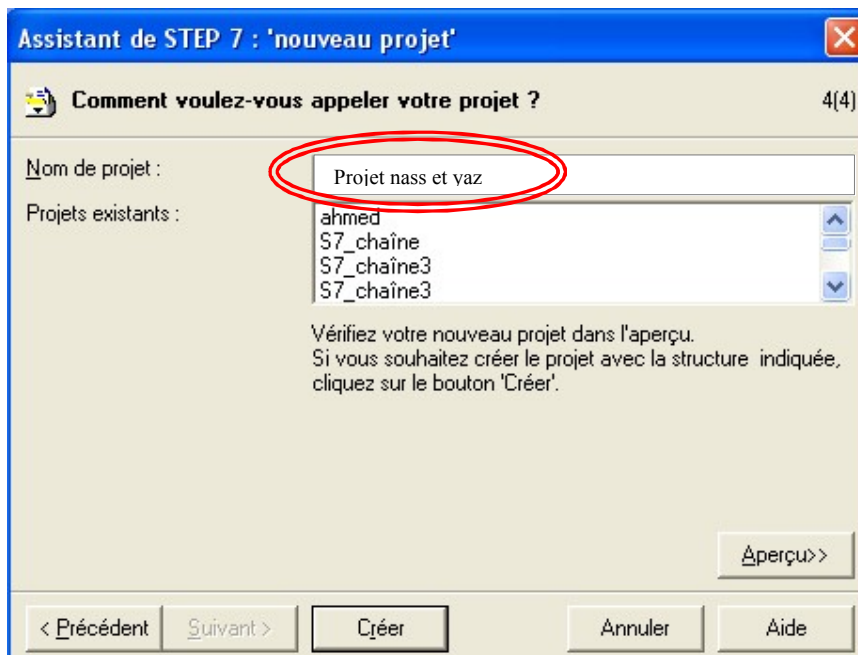


Figure IV.16 : nomination du projet.

7- On clique sur créer, la fenêtre suivante apparaît.

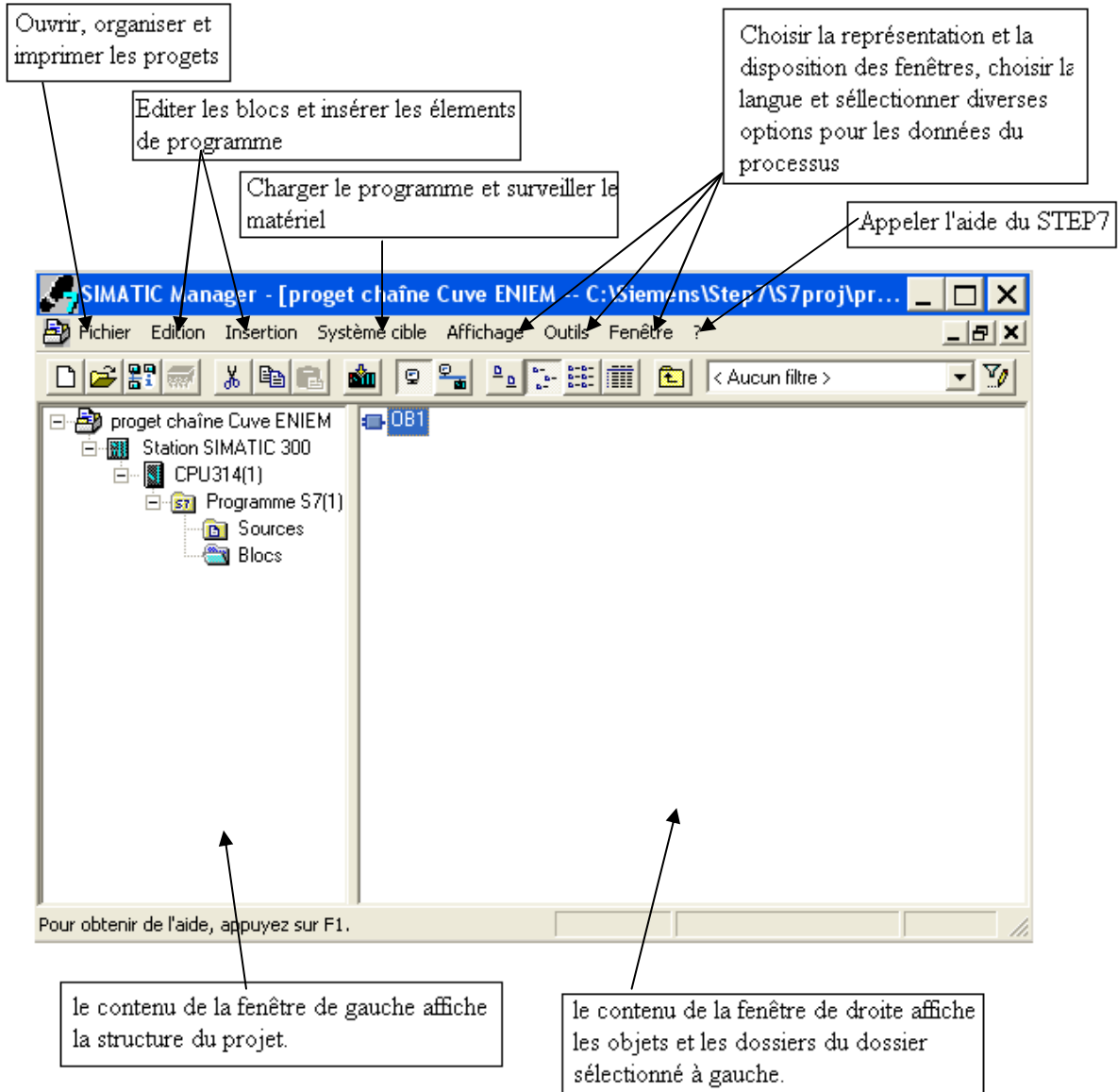


Figure-IV.17 : vue des composants d'un projet S7.

La gestion des fichiers en STEP 7 s'effectue avec 'SIMATIC Manager'. Ici il est possible par exemple de copier les blocs de programme ou d'appeler d'autres traitements avec d'autres outils en cliquant avec la souris. L'utilisation correspond aux standards habituels WINDOWS. (Donc on a entre autres la possibilité de faire apparaître un menu contextuel sur chaque composant par un bouton droit de la souris).

En STEP 7, chaque projet est stocké dans une structure fixe prédéfinie.

Dans les répertoires ,Station SIMATIC 300' et ,CPU' on met en place la conception matérielle du SPS. En conséquence, on voit toujours un tel projet d'un point de vue de son matériel.

Les programmes sont enregistrés dans les répertoires suivants :

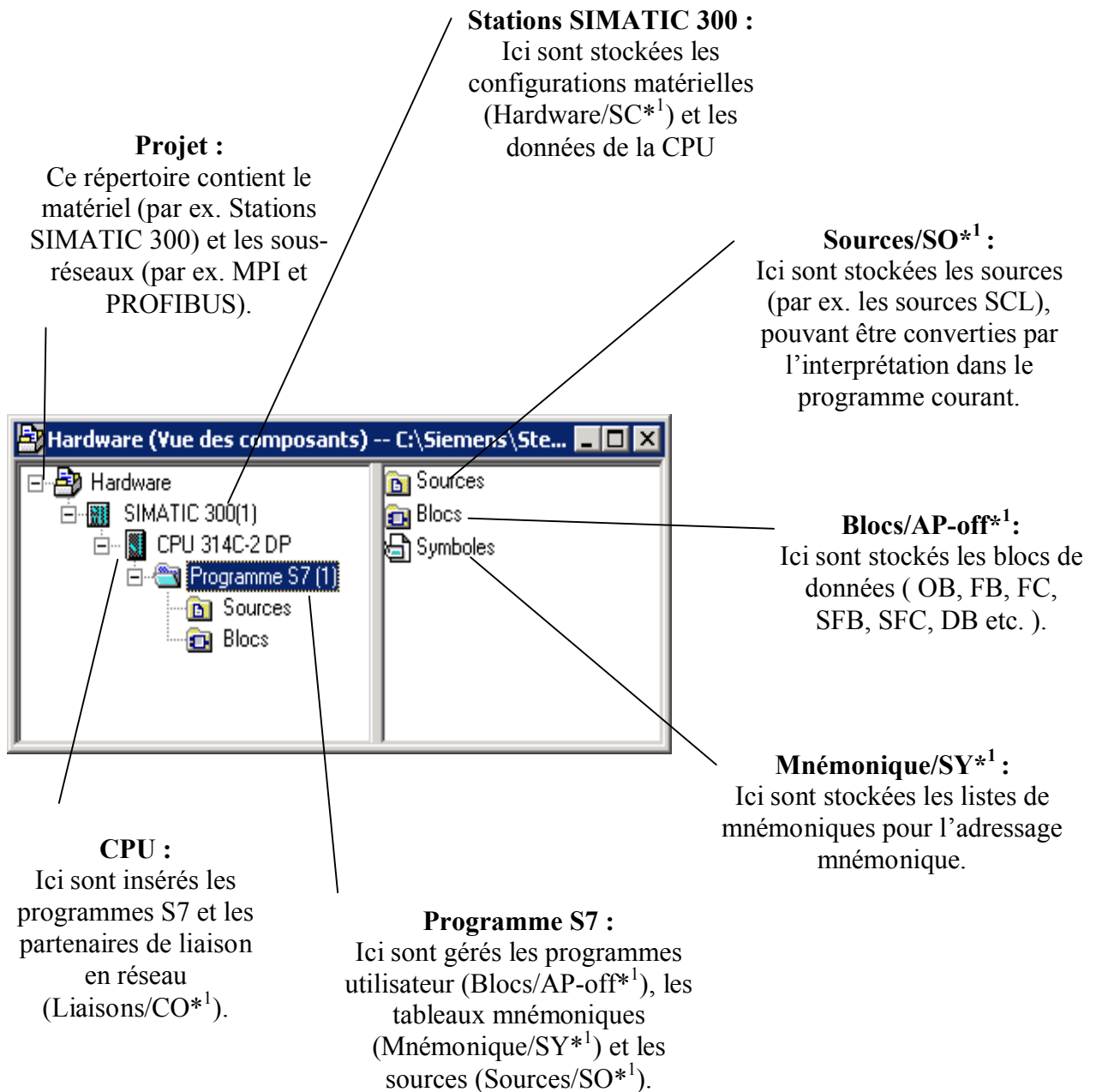


Figure IV.18 : répertoire de la station SIMATIC et de la CPU.

IV -4-9 Configuration et paramétrage de l'automate :

On entend Par "configuration", la disposition de profilés support ou châssis, de modules, d'appareils de la périphérie décentralisée et de cartouches interface dans une fenêtre de station. Les profilés support ou châssis sont représentés par une table de configuration, dans laquelle l'on peut enficher un nombre défini de modules, tout comme dans les profilés support ou châssis "réels".

STEP 7 affecte automatiquement une adresse à chaque module dans la table de configuration. On peut modifier les adresses des modules d'une station, à condition que la CPU permette l'adressage libre.

Avec l'outil 'Configuration HW' il est possible de créer une configuration théorique dérivant de cette dernière et ainsi de configurer une nouvelle conception. De plus, on peut aussi charger une configuration existante depuis une CPU. En plus des modules comme la CPU, d'autres paramètres peuvent être prédéfinis (par ex. comportement de démarrage et de cycle d'une

IV -4-9-1 Marche à suivre pour la configuration d'une station :

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

1. Sélectionnez un composant matériel dans la fenêtre "Catalogue du matériel".
2. Amenez le composant sélectionné dans la fenêtre de station en utilisant la fonction glisser-lâcher.

La figure suivante illustre les manipulations de base :

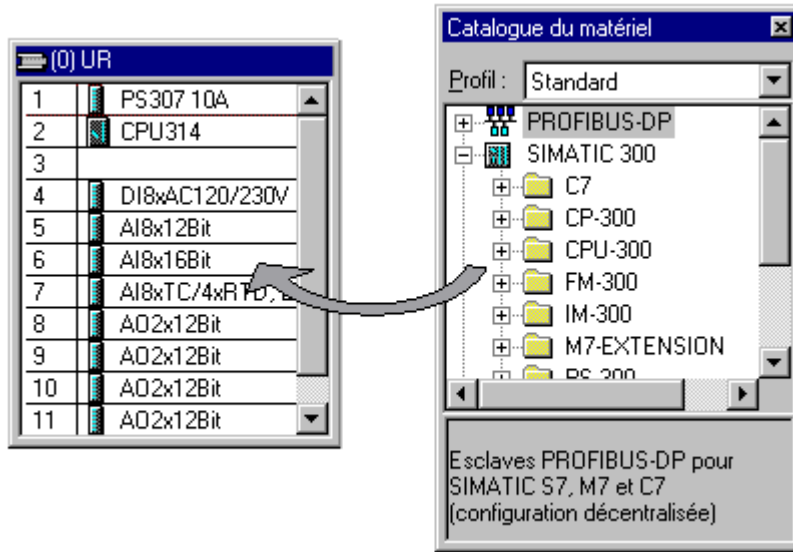


Figure IV.19 : manipulation de base pour une configuration de matérielle.

La figure suivante illustre la configuration matérielle de notre automate.

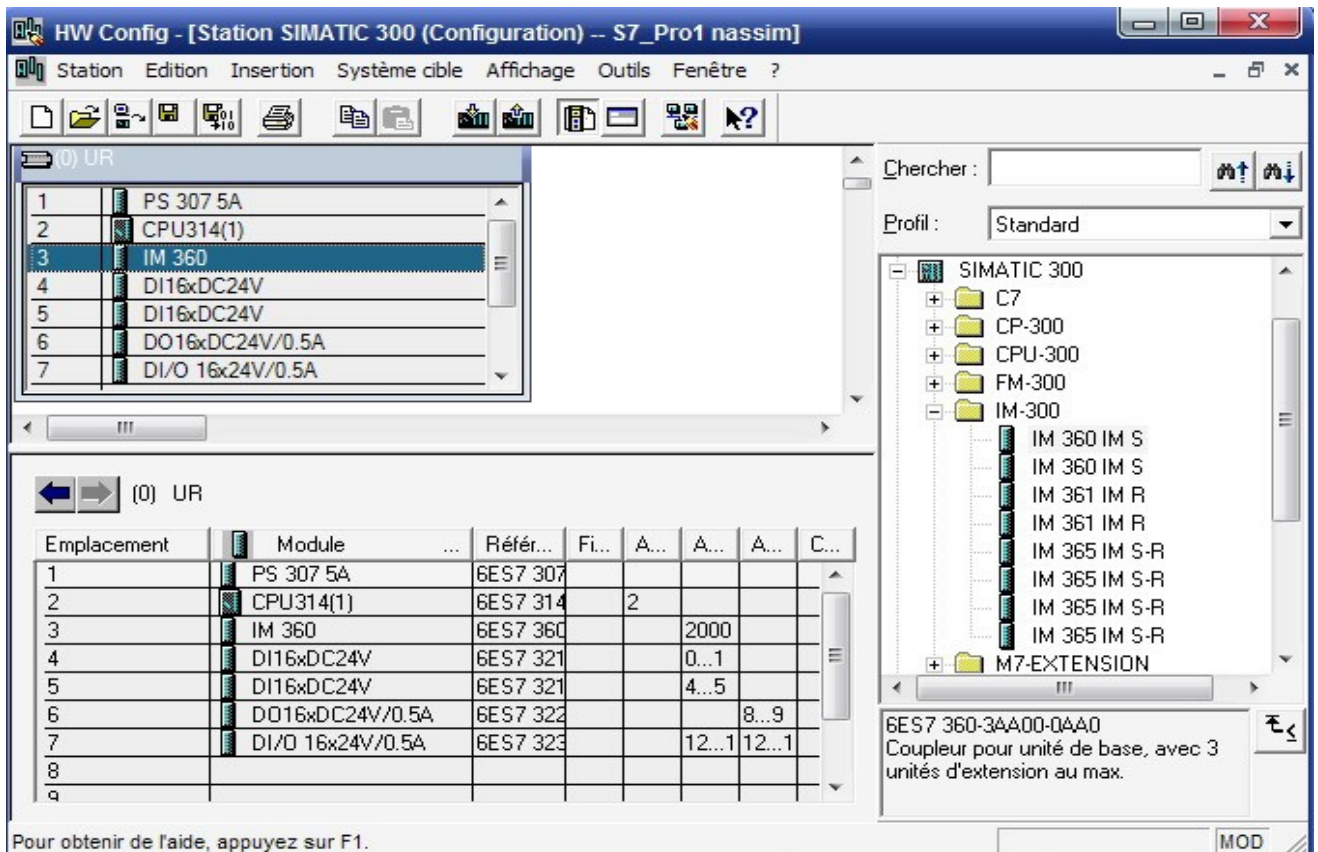


Figure IV.20 configuration matérielle de l'automate

Conclusion

L'automate programmable industriel (API) est un outil adéquat pour les solutions d'automatisation, et conduit des procédés de l'industrie vu les avantages suivants :

- Modification du programme facile a effectué par rapport a une logique câblée.
- Enormes possibilités d'exploitation.
- Fiabilité professionnelle.
- Sa modularité lui permet de réaliser des fonctions diverses.
- Il offre une bonne sécurité pour les procédés et pour lui-même (le chien de garde, les alarmes).

V. Simulation et validation avec PLCSIM :

V -1 Introduction :

Après l'élaboration du programme de commande de notre système à automatiser, nous arrivons à l'étape décisive du travail effectué. Cette étape est la validation du programme par simulation et vérification de son bon fonctionnement.

Pour cela nous avons utilisé le logiciel S7 PLCSIM qui est un logiciel optionnel de STEP 7.

L'application de simulation de modules S7-PLCSIM nous permet d'exécuter et de tester notre programme dans un automate programmable. La simulation étant complètement réalisée au sein du logiciel STEP 7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou modules de signaux). L'objectif de ce logiciel est le test des programmes STEP 7 pour les automates S7-300 et S7-400 qu'on ne peut pas tester immédiatement sur le matériel et ceci pour différentes raisons, l'application est critique, car elle peut occasionner des dommages matériels ou blessures corporelles en cas d'erreurs de programmation, mais la simulation permet de corriger ces erreurs pendant le test de la simulation.

V -2 Présentation du S7 PLCSIM :


L'utilisation du simulateur de modules physiques S7-PLCSIM nous permet d'exécuter et de tester le programme dans un automate de simulation que nous simulons dans un ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7.

Le S7-PLCSIM dispose d'une interface simple nous permettant de visualiser et de forcer les différents paramètres utilisés par le programme (comme activer ou désactiver des entrées.). Tout en exécutant le programme dans l'API de simulation, nous avons également la possibilité de mettre en œuvre les diverses applications du logiciel STEP7 comme, par exemple, le test de bloc afin de visualiser les variables d'entrées et de sorties.

V -3 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.
- ✓ Cliquez sur  ou sélectionnez la commande Outils > simulation de modules.

Cela lance l'application S7-PLCSIM et ouvre une fenêtre CPU (figure V-1):

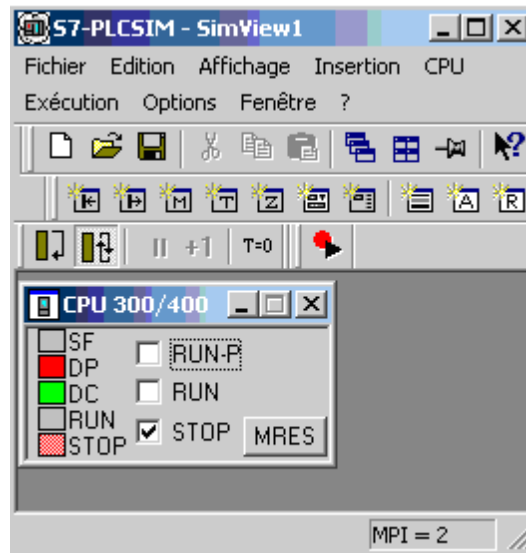






Figure-V-2 : fenêtre du S7-PLCSIM

- ✓ Dans le gestionnaire de projets SIMATIC, chercher le projet-exemple, projet (machine à garnir les encoches) et dans ce dernier, chercher le dossier Blocs.
- ✓ Dans le gestionnaire de projet SIMATIC, cliquez sur  ou choisir la commande Système cible (charger) pour charger le dossier blocs dans l'API de simulation.

Dans l'application S7-PLCSIM, on crée de nouvelles fenêtres pour visualiser les informations provenant de l'API de simulation :

- ✓ Cliquez sur  ou choisissez la commande Insertion (Entrée) pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des entrées (zone E). Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut EB0. Mais on peut modifier l'adresse (EB1, EB2...).
- ✓ Cliquez sur  ou choisissez la commande Insertion (Sortie) pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des sorties

(zone A). Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut AB0. Mais on peut modifier l'adresse (AB1, AB2...).

- ✓ Cliquez sur  ou choisissez la commande Insertion (Temporisation) pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer les temporisations utilisées par le programme. Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut T 0.

Choisir le menu CPU dans la fenêtre du s7-PLCSIM et vérifier que la commande à Mettre sous tension est activée (figure V-3).

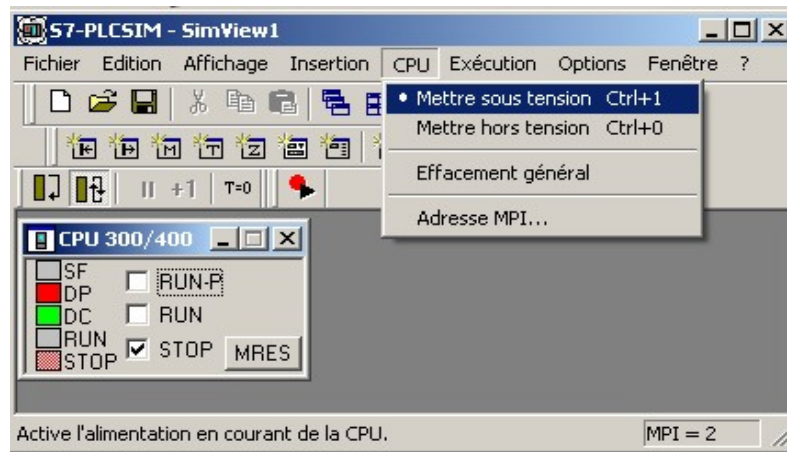


Figure-V-3 : Mise sous tension de la CPU

- ✓ Choisir la commande Exécution (Mode d'exécution) et vérifier que la commande cycle continue est activée. (figure-V-4)

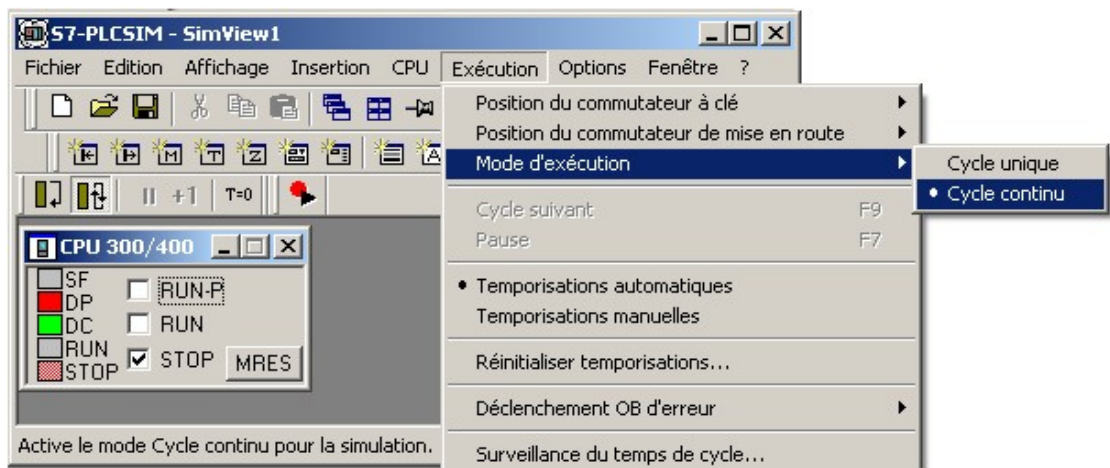



Figure V-4 : Choix du cycle continu.

Mettre la CPU de simulation en marche en cliquant sur l'une des cases à cocher RUN ou RUN-P (figure-V-5)



Figure -V-5 : Mise en marche de la CPU

Pour sauvegarder la version actuelle de la simulation d'AP, cliquez  ou choisissez la commande Fichier (Enregistrer CPU).

Une fois toutes les fenêtres d'entrées et de sorties sont prêtes, nous activons les entrées voulues pour lire l'état des sorties. Dans notre cas, par exemple, pour faire avancer la bonde de papier isolant. (Figure V-6)

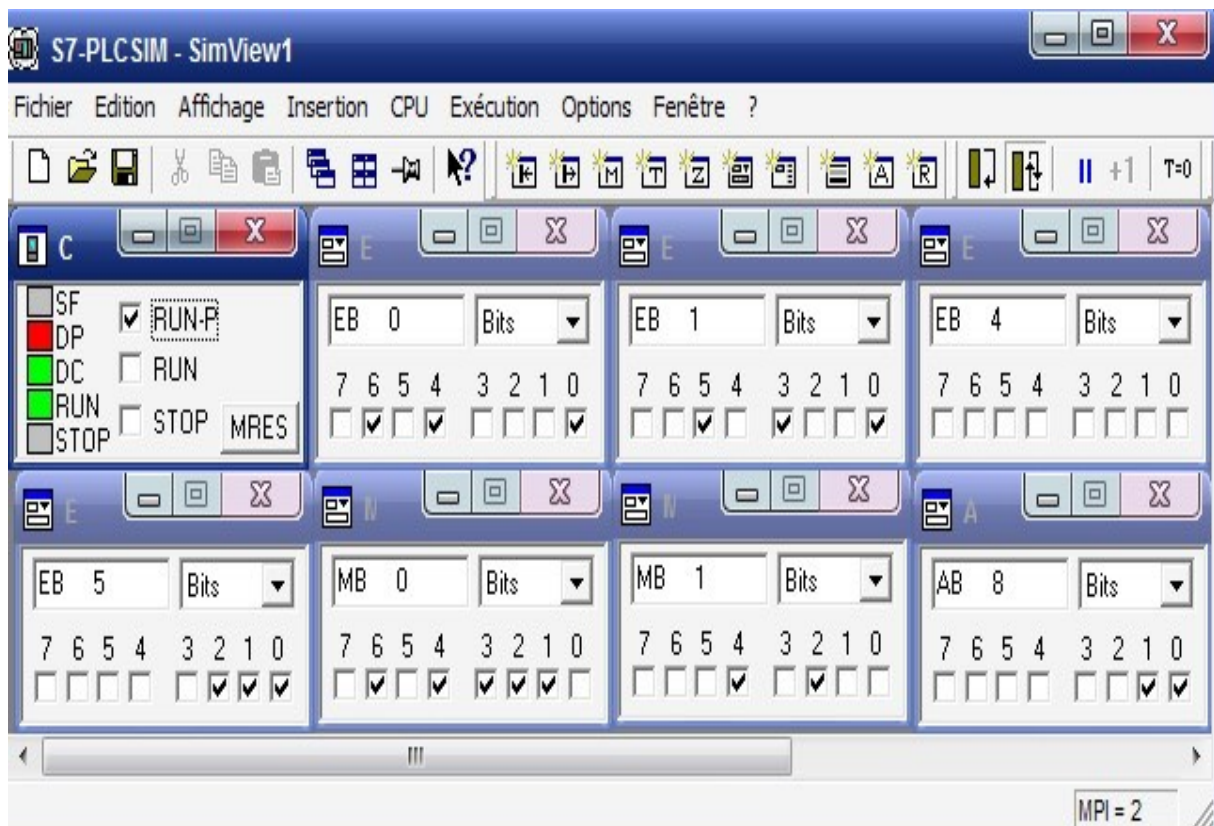


Figure V-6 : Simulateur S7-PLCSIM

V-4 Visualisation de l'état du programme :

Après le chargement du programme dans la CPU du simulateur et la mise de cette dernière en mode « RUN » le STEP 7 nous permet de visualiser l'état du programme soit en cliquant sur l'icône



ou en sélectionnant la commande Test > Visualiser.

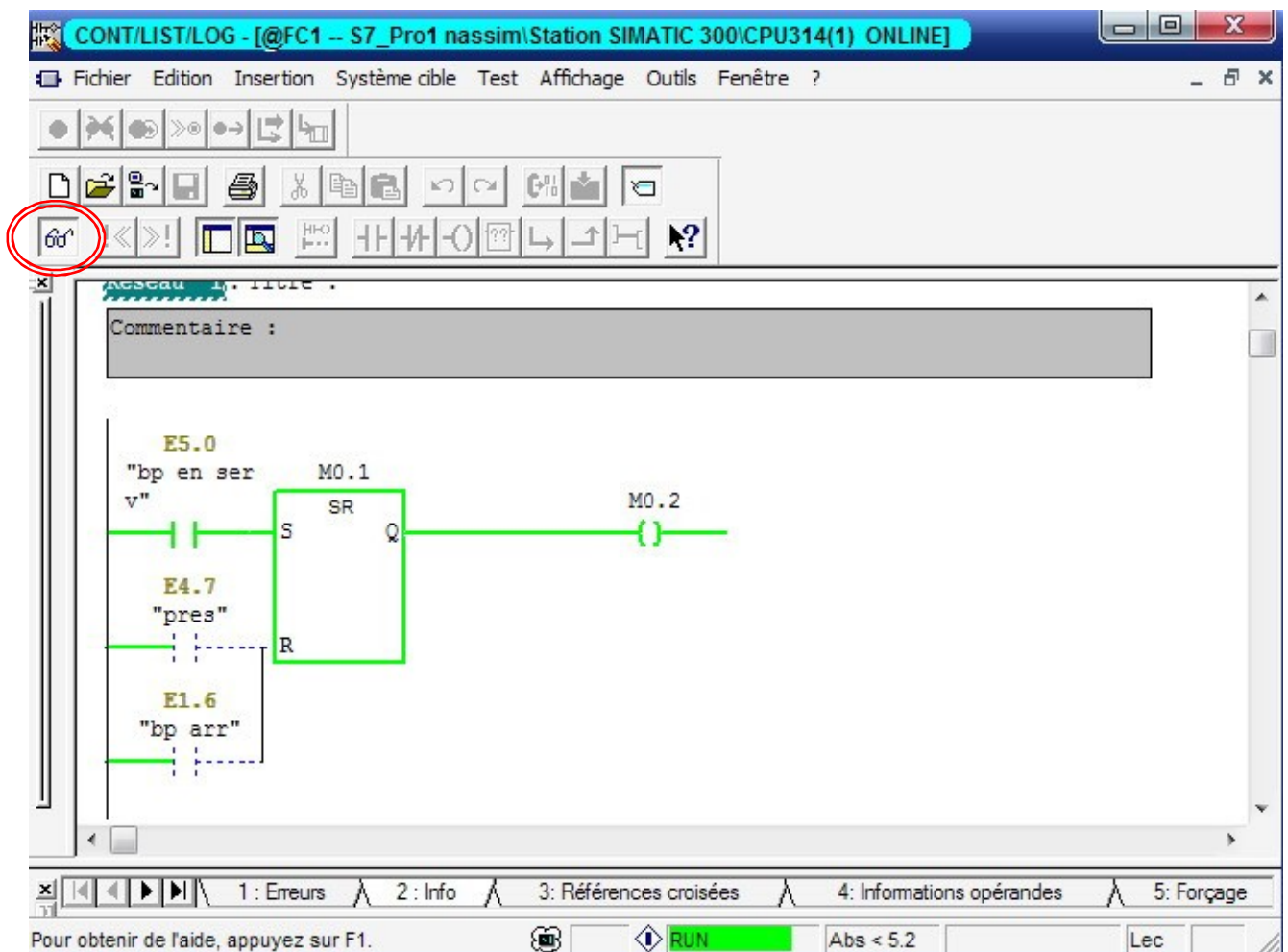


Figure V-7 : Visualisation de l'état du programme.

Conclusion :

En absence d'un automate réel, le logiciel S7-PLCSIM devient l'outil indispensable pour la simulation des programmes et des concepts de commandes automatisés avant leur implantation sur un système réel.

Ce logiciel nous a permis de corriger toutes les erreurs commises, et d'apporter les modifications sur le programme avec une très grande facilité. Il nous a aussi permis de valider et de visualiser le comportement des sorties et de simuler le programme de l'unité.

L'Entreprise Nationale d'Electro-Industries a été créée sous sa forme actuelle en janvier 1999, après la scission de l'Entreprise mère ENEL (Entreprise Nationale des Industries Electrotechniques). Son activité de production remonte à 1986, dans les domaines de fabrication des moteurs Electriques, Alternateurs et transformateurs de distribution.

Les produits fabriqués par l'Electro-Industries sont conformes aux recommandations CEI et aux normes Allemandes DIN / VDE.

La production actuelle est écoulee sur le marché Algérien et génère un chiffre d'affaire de 1,8 Milliards de Dinars.

La capacité de production de transformateur de l'entreprise couvre les besoins du marché à 70 pour cent environ, Et les ventes de moteurs représentent 30 pour cent environ de la capacité totale de production.

L'entreprise emploie un effectif de 776 agents dont 15 pour cent de cadres, 34 pour cent de maîtrise et 51 pour cent d'exécution. En matière de qualité l'Electro-Industries dispose de ses propres laboratoires d'essai et de mesure, ainsi que de ses produits pour le contrôle des différents matériaux utilisés dans sa fabrication.

Pour le système documentaire, l'entreprise utilise 252 normes internes en plus des normes DIN / VDE et CEI, et les différentes valeurs d'essais et de mesure sont consignées sur des procès-verbaux et des cartes de contrôle.

- **Les grandes structures du complexe :**

L'entreprise est composée de deux (02) unités qui se trouvent toutes les deux sur un même site :

- **Département moteurs, alternateurs et prestation technique**

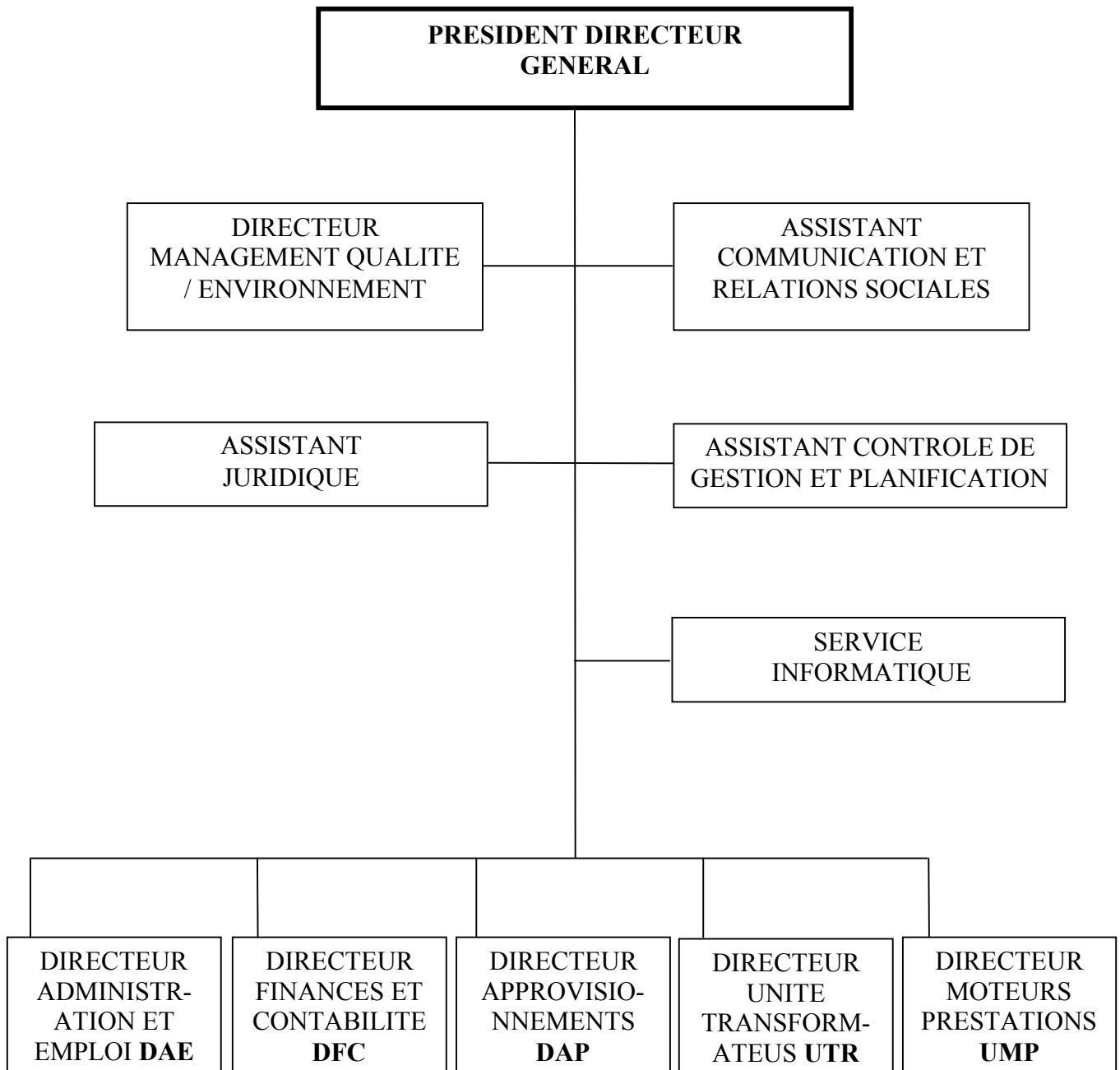
Celui-ci est composé de deux bâtiments (2 et 2A), et est chargé des activités suivantes

- L'usinage ;
- La fonderie aluminium sous pression ;
- Le bobinage ;
- L'imprégnation ;
- Le montage des moteurs et alternateurs ;
- Maintenance de l'ensemble des équipements de l'entreprise ;

➤ **Département transformateur**

Il est composé de deux bâtiments (3 et 3A), l'un pour la construction métallique des cuves et des vases d'expansions et l'autre pour le bobinage, le découpage des tôles magnétiques utilisées dans les noyaux et le montage des transformateurs.

ORGANIGRAMME DE L'EPE / ELECTRO-INDUSTRIES / SPA, AZAZGA



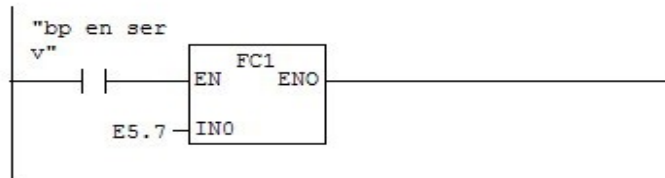
Etat	Mnémonique /	Opérande	Type de don	Commentaire
	bp arr	E 1.6	BOOL	
	bp aut	E 1.7	BOOL	
	bp av	E 4.1	BOOL	
	bp div	E 4.6	BOOL	
	bp en serv	E 5.0	BOOL	
	bp ench	E 4.3	BOOL	
	bp encl	E 4.4	BOOL	
	bp f-c	E 4.2	BOOL	
	bp indx	E 4.5	BOOL	
	bp pas	E 4.0	BOOL	
	comp	E 5.1	BOOL	
	Cycle Execution	OB 1	OB 1	
	pres	E 4.7	BOOL	
	s1	E 0.0	BOOL	
	s10	E 1.1	BOOL	
	s11	E 1.2	BOOL	
	s12	E 1.3	BOOL	
	s13	E 1.4	BOOL	
	s14	E 1.5	BOOL	
	s2	E 0.1	BOOL	
	s3	E 0.2	BOOL	
	s4	E 0.3	BOOL	
	s5	E 0.4	BOOL	
	s6	E 0.5	BOOL	
	s7	E 0.6	BOOL	
	s8	E 0.7	BOOL	
	s9	E 1.0	BOOL	
	sel ajust	E 5.4	BOOL	
	sel aut	E 5.2	BOOL	
	sel ein	E 5.3	BOOL	
	y0m	A 8.0	BOOL	
	y1a	A 8.1	BOOL	
	y2a	A 8.2	BOOL	
	y3a	A 8.3	BOOL	
	y4a	A 8.4	BOOL	
	y5a	A 8.5	BOOL	
	y6a	A 8.6	BOOL	

OB1 : "Main Program Sweep (Cycle)"

Commentaire :

Réseau 1: Titre :

Commentaire :

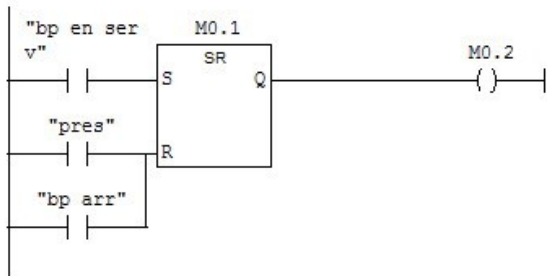


FC1 : Titre :

Commentaire :

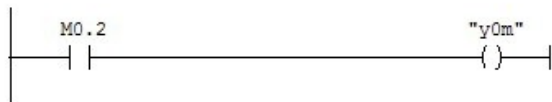
Réseau 1 : Titre :

Commentaire :



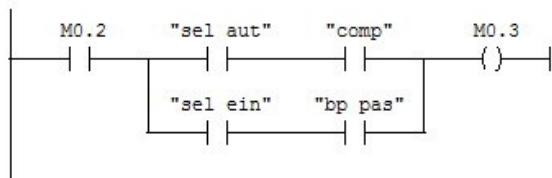
Réseau 2 : Titre :

Commentaire :



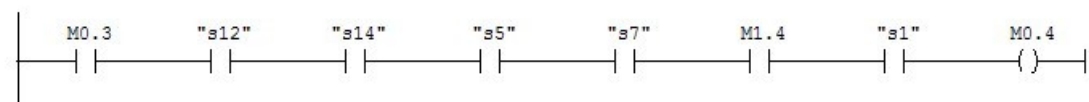
Réseau 3 : Titre :

Commentaire :



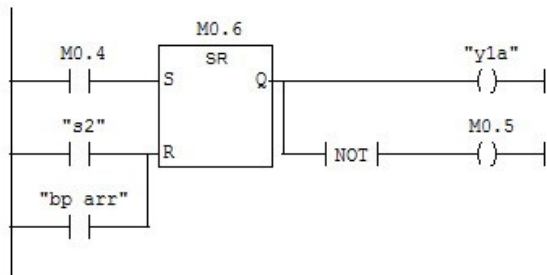
Réseau 4 : Titre :

Commentaire :



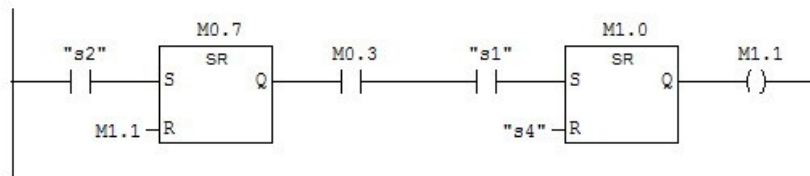
Réseau 5 : Titre :

Commentaire :



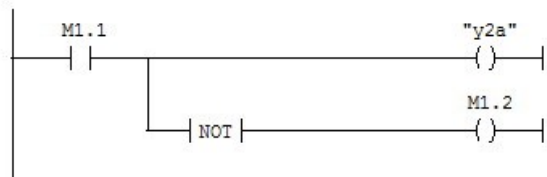
Réseau 6 : Titre :

Commentaire :



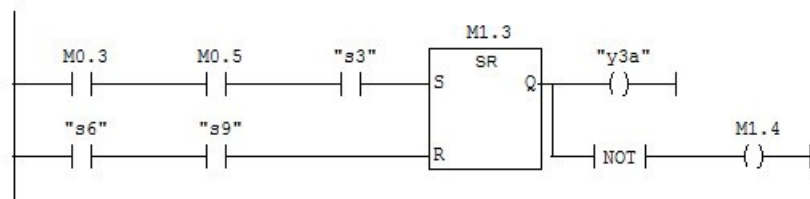
Réseau 7 : Titre :

Commentaire :



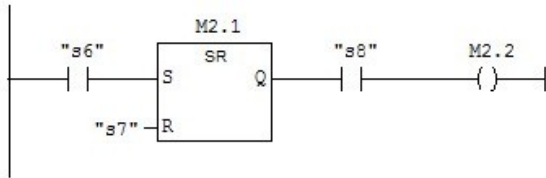
Réseau 8 : Titre :

Commentaire :



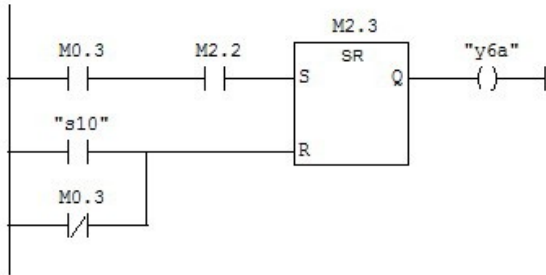
Réseau 9 : Titre :

Commentaire :



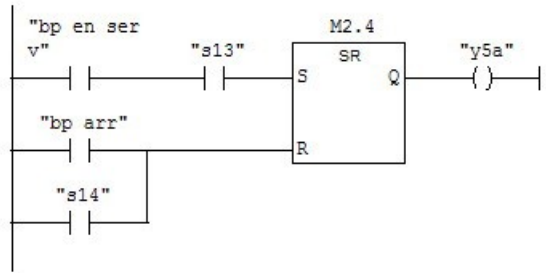
Réseau 10 : Titre :

Commentaire :



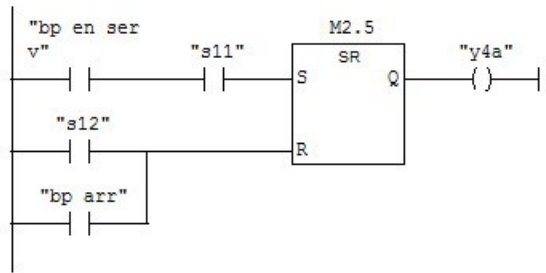
Réseau 11 : Titre :

Commentaire :



Réseau 12 : Titre :

Commentaire :



Conclusion générale

Notre projet de fin d'étude a été effectué en grande partie au sein de l'entreprise d'électro-industrie (E.I) d'Azazga, dans le but de concevoir une solution programmable pour la machine à garnir les encoches et remplacer le système de commande à base de logique câblée déjà existant par un API.

A l'issue de notre travail, nous pouvons conclure que :

Pour être à la page avec les exigences introduites par l'évolution des industries, la commande des processus avec un automate programmable industriel est la solution souvent recherchée, vue la justesse des traitements numériques que les API effectuent pour générer la commande adéquate à tout moment et dans toutes les conditions.

L'évolution des API ne cesse de continuer et notamment leurs logiciels de programmation, l'API S7-300 procure plusieurs avantages tels que la flexibilité, la facilité d'extension de ses modules et la possibilité de visualisation et validation du programme établie avant son implantation grâce à son logiciel de simulation S7-PLCSIM.

Une bonne automatisation d'un procédé doit être performante et d'un coût optimal, cela est obtenu en passant par :

- ✓ L'élaboration d'un cahier des charges qui comprend tous les aspects fonctionnels du processus.
- ✓ La modélisation du cahier des charges par un des outils de modélisation par exemple le GRAFCET.
- ✓ Le choix optimal de la partie commande (API), de la partie opérative (actionneurs) et les moyens de dialogues (capteurs).

Ce que nous retenons du travail que nous avons effectué durant ces trois mois à l'ENEL est primordial dans notre vie d'ingénieur. Ce stage pratique nous a été bénéfique à plus d'un titre, compte tenu des nombreux avantages qu'il présente. La découverte du monde industriel, la mise en

application de la théorie acquise lors de notre cursus et l'expérience engrangée lors de notre collaboration avec les ingénieurs sur les lieux.

Nous espérons que notre travail verra naître sa concrétisation sur le plan pratique et que les promotions à venir puissent en tirer profit.

BIBLIOGRAPHIE

[1] : **Christian Merlaud, Jacques Perrin, Jean-Paul Trichard** “ Automatique et informatique industriel”.

Edition DUNOD, Année 1995.

[2] : **Jacques Faisandier**, “Mécanismes hydrauliques Et pneumatiques”.

Édition DUNOD 1999.

[3]: **HENRY NEY**, “Eléments d’automatismes”.

Edition Fernand Nathan 1985.

[4]: **J -M Bleux. J.-L Fanchon**. “Les automates industriels” Collection Etapes.

Edition Nathan, année 1996.

[5]: **Moreno et Epeulot**, “ La pneumatique dans les systèmes automatisés de production”

[6] : **Réalisation technologique du GRAFCET** par Daniel Dupont et David Dubois.

[7] : documentations technique de l’ENEL.

[8] : document festo année 2002.

[9] : **Documentation techniques de Siemens**,

[10]: Mémoire de fin d’étude, **Mr B. Ideki et Mr M. Tesbia**, “Utilisation du GRAFCET pour l’automatisation d’une plieuse taraudeuse”, année 2004/2005

[11] : Mémoire de fin d’étude, **Mr M. Boudissa et Mr A. Benlalli**, “Conception d’une commande programmable d’une plieuse de tôle à base d’un automate programmable S7-300”

[12]: site web; **www.STielec-ac-aix-marseille.fr /électro technique/cour.htm**.

Détecteurs de position et de proximité Jean-Louis Hu.

[13] : Site internet : Patrick TRAUM. “ **Le GRAFCET et sa mise en œuvre**” cours disponible sur [http :\ pat.fr.st](http://pat.fr.st).

- Compact disc (cd)

- ✓ Logiciel step 7version 5.3, année 2002.

- ✓ Simatic manuelle collection, année 2002.

- ✓ Logiciel de simulation manager en Français, année 2002.

- ✓ Logiciel Fluid Sim, année 2002.