

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud Mammeri de Tizi-Ouzou**  
**Faculté de Génie Electrique et de l'Informatique**  
**Département Electronique**



# Mémoire

*De fin d'études*

*En vue de l'obtention d'un diplôme  
Master Professionnel  
Option : Electronique Industrielle*

*Thème*

**Etude et automatisation par API S7-300 de la  
machine à fabriquer les cales d'encoches**

**Dirigé par :**

Mr. Y.ATTAF

**Présenté par :**

Mr. TALMAT AMAR Ahmed

**Promotion : 2011/ 2012**

# Remerciement

Ce n'est pas facile d'arriver à accomplir notre travail sans aide ni conseils à cause des problèmes rencontrés tant pratiques que théoriques.

Pour cela, tout mes remerciements au promoteur, Monsieur ATTAF qui a accepté de m'encadrer et pour sa disponibilité et ses orientations pour mener à bien mon travail.

Mes remerciements s'adressent aussi à toutes les personnes qui m'ont encouragé pour aller au bout de notre travail.

Mes derniers remerciements vont aux membres de jury qui me feront l'honneur de juger mon travail.

# *Dédicaces*

Je dédie ce travail :

A mes très chers parents, en témoignant de ma plus profonde gratitude et mon incontestable reconnaissance ;

A mes grans parents ;

A mon frère « Idir » ;

A ma soeur « Micha » ;

A ma fiancé « Djamila » et sa famille;

A toute la famille TALMAT AMAR ;

A tous mes amis ;

A tous mes collègue d'étude ;

A tous ce qui mon chère(s).

Introduction générale.....	1
----------------------------	---

## **Chapitre I : Etude de la machine à fabriquer les cales d’encoches.**

I. Présentation de l’organe d’accueil.....	3
I.1. Composition du complexe d’Electro-Industries.....	3
II. Description de la machine à fabriquer les cales d’encoches.....	5
II.1. Unités de la machine à fabriquer les cales d’encoches.....	6
II.1.1. Unité bobine à ruban.....	6
II.1.2. Unité de freinage.....	7
II.1.3. Unité de chauffage.....	8
II.1.4. Unité de mise en forme.....	9
II.1.5. Unité de découpage.....	10
II.1.6. Panneau de commande.....	11
II.1.7. Armoire électrique.....	12
II.1.8. Goulotte de sortie.....	13
II.2. Partie opérative.....	14
II.2.1. Pré-actionneurs.....	14
a. Distributeur pneumatique.....	14
b. Electrovanne « tout ou rien ».....	14
II.2.2. Les actionneurs.....	15
II.2.3. Les capteurs.....	16
a. Capteur de proximité inductif.....	16
b. Interrupteurs à lames souples.....	16
c. Thermostat.....	17
d. Encodeur rotatif.....	18
II.2.4. Les circuits électriques de protection.....	18
a. Les circuits RC.....	18
b. Les fusibles.....	19
c. Les contacteurs.....	19
d. Transformateur galvanique (220/220 V).....	19
II.2.5. Variateur de vitesse électronique.....	20
II.2.6. Moteur à courant continu.....	20
II.2.7. Résistances chauffantes.....	20
II.3. Partie commande.....	20
II.3.1. Pupitre de commande.....	20
II.3.2. Commande de la machine.....	20
III. Fonctionnement de la machine à fabriquer les cales d’encoches.....	21
IV. Conclusion.....	21

## **Chapitre II : Modélisation de la machine à l’aide du GRAFCET.**

I. Introduction .....	22
II. Définition du GRAFCET.....	22
III. Composition du GRAFCET.....	22
III.1. Les étapes.....	22
III.2. Les transitions.....	23
III.3. Les liaisons orientées.....	25
III.4. Les actions.....	25

III.5. Les règles d'évolution du GRAFCET.....	27
III.6 Niveau d'un GRAFCET.....	28
IV. Création et simulation du GRAFCET.....	28
V. Définition du logiciel Automgen V7.....	28
V.1. Lancement d'automgen.....	28
V.2. Le projet.....	28
V.3. La vue générale.....	29
V.4. La cible.....	29
V.5. Les folios.....	30
V.6. Iris.....	30
V.7. Le choix de la palette.....	31
V.8. Exécution de l'application.....	31
V.8.1. Compilateur.....	31
V.8.2. Exécution.....	31
VI. Liste des actionneurs, pré-actionneurs et capteurs.....	32
VI.1. Les actionneurs et pré-actionneurs.....	32
VI.2. Les capteurs.....	32
VI.3. Les boutons.....	32
VI.4. Compteur.....	32
VII. Conclusion.....	33

### **Chapitre III : automatisation de la chaine par automate programmable industriel S7-300.**

I. Introduction aux automates programmable.....	34
I.1. Automates programmables industriels (API).....	34
I.2. Structure d'un système automatisé.....	35
I.3. Choix d'un automate.....	36
II. Présentation générale de l'automate S7-300.....	36
II.1. Caractéristiques de l'automate S7-300.....	37
II.2. Constitution de l'automate S7-300.....	37
II.2.1. Module d'alimentation (PS).....	38
II.2.2. Description de la CPU.....	38
II.2.3. Modules de coupleurs (IM).....	39
II.2.4. Modules de signaux (SM).....	39
II.2.5. Modules d'entrées et de sorties tout ou rien (TOR).....	39
II.2.6. Modules d'entrée et sortie analogique.....	40
II.2.7. Module de fonction (FM).....	40
II.2.8. Module de communication (CP).....	40
II.2.9. Châssis d'extension (UR).....	40
II.3. Fonctionnement de base d'un API.....	40
II.3.1. Le module central CPU.....	40
II.3.2. Réception des informations sur les états du système.....	40
II.3.3. Exécution du programme utilisateur.....	41
II.3.4. La commande du processus.....	41
III. Programmation de l'automate S7-300.....	41
III.1. Création d'un projet STEP7.....	43
III.2. Configuration et paramétrage de l'automate.....	47
III.3. Marche à suivre pour la configuration d'une station.....	48
III.4. Adressage des modules du S7-300.....	49

III.4.1. L'adressage des modules lié a l'emplacement.....	49
III.4.2. L'adressage absolu des modules de signaux.....	50
III.5. Mémento.....	50
III.6. L'instruction de commande.....	51
III.7. Traitement du programme par l'automate.....	51
III.8. Principe de conception d'une structure de programme.....	51
III.8.1. Le rôle de système d'exploitation.....	51
III.8.2. Rôle du programme utilisateur.....	52
III.9. Blocs fonctionnels système et fonctions système.....	54
III.9.1. Bloc fonctionnel système (SFB).....	54
III.9.2. Fonction système (SFC).....	54
III.9.3. Bloc de données système (SDB).....	54
III.10. Implantation du Grafset dans le S7-300.....	54
IV. Conclusion.....	55

## **Chapitre IV : Simulation avec le logiciel S7-PLCSIM et la configuration du panel tactile à l'aide du WinCC Flexible.**

I. Simulation et validation avec PLCSIM.....	56
I.1. Introduction.....	56
I.2. Présentation du S7-PLCSIM.....	56
I.3. Mise en route du logiciel S7-PLCSIM.....	56
I.4. Visualisation de l'état du programme.....	59
I.5. Simulation d'un programme.....	59
II. Conclusion.....	59
III. WinCC Flexible.....	60
III.1. Introduction au WinCC.....	60
III.1.1 WinCC Flexible Advanced.....	60
III.1.2. WinCC Flexible Runtime.....	60
IV. Choix du panneau tactile TP177A.....	61
V. Etapes de configuration.....	62
V.1. créer un projet.....	62
V.2. Configuration de la liaison.....	64
V.3. Création d'une vue (Menu).....	64
V.4. Création de variables.....	64
V.5. Configuration d'une alarme.....	65
V.6. Configurer l'affichage d'alarmes.....	67
V.7. Ajout de changements de vue.....	70
V.8. Définir la fonction STOP_RUNTIME.....	71
V.9. Teste et simulation du projet.....	72
V.9.1.Exemple de test du projet (Compilation).....	72
V.9.2. Exemple de simulation du projet.....	72
VI. Conclusion.....	74
<b>Conclusion générale.....</b>	<b>75</b>

# **INTRODUCTION**

# **GENERALE**

### Introduction générale :

La première révolution industrielle a commencé vers 1814 avec l'apparition de la machine à vapeur, puis de la dynamo électrique. Un grand nombre de spécialistes pensent que la seconde révolution industrielle a commencé en 1955 avec la mise au point du premier transistor. Mais, c'est au 1980 que la troisième révolution industrielle a commencé avec l'éclosion de la micro-électronique et de la micro informatique. Ces nouvelles technologies permettent de réaliser des systèmes complexes dont on ne mesure pas encore les limites. Dès lors, les êtres humains confient à des machines et des robots les tâches pénibles.

L'automatisation est un domaine pluridisciplinaire qui associe la notion de la mécanique, de l'électronique et de l'informatique. Elle permet d'exécuter des tâches industrielles avec une intervention humaine très réduite. Elle redevient ces dernières années le centre de préoccupation d'un bon nombre d'entreprises à cause de son évolution vers l'étude et la maîtrise des systèmes de plus en plus complexes, permettant l'exécution et le contrôle de tâches techniques par des machines fonctionnant sans intervention humaine, ou avec intervention réduite. Avec les progrès technologiques, 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.

L'entreprise Electro-Industries sise à mi-chemin, entre les deux localités de Fréha et Azazga, s'est équipée de plusieurs machines dont la commande est moins complexe. Les premières machines étaient dotées de commande électromécaniques et de logique câblée. Les plus récentes sont commandées par des automates programmables.

L'entreprise Electro-Industries a choisi comme partenaire la firme allemande de marque « SIEMENS » pour automatiser ses machines de production.

Dans les chaînes de production, les automatismes occupent une place importante. Le complexe Electro-Industrie représente un complexe typique. Cette entreprise, possède deux principales chaînes de production, celle des transformateurs, et celle des moteurs électriques.

Dans ces deux chaînes, de nombreux postes de production utilisent des automatismes construits avec la technologie de la logique câblée. Ce type d'inconvénients comme la durée de vie limitée, les pannes fréquentes et l'impossibilité d'apporter des modifications ou des améliorations aux systèmes.

Le remplacement de ces automatismes par les automates programmables industriels (API), offre un apport considérable à la conduite des processus de fabrication dans l'augmentation de la production, et de l'optimisation de la productivité.

Le thème qui nous a été proposé consiste à automatiser « la machine à fabriquer les cales d'encoches ».

En effet, notre travail a été de remplacer la logique câblée en une logique programmée par un automate programmable, de mettre un écran tactile à la place du pupitre existant (à entrées physique) pour un meilleur suivi et une meilleure commande de la machine et d'apporter quelque amélioration au système.

Pour se faire, nous avons organisé ce manuscrit comme suit :

- Chapitre I : Etude de la machine à fabriquer les cales d'encoches.
- Chapitre II : Modélisation de la machine à l'aide du GRAFCET.
- Chapitre III : automatisation de la chaîne par automate programmable industriel S7-300.
- Chapitre IV : Simulation avec le logiciel S7-PLCSIM et la configuration du panel tactile à l'aide du WinCC Flexible.

Nous terminons ce travail par une conclusion générale, bibliographie et annexes.

# **Chapitre I :**

## **Etude de la machine à fabriquer les cales d'encoches.**

## I. Présentation de l'organe d'accueil :

### I.1. Composition du complexe d'Electro-Industries :

L'entreprise est divisée en trois (03) unités, toutes situées sur un même site :

#### Ø Unité fabrication des transformateurs de basse et moyenne tension :

C'est le cœur de l'activité de l'entreprise. Elle produit des transformateurs de puissance variant entre 50 et 2000 KVA avec une capacité de production de 5000 transformateurs/an.

Cette unité dispose de l'ensemble des équipements nécessaire à la fabrication et aux essais des produits finis.

#### Ø Unité de fabrication de moteurs électriques, alternatives et montage de groupes :

Elle vient en second place en matière d'activité de l'entreprise, sa capacité de production est de :

- 45000 moteurs de 0,25 à 15KW.
- 4500 moteurs de 1,5 à 40KW.
- 2000 alternateurs de 16 à 180KVA.
- Montage de 400 groupes électrogènes (22, 35, 52KVA).

#### Ø Unité prestation technique chargée de la maintenance des équipements et installations pour les deux unités précédentes.

#### Machine électrique :

Une machine électrique est un dispositif électromécanique permettant la conversion d'énergie électrique en travail ou énergie mécanique; ce processus est réversible et peut servir à produire de l'électricité:

- Les machines électriques produisant de l'énergie électrique à partir d'une énergie mécanique sont appelées des génératrices, dynamos ou alternateurs suivant la technologie utilisée.
- Les machines électriques produisant une énergie mécanique à partir d'une énergie électrique sont appelées des moteurs.

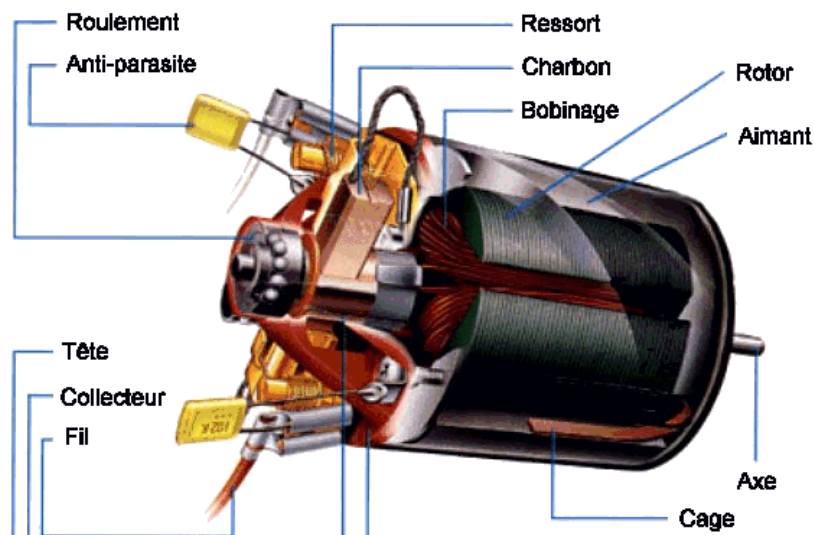


Figure I.1 : Machine électrique (moteur).

Les deux principales parties d'une machine électrique sont :

- Le rotor est la partie rotative d'une machine
- Le stator est la partie stationnaire d'un moteur électrique ou d'un alternateur. Selon la configuration de la machine, le stator peut créer un champ magnétique qui par interaction avec le champ magnétique rotorique produit le couple électromécanique. Il est constitué d'un empilage de tôles magnétique sur lesquelles est surmoulée la carcasse. Ces tôles comportent des encoches profondes dans lesquelles sont logés les bobinages. Un système d'isolement est nécessaire et qui est assuré par ce qu'on appelle « les cales ».

#### ✚ Les cales :

Les cales sont des feuilles isolantes, qui servent à fermer les encoches du stator de la machine électrique et d'isoler les différentes phases des bobines des encoches. Ces dernières sont produites par la machine qui fait l'objet de notre étude.



Figure I.2 : Cales d'encoche d'un stator.

II. Description de la machine à fabriquer les cales d'encoches :

La machine se compose de différentes unités, comme il est illustré sur la figure suivante :

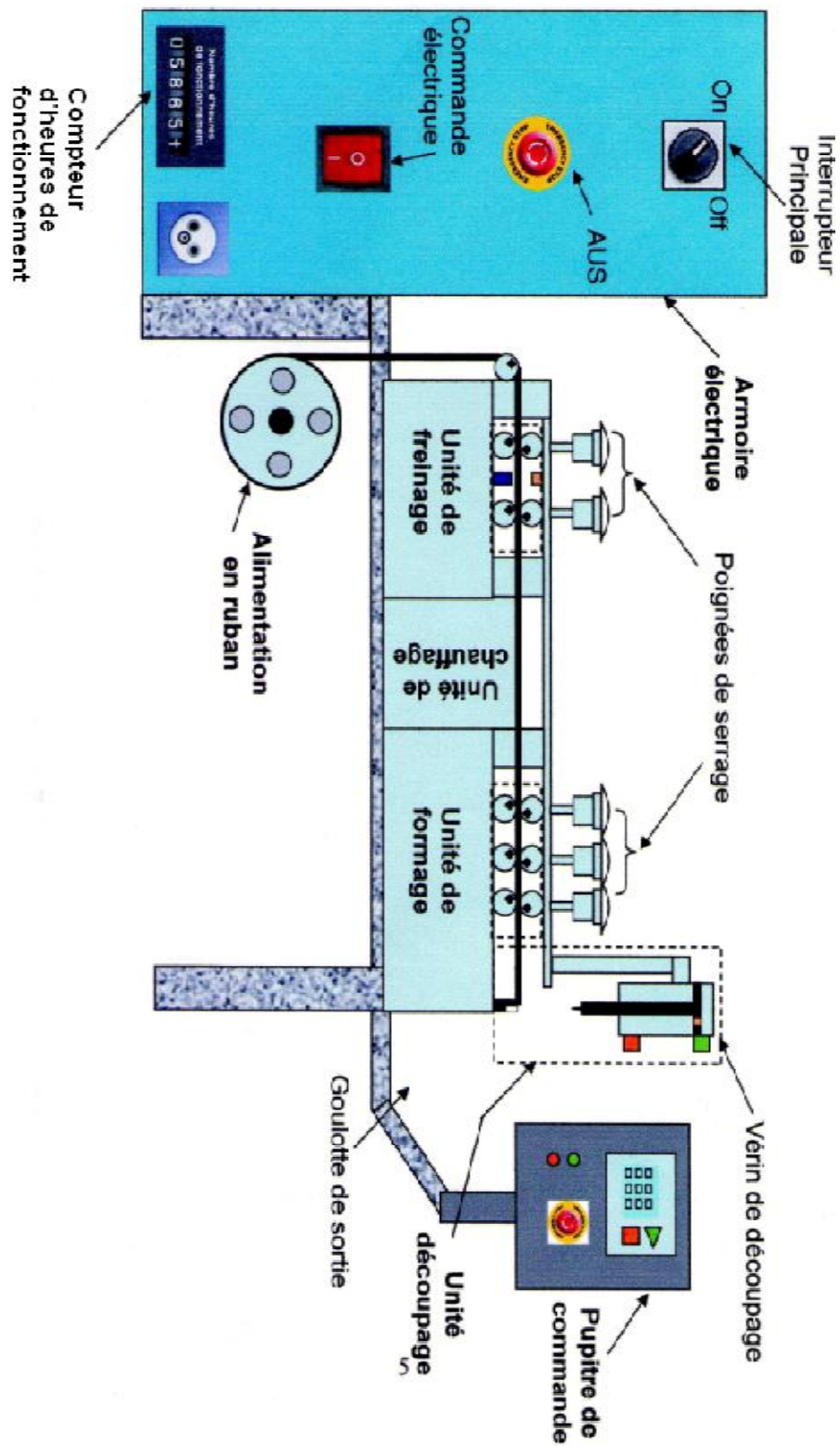


Figure 3 : Machine à fabriquer les cales d'encoches.

## II.1. Unités de la machine à fabriquer les cales d'encoches :

### II.1.1. Unité bobine à ruban :

C'est la source d'alimentation en ruban. Elle comporte :

- Un bouton d'arrêt du tambour.
- Un disque frontal
- Une poignée étoile, avec laquelle se fait le réglage des disques selon la largeur des cales d'encoches.

Pour alimenter le tambour d'un nouveau rouleau de ruban pour cales, on procéde comme suite :

- Débloquer le bouton d'arrêt du tambour et remonter le disque frontal.
- Verrouiller le bouton d'arrêt.
- Régler les disques à l'aide de la poignée étoile selon la largeur des cales d'encoches.

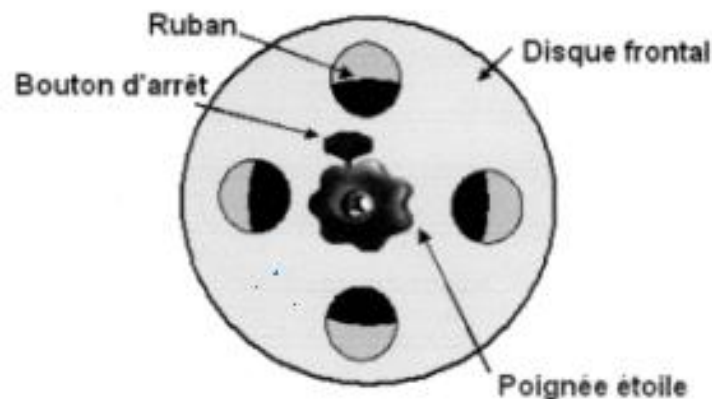


Figure I.4: bobine a ruban.

### II.1.2. Unité de freinage :

Elle est destinée à maintenir le ruban tendu pendant le découpage. Elle comporte :

- Un capot de sécurité.
- Des poignées étoiles, servent à varier la pression (serrage ou desserrage des poignées).
- Un rouleau de pression.
- Un levier d'interrupteur de fin de course, démontable de façon à introduire le ruban puis le poser sur ce dernier. Il nous renseigne sur la présence ou l'absence du ruban.
- Un dispositif d'arrêt.

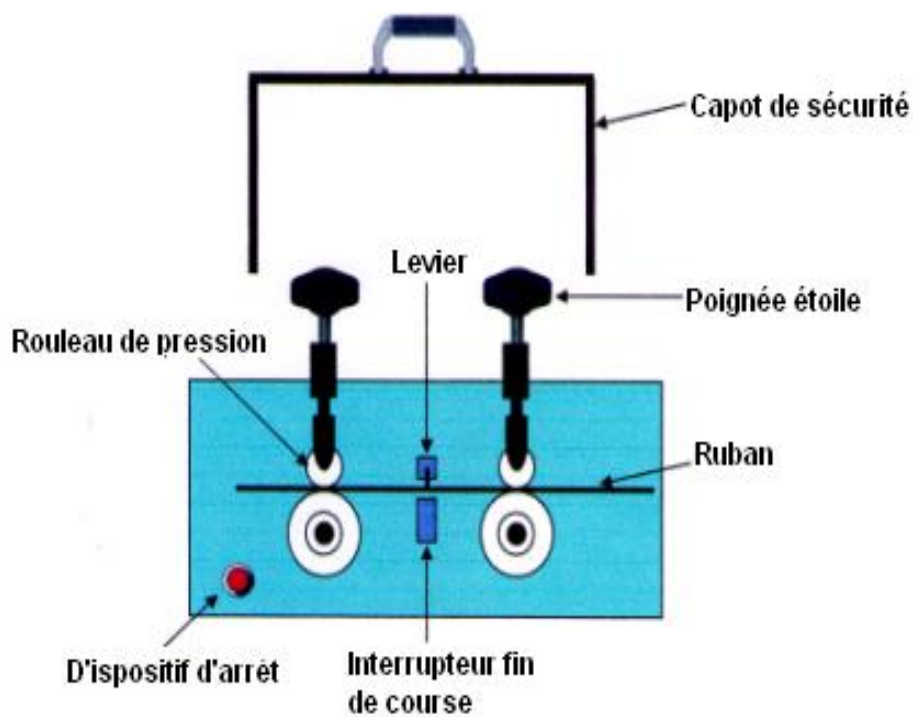


Figure I.5 : Unité de freinage.

### II.1.3. Unité de chauffage :

Elle est destinée à chauffer le ruban par des résistances chauffantes, afin de lui offrir une certaine élasticité et faciliter sa mise en forme. Elle comporte :

- Un capot de sécurité.
- Une partie supérieure, où le ruban passe.
- Un régulateur de température (thermostat), qui permet de garder une température constante.

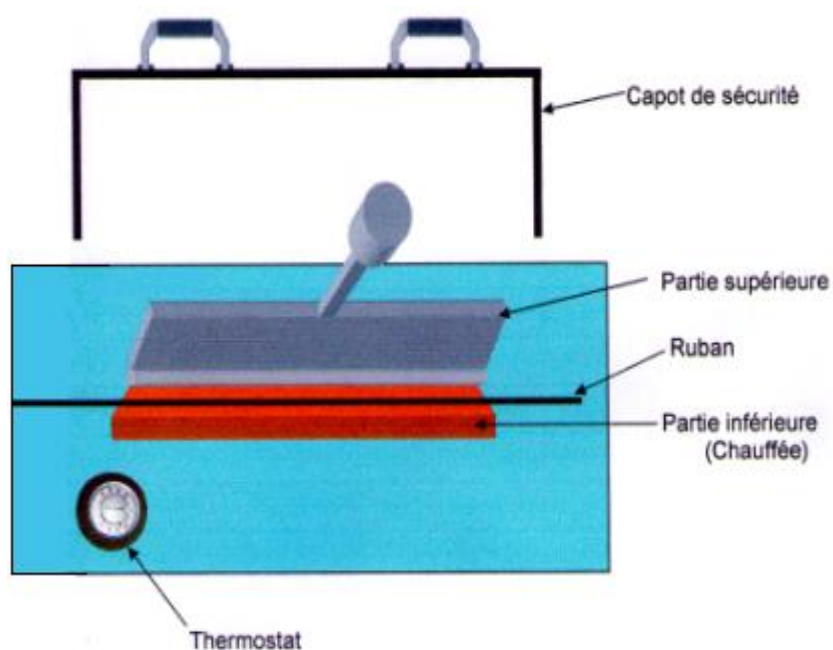


Figure I.6 : Unité de chauffage.

#### II.1.4. Unité de mise en forme :

L'unité de mise en forme donne la forme d'un « U » au ruban, de manière à obtenir des cales d'encoches appropriées aux encoches des stators des moteurs à fabriquer. Elle comporte :

- Un capot de sécurité.
- Un moteur continu, qui est solidaire aux roues de mise en forme inférieures par l'intermédiaire d'un réducteur (système mécanique à engrenage), entraînant ainsi l'avance du ruban vers l'unité de découpage.
- Un capteur rotatif (encodeur), solidaire à l'arbre du moteur continu. Il sert à donner des impulsions qui se traduiront en longueur de cale.
- Roues de mise en forme, positionnées par paires :
  - une paire de roues de préformage.
  - Une paire de roues principales.
  - Une paire de roues de finition.
- Languettes de guidage, qui est ajusté selon la largeur du ruban.
- Poignées étoiles, servent à varier la force de pression des roues de mise en forme.
- Les écrous de fixation.

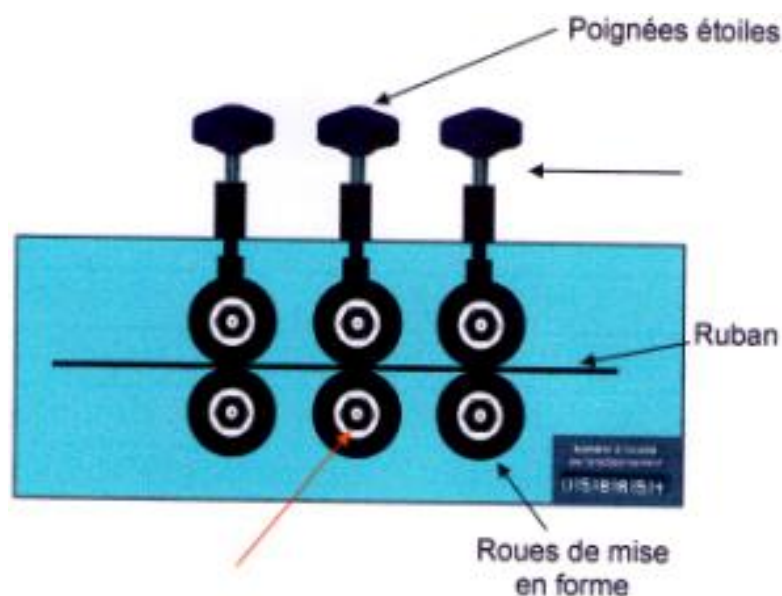
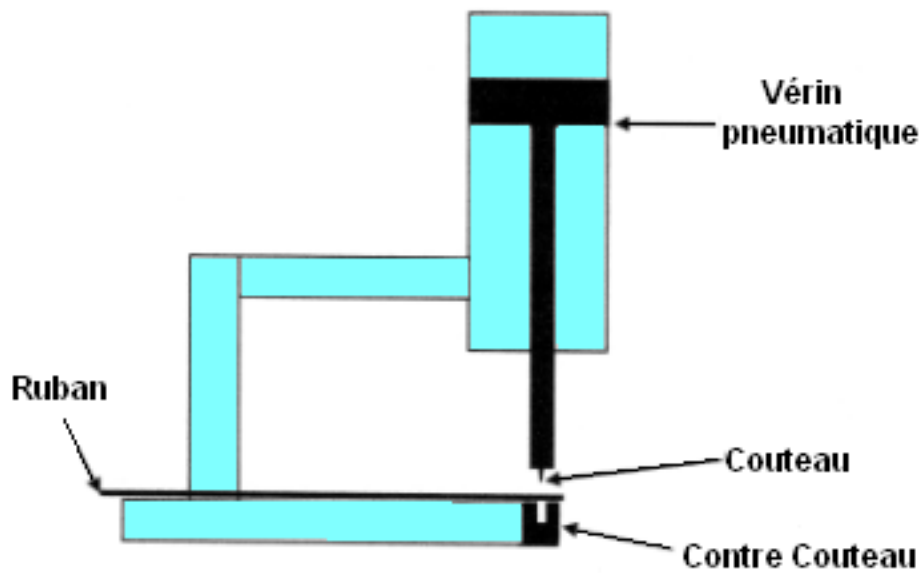


Figure I.7 : Unité de mise en forme.

**II.1.5. Unité de découpage :**

Elle est destinée à découper le ruban sur la longueur présélectionnée des cales. Elle comporte :

- Un vérin pneumatique.
- Un couteau.
- Un contre-couteau.



**Figure I.8 : Unité de découpage.**

### II.1.6. Panneau de commande :

Il comprend les boutons de commande suivants :

- Sélecteur Réglage-Auto : choisit le mode de travail de la machine (Manuel ou Automatique).
- Présélecteur longueur des cales : sélectionne la longueur de la cale souhaitée.
- Avance : commande le moteur manuellement.
- Bouton d'arrêt d'urgence.
- Boutons Retour et Découpe : commande manuelle le vérin.
- START : démarre le cycle automatique.
- STOP : met fin au cycle automatique.



Figure I.9 : Panneau de commande.

### II.1.7. Armoire électrique :

Elle comporte les circuits de commande et d'alimentation, ainsi qu'un bouton de mise sous ou hors tension et un bouton d'arrêt d'urgence.

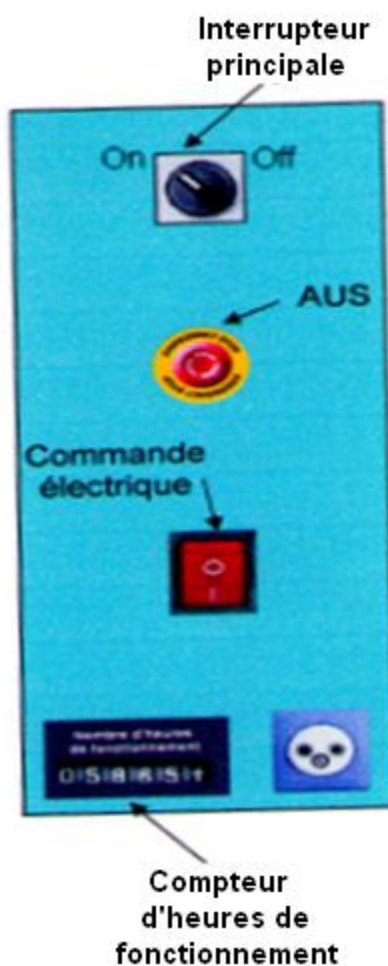
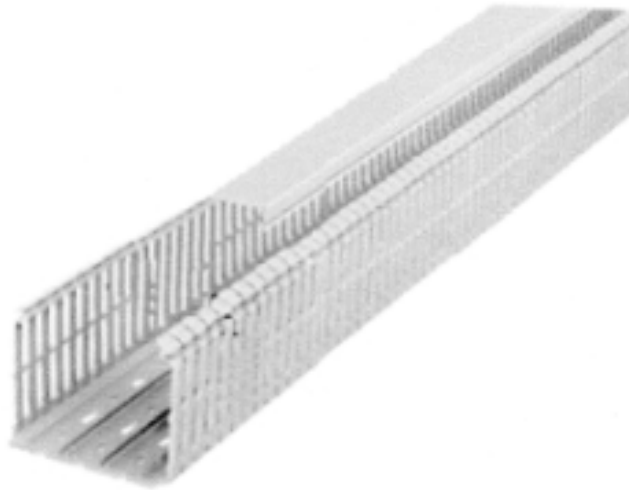


Figure I.10 : Armoire électrique.

**II.1.8. Goulotte de sortie :**

Elle est destinée à trier les cales selon leur longueur.



**Figure I.11 : Goulotte de sortie.**

**Remarque :**

- Les unités de freinage, de chauffage, de mise en forme et de découpage sont protégées par des capots protecteurs dans les zones dangereuses.
- En position automatique et manuelle, les capots protecteurs des unités de freinage et de mise en forme doivent être fermés.

## II.2. Partie opérative :

Elle comporte le processus à automatiser. Elle comprend :

- Les pré-actionneurs.
- Les actionneurs.
- Les capteurs.

### II.2.1. Pré-actionneurs :

Un pré-actionneur permet de distribuer de l'énergie à un actionneur, ce dernier convertissant l'énergie reçue en énergie utile. On citera le distributeur pneumatique et l'électrovanne « tout ou rien ».

#### a- Distributeur pneumatique :

Les distributeurs pneumatiques sont les pré-actionneurs privilégiés des actionneurs pneumatiques (vérins). A chaque type d'actionneur correspond un distributeur pneumatique que l'on caractérise par :

- son dispositif de commande (mécanique, électrique ou pneumatique)
- sa stabilité (monostable ou bistable)
- le nombre d'orifices de passage de fluide qu'il présente dans chaque position.

Dans notre machine, le vérin de découpage est un vérin à double effet et comporte donc deux orifices. On utilise un distributeur 4/2 comportant 4 orifices et deux chambres, actionné par une électrovanne et un retour à ressort. Représentée sur la figure suivante :

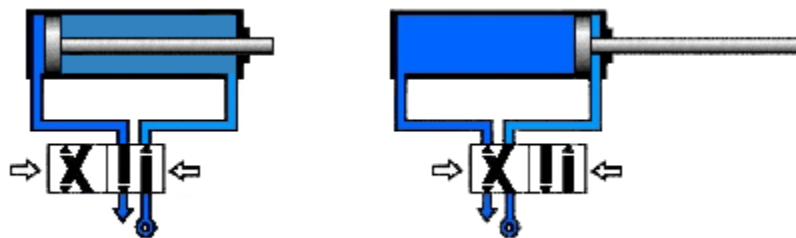


Figure I.12 : Distributeur 4/2.

#### b- Electrovanne « tout ou rien » :

L'électrovanne est un dispositif commandé électriquement permettant d'autoriser ou d'interrompre par une action mécanique la circulation de l'air dans un circuit pneumatique.

Les électrovannes dites de « tout ou rien » ne peuvent s'ouvrir qu'en entier ou pas du tout.

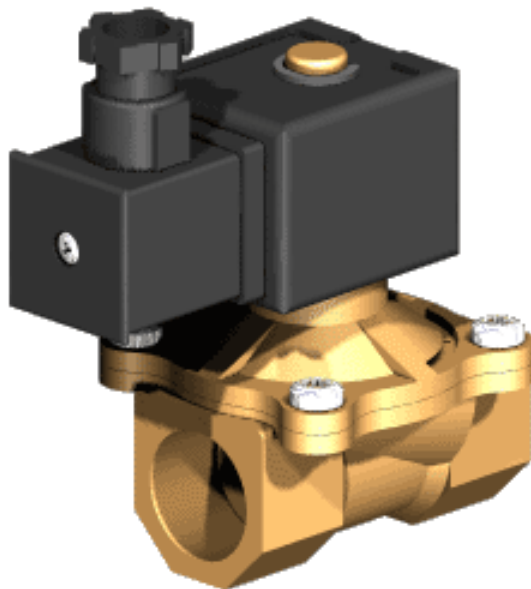


Figure I.13 : Electrovanne.

## II.2. Les actionneurs :

La fonction globale d'un actionneur est de convertir une énergie d'entrée, disponible sous une certaine forme, en une énergie de sortie utilisable pour obtenir un effet cherché. On citera le vérin pneumatique a double effet.

### - Vérin pneumatique a double effet :

Un vérin est dit double effet quand l'air comprimé permet de sortir la tige et de la rentrer. On l'utilise quand on a besoin d'un effort important dans les deux sens de déplacement.

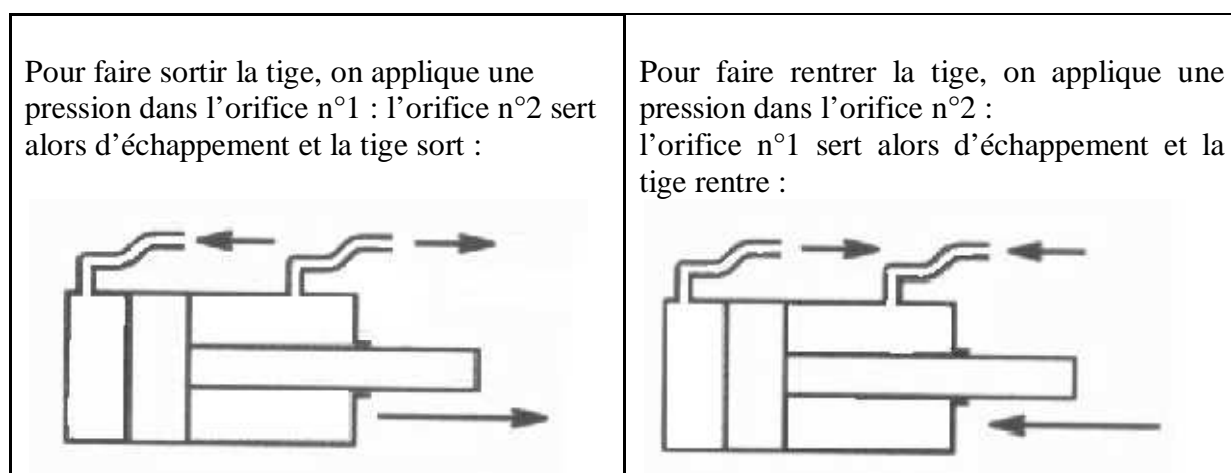


Figure I.14 : Vérin pneumatique a double effet.

- ✓ on le retrouve dans l'unité de découpage. Une fois la longueur sélectionnée atteinte, la bobine de l'électrovanne est excitée. Ce qui conduira à une alternance de pression sur les deux orifices de vérin, et ça descend de façon à découper le ruban. La position basse du vérin désactive la bobine pour un retour à l'état initial par effet du ressort.

### II.2.3. Les capteurs :

Les capteurs sont des composants d'automatisme qui ont pour but de récolter une information sur la partie opérative et de la retransmettre à la partie commande qui pourra ainsi la traiter.

#### a. Capteur de proximité inductif :

La détection se fait sans contact. Un circuit électronique à effet inductif transforme une perturbation magnétique due à la présence de l'objet en commande d'ouverture ou de fermeture statique (par transistor) du circuit d'information. La face sensible crée un champ magnétique local.

Lorsque l'objet pénètre dans le champ magnétique, l'oscillateur se met en route et la sortie est activée.

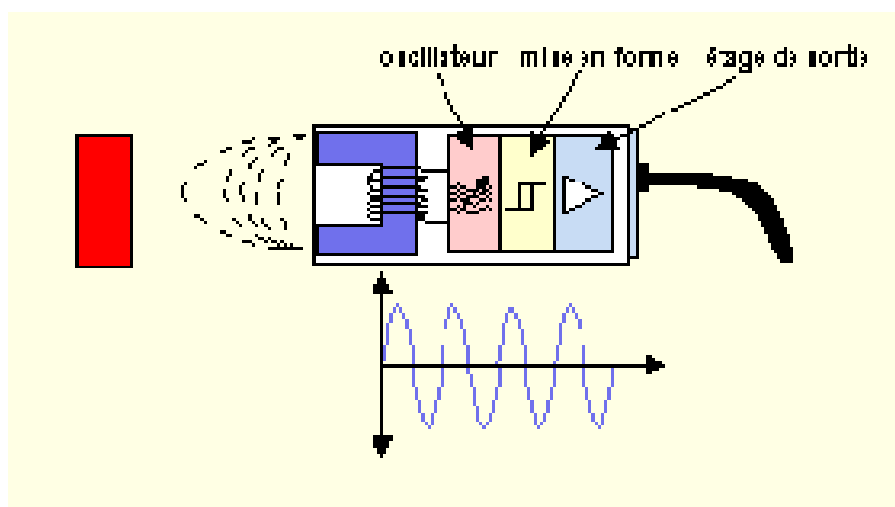


Figure I.15 : Capteur de proximité inductif

- ✓ On retrouve ces capteurs sous les capots de sécurité de l'unité de freinage, celle de mise en forme ainsi que dans le levier de l'unité de freinage. Ils nous renseignent sur l'ouverture ou fermeture des capots de sécurité et l'absence ou présence du ruban. Ces conditions sont nécessaires au démarrage du moteur.

#### b. Interrupteurs à lames souples :

Un interrupteur à lame souple est constitué d'un corps à l'intérieur duquel est placé un contact électrique métallique souple sensible aux champs magnétiques.

Lorsqu'un champ magnétique est dirigé sur la face sensible du capteur, le contact s'établit entre les deux bornes du capteur.

Ce type de détecteurs est souvent monté directement sur le corps de vérins en tant que fin de course (dans ce type de montage, le piston du vérin est magnétisé).



**Figure I.16 : Interrupteur à lames souples.**

**c. Thermostat :**

Un thermostat est un système électronique à sonde thermique. Il comporte un potentiomètre pour sélectionner une température de consigne.



**Figure I.17 : Thermostat.**

- ▼ Dans l'unité de chauffage, une fois arrivé à la température souhaitée le contact du thermostat s'ouvre, la bobine du contacteur relié aux résistances chauffantes est désexcitée, et alors le ruban n'est plus chauffé. Sinon (le cas où la température est au-dessous de la consigne), le contact se ferme, la bobine s'excite et alors le ruban est chauffé.

**d. Encodeur rotatif :**

Le codeur rotatif est un capteur de position angulaire. Lié mécaniquement à un arbre qui l'entraîne, son axe fait tourner un disque qui lui est solidaire. Ce disque comporte une succession de parties opaques et transparentes.

Une lumière émise par des diodes électro luminescentes, (DEL) traverse les fentes de ce disque et crée sur les photodiodes réceptrices un signal analogique. Une interface électronique (qui est incluse dans le codeur) amplifie ce signal puis le convertit en signal carré qui est alors transmis à un système de traitement (généralement un API).



**Figure I.18 : Encodeur rotatif**

- ▼ L'encodeur de notre machine est solidaire à l'arbre de rotation du moteur, il se charge de délivrer des impulsions générées à partir du moteur. Ces dernières seront traitées directement par des cartes électroniques adéquates à l'encodeur. Leur conversion d'un signal analogique en numérique permet la commande du déplacement du ruban.

#### **II.2.4. Les circuits électriques de protection :**

Ils servent d'une part à la protection des composants électriques, et d'une autre à l'isolement des parties puissance et commande. On a :

##### **a. Les circuits RC :**

La mise hors tension de la bobine d'un contacteur provoque des surtensions. En effet l'inductance de la bobine cause un maintien du flux électrique au moment de la coupure et fermeture du circuit de courant de fait de la capacité propre de la bobine.

Communément, on atténue donc les surtensions de commutation des bobines de contacteur en raccordant des organes de protection telle que les circuits RC.

Les organes de circuits sont essentiellement raccordés aux contacteurs à l'alimentation en courant alternatif. Ils peuvent également être raccordés aux contacteurs à l'alimentation en courant continu. Ils sont montés en parallèle à la bobine du contacteur.

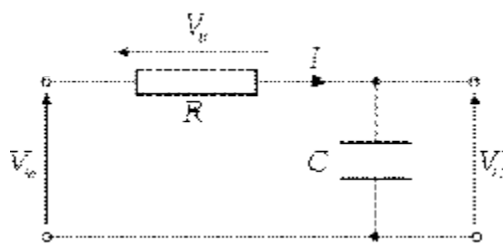


Figure I.19 : Circuit RC.

### b. Les fusibles :

La fonction du fusible est d'assurer la protection des circuits électriques contre les courts-circuits.

Son principe est le suivant, lorsque le courant demandé par le circuit électrique dépasse le calibre de fusible, la partie conductrice intérieure fond et ouvre ainsi le circuit.

### c. Les contacteurs :

Le contacteur est un appareil de commande capable d'établir ou d'interrompre le passage de l'énergie électrique. Il assure la fonction COMMUTATION

Lorsque la bobine située dans le circuit de commande est parcourue par un courant électrique celle-ci est excitée : le circuit magnétique attire les contacts (mobiles sur les fixes) de puissance et de commande. La charge est alimentée.

Lorsque la bobine n'est plus alimentée, les contacts s'ouvrent sous l'action du ressort de rappel : la charge n'est ainsi plus alimentée.

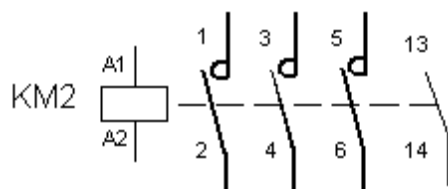


Figure I.20 : Contacteur.

### d. Transformateur galvanique (220/220 V) :

Il est utilisé pour la sécurité du matériel. Il permet l'isolation électrique du circuit de commande du réseau. Les surtensions ou les défauts qui peuvent survenir au niveau du réseau sont subits uniquement par le transformateur.

## II.2.5. Variateur de vitesse électronique :

Un variateur de vitesse est un dispositif électronique destiné à commander la vitesse d'un moteur électrique à l'aide d'un potentiomètre ou on ajuste la vitesse de consigne.

- ✓ Le système fonctionne à une vitesse constante tout au long du cycle de fabrication.

### **II.2.6. Moteur à courant continu :**

Le moteur à courant continu de l'unité de mise en forme, sert à entraîner les roues de l'unité mise en forme à l'aide du réducteur. Il ne peut y avoir démarrage du moteur que si l'entrée de déblocage du variateur de vitesse relié au moteur est activée.

### **II.2.7. Résistances chauffantes :**

C'est un ensemble de quatre résistances de 600 watts alimentées par 220 V.

- ✓ On les retrouve dans l'unité de chauffage et servent à chauffer le ruban pour le rendre plus élastique et facilite à être mise en forme.

## **II.3. Partie commande :**

### **II.3.1. Pupitre de commande :**

La partie commande et la partie opérative ne peuvent dialoguer que si un opérateur provoque le démarrage de la machine. D'une part, l'opérateur a besoin de savoir si le processus se déroule correctement, il peut alors être amené à agir en de mauvais fonctionnement (à l'aide du bouton d'arrêt d'urgence), ou bien d'assurer certain réglage préalables avant le démarrage de la machine.

Tout ceci s'effectue au niveau du pupitre ou se situent tous les boutons de commande.

### **II.3.2. Commande de la machine :**

La machine à fabriquer les cales d'encoches, peut être réglé sur deux modes :

a) Mode réglage (Manuel) :

Le mode réglage est choisi en actionnant le commutateur sélecteur sur réglage. L'opérateur fait le réglage des différents organes de la machine. Ce mode est aussi choisi lorsqu'il y a changement de la bobine à ruban.

b) Mode automatique :

Le mode automatique est choisi en actionnant le commutateur sur l'automatique. L'intervention de l'opérateur n'est donc pas nécessaire, et le cycle de production s'effectue automatiquement et ne s'arrête que par l'absence du ruban ou par un STOP cycle ou encor un arrêt simple ou celui d'urgence.

## **III. Fonctionnement de la machine à fabriquer les cales d'encoches :**

De l'unité de bobine d'alimentation, le ruban isolant est entraîné vers l'unité de freinage, qui grâce à son système, tend le ruban de façon à faciliter sa manipulation.

L'unité de chauffage donne au ruban l'élasticité nécessaire pour la mise en forme, et ceci par une température constante maintenue par un thermostat existant.

Les trois roues de mise en forme inférieures sont reliées entre elles par une chaîne tendue par deux roues tendeuse.

L'axe de la roue de mise en forme centrale est entraîné par un moteur à courant continu, au moyen d'une courroie dentée et d'un réducteur à roues coniques.

C'est en effet ces roues dont le mouvement est engendré par le moteur, qui entraînent le ruban d'une extrémité à une autre, c'est à dire de l'unité de la bobine d'alimentation à l'unité de mise en forme.

Elle est équipée aussi d'un capteur rotatif ou encor l'encodeur rotatif dans l'arbre est solidaire à celui du moteur. Son principal rôle est de délivrer des impulsions qui seront traitées par des cartes électroniques, de façon à traduire le signal analogique en un signal numérique et ainsi de commander le déplacement du ruban.

Chaque fois que la longueur présélectionnée est atteinte, le moteur s'arrête et le ruban est coupé dans l'unité de découpage, et le cycle se répète et ne s'arrête que lorsqu'il n'y a plus de détection de ruban, par le capteur magnétique inductif. De ce fait la machine coupera une dernière fois et s'arrêtera jusqu'à nouvel ordre de façon à reproduire le même cycle.

Le cycle s'arrêtera aussi après avoir actionnée l'un des boutons suivants.

- Bouton stop cycle.
- Bouton arrêt.
- Bouton arrêt d'urgence.

- ✓ En vue de répondre aux exigences du système en termes de précision, de sécurité et d'un meilleur contrôle et commande de la machine à fabriquer les cales d'encoches, une politique d'automatisation serait opportune.

### **Conclusion :**

Devant la complexité du système et les techniques utilisées, l'étude de la machine nous a permis de bien comprendre et d'assimiler au mieux le fonctionnement et la succession des étapes, qui forment le processus de fabrication des cales d'encoches.

Ainsi on a estimé que dans un premier temps, il serait alors nécessaire d'élaborer une modélisation plus explicite pour notre système. Et qui fera donc l'objet du chapitre suivant.

# **Chapitre II :**

## **Modélisation de la machine à l'aide du GRAFCET.**

## I. Introduction :

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

## II. Définition du GRAFCET :

Le GRAFCET (GRAphe Fonctionnel de Commande des étapes et Transitions) permet l'établissement des descriptions de la fonction et du comportement des systèmes de commandes en établissant une représentation graphique indépendante de la réalisation technologique.

Le GRAFCET est destiné à représenter des automatismes logiques (ou discrets), c'est à dire des systèmes à événements discrets dans lesquels les informations sont de type booléennes ou peuvent s'y ramener (par exemple la comparaison d'une température avec un seuil). Le GRAFCET est utilisé généralement pour spécifier et concevoir le comportement souhaité de la partie commande d'un système, mais il peut également être utilisé pour spécifier le comportement attendu de la partie opérative ou bien de tout le système de commande.

## III. Composition du GRAFCET :

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

### III.1. Les étapes :

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

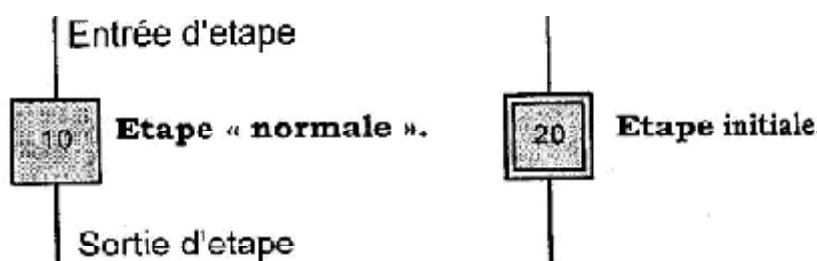


Figure II.1 :Les étapes.

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

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

$X_i = 1$  si  $i$  est active  
 Remarquez, que l'on remplace le numéro d'étape par un  $X$  étant le symbole d'une étape.  
 $X_i = 0$  si  $i$  est inactive dans le cas présent.



**Figure II.2 : Etape active et étape inactive.**

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

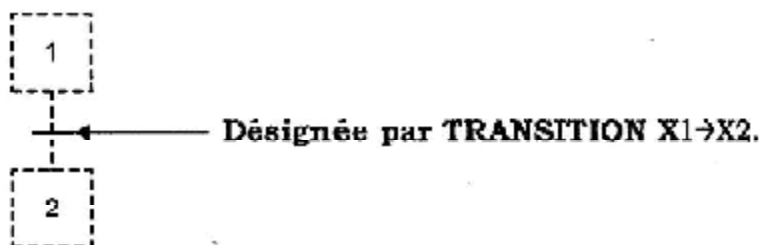
Les actions sont décrites de façon littérale ou symbolique à l'intérieur d'un ou plusieurs rectangles reliés au symbole de l'étape à laquelle elles sont associées.



**Figure II.3 : les actions.**

### III.2. Les transitions :

Caractéristique essentielle, elle indique la possibilité d'évolution entre étapes. La transition peut être considérée comme une porte entre deux étapes, et la réceptivité comme la clé ou le code nécessaire pour ouvrir la porte. La représentation est comme suit :



**Figure II.4 : une transition.**

La réceptivité associée à une transition est une fonction logique des entrées, des variables auxiliaires et/ou de l'activité d'étapes du GRAFCET. Elle peut s'écrire sous forme littérale ou sous forme logique (expression booléenne). La réceptivité regroupe toutes les conditions et uniquement celles qui sont nécessaires au franchissement de la transition. Une réceptivité est dite vraie si la condition ou l'équation booléenne, associée est vérifiée et égale à 1, et inversement.

Les réceptivités associées sont décrites de façon littérale ou symbolique à droite de la transition à laquelle elles sont associées.

1<sup>er</sup> cas :

La réceptivité est une variable ou une fonction logique de variables d'entrée issues de capteurs, boutons poussoirs etc. ...

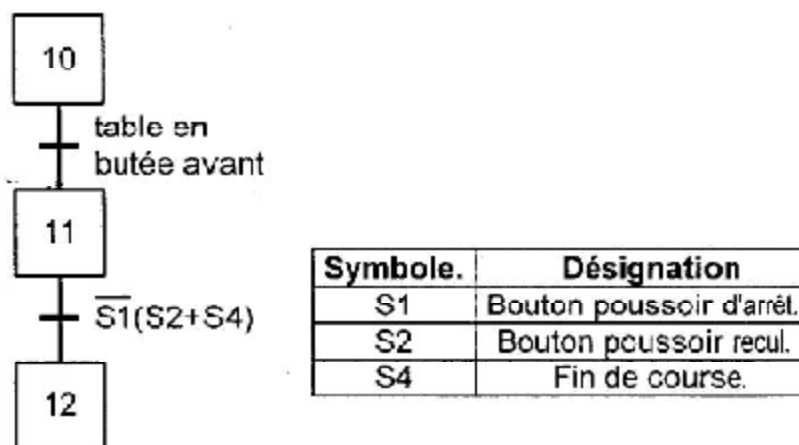


Figure II.5 : 1<sup>er</sup> cas de la réceptivité.

2<sup>ème</sup> cas :

La réceptivité est une variable ou une fonction logique de variables d'entrée issues de l'activité d'une ou plusieurs étapes de GRAFCET.

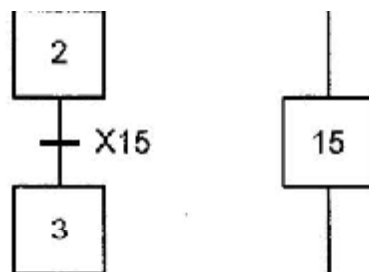


Figure II.6 : 2<sup>ème</sup> cas de réceptivité.

### III.3. Les liaisons orientées :

Les liaisons relient les étapes aux transitions et les transitions aux étapes.

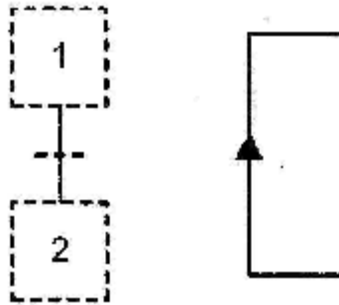


Figure II.7 : les liaisons orientées.

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

Véritable code de la route du GRAFCET, les règles d'évolution réglementent la situation initiale, le franchissement des transitions, l'évolution des étapes actives, les évolutions simultanées, l'activation, la désactivation.

### III.4. Les actions :

L'action associée à l'étape peut être de 3 types : continue, conditionnelle ou mémorisée.

- **Actions continues :**

La ou les sorties correspondant à l'ordre A sont mises à 1 tant que l'étape associée est active. Lorsque l'étape devient inactive, la ou les sorties sont mises à 0.

- **Actions conditionnelles :**

Une action conditionnelle n'est exécutée que si l'étape associée est active et si la condition associée est vraie. Elles peuvent être décomposées en 3 cas particuliers :

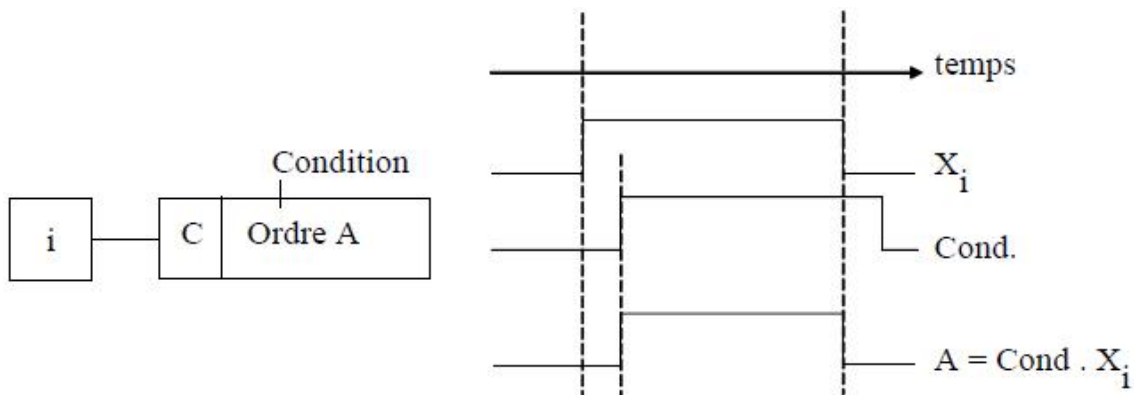
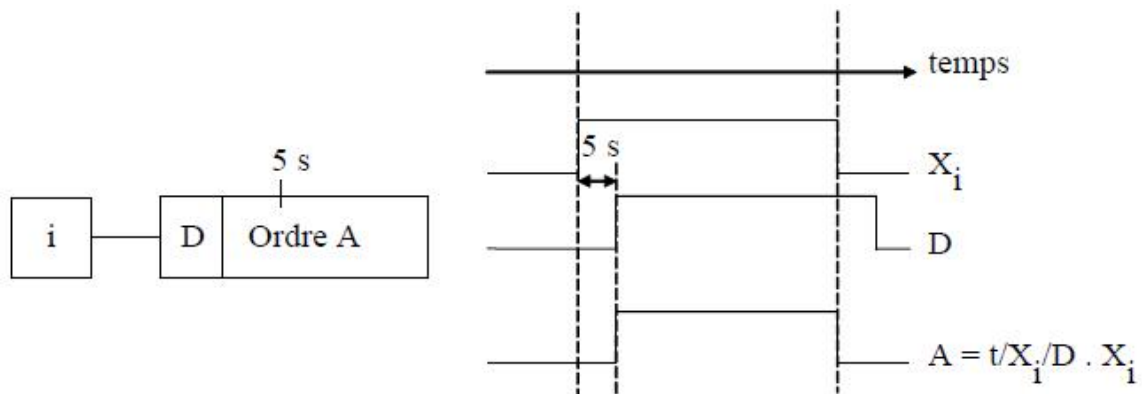
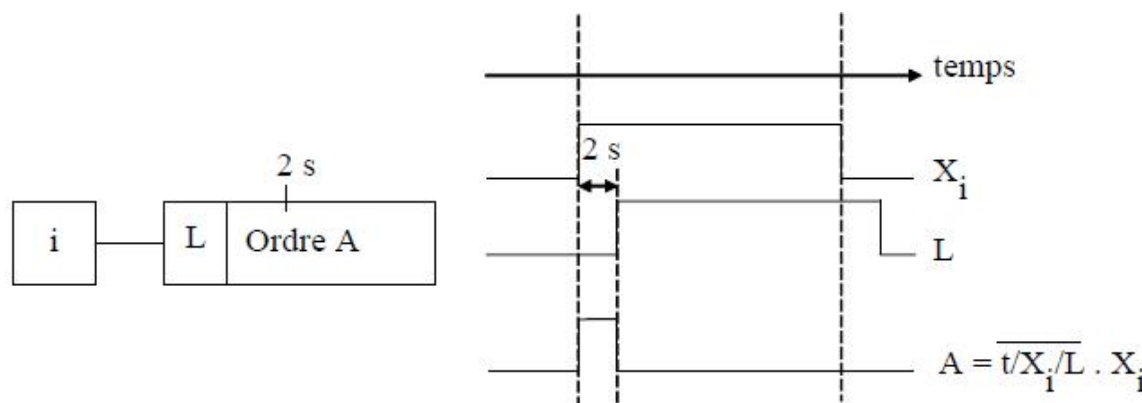


Figure II.8 : Action conditionnelle simple : Type C.



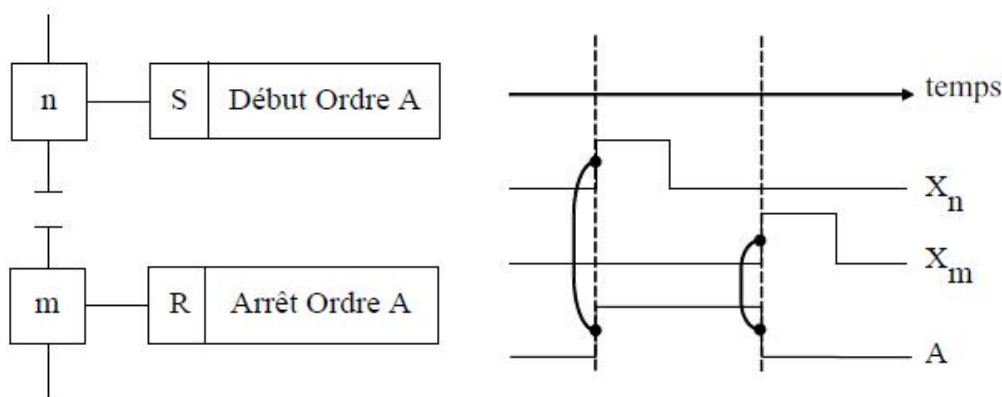
**Figure II.9 : Action retardée : Type D (Delay).**

D est la durée de la temporisation, c'est à dire le retard mis pour passer à 1 après que Xi soit passé à 1.



**Figure II.10 : Action limitée dans le temps : Type L (limited).**

- **Actions mémorisées :**



**Figure II.11 : Action mémorisée.**

On peut ainsi donner l'équation d'un ordre A en fonction des états des étapes, des conditions éventuelles et du temps.

### III.5. Les règles d'évolution du GRAFCET

#### Règle N°1 : Condition initiale

A l'instant initial, seules les étapes initiales sont actives.

#### Règle N°2 : Franchissement d'une transition

Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, ET seulement si la réceptivité associée est vraie.

#### Règle N°3 : Evolution des étapes actives

Le franchissement d'une transition entraîne obligatoirement dans cet ordre la désactivation de toutes ces étapes amont et l'activation de ses étapes aval.

#### Règle N°4 : Franchissement simultané

Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.

#### Règle N°5 : Conflit d'activation

Si une étape doit être simultanément désactivée par le franchissement d'une transition aval, et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes au procédé) non désirées.

### III.6 Niveau d'un GRAFCET :

- **GRAFCET de niveau 1 :**

Appelé aussi niveau de la partie commande, il décrit l'aspect fonctionnel du système et les actions à faire par la partie commande, en réaction aux informations provenant de la partie opérative. Les réceptivités sont décrites en mots et non en abréviations.

- **GRAFCET de niveau 2 :**

Appelé aussi niveau de la partie opérative, il tient compte de plus de détails des actionneurs, des pré-actionneurs et des capteurs. La représentation des actions et réceptivités est écrite en abréviations et non en mots.

### IV. Création et simulation du GRAFCET :

Après avoir choisi de modéliser le système par le biais du GRAFCET, nous avons estimé préférable de pouvoir créer le GRAFCET, et en suite le simuler de façon à se rapporter à la réalité du fonctionnement. Pour ce faire, nous avons utilisé le logiciel Automgen 7.

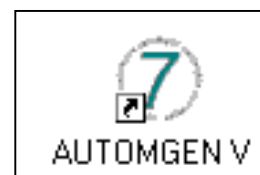
### V. Définition du logiciel Automgen V7 :

Automgen V7 est un logiciel de conception d'automatisme, propriétaire édité par la société française IRAI. Fonctionnant sur un PC sous système d'exploitation Windows, il est utilisé pour l'apprentissage des automatismes et dans l'industrie pour le développement d'applications.

Automgen V7 peut travailler avec plusieurs outils de représentation graphiques, comme les logigrammes, Ladder...etc...et bien sur le GRAFCET, outil avec lequel on va travailler pour réaliser la modélisation du dispositif.

#### V.1. Lancement d'automgen :

Pour démarrer Automgen, on clique deux fois sur l'icone



#### V.2. Le projet :

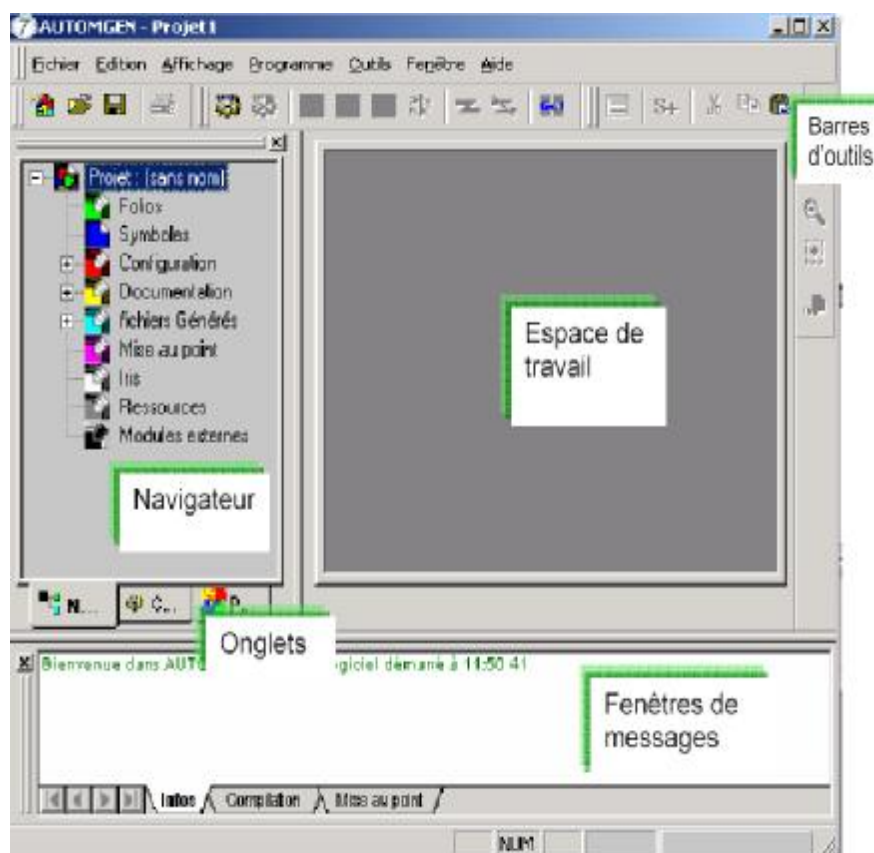
Un projet regroupe l'ensemble des éléments composant une application (folios, symboles, configuration, objets IRIS, etc ...).

<p>Pour créer un nouveau projet</p> 	<p>Pour ouvrir un projet existant</p> 	<p>Pour sauvegarder un projet</p> 
---	---	---

### V.3. La vue générale :

Élément central de la gestion des applications, le navigateur permet un accès rapide aux différents éléments d'une application :

- *folios*,
- *symboles*,
- *configuration*,
- *documentation*,
- *objets IRIS*,
- *etc ...*



### V.4. La cible :

La cible est associée à l'automate utilisé.

L'onglet "Cibles" permet d'accéder à la liste des post-processeurs installés.

La cible active est marquée d'une coche verte.

Pour modifier la cible courante, double cliquez sur la ligne correspondante

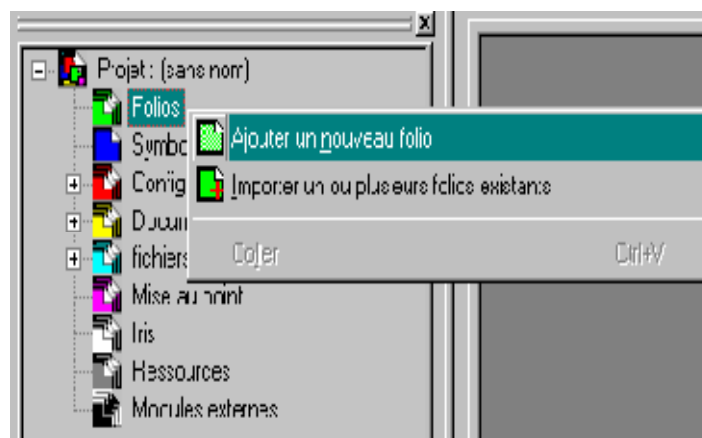


## V.5. Les folios :

Un folio est une page sur laquelle est dessinée un grafcet ou une partie de grafcet

Pour ajouter un nouveau folio, dans le navigateur :

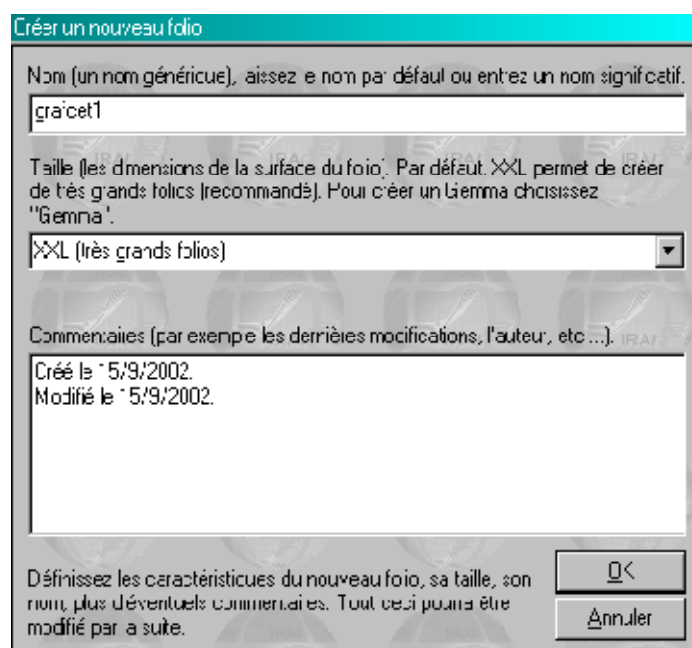
- Pointer avec la souris l'élément "Folios"
- et Cliquer avec le bouton droit
- puis choisissez "Ajouter un nouveau folio".



Le nom du folio peut être quelconque mais doit rester unique pour chaque folio du projet.

Choisissez la taille du folio (XXL est le format recommandé)

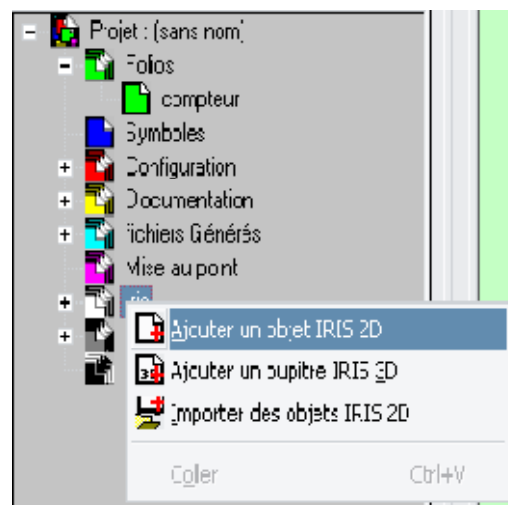
La zone commentaire est laissée à votre discrétion pour l'évolution des modifications ou autres informations relatives à chacun des folios.



## V.6. Iris :

permet de créer des objets 2D ou 3D de supervision et de simulation.

Une fois l'objet créé, on se doit alors de lui associer les boutons nécessaires à la commande du système et les configurer en utilisant les symboles adéquats du GRAFCET conçus.

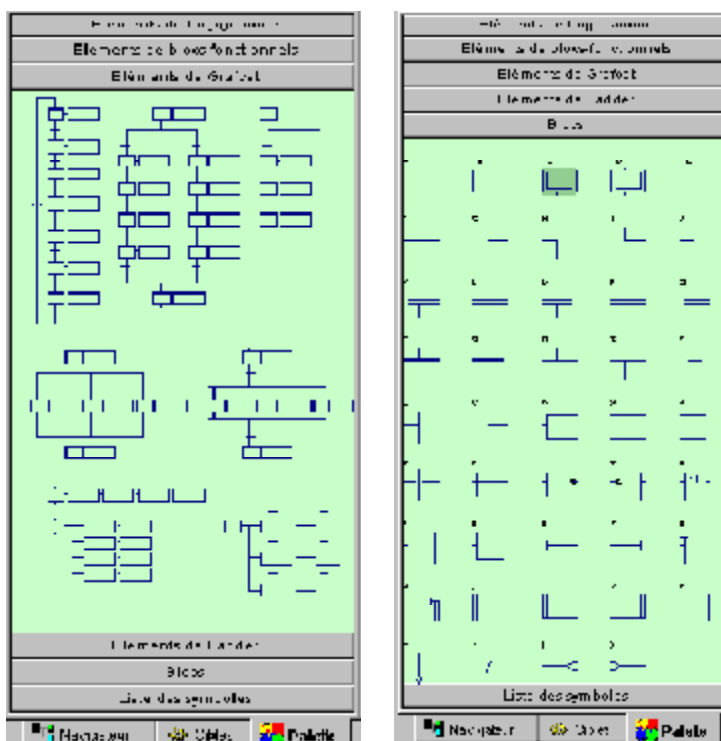


## V.7. Le choix de la palette :

Un onglet "Palette" permet d'accéder à des éléments de dessin de programmes.

La palette donne un ensemble d'éléments qui peuvent être sélectionnés et déposés sur les folios.

Pour dessiner les grafquets, nous utiliserons essentiellement les palettes "Blocs" et "Éléments de Grafcet".



## V.8. Exécution de l'application :

### V.8.1. Compilateur :

Le compilateur permet de localiser les erreurs, après un double clic sur le message d'erreur sur l'erreur.




### V.8.2. Exécution :

Le bouton GO représente la méthode la plus rapide pour observer le résultat de l'exécution d'une application. Il active le mécanisme suivant :

- Compilation de l'application si elle n'est pas à jour.
- Installation du module d'exécution.
- Passage de la cible en RUN.
- Activation de la visualisation dynamique.

Sinon on peut lancer par le bouton RUN, une fois qu'on a compilé le projet sans erreurs.

### Remarque :

L'activation de la visualisation dynamique se fait par le bouton  Au niveau de la visualisation en dynamique sur les folios :

- Vert correspond à l'état 0
- Jaune correspond à l'état 1

## **VI. Liste des actionneurs, pré-actionneurs et capteurs :**

### **VI.1. Les actionneurs et pré-actionneurs :**

Ø Moteur :

S EV0 : Démarrage du moteur.

R EV0 : Arrêt du moteur.

Ø Electrovanne :

S Y1 : Descente du vérin.

R Y1 : Remontée du vérin.

**Remarque :** Le S (Set) est pour l'excitation et le R (Reset) pour la d'excitation.

### **VI.2. Les capteurs :**

S0 : Capteur magnétique, détecte la position haute du vérin.

S1 : Capteur magnétique, détecte la position basse du vérin.

S2 : Capteur magnétique, détecte la fermeture du capot 1.

S3 : Capteur magnétique, détecte la fermeture du capot 2.

S4 : Capteur magnétique, détecte la présence du ruban.

### **VI.3. Les boutons :**

BM : Bouton de mise en marche de la machine.

BPM1 : Sélection du mode automatique.

BPM1 : Sélection du mode manuel.

BP0 : Bouton poussoir Début de cycle automatique.

BPAV : Bouton poussoir de moteur marche.

BPDEC : Bouton poussoir Descente vérin.

BPRET : Bouton poussoir Remontée vérin.

### **VI.4. Compteur :**

C0 : Compte le nombre de pièce découpée.

## **VII. Conclusion :**

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

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

# **Chapitre III :** automatisation de la chaine par automate programmable industriel S7-300.

## **I. Introduction aux automates programmables :**

Les automates programmables industriels API (programmable logic controller PLC), sont apparus vers 1969 aux Etats unis pour répondre aux besoins de l'industrie automobile. Ils ont été conçus pour l'automatisation des chaînes de fabrication et réaliser des fonctions logiques combinatoires et séquentielles en remplacement des armoires à relai très volumineuses.

Depuis, leur utilisation s'est largement répandue dans l'industrie, ou ils représentent l'outil de base de l'automatisation des systèmes de production. Les API ont trouvé leur place dans les domaines les plus variés comme dans les chaînes de fabrication (usinage, montage,...etc.), dans les opérations de manutention (stockage, chargement,...etc.), ou encore dans le système de contrôle (installation de climatisation, frigorifique, de chauffage, détection des incendies, industrie nucléaire,...etc.)

Avec le temps, on a trouvé sur le marché différentes variétés ; ceci est dû à la diversité des constructeurs. Pour la réalisation de notre travail on utilise l'automate SIMATIC S7-300 de SIEMENS.

### **I.1. Automates programmables industriels (API) :**

L'automate programmable industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande des pré-actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques.

A partir ces définitions, nous distinguerons dans les fonctions que l'automate doit remplir :

- un rôle de commande ou il est un composant d'automatisme, élaborant des actions, suivant un algorithme appropriée, à partir des informations que lui fournissent des détecteurs (Tout ou Rien) ou des capteurs (analogiques ou numériques) ;
  - un rôle de communication dans le cadre de la production :
- Avec des opérateurs humains : c'est le dialogue d'exploitation.
  - Avec d'autres processeurs, hiérarchiquement supérieurs (calculateur de gestion de production), égaux (autres automates intervenant dans la même chaîne) ou inférieurs (instrumentation intelligente).

Se poseront aussi, pour le rendre ou le faire demeurer opérationnel, les problèmes de la programmation et de la maintenance.

## I.2. Structure d'un système automatisé :

Tout système automatisé peut se décomposer selon le schéma ci-dessous :

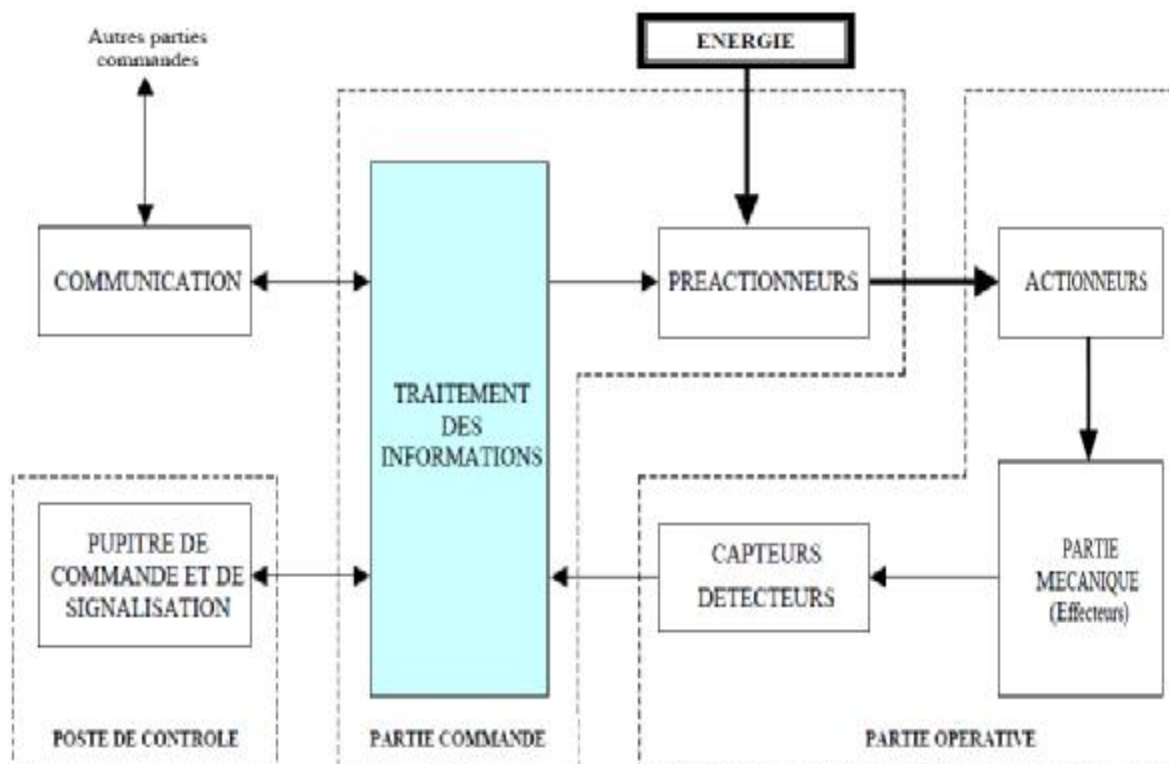


Figure III.1 : Structure d'un système automatisé.

- **Partie opérative :**

Elle agit sur la matière d'œuvre afin de lui donner sa valeur ajoutée.

Les actionneurs (moteurs, vérins) agissent sur la partie mécanique du système qui agit à son tour sur la matière d'œuvre.

Les capteurs / détecteurs permettent d'acquérir les divers états du système.

- **Partie commande :**

Elle donne les ordres de fonctionnement à la partie opérative.

Les pré actionneurs permettent de commander les actionneurs ; ils assurent le transfert d'énergie entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs. Exemple : contacteur, distributeur ...

Ces pré actionneurs sont commandés à leur tour par le bloc traitement des informations. Celui-ci reçoit les consignes du pupitre de commande (opérateur) et les informations de la partie opérative transmises par les capteurs / détecteurs.

En fonction de ces consignes et de son programme de gestion des tâches (implanté dans un automate programmable ou réalisé par des relais (on parle de logique câblée)), elle va commander les pré actionneurs et renvoyer des informations au pupitre de signalisation ou à d'autres systèmes de commande et/ou de supervision en utilisant un réseau et un protocole de communication.

- **Poste de contrôle :**

Composé des pupitres de commande et de signalisation, il permet à l'opérateur de commander le système (marche, arrêt, départ cycle ...).

Il permet également de visualiser les différents états du système à l'aide de voyants, de terminal de dialogue ou d'interface homme-machine (IHM).

### I.3. Choix d'un automate :

En principe, le choix se fait selon le cahier des charges du système étudié et en prenant compte des critères bien précis et importants tel que :

- Le nombre et la nature des entrée/sortie (numériques, analogiques).
- La nature de traitement (temporisation, comptage...etc.).
- Le dialogue (la console détermine le langage de programmation).
- La communication avec d'autres systèmes.
- Le moyen de sauvegarde de programme (diskette, disc amovible,...etc.).
- La fiabilité, la robustesse et l'immunité aux parasites et aux bruits.
- La documentation, la formation et le service après vente.
- La durée de garantie.

## II. Présentation générale de l'automate S7-300 :

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

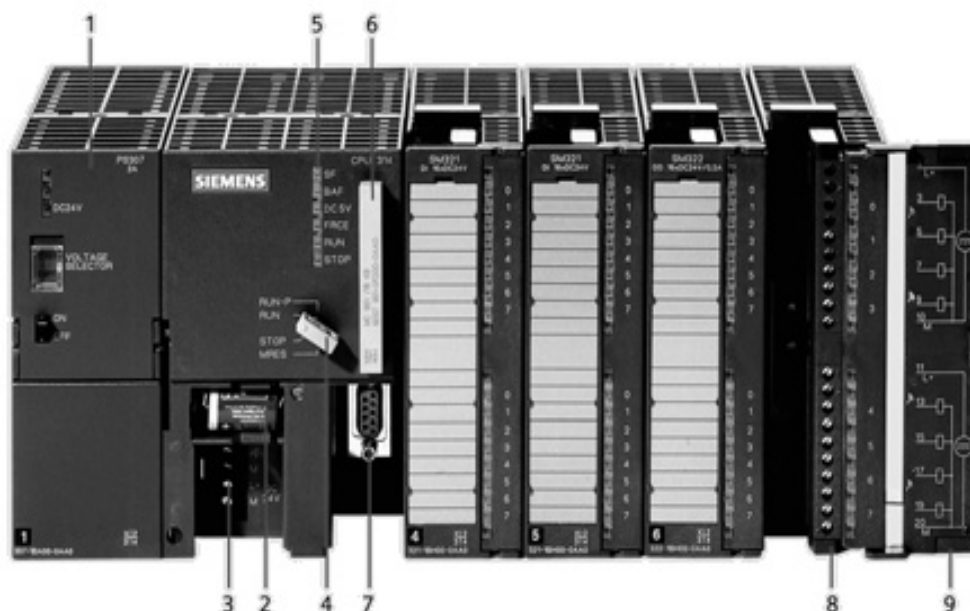


Figure III.2 : Automate modulaire SIEMENS.

- |                                  |                                |
|----------------------------------|--------------------------------|
| 1 : Module d'alimentation.       | 6 : Carte mémoire.             |
| 2 : Pile de sauvegarde.          | 7: Interface multipoint (MPI). |
| 3 : Connexion 24 Vcc.            | 8 : Connecteur frontal.        |
| 4 : Commutateur de mode (à clé). | 9 : Volet en face avant.       |
| 5 : LED de signalisation.        |                                |

Les automates programmables SIEMENS sont des appareils fabriqués en série, conçus indépendamment d'une tâche précise. Tout les éléments logiques, fonctions de mémoire, temporisations, compteurs ...etc, nécessaire à l'automatisation sont prévus par le fabricant et sont intégrés à l'automate. Ils se distinguent principalement par le nombre des :

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

## II.1. 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.

## II.2. Constitution de l'automate S7-300 :

L'automate programmable S7-300 est d'une forme compacte et permet un vaste choix de gamme de module suivants :

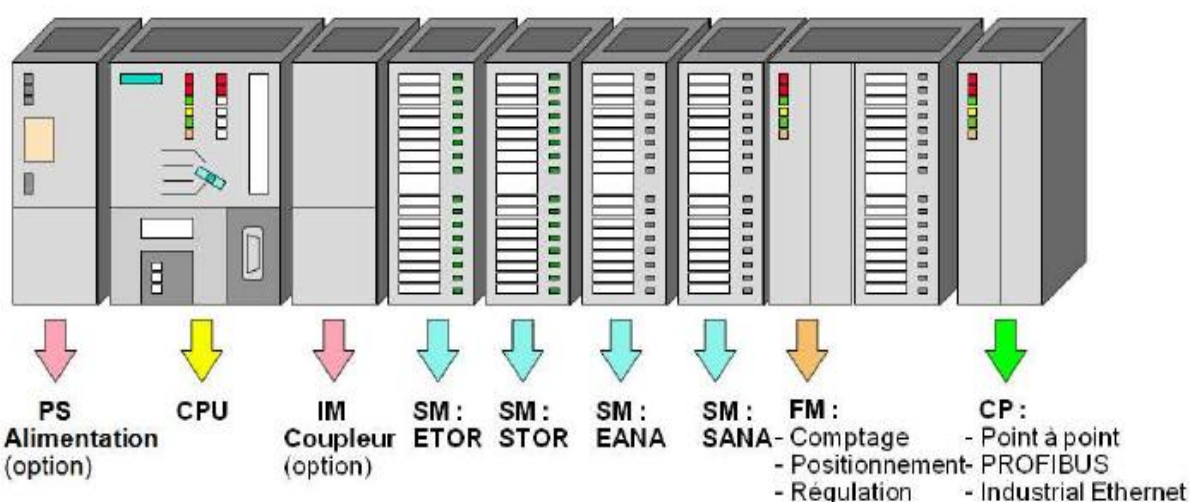


Figure III.3: Constitution de l'automate S7-300

- Module d'alimentation (PS) 2A, 5A, 10A ;
- Unité centrale CPU 314 travaillant avec une mémoire de 48 Ko, sa vitesse d'exécution est de 0,3ms/1Ko instructions ;
- Modules d'extension (IM) pour configuration multi rangée de S7-300 ;
- Modules de signaux (SM) pour entrées et sorties TOR et analogiques ;
- Module de fonction (FM) pour les fonctions spéciales ;
- Processus de communication (CP) pour la connexion au réseau ;
- Châssis d'extension (UR).

### II.2.1. Module d'alimentation (PS) :

Il transforme la tension secteur en tension d'alimentation pour les modules électriques de l'automate programmable. Cette tension s'élève à 24 volts.

Les tensions qui dépassent 24 volts comme pour les capteurs, actionneurs, et voyons lumineux sont fournies par des blocs d'alimentation ou transformateurs supplémentaires.

### II.2.2. Description de la CPU :

La CPU (Central Processing Unit) est le cerveau de l'automate, elle lit les états des entrées, ensuite, elle exécute le programme utilisateur en mémoire et enfin, elle commande les sorties. Elle comporte une unité de commande et de calcul, des mémoires, un programme système et des interfaces vers les modules de signaux.

La CPU est constituée de :

#### - Interfaces MPI :

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

#### - Commutateur de mode de fonctionnement :

Le commutateur de mode de fonctionnement permet de changer le mode de fonctionnement.

Les modes de fonctionnement suivants sont possibles :

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

#### - Signalisation des états :

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

**SF** : signalisation groupée des défauts, défauts de la CPU ou d'un module avec fonction diagnostique.

**BATF** : défaut de pile, pile à plat ou absente.

**DC5v** : signalisation de tension d'alimentation 5v (allumé : les 5v sont présentes, clignote : surcharge courant).

**FCRE** : forçage signalisation qu'au moins une entrée ou une sortie est forcée de manière permanente.

**RUN** : clignotement lors de la mise en route de la CPU, allumage continu en mode RUN.

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

- La carte mémoire :

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

- La pile :

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

- Borne pour l'alimentation et la terre fonctionnelle :

Ce bloc est commun à la majorité des CPU des S7-300, on trouve les différentes bornes d'alimentation tel que le cavalier amovible pour montage sans liaison à la terre.

### **II.2.3. Modules de coupleurs (IM) :**

Les coupleurs peuvent être utilisés pour un couplage sur des courtes distances. Pour un couplage sur de long distance, il est recommandé d'émettre les signaux via le bus Profibus. Les coupleurs IM 360/IM 361 ou IM 365 permettent de réaliser des configurations à plusieurs châssis.

### **II.2.4. Modules de signaux (SM) :**

Il comporte plusieurs signaux tel que : STOR ; ETOR ; SANA ; EANA ; ou E/SANA, et E/STOR, ils ont comme fonction l'adaptation des niveaux de signaux entre le processus et le S7-300.

### **II.2.5. Modules d'entrées et de sorties tout ou rien (TOR) :**

Les API offrent une grande variété d'entrées/sorties TOR adaptées aux milieux auxquels ils sont soumis. Ces entrées/sorties peuvent accepter des informations en courant ou en tension, alternatifs ou continus.

#### **a) Modules d'entrées TOR :**

Un module d'entrée doit permettre à l'unité centrale de l'automate, d'effectuer une lecture de l'état logique des capteurs qui lui sont associés (modules 4, 8, 16 ou 32 entrées).

A chaque entrée correspond une voie qui traite le signal électrique pour élaborer une information binaire.

Le processeur de l'automate vient questionner le module ; le contenu du mot d'entrées du module est alors recopié dans la mémoire de données de l'automate programmable.

Les modules d'entrées « Tout Ou Rien » permettent de raccorder à l'automate différents capteurs logiques et tel que les boutons poussoirs, les fins de course, ...etc.

#### **b) Modules de sorties TOR :**

Un module de sortie permet à l'automate programmable d'agir sur les actionneurs. Il réalise la correspondance : état logique → signal électrique.

Les modules de sortie Tout Ou Rien permettent de raccorder à l'automate différents pré-actionneurs tels que : les électrovannes, les contacteurs ...etc.

### **II.2.6. Modules d'entrée et sortie analogique :**

Les modules d'entrées/sorties analogiques réalisent la conversion des signaux analogiques issus de processus en signaux numériques pour le traitement interne dans le S7-300 et des signaux numériques du S7-300 en signaux analogiques destinés au processus.

### **II.2.7. Module de fonction (FM) :**

Il a pour rôle l'exécution de tâche du traitement des signaux du processus à temps critique et nécessitant une importante capacité mémoire comme le comptage, le positionnement et la régulation.

### **II.2.8. Module de communication (CP) :**

Il permet d'établir des liaisons homme-machine ou machine-machine ; ces liaisons sont effectuées à l'aide des interfaces de communication :

- Point à point.
- Profibus.
- Industriel Ethernet.

### **II.2.9. Châssis d'extension (UR) :**

Il est constitué d'un profilé support en aluminium et bus de fond de panier avec connecteur. Il permet le montage et le raccordement électrique de divers modules tels que : les modules d'entrées/sorties et l'alimentation. Il est possible d'utiliser plusieurs racks en fonction du nombre d'entrées/sorties.

## **II.3. Fonctionnement de base d'un API :**

### **II.3.1. Le module central CPU :**

La tension venant du signaleur est connectée sur la barrette de connexion du module d'entrée. Dans la CPU (Module central), le processus traite le programme se trouvant dans la mémoire et interroge les entrées de l'appareil pour savoir si elles délivrent de la tension ou pas. En fonction de cet état des entrées et du programme se trouvant en mémoire, le processus ordonne au module de sortie de commuter sur le connecteur de la barrette de connexion correspondante. En fonction de l'état de tension sur les connecteurs des modules de sortie, les appareils à positionner et les lampes indicatrices sont connectés ou déconnectés.

### **II.3.2. Réception des informations sur les états du système :**

Le S7-300 reçoit des informations sur l'état du processus via les capteurs de signaux reliés aux entrées.

Le S7-300 met à jour la mémoire image des entrées au début de chaque cycle de programme en transférant le nouvel état des signaux d'entrées des modules vers la mémoire image des entrées ce qui permet à la CPU de savoir l'état du processus.

### II.3.3. Exécution du programme utilisateur :

Après avoir acquis les informations d'entrées et exécuter le système d'exploitation, la CPU passe à l'exécution du programme utilisateur, qui contient la liste d'instruction à exécuter pour faire fonctionner le procédé. Il est composé essentiellement en bloc de données, en bloc de codes et en bloc d'organisation.

### II.3.4. La commande du processus :

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

## III. 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. La présentation en STEP7 présente trois modes de représentation qui peuvent être combinés dans une même application :

- Schéma à contacts (CONT).
- Logigramme (LOG).
- Liste d'instruction (LIST).

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

#### a) Schéma à contacts (CONT) :

Le schéma à contacts est une représentation graphique de la tâche d'automatisation ayant recours aux symboles. C'est un langage des habitués des schémas électriques.

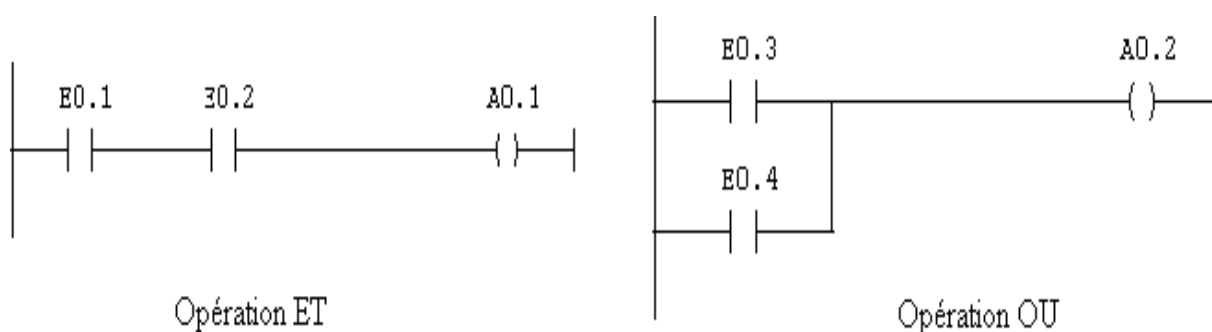


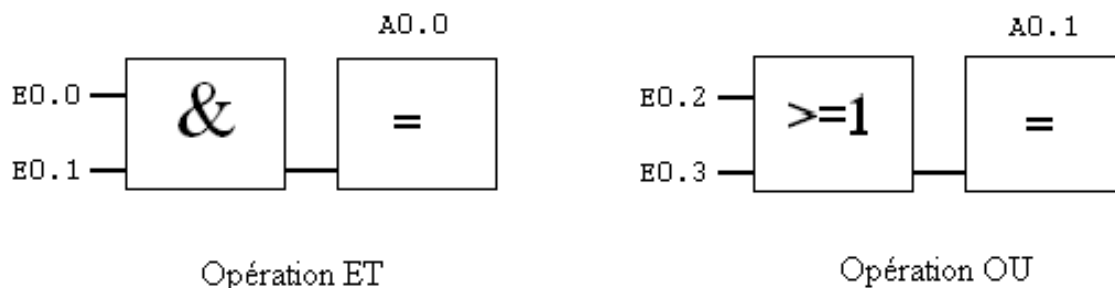
Figure III.4: Représentation des opération ET, OU par CONT .

**b) logigramme (LOG) :**

Le logigramme est une représentation graphique ayant recours aux symboles de la logique.

Les différentes fonctions sont représentées par un symbole avec un indicateur de fonction.

Les entrées sont disposées à gauche du symbole, et les sorties à sa droite.



**Figure III.5: Représentation des opérations ET, OU par LOG.**

**c) Liste d'instructions (LIST) :**

La tâche d'automatisation est écrite dans ce cas à l'aide des différentes instructions. C'est un langage qui s'apparente au langage machine.

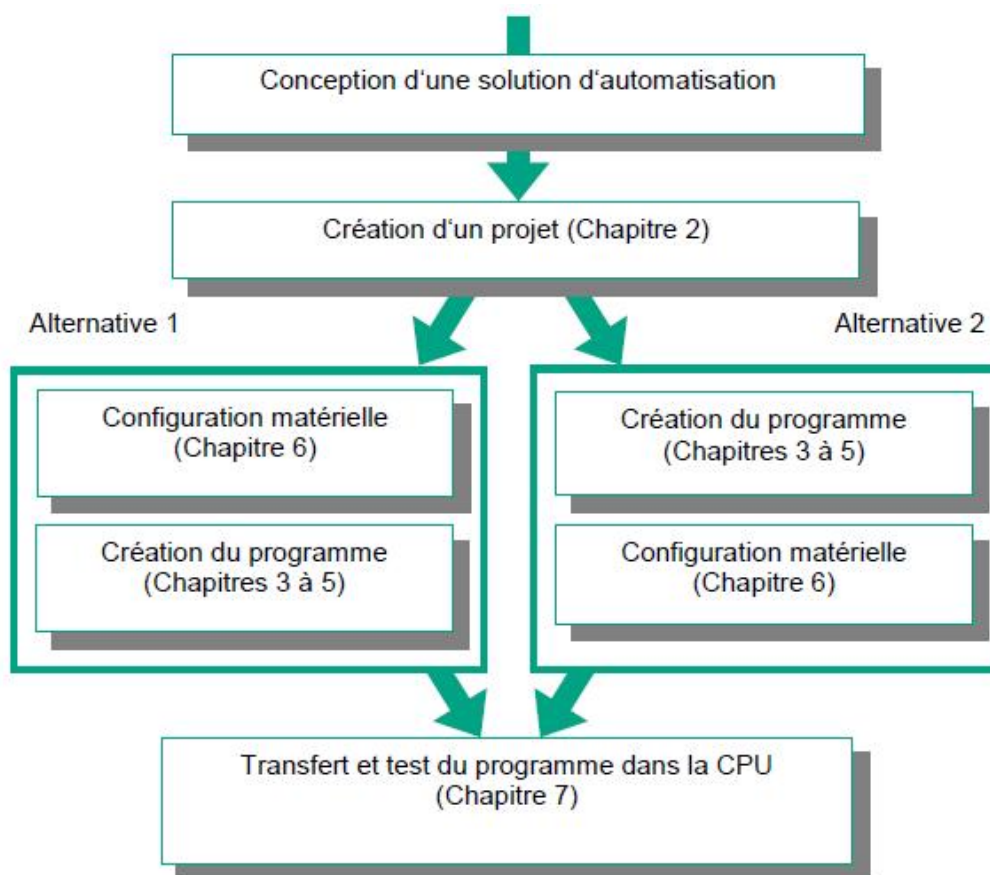
Les programmes en CONT et en LOG sont en principe toujours traduisibles en LIST.

### III.1. 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 illustre les deux solutions possibles lors de la conception d'une solution d'automatisation :



**Figure III.6: Les deux solutions possibles pour la programmation.**

L'application de la configuration matérielle de STEP7 présente l'avantage de la sélection automatique des adresses.

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

1- Double clique sur l'icone SIMATIC Manager  sur le bureau Windows ; ceci lance l'assistant de STEP7.

2- La fenêtre illustrée en figure III.7 apparaît, elle permet la création d'un nouveau projet.



**Figure III.7:Assistant de STEP7 : 'nouveau projet'.**

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

Pour notre projet on a choisi la CPU 314 .



**Figure III.8: fenêtre de choix de la CPU.**

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).

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

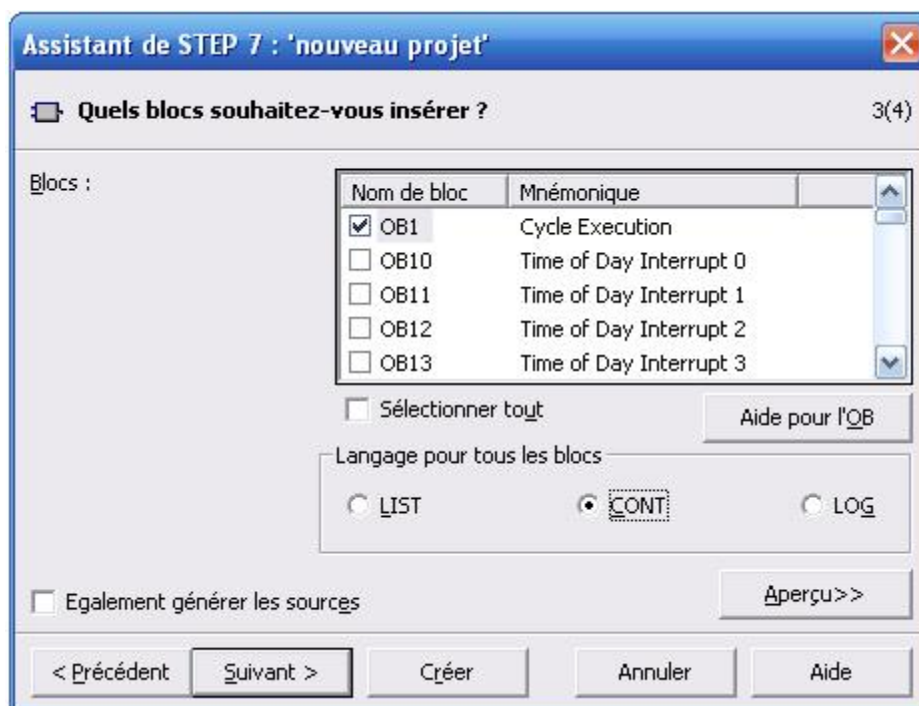


Figure III.9: choix des blocs à utilisés et de langage.

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

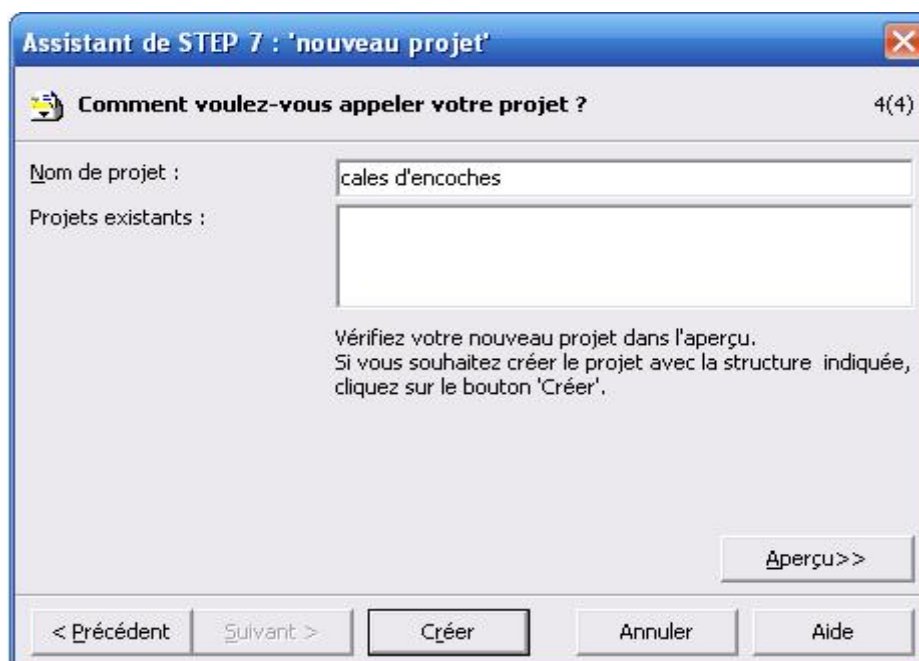
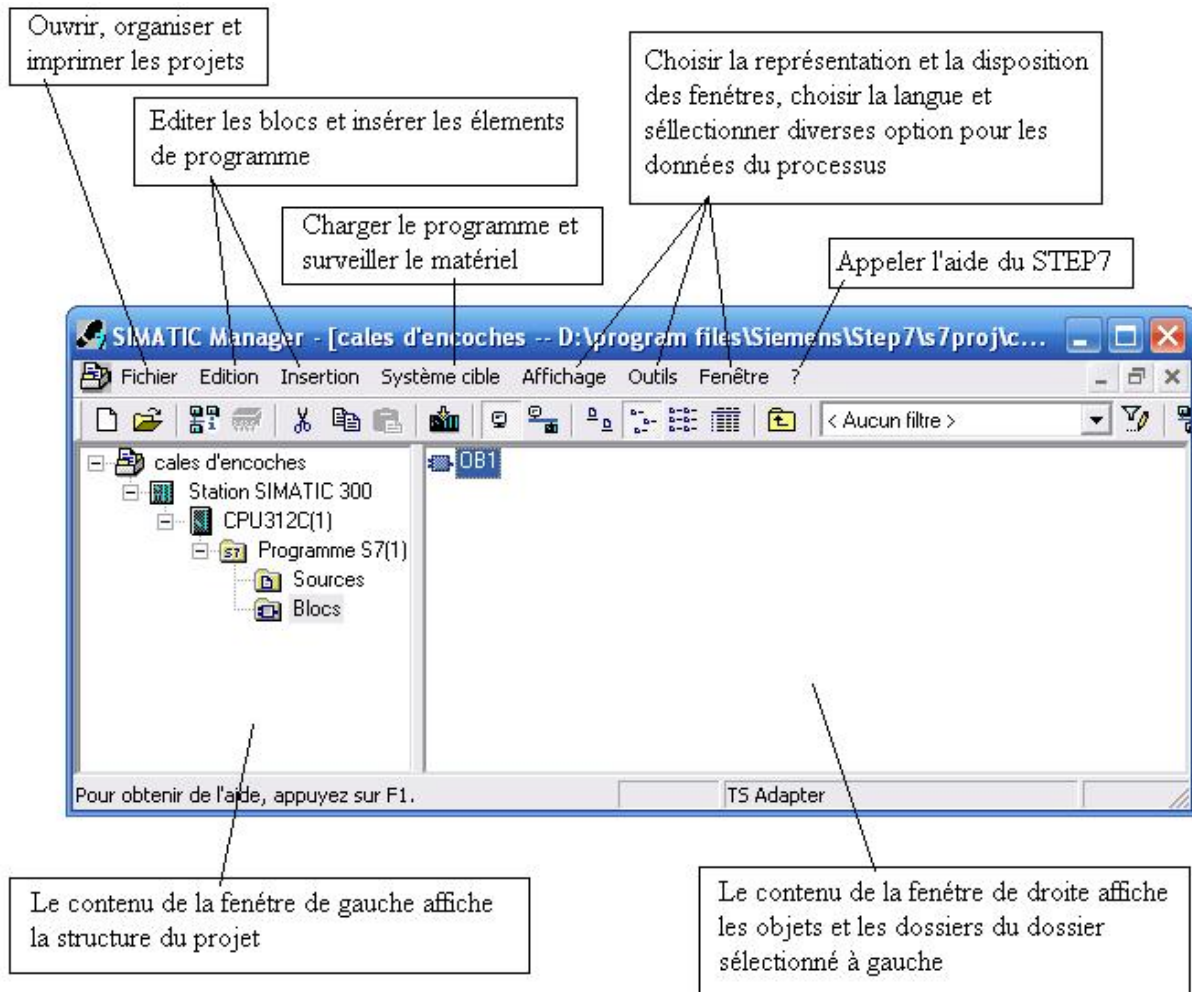


Figure III.10: nomination du produit.

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



**Figure III.11: vue des composants d'un projet S7.**

La gestion des fichiers en STEP7 s'effectue avec 'SIMATIC Manager'. Ici il est possible par exemple de copier les blocs de programme ou d'appeler d'autre traitement avec d'autre 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 clic sur le bouton droit de la souris).

En STEP7, 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 :

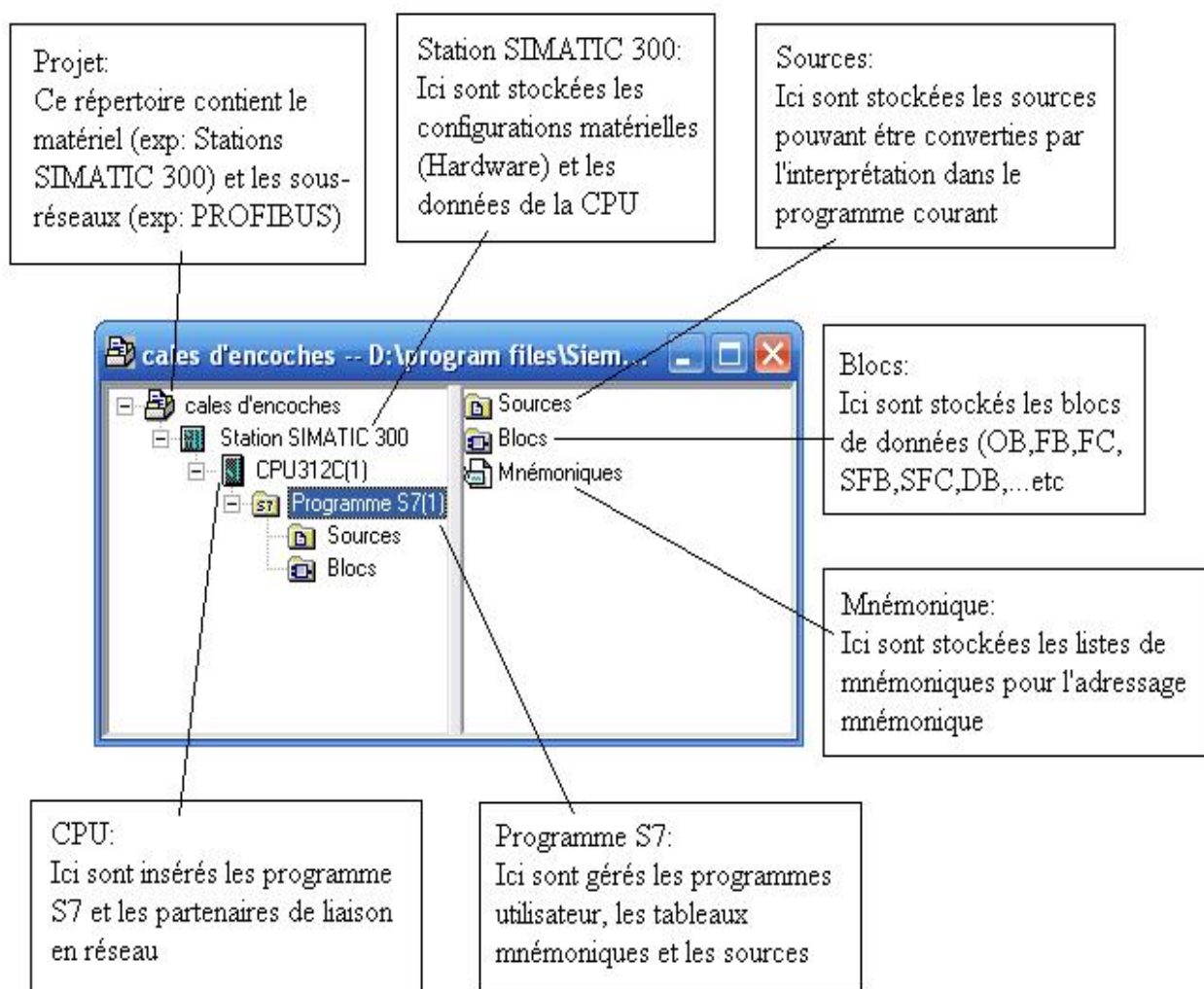


Figure III.12: répertoire de la station SIMATIC et de la CPU.

## III.2. Configuration et paramétrage de l'automate :

On entend par 'configuration', la disposition de profilés support ou châssis, de module, d'appareils de la périphérie décentralisée et de cartouche 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'.

STEP7 effectue automatiquement une adresse à chaque module dans la table de configuration. On peut modifier les adresses des modules d'une station. A 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 exemple : comportement de démarrage et de cycle d'une CPU).

### III.3. 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-y Sélectionnez un composant matériel dans la fenêtre " Catalogue du matériel ".
- 2-y Amenez le composant sélectionné dans le fenêtre de station en utilisant la fonction glisser-lâcher.

La figure suivante illustre les manipulations de base :

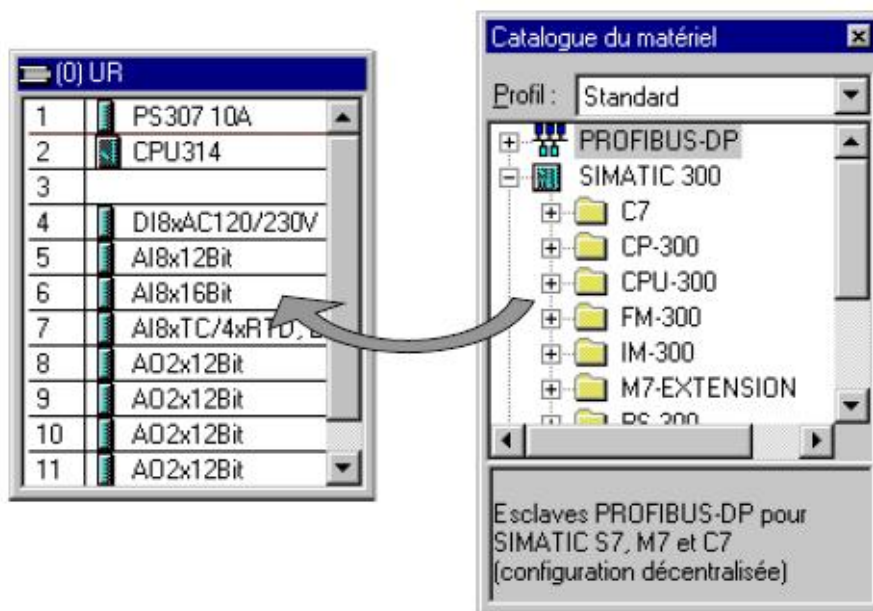


Figure III.13: manipulation de base pour une configuration de matérielle.

La figure suivante illustre la configuration matérielle de notre automate.

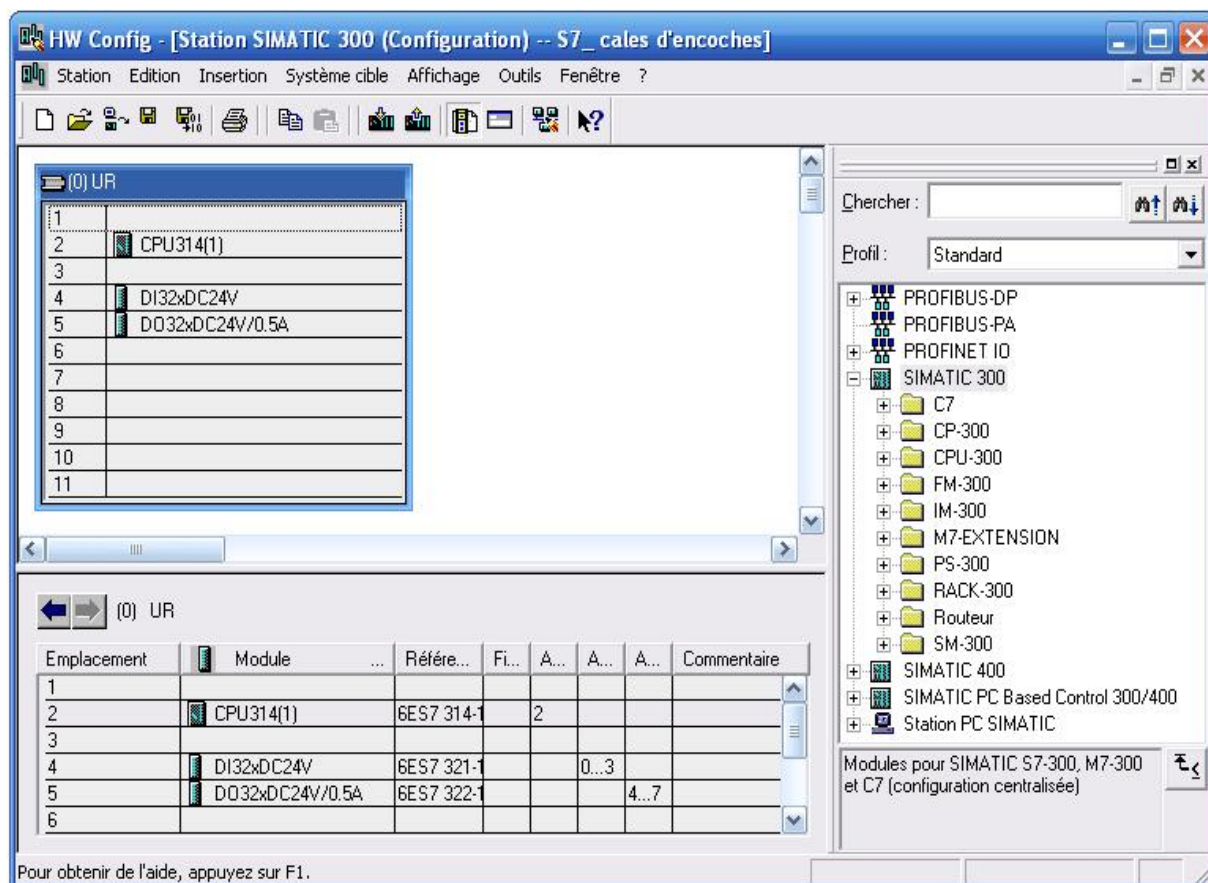


Figure III.14: configuration de matérielle de l'automate.

### III.4. Adressage des modules du S7-300 :

On a deux type d'adressage :

- Adressage lié à l'emplacement : il s'agit du mode d'adressage par default, 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 CPU.

#### III.4.1. L'adressage des modules lié a l'emplacement :

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

### III.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 III.15 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 d'E/S analogique ont la même adresse de début.

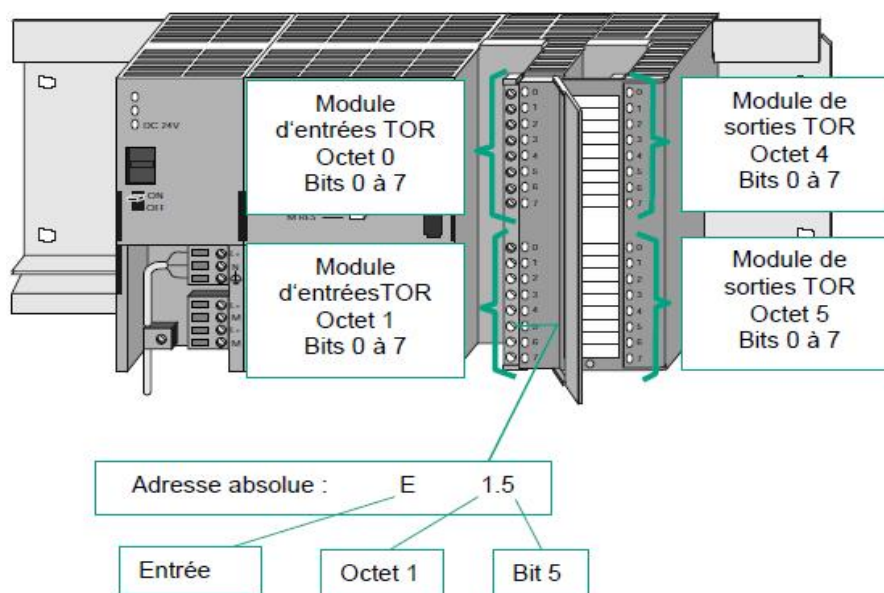


Figure III.15: exemple d'adressage absolu d'un module TOR.

### III.5. Mémento :

Les mémento sont utilisés pour les opérations un ternes à l'automate pour lesquelles l'émission d'un signal n'est pas nécessaire. Les mémentos sont les bistables servant à mémoriser les états logique 1 ou 0.

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.

### **III.6. L'instruction de commande :**

Pour le traitement par une commande à mémoire programmable, la tâche de programmation est décomposée en instruction élémentaires de commande. Une instruction de commande est l'unité autonome d'un programme de commande. Elle représente une directive de travail pour le système de contrôle.

### **III.7. Traitement du programme par l'automate :**

La CPU traite le programme d'une manière cyclique en plusieurs phases :

- Phase(1) : le système d'exploitation démarre la surveillance de temps de cycle.
- Phase(2) : la CPU lit l'état des entrées dans les modules d'entrées et met à jour la mémoire image des entrées.
- Phase(3) : à cette étape, la CPU exécute les instructions de programme utilisateur.
- Phase(4) : la CPU écrit les résultats dans la mémoire image des sorties, puis elle transfère ces derniers vers les modules de sorties.
- Phase(5) : à la fin de cycle, le système d'exploitation exécute les travaux en attente, tel que le chargement et l'effacement des blocs ou la réception et l'émission des données globales.
- Phase(6) : la CPU revient alors au début du cycle et démarre à nouveau la surveillance du temps du cycle.

### **III.8. Principe de conception d'une structure de programme :**

Pour chaque solution d'automatisation, la CPU exécute deux types de programmes différents, le système d'exploitation et le programme utilisateur.

#### **III.8.1. Le rôle de système d'exploitation :**

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

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

### III.8.2. Rôle du programme utilisateur :

Le programme utilisateur est chargé dans la CPU. Il contient toutes les fonctions nécessaires au traitement de la tâche d'automatisation, en plus il doit prendre en charge :

- La détermination des conditions pour le démarrage et le redémarrage de la CPU (exp : initialisation des signaux).
- La réaction aux alarmes.
- Le traitement des données de processus (exp : combiner des signaux binaires , écrire des valeurs analogiques)

#### a) blocs dans le programme utilisateur :

il faut subdiviser le procédé à automatiser en ces différentes tâches .

les parties d'un programme utilisateur structuré correspondant à ces différentes tâches sont les blocs de programmes.

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

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

#### b) Les différents 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 :

##### **b-1) Bloc d'organisation (BO) :**

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.

Nous pouvons 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.

##### **b-2) 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.

### b-3) Fonction (FC) :

Les fonctions font partie des opérations que le concepteur du programme d'utilisateur utilise, elles ne possèdent 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 des 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.

### b-4) Bloc de données (DB) :

Les DB sont utilisées 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 de paramètres effectifs et des données statiques de FB, on appelle l'instance l'appel d'un FB pour un DB.
- Bloc de données globaux (DB) : contrairement aux blocs de code (OB, FC, FB), les DB ne contiennent pas d'instruction 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 lorsque on quitte le DB.

Le rôle des DB dans un programme utilisateur est illustré en figure III.16.

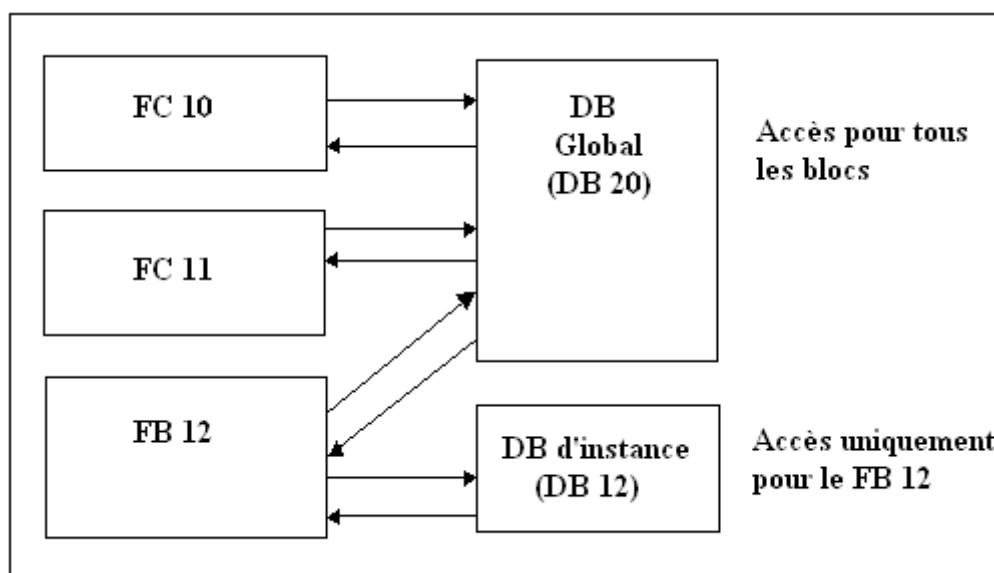


Figure III.16: Illustration de rôle des DB dans un programme utilisateur.

### **III.9. Blocs fonctionnels système et fonctions système :**

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

#### **III.9.1. 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 des mémoires, nous devons donc leur associer des blocs de données d'instance que nous chargeons dans la CPU en tant que partie de programme.

#### **III.9.2. Fonction système (SFC) :**

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

#### **III.9.3. Bloc de données système (SDB) :**

C'est une zone de mémoire dans le programme, configurée par différentes applications STEP7 pour le stockage des données dans le système d'automatisation.

### **III.10. Implantation du Grafcet dans le S7-300 :**

Après avoir bien étudié le procédé à automatiser et réaliser son grafcet, on doit l'implanter dans le S7-300 ou la version du S7 dont nous disposons ne possède pas de langage grafcet qui permet la transition de modélisation à la commande.

Avec la possibilité offerte par le S7-300 en matière de langage il existe

Toujours un moyen d'implanter le grafcet dans l'automate en vue de son exécution, pour cela nous devons suivre certaines étapes :

- 1) Analyser et valider le grafcet de la machine (blocage, conflit,...).
- 2) Détermination des conditions d'activation et désactivation des étapes du grafcet.
- 3) Définition des exigences en matière de sécurité (les alarmes, AUD et AUD ...).
- 4) Affectation des mementos aux variables intermédiaires, et des adresses effectives et formelles aux entrées et sorties.
- 5) Écriture des équations des étapes.
- 6) Traduction de ces équations en programme séquentiel, en utilisant l'un des langages offerts par le STEP7 (List, Cont et Log).

- 7) Raccorder toutes les entrées, sorties et la PG à l'automate.
- 8) Après avoir choisi la configuration matérielle et simuler le programme, on procède au chargement du projet dans la CPU physique.
- 9) On lance la machine en la mettant en mode RUN.
- 10) Après validation du projet, on peut commencer la production.

#### **IV. Conclusion :**

Ce chapitre nous a permis de présenter les différents modules constituant l'ensemble de l'automate S7-300 et son langage de programmation.

La constitution modulaire, la facilité de réalisation d'architecture décentralisée et la facilité d'emploi font du S7-300 la solution économique et conviviale pour les tâches les plus diverses dans les petites et moyennes applications.

Le logiciel de programmation STEP 7 constitue le lien entre l'utilisateur et l'automate programmable S7-300 car ce dernier ne peut gérer ses fonctionnalités sans un programme approprié. Mais avant l'élaboration du programme il faut d'abord modéliser le procédé. Cela a fait l'objet du chapitre précédent.

# **Chapitre IV :**

## **Simulation avec le logiciel S7-PLCSIM et la configuration du panel tactile à l'aide du WinCC Flexible.**

## I. Simulation et validation avec PLCSIM :

### I.1. Introduction :

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

Pour cela nous avons utilisé le logiciel S7-PLCSIM qui est un logiciel optionnel de STEP7.

L'application de simulation de module S7-PLCSIM nous permet d'exécuter et de tester notre programme dans un automate programmable. La simulation étant complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux). L'objectif de ce logiciel est le test des programme STEP7 pour les automates S7-300 et S7-400 qu'on ne peut pas tester immédiatement sur le matériels ou blessures corporelles en cas d'erreurs de programmation, mais la simulation permet de corriger ces erreurs pendant le test de la simulation.

### I.2. Présentation du S7-PLCSIM :

L'utilisation du simulateur de modules physiques S7-PLCSIM nous permet d'exécuté 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.

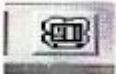
Le S7-PLCSIM dispose d'une interface simple nous permet de visualiser et de forcer les différents paramètres utilisés par le programme (comme activer ou désactiver des entrées). Tout en exécutant le programme dans l'API de simulation, nous avons également la possibilité de mettre en œuvre les diverses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

### I.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 IV.1).

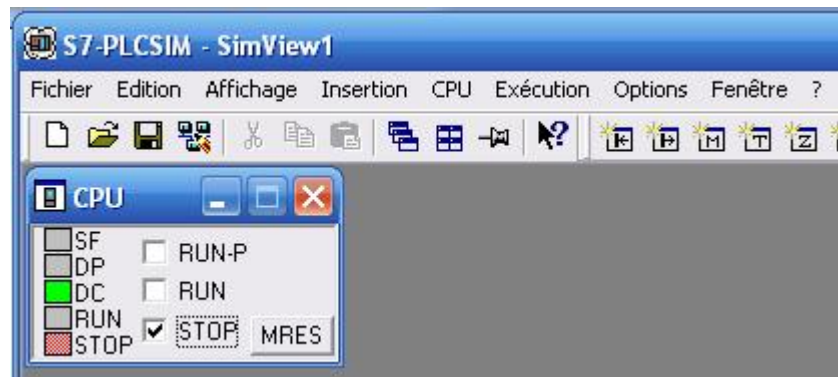






Figure IV.1 : Fenêtre du S7-PLCSIM.

- ✚ Dans le gestionnaire de projets SIMATIC, chercher le projet-exemple projet cales d'encoche.
- ✚ Dans le projet exemple « projet cales d'encoche », 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 créer de nouvelles fenêtre pour visualiser les informations provenant de l'API de simulation :

- ✚ Cliquez sur  ou choisissez la commande Insertion - Entrée pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des entrées (zone E ) cette fenêtre s'ouvre avec l'adresse de mémoire par défaut EB0. Mais peut modifier
- ✚ Cliquez sur  ou la commande Insertion-sortie pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des sorties (zone A) . cette fenêtre s'ouvre avec l'adresse de mémoire par défaut Ab0. Mais on peut modifier l'adresse (AB1, AB2... ).
- ✚ Cliquez sur  ou choisissez la commande Insertion-Temporisation pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer les temporisations utilisées par le programme. Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut T0.

Choisir le menu CPU dans la fenêtre du S7-PLCSIM et vérifier que la commande mettre sous tension est activé.

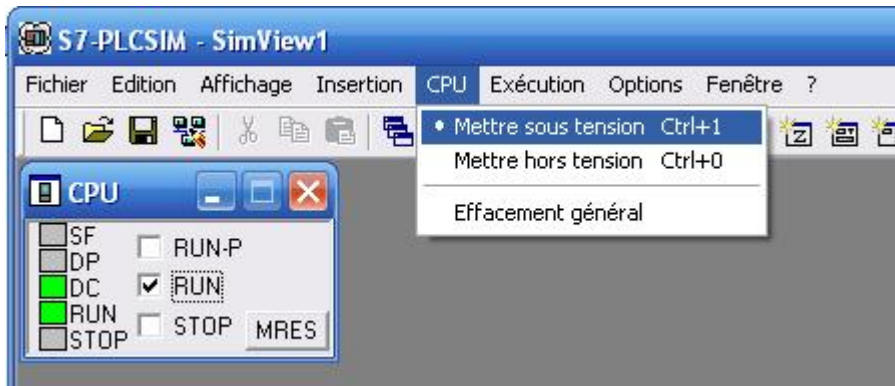


Figure IV.2: Mise sous tension de la CPU.

- ✚ Choisir la commande Exécution - Mode d'exécution et vérifier que la commande cycle continue est activée.

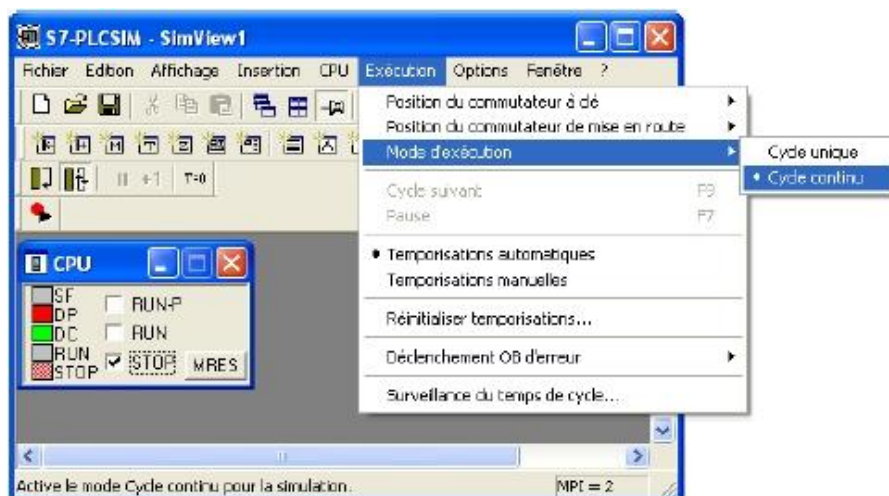


Figure IV.3: choix du cycle continu.

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



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

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

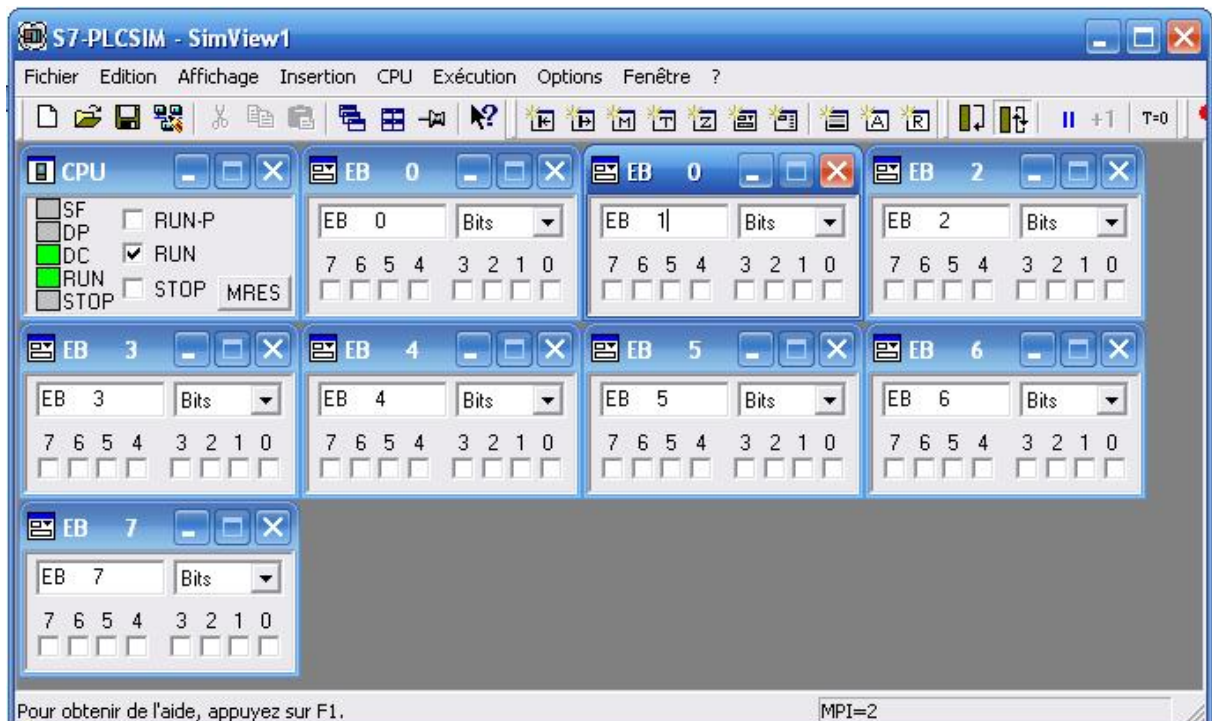


Figure IV.5: Simulateur S7-PLCSIM.

#### I.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 STEP7 nous permet de visualiser l'état du programme on sélectionnant la commande Test – Visualiser.

#### I.5. Simulation d'un programme :

La simulation d'un programme ce faite en deux étapes :

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

## II. Conclusion :

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

Les actions de chaque équipement de la chaîne sont programmées dans un FC dans le but de repérer et rendre facile les modifications à apporter, alors que les FC programmés sont insérés dans le bloc d'organisation OB1.

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

Le logiciel de simulation S7-PLCSIM nous a permis de tester la solution programmée que nous avons développée pour la commande de la chaîne, de corriger toutes les erreurs que nous avons commises et apporter les modifications nécessaires, comme il nous a permis de valider et de visualiser le comportement des sorties.

## III. WinCC Flexible :

### III.1. Introduction au WinCC :

SIMATIC WinCC Flexible est un logiciel innovant sous Windows, qui convient pour toutes les applications au pied de la machine.

WinCC Flexible est l'interface homme-machine (IHM). Il permet de disposer d'un logiciel d'ingénierie pour tous les pupitres opérateur SIMATIC IHM, du plus petit micro panel jusqu'au multi panel. Il comprend aussi un logiciel de supervision Runtime pour solutions basées sur PC.

Pour ce faire, on sollicite deux composants essentiels du logiciel SIMATIC WinCC Flexible :

#### III.1.1 WinCC Flexible Advanced :

Le logiciel de configuration WinCC Flexible Advanced permet de créer les configurations sous Windows, à partir de l'ordinateur de configuration (PC ou console de programmation).

### III.1.2. WinCC Flexible Runtime :

WinCC Flexible runtime est un logiciel performant et facile à utiliser pour la visualisation du processus des projets créés avec le logiciel de configuration winCC Flexible advanced sous windows.

Le logiciel runtime se distingue par son interface utilisateur entièrement graphique, basée sur la technique des fenêtres. Il permet grâce à des temps de réaction rapides une conduite de processus sûre.

Il est également exécuté sur l'ordinateur de configuration pour tester et simuler le fichier projet compilé.

On dispose avec le WinCC Flexible runtime de plusieurs options qu'on peut acquérir en complément selon nos besoins.

#### Ø Fonctions :

WinCC Flexible Runtime offre les fonctions suivantes :

- Représentation conviviale du processus, grâce à une interface utilisateur conforme
- Windows.
- Large choix de champs d'entrée/sortie standard, paragraphes, affichage de courbe.
- Graphique vectoriel et boutons.
- Système d'alarme intégré
- Positionnement dynamique des objets.
- Archivage des alarmes et des valeurs de processus.
- Recettes
- Couplages standard à SIMATIC S7, SIMATIC S5 et SIMATIC 505 et aux automates d'autres fabricants.
- Protection de la saisie via les groupes d'utilisateurs, mots de passe et temps de validité
- Déconnexion.

✓ SIMATIC HMI (Human Machine Interface) propose une vaste gamme de pupitres opérateur pour toutes les applications de conduite et de supervision.

Parmi ces pupitres, celui qu'on a proposé est le pupitre écran tactile TP177A.

### IV. Choix du panneau tactile TP177A :

Le choix s'est porté sur le panel tactile TP177A, pour ces avantages qu'il présente :

- La possibilité de création de plusieurs menus de commande, et donc une meilleure commande du système et d'une manière plus précise et juste.
- L'ensemble graphique auquel il ouvre droit, de façon à mieux présenter l'état de la machine, et de manière plus compréhensive vis-à-vis de l'opérateur.
- La possibilité d'être monté verticalement pour la réception d'autre application.
- Les temps de configuration et de mise en service, qui se singularisent par une courte durée de configuration dans le WinCC flexible.

### Ø Panneau tactile TP 177A :

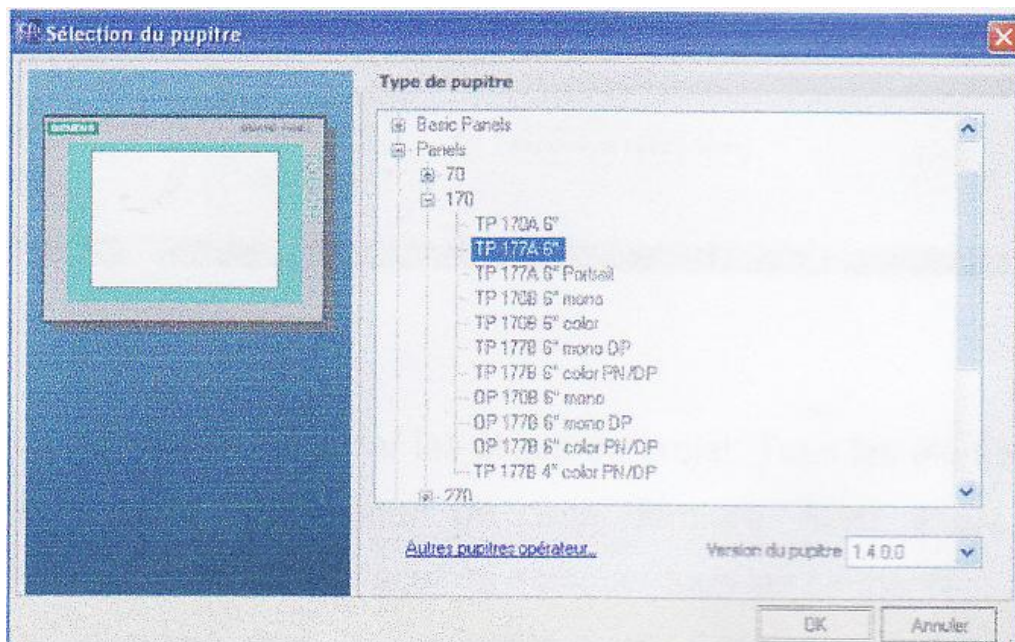
Les afficheurs TP177A micro sont conçus pour fonctionner avec le S7-300. Ils fournissent des fonctions de commande et de surveillance pour des machines et des installations.

## V. Etapes de configuration :

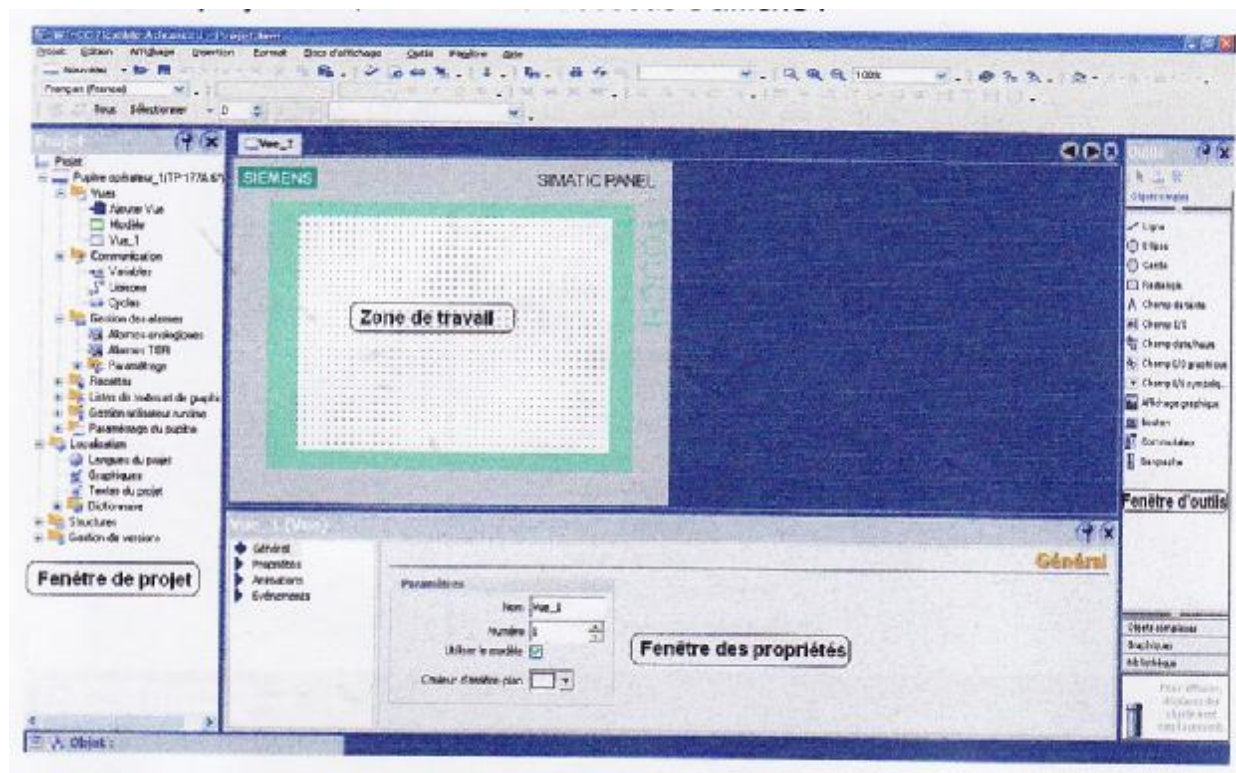
### V.1. créer un projet :

Le projet est à la base de la configuration de l'interface graphique du pupitre. il sert à créer tous les objets indispensables à la commande et au contrôle de notre machine.

Pour créer un projet, on ouvre le SIMATIC WinCC Flexible, et on appuie sur créer projet vide puis choisir des pupitres 170 le pupitre 177A, comme suit :



Une fois le projet créé, la fenêtre ci-dessous s'affiche :

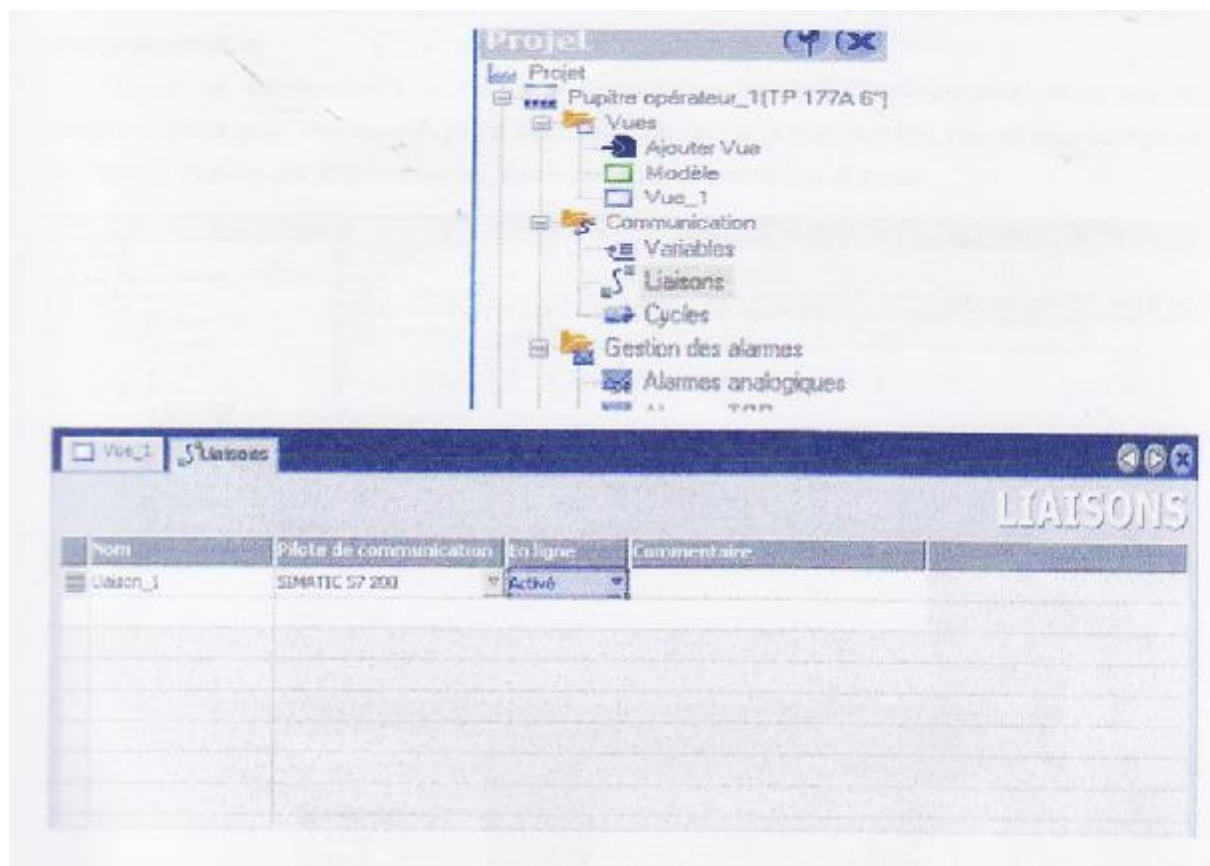


On définira :

- Zone de travail : sert à éditer les objets du projet. tous les éléments du WinCC Flexible sont disposés autour de cette dernière. Ainsi, il est possible de disposer et configurer, déplacer ou masquer tous les éléments.
- Fenêtre de projet : tous les éléments et les éditeurs disponibles d'un projet sont affichés dans l'arborescence et peuvent y être ouverts. De plus, il est possible d'accéder aux propriétés du projet et au paramétrage du pupitre utilisateur.
- Fenêtre d'outils : propose une sélection d'objets qui peuvent être insérés dans une vue, par exemple des objets graphiques et des éléments de commande. elle met aussi à notre disposition des bibliothèques d'objets et de collections de bloc d'affichage déjà prêt à l'emploi.
- Fenêtre des propriétés : édite les propriétés des objets, p.ex. : la couleur des objets graphiques.

## V.2. Configuration de la liaison :

Une fois notre projet créé, on doit configurer la liaison entre le pupitre et l'automate. Pour cela, on appuie sur Liaison dans la fenêtre Projet, et on choisit S7-300.



## V.3. Création d'une vue (Menu) :

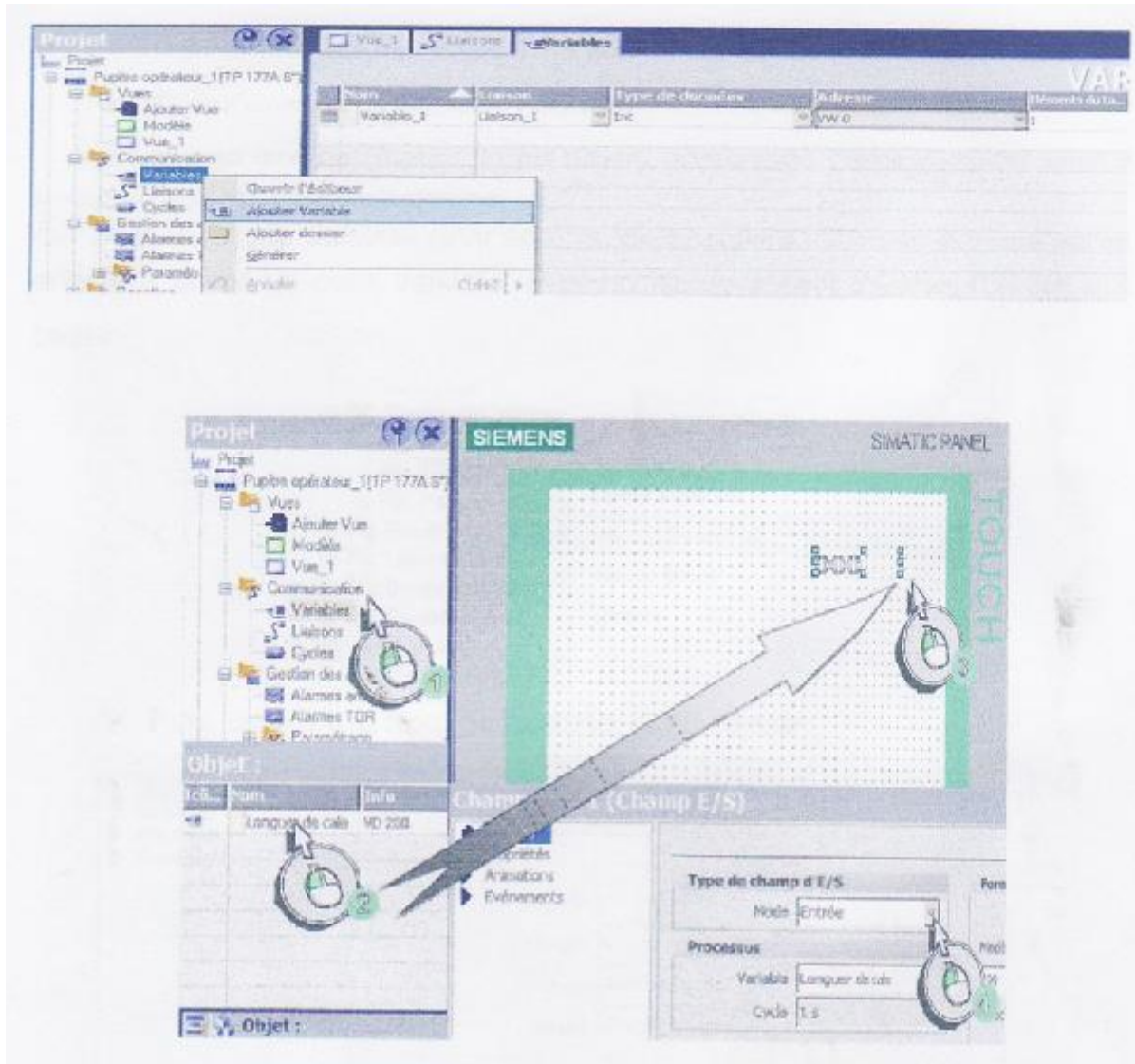
Les vues sont les éléments principaux du projet. Elles permettent de commander et de contrôler la machine. Elles peuvent comporter des champs de Sorties (Variables de type sortie), ou des champs d'Entrées (Variable de type entrée). Elles peuvent contenir d'autres informations telles que : zones de textes, des paragraphes,...

On peut créer d'autant plus de vues, selon le nombre de menus nécessaire à la commande du dispositif, pour cela un double clique sur Ajouter Vue, du menu Vues de la fenêtre de Projet.

## V.4. Création de variables :

Les variables transmettent les données entre l'installation et le pupitre (et donc champ d'entrée) et inversement (champ de sortie), ou encore des variables servant à communiquer le mode de fonctionnement (auto ou manuel), et ceci via l'automate programmable.

Pour se faire, dans la fenêtre projet, on choisit communication, et on avec le bouton droit sur Variables puis Ajouter Variable. Une fenêtre s'ouvre pour configurer le nom, types de données et adresse de variable à créer.



**V.5. Configuration d'une alarme :**

Les alarmes montrent les événements ou les états de fonctionnement qui se produisent sur la machine et qui nécessitent l'avertissement de l'opérateur. Elles servent au diagnostic des erreurs.

Il existe deux types d'alarmes : Alarmes TOR et Alarmes Analogiques.

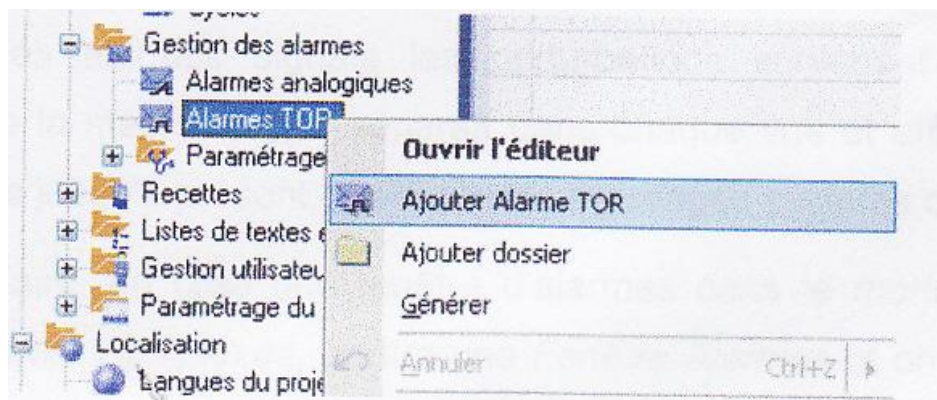
- ✚ Dans notre cas, on utilisera les Alarmes TOR pour indiquer : la fin du ruban, Nombre de pièces à découper atteint et bouton d'arrêt d'urgence actionné (système Tout Ou Rien).

Pour configurer une alarme, on suit les étapes suivantes :

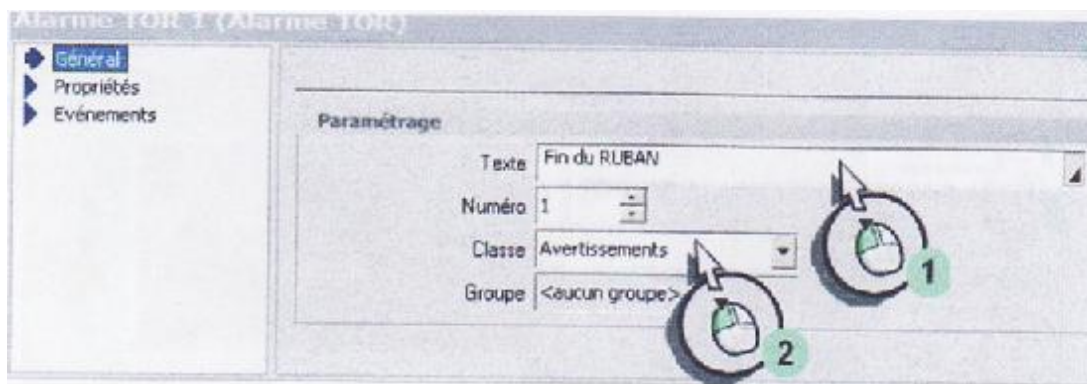
Ø Créer une variable pour l'enregistrement des états :

On entend dire par états : fin du ruban, nombre de pièces à découper atteint et bouton d'arrêt d'urgence actionné.

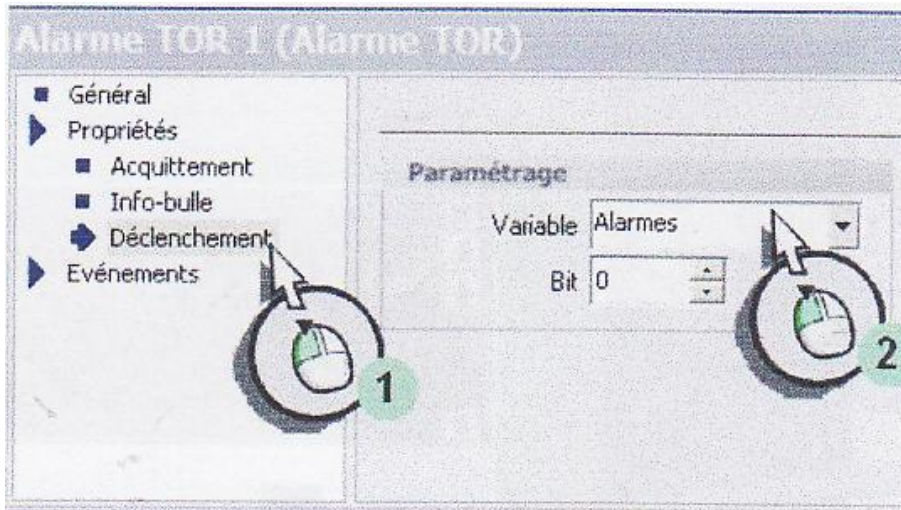
On crée alors une variable (VW 900) qu'on appellera : Alarmes, et chaque état est affecté à un bit de cette variable. Puis on ajoute autant d'Alarmes TOR dont on a besoin.



Ø Puis, on entre le nom de l'alarme, et sa classe :



Ø Et enfin, on sélectionne la variable et le bit ou l'état est enregistré :




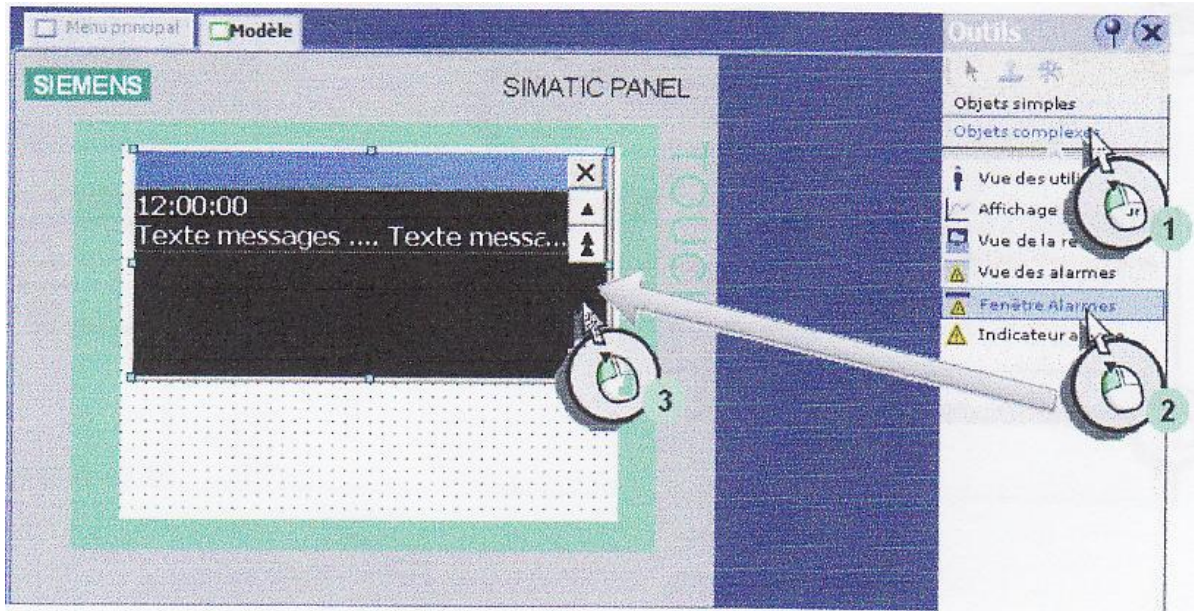
On procède de la même manière pour configurer l'alarme de « Nombre de pièces à découper atteint » et celle du « Bouton d'arrêt d'urgence actionné ».

Texte	INDIC	Classe	Variable de déclenchement	Numéro de bit
Pin du RUBAN	1	Avertissements	Alarmes-Ruban	1
Nombre de pièces à découper atteint	2	Avertissements	Alarmes-PIECE_FIN	0
Bouton d'arrêt d'urgence actionné	3	Avertissements	Alarmes-AUS	0

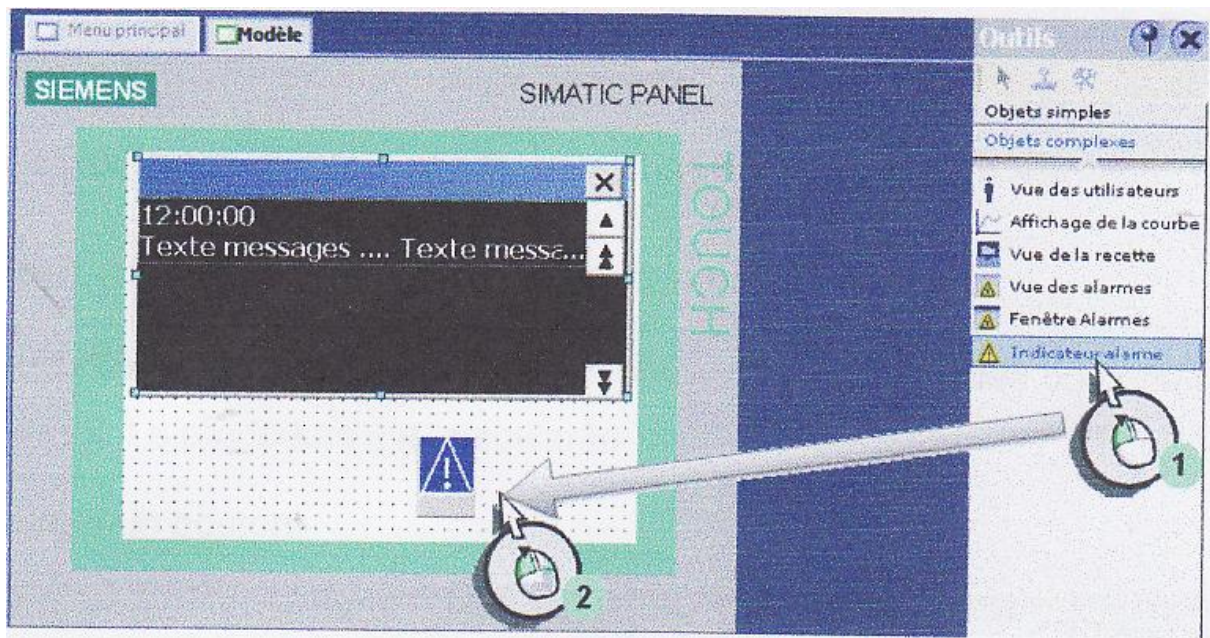
### V.6. Configurer l'affichage d'alarmes :

La fenêtre des alarmes signale les perturbations survenant au cours du fonctionnement de la machine. Elle apparaît dans chaque vue et affiche toutes les alarmes survenues jusqu'à présent, ainsi que les messages système correspondants.

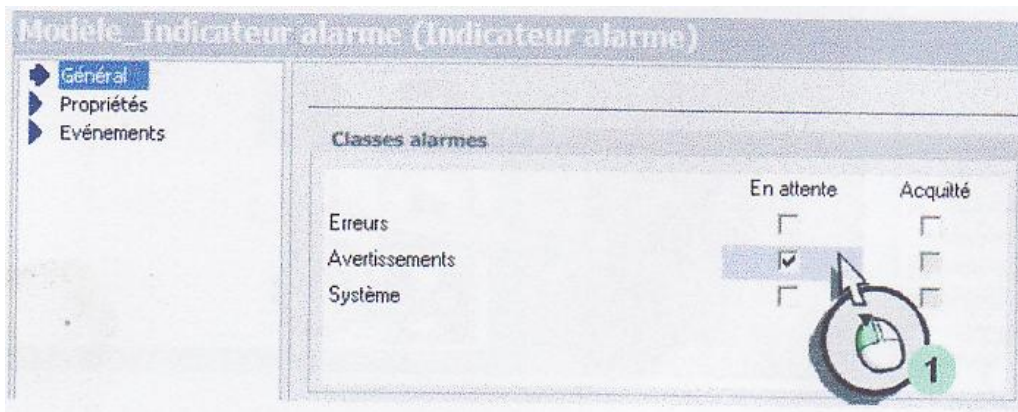
- Ø Pour se faire, on crée une fenêtre d'alarmes dans le modèle à partir du menu outil  Objets Complexes, on tire une fenêtre Alarmes et on ajuste la taille qui convient comme le montre la figure suivante :



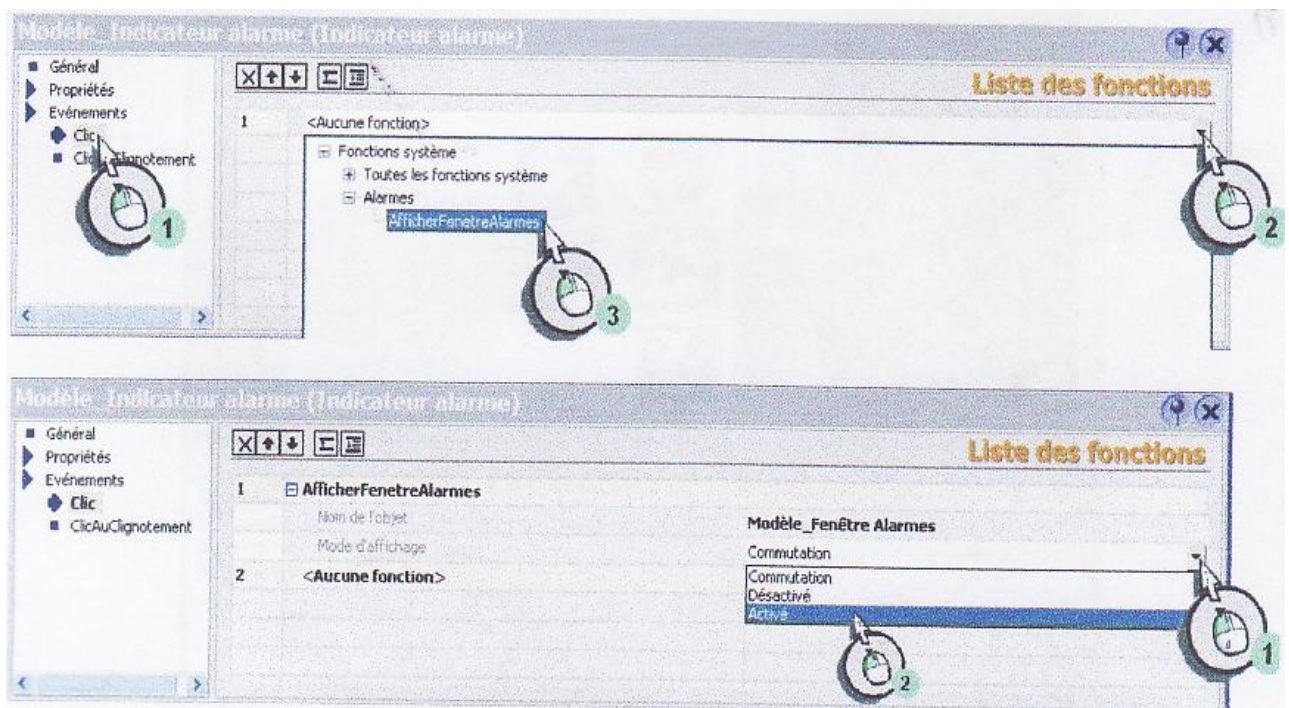
∅ Puis on insère un indicateur d’alarme comme le montre la figure :



Ø On configure ensuite l'indicateur d'alarme.



Ø Puis on associe l'indicateur d'alarme à la fenêtre créée pour qu'elle soit affichée une fois qu'on ait appuyé sur l'indicateur.



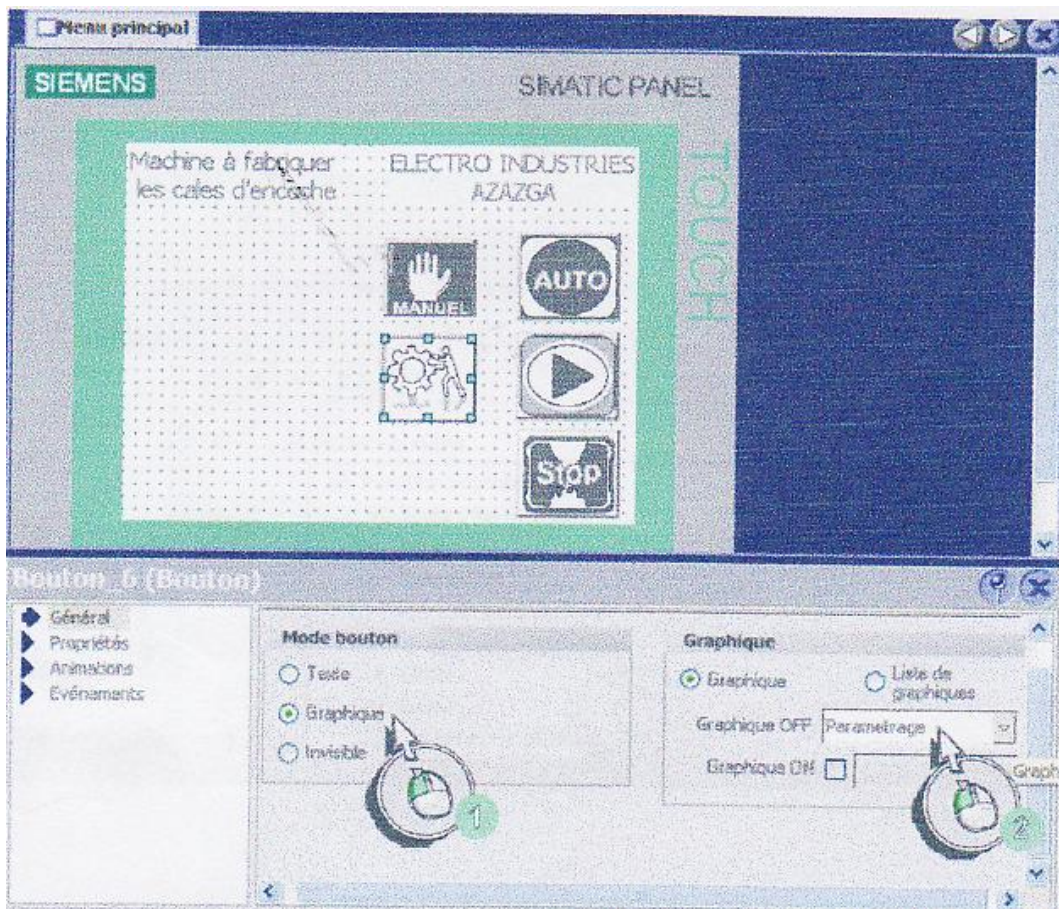
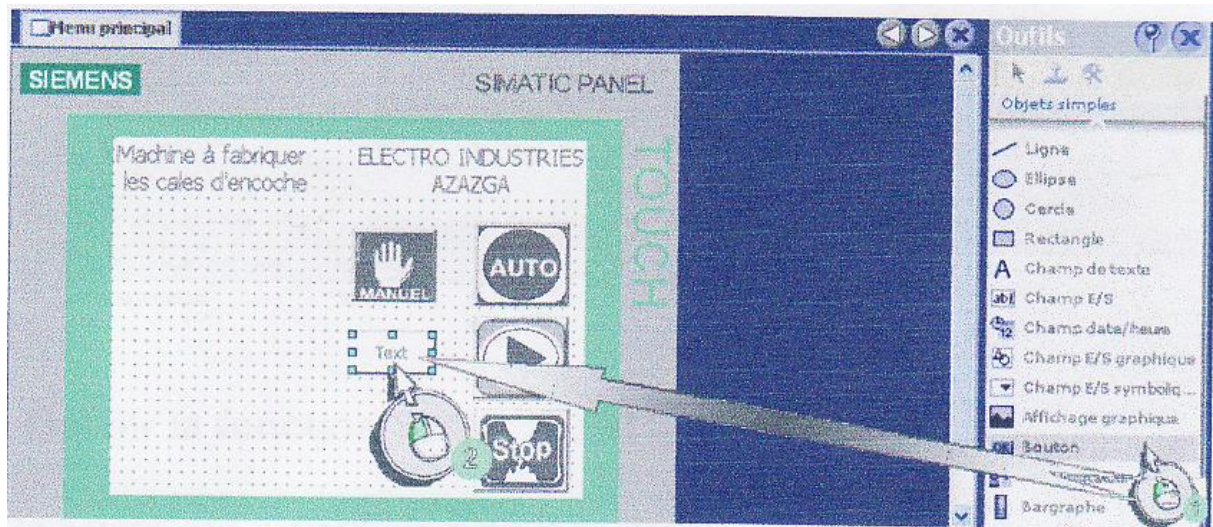
Ainsi, à chaque fois que le ruban est fini, nombre de pièces à découper atteint ou encore bouton d'arrêt d'urgence actionné, l'indicateur d'alarme s'affiche en premier plan ainsi que le nombre d'alarmes déclenchées.

Pour visualiser la nature des alarmes, on appuie sur l'indicateur d'alarmes.

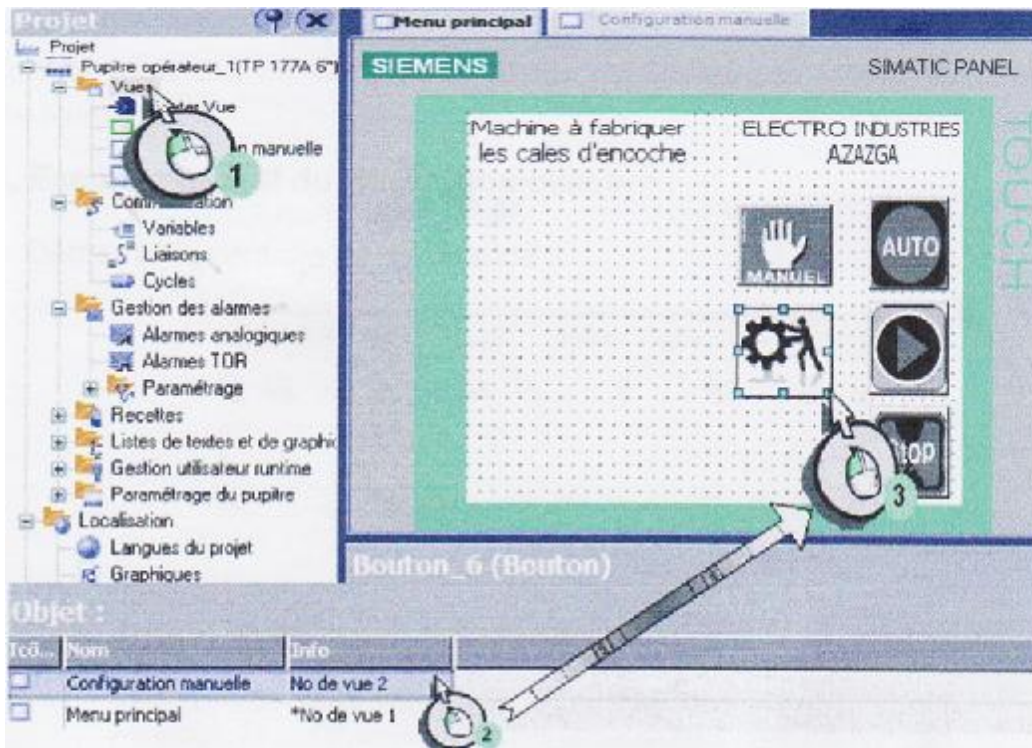
V.7. Ajout de changements de vue :

Afin de naviguer entre les différents menus, on doit créer des boutons qui nous le permettront. La procédure est la suivre :

- Ø Créer un bouton.

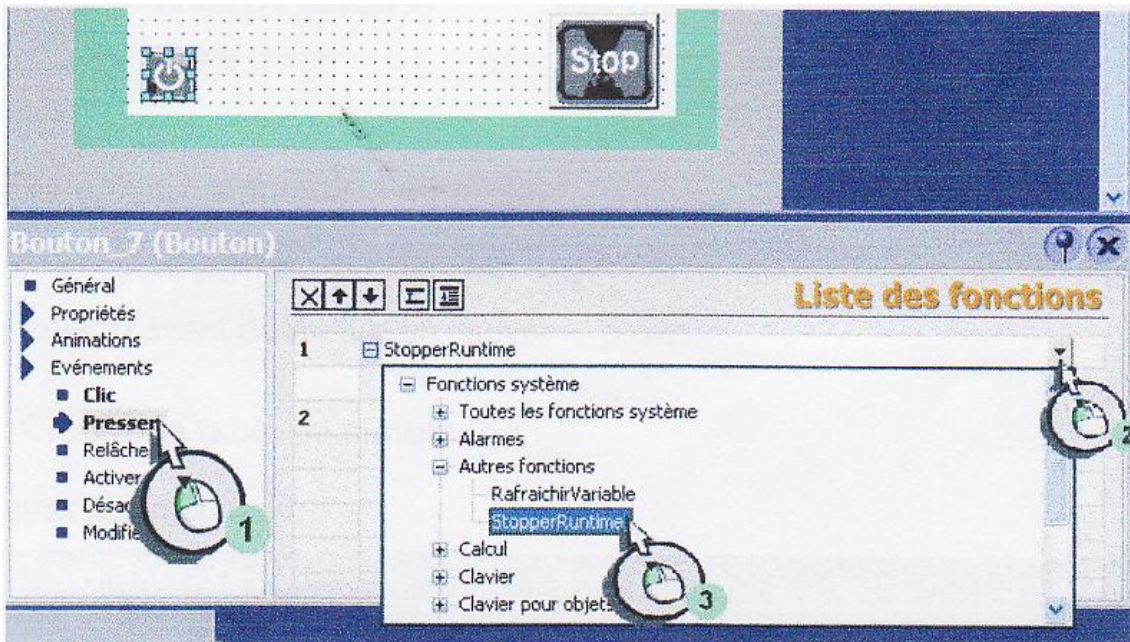


Ø Associer le bouton à une vue :



**V.8. Définir la fonction STOP\_RUNTIME :**

Ce bouton sert à arrêter la simulation. Pour le configurer, on crée un bouton et on procède comme le montre la figure suivante :

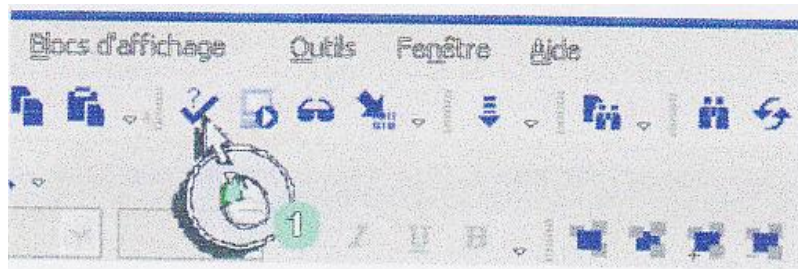


**V.9. Teste et simulation du projet :**

Avant de terminer la configuration, on teste et simule le projet à l'aide de la fonction de contrôle de cohérence et du simulateur de WinCC flexible (WinCC flexible runtime).

**V.9.1.Exemple de test du projet (Compilation) :**

Ø Démarrer le contrôle de cohérence :



Fenêtres des erreurs et avertissements		
Heure	Catégorie	Description
20:35:59.71	Compilateur	Conversion des images en cours ...
20:35:59.71	Compilateur	Polices en cours de conversion ...
20:35:59.71	Compilateur	Vérification des résultats en cours ...
20:35:59.71	Compilateur	Ecriture des fichiers de sortie en co...
20:35:59.71	Compilateur	Nombre de Power Tags utilisés : 4
20:35:59.71	Compilateur	Terminé avec 0 erreur(s), 0 avertisse...
20:35:59.71	Compilateur	Horodatage : 22/05/2010 20:29:22 - ...
20:35:59.71	Compilateur	Compilation terminée !

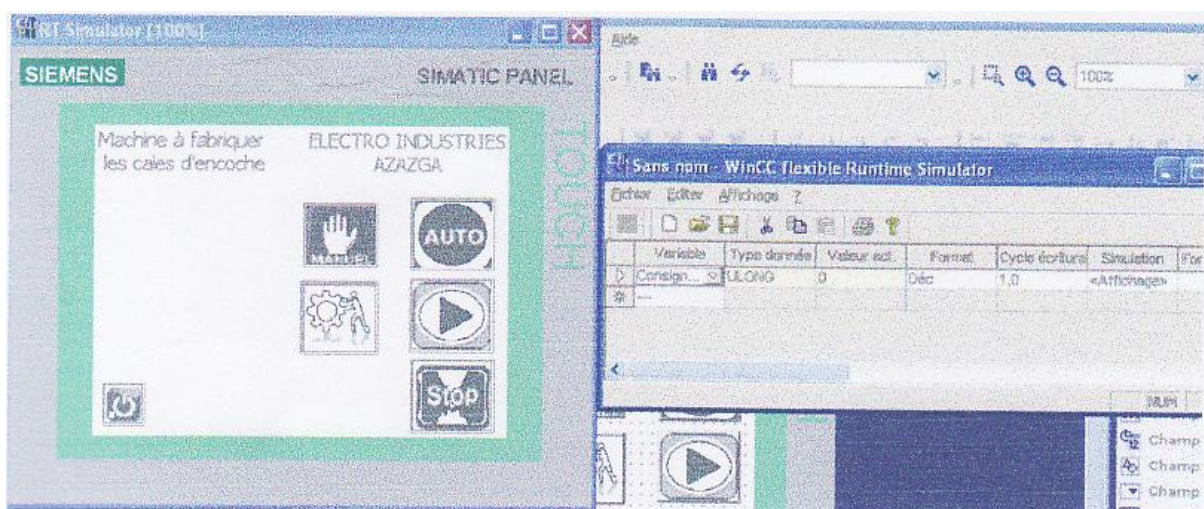
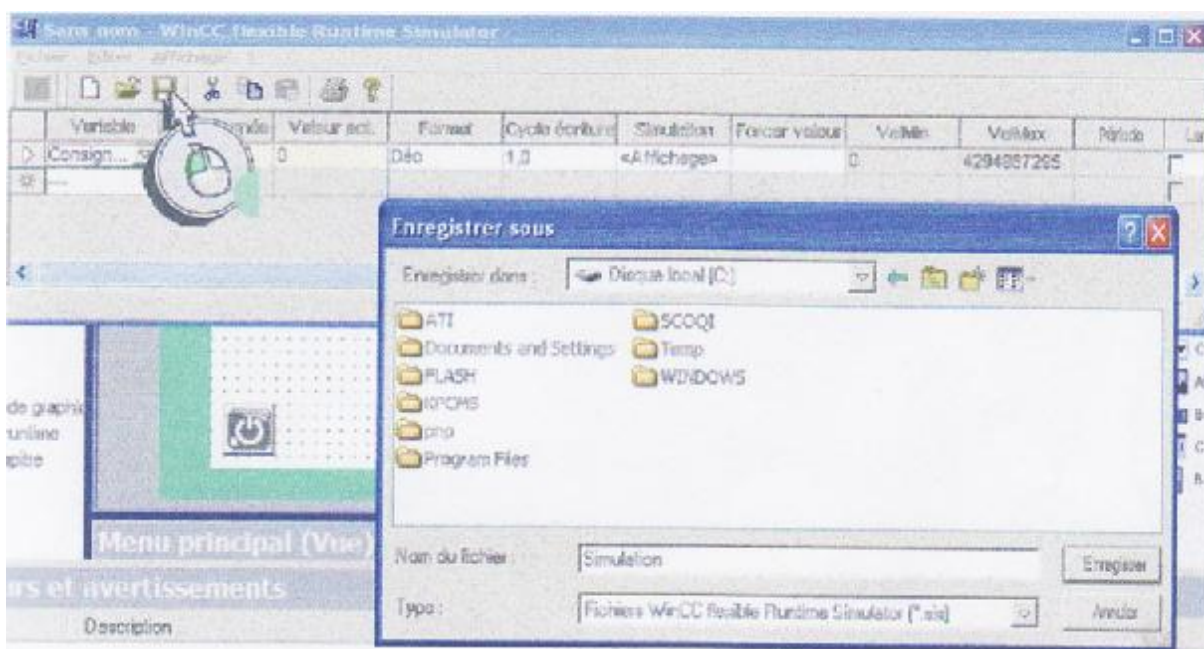
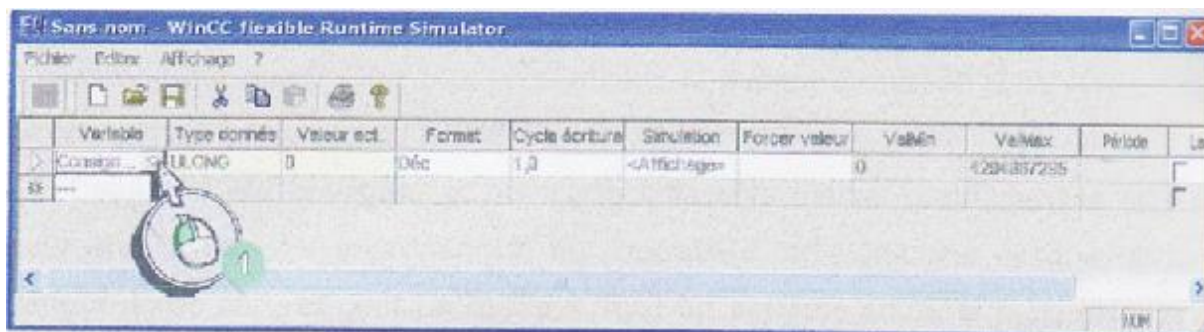
**V.9.2. Exemple de simulation du projet :**

La simulation permet de détecter des erreurs de configuration, et d'avoir un meilleur aperçu avant de mettre en œuvre le projet.

Ø Créer une table de simulation :

1. Démarrer le simulateur :





**VI. Conclusion :**

Dans ce chapitre, nous avons présenté notre correction en ce qui concerne la commande par panel. Ceci tout en éliminant le pupitre à entrées physiques par son remplacement par un panel tactile.

Celui-ci a été configuré et comporte différents menus graphiques et alarmes, pour une meilleure compréhension de l'opération, en particulier dans le cas d'un danger d'une part, et pour un meilleur suivi du système selon le programme de la machine crée d'autre part.

**CONCLUSION**

**GENERALE**

## **Conclusion générale :**

Le travail présenté dans ce mémoire, est axé sur une amélioration de la partie commande d'une machine à fabriquer les cales d'encoches et de son automatisation par un automate programmable S7-300 de la firme SIEMENS ainsi que sur l'amélioration du système de supervision par un panel tactile TP177A.

Le GRAFCET et les outils qui lui sont associés ont apporté bien des progrès en matière de méthodologie d'élaboration de cahier des charges, de réalisation et de programmation des systèmes automatisés.

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 que les API effectuent.

L'évolution des API ne cesse de continuer et notamment leurs logiciel 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 établi avant son implantation grâce à son logiciel de simulation S7-PLCSIM. Une bonne automatisation d'un procédé doit être performante et d'un cout optimal, cela est obtenu en passant par :

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

Toutes fois notre travail reste une solution qui participera, du moins nous l'espérons, d'une manière ou d'une autre, à apporter un éclairage et des possibilités nouvelles aux lecteurs qui seront amenés à utiliser l'automate S7-300 de Siemens.

# **BIBLIOGRAPHIE**

# BIBLIOGRAPHIE

- [1] : Claude Ducos « Automatismes Oléo-Hydraulique et pneumatique ».
- [2] : René David et Hassane Alla « Du Grafcet aux Réseau de Petri ».
- [3] : Gilles Michel « Les API Architecture et Application des Automates Programmables Industriels ».
- [4] : Site web : <http://cvdjafarhadiouche.ifrance.com/>
- [5] : Site web : <http://lamaintenance.fr/electricite/electricite-le-contacteur/>
- [5] : Site web : <http://wikipedia.org/wiki/Electrovanne>
- [6] : Siteweb :  
[http://industrial.omron.fr/fr/products/catalogue/sensing/inductive\\_sensors/special\\_models/e2fm/default.html](http://industrial.omron.fr/fr/products/catalogue/sensing/inductive_sensors/special_models/e2fm/default.html)
- [7] : Site web : <http://pagesperso-orange.fr/geea.org/PNEUM/Distributeurs.htm>
- [8] : Mémoire de fin d'étude : Mr Belkacemi Ahmed, Mr Boukherroub Samir « Etude de l'automatisation par automate programmable S7-300 de la chaîne de fabrication des armoires frigorifiques de l'ENIEM. Projet de fin d'études d'ingénieur. Département Automatique. FGEI. UMMTO. Année 2007.
- [9] : Documentation techniques de SIEMENS.

# **ANNEXES A :**

# **Logique câblée**

**ANNEXES B :**

**Menu du**

**Panel tactile**