

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE MOULOU MAMMERI, TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE  
DEPARTEMENT D'ELECTRONIQUE

# Mémoire de fin d'études

Présenté en vue de l'obtention  
du Diplôme d'Ingénieur d'Etat en Electronique

Option : Contrôle

**Thème :**

**Modernisation d'une machine à fabriquer  
les cales d'encoche par un API S7-200 et un  
Panel tactile TP177A**

**Proposé par :**

M<sup>f</sup>: A. HARBIT (Electro-Industries)

**Encadré par :**

M<sup>me</sup> : N. DJOUAHER

**Réalisé par :**

M<sup>elle</sup> : TOUDJINE Dyhia Meriem

M<sup>f</sup> : MOUAZER Salah

**Année universitaire 2009/2010**

## Remerciements

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, on exprime tout d'abord nos gratitudees et nos plus vifs remerciements à notre encadreur M<sup>r</sup> A. HARBIT pour son soutien, sa patience, ses conseils judicieux et pertinents.

On renouvelle notre gratitude envers notre promotrice M<sup>me</sup> N. DJOUAHER qui a assisté ce travail et qui nous a aidé à le concrétiser.

Nos remerciements s'adressent aussi à tous nos enseignants pour la formation qu'ils nous ont inculquée.

Et finalement, à tous ceux qui nous ont aidé de

## **Résumé :**

L'automatisation, dont l'histoire est déjà assez longue redevient ces dernières années le centre de préoccupation d'un nombre d'entreprises, c'est pour cette raison qu'on a alors proposé ce travail, qui est axé sur une amélioration de la partie commande d'une machine à fabriquer les cales d'encoche et de son automatisation par un automate programmable S7.200(CPU226)

De la firme SIEMENS, ainsi que sur l'amélioration du système de supervision par un panel tactile TP177A.

L'utilisation du grafcet comme outils de modélisation, méthode d'analyse fiable, juste, robuste, et qui nous a faciliter la transcription de ce model en un langage de programmation avec le logiciel STEP7 MICROWIN V4.0. Ceci nous à faciliter l'élaboration d'une solution programmable à cette chaine de traitement, et par la suite configurer le panel tactile TP177A et par le logiciel WINCC FLEXIBLE.

## Introduction générale :

L'automatisation, dont l'histoire est déjà assez longue redevient ces dernières années le centre de préoccupation d'un bon nombre d'entreprises, grâce à son évolution vers l'étude et la maîtrise des systèmes de plus en plus complexe. Elle permet l'exécution et le contrôle des tâches techniques par des machines fonctionnant sans intervention humaine, ou à l'aide d'une intervention réduite. Par conséquent, elle améliore la vie quotidienne et permet une augmentation constante de la productivité, ce qui permet de réduire considérablement le temps de travail.

Pour automatiser un système industriel, on doit d'abord connaître toutes ses structures et toutes les exigences que son fonctionnement présente. Ensuite, à l'aide des outils méthodiques, on entame une suite d'analyses, d'études et d'essais. Cette démarche peut se résumer en deux phases : Phase d'étude et phase de conception (programmation).

La part que prennent les automates programmables dans l'industrie algérienne est de plus en plus importante. Cependant, les personnes qui ont effectivement accès à un automate pour mettre en œuvre leurs connaissances et réaliser un projet complet sont rares. Elles relèvent principalement des entreprises spécialisées dans ce domaine.

Il nous a alors paru intéressant de nous attaquer à cette question, qui consiste à explorer le plus largement possible cette voie, en nous intéressant aussi bien aux aspects matériels que logiciels des automates, tout en restant autant que possible dans le domaine du concret.

Elle consiste aussi à explorer les nouveaux systèmes de commande par les panels tactiles, qui de nos jours, remplacent tous les pupitres à entrées physiques grâce à leurs multiples avantages.

Depuis son ouverture en 1983, 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 commandes 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 « **SIEMENS** » pour automatiser ses machines de production.

Dans les chaînes de production, les automatismes occupent une place importante. Le complexe **Electro-Industries** 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 de technologie nuit énormément à la productivité en présentant beaucoup 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'encoche ».

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 physiques) pour un meilleur suivi et une meilleure commande de la machine et d'apporter quelques améliorations 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 avec Grafcet ;
- Chapitre III : Description de l'automate programmable (API) ;
- Chapitre IV : Programmation ;
- Chapitre V : Configuration du panel tactile à l'aide du WinCC Flexible ;

## Sommaire

Introduction Générale .....	1
<b>Chapitre I : Etude de la machine à fabriquer les cales d'encoches</b>	
I. Présentation de l'organe d'accueil .....	3
I.1. Composition du complexe d'Electro-Industries .....	3
I.1.1. Unité de transformateurs .....	3
I.1.2 Unité moteur, alternateur et groupe électrogène (UMAGE) .....	3
II. Description de la machine .....	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 .....	17
II.2.3. Capteurs .....	18
a. Capteurs magnétiques .....	18
b. Thermostat .....	20
c. Encodeur rotatif .....	21
II.2.4. Le circuit électrique de protection .....	22
a. Circuits RC .....	22
b. Les fusibles .....	23
c. Les contacteurs .....	23
d. Transformateur galvanique (220/220 V) .....	24
II.2.5. Variateur de vitesse électronique .....	24
II.2.6. Moteur à courant continu .....	24
II.2.7. Résistances chauffantes .....	25

II.3. Partie commande .....	25
II.3.1. Pupitre de commande .....	25
II.3.2. Commande de la machine .....	25
III. Fonctionnement de la machine .....	26
Conclusion .....	28
Cahier des charges .....	29

## **Chapitre II : Modélisation avec l'outil GRAFCET**

I. Introduction .....	30
II. Définition .....	30
III. Les éléments de base du GRAFCET .....	31
III.1. Les étapes .....	31
III.2. Les actions associées à une étape .....	32
III.3. Les transitions et les réceptivités associées .....	33
III.3.1. Transition .....	33
III.3.2. Réceptivité .....	33
III.4. Les règles d'évolution du GRAFCET .....	34
III.5. Sélection de séquence et séquence simultanées .....	35
IV. Niveaux d'un GRAFCET .....	36
IV.1. GRAFCET de niveau 1 .....	36
IV.2. GRAFCET de niveau 2 .....	36
V. Création et simulation du GRAFCET .....	36
V.1. Présentation du logiciel Automgen 8 .....	36
V.2. Utilisation d'Automgen 8 .....	37
V.2.1. Lancement d'Automgen 8 .....	37
V.2.2. Ouvrir un nouveau projet .....	37
V.2.3. Exécution de l'application .....	40
V.2.3.1. Compilateur .....	40
V.2.3.2. Exécution .....	40
V.2.3.3. Exemple de simulation du GRAFCET .....	41
V.2.3.4. Exemple de simulation de « Automatism » et « IRIS » .....	41
VI. Liste des actionneurs, pré actionneurs et capteurs .....	42
VI.1. Les actionneurs et pré-actionneurs .....	42

VI.2. Les capteurs .....	42
VI.3. Les boutons .....	42
VI.4. Compteur .....	42
Conclusion .....	43

### **Chapitre III : Description de l'automate programmable (API)**

I. Introduction .....	44
II. Structure interne d'un automate programmable .....	45
II.1. L'unité centrale (UC) .....	46
II.2. Bloc d'alimentation .....	46
II.3. Les modules enfichables .....	46
II.4. Modules d'entrées / sorties .....	47
III. Fonctions d'un automate programmable .....	48
IV. Présentation de l'automate S7 200 .....	49
IV.1. Introduction .....	49
IV.2. Principaux composants de l'API S7-200 .....	50
IV.2.1. CPU S7-200.....	50
IV.2.2. Module d'extension .....	51
V. Choix de la CPU 226 .....	52
Conclusion .....	53

### **Chapitre VI : Programmation**

I. Logiciel de programmation STEP7-Micro/Win .....	54
I.1. Définition .....	54
I.2. Organisation du programme de commande .....	54
I.2.1. Programme principal (PPAL).....	54
I.2.2. Sous-programme (SBR_i) .....	54
I.2.3. Programmes d'interruption (INT_i) .....	54
II. Fonctionnement du programme de commande .....	56
III. Accès aux données dans les zones de mémoire .....	56
III.1. Mémoire image des entrées (I) .....	57
III.2. Mémoire image des sorties (Q) .....	57
III.3. Mémoire des variables (V) .....	57



III.4. Mémentos (M).....	57
III.5. Temporisations (T).....	57
III.6. Compteur (C).....	58
III.7. Accumulateurs (AC).....	58
III.8. Mémentos spéciaux (SM).....	58
III.9. Compteurs rapides (HC).....	58
IV. Fonctionnement des compteurs rapides (HC).....	59
IV.1. Programmation d'un compteur rapide.....	61
IV.1.1. Définition du compteur et de son mode.....	61
IV.1.2. Définition de l'octet de commande.....	61
IV.1.3. Définition de la valeur en cours et prédéfinie.....	62
IV.1.4. Affectation et validation du programme d'interruption.....	62
IV.1.5. Validation du programme d'interruption.....	63
IV.1.6. Activation du compteur rapide.....	63
IV.1.7. Programme d'interruption (EG_COMPT).....	63
IV.2. Programmation de la commande de la machine.....	64
IV.2.1. Programme Principale (PPAL).....	64
IV.2.2. Sous-programme des (SORTIES).....	71
IV.2.3. Sous-programme d'Alarmes.....	72
IV.2.4. Programme d'initialisation du compteur rapide.....	73
IV.2.5. Programme d'interruption (EG_COMPT).....	74
Conclusion.....	75

## **Chapitre V : Configuration du panel tactile à l'aide du WinCC Flexible**

I. WinCC Flexible.....	76
I.1. Introduction au WinCC.....	76
I.1.1. WinCC flexible Advanced.....	76
I.1.2. WinCC flexible Runtime.....	76
II. Choix du panneau tactile TP 177A.....	78
III. Etapes de configuration.....	79
III.1. Créer un projet.....	79
III.2. Configuration de la liaison.....	81
III.3. Création d'une vue (Menu).....	81
III.4. Création de variables.....	82

III.5. Configuration d'une alarme .....	83
III.6. Configurer l'affichage d'alarmes .....	84
III.7. Ajout de changements de vue .....	86
III.8. Définir la fonction STOP_RUNTIME .....	87
III.9. Test et simulation du projet .....	88
III.9.1. Test du projet (Compilation) .....	88
III.9.2. Simulation du projet .....	88
Conclusion .....	90
Conclusion générale .....	91
Bibliographie	
Annexes A	
Annexes B	
Annexes C	

## **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 :

- **Unité fabrication des transformateurs** de basse et moyenne tension.
- **Unité de fabrication de moteurs électriques, alternateurs et montage de groupes.**

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

#### **I.1.1. Unité de transformateurs :**

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 nécessaires à la fabrication et aux essais des produits finis.

En vue de développer ses transformateurs en les rendant secs et hermétiques, du coup, moins encombrant et peu de matières premières, l'entreprise est en partenariat avec des entreprises étrangères.

#### **I.1.2. Unité moteur, alternateur et groupe électrogène (UMAGE) :**

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

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

Les produits fabriqués dans cette unité sont :

Alternateurs synchrones, groupes électrogènes et les moteurs.

### ✚ **Les moteurs :**

Les moteurs fabriqués par cette unité sont :

- Moteur asynchrone triphasé.
- Moteur asynchrone triphasé à deux vitesses.
- Moteur asynchrone monophasé.
- Moteur asynchrone monophasé à deux bouts d'arbre.

Les moteurs à rotor triphasés sont les plus utilisés dans l'industrie, du fait de leur robustesse, de leur prix peu élevé et de leur grande durée de vie.

Les deux principales parties d'un moteur à rotor triphasé sont :

- *Le STATOR* qui produit un champ magnétique tournant ;
- *Le ROTOR* qui, entraîné par ce champs tournant, produit de l'énergie mécanique.

Le stator est constitué d'un empilage de tôles magnétiques 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 », comme l'indique la Figure 1.



**Fig.1 – Cales d'encoche d'un stator <sup>(1)</sup>**

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

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

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

## 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 du ruban. Elle comporte :

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

largeur des cales d'encoches.

— Pour alimenter le tambour d'un nouveau rouleau de ruban pour cales, on procède comme suit :

- 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.

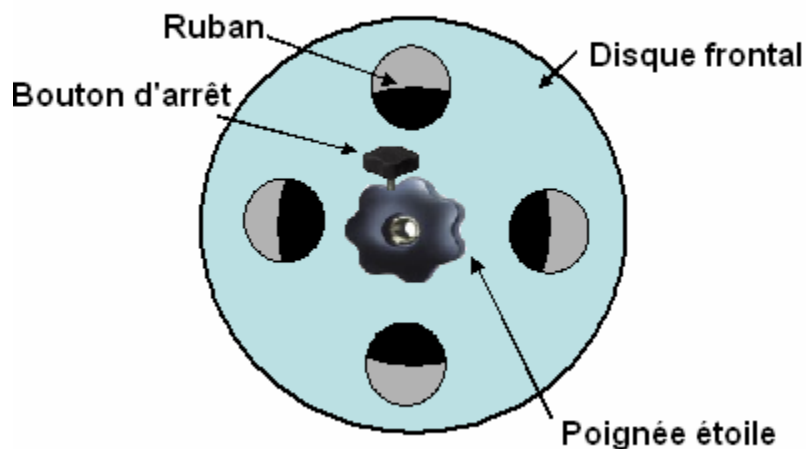
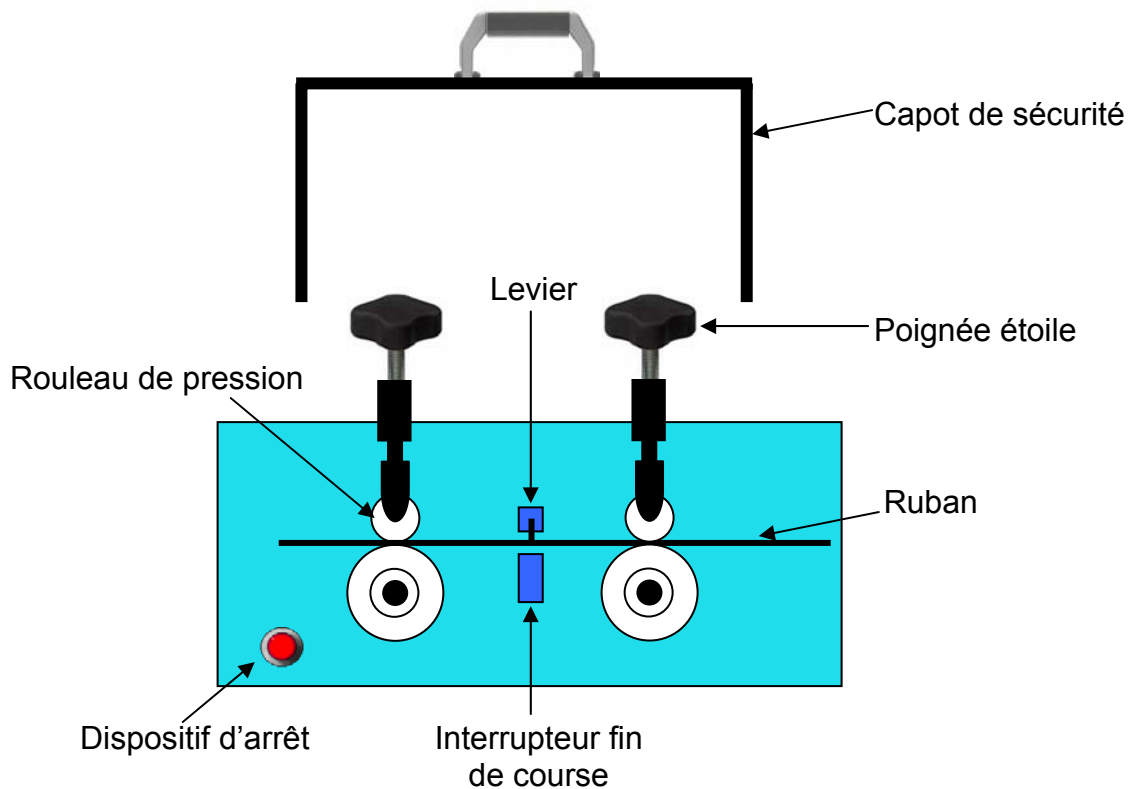


Fig.3 – Bobine à ruban.

### 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, servant 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 ;



**Fig.4 – 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 ;

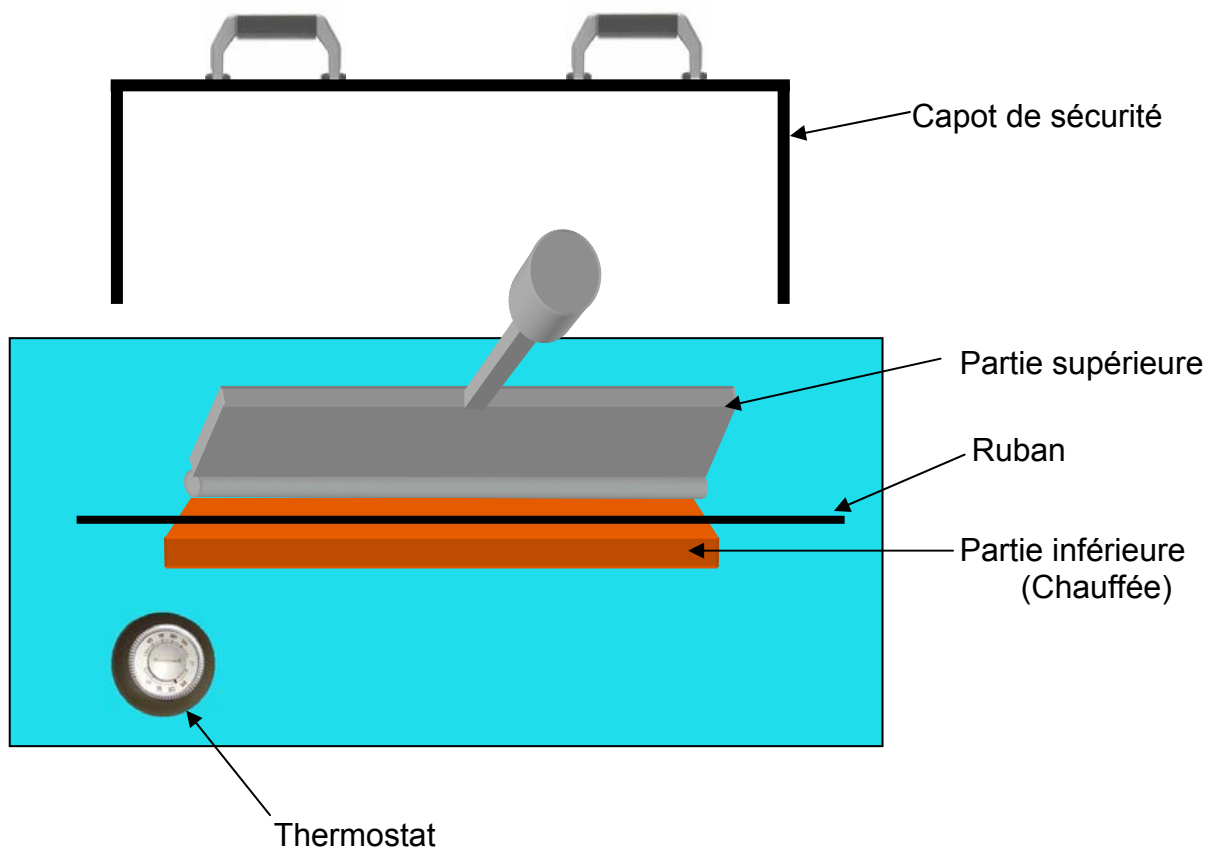


Fig. 5 – Unité de chauffage



### 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 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 à ajuster selon la largeur du ruban.
- Poignées étoiles, servant à varier la force de pression des roues de mise en forme.
- Les écrous de fixation.

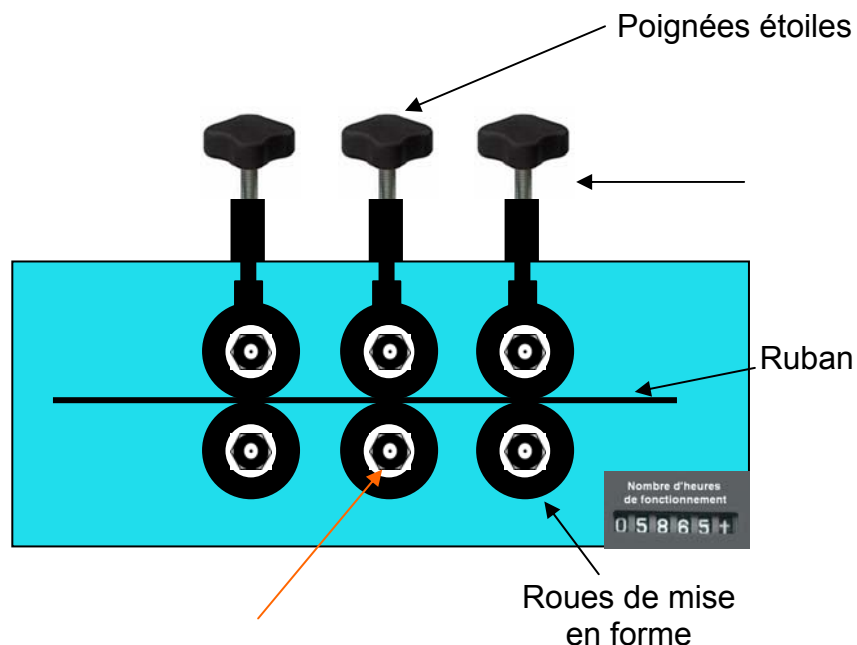


Fig.6 – Unité de mise en forme

### 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 ;

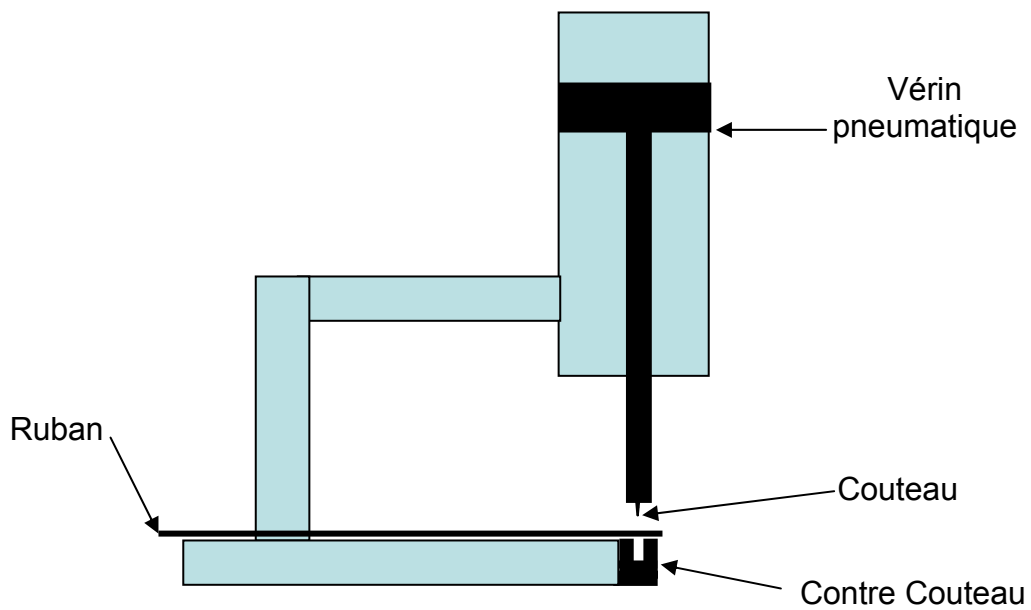
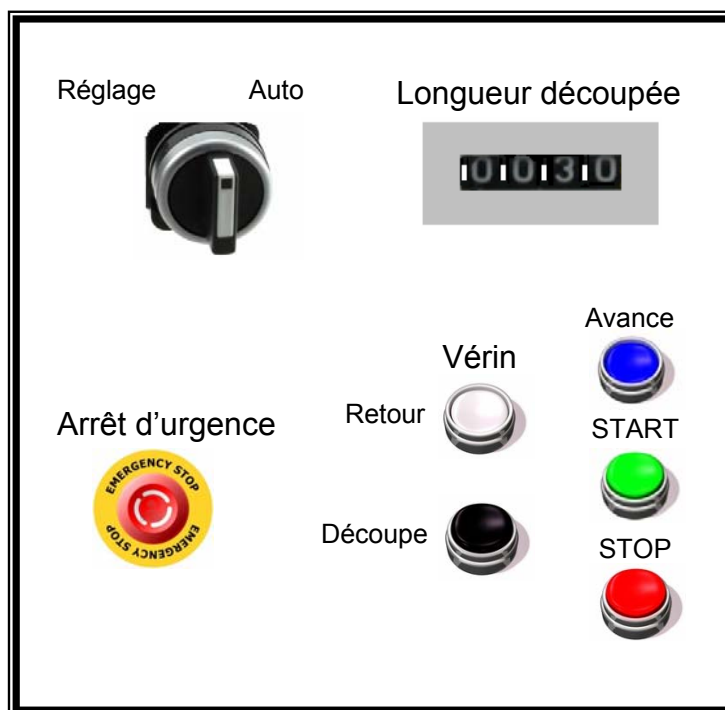


Fig. 7 – Unité de découpage

### II.1.6. Panneau de commande :

Il comprend les boutons de commande suivants :

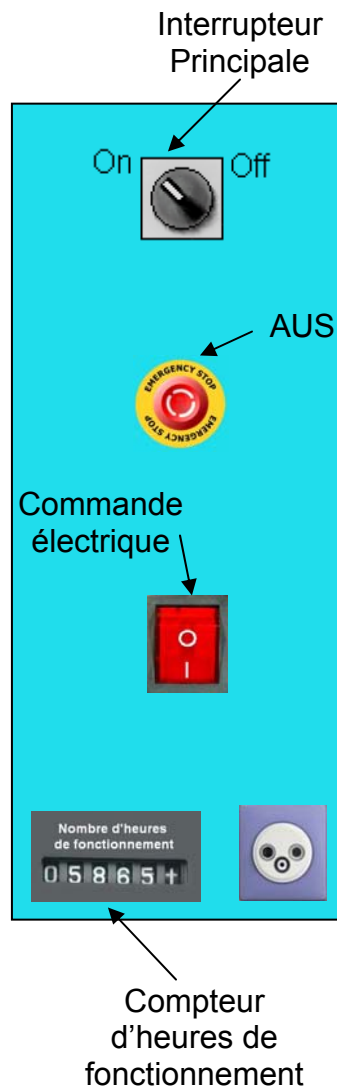
- Sélecteur Réglage-Auto : choisit le mode de travail de la machine (Manuel ou Automatique) ;
- Présélecteur longueur des cales : sélectionne la longueur de la cale souhaitée ;
- Avance : commande le moteur manuellement. ;
- Bouton d'arrêt d'urgence ;
- Boutons Retour et Découpe : commande manuelle le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



**Fig. 8 – Panneau de commande (Pupitre)**

### 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.



**Fig. 9 – Armoire électrique**

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

Elle est destinée à trier les cales selon leur longueur.



**Fig. 10 – 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 est un composant de gestion de l'énergie de commande d'un actionneur, auquel il est associé. On citera le distributeur pneumatique et l'électrovanne « Tous Ou Rien ».

#### a. Distributeur pneumatique :

Ils ont pour fonction essentielle de distribuer l'air dans des canalisations qui aboutissent aux chambres des vérins. Le distributeur est donc le pré-actionneur associé à un vérin pneumatique.

Comme le vérin de découpage de notre machine est un vérin à double effet et comporte donc deux orifices sur lesquels il faut alterner les états de pression et d'échappement. On utilise un distributeur 4/2 comportant 4 orifices : deux orifices de sortie « 1 et 2 », un orifice d'entrée « P » (à partir d'un compresseur : source d'air comprimé) et un orifice d'échappement « T », et deux chambres, actionné par une électrovanne et un retour à ressort. Sa représentation schématique est la suivante :

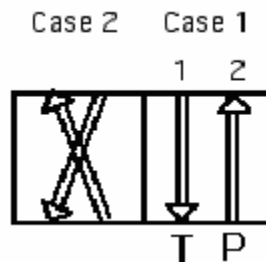


Fig. 11 – Distributeur 4/2

#### b. Electrovanne « Tout Ou Rien » ou TOR :

Lorsque le traitement de l'information est réalisé par l'automate, il est nécessaire que les distributeurs soient équipés d'une ou de deux électrovannes, dont le rôle est de transformer un signal électrique en un signal pneumatique de pilotage du distributeur.

Une électrovanne est un dispositif commandé électriquement. Elle autorise ou interrompt par une action mécanique, la circulation de l'air dans un circuit pneumatique.

Les électrovannes TOR sont des électrovannes qui ne peuvent s'ouvrir qu'en entier ou pas du tout. L'état change suivant qu'elle soit alimentée électriquement ou non.

La figure suivante illustre une électrovanne montée sur un distributeur 4/2.



**Fig. 12 – Electro-distributeur <sup>(2)</sup>**

➤ **Principe de fonctionnement :**

Une fois l'électrovanne est excitée par un signale de commande, la bobine crée un champ magnétique qui attire le noyau. L'air circule de P vers 1, et de 2 vers T (Fig. 13) et engendre la descente du vérin.

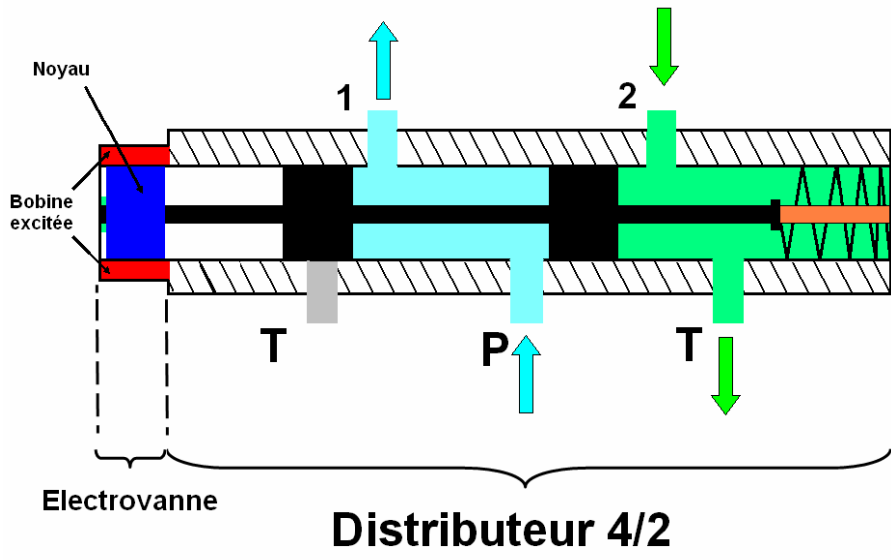


Fig. 13 – Electrovanne excitée

De même, quand l'électrovanne est désexcitée (absence du signal de commande), la bobine ne crée plus de champ magnétique, et le noyau revient à sa position de repos grâce au ressort. L'air circule donc de P vers 2 et de 1 vers T (Fig. 14) et entraîne la remontée du vérin.

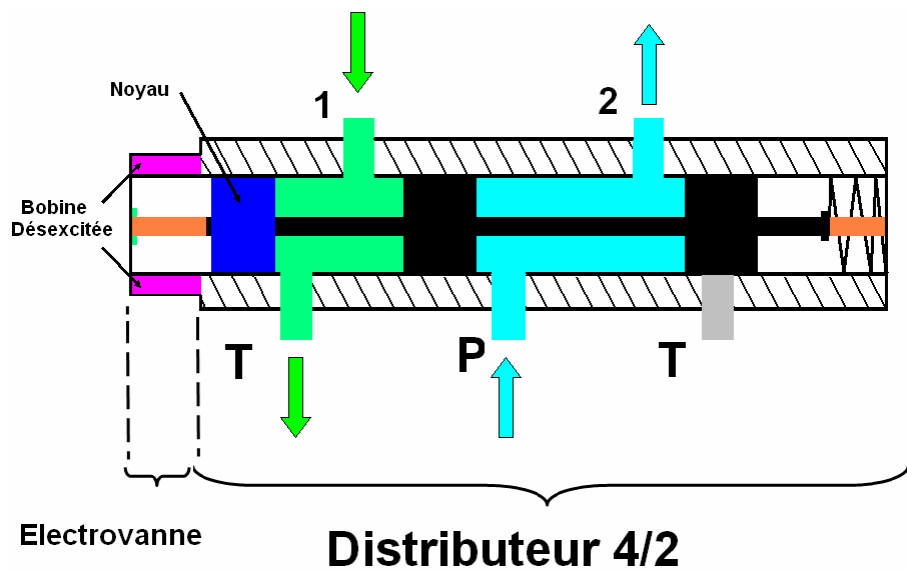


Fig. 14 – Electrovanne désexcitée



## II.2.2. Les actionneurs :

Ils convertissent l'énergie de puissance pneumatique en énergie mécanique de translation. Parmi eux, on citera le vérin pneumatique à double effet.

### ✓ Vérin pneumatique double effet :

Un vérin pneumatique est un actionneur linéaire dans lequel l'énergie de l'air comprimé est transformée en travail mécanique.

Cet actionneur de conception robuste et de simplicité de mise en œuvre, est utilisé dans toutes les industries manufacturières.

- ✚ 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 du vérin, et ça descend de façon à découper le ruban. La position basse du vérin désexcite la bobine pour un retour à l'état initial par effet du ressort.

La figure suivante représente notre vérin en position repos et découpe

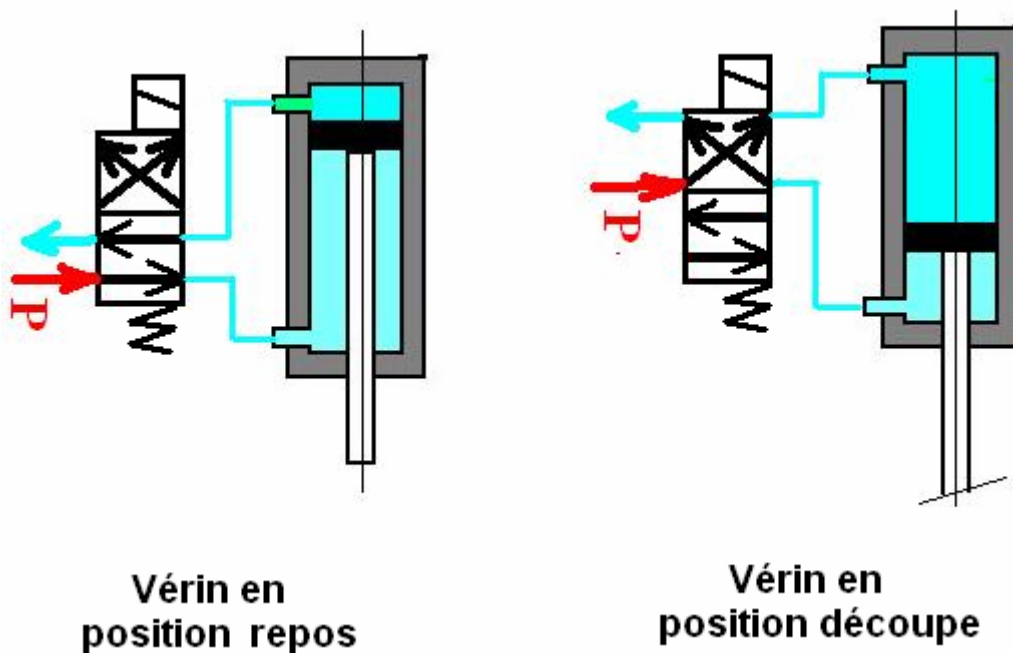


Fig. 15 - Vérin en position repos et découpe

### II.2.3. Capteurs :

Ils sont utilisés pour la détection des événements qui se produisent dans le procédé. Ils émettent des signaux après traitement de l'information de la partie opérative vers celle de commande, pour commander les pré-actionneurs et actionneurs.

#### a. Capteurs magnétiques :

Les capteurs magnétiques existants sur notre machine sont des capteurs de position (proximité) magnétiques. Ils sont caractérisés par l'absence de liaison mécanique entre le dispositif de contrôle (capteur) et l'objet cible. Ils se caractérisent par rupture ou pas du champ magnétique.

On y distingue deux types : Les capteurs excités par un objet métallique (capteurs inductifs), et ceux excités par la présence d'un aimant permanent (Interrupteur à lames souples).

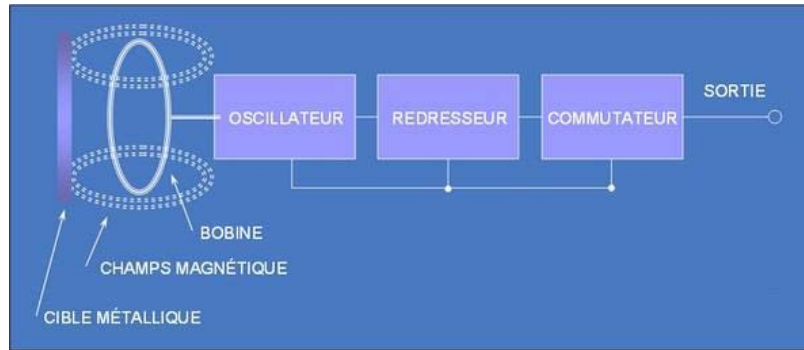
#### ❖ Capteurs de proximité inductifs :



Fig. 16 – Capteur de proximité inductif <sup>(3)</sup>

➤ **Principe de fonctionnement :**

Les capteurs inductifs produisent à l'extrémité de leur tête de détection un champ magnétique oscillant. Lorsqu'un objet métallique pénètre dans ce champ, il est perturbé puis atténué. Cette variation est exploitée par un amplificateur, qui délivre un signal de sortie et donc le capteur commute comme le montre la figure suivante :



**Fig. 17 – Principe de fonctionnement d'un capteur 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 continu.

❖ **Interrupteurs à Lames Souples (I.L.S) :**



**FIG. 18 – Interrupteur à lames souples <sup>(4)</sup>**

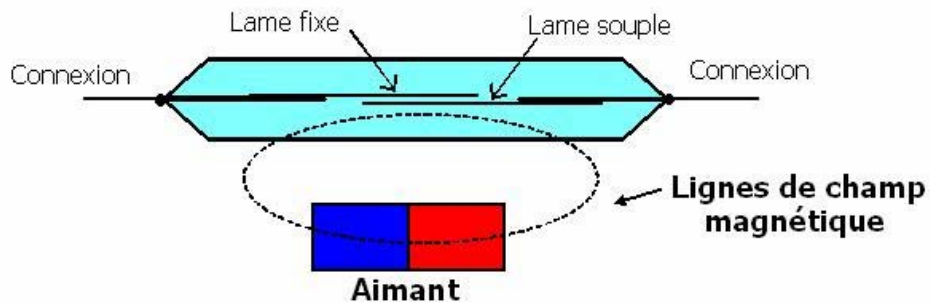
➤ **Principe de fonctionnement :**

Un I.L.S est un Interrupteur à Lame Souple, comportant deux lames dans une enveloppe de verre (FIG.18), reliée chacune à une électrode extérieure que l'on peut souder.

En rapprochant un aimant de l'I.L.S (présence d'un champ magnétique), la lame souple se rapproche de celle rigide et constitue ainsi un interrupteur fermé.

En éloignant l'aimant, la lame souple reprend sa position initiale (interrupteur ouvert). Il est donc assez aisé de mettre en place une détection d'ouverture de contact, en plaçant par exemple deux I.L.S sur les deux extrémités du vérin et un aimant sur le piston, de telle sorte que, les deux éléments soient proches l'un de l'autre quand le piston est en position haute ou basse.

La figure suivante est un schéma d'un ILS en présence d'un aimant permanent.



**Fig. 19 – ILS en présence d'un champ magnétique**

- ✚ Ces capteurs se trouvent sur les deux extrémités du vérin de l'unité de découpage. Leur but est la détection des positions haute et basse du vérin, grâce à un aimant solidaire au piston du vérin. Ces informations sont nécessaires au démarrage du moteur et à la descente du vérin.

### ***b. Thermostat :***

Un thermostat comme décrit dans la (Fig. 20), est un système électronique à sonde thermique. Il comporte un potentiomètre pour sélectionner une température de consigne.

Il est aussi constitué d'un système à bilame qui ouvre un circuit électrique (le contact du thermostat), quand la température de consigne est atteinte, et ferme ce circuit dès que l'on descend en dessous de cette même température.



**Fig. 20 – Thermostat <sup>(5)</sup>**

- ✚ 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 en dessous de la consigne), le contact se ferme, la bobine s'excite et alors le ruban est chauffé.

### ***c. Encodeur rotatif :***

La croissance de la puissance des systèmes de traitements ainsi que les impératifs de productivité, appellent dans tous les domaines de productions industrielles un besoin d'information continue sur : le déplacement, la position et la vitesse des outils ou des produits. L'encodeur répond parfaitement aux besoins de précision et flexibilité par rapport aux systèmes de détection « tous ou rien » tel que les interrupteurs.

Un encodeur rotatif, également appelé codeur, est un dispositif électromécanique qui convertit l'angle de position d'un arbre ou d'un essieu à un code numérique ou analogique.

- ✚ l'encodeur de notre machine est solidaire à l'arbre de rotation du moteur (Fig. 21), Il se charge de délivrer des impulsions générées à partir du moteur continu.

Ces dernières seront traitées directement par des cartes électroniques adéquates à l'encodeur. Leur conversion d'un signal analogique (délivré) en numérique permet la commande du déplacement du ruban.

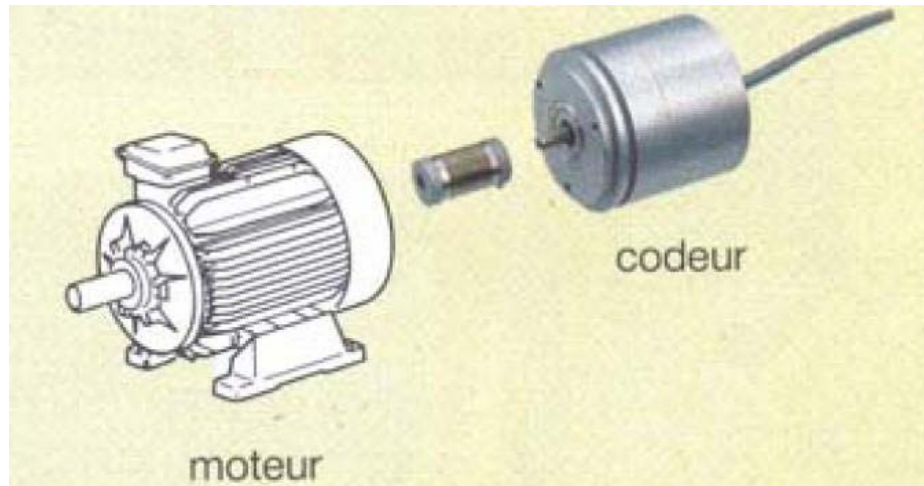


Fig. 21 – Montage de l'encodeur rotatif sur l'arbre du moteur

#### 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 cite :

- *Les circuits RC ;*
- *Les fusibles ;*
- *Les contacteurs ;*

##### **a. 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 du 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 tel que les circuits RC.

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

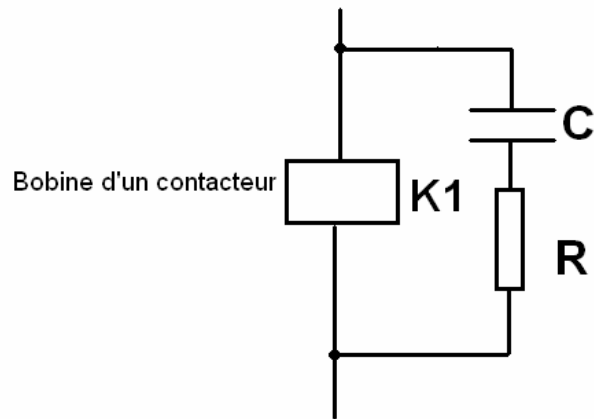


Fig. 22 – Circuit RC de protection pour Bobine de contacteur

**b. Les fusibles :**

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

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



FIG. 23 – Symbolisation de fusible

**c. Les contacteurs :**

Le contacteur est un appareil mécanique de connexion ayant une seule position repos, commandé, capable d'établir, de supporter et d'interrompre des courants dans les conditions normales du circuit, y compris les conditions de surcharge en service.

La constitution d'un contacteur est faite d'un organe de commande (bobine) et d'organes commandés (contacts).

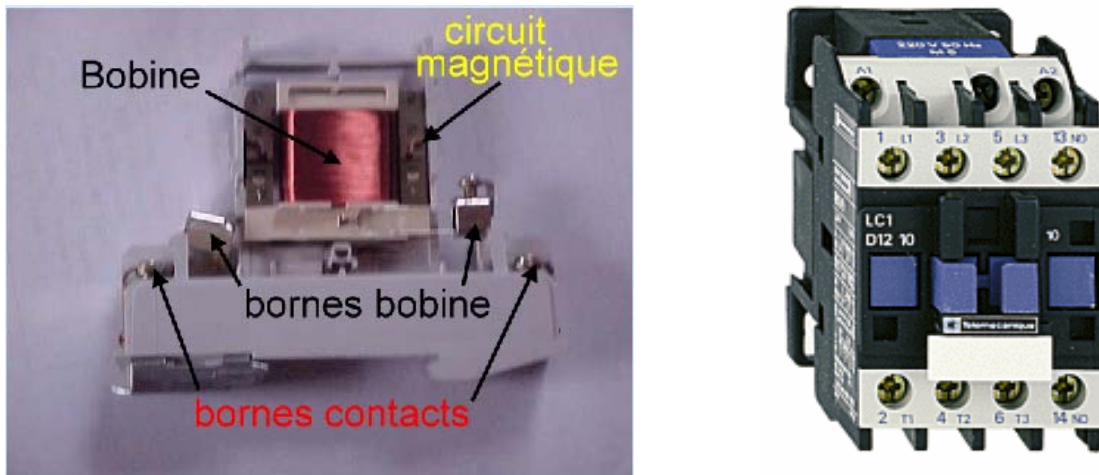


Fig. 24 – Contacteurs <sup>(6)</sup>

#### **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 :**

C'est une carte électronique. Elle a pour rôle la variation de la vitesse du moteur à l'aide d'un potentiomètre où on ajuste la vitesse de consigne. Possédant également une entrée de déblocage de façon à activer le variateur.

✚ 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 (le déplacement) 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 (04) résistances de 600 Watts chacune alimentées par 220 V.

- ✚ On les retrouve dans l'unité de chauffage et servent à chauffer le ruban pour le rendre plus élastique et facile à être mis en forme par l'unité suivante.

## 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 cas de mauvais fonctionnement (à l'aide du bouton d'arrêt d'urgence), ou bien d'assurer certains réglage préalables avant le démarrage de la machine (mode Réglage).

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

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

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

- 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.

- 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 encore un arrêt simple ou celui d'urgence.

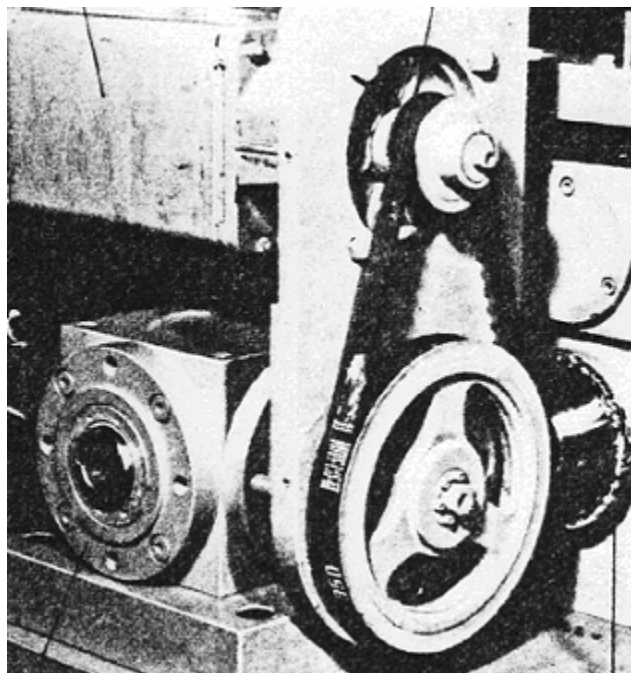
### III. Fonctionnement de la machine à fabriquer les cales d'encoches :

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 tendeuses.

L'axe de la roue de mise en forme centrale est entraîné par un moteur à courant continu, au moyen d'une courroie dentée (réduction 1 : 3) et d'un réducteur à roues coniques (réduction 1 : 5).



**Fig. 25 – Système d'entraînement des roues de mise en forme (réducteurs)**

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 encore l'encodeur rotatif dont 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 lorsque il n'y a plus de détection de matière et donc 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 actionné 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.

## Position du problème et solutions apportées :

✚ La commande actuelle de la machine à fabriquer les cales d'encoches (technologie logique câblée), présente plusieurs insuffisances dont :

- Espace important : occupé par les différentes cartes électriques, de la logique câblée (CAN, portes logiques, compteurs, bascules,...), ainsi que leur câblage.
- Modification de la commande implique des modifications de câblage.
- Difficulté de maintenance en cas de pannes.

Afin de pallier à ces insuffisances nous avons opté pour une commande programmable par API (et donc une automatisation) dont les avantages principaux seraient :

➤ Au niveau de l'armoire électrique, toute la logique câblée sera supprimée ainsi que les cartes électroniques reliées à l'encodeur (gain d'espace).

➤ L'alimentation des différents capteurs et de l'électrovanne se fera directement à partir de l'automate, qui commandera aussi la bobine du contacteur relié aux résistances chauffantes, de manière à fermer ou ouvrir tous les contacts de puissances.

➤ L'activation de l'entrée de déblocage du variateur de vitesse se fera désormais directement par l'automate (par un 24V).

✚ Le pupitre actuel (à entrées physiques), présente des restrictions sur le contrôle de la commande et la commande elle même. Par exemple :

- L'utilisateur ne peut pas suivre le déplacement du ruban (en particulier dans le mode manuel).

De manière à pouvoir tirer un trait sur les différentes insuffisances, nous avons proposé d'insérer un pupitre tactile. Offrant ainsi une amélioration en créant plusieurs menus au lieu d'un (pupitre actuel), sur lesquels on élargira la commande et le suivi du système. Par exemple Créer un menu :

- Qui commandera le nombre de ruban à couper.
- Qui pourra suivre le déplacement du ruban.

La configuration de cet Ecran Tactile se fera par le biais du logiciel WinCC Flexible.

## **I. Introduction :**

Pour être analysé, communiqué et compris, le fonctionnement d'un système automatisé doit être modélisé selon les moments de la vie de ce dernier. Pour répondre à des besoins précis et variés, les modèles sont différents (dessin mécanique, algorithme, etc....).

Le fonctionnement d'un système automatisé peut être représenté graphiquement par un ensemble appelé GRAFCET.

## **II. Définition :**

Le Graphe de Commande Etapes Transitions (GRAFCET), est un diagramme fonctionnel dont le but est de décrire graphiquement, suivant un cahier de charges, les différents comportements et évolution d'un système séquentiel. Il est à la fois simple à utiliser et constitue un outil de dialogue entre toutes les personnes collaborant à la conception, à l'utilisation ou à la maintenance de la machine à automatiser.

### III. Les éléments de base du GRAFCET :

La symbolisation du GRAFCET est représentée par la figure suivante :

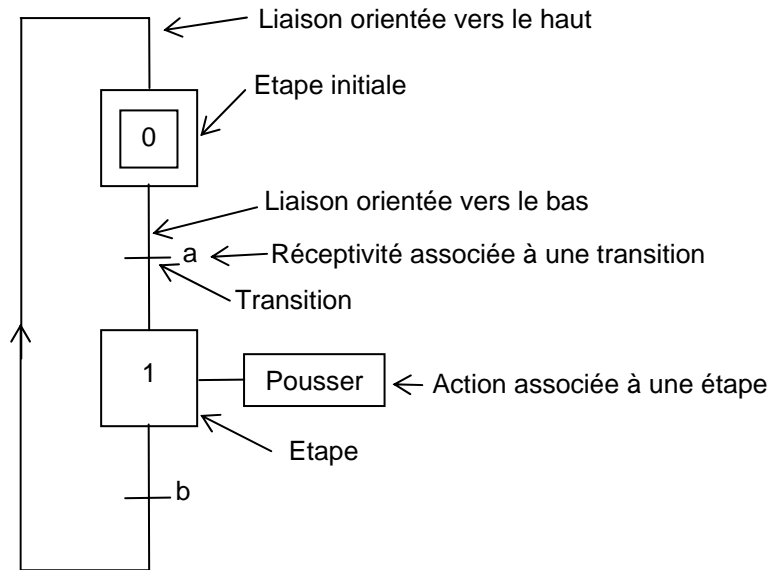


Fig.1 – La symbolisation du GRAFCET

#### III.1. Les étapes :

Une étape correspond à une phase durant laquelle on effectue une action pendant une certaine durée, l'action doit être stable, c'est-à-dire que l'on fait la même chose pendant la durée de l'étape. Lorsqu'une étape est active on peut le préciser par un point (marque).

Chaque étape porte son propre numéro. Lorsqu'on désigne une étape on fait précéder ce numéro de la lettre X majuscule.

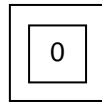
- $X_i=1$ , lorsque i est active.
- $X_i=0$ , lorsque i est inactive.



Fig.2 – Etape active (a) / Etape inactive (b)

➤ **Etape initiale :**

L'étape initiale représente l'étape active du système au début du fonctionnement (position initiale de la partie opérative). Elle se différencie en doublant les côtés du carré.



**Fig.3 – Etape initiale**

### III.2. Les actions associées à une étape :

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



**Fig.4 – Etape simple (e) / Etape à deux actions (f)**

- ✓ Une étape peut n'avoir aucune action, par exemple si elle correspond à une attente d'un événement.



### III.3. Les transitions et les réceptivités associées :

#### III.3.1. Transitions :

Une transition représente une possibilité de changement de comportement. Le passage d'un comportement au suivant correspond au franchissement de la transition. On représente une transition par un petit trait horizontal sur une liaison vertical.

#### III.3.2. Réceptivité :

La réceptivité est une condition logique qui doit être remplie pour le franchissement d'une transition, la réceptivité est inscrite à droite de la barre représentant la transition.

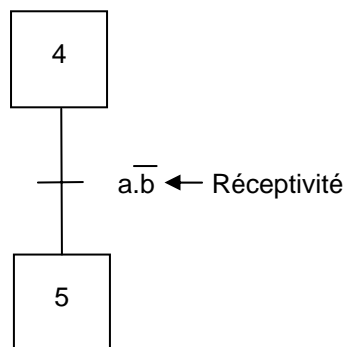


Fig.5 – Réceptivité associée à l'étape 4

#### Type de réceptivités :

##### ✓ Temporisation :

La temporisation est une réceptivité qui permet une prise en compte du temps, elle implique l'utilisation d'un temporisateur. Ce type de réceptivité est notée  $Q / X_i$ , où « i » est le numéro de l'étape comportant l'action de temporisation, et Q est la durée écoulée depuis l'activation de l'étape  $X_i$ .

##### ✓ Réceptivité à niveau :

La réceptivité à niveau est une réceptivité faisant intervenir une condition logique ( $a.\bar{b}$ ).

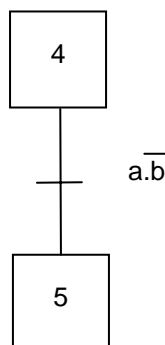


Fig.6 – Réceptivité à niveau

### III.4. Les règles d'évolution du GRAFCET :

- **Règle 1 (situation initiale) :**

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

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

Une transition est dite validée lorsque toutes les étapes en amont sont actives. Le franchissement d'une transition se produit, lorsque la transition est validée et la réceptivité associée à cette dernière est vraie.

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

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

- **Règle 4 (évolution simultanée) :**

Plusieurs transitions simultanément franchissables sont simultanément franchies,

- **Règle 5 (activation et désactivation simultanée d'une même étape) :**

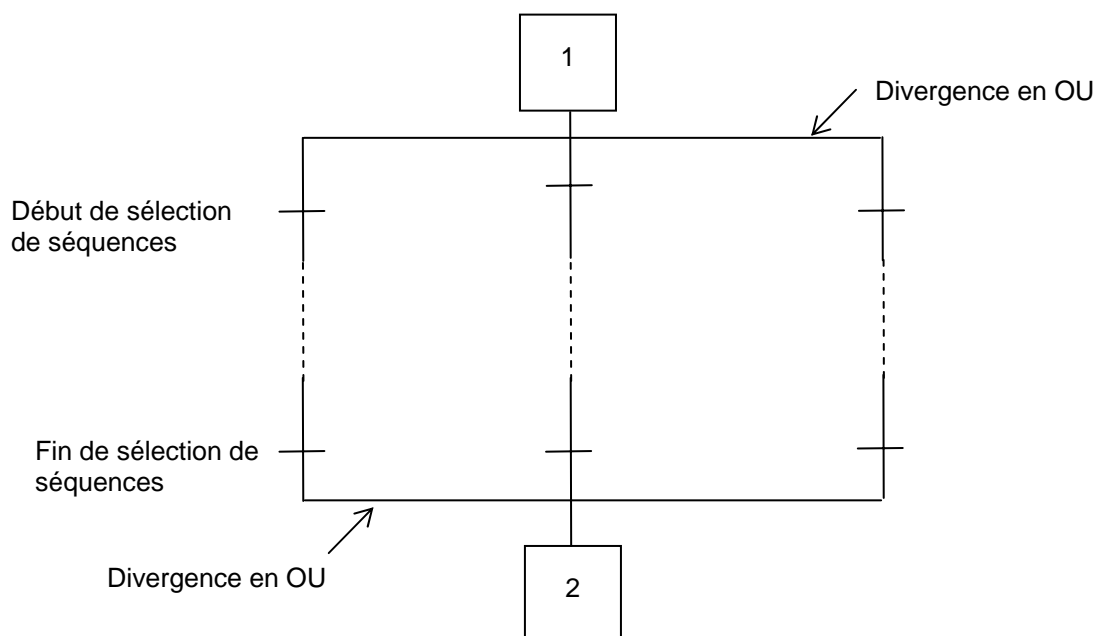
Si au cours du fonctionnement la même étape est simultanément activée et désactivée elle reste active.

### III.5. Sélection de séquence :

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

✓ **Sélection de séquence :**

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



**Fig.7 – Représentation graphique d'une sélection de séquence**

## **IV. Niveaux d'un GRAFCET :**

### **IV.1. 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.

### **IV.2. 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.

## **V. 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 8.

### **V.1. Présentation du logiciel Automgen 8 :**

Automgen 8 est un logiciel de conception d'automatismes, propriétaire édité par la société française IRAI; atelier d'automatisme, de supervision et de simulation de parties opératives 2D et 3D. Fonctionnant sur PC sous système d'exploitation WINDOWS, il est utilisé pour l'apprentissage des automatismes et dans l'industrie pour le développement d'applications.

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

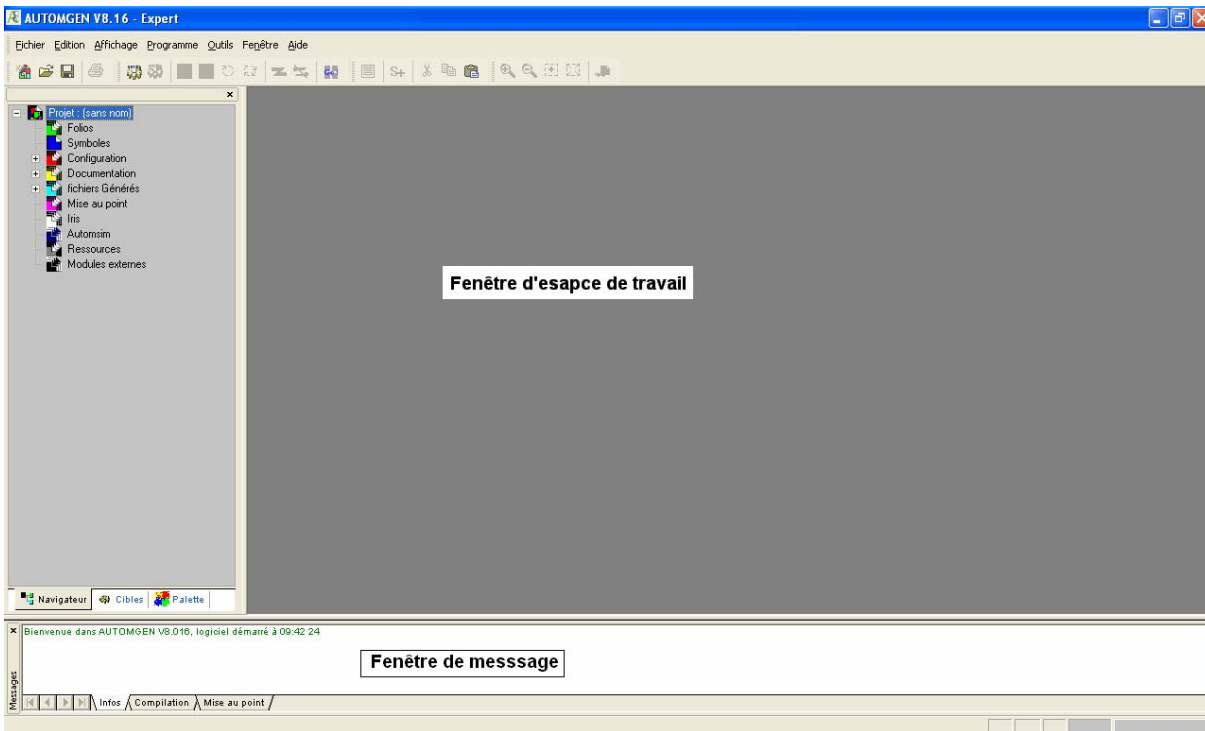
## V.2. Utilisation d'Automgen 8 :

### V.2.1. Lancement d'Automgen 8 :

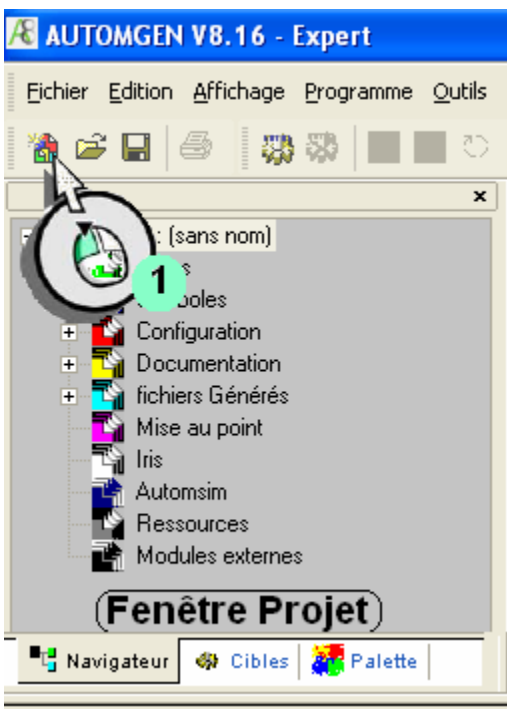
Pour démarrer AUTOMGEN, on clique deux fois sur l'icône



La fenêtre principale s'affiche :



### V.2.2. Ouvrir un nouveau projet :



La fenêtre projet permet le choix entre 3 menus :

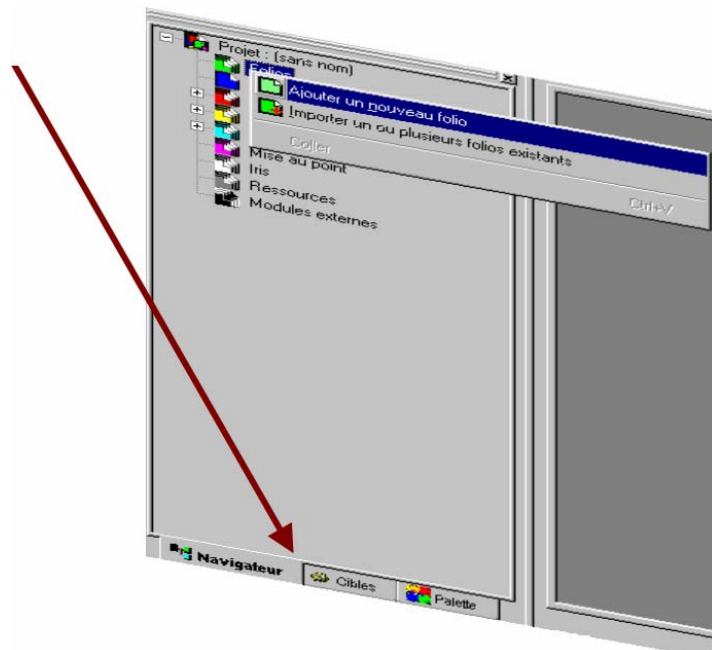
- Navigateur
- Cible
- Palette

**1. Cibles** : permet de sélectionner le type de traducteur (exécuteur PC ou automate programmable) avec laquelle on va effectuer la compilation du projet.

**2. Palette** : permet d'accéder à des éléments de logigrammes, de blocs fonctionnels, de GRAFCET, de ladder et la liste des symboles à utiliser lors de la création des programmes dans les folios (page GRAFCET).

**3. Navigateur** : il permet de créer les différents éléments d'une application et d'y accéder rapidement :

- Folios ;
- Table de symbole ;
- Configuration du Post-processeur (adressage, ..) ;
- Configuration matérielle ;
- Monitoring (Mise au point) ;
- Modules IRIS 2D et 3D ;
- Ressources (objets IRIS 2D, objets 3D, ....) ;
- Modules externes ;



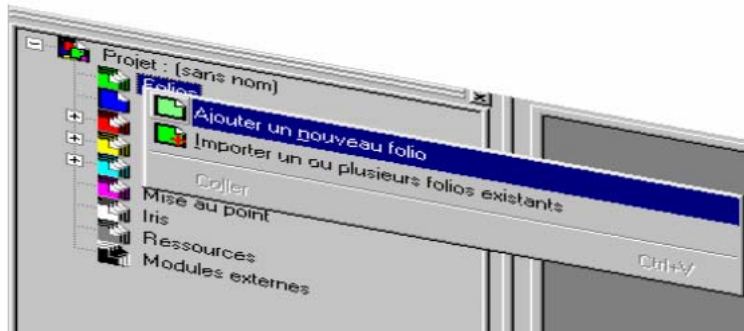
✚ Les menus suivants sont ceux utilisés dans la création de notre projet :

✓ **FOLIOS :**

Un folio est une page sur laquelle est dessiné le GRAFCET.

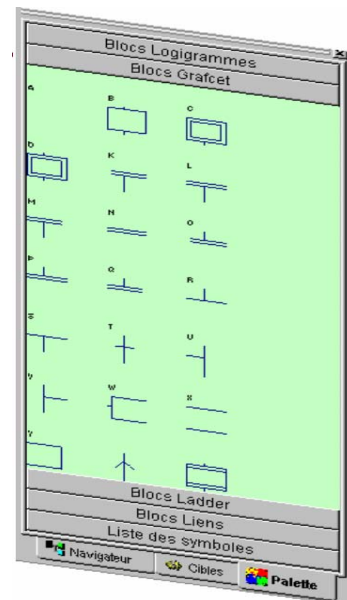
• Dans le menu « *Navigateur* » on clique avec le bouton droit de la souris sur folios et on choisit:

- Ajouter un folio,
- La taille du folio souhaitée (ex : A4 portrait) puis OK.



• Afin de dessiner les GRAFCET sur le nouveau folio, on procède comme suit :

Dans l'espace de gauche, on choisit «*Palette*» ensuite «*Blocs GRAFCET*». Enfin, on sélectionne et glisse les éléments souhaités sur le folio.



✓ **Symboles :**

La liste des symboles donne la correspondance entre des noms « symboliques » et des noms de variables.

On commence par créer la table des symboles : pour cela on clique avec le bouton droit de la souris sur "*Symboles*" et on choisit "*Créer une table de symboles*". Ensuite, on clique avec le bouton droit de la souris sur "*Symboles*", et on choisit

"*Ajouter*" et on crée ainsi la table des symboles que l'on utilisera dans les GRAFCET.



✓ **IRIS :**

IRIS 2D permet de créer des pupitres, des applications de supervision et des applications de simulation de parties opératives 2D.

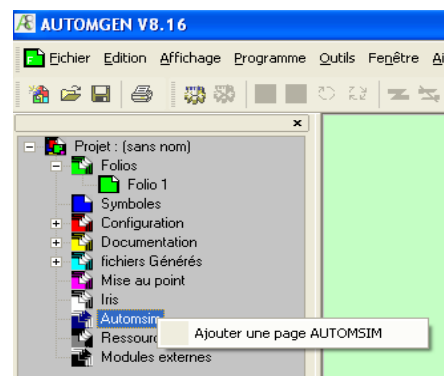
- Pour ajouter un pupitre ou un objet IRIS 2D, on sélectionne <Ajouter un objet IRIS 2D>, et on choisit « Pupitre ».



Une fois le pupitre 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.

✓ **Automatisme :**

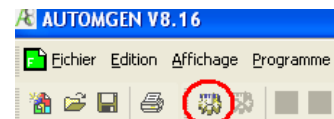
Pour simuler des schémas électriques, on crée une fenêtre « *Automatisme* » qui offre une bibliothèque élargie sur des composants électriques et pneumatiques, pouvant ainsi refléter le fonctionnement d'une manière la plus proche possible.



**V.2.3. Exécution de l'application :**

**V.2.3.1. Compilateur :**

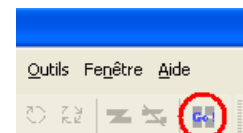
Le compilateur permet de localiser une erreur, après un double clic sur le message d'erreur, le logiciel renvoie sur l'erreur.



**V.2.3.2. Exécution :**


Le bouton « GO » représente la méthode la plus rapide pour observer le résultat de l'exécution d'une application. Il active les mécanismes suivants :

- Compilation de l'application si elle n'est pas à jour ;
- Passage de la cible en RUN ;
- Activation de la visualisation dynamique ;





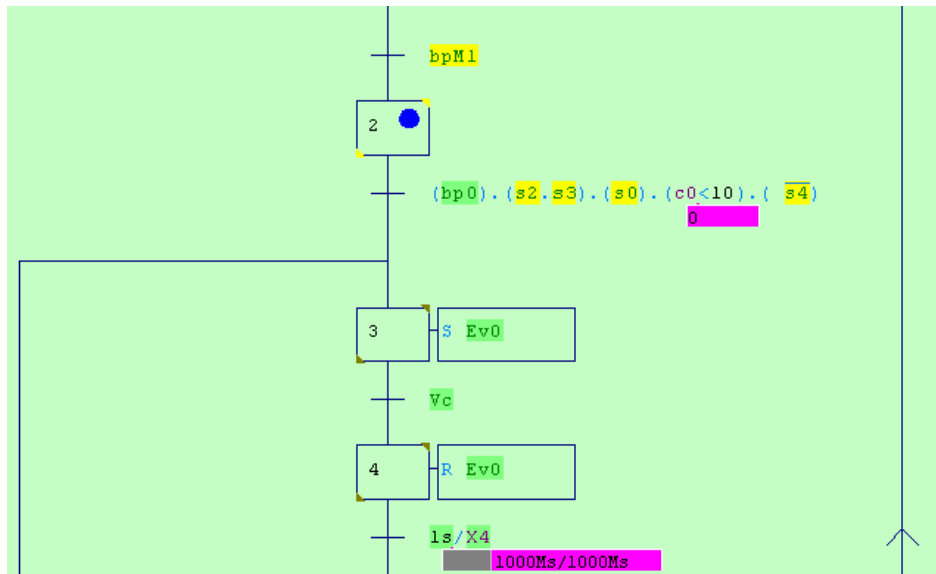
Remarque :

L'activation de la visualisation dynamique peut se faire par le bouton .

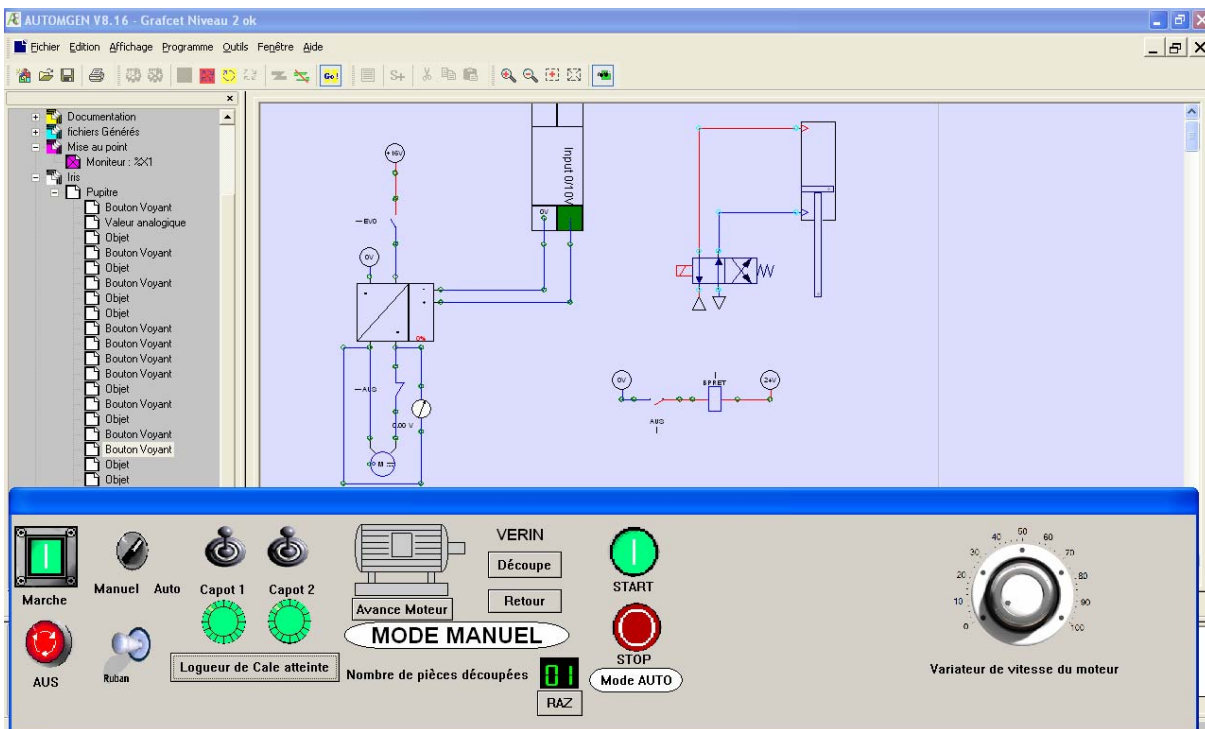
Au niveau de la visualisation en dynamique sur les folios :

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

V.2.3.3. Exemple de simulation du GRAFCET :



V.2.3.4. Exemple simulation de « Automatism » et « IRIS » :



Après avoir présenté le logiciel de modélisation et simulation du GRAFCET, on peut à présent concevoir le GRAFCET régissant le fonctionnement de la machine.

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

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

✓ Moteur :

S EV0 : Démarrage du Moteur.

R EV0 : Arrêt du moteur.

✓ Electrovanne :

S Y1 : Descente du vérin.

R Y1 : Remontée du vérin.

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

### 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 du 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 pièces découpées.

+ C0 : Incrémente le compteur.

R C0 : Remise à zéro du compteur.

**Conclusion :**

Dans ce chapitre nous avons présenté l'outil GRAFCET, qui nous a permis de décrire d'une manière plus explicite le fonctionnement de la machine et ainsi la succession des étapes formant le processus de fabrication des cales d'encoches.

Après avoir organisé la modélisation par GRAFCET, et avoir bien assimilé le système, nous avons pu envisager une transcription vers un programme dans une unité intelligente, qui n'est autre qu'un automate programmable. Et qu'on définira dans le chapitre suivant.

## I. Introduction :

Un automate programmable industriel (API) est essentiellement un dispositif électronique possédant l'architecture d'un ordinateur mais destiné à être utilisé en milieu industriel. Cette vocation induit des caractéristiques distinctives dont les principales sont :

- Une grande capacité de communication avec un environnement industriel.
- Une grande fiabilité (nécessaire par le caractère industriel de son utilisation).
- Des logiciels de programmation non standardisés (Chaque constructeur d'API développe ses propres langages de programmation).
- Outre ses capacités de communication avec un procédé industriel, un API possède également des possibilités de dialogue homme — machine.
- Dans la figure suivante, on présente la commande par un API dans son contexte industriel.

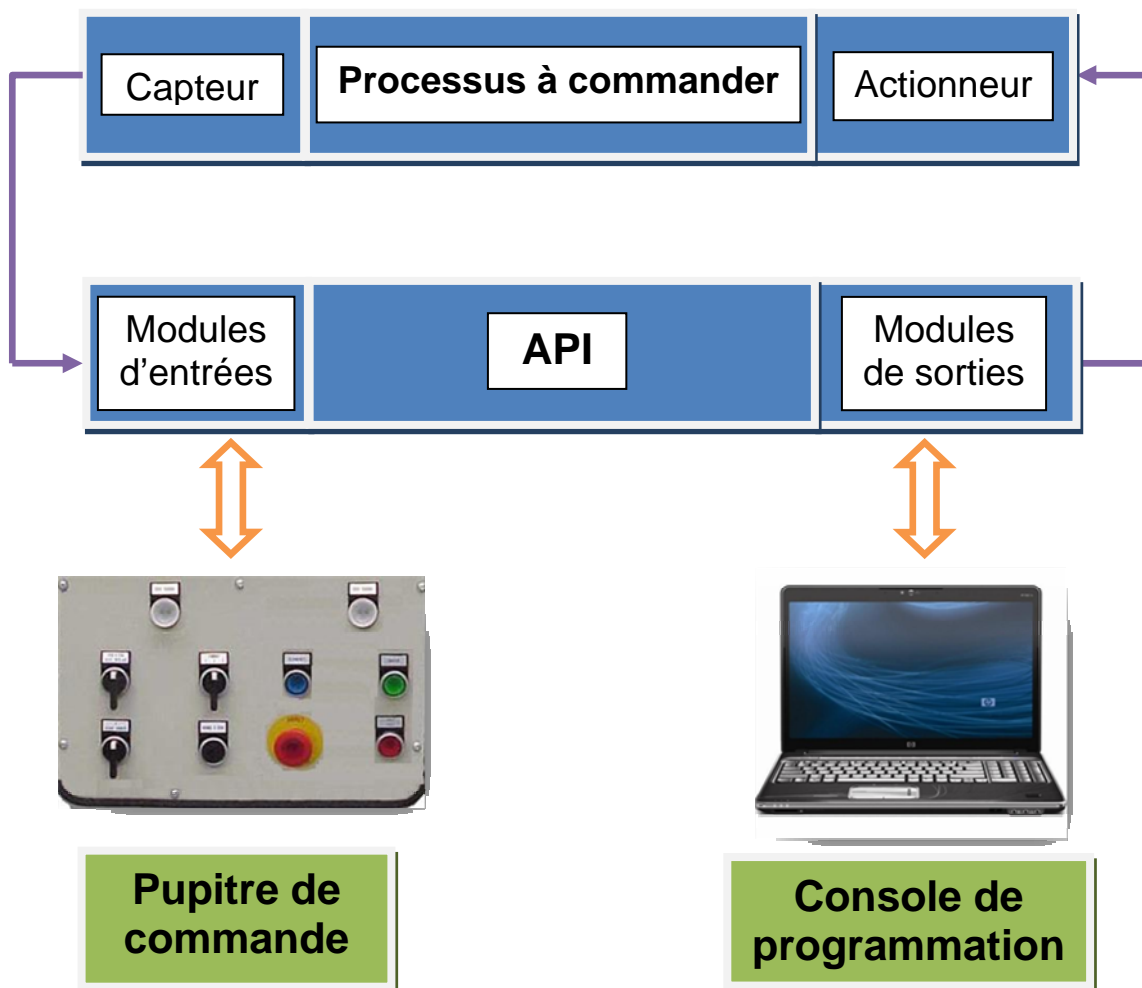


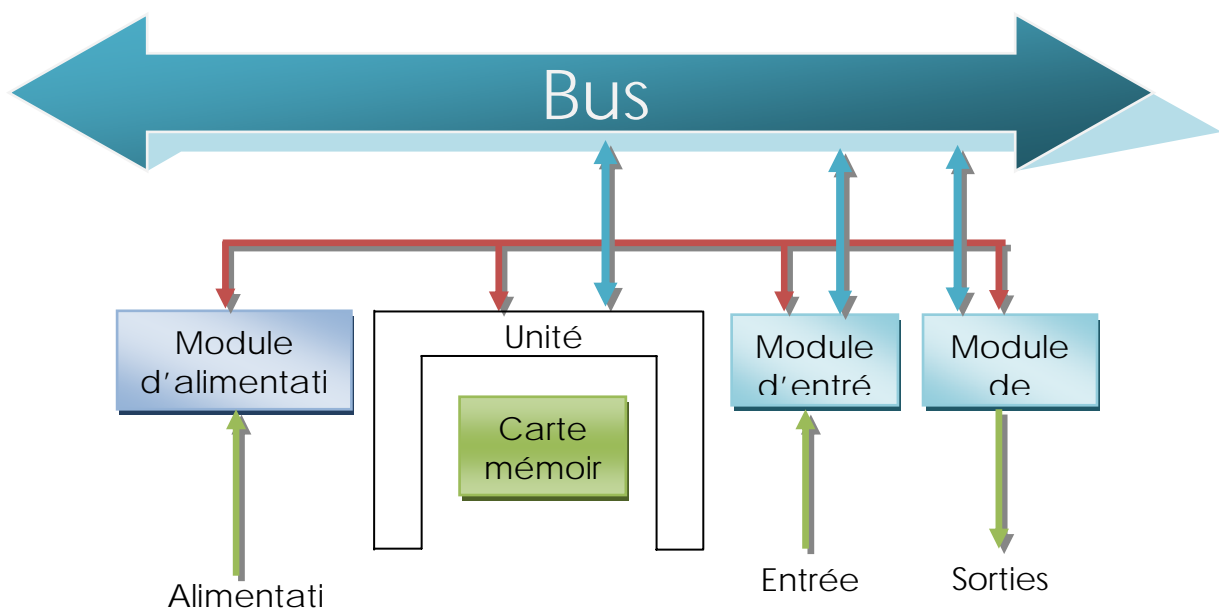
Fig.1 – Insertion d'API dans un procédé et dialogue H-M.

➤ **Définition d'un automate programmable industriel :**

C'est un appareil électronique qui comporte une mémoire programmable par un utilisateur à l'aide d'un langage adapté. Il commande, mesure et contrôle au moyen des modules d'entrées et de sorties des processus.

## II. Structure interne d'un automate programmable :

La structure d'un automate programmable (AP), ressemble à celle d'un micro-ordinateur comme montre la figure suivante :



**Fig.2 – Structure interne d'un système d'automate programmable.**

L'automate comporte principalement les parties suivantes :

- L'unité centrale ;
- Bloc d'alimentation ;
- Les modules d'entrées / sorties ;
- Des modules enfichables ajoutés en guise de périphérie. Ces parties sont reliées entre elles par un bus ;

## II.1. L'unité centrale (UC) :

C'est l'ensemble des dispositifs nécessaires au fonctionnement logique interne de l'API, elle comprend ce qui suit :

**1. Le processeur** : c'est la partie intelligente de l'UC, appelé l'unité de traitement (UT). Il est chargé d'assurer le contrôle de l'ensemble de la machine et d'effectuer le traitement demandé par l'instruction du programme. Il comprend différents registres :

- a. Registre interne ;
- b. Compteur ordinal (pointeur) ;
- c. Registre d'instruction (code opératoire) ;
- d. Registre d'adresse (adresse opérande) ;
- e. Registre accumulateur ;

**2. La mémoire** : elle est conçue en vue de stocker toutes les informations du système; elle dialogue d'une part avec le processeur et d'autre part avec les organes d'entrées / sorties.

## II.2. Bloc d'alimentation:

Il permet de fournir à l'automate l'énergie nécessaire à son fonctionnement; à partir d'une alimentation en 220 volts alternatifs. Ce bloc délivre les tensions dont l'automate a besoin : 24 ou 12 ou 5 volts.

## II.3. Les modules enfichables:

Le raccordement d'autres modules à l'automate est prévu pour l'obtention des fonctions supplémentaires ; ces modules sont :

1. Les cartes mémoires ;

2. Les coupleurs : ce sont des cartes électriques qui assurent la communication et l'échange d'information entre l'UC et :

- Les modules d'entrées / sorties par l'intermédiaire d'un bus interne.
- Les périphériques de l'automate (console, imprimante, micro ordinateur) par un bus externe.

3. Les cartes d'entrées / sorties ;

4. La console de programmation ;

## II.4. Modules d'entrées / sorties:

1. Les modules d'entrées tout ou rien (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 (Module 4, 8, 16 ou 32 entrées). A chaque entrée, correspond une voie qui traite le signal électrique pour élaborer une information binaire. Une diode électroluminescente « LED » située sur la carte donne l'état de chaque entrée.

2. Les modules de sorties tout ou rien (TOR) : un module de sortie permet à l'API d'agir sur les actionneurs. Il réalise la correspondance (état logique – signal électrique) et délivre l'information de l'automate vers le processus.

L'état de chaque sortie est visualisé par une diode électroluminescente (LED).

3. Des modules (d'entrées/sorties) analogique : auquel on n'aura pas à faire dans notre programme).

### III. Fonctions d'un automate programmable :

Les fonctions susceptibles d'être amènes par un automate programmable sont :

- La détection, grâce à des capteurs placés sur la machine ;
- Le dialogue d'exploitation ;
- La commande d'action, vers les actionneurs et pré-actionneurs ;
- Le dialogue de supervision ;

#### ➤ **Choix d'un automate programmable industriel :**

Les critères de choix d'un API sont :

- Le nombre et la nature des entrées / sorties ;
- La nature du traitement (temporisation, comptage...) ;
- Le dialogue (la console détermine le langage de programmation) ;
- Les moyens de sauvegarde du programme ;
- La fiabilité et robustesse ;
- L'immunité aux parasites ;
- Le service après vente et la durée de garantie ;
- Le rapport coût / qualité ;

🚦 Le choix de l'automate programmable S7.200 (CPU 226) de la firme SIEMENS, s'est fait selon différents critères dont les suivants :

- **Le choix de la firme SIEMENS**, est relié au fait que l'évolution automatique de l'Electro Industries s'est longtemps faite par des automates programmable de la firme SIEMENS de la première génération d'automate jusqu'au S7, et ceci grâce à leur réputation en terme de fiabilité et de robustesse.

- **Du type S7.200**, car il répond entièrement aux besoins et aux exigences de la programmation, du fait qu'il met à notre disposition l'utilisation des compteurs rapides qui est un élément principal dans notre sujet, et qui n'est pas disponible dans les autres automates programmables SIEMENS (S7.300,400,...).

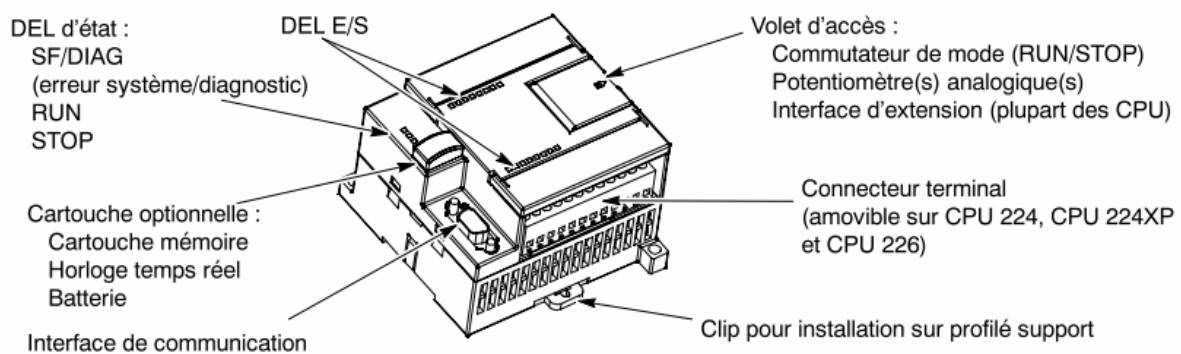


## IV. Présentation de l'automate S7 200 :

### IV.1. Introduction :

La famille S7-200 est constituée de micro-automates programmables pouvant commander une large gamme d'appareils afin de répondre au besoin de l'opérateur en matière d'automatisation.

Le S7-200 surveille les entrées et modifie les sorties conformément au programme utilisateur, qui peut contenir des opérations booléennes, des opérations de comptage, des opérations de temporisation, des opérations arithmétiques complexes et des opérations de communication unités intelligentes. Sa forme compacte, sa configuration souple et son important jeu d'opérations en font une solution idéale pour la commande d'applications très variées.



**Fig.3: Automate programmable S7-200.**

## IV.2. Principaux composants de l'API S7-200 :

### IV.2.1. CPU S7-200 :

Le module CPU S7-200 est un automate compact comprenant une unité centrale (CPU ou UC), une alimentation et des entrées / sorties discrètes.

- La CPU exécute le programme et sauvegarde les données par la commande du processus ou de la tâche d'automatisation.
- L'alimentation fournie de l'énergie électrique à l'appareil de base et à tout module extension connectée.
- Les entrées et les sorties sont les points de commande du système : les entrées surveillent les signaux des appareils sur site (tels que capteurs et commutateurs) et les sorties commandent pompes, moteurs et autres appareils dans le processus.
- L'interface de communication permet de connecter la CPU à une console de programmation ou à d'autres appareils. Certaines CPU S 7-200 disposent de deux interfaces de communication.

 Dans notre cas, la technologie de communication est le MPI.

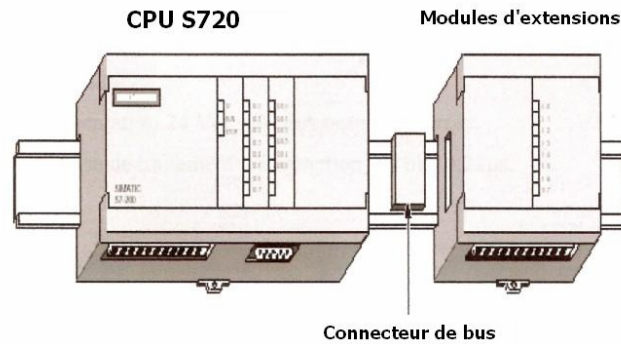
Le MPI (Multi point Interface) est une interface propriétaire des appareils SIMATIC S7. Elle est utilisée pour le raccordement des stations de programmation (PG ou PC), les pupitres opérateurs, ainsi que pour d'autres appareils appartenant à la famille Simatic. C'est cette technologie qui a inspiré le développement du protocole profibus.

L'interface MPI fonctionne avec une vitesse de transfert de 187,5kbaud. Les fabricants utilisant cette technologie MPI peuvent proposer toute une gamme de connexions vers un PC : Cartes MPI, cartes PCMCIA, adaptateurs USB ou Ethernet.

- Des témoins (DEL), donnent des informations visuelles sur l'état de fonctionnement de la CPU (Marche – RUN ou Arrêt – Stop), l'état en vigueur des E/S locales et la détection éventuelle d'une défaillance système.

#### IV.2.2. Modules d'extensions :

La CPU S7-200 comporte des entrées / sorties locales. Les modules d'extension permettent d'ajouter des entrées / sorties à l'appareil de base comme illustré la figure suivante. Un connecteur de bus fourni avec le module d'extension permet de connecter ce dernier à l'appareil de base.



**Fig.4 – Unité centrale avec un module d'extension.**

- ✚ La famille S7-200 comporte un large choix de CPU. Elle dispose ainsi de différentes fonctions permettant de concevoir une solution d'automatisation au meilleur prix possible parmi elles : la CPU 226.

## V. Choix de la CPU 226:

Le choix s'est porté sur la CPU 226 pour sa haute performance compacte, elle a 24 entrées et 16 sorties intégrées, extensible avec un maximum de 7 modules d'extension.

### ❖ Les caractéristiques techniques de la CPU 226 :

- 24 entrées (24 Vcc logique positive) ;
- 16 sorties (24 Vcc) ;
- Mémoire de programmation 16384 octets ;
- Mémoire de données 10240 octets ;
- Potentiomètres analogiques (02) ;
- Deux interfaces de communication (RS-485) ;
- Sauvegarde des données utilisateur (environ 5 jours par le biais d'un condensateur interne) ;
- Deux voies de comptage/décomptage pour compteurs rapides, de 6 à 30 kHz pour les compteurs rapides à une phase, et de 4 à 20 kHz pour ceux à deux phases ;
- Alimentation 24 Vdc 280 mA pour les entrées ;
- Taille de la mémoire image d'E/S TOR : 128 entrées, 128 sorties ;
- Vitesse de traitement d'instruction sur bits 0,22  $\mu$ s ;

**Conclusion :**

Ce chapitre nous a permis de présenter l'API que nous avons proposé (S7-200) et ses différentes caractéristiques. L'étape suivante, est la programmation qui se fera à l'aide du progiciel de programmation : STEP-7 MicroWin v4.0 qui constitue le lien entre l'utilisateur et l'API car ce dernier ne peut gérer ses fonctionnalités sans un programme approprié.

L'exploitation du logiciel ainsi que les programmes créés fera l'objet du chapitre suivant.

## **I. Logiciel de programmation STEP7-Micro/Win :**

### **I.1. Définition :**

STEP 7-Micro/WIN v4.0, permet d'associer un programme à toutes les informations nécessaires, de façon à communiquer avec un API et charger le programme dans cet API.

### **I.2. Organisation du programme de commande :**

Un programme de commande pour une CPU S7-200 comporte les types suivants d'unités d'organisation de programme (UOP) :

#### **I.2.1. Programme principal : PPAL**

Il comporte les opérations qui commandent l'application. Les opérations dans le programme principal sont exécutées séquentiellement, une fois par cycle de la CPU.

#### **I.2.2. Sous-programmes : SBR\_i**

Ces éléments du programme ne sont exécutés que lorsqu'ils sont appelés par le programme principal, par un programme d'interruption ou par un autre sous-programme.

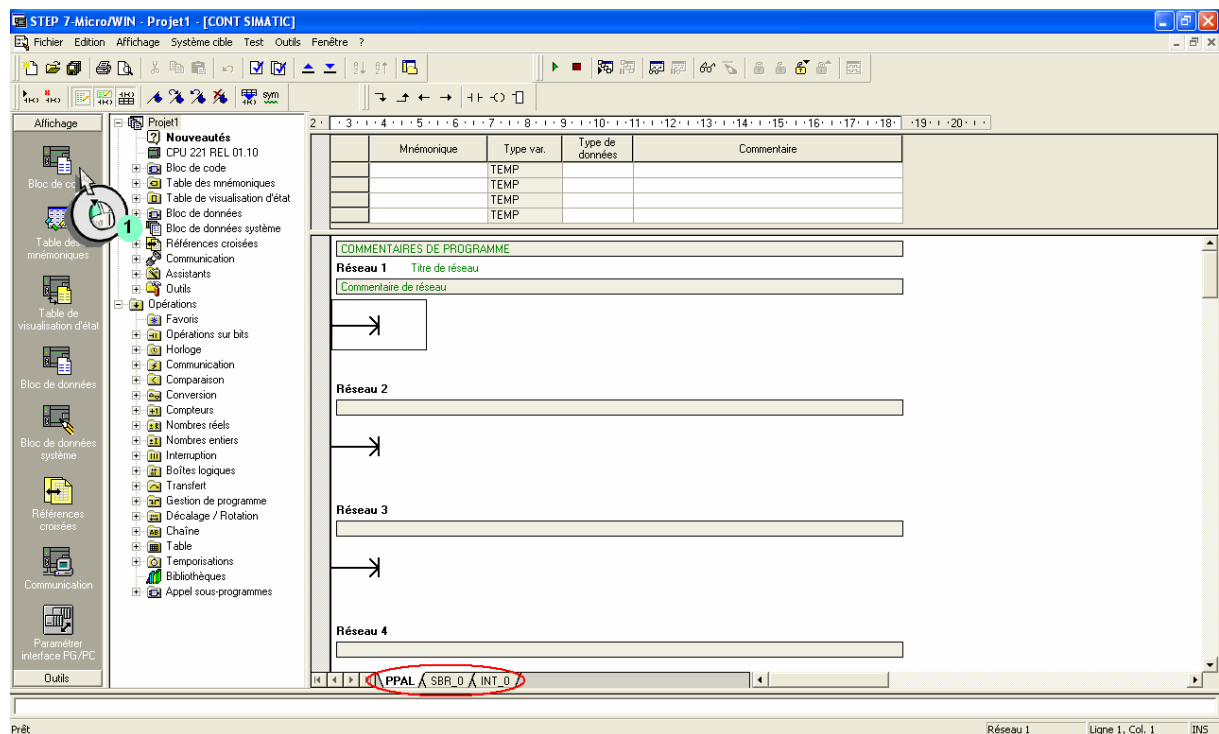
#### **I.2.3. Programmes d'interruption : INT\_i**

Un programme d'interruption est un ensemble d'opérations situées dans un bloc distinct et qui sont exécutées uniquement lorsqu'une interruption se produit.

**Remarque :**

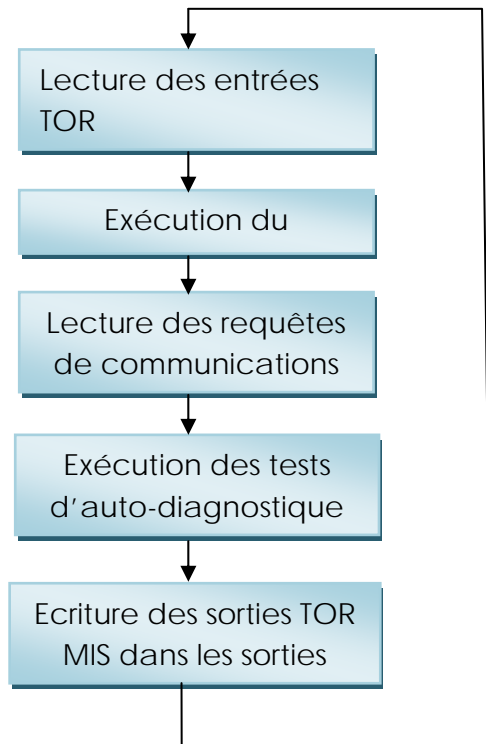
STEP 7-Micro/WIN v4.0 organise le programme de façon à fournir des onglets distincts pour chaque UOP dans la fenêtre de l'éditeur de programme. Le programme principal, PPAL, constitue toujours le premier onglet ; il est suivi des onglets des sous programmes et programmes d'interruption préalablement créés.

La figure suivante représente l'organisation des « PPAL, SBR\_i et INT\_i » sur le bloc de code.



## II. Fonctionnement du programme de commande :

Une fois le programme chargé dans l'API, et avoir fait passer l'API à l'état de fonctionnement "**Marche**" (**RUN**), l'unité centrale (CPU) de l'API S7-200 exécute les tâches dans l'ordre suivant :



- Lecture des entrées TOR : lecture de l'état de toutes les entrées connectées à l'automate programmable. Ces données sont rangées dans la zone de mémoire d'entrée ou mémoire image des entrées (MIE). Elle utilise ces entrées pour évaluer (exécuter) la logique du programme de commande.
- Exécution du programme : Pendant l'évaluation du programme, la CPU range les résultats de la logique de programme dans la zone de mémoire de sortie ou mémoire image des sorties.
- Traitement des requêtes de communication.
- Exécution des tests d'auto-diagnostic de la CPU.
- Ecriture dans les sorties TOR : A la fin du programme, la CPU écrit les données de la mémoire image des sorties (MIS) dans les sorties sur site.

## III. Accès aux données dans les zones de mémoire :

L'accès à la mémoire image des entrées (I) et celles des sorties (Q), mémoire des variables (V), mémentos (M), temporisations (T), compteurs (C), compteurs rapides (HC), accumulateurs (AC) et les mémentos spéciaux (SM) se fait par bits, octets, mots ou doubles mots.

- Bit : I [adresse d'octet].[adresse de bit], ex : I0.1, Q2.1, M1.0, ...
- Octet (B), mot (W) ou double mot (DW) : I [taille] [adresse d'octet de départ],  
Ex : IB4, VD100, QW0, SMD48, ...



### III.1. Mémoire image des entrées : I

Le S7-200 lit les entrées physiques au début de chaque cycle et écrit ces valeurs dans la mémoire image des entrées.

### III.2. Mémoire image des sorties : Q

A la fin du cycle, le S7-200 copie dans les sorties physiques les valeurs contenues dans la mémoire image des sorties.

### III.3. Mémoire des variables : V

La mémoire des variables (mémoire V) peut être utilisée pour sauvegarder des résultats intermédiaires d'opérations exécutées par la logique de commande dans un programme.

- ✚ Dans notre programme, les mémoires (V) ont été consacrées pour toutes les variables auxquelles on fera appel à partir de l'écran tactile.

### III.4. Mémentos : M

Les mémentos internes (mémoire M) « ou relais de commande » fournissent de l'espace mémoire pour l'état intermédiaire d'une opération ou d'autres informations de commande.

### III.5. Temporisations : T

Le S7-200 fournit des temporisations qui comptent des incréments.

- ✚ Dans notre cas, on a utilisé la temporisation T37 dont deux variables lui sont associées :
  - **la valeur en cours** : ce nombre entier signé de 16 bits contient la durée comptabilisée par la temporisation.
  - **le bit de temporisation** : ce bit (T37) est mis à 0 si la valeur en cours (Vc) est inférieure à la valeur prédéfinie (Vp) ; sinon il est mis à 1.

### III.6. Compteur : C

Le S7-200 fournit trois types de compteurs qui comptent chaque transition du niveau bas au niveau haut aux entrées de comptage : les compteurs qui incrémentent, ceux qui décrémentent et ceux pouvant à la fois incrémenter et décrémenter.

✚ Dans notre cas, on aura à utiliser un compteur incrémental dont deux variables lui sont associées :

- **la valeur en cours** : ce nombre entier signé de 16 bits contient le total cumulé.
- **le bit de compteur** : ce bit (C0) est mis à 0 si la valeur en cours (Vc) est inférieure à la valeur prédéfinie (Vp) ; sinon il est mis à 1.

### III.7. Accumulateurs : AC

Les accumulateurs sont des éléments en lecture/écriture pouvant être utilisés comme mémoire.

Par exemple pour sauvegarder des valeurs intermédiaires utilisées dans un calcul. Le S7-200 dispose de quatre accumulateurs de 32 bits : AC0, AC1, AC2 et AC3.

### III.8. Mémentos spéciaux : SM

Les mémentos spéciaux permettent l'échange d'informations entre la CPU et le programme. Et c'est possible de faire appel à ces bits pour sélectionner et commander certaines fonctions spéciales de la CPU S7-200, parmi eux on citera :

- SM 0.0 : ce bit est toujours mis à 1 dès que l'automate est allumé.
- SM 0.1 : Ce bit est mis à 1 au premier cycle. Il sert, entre autres, à l'appel d'un sous-programme d'initialisation.
- SMD 38 et SMD 42 : ces doubles mots sont consacrés aux compteurs rapides.

### III.9. Compteurs rapides : HC

Les compteurs rapides comptent des événements très rapides indépendamment du cycle de la CPU.

## IV. Fonctionnement des compteurs rapides (HC) :

Ils servent typiquement d'entraînement pour dispositifs de comptage où un arbre en rotation (du moteur à courant continu) à vitesse constante est muni d'un encodeur rotatif. L'encodeur fournit un nombre défini de valeurs de comptage par tour (des impulsions). Les horloges (A/B) de l'encodeur constituent les entrées du compteur rapide (I0.0 et I0.1).

Une valeur prédéfinie est chargée dans le compteur rapide et la sortie désirée est activée pour la durée où la valeur de comptage en cours est inférieure à la valeur prédéfinie en vigueur. Le compteur est défini de telle façon qu'une interruption est générée si la valeur en cours est égale à la valeur prédéfinie.

Ils ont une valeur de comptage (ou valeur en cours) entière signée de 32 bits. Pour accéder à la valeur de comptage d'un compteur rapide, on indique l'adresse du compteur rapide, comprenant l'identificateur de zone HC et le numéro du compteur (HC1, pour le compteur numéro 1 par exemple).

Le S7-200 (CPU 226) comporte six compteurs rapides, et 12 modes de fonctionnement.

Le tableau suivant montre les entrées utilisées pour les fonctions d'horloge, de gestion du sens de comptage, de mise à zéro et de démarrage associées aux compteurs rapides.

Mode	Description	Entrées			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I1.0	I1.1
	HSC2	I1.2	I1.3	I1.4	I1.5
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	Compteur monophasé avec gestion interne du sens de comptage	Horloge			
1		Horloge		Mise à 0	
2		Horloge		Mise à 0	Démarrage
3	Compteur monophasé avec gestion externe du sens de comptage	Horloge	Sens de comptage		
4		Horloge	Sens de comptage	Mise à 0	
5		Horloge	Sens de comptage	Mise à 0	Démarrage
6	Compteur biphasé avec deux entrées d'horloge	Horloge, incrémentation	Horloge, décrémentation		
7		Horloge, incrémentation	Horloge, décrémentation	Mise à 0	
8		Horloge, incrémentation	Horloge, décrémentation	Mise à 0	Démarrage
9	Compteur en quadrature de phase A/B	Horloge phase A	Horloge phase B		
10		Horloge phase A	Horloge phase B	Mise à 0	
11		Horloge phase A	Horloge phase B	Mise à 0	Démarrage
12	Seuls HSC0 et HSC3 acceptent le mode 12. HSC0 compte le nombre d'impulsions sortant de Q0.0. HSC3 compte le nombre d'impulsions sortant de Q0.1.				

**Tableau 1 – Entrées pour les compteurs rapides**

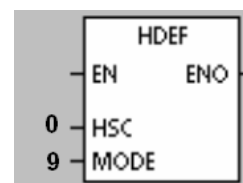
- ✚ Le compteur rapide utilisé dans notre programme est le HC0 fonctionnant en mode 9.
  - ✚ L'introduction des compteurs rapides, nous a permis de supprimer un bon nombre de cartes électroniques, celles qui servent à faire la Conversion (impulsions  $\Leftrightarrow$  millimètres), et celles responsables du comptage à partir de l'encodeur incrémental.
- De ce fait, le comptage se fera désormais par le compteur rapide, et la conversion par un programme dans PPAL consacré à cette opération.

## IV.1. Programmation d'un compteur rapide :

Les étapes de configuration d'un compteur rapide sont les suivantes :

### IV.1.1. Définition du compteur et de son mode :

L'opération « Définir mode pour compteur rapide » (HDEF) sélectionne le mode de fonctionnement (MODE) d'un compteur rapide spécifique (HSCx). La sélection du MODE définit les fonctions d'horloge, de sens, de démarrage et de remise à zéro du compteur rapide.



### IV.1.2. Définition de l'octet de commande :

HSC0	Description (utilisé uniquement à l'exécution de HDEF)
SM37.0	Bit de commande du niveau d'activité pour la mise à zéro <sup>1</sup> : 0 = niveau haut                      1 = niveau bas
---	Bit de commande du niveau d'activité pour le démarrage <sup>1</sup> : 0 = niveau haut                      1 = niveau bas
SM37.2	Sélection de la vitesse de comptage pour compteurs en quadrature de phase : 0 = fréquence quadruple,          1 = fréquence simple
SM37.3	Bit de commande du sens de comptage : 0 = décrémentation                  1 = incrémementation
SM37.4	Ecrire sens de comptage dans le compteur rapide : 0 = pas de mise à jour 1 = mise à jour du sens
SM37.5	Ecrire nouvelle valeur prédéfinie dans le compteur rapide : 0 = pas de mise à jour 1 = mise à jour de la valeur prédéfinie
SM37.6	Ecrire nouvelle valeur en cours dans le compteur rapide : 0 = pas de mise à jour 1 = mise à jour de la valeur en cours
SM37.7	Valider le compteur rapide : 0 = inhiber le compteur rapide 1 = valider le compteur rapide

**Tableau 2 – Octet de commande**

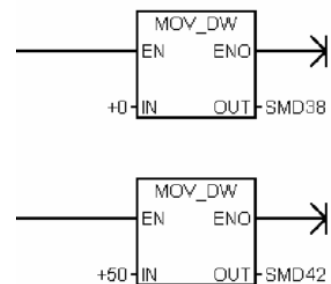
L'octet de commande qui définit le compteur rapide utilisé est :

SM 37.7	SM 37.6	SM 37.5	SM 37.4	SM 37.3	SM 37.2	SM 37.1	SM 37.0
1	1	1	1	1	0	0	0
<b>F</b>				<b>8</b>			

L'octet de commande est donc : 16#F8 (en hexadécimal).

**IV. 1.3. Définition de la valeur en cours et prédéfinie :**

Chaque compteur rapide a une valeur en cours (VC) et une valeur prédéfinie (VP). Ce sont toutes deux des nombres entiers de 32 bits signés. Le chargement de ces valeurs dans le compteur rapide, se fait après la configuration de l'octet de commande, et les mémentos spéciaux qui contiennent ces valeurs.



Le Tableau (3) présente les mémentos spéciaux pour les valeurs en cours et prédéfinies.

Valeur à charger	HSC0
valeur en cours	SMD38
valeur prédéfinie	SMD42

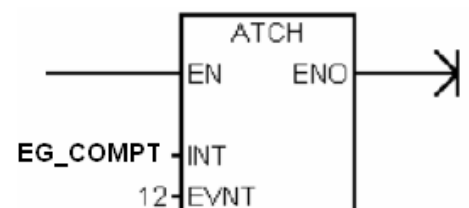
Compteur	HSC0
Valeur en cours	HC0

**Tableau 3 – Mémentos spéciaux : Vc, Vp**

**Tableau 4 – Lecture de Vc**

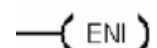
**IV.1.4. Affectation et validation du programme d'interruption :**

La commande ATCH, associe l'événement 12 (propre au compteur HC0) au programme d'interruption EG\_COMPT (préalablement programmé) lorsque la valeur en cours est égale à la valeur prédéfinie (Vc=Vp).



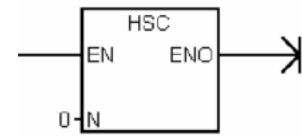
**IV.1.5. Validation du programme d'interruption :**

L'opération « Valider tous les événements d'interruption » (ENI), valide le traitement de tous les événements d'interruption objets d'une association.



#### IV.1.6. Activation du compteur rapide :

L'opération « *Activer compteur rapide* » (HSC), configure et commande le compteur rapide, selon l'état des mementos spéciaux pour ce dernier. Le paramètre N précise le numéro du compteur rapide.



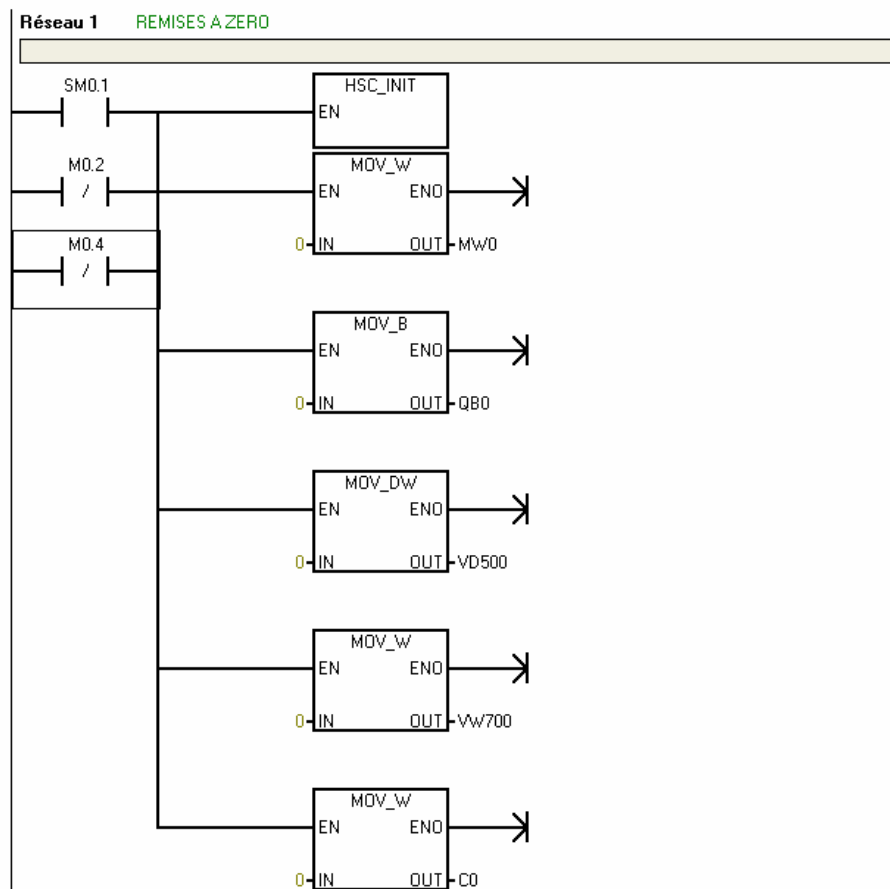
#### IV.1.7. Programme d'interruption (EG\_COMPT) :

Les interruptions du compteur rapide permettent de réagir à des situations telles que *valeur en cours égale à valeur prédéfinie* ( $V_c=V_p$ ).

Les interruptions sont gérées par des événements. Avant d'appeler un programme d'interruption, il faut associer l'événement d'interruption à la partie de programme à exécuter à l'apparition de cet événement ( $V_c=V_p$ ). Pour ce faire, on fait appel à l'opération ATCH en précisant le numéro d'événement d'interruption (événement 12) et le nom du programme d'interruption correspondant (**EG\_COMPT**).

## IV.2. Programmation de la commande de la machine :

### IV.2.1. Programme Principale : PPAL



✚ L'objectif de ce réseau est la réinitialisation.

- La 1<sup>ère</sup> ligne du réseau : appel le sous-programme de réinitialisation du compteur rapide.
- La 2<sup>ème</sup> ligne du réseau : remise à zéro de tous les mementos utilisés (de M0.0 jusqu'à M1.7).
- La 3<sup>ème</sup> ligne du réseau : remise à zéro des sorties.
- La 4<sup>ème</sup> ligne du réseau : remise à zéro de la valeur en cours en millimètre (celle visualisée sur le panel tactile).
- La 5<sup>ème</sup> ligne du réseau : remise à zéro du nombre de pièces découpées.
- La 6<sup>ème</sup> ligne du réseau : remise à zéro du compteur de nombre de pièces découpées.



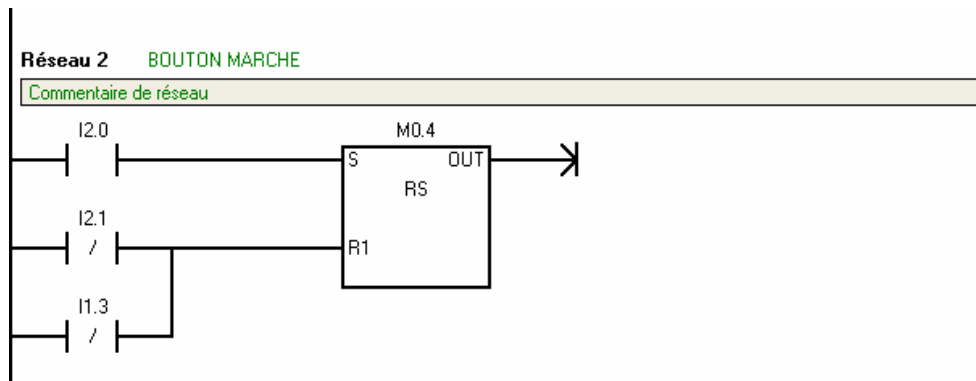
Et ceci est déclenché lors du :

- Premier cycle de l'automate.
- L'action sur le bouton Stop cycle de la machine.
- L'action sur le bouton Arrêt ou Arrêt d'urgence.

SM0.1 : mis à 1 lors du premier cycle de l'automate.

M0.2 : maintien du bouton Start cycle automatique.

M0.4 : maintien du bouton Marche.



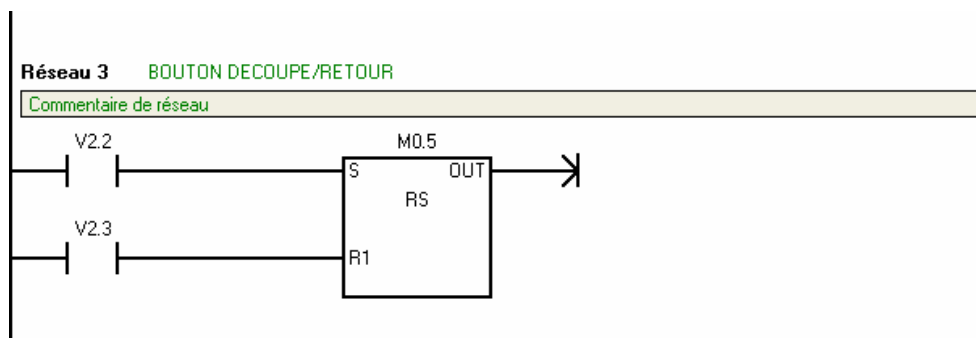
L'objectif essentiel de ce réseau est le verrouillage des boutons, et ceci en utilisant une bascule RS.

De cette manière, une fois le bouton marche actionné, l'information sera mémorisée dans le memento (M0.4) et ne sera remise à zéro que lorsque l'un des boutons arrêt ou arrêt d'urgence sera actionné sur l'entrée R de la bascule.

I2.0 : Bouton marche.

I2.1 : Bouton arrêt

I1.3 : Bouton arrêt d'urgence.

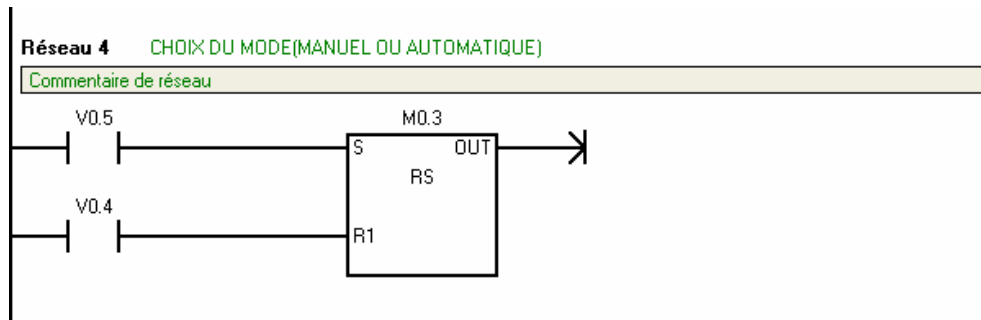


L'objectif est le verrouillage des boutons et ceci en utilisant une bascule RS.

De ce fait, l'action sur le bouton de découpe (V2.2) est maintenue dans le memento (M0.5). Il n'est remis à zéro qu'après action sur le bouton de retour vérin (V2.3) sur l'entrée R de la bascule.

V2.2 : Bouton découpe.

V2.3 : Bouton retour (du vérin).

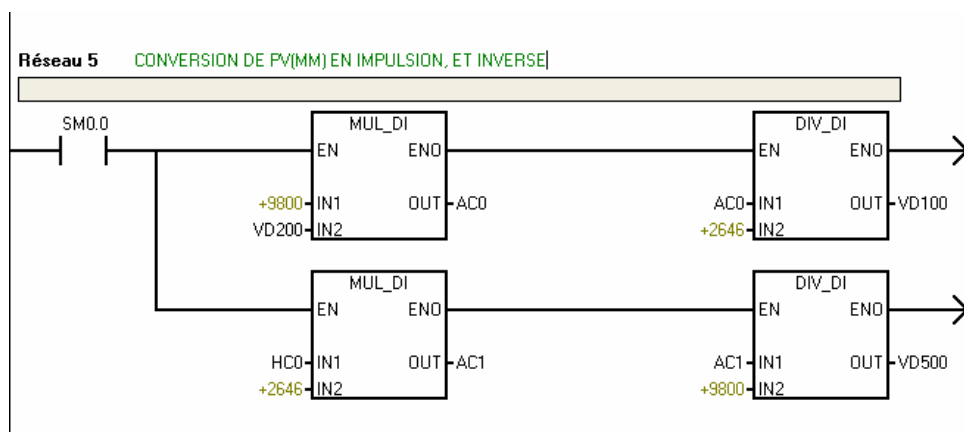


✚ L'objectif est le verrouillage des boutons en utilisant une bascule RS.

De ce fait, l'action du bouton mode automatique (V0.5), est maintenu dans le memento (M0.3). Il n'est remis à zéro qu'après action sur le bouton du mode manuel (V0.4) sur l'entrée R de la bascule.

V0.5 : Bouton mode automatique.

V0.4 : Bouton mode manuel.



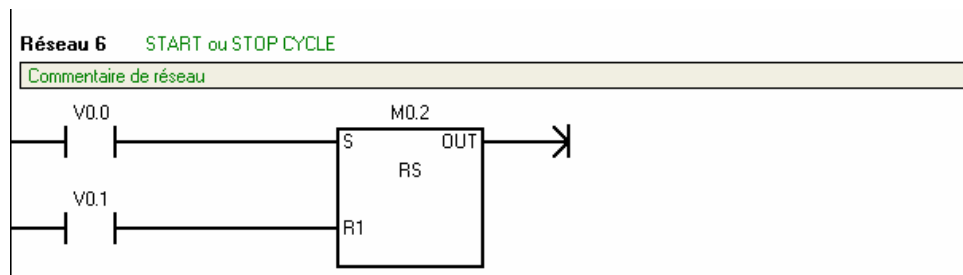
✚ L'objectif est de faire la conversion « millimètre  $\Leftrightarrow$  impulsion ». Après une étude expérimentale, les résultats suivants ont été donnés :

Pour un nombre d'impulsions égale à 9800, lui correspond une longueur du ruban égale à 2646 millimètres.

Nous avons utilisées ces résultats pour faire une conversion par la règle de trois, et ceci en utilisant des opérations arithmétiques telles que la multiplication et la division.

- La 1<sup>ère</sup> ligne du réseau : conversion de la *valeur de consigne* en millimètres (donnée à partir du panel tactile), en impulsions.

- La 2<sup>ème</sup> ligne du réseau : conversion de la *valeur en cours* (HC0), correspondante au déplacement du ruban fournie par le compteur rapide (impulsions), en millimètre.

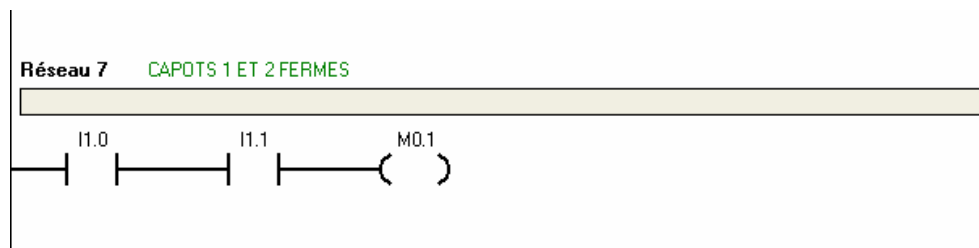


✚ L'objectif est le verrouillage des boutons en utilisant une bascule RS.

De ce fait, l'action sur le bouton démarrage cycle (V0.0), est maintenue dans le memento (M0.2). Il n'est remis à zéro qu'après action sur le bouton du mode manuel (V0.1) sur l'entrée R de la bascule.

V0.0 : Bouton mode automatique.

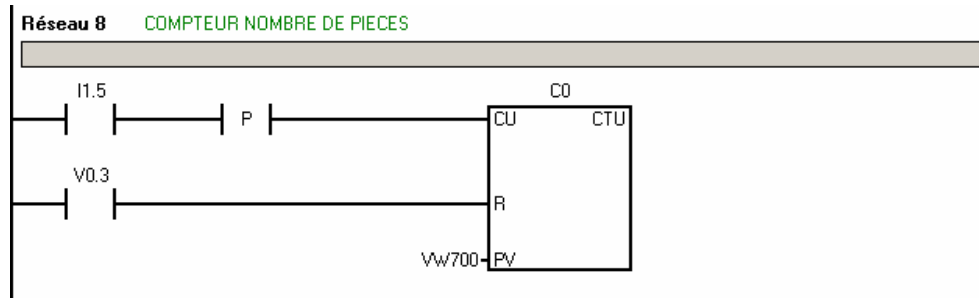
V0.1 : Bouton mode manuel.



✚ L'objectif est de regrouper les deux conditions sur les capots de sécurité des unités de freinage et de mise en forme (information venant des capteurs magnétiques de présences dans les deux unités), en un seul memento (M0.1).

I1.0 : capot de l'unité de freinage.

I1.1 : capot de l'unité de mise en forme.

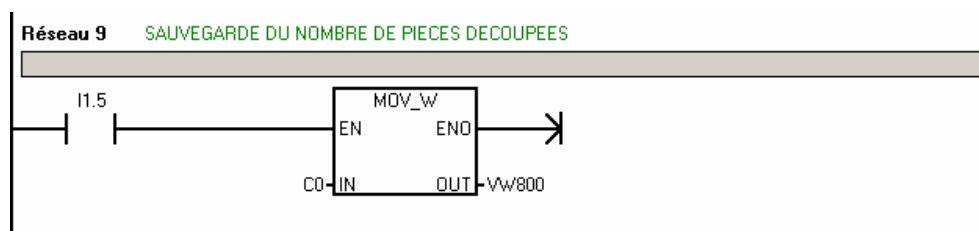


L'objectif de ce réseau, est de créer une consigne de nombre de pièces à découper, qui sera fournie à partir du panel tactile. Et ceci par un compteur incrémental (C0) de nombre de pièces qui est incrémenté à chaque fois que le vérin atteint la position basse (à chaque front montant sur le contact du capteur de position basse). Le compteur est remis à zéro par V0.3 (bouton).

I1.5 : contact du capteur magnétique de position basse du vérin.

I P I: front montant de I1.5.

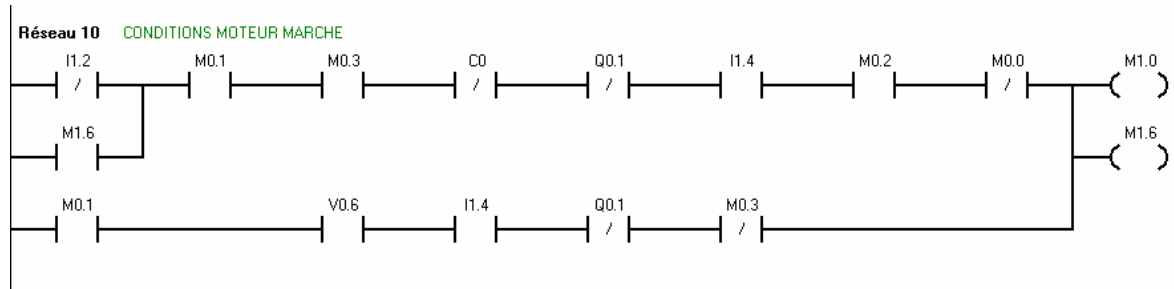
V0.3 : bouton de remise à zéro sur l'écran tactile.



L'objectif de ce réseau est de visualiser sur le panel le nombre de pièces cumulées ou découpées, en sauvegardant la valeur en cours du compteur incrémental (C0) dans la variable VW800.

C0 : valeur de comptage en cours.

I1.5 : capteur de position basse du vérin.



- Le rôle de ce réseau est de réunir les conditions nécessaires à la mise en marche du moteur en un memento (M1.0).

- La 1<sup>ère</sup> ligne du réseau, comporte les conditions de mise en marche du moteur en mode Automatique :

I1.2 : présence de ruban.

M0.1 : capots 1 et 2 fermés.

M0.3 : maintien du bouton *mode automatique*.

C0 : bit à zéro quand le nombre de pièces découpées est inférieur à la consigne.

Q0.1 : sortie du vérin (découpage), utilisé comme verrouillage.

I1.4 : capteur de position haute du vérin.

M0.2 : maintien du bouton *Start cycle automatique*.

M0.0 : impulsion lorsque  $V_c = V_p$ .

- La 2<sup>ème</sup> ligne du réseau, comporte les conditions de mise en marche du moteur en mode manuel :

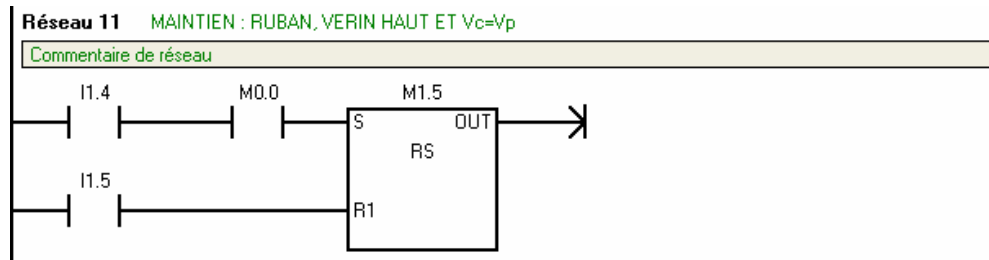
M0.1 : capots 1 et 2 fermés.

V0.6 : bouton moteur en marche manuel sur l'écran tactile.

I1.4 : capteur de position haute du vérin.

Q0.1 : sortie du vérin (découpage), utilisé comme verrouillage.

M0.3 : maintien du bouton *mode manuel*.

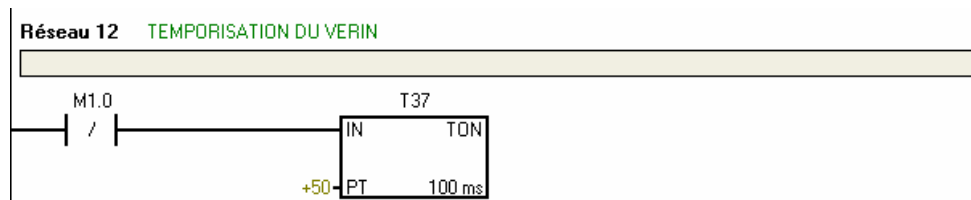


L'objectif de ce réseau est de regrouper les deux conditions (niveau haut du vérin, l'impulsion de Vc=Vp). Il suffit que le vérin atteigne la position basse pour que le memento M1.5 soit remis à zéro.

I1.4 : contact du capteur magnétique de position haute du vérin.

I1.5 : contact du capteur magnétique de position basse du vérin.

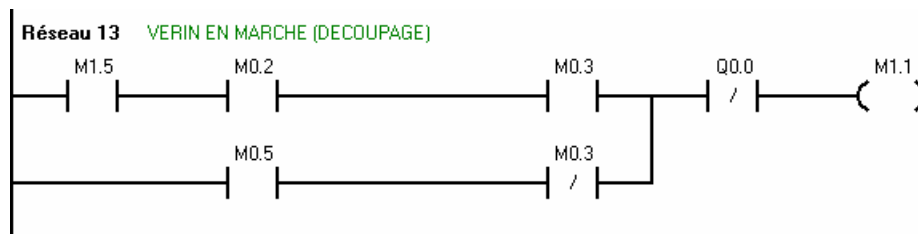
M0.0 : impulsion lorsque Vc=Vp.



Ce réseau a pour rôle de temporiser la descente du vérin d'une seconde.

M1.0 : memento donnant l'état moteur.

T 37 : temporisation d'une seconde après l'arrêt du moteur.



L'objectif de ce réseau est de regrouper les conditions nécessaires à la mise en marche du vérin en un memento (M1.1).

- La 1<sup>ère</sup> ligne du réseau, comporte les conditions nécessaires pour le mode automatique dont ces dernières :

M1.5 : maintien du niveau haut du vérin et l'impulsion de Vc=Vp.

M0.2 : maintien du bouton *Start cycle automatique*.

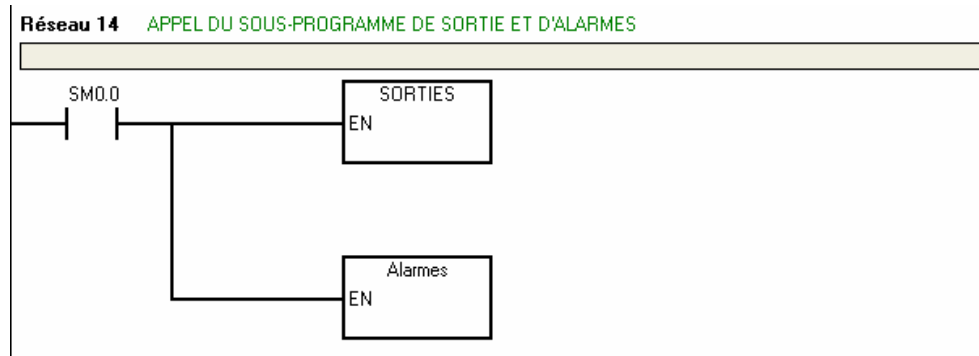
M0.3 : maintien du bouton *mode automatique*.

Q0.0 : sortie moteur en marche, utilisé comme verrouillage.

- La 2<sup>ème</sup> ligne du réseau, comporte les conditions nécessaires pour le mode manuel :

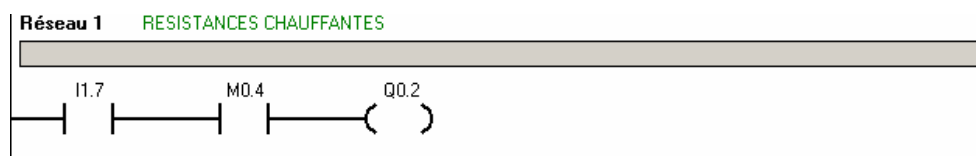
M0.5 : maintien du bouton de *découpage manuel*.

M0.3 : maintien du bouton *mode automatique*.



Les sous-programmes des *sorties* et *d'alarme* sont constamment appelés (SM0.0 est toujours à 1).

#### IV.2.2. Sous-programme des SORTIES :

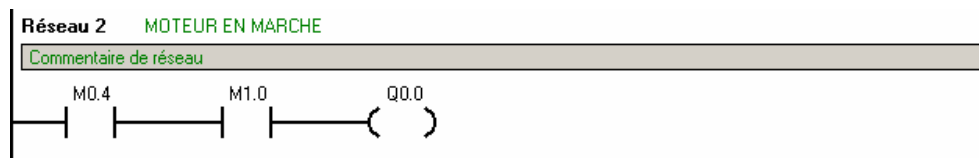


L'objectif de ce réseau est d'actionner le chauffage (par résistances chauffantes).

I1.7 : contact du thermostat.

M0.4 : maintien du bouton *Marche*.

Q0.2 : bobine du contacteur des résistances chauffantes.



Ce réseau a pour rôle de débloquent le variateur de vitesse et ainsi mettre le moteur en marche, quand les conditions nécessaires sont réunies.

M0.4 : maintien du bouton *Marche*.

M1.0 : condition de mise en marche du moteur.

Q0.0 : sortie vers l'entrée de déblocage du variateur de vitesse.



✚ Ce réseau a pour rôle d'exciter la bobine de l'électrovanne et entraîner la descente du vérin, quand les conditions de découpage sont réunies.

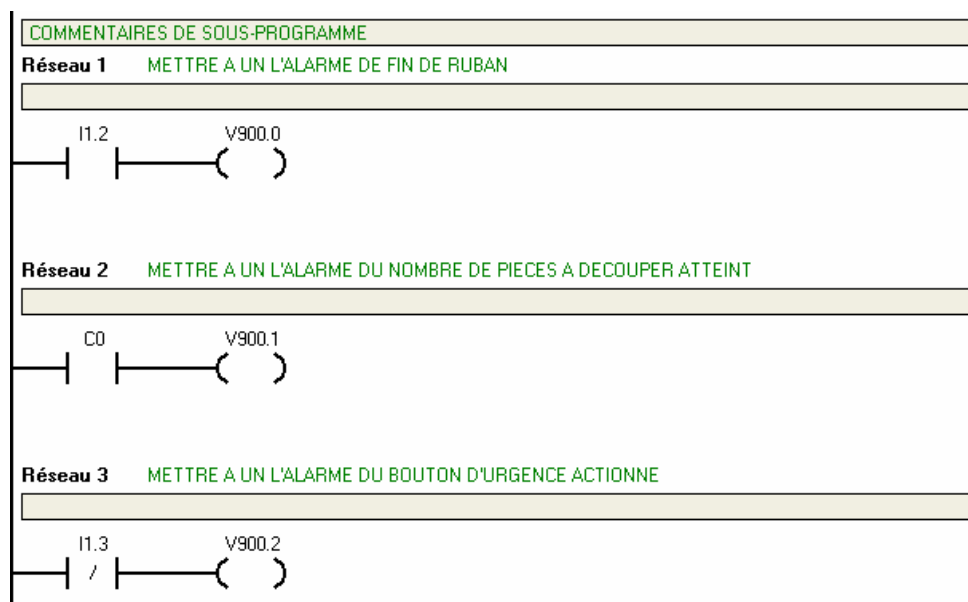
M0.4 : maintien du bouton *Marche*.

T37 : temporisation d'une seconde après l'arrêt du moteur.

M1.1 : conditions nécessaires pour le découpage.

Q0.1 : sortie vers la bobine de l'électrovanne.

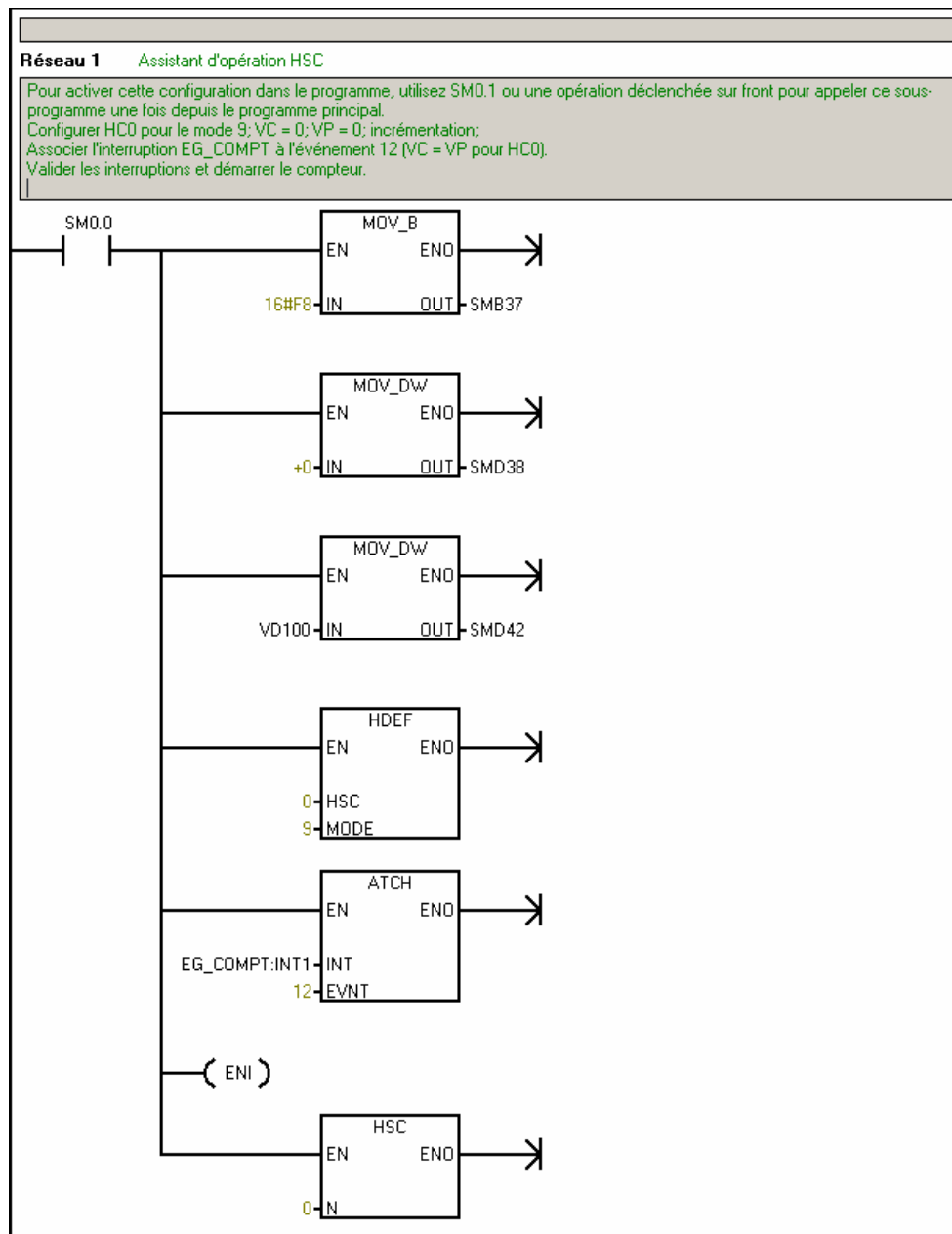
### IV.2.3. Sous-programme d'Alarmes :



✚ Réseaux consacrés aux alarmes visualisées sur l'écran tactile.



## IV.2.4. Programme d'initialisation du compteur rapide :

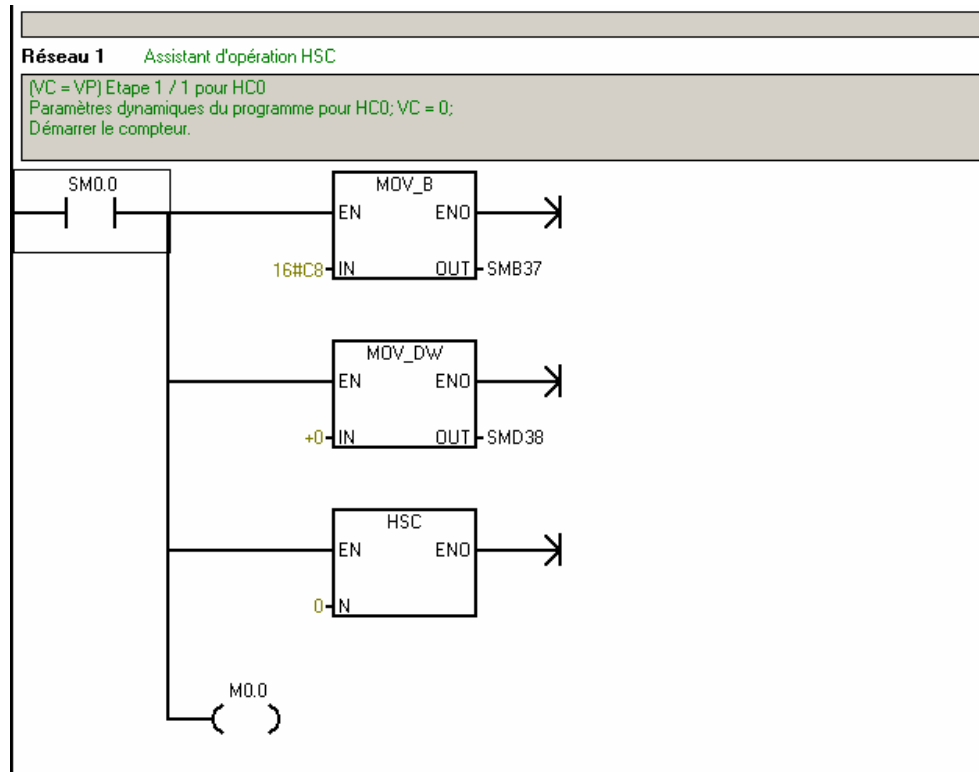


🔧 Programme d'initialisation du compteur rapide. Il a pour rôle :

- La 1<sup>ère</sup> ligne du réseau : configuration de l'octet de commande.
- La 2<sup>ème</sup> ligne du réseau : remise à zéro de la *valeur en cours*.
- La 3<sup>ème</sup> ligne du réseau : configuration de la *valeur de consigne*.
- La 4<sup>ème</sup> ligne du réseau : définition du compteur rapide et de son *mode de fonctionnement*.
- La 5<sup>ème</sup> ligne du réseau : association de l'événement d'interruption (12) au programme d'interruption (EG\_COMPT).

- La 6<sup>ème</sup> ligne du réseau : validation du programme d'interruption.
- La 7<sup>ème</sup> ligne du réseau : activation du compteur rapide HSC0.

#### IV.2.5. Programme d'interruption : EG\_COMPT



📌 Le programme est exécuté lorsque la valeur de consigne est atteinte ( $V_c=V_p$ ).

- La 1<sup>ère</sup> ligne du réseau : écriture de la nouvelle *valeur en cours* dans le compteur rapide.
- La 2<sup>ème</sup> ligne du réseau : remise à zéro de la *valeur en cours*.
- La 3<sup>ème</sup> ligne du réseau : activation du compteur rapide HSC0.
- La 4<sup>ème</sup> ligne du réseau : impulsion générée lorsque  $V_c=V_p$  dans le memento M0.0.

**Conclusion :**

Après avoir mis en œuvre le programme régissant le fonctionnement de la machine, tout en apportant des améliorations à sa commande à savoir l'ajout du compteur « Nombre de pièces à découper » ainsi que les « Alarmes », nous avons cherché après un meilleur contrôle de la commande en introduisant de nouvelles consignes par un panel tactile. Ce qui fera l'objet du dernier chapitre.

## **I. WinCC Flexible :**

### **I.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 :

#### **I.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).

#### **I.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.

**✓ 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, bargraphes, 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.

## II. Choix du panneau tactile TP 177A :

Le choix s'est porté sur le panel tactile TP 177A, pour ces avantages qu'il présente dont :

- 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'autres applications.
- Les temps de configuration et de mise en service, qui se singularisent par une courte durée de configuration dans le WinCC Flexible.

### ➤ **Panneau Tactile TP 177A :**

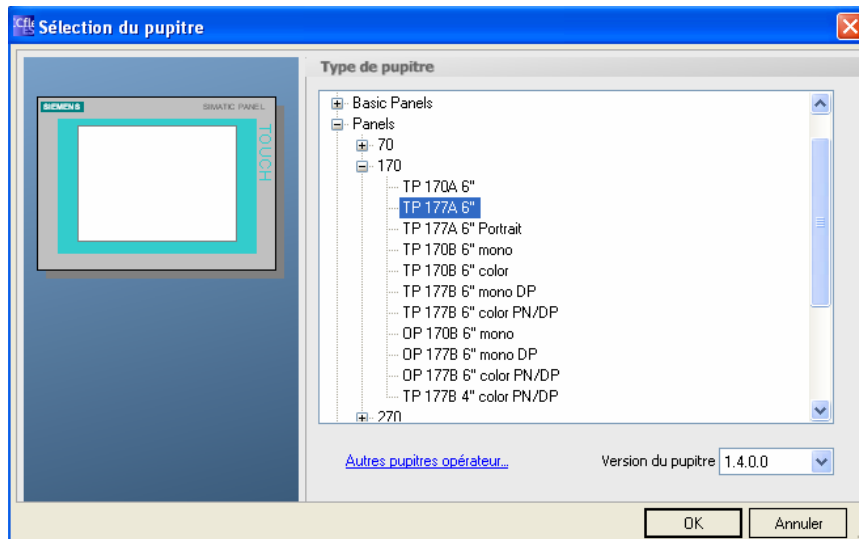
Les afficheurs TP177A micro sont conçus pour fonctionner avec le S7-200. Ils fournissent des fonctions de commande et de surveillance pour des machines et des installations.

### III. Etapes de configuration :

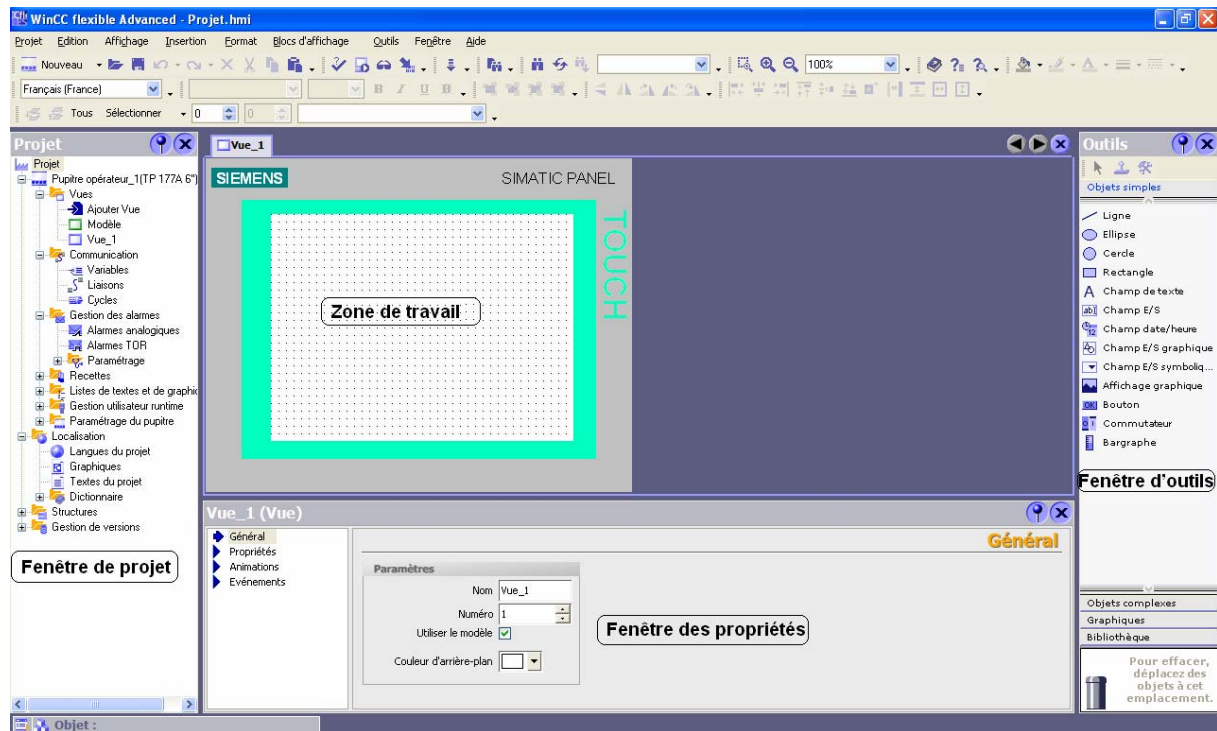
#### III.1. Créer un projet :

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 dans la famille des pupitres 170 le pupitre 177A'', comme suit :



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



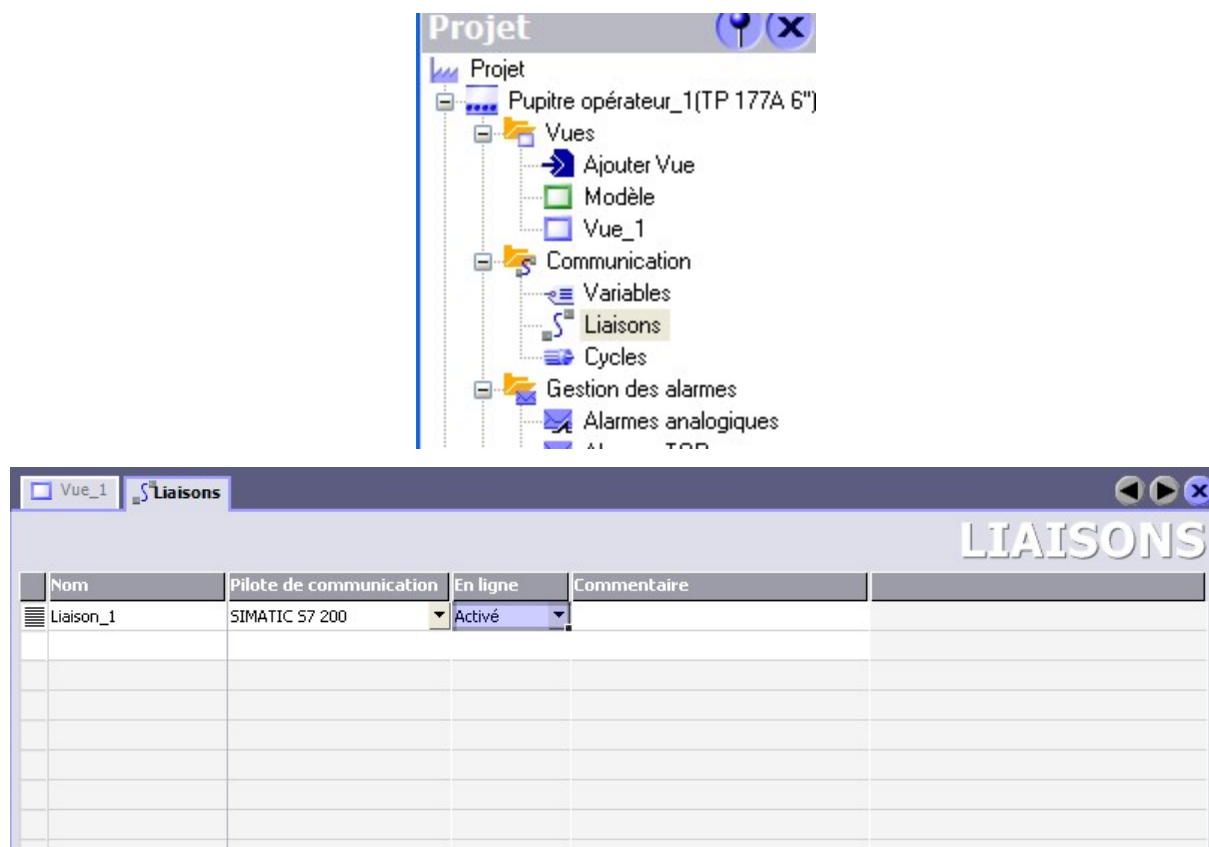
On définira :

- *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.



### III.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 S7-200.



### III.3. Création d'une vue (Menu) :

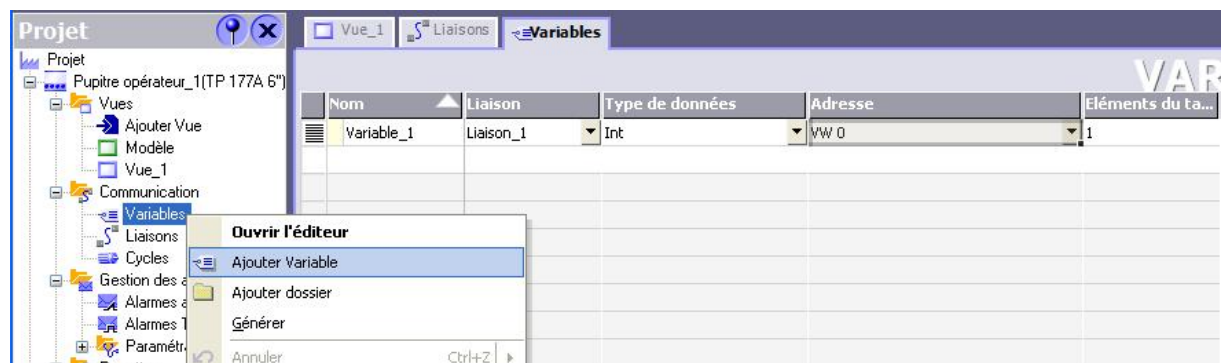
Les vues sont les éléments principaux du projet. Elles permettent de commander et de contrôler la machine. Elles peuvent comporter des champs de Sorties (Variable de type sortie), ou des champs d'Entrées (Variable de type entrée). Elles peuvent contenir d'autres informations telles que : zones de textes,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

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

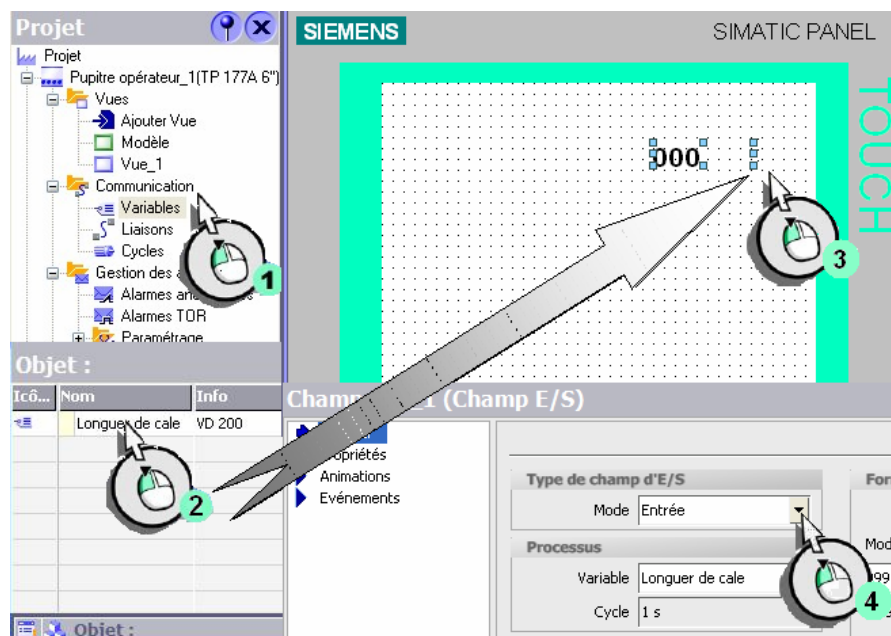
### III.4. Création de variables :

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 la variable à créer.



Et pour ajouter une variable sur une vue créée, on procède comme suit :



### III.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*.

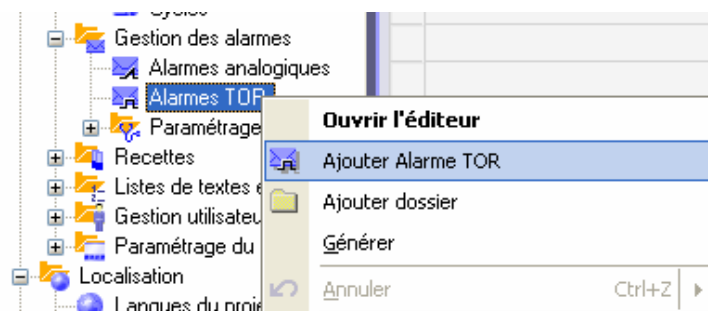
- ✚ Dans notre cas, on utilisera les *Alarmes TOR* pour indiquer : la fin du ruban, nombre de pièces à découper atteint et bouton d'arrêt d'urgence actionné (système Tout Ou Rien).

Pour configurer une alarme, on suit les étapes suivantes :

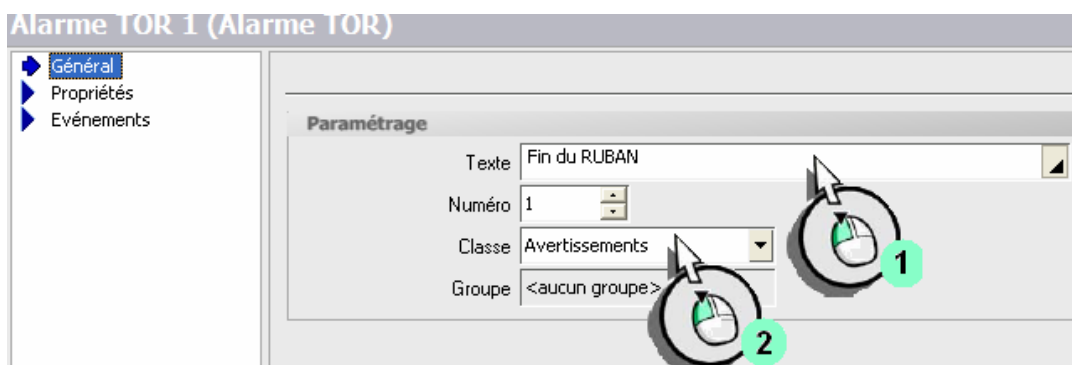
- ✓ **Créer une variable pour l'enregistrement des états :**

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

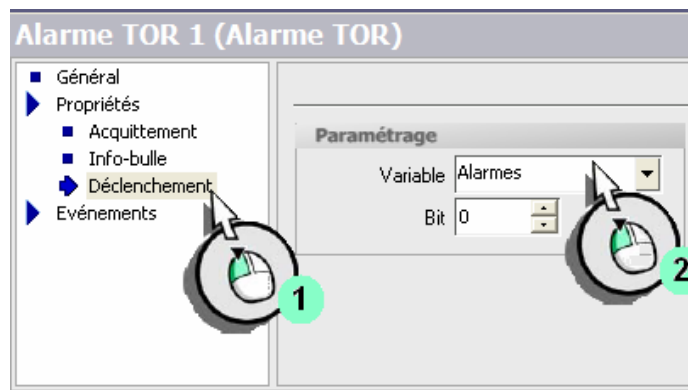
On crée alors une variable (VW 900) qu'on appellera : Alarmes, et chaque état est affecté à un bit de cette variable. Puis on ajoute autant d'*Alarmes TOR* dont on a besoin.



- Puis, on entre le Nom de l'alarme, et sa classe :



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



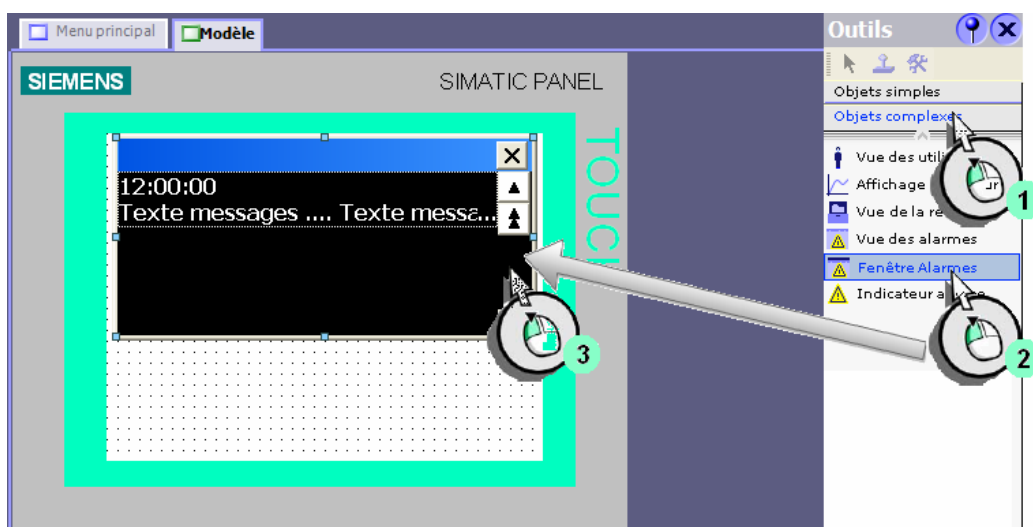
On procède de la même manière pour configurer l'alarme de « Nombre de pièces à découper atteint » et celle du « Bouton d'arrêt d'urgence actionné ».

Texte	Numéro	Classe	Variable de déclenchement	Numéro de bit
Fin 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

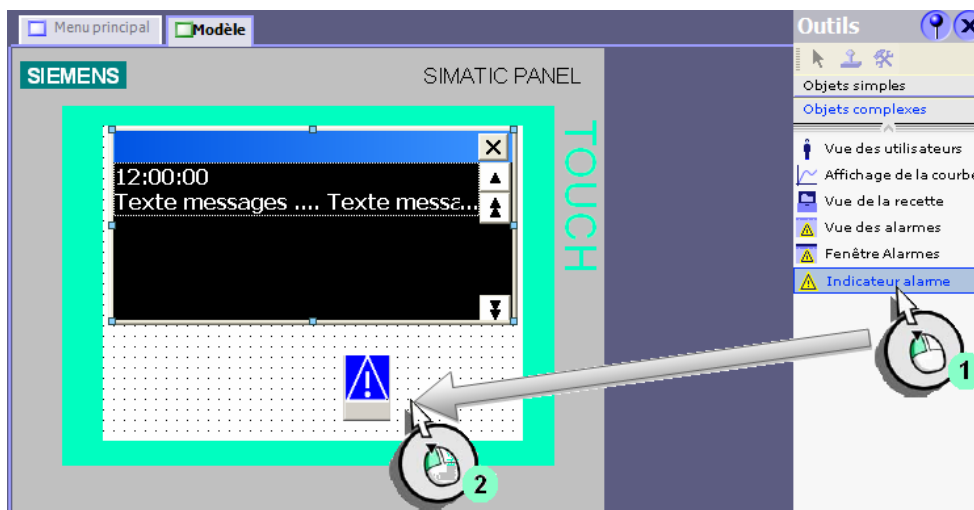
### III.6. Configurer l'affichage d'alarmes :

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.

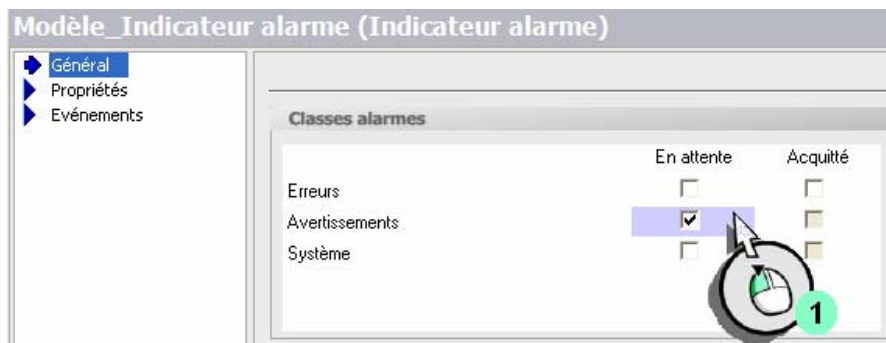
- Pour se faire, on crée une fenêtre d'alarmes dans le *modèle* à partir du menu *Outils* ⇒ *Objets Complexes*, on tire une *Fenêtre Alarmes* et on ajuste la taille qui convient comme le montre la figure suivante :



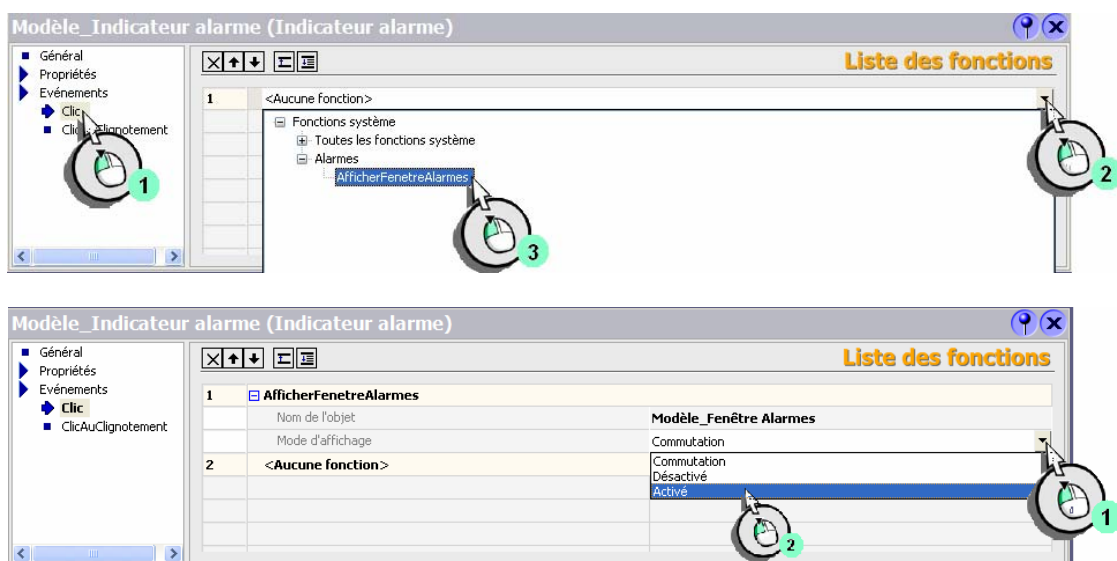
- Puis on insère un *Indicateur d'alarme* comme le montre la figure :



- On configure ensuite *l'indicateur d'alarme*.



- 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.



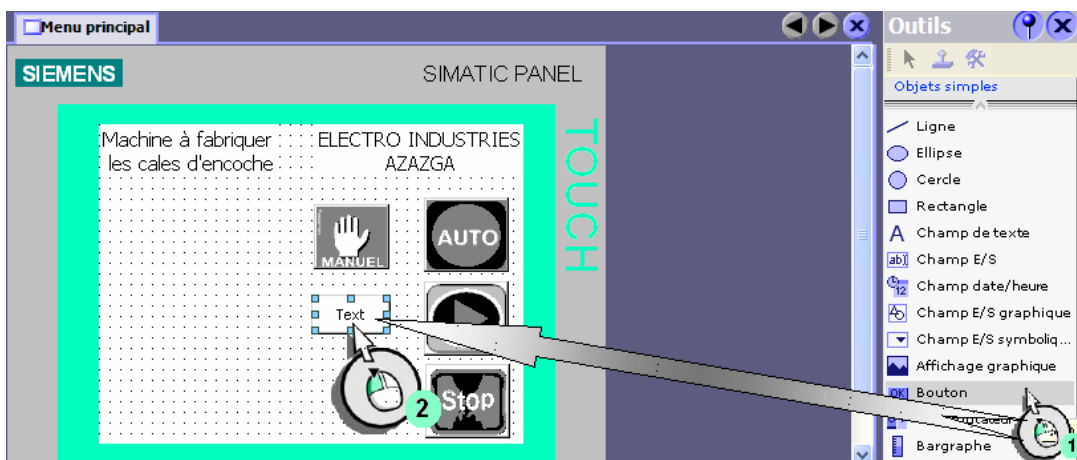
Ainsi, à chaque fois que le ruban est fini, nombre de pièces à découper atteint ou encore bouton d'arrêt d'urgence actionné, l'indicateur d'alarme s'affiche en premier plan ainsi que le nombre d'alarmes déclenchées.

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

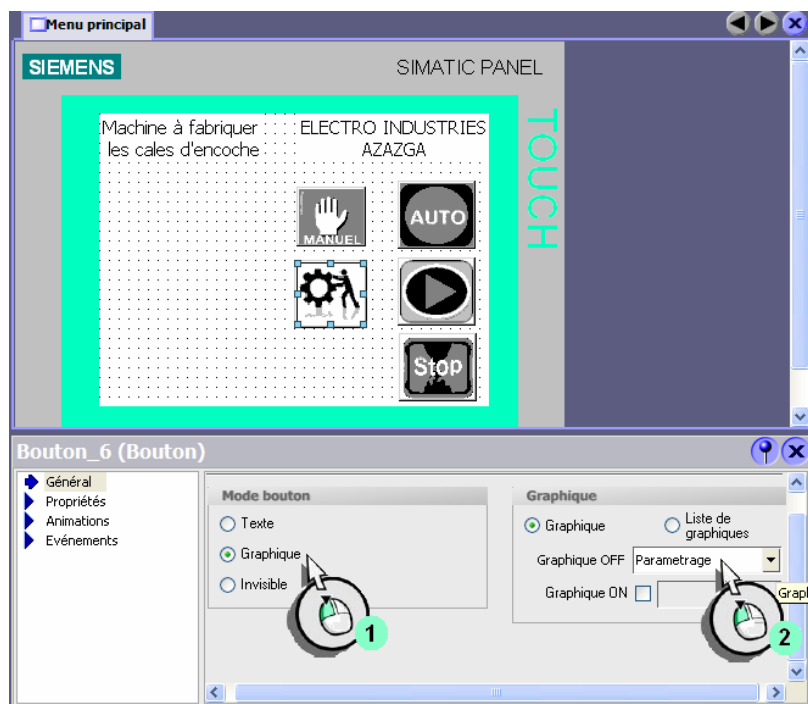
### III.7. Ajout de changements de vue :

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

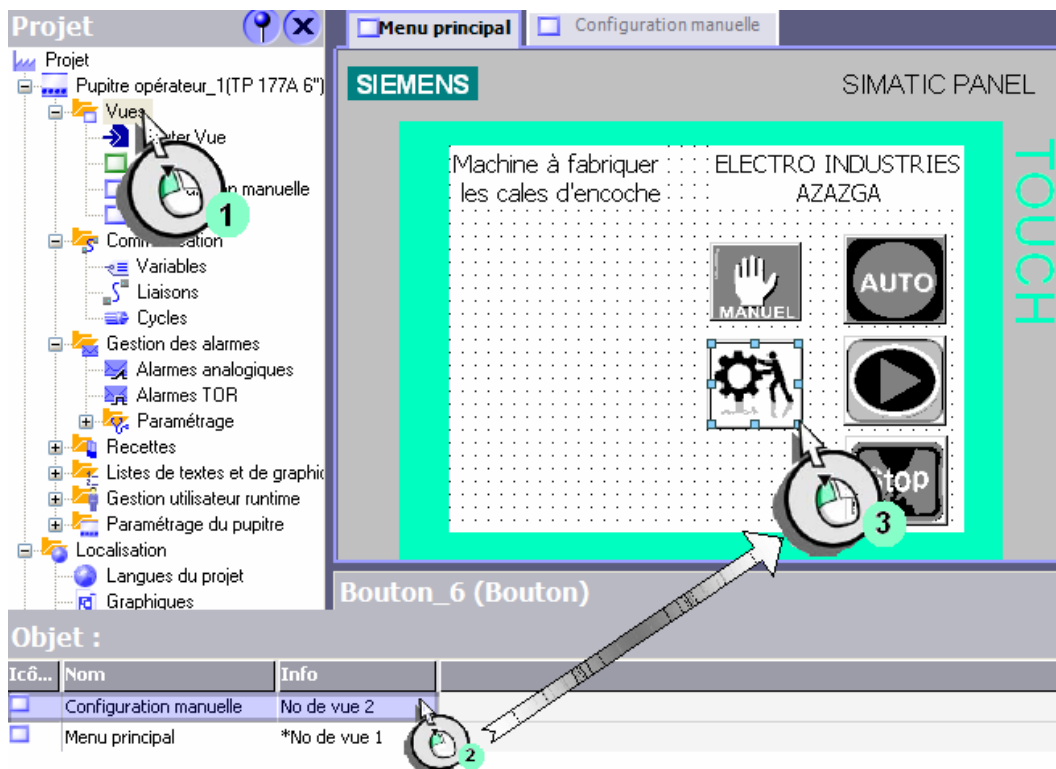
- Créer un bouton.



- Choisir *Mode Graphique* et associer une image (icône) au bouton.

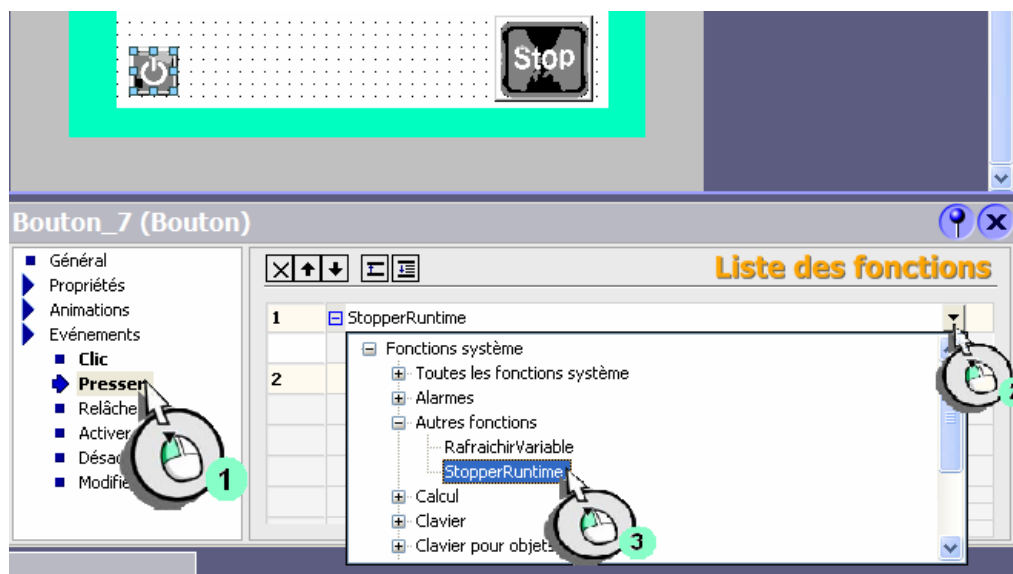


- Associer le bouton à une vue :



### III.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 :



### III.9. Test 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).

#### III.9.1. Exemple de test du projet (Compilation) :

➤ Démarrer le contrôle de cohérence :



Le résultat du contrôle de cohérence est affiché dans la fenêtre des erreurs et avertissements :

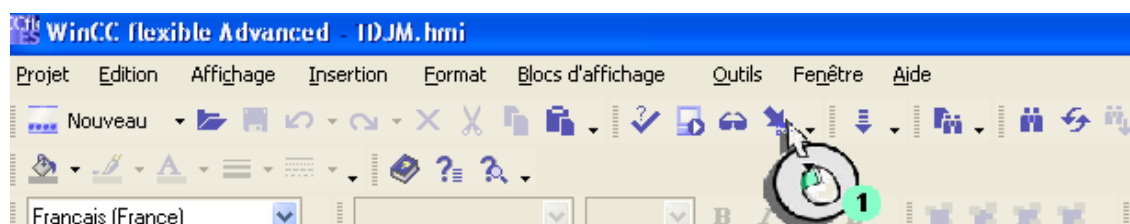
Fenêtres des erreurs et avertissements		
Heure	Catégorie	Description
20:35:59.71	Compilateur	Conversion des images en cours ...
20:35:59.71	Compilateur	Polices en cours de conversion ...
20:35:59.71	Compilateur	Vérification des résultats en cours ...
20:35:59.71	Compilateur	Ecriture des fichiers de sortie 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 !

#### III.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.

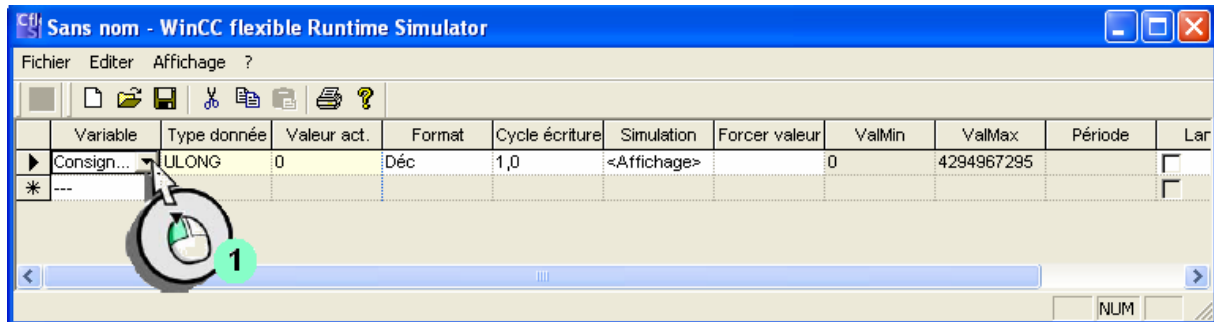
➤ **Créer une table de simulation :**

1. Démarrer le simulateur :

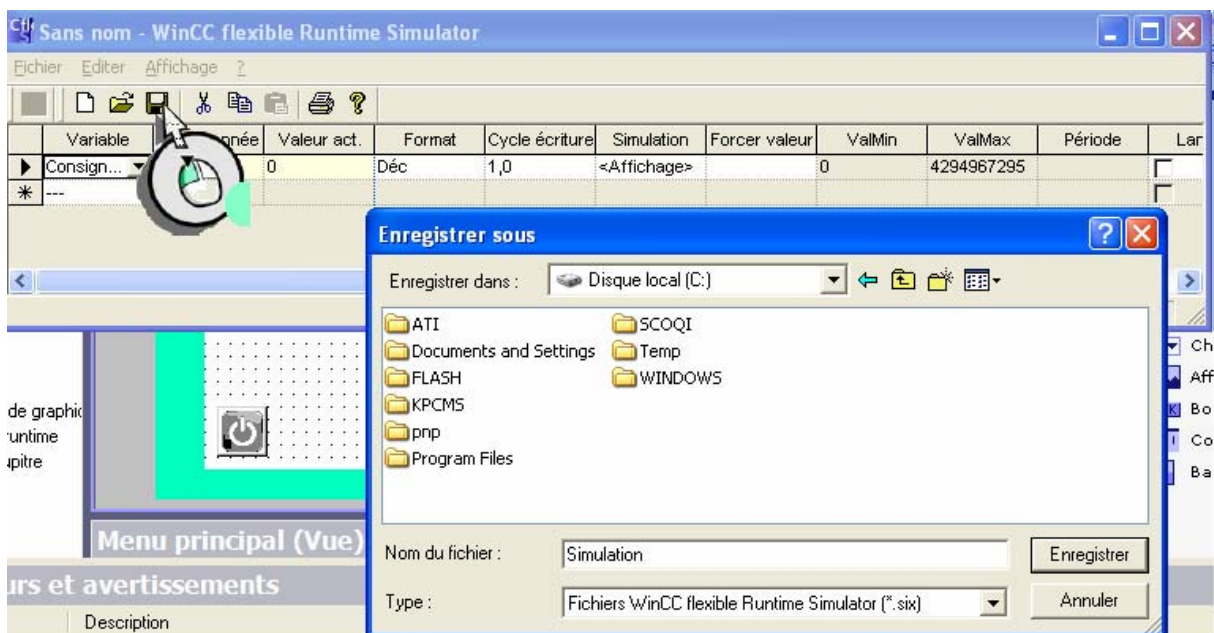




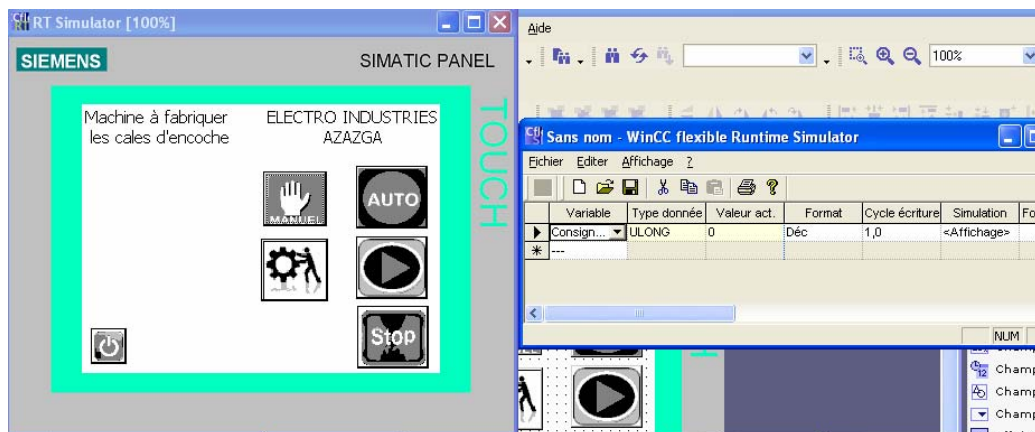
## 2. Sélectionner les variables à simuler :



## 3. Enregistrer la simulation :



## ➤ Simulation du projet :



**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'opérateur, en particulier dans le cas d'un danger d'une part, et pour un meilleur suivi du système selon le programme de la machine créé d'autre part.

## **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 S7-200 (CPU 226) de la firme SIEMENS ainsi que sur l'amélioration du système de supervision par un panel tactile TP177A.

Le stage pratique en milieu professionnel nous a été bénéfique dans la mesure où on a pu intégrer et approcher l'environnement industriel en appliquant nos acquis théoriques à la réalité de la fabrication industrielle.

L'utilisation du GRAFCET comme outil de modélisation, méthode d'analyse et de synthèse fiable, robuste et simple, nous a facilité la transcription de ce modèle en langage de programmation avec le logiciel STEP7 MICROWIN V4.0.

Ceci nous a également facilité l'élaboration d'une solution programmable à cette chaîne de traitement, et par la suite configurer le panel tactile TP177A par le logiciel WINCC FLEXIBLE.

Une fois l'automatisation réalisée, toutes les modifications souhaitées peuvent être configurées directement sur le programme sans pour autant manipuler la circuiterie.

Toutefois 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 S7-200 de Siemens.

## Bibliographie :

### 🔗 Ouvrages :

- C. Merlaud, J. Perrin, « Automatique, informatique industrielle » 1995 ;
- Alain Jacques, Jean-Claude LAFONT, Jean-Paul VABRE « Logique programmable et GRAFCET », 1987 ;

### 🔗 Manuels d'utilisation :

- Manuel d'utilisation de la machine à fabriquer les cales d'encoches (Electro-Industries) ;
- Manuel d'utilisation d'Automgen 8.0 (Ed. 2006) ;
- Manuel d'utilisation du Step 7 MicroWin v4.0 (Ed. 2004) ;
- Manuel d'utilisation du WinCC Flexible 2008.

### 🔗 Liens Internet :

- (1) <http://cvdjafarhadiouche.ifrance.com/>
- (2) <http://www.hellopro.fr/Electrovannes-pneumatiques-type-4-2--voies-2001704-787008-produit.html>
- (3) [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)
- (4) [http://www.sonelec-musique.com/electronique\\_realisations\\_alarme\\_001.html](http://www.sonelec-musique.com/electronique_realisations_alarme_001.html)
- (5) <http://hiconversion.net/conversion-rate-optimization-blog/321/inbound-marketing/be-a-thermostat-not-a-thermometer>
- (6) <http://lamaintenance.fr/electricite/electricite-le-contacteur/>