

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



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

**PROJET DE FIN D'ETUDES**

En vue de l'obtention du diplôme

*D'INGENIEUR D'ETAT EN AUTOMATIQUE*

**Thème**

Etude et adaptation d'un API sur une affûteuse

« Machine Walter AMS 1000 »

**Proposé par :**

M<sup>r</sup> I. TEBANI

**Dirigé par :**

M<sup>r</sup> A. DIRAMI

**Présenté par :**

M<sup>r</sup> IFRENE

Ahmed

M<sup>r</sup> OUKHERFELLAH Mourad

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

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

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

M<sup>r</sup> S. AKROUF

*Promotion 2009*

Projet réalisé à la Société Nationale de Véhicules Industriels « SNVI ».

## *REMERCIEMENTS*

*Nous remercions le dieu le tout puissant de nous  
avoir procuré la volonté, le courage et la force  
nécessaires pour mener à terme ce travail.*

*Nous remercions tous les gens qui nous ont aidé à  
faire ce travail, en particulier notre promoteur Mr  
DIRAMI, ainsi que notre encadreur  
Mr TEBANI.*

# Dédicace

*Pour que ma réussite soit complète je la partage avec  
tous les personnes que j'aime, je dédie ce modeste travail*

*à :*

*Mes très chers parents pour leur amour et les  
sacrifices qu'ils n'ont pas épargné pour  
contribuer à ma réussite*

*Mes chers frères et soeurs*

*Ma belle sœur*

*Tous mes neveux et mes nièces en particulier  
à la petite Inès*

*Toute la famille*

*Tous mes amis sans exception*

*Ahmed*

# Dédicace

*Pour que ma réussite soit complète je la partage avec  
tous les personnes que j'aime, je dédie ce modeste travail*

*à :*

*Mes très chers parents pour leur amour et pour  
le courage et la volonté qu'ils m'ont inculqué.*

*Mes chers frères, Amar, Toufik, Djamel et*

*Mokrane*

*Toute la famille*

*Tous mes amis sans exception*

*Mourad*

## SOMMAIRE

<b>Introduction générale</b> .....	1
<b>Présentation de la société</b> .....	2

### **Chapitre I : DESCRIPTION ET FONCTIONNEMENT DE LA MACHINE**

<b>I- Introduction</b> .....	4
<b>II- Description de la machine</b> .....	5
II-1- Partie mécanique .....	5
II-1-1- Le bâti .....	5
II-1-2- La tête porte meule .....	5
II-1-3- La broche porte outil .....	6
II-1-4- Le doigt diviseur .....	6
II-1-5- Le plateau basculant et ses butées .....	6
II-1-6- les butées du chariot porte meule .....	7
II-2- Partie hydraulique .....	7
II-2-1- Graissage .....	7
II-2-2- Installation hydraulique .....	7
II-2-2-1- Les vérins et moteurs hydrauliques .....	8
II-2-2-2- Les distributeurs .....	8
II-2-2-3 Pupitre de commande hydraulique .....	9
II-3- Partie électrique .....	11
II-3-1- Les relais .....	11
II-3-2- Les relais thermiques .....	12
II-3-3 Les électrovannes .....	13
II-3-4- Les moteurs .....	13
II-3-5- Les transformateurs .....	14

II-3-6- Pupitre de commande .....	14
II-3-7- Les capteurs .....	16
II-3-8- Les signalisations (voyants) .....	16
<b>III Description du fonctionnement a partir du schéma électrique .....</b>	<b>17</b>
<b>VI Conclusion .....</b>	<b>19</b>

## **Chapitre II : Modélisation du fonctionnement de la machine**

<b>I- Introduction .....</b>	<b>20</b>
<b>II- Définition du GRAFCET .....</b>	<b>20</b>
<b>III- Structure du GRAFCET .....</b>	<b>20</b>
III-1- Etape .....	21
III-2- Transition .....	21
III-3- Liaisons orientées .....	22
III-4- Règles d'évolution du GRAFCET .....	22
III-5- Sélection de séquence et séquence simultanée .....	24
III-6 Saut d'étapes .....	25
III-7- Reprise de séquence .....	26
<b>IV- Niveau d'un Grafcet .....</b>	<b>27</b>
IV-1- Grafcet niveau 1 .....	27
IV-2- Grafcet niveau 2 .....	27
<b>V- Mise en équation d'un Grafcet .....</b>	<b>28</b>
<b>VI- application du GRAFCET pour modéliser la machine .....</b>	<b>30</b>
VI- 1 GRAFCET de sécurité niveau 1 .....	30
VI- 2 GRAFCET de sécurité niveau 2 .....	31

VI-3 GRAFCET de fonctionnement niveau 1 .....	32
VI- 4 GRAFCET de fonctionnement niveau 2 .....	35

**Chapitre III : Automatisation de la machine par l'Automate Programmable Industriel S7-300**

<b>I- Introduction</b> .....	38
<b>II- Automates programmables industriels API</b> .....	38
II-1- Définition .....	38
II-2- Architecture et gammes d'automates .....	38
II-3- Définition d'un système automatisé .....	39
II-4- Structure générale d'un système .....	40
II-5 Choix de l'automate .....	42
III-1- Caractéristiques de l'API S7-300 .....	42
III- Différents parties de l'API S7-300 .....	42
III-2-1 Module d'alimentation .....	43
III-2-2 L'unité centrale .....	43
III-2-3 Les modules d'entrée/sortie .....	43
III-2-3-a Modules d'entrée .....	43
III-2-3-b Les modules de sortie .....	44
III-2-4 Les coupleurs .....	44
III-2-5 Modules de fonction (FM) .....	44
III-3 Fonctionnement de l'API .....	44
III-3-1 Réception des informations sur les états du système .....	45
III-3-2 Système d'exploitation .....	45
III-3-3 Exécution du programme utilisateur .....	45
III-3-4 Commande de processus .....	45
III-4 Nature des informations traitées par l'API .....	45
<b>IV- Programmation</b> .....	46

IV-1	Création d'une solution d'automatisation dans STEP 7.....	46
IV-2	Les langages de programmation .....	47
IV-2-1	Le langage contact (CONT) .....	47
IV.2.2.	Logigramme (LOG) .....	48
IV.2.3.	Liste d'instructions (LIST) .....	48
IV-3	L'opération de temporisation .....	49
IV.4	Création d'un programme utilisateur .....	49
IV.4.1.	Lancement du logiciel .....	49
IV.4.2.	Création du projet .....	49
IV.4.3.	Création de la table des mnémoniques .....	52
<b>V</b>	<b>Configuration et paramétrage du matériel .....</b>	<b>52</b>
V.1	Adressage des signaux d'entrées/sorties .....	53
V.2	Mémentos .....	54
V.3	Traitement du programme par l'automate .....	54
<b>VI.</b>	<b>Principe de conception d'une structure de programme .....</b>	<b>55</b>
VI.1.	Système d'exploitation .....	55
VI.2.	Programme utilisateur .....	55
VI.3	Blocs dans le programme utilisateur .....	56
VI.3.1.	Bloc utilisateur .....	56
VI.3.2.	Blocs système pour fonctions standard et fonction système .....	57
<b>VII.</b>	<b>Traitement du programme par la CPU .....</b>	<b>57</b>
VII.1.	Traitement linéaire du programme .....	57
VII.2	Traitement structuré du programme .....	57
<b>VIII.</b>	<b>Différents blocs d'organisation des S7-300 .....</b>	<b>58</b>
<b>IX.</b>	<b>Implantation du GRAFCET dans le S7-300 .....</b>	<b>58</b>
<b>IIX.</b>	<b>conclusion .....</b>	<b>59</b>

## **Chapitre IV : SIMULATION ET VALIDATION PAR S7-PLCSIM**

<b>I introduction</b> .....	60
<b>II Présentation du S7-PLCSIM</b> .....	60
<b>III- Commande de la CPU</b> .....	60
III-1- La fenêtre CPU .....	60
III-2- les modes de fonctionnement de la CPU .....	61
III-3 Mise en route du logiciel S7-PLCSIM .....	61
IV-1 Visualisation de l'état du programme .....	63
IV-2 Simulation du programme de l'affûteuse .....	63
V Conclusion .....	64
<b><i>Conclusion générale</i></b> .....	65
<b>ANNEXE</b>	

# Introduction générale

---

Historiquement, l'évolution des automatismes s'est déroulée en trois étapes avec deux grandes ruptures. Chaque étape correspond à un type de logique de pensée. Ainsi on passe d'une vision imitative (imitations des vivants) à une vision mécaniste dans les années 50 (opérationnalisation des machines, la machine remplace l'homme dans les tâches difficiles et répétitives : Ford, Taylor) pour terminer vers une vision actuelle systématique (c'est l'ère de l'intelligence artificielle qui est née de la cybernétique de Wiener, la machine réfléchie, analyse, s'adapte, la spécialisation a fait place à la polyvalence et à la capacité d'adaptation).

D'un point de vue économique et sociale, les automatismes ont des impacts économiques très importants dans le monde industriel telle que le gain de temps dû à la robotisation, une meilleure productivité, un rendement optimisé ..., mais aussi ils amènent des incidences sociales fortes autant, pour le travailleur (sécurité renforcée, adaptabilité par la polyvalence ...), que pour le citoyen (communication variée, commande à distance ...).

Techniquement, les applications des automatismes sont vastes : Industrielles, environnementaux, domestiques (systèmes de régulation et de sécurité), gestion à distance ... etc.

Dans ce travail on se limite dans l'automatisme industriel, et bien précisément l'étude d'une machine-outil commandée en logique câblée (relais, temporisateurs, ...) en vue de la modernisation de cette partie de commande en adaptant un API (Automate Programmable Industriel) et en tenant compte de tous les changements nécessaires à cet effet.

Le remplacement de la commande en logique câblée par celle en logique programmée présente plusieurs avantages, telle que :

- Un temps de réponse des actionneurs très réduit.
- Une commande très précise.
- Une armoire de commande moins encombrante.
- Une parfaite accessibilité pour la maintenance.
- Un pouvoir de modification de programme de fonctionnement.
- Un pouvoir de communication avec l'automate.

# Présentation de la société

---

## PRESENTATION DE LA SOCIETE

Pour répondre aux besoins d'intégration de la métropole française, BERLIET entreprend en juin 1957 la construction d'une usine de poids lourds dans la zone industrielle de ROUIBA sise à 30 km à l'est d'ALGER.

Par suite de la série de nationalisations des entreprises, en 1967, BERLIET ALGERIE est devenue SONACOME (Société Nationale de Constructions Mécaniques). En fin, le schéma de reconstruction de la SONACOME retenu a donné naissance à 11 entreprises nouvelles, dont la SNVI (Société Nationale de Véhicules Industriels) en 12 Décembre 1981 sous le décret n° 81/48.

La superficie de cet ensemble industriel s'étend sur 260 Hectares dont 30 sont couverts.

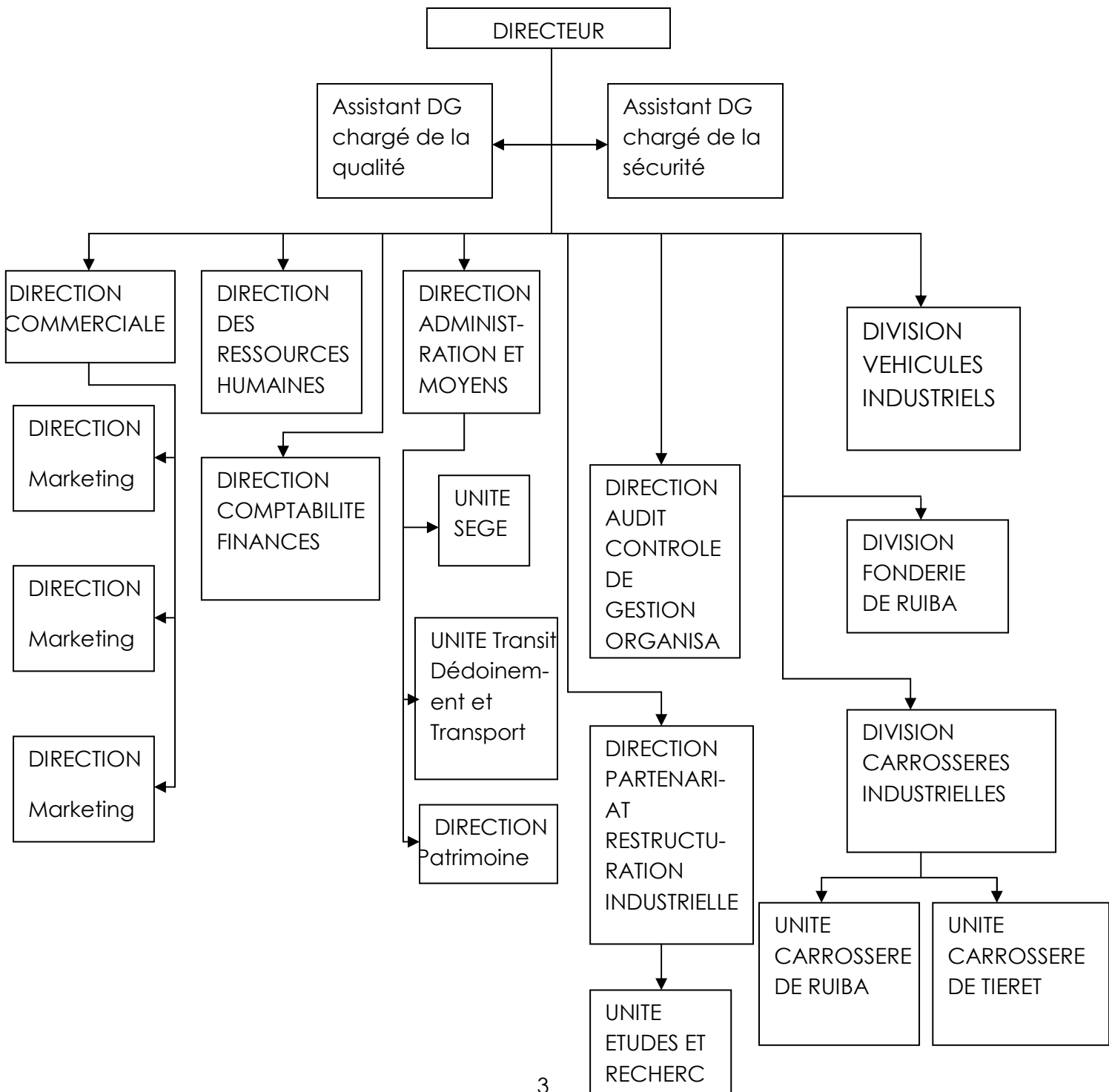
Les missions de la SNVI sont de fabriquer, distribuer, maintenir et exporter des véhicules industriels. Actuellement la SNVI ROUIBA se compose de 3 unités qui sont : la DVI ; la Carrosserie, la Fonderie ainsi que la Carrosserie de Tiaret (Annexe).

La DVI (Division Véhicules Industriels) est l'une des plus importantes unités de la société, elle se compose de :

- Direction des Fabrications.
- Bâtiment Mécanique.
- Bâtiment Forgeage.
- Tolerie -Emboutissage.
- Bâtiment de Montage Autobus.
- Bâtiment de Montage Camions.
- Département de Gestion Industrielle.

On a effectué notre stage au niveau du Secteur Affûtage qui fait partie du Bâtiment Mécanique.

## ORGANIGRAMME DE LA SNVI



# **Chapitre I**

## **DESCRIPTION ET FONCTION DE LA MACHINE**

**I- Introduction :**

L'affûteuse est une machine qui sert affûter les outils d'usinage. C'est l'une des machines qui forment le secteur affûtage au niveau du bâtiment mécanique dans la Division Véhicules Industriels.

L'affûteuse de type WALTER AMS 1000 joue un rôle très important au niveau du secteur affûtage vu sa fiabilité et sa robustesse.

Dans le présent chapitre, nous présentons la machine étudiée dans le cadre de notre stage, en décrivons particulièrement son fonctionnement et ses différentes parties.

La figure (I-1) illustre l'allure de cette machine :



Figure (I-1) : machine à affûter les fraises de type WALTER AMS 1000.

- 1- Chariot porte meule.
- 2- Tête porte meule.
- 3- Le doigt diviseur.
- 4- Broche porte outil.
- 5- Chariot porte outil.
- 6- Pupitre de commande électrique.
- 7- Monte et baisse.
- 8- Le bâti.
- 9- Plateau basculant.
- 10- Butée du plateau basculant.
- 11- Pupitre de commande hydraulique.

## **II- Description de la machine :**

La machine peut être décrite en la décomposant en trois parties principales :

- Une partie mécanique
- Une partie électrique
- Une partie hydraulique

### **II-1- Partie mécanique :**

#### **II-1-1- Le bâti :**

Il forme la structure porteuse de l'ensemble des parties de la machine (unités de travail, tableau de commande, ...)

#### **II-1-2- La tête porte meule :**

La tête porte-meule peut être déplacée verticalement sur le chariot porte-meule à l'aide d'un volant, et peut pivoter de 360° autour de son axe vertical. Le moteur d'entraînement et la broche porte-meule peuvent en outre basculer autour d'un axe horizontal.

Le moteur de la meule est à courant continu alimenté par le groupe Léonard qui se trouve monté dans le bâti de la machine. Ceci permet un réglage progressif de la vitesse de la meule.

La broche porte-meule est montée sur un roulement à billes et elle est uniquement bloquée sur son diamètre extérieur, de sorte qu'elle pourra être facilement changée.

### **II-1-3- La broche porte outil :**

La broche porte-outil est commandée par un moteur hydraulique à vitesse réglable par l'intermédiaire de pignons. Deux tuyaux fournissent l'huile sous pression à partir du bloc de commande. Lorsqu'on affûte avec division automatique dent par dent, on intercale les pignons réducteurs pour obtenir une rotation lente.

Le levier d'accouplement doit être en position « à bas ».

Pour réaliser l'accouplement, on tire le levier d'accouplement en arrière afin de passer le verrouillage, on tourne alors lentement la broche porte-outil, jusqu'à ce que les dents de l'embrayage viennent en prise. Après cela on repousse le levier qui se verrouille. Pour l'affûtage sur la périphérie des fraises, et, en particulier, pour les petits diamètres, il faut se mettre en prise directe par les pignons afin d'avoir une vitesse plus grande. En réglant le débit d'huile on peut, dans les deux cas, faire varier la vitesse de rotation de la broche porte-outil dans de grandes limites.

### **II-1-4- Le doigt diviseur :**

Le doigt diviseur donne les ordres pour l'opération de division. Lorsqu'on effectue la division d'une dent à l'autre, la pointe du doigt diviseur tourne autour d'une articulation. Deux leviers transmettent ce mouvement à l'interrupteur de fin de course b17 qui déclenche par l'intermédiaire de la commande hydraulique la rotation en arrière de la fraise jusqu'à ce que la face de coupe de la dent vienne buter. Lorsque la face de coupe de la dent vient buter, la douille recule de quelques millimètres en comprimant son ressort, jusqu'à ce que le fin de course b18 provoque l'approche du chariot porte-meule. Grâce aux vis du bloc hydraulique, il faut réduire la vitesse du mouvement de division suffisamment pour que les mouvements de commande puissent s'effectuer avec précision.

### **II-1-5- Le plateau basculant et ses butées :**

Les butées du plateau basculant servent à régler l'angle que font la meule boisseau et la meule tangente. Le réglage approché de ces butées se fait à l'aide de noix coulissantes dans les rainures en T circulaires. Les parties mobiles des butées permettent un réglage précis de ces angles en combinaison avec les index qui se trouvent fixés sur ces parties et avec le grand cadran circulaire. Les cames 48 et 49, actionnent les interrupteurs de fin de course b19 et b20 qui déclenchent l'inversion du mouvement de basculement et le dégagement du chariot porte-meule. Le mouvement de basculement se fait toujours jusqu'à ce que les butées 45 et 46

viennent contre les butées 50 et 51. La première partie des cames 48 et 49 sert à réduire la vitesse du mouvement de basculement par l'intermédiaire d'une soupape électro-magnétique afin que le chariot ne vienne pas heurter trop fortement contre les butées 50 et 51.

### **II-1-6- les butées du chariot porte-meule :**

Servent à limiter la course du chariot lors du mouvement de dégagement, mais ces butées sont en outre combinées avec des interrupteurs de fin de course électriques b21 et b22 (voir schéma électrique). Ces fins de course sont montés à demeure et ils fonctionnent chaque fois que le chariot arrive en butée. Chaque fois que le chariot pousse b21 (à droite) et b22 (à gauche), ils déclenchent l'opération de division, pour passer à la dent suivante (lorsqu'on est en fonctionnement automatique) et chaque fois que le chariot les quitte, ils actionnent le mouvement de pivotement du plateau basculant.

### **II-2- Partie hydraulique :**

Tous les mouvements principaux de l'AMS 1000 sont à commande hydraulique. Les distributeurs et vannes sont à commande électromagnétique.

#### **II-2-1- Graissage :**

Graisser tous les graisseurs en suivant le plan de graissage.

Pour le transport, les butées du chariot porte-meule à gauche et à droite, ainsi que les butées pour le mouvement de basculement sont bloquées sur zéro. Pour pouvoir déplacer le chariot porte-meule, il faut tourner les vis moletées des deux cotés vers la droite, jusqu'à ce que le système basculant ait suffisamment de jeu pour déclencher le mouvement d'inversion de marche. Les butées du plateau basculant doivent être décalées suffisamment pour permettre un pivotement de 30°.

#### **II-2-2- Installation hydraulique :**

Elle comporte une centrale hydraulique qui génère différents débits permettant l'entraînement de tout le système hydraulique.

Le système hydraulique se compose de trois parties :

- Une pompe hydraulique qui a pour rôle l'entraînement des différents vérins et moteurs hydrauliques.
- Un ensemble de distributeurs qui servent de commande pour le circuit hydraulique.

- Les vérins et moteurs hydrauliques pour agir sur la partie mécanique afin de réaliser différents mouvements nécessaires pour le fonctionnement de la machine.

En plus des trois parties précitées s'ajoute la partie contrôle qui se résume dans les différents pointeaux et vis de réglage.

### II-2-2-1- Les vérins et moteurs hydrauliques :

On peut distinguer des vérins à simple effet, à double effet, ou des vérins rotatifs.

Les vérins et moteurs hydrauliques qui existent sur la machine sont récapitulés dans le tableau ci-après :

Type	Rôle
Vérin simple effet à retour à ressort	Avance automatique
Vérin double effet	Déplacement du chariot porte meule
Vérin simple effet	Support basculant
Moteur hydraulique	Rotation de la fraise
Moteur hydraulique	Plateau basculant

Tableau (I-1) : Récapitulatif des vérins et moteurs hydrauliques

### II-2-2-2- Les distributeurs :

Il existe des distributeurs pneumatiques et hydrauliques commandés électriquement (par électrovannes), ou par pilotage (hydrauliques, pneumatiques), leur fonction est la commande des vérins.

Les distributeurs qui existent sur la machine sont récapitulés dans le tableau (I-2):

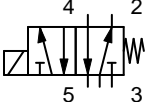
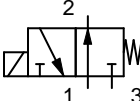
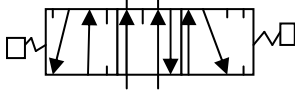
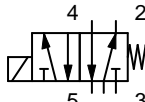
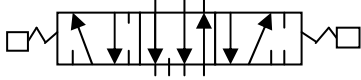
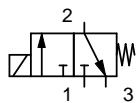
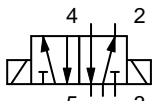
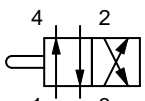
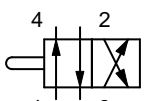
Nomenclature	Type	Rôle
S 1	Distributeur électro-magnétique 5/2 	Avance et dégagement du chariot porte meule
S 2	Distributeur électro-magnétique 3/2 	Commande le support basculant
S 3	Distributeur électro-magnétique 7/3 	Commande la rotation de la fraise
S 4	Distributeur électro-magnétique 5/2 	Ralentissement du mouvement du support basculant
S 5	Distributeur électro-magnétique 7/3 	Mouvement de basculement
A	Distributeur 3/2 	Avance automatique
-	Distributeur 5/2 	Déplacement du chariot porte meule
<b>D</b>	Distributeur 4/2 	Commutation mécanique automatique
-	Distributeur 4/2 	Blocage du chariot porte meule

Tableau I-2 : Récapitulatif des distributeurs

### II-2-2-3 Pupitre de commande hydraulique :

Sur le pupitre de commande hydraulique, on dispose de pointeaux et de vis de réglage pour régler les pressions et les débits.

La figure (I-2) représente le pupitre de commande hydraulique de la machine.

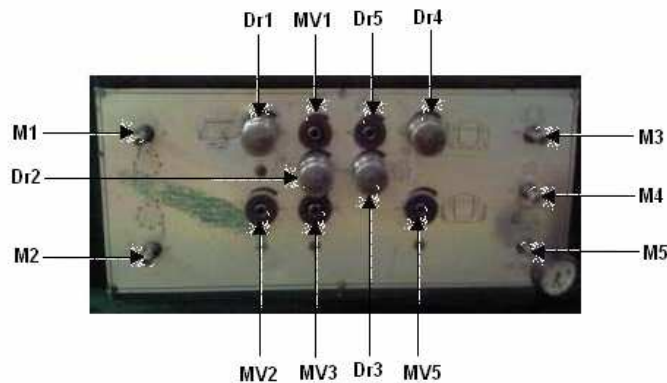


Figure (I-2) : Pupitre de commande hydraulique

M1, M2 : Manostats, donnent les pressions dans les canalisations d'amenée au moteur hydraulique pour la rotation de la fraise.

M3 : Manostat, donne la pression dans les canalisations d'amenée au moteur hydraulique pour le plateau basculant.

M4 : Manostat, donne la pression dans la canalisation d'amenée au distributeur pour le chariot porte meule.

M5 : Manostat, donne la pression dans la canalisation d'amenée au piston pour le dispositif d'oscillation.

MV1 : Sert à régler les pressions de service pour le mouvement de va-et-vient de la tête porte-meule.

MV2 : Sert à régler les pressions de service pour le mouvement du support basculant.

MV3 : Sert à régler les pressions de service pour le mouvement du chariot port fraise.

MV5 : Sert à régler les pressions de service pour le mouvement du plateau basculant.

Dr1 : Sert à régler le débit pour le mouvement de va-et-vient de la tête porte-meule.

Dr2, Dr3 : Servent à régler le débit d'huile pour la rotation de la fraise

Dr4 : Sert à régler le débit du moteur du plateau basculant.

Dr5 : Sert à régler le débit pour le ralentissement du mouvement de basculement.

### II-3- Partie électrique :

La partie électrique est constituée d'une armoire, d'un tableau de commande et d'organes sur la machine interconnectés par l'intermédiaire de câbles électriques.

#### II-3-1- Les relais :

Le tableau (I-3) cite l'ensemble des relais existants dans la machine.

Nomenclature	Type	Rôle
c1	Relais	Commande la pompe hydraulique
c2	Relais	Commande le groupe Léonard
c3	Relais	Commande la pompe d'aspiration de poussière et d'arrosage
d1	Relais	Commande le démarrage de la meule
d2	Relais	Commande la rotation de la fraise
d3	Relais	Commande l'avance du chariot porte meule en position de travail
d4	Relais	Commande l'avance du chariot porte meule en position de travail
d5	Relais	Commande la rotation de la fraise dans le sens inverse
d6	Relais	Commande la marche en sens inverse de la pièce
d7	Relais	Basculement automatique du plateau à droite
d8	Relais	Met en route le basculement automatique du plateau
d9	Relais	Met en route le basculement automatique du plateau en sens inverse
d10	Relais	Basculement automatique du plateau à gauche
d11	Relais	Basculement du plateau à droite
d12	Relais	Actionne le ralentissement du plateau basculant
d14	Relais temporisé	Shunte pendant un temps très court le contact d15
d15	Relais	Dégagement du chariot porte-meule

	temporisé	
d16	Relais	Actionne à retardement le dispositif de commande automatique
d18	Relais temporisé	Avance du chariot en position de travail
d19	Relais temporisé	Permet le dégagement du chariot
d20	Relais temporisé	Actionne à retardement le basculement du plateau
d21	Relais temporisé	Déclanche le mouvement de basculement
d22	Relais temporisé	Commande l'avance du chariot porte-meule
d23	Relais temporisé	Retarde le moment où la dent vient contre le doigt

Tableau (I-3) : Les relais

### II-3-2- Les relais thermiques :

Les relais de protection thermique servent comme une protection pour les moteurs contre la surchauffe.

On résume dans le tableau (I-4) suivant le rôle des relais thermiques qui se trouvent dans la machine.

Nomenclature	Type	Rôle
e11	Relais thermique	Protection du moteur m1
e12	Relais thermique	Protection du moteur m2
e13	Relais thermique	Protection du moteur m3
e14	Relais thermique	Protection du moteur m4
e15	Relais thermique	Protection du moteur m5

Tableau (I-4) : Récapitulatif des relais thermiques

### II-3-3 Les électrovannes :

Une électrovanne est un dispositif qui transforme un signal de commande électrique en une action sur un distributeur pneumatique ou hydraulique.

Le tableau (I-5) récapitule l'ensemble des électrovannes ainsi que leurs rôles.

Nomenclature	Type	Rôle
<b>S1</b>	Electro-aimant simple	Avance et dégagement du chariot porte meule
<b>S2</b>	Electro-aimant simple	Commande le support basculant
<b>S3.1</b> <b>S3.2</b>	Electro-aimant double	Commande la rotation de la fraise
<b>S4</b>	Electro-aimant simple	Ralentissement du mouvement du support basculant
<b>S5.1</b> <b>S5.2</b>	Electro-aimant double	Mouvement de basculement

Tableau (I-5) : récapitulatif des électrovannes

#### II-3-4- Les moteurs :

Les moteurs jouent un rôle essentiel dans la chaîne de commande. Ils peuvent être classés suivant plusieurs paramètres.

Le tableau (I-6) définit les caractéristiques des moteurs de la machine.

Nomenclature	Type	Rôle	Vitesse nominale	Puissance
m1	Moteur 3ph alternatif	Entraînement de la pompe hydraulique qui délivre : - Débit 6.3 l/mn - Débit 6.3 l/mn - Débit 20 l/mn - Débit 20 l/mn - Débit 6 l/mn	1500 tr/mn	4 kW
m7	Moteur 3ph Alternatif	Monte et baisse de la broche porte meule	-	0.37 kW
m3	Moteur 3ph		-	

m10	Génératrice	Variation de vitesse		2.6 kW
m11	Génératrice			
m9	moteur à courant continu	Rotation de la meule	2800 tr/mn	2.3 kW
m4	Moteur 3ph	Aspiration de la poussière	-	0.55 kW
m5	Moteur 3ph	Rotation de la pompe d'arrosage	-	0.15 kW

Tableau (I-6) : Récapitulatif des moteurs

### II-3-5- Les transformateurs :

Ce sont des machines électriques statiques. Ils ont pour rôle la transformation de la tension électrique.

On résume les caractéristiques des transformateurs utilisés sur l'affûteuse dans le tableau (I-7) suivant.

nomenclature	tensions	Fusibles de protection	Rôle
<b>T1</b>	380/30V	e8	Alimentation du circuit de commande.
<b>T2</b>	380/110V	e9	Alimentation du circuit de commande des distributeurs électromagnétiques.
<b>T3</b>	380/24V	e7	Alimentation de la prise.

Tableau (I-7) : Résumé des caractéristiques des transformateurs

### II-3-6- Pupitre de commande :

Elle est l'interface qui permet à l'opérateur de communiquer avec l'automatisme.

On cite tous les boutons qui forment le tableau de commande dans le tableau (I-8).

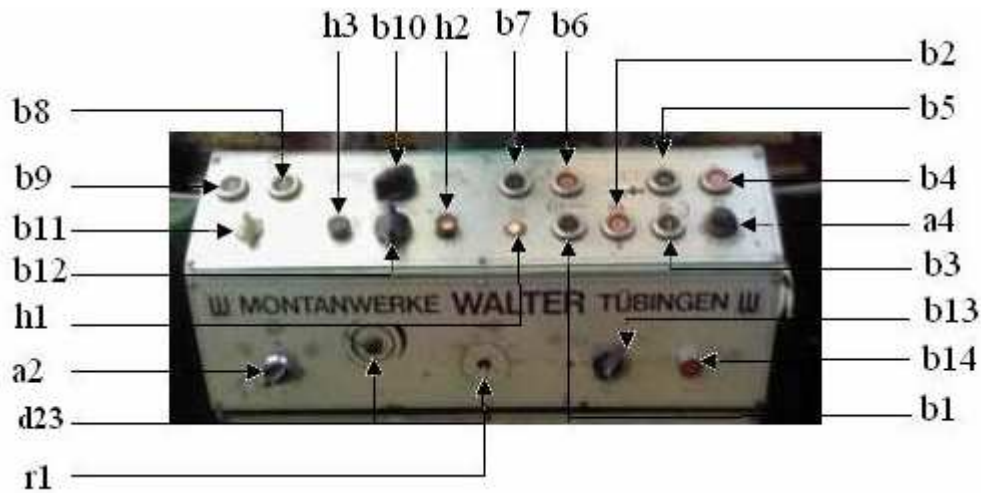


Figure (I-3) : Pupitre de commande électrique

Nomenclature	Type	Rôle
a1	interrupteur	Mise sous tension de la machine
a2	Inverseur	Donne deux sens de rotation et une position d'arrêt pour la meule
a3	Commutateur	Commute entre l'aspiration de la poussière et l'arrosage
a4	Commutateur	Commande la monte et baisse du chariot
b1	Bouton poussoir	- Met en marche la pompe hydraulique - Alimente le circuit de commande
b2	Bouton poussoir	Commande l'arrêt de la pompe hydraulique et le groupe Léonard
b3	Bouton poussoir	Met en route le groupe Léonard
b4	Bouton poussoir	Commandent ensemble le dispositif d'aspiration de poussière et la pompe d'arrosage
b5	Bouton poussoir	
b6	Bouton poussoir	Arrêt du cycle va-et-vient
b7	Bouton poussoir	Bouton va-et-vient
b8	Bouton poussoir	Basculement du plateau à droite
b9	Bouton poussoir	Basculement du plateau à gauche
B10	Commutateur	Commande l'affûtage périphérique
b11	Commutateur	Blocage du plateau à gauche ou à droite

b12	Commutateur	Commande l'affûtage périphérique
b13	Sélecteur	Permet de choisir un des trois modes de fonctionnement
b14	Commutateur	Commande le dispositif d'affûtage hélicoïdal
r1	Potentiomètre	Permet de régler la vitesse de rotation de la meule

Tableau (I-8) : Récapitulatif des boutons

### II-3-7- Les capteurs :

Les capteurs sont des composants de la chaîne d'acquisition dans une chaîne fonctionnelle.

Le tableau (I-9) récapitule les capteurs qui assurent l'acquisition des informations sur la machine.

Nomenclature	désignation	Information délivrée
b17	Fin de course	Le doigt diviseur a dépassé l'arrête de la dent
b18		La fraise a avancé d'une dent
b19		Le plateau est arrivé à la butée droite
b20		Le plateau est arrivé à la butée gauche
b21		Interrupteur pour le dégagement du chariot vers la gauche
b22		Interrupteur pour le dégagement du chariot vers la droite
b23		Amorce le dégagement après la course double

Tableau (I-9) : Récapitulatif des capteurs

### II-3-8- Les signalisations (voyants) :

Les voyants ont pour rôle d'informer l'opérateur de l'état du système.

On retrouve dans le tableau (I-10) tous les voyants qui se trouvent sur la machine.

Nomenclature	Type	Rôle
h1	Voyant lumineux	Machine sous tension
h2	Voyant lumineux	Le doigt diviseur dépasse la pointe de la dent
h3	Voyant lumineux	Le doigt diviseur est appuyé contre la fraise

Tableau (I-10) : Récapitulatif des voyants

### III- Description du fonctionnement à partir du schéma électrique

Une fois que l'on a appuyé sur le bouton b 1 « pompe hydraulique », le dispositif de commande est prêt à servir. Avec le sélecteur b 13, on peut choisir un des programmes suivants :

#### 1) Sélecteur placé sur « Régler »

Dans cette position du sélecteur, on peut faire les réglages et les configurations nécessaires pour l'un des trois modes de fonctionnement avant de sélectionner la position correspondante sur le même sélecteur.

#### 2) Sélecteur placé sur « Affûtage périphérique avec va-et-vient et basculement automatique »

Avant que le sélecteur ne soit amené sur cette position, il faut amener le support basculant soit contre la butée 50 à gauche ou 51 à droite (après avoir placé le sélecteur sur « Régler ») afin que le programme puisse se dérouler automatiquement.

- Dans cette position, on actionne le mode automatique.
- Pour déclencher le programme « affûtage périphérique avec basculement automatique » il faut placer le commutateur b 10 sur la position II et le commutateur b 12 sur la position « affûtage périphérique à droite » ou « affûtage périphérique à gauche », qui déclenche la rotation de la fraise.
- Après cela il faut appuyer sur le bouton b 7 « Va-et-vient I » qui déclenche l'avance du chariot en position de travail, actionne à retardement le mouvement de basculement automatique du chariot.

- Si le support basculant est au début contre la butée de droite. le mouvement de basculement vers la gauche sera enclenché.
- Le support basculant s'en va jusqu'en butée à gauche. le mouvement de basculement ralentit. Puis le support basculant vient pousser lentement le fin de course b 20. Ce contact fait déclencher le mouvement de basculement dans le sens inverse.

Le mouvement de va-et-vient se poursuit jusqu'à ce que le bouton va-et-vient 0 soit appuyé.

### **3) Sélecteur placé sur « Avance du chariot, mouvement de va-et-vient, dégagement du chariot et avance dent par dent sans basculement ».**

Il est recommandé d'amener le plateau basculant auparavant contre la butée de droite ou de gauche afin qu'il soit appuyé sur la butée pendant le déroulement du programme.

- Si l'on pousse le bouton « Va-et-vient I », le chariot sera renvoyé en position de travail et son mouvement de va-et-vient sera déclenché.
- La durée du mouvement de va-et-vient est réglée par les butées du chariot.
- Une fois le chariot arrive en butée, le mouvement de dégagement sera enclenché.
- Le dégagement du chariot provoque la rotation de la pièce dans un sens ou dans l'autre.
- Du fait de cette rotation, le doigt de butée (doigt diviseur) se trouve dévié par la lame et commande la marche en sens inverse de la pièce, de sorte que le doigt diviseur soit appuyé contre la fraise.
- La lampe témoin h 3 indique si le doigt diviseur est dévié par la dent de la fraise.
- La lampe témoin h 2 montre si le doigt diviseur est appuyé contre la fraise.
- Le chariot porte-meule avance de nouveau en position de travail.
- Cette opération se répète jusqu'à ce que l'on appuie sur le bouton b 6 « Va-et-vient O ».

### **4) Sélecteur placé sur « Avance du chariot porte meule, va-et-vient, basculement, redressement, dégagement du chariot et avance dent par dent en fonctionnement automatique.**

Avant de placer le sélecteur sur cette position, on amène le plateau basculant à droite ou à gauche, contre l'une des butées 50 ou 51.

- Si l'on pousse le bouton « Va-et-vient I » le chariot sera renvoyé en position de travail et son mouvement de va-et-vient sera déclenché et le basculement automatique se fera à retardement.
- Si le support basculant est au début contre la butée de droite, le mouvement de basculement vers la gauche sera enclenchée.

- Sur le pupitre de commande se trouve le commutateur b 11 à l'aide duquel on peut faire déclencher le mouvement d'avance dent par dent vers la droite ou la gauche, lorsque le plateau basculant est en butée à gauche ou à droite.
- Nous admettons que ce commutateur est placé vers la gauche, le plateau basculant dépasse la butée gauche, le mouvement de basculement sera ralenti. Puis le support basculant vient pousser lentement le fin de course b 20. Ce contact fait déclencher le mouvement de basculement dans le sens inverse.
- L'inversion du basculement provoque le dégagement du chariot porte-meule et arrête le mouvement de va-et-vient.
- En dégageant, le chariot porte-meule, l'avance dent par dent s'effectue comme il a été décrit.
- Lorsque l'avance d'une dent est terminée, le doigt diviseur est appliqué sur la fraise, ce qui provoque le mouvement de basculement automatique.
- Le plateau revient dans la position de départ de sorte que l'opération peut se reproduire.
- Dans les machines où la division se fait en sautant des dents, on a disposé comme le montre le schéma, d'un relais temporisé d 23 supplémentaire à l'avant du pupitre de commande. Ce relais permet de retarder le moment où la dent vient contre le doigt de division. Ceci permet de sauter une ou deux dents lors de l'avance dent par dent.

### **IV Conclusion :**

Les solutions câblées à relais ne sont envisageables que pour de petites applications (une dizaine d'E/S TOR) avec de faibles besoins de communication, au-delà et ce qui est le cas de notre machine, cette technologie montre plusieurs inconvénients :

- Difficulté de situer une éventuelle panne, un dysfonctionnement d'un simple contact peut causer l'arrêt de la machine pour une longue durée.
- Un grand arbre de défaillance, implique une multitude de pannes.
- Un dialogue homme-machine très limité.
- Manque de flexibilité car la machine est limitée aux tâches qu'elle est destinée à faire.

L'analyse du cahier des charges va nous permettre d'élaborer le Grafcet qui est un outil de modélisation très puissant.

## **Chapitre II**

# **MODELISATION DU FONCTIONNEMENT DE LA MACHINE**

### I- Introduction

Une conception d'un système automatisé industriel comporte un cahier des charges, établi en collaboration avec les différents services utilisant ce système. Outre les contraintes techniques, il comporte des instructions impératives reliant la partie commande à la partie opérative, ainsi que le dialogue avec l'opérateur.

La conception, l'étude et la réalisation de d'automatisme nécessite une démarche structurale qui fait appel à un outil de description des systèmes automatisés séquentiels dans l'ordre chronologique des étapes tels que : l'organigramme, le GRAFCET ...

Afin de modéliser notre système industriel, nous avons choisi d'utiliser le GRAFCET qui est considéré comme un outil simple, permettant la modélisation parfaite du système en tenant compte des contraintes physiques et logiques du fonctionnement.

### II- Définition du GRAFCET

Le GRAFCET (**GRA**phe **F**onctionnel de **C**ommande par **E**tapes et **T**ransitions) est un diagramme fonctionnel, dont le but est de décrire graphiquement les différentes évolutions d'un automatisme. Il est parfois simple à utiliser et rigoureux sur le plan formel, et constitue un unique outil de dialogue entre toutes les personnes collaborant à la conception, à l'utilisation ou à la maintenance de la machine à automatiser.

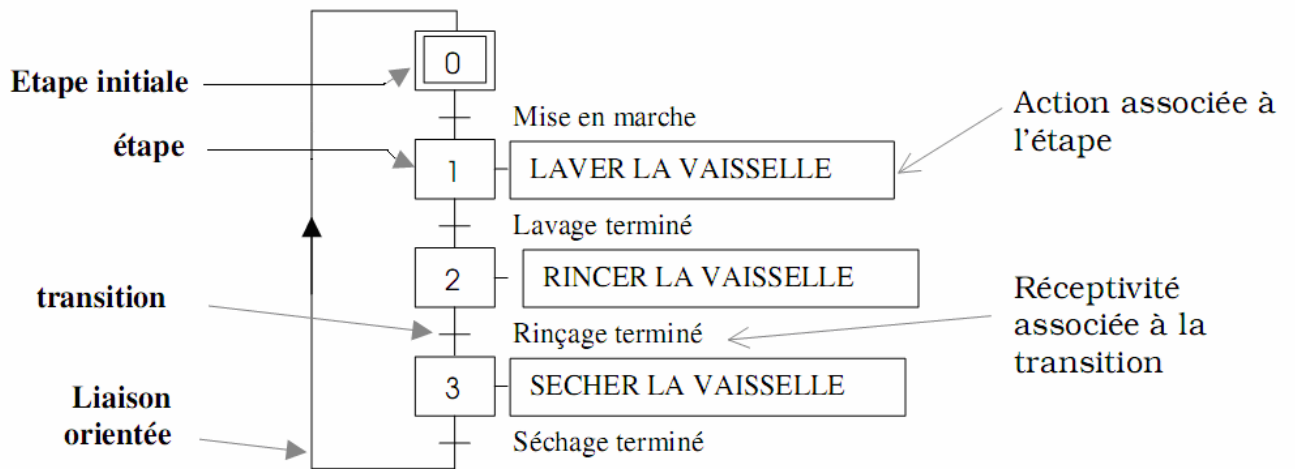
Le GRAFCET permet de construire des modèles ayant une structure graphique à laquelle on associe une interprétation. De plus, ce modèle possède un comportement dicté par des règles d'évolution, complétées pour l'implantation par des algorithmes d'application de ces règles.

### III- Structure du GRAFCET

La structure du modèle de description GRAFCET est composée essentiellement des éléments graphiques de bases suivants :

- Les étapes auxquelles sont associées des actions.
- Les transitions auxquelles sont associées des réceptivités.
- Les liaisons orientées reliant entre étapes et transitions.

La figure (II-1) montre les éléments de base d'un GRAFCET.



La figure (II-1) : Symbolisation d'un grafcet.

### III-1- Etape

Une étape est une situation dans laquelle les variables d'entrée et de sortie de la partie commande restent inchangées. Autrement dit, une étape représente un état dans lequel les informations d'entrée et les informations de sorties de la partie commande restent identiques à elles mêmes. L'étape est représentée par un carré repéré numériquement.

Les **actions** associées sont marquées en clair dans un rectangle à droite du carré représentant à l'étape. Elles sont réalisées lorsque l'étape associée à l'action est active. Il est possible de définir des actions continues, conditionnelles, temporisées, mémorisées et impulsionnelles,...

La situation initiale d'un système automatisé est indiquée par une étape dite étape initiale et représentée par un carré double

### III-2- Transition

Elle est située entre deux étapes consécutives, son franchissement indique l'évolution d'une étape vers l'étape suivante du système.

### **Réceptivité :**

Une réceptivité est associée à chaque transition, c'est la condition qui détermine la possibilité ou non de l'évolution du système par cette transition. Une réceptivité s'exprime comme étant une expression booléenne ou numérique.

### **Temporisation :**

La temporisation est une réceptivité qui permet une prise en compte du temps, elle implique l'utilisation d'un temporisateur. Ce genre de réceptivité est noté comme suit :  $T/ X_i/ t$ , où  $i$  est le numéro de l'étape comportant l'action de la temporisation, et  $t$  est la durée d'activation de l'étape  $X_i$ .

### **III-3- Liaisons orientées**

Les liaisons (arcs) indiquent les voies d'évolution du GRAFCET. Elles relient soit une étape à une transition, soit une transition à une étape.

Les arcs sont implicitement orientés du haut vers le bas. Si un arc est orienté du bas vers le haut il doit porter une flèche qui l'indique.

### **III-4- Règles d'évolution du GRAFCET**

Le GRAFCET permet de déterminer les évolutions dynamiques de n'importe quel système logique (qu'il soit concret ou abstrait). Il est normalisé, et son fonctionnement est régi par cinq règles d'évolution. Si une des règles n'est pas respectée, le graphe n'est pas un GRAFCET.

L'alternance étape - transition et transition - étape doit toujours être respectée quelle que soit la séquence parcourue.

#### **• 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 à l'ensemble des étapes actives au début du fonctionnement. Elle traduit généralement un comportement de repos.

Dans un grafcet, il doit y avoir au moins une étape initiale.

- **Règle 2 : Franchissement d'une transition**

Une transition est franchissable et obligatoirement franchie si les deux conditions suivantes sont remplies :

- ✓ La transition est validée (toutes les étapes d'entrée de cette transition sont actives).
- ✓ La réceptivité associée à cette transition est vraie.

- **Règle 3 : Evolution de la situation**

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

**Remarque :**

La durée de franchissement d'une transition est considéré comme infiniment petite mais non nulle.

- **Règle 4 : Evolutions simultanées**

Plusieurs transitions simultanément franchissables sont simultanément franchies.

- **Règle 5 : Activation et désactivation simultanées d'une étape**

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

### III-5- Sélection de séquence et séquence simultanée

Le GRAFCET présente deux structures particulières :

- ✓ La sélection de séquences.
- ✓ Les séquences simultanées.

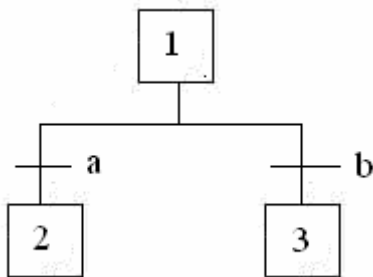
**a- Sélection de séquences**

**-Séquence :** On appelle séquence tout ensemble d'étapes successives où chaque étape est suivie d'une seule transition et chaque transition n'est validée que par une seule étape.

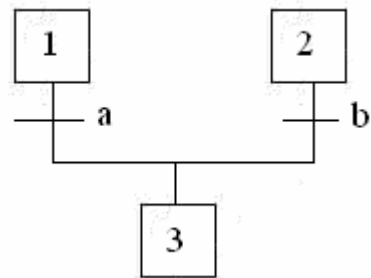
**-Sélection de séquence :** elle représente une alternative d'évolution vers plusieurs étapes. Les réceptivités associées aux transitions d'un aiguillage doivent être exclusives (leur ET logique doit être nul).

La sélection de séquence est aussi appelée divergent OU où divergence de sélection de séquences (figure II-4-a).

La convergence de plusieurs séquences est appelée convergent OU où convergence de sélection de séquences (figure II-4-b).



a- Divergence en OU.



b- Convergence en OU.

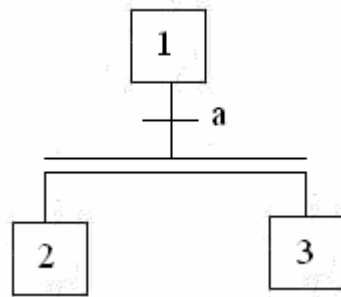
Figure (II-4) : Représentation graphique d'une sélection de séquences.

**b-Séquences simultanées**

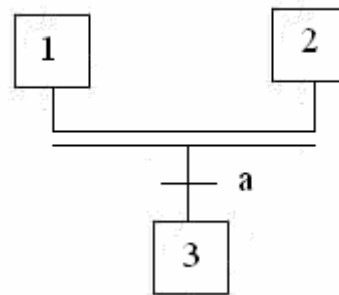
C'est un ensemble de séquences pouvant évoluer indépendamment, à partir du franchissement d'une transition activant simultanément plusieurs étapes.

Une transition qui possède plusieurs étapes de sortie représente l'exécution en parallèle de plusieurs séquences. On appelle cette structure divergent ET ou divergence de séquences simultanées Figure (II-5-a).

Une transition qui possède plusieurs étapes d'entrée représente la synchronisation de plusieurs séquences. On appelle cette structure convergent ET ou convergence de séquences simultanées Figure (II-5-b).



a- Divergence en ET



b- Convergence en ET

Figure (II-5) : Représentation graphique d'une séquence simultanée.

### III-6 Saut d'étapes

Le saut d'une ou plusieurs étapes est réalisé lorsque les actions associées à ces étapes deviennent inutiles Figure (II-6)

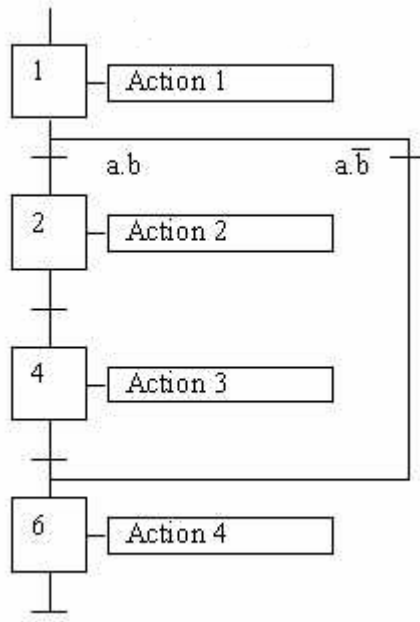


Figure (II-6) : Saut d'étape.

### III-7- Reprise de séquence

La reprise de séquence permet de répéter une même séquence jusqu'à ce que la réceptivité de fin de séquence soit vraie Figure (II-7).

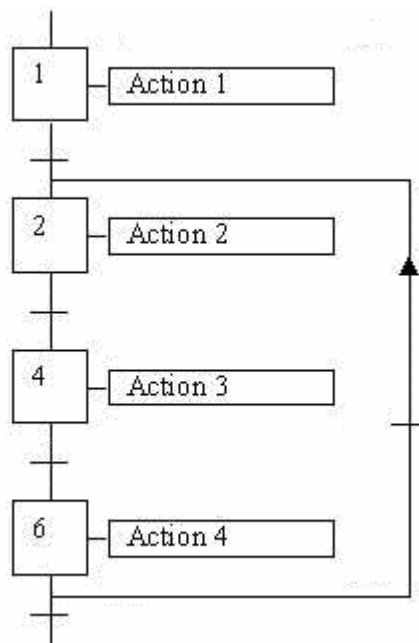


Figure (II-7) : Reprise de séquence.

#### IV- Niveau d'un Grafcet

##### IV-1- Grafcet niveau 1

Il est 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 indépendamment de la technologie utilisée. Les réceptivités sont décrites en mots et non en abréviations, on associe le verbe à l'infinitif pour les actions. Ici on regarde les pré-actionneurs : distributeurs, contacteurs ...et les capteurs qui réalisent les réceptivités Figure (II-8-a).

##### IV-2- Grafcet niveau 2

Appelé aussi niveau de la partie opérative, il tient compte de plus de détails des actionneurs, et des capteurs, la représentation des actions et réceptivités est écrite en abréviation et non en mot, on associe une lettre majuscule à l'action et lettre minuscule à la réceptivité

Figure (II-8-b).

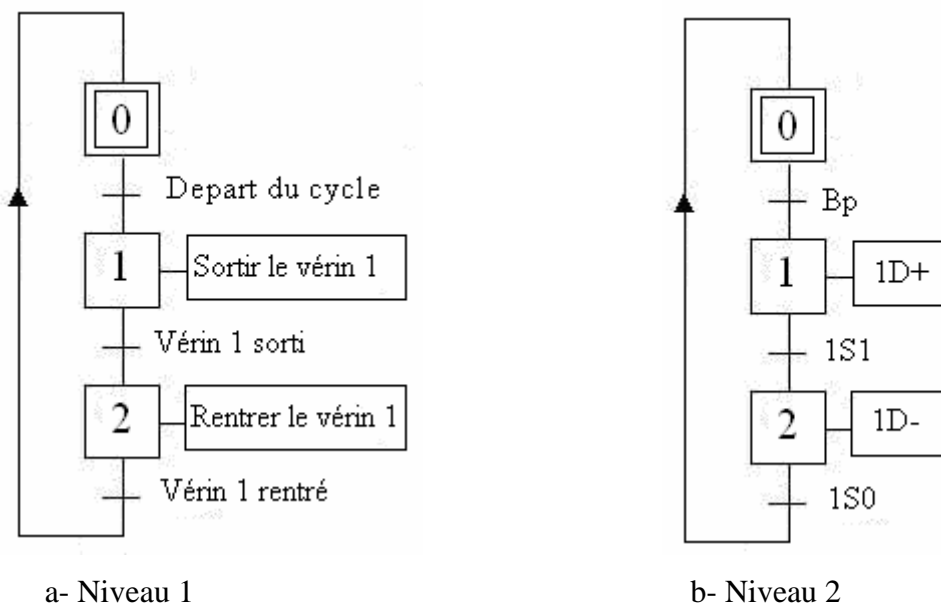
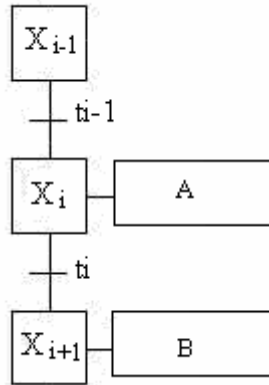


Figure (II-8) : Les niveaux de GRAFCET

**V- Mise en équation d'un Grafcet**

Soit le grafcet de la figure suivante :



L'état d'une étape  $X_i$  peut être noté comme suit :

- $X_i = 1$  si l'étape est active.
- $X_i = 0$  si l'étape est inactive.

La réceptivité qui est une variable binaire a pour valeur :

- $t_i = 1$  si la réceptivité est vraie.
- $t_i = 0$  si la réceptivité est fausse.

Soit la variable d'arrêt d'urgence dur (AUD) et d'arrêt d'urgence doux (Aud) tel que :

- AUD = 1 désactivation de toutes les étapes.
- Aud = 1 désactivation des actions, les étapes restent actives.

On la définit l'étape initiale comme suit :

- Init = 1 initialisation du grafcet (mode d'arrêt).
- Init = 0 déroulement du cycle (mode marche).

- Equation d'une étape initiale :

$$X_i = (CAX_i + CDX_i * X_i + Init) * \overline{AUD}$$

Avec :

$$CAX_i = (X_{i-1} * t_{i-1} + Init) * AUD$$

$$CDX_i = (X_{i+1} * \overline{Init}) + AUD$$

CAX<sub>i</sub> : sont les conditions d'activation de l'étape i.

CDX<sub>i</sub> : sont les conditions de désactivation de l'étape i.

- Equation d'une étape non initiale :

$$X_i = (CAX_i + CDX_i * X_i) * Init * \overline{AUD}$$

Avec :

$$CAX_i = X_{i-1} * t_{i-1} * Init * AUD$$

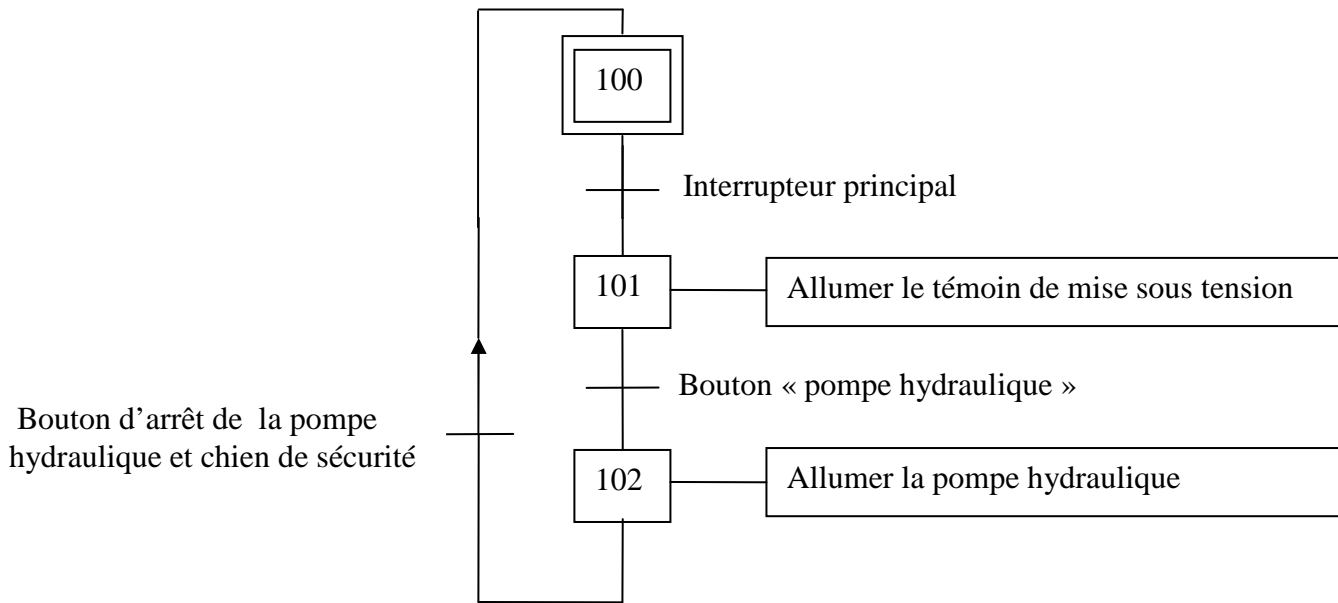
$$CDX_i = X_{i+1} * Init * AUD$$

- Equation des actions

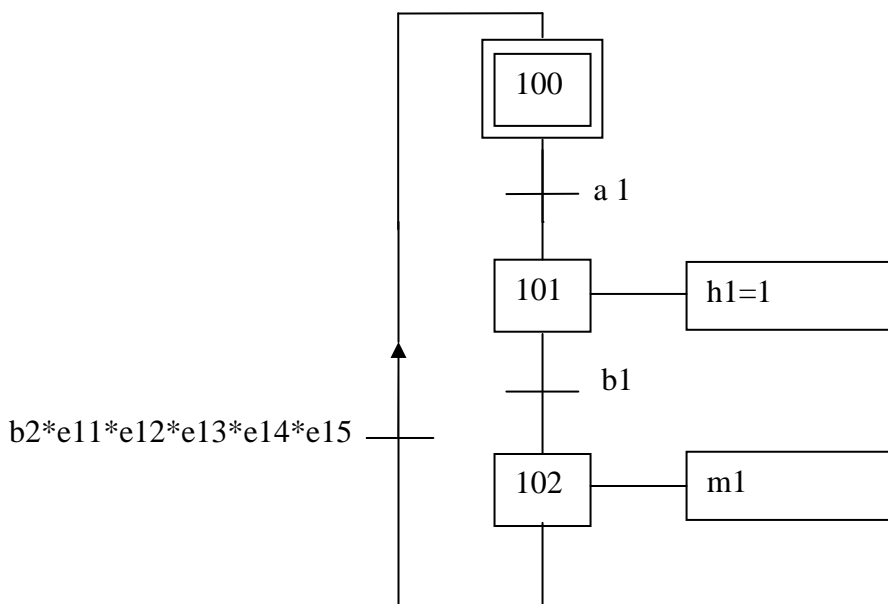
$$A = X_i * AUD$$

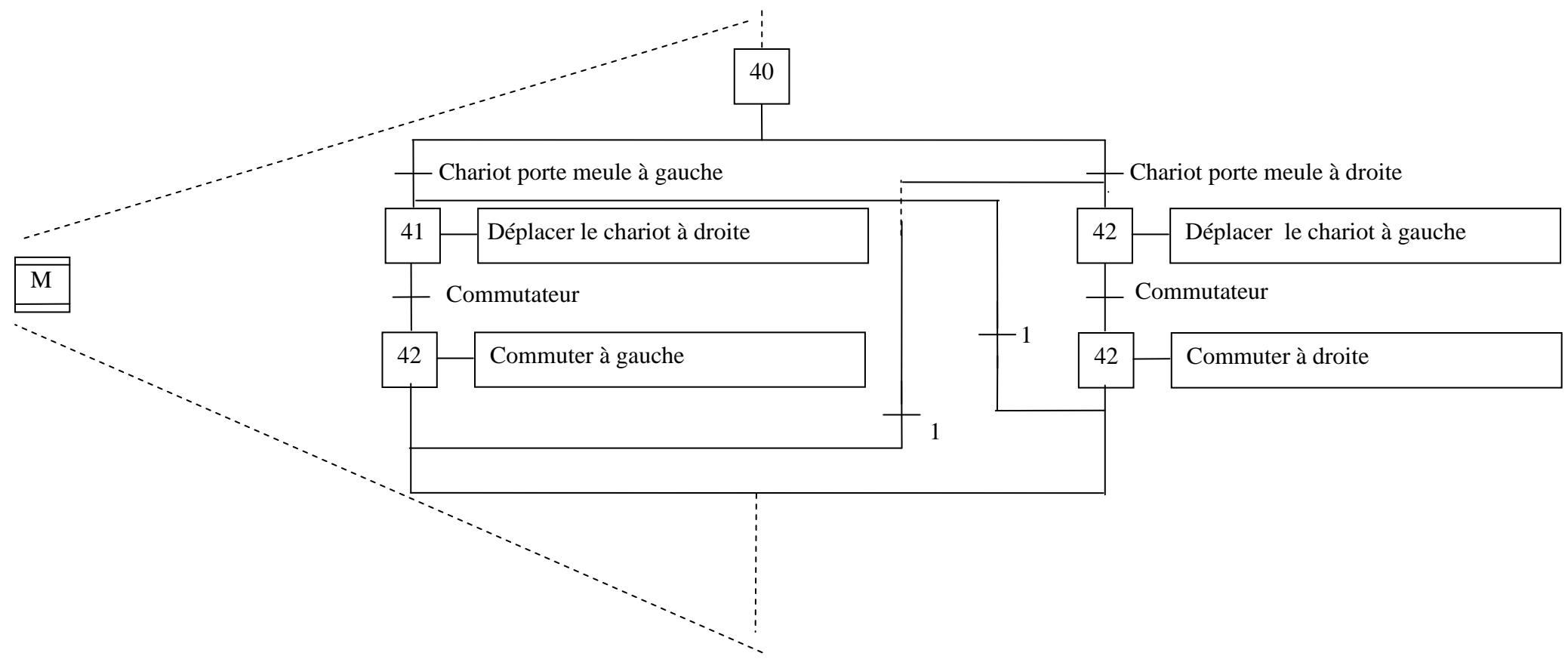
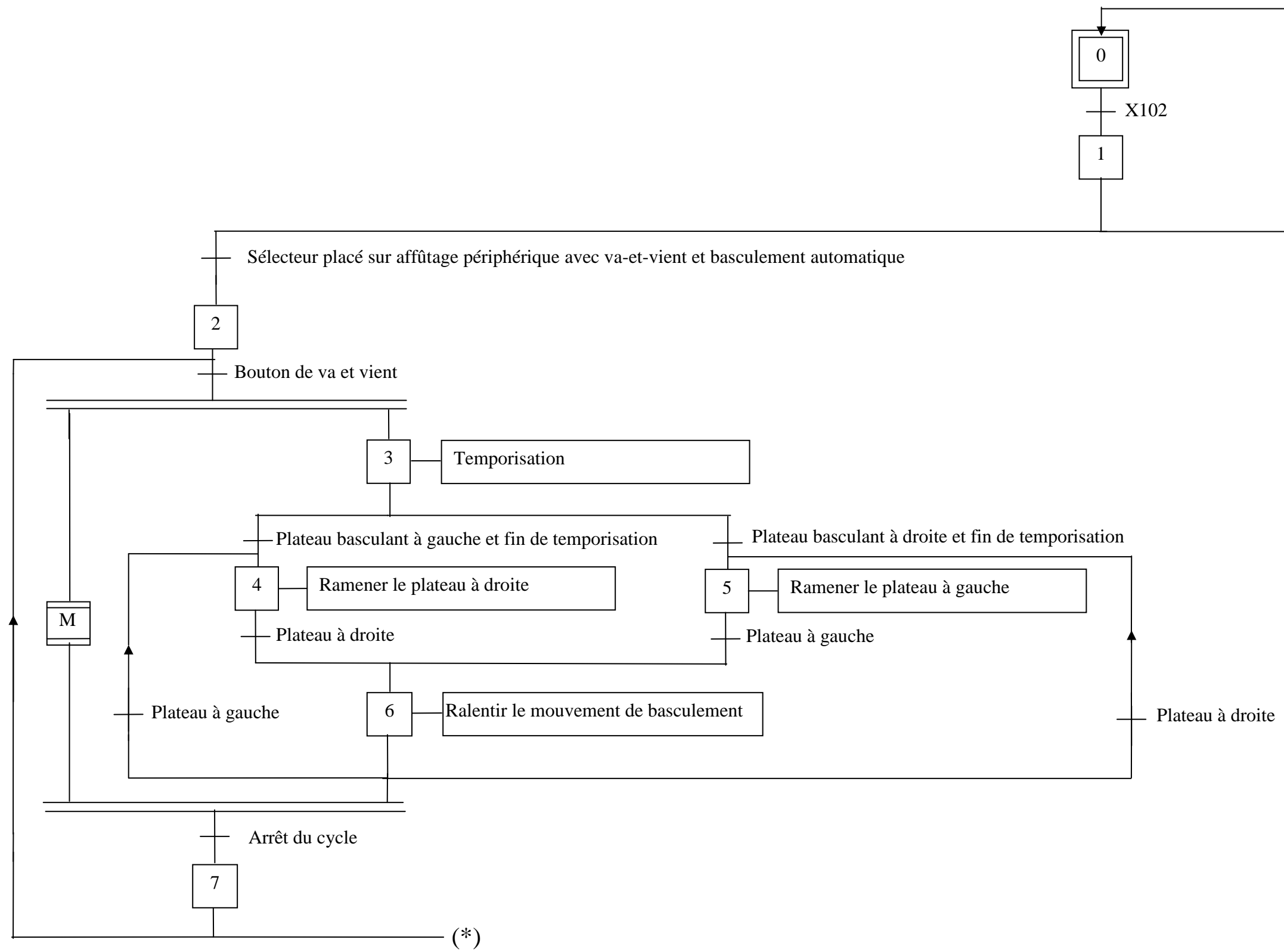
## VI- application du GRAFCET pour modéliser la machine

### VI- 1 GRAFCET de sécurité niveau 1



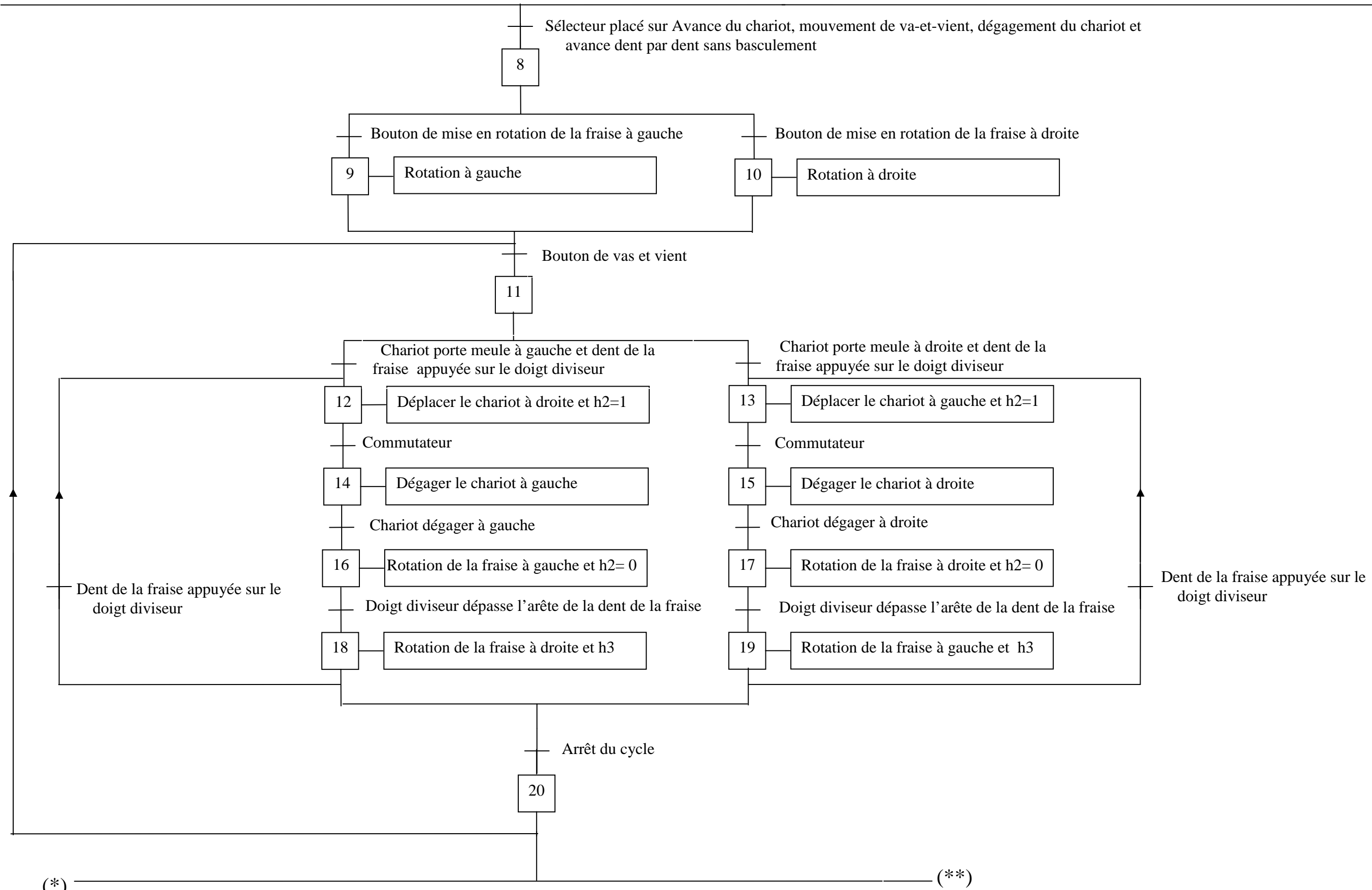
**IV- 2 GRAFCET de sécurité niveau 2**





(\*\*\*)

(\*\*\*)



(\*\*\*)



(\*\*)

(\*\*\*\*)

0

X102

1

b13\_1

2

b7

3

T1

$(t/X4/5s)*b20$

4

S5\_2

$b19*\overline{b20}$

$(t/X4/5s)*b19$

5

S5\_1

$b20*\overline{b19}$

$(t/X7/3s)*b20$

6

S4

T2

$(t/X7/3s)*b19$

b6

7

(\*)

40

b22

41

V\_d

b23

43

D\_d

b21

42

V\_g

b23

44

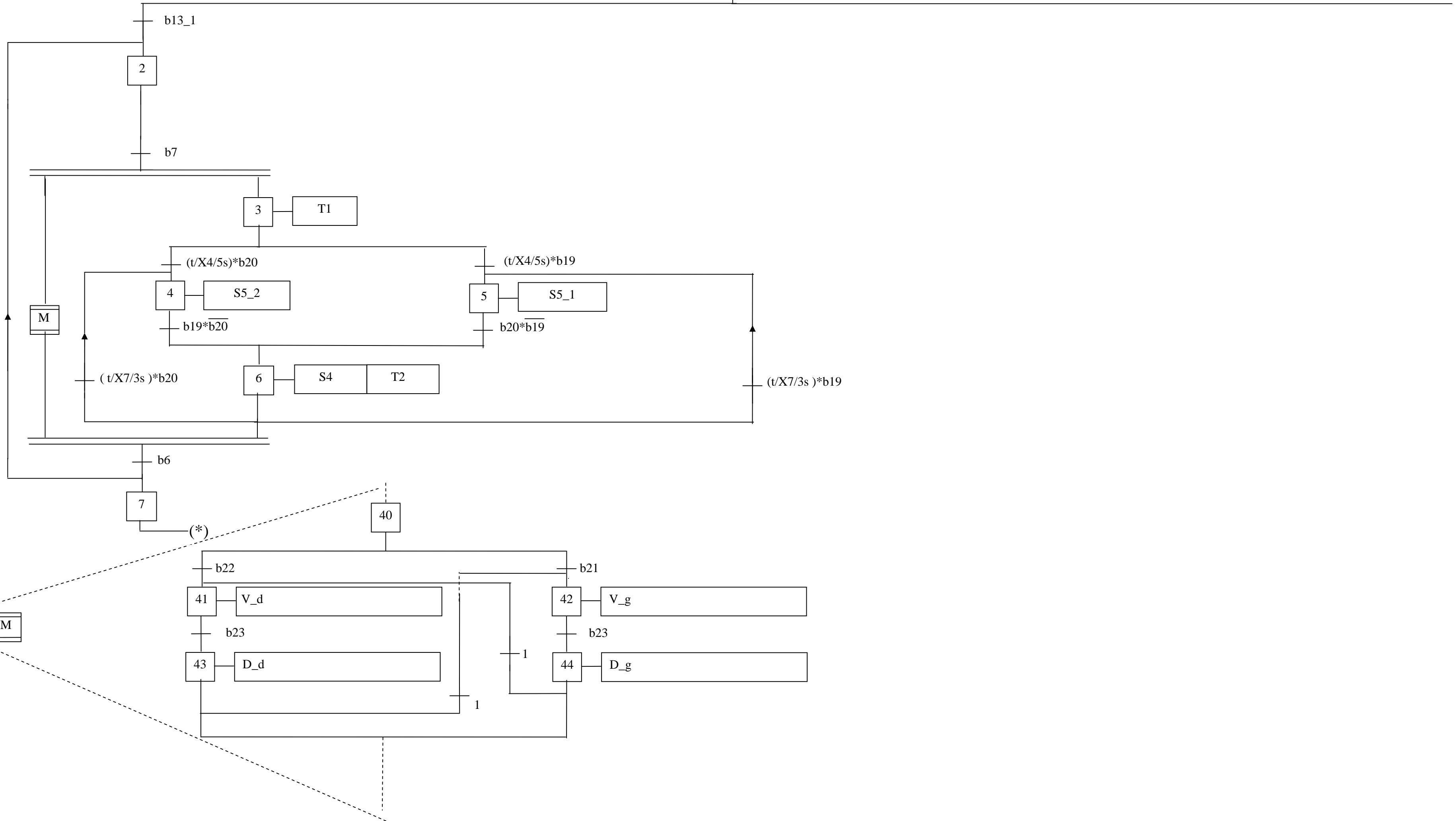
D\_g

1

1

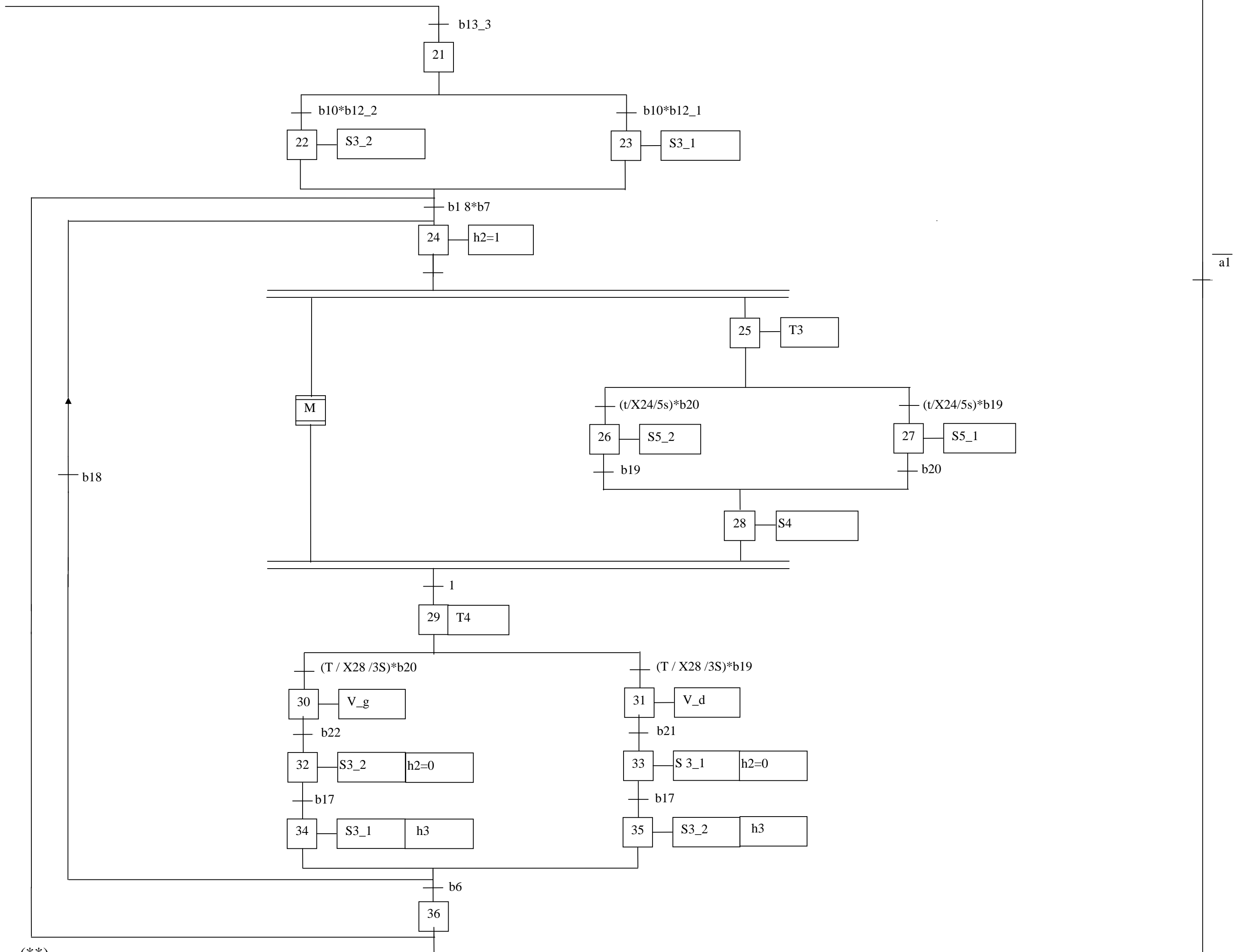
M

M





(\*\*\*)



a1

(\*\*)

## **Chapitre III**

# **AUTOMATISATION DE LA MACHINE PAR L'AUTOMATE PROGRAMMABLE INDUSTRIEL S7-300**

## **I- Introduction**

Les automates programmables industriels (API) sont apparus à la fin des années soixante, à la demande de l'industrie automobile américaine (GM), qui réclamait plus d'adaptabilité de leurs systèmes de commande.

Depuis le début des années 80, l'intégration des automates programmables pour le contrôle des différents processus industriels est plus qu'indispensable. A l'origine, l'automate programmable a été considéré comme une machine séquentielle, capable de suppléer des automatismes réalisés en logique traditionnelle, en apportant toutefois de profonds bouleversements dans la manière de concevoir et d'organiser le contrôle d'un processus.

L'intégration de l'automate programmable renforce le degré de fiabilité de l'équipement et offre une très grande adaptabilité face aux évolutions de l'environnement.

## **II- Automates programmables industriels API**

### **II-1- Définition**

Un automate Programmable Industriel (API) est une machine électrique programmable, adapté à l'environnement industriel et destiné à piloter des procédés. Son fonctionnement est défini par programme, il donne des ordres aux préactionneurs de la partie opérative à partir des données d'entrées (capteurs, détecteurs ...), rend des comptes en permanence de son état et dialogue avec l'opérateur et le processus. L'automate peut traiter :

- Des commandes de type logique, séquentiel, et analogique.
- Des fonctions de calcul arithmétique ; temporisation, comptage, comparaison.
- Des liaisons avec d'autres appareils (imprimantes, calculateurs ...)

Comme il peut aussi réaliser des fonctions de régulation.

### **II-2- Architecture et gammes d'automates**

Un automate programmable est constitué de plusieurs éléments. Quelque soit la taille et la puissance de calcul de la machine, l'architecture est similaire.

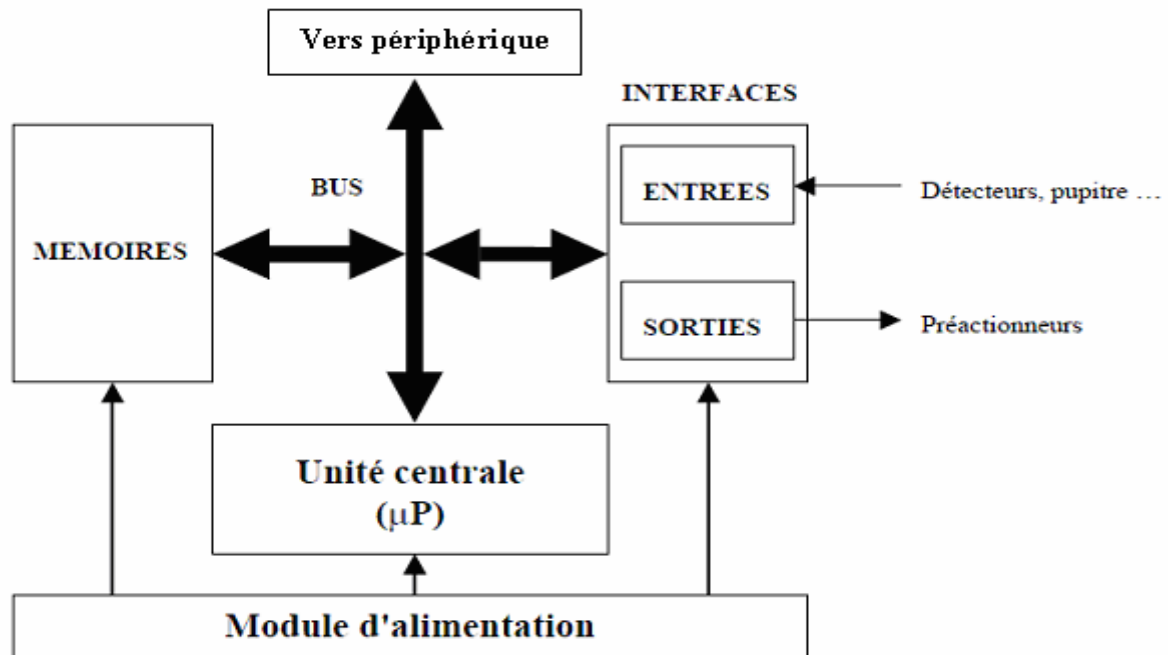


Figure III-1 : Architecture interne d'un automate programmable

- L'élément central est l'unité de traitement arithmétique et logique (CPU) qui effectue les séquences de programme et les calculs.
- Les programmes sont enregistrés dans une mémoire qui garde l'information même quand l'alimentation électrique est coupée. Une autre mémoire est dédiée au stockage des données, cette partie de la mémoire peut être volatile.
- Les entrées-sorties sont les liens entre l'automate et son environnement. Leur type dépend des caractéristiques du signal qu'elles doivent capter ou gérer : tout ou rien (digitales) pour les signaux binaires, analogiques pour les signaux de mesure ou de consigne.
- Une alimentation pour les circuits électriques internes. Celle-ci est galvaniquement isolée des circuits de commande.
- Des interfaces de communication (Xcom) qui servent à l'échange d'informations numériques avec le monde extérieur par bus de terrain ou réseaux informatiques. Une de ces interfaces est utilisée pour charger le programme dans la mémoire.

### II-3- Définition d'un système automatisé

Un système de production est dit automatisé, lorsqu'il peut gérer de manière autonome un cycle de travail préétabli qui se décompose en séquences ou étapes.

## II-4- Structure générale d'un système

Les systèmes automatisés utilisés dans le domaine de l'industrie, possèdent une structure de base identique.

Ils sont constitués de plusieurs parties plus ou moins complexes reliés entre elles (voir figure III-2) :

- partie opérative (PO)
- partie commande (PC)
- partie contrôle

### Description des différentes parties

#### La partie opérative (PO)

C'est la partie visible du système. Elle comporte les éléments mécaniques du mécanisme avec :

- **Des prés actionneurs** (distributeurs, contacteurs ...), qui reçoivent des ordres de la partie commande.
- **Des actionneurs** (vérins, moteurs, ...) qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie électrique, pneumatique (air comprimé) ou hydraulique (huile sous pression) en énergie mécanique.
- **D'une détection** (capteurs) qui informe la partie commande de l'exécution de travail.

Dans un SAP (système de production automatisé), le détecteur représente le service de surveillance et de renseignement de mécanisme. Il contrôle, mesure, surveille et informe la Partie Commande sur l'évolution du système.

#### La partie commande (PC)

Ce secteur de l'automatisme gère dans une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs situés dans la Partie Opérative, et restitue vers cette même direction des prés actionneurs (distributeurs).

#### La partie contrôle (P R)

Sa complexité et sa taille dépendent de l'importance du système. Il résume les différentes commandes nécessaires au bon fonctionnement de l'automatisme : marche, arrêt, arrêt d'urgence, marche automatique, ...

La figure ci dessous montre le schéma général d'un automatisme.

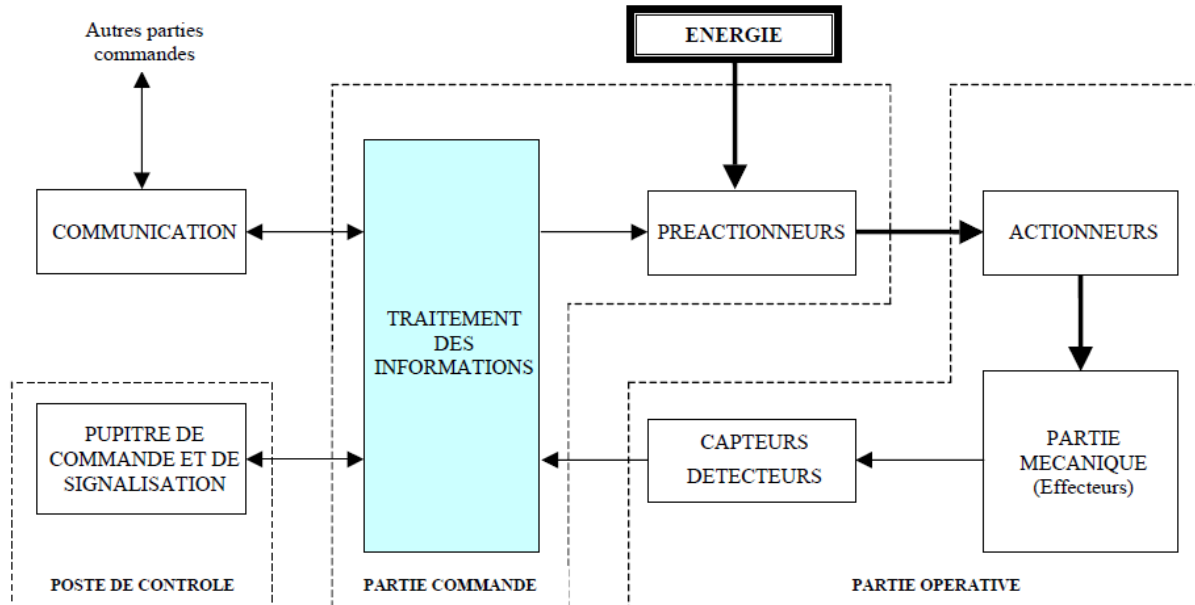


Figure III-2 : Structure d'un système automatisé

Un échange d'informations s'effectue continuellement entre les trois parties de cette structure, permettant le bon déroulement du cycle de fonctionnement défini dans le cahier des charges.

## II-5 Choix de l'automate

Différents critères et contraintes définissant le cahier des charges de notre problème d'automatisation, l'automate choisi doit répondre donc aux critères suivants:

- La capacité de traitement du processeur.
- Le nombre d'entrées/sorties.
- La nature des entrées/sorties (analogiques, digitales ou booléennes).
- La fiabilité.
- La qualité du service après vente.

### **III Présentation de l'automate S7-300**

L'automate S7-300 est constitué d'une alimentation, de modules d'entrée et de modules de sortie. A ceux-ci peuvent s'ajouter des processus de communication et des modules de fonctions qui se chargeront des fonctions spéciales, tel que la commande d'un moteur pas à pas, par exemple.

Comme tout API, l'automate S7-300 est un système d'automatisation modulaire offrant la gamme des modules suivants :

- Module d'alimentation (PS) avec un courant de 2A, 5A ou 10A.
- Unité centrale (CPU314).
- Modules d'entrée et de sortie digitale (SM : ETOR et SM : STOR).
- .Module coupleur IM servant d'extension pour un autre rack.
- Module de fonction (FM) pour réaliser des tâches spéciales.
- Processeur de communication (CP) pour la connection en réseau.

#### **III-1- Caractéristiques de l'API S7-300**

Le S7-300 présente les caractéristiques suivantes.

- Variété de la gamme de CPUs.
- Différent types de modules permettant de réaliser des tâches d'automatisation plus efficaces.
- Possibilité d'exécution jusqu'à 32 module.
- Possibilité d'appartenance en réseau avec les options MPI, PROFIBUS, INDUSTRIAL ETHERNET.
- Raccordement central de la PG avec accès à tous les modules.
- Configuration et paramétrage à l'aide de l'outil configuration matériel.

#### **III- Différents parties de l'API S7-300**

Elles sont montrées dans la figure III-3.

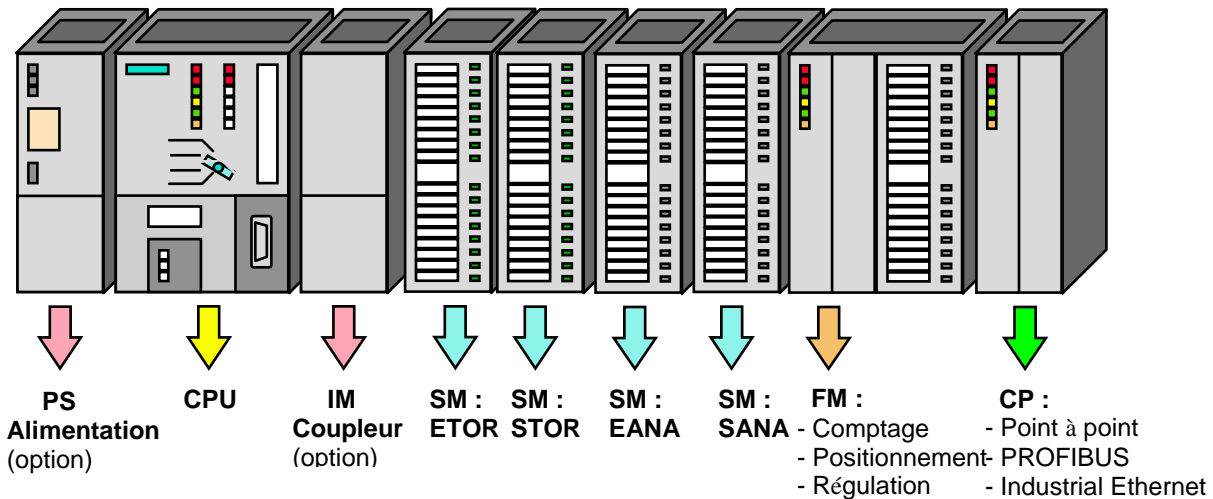


Figure III-3: Les différents composants de l'automate programmable

### III-2-1 Module d'alimentation

L'API S7- 300 nécessite une tension de 24Vcc. Le module d'alimentation assure cette condition en transformant la tension du secteur 380/220Vca en 24Vcc. Il permet de sauvegarder le contenu de la RAM au moyen de la pile de sauvegarde ou d'une alimentation externe. Sur le module d'alimentation nous trouvons l'interrupteur de mise sous tension de l'automate.

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

### III-2-2 L'unité centrale

La CPU est le centre des opérations arithmétiques et logiques de l'automate. Elle reçoit les informations sur l'état des entrées, exécute le programme et met à jour les sorties. Elle permet le réglage du comportement au démarrage et le diagnostic de défauts par des voyants lumineux.

L'utilisateur peut choisir une des différentes gammes de CPU offertes par le constructeur selon les performances désirées.

### III-2-3 Les modules d'entrée/sortie

#### III-2-3-a Modules d'entrée

Il existe deux types d'entrées :

- ✓ Les modules d'entrées **Tout ou Rien**

Un module d'entrées Tout ou Rien permet à l'unité centrale d'effectuer une lecture de l'état logique des capteurs qui lui sont reliés et de le matérialiser par un bit image de l'état du capteur.

✓ **Les modules d'entrées analogiques**

Les modules d'entrées analogiques permettent de :

- Sélectionner la gamme d'entrées de chaque voie.
- Scruter des voies d'entrées par le multiplexage et l'acquisition des valeurs.
- Faire une conversion analogique/numérique des mesures d'entrées.

**III-2-3-b Les modules de sortie**

Il existe deux types de modules de sorties :

✓ **Les modules de sortie Tout Ou Rien**

Les modules de sortie Tout Ou Rien permettent à l'automate programmable d'agir sur les pré actionneurs ou de signaler à l'opérateur.

✓ **les modules de sortie analogique**

Les différentes fonctions du module de sortie analogique sont :

- La sélection de la gamme pour chaque sortie
- La conversion numérique / analogique

**III-2-4 Les coupleurs**

Les coupleurs permettent à l'automate de communiquer avec le milieu extérieur (console, imprimantes ...) ou de le relier avec d'autres automates.

**III-2-5 Modules de fonction (FM)**

Ces modules permettent les fonctions suivantes : le comptage, la régulation, le positionnement.

**III-3 Fonctionnement de l'API**

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

### **III-3-1 Réception des informations sur les états du système**

L'automate reçoit des informations sur l'état du processus par les capteurs reliés aux entrées. Il va mettre à jour l'image des entrées au début de chaque cycle de programme, en transformant le nouvel état des signaux d'entrées des modules vers l'image des entrées, ce qui permet à la CPU de connaître l'état du processus.

### **III-3-2 Système d'exploitation**

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

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

### **III-3-3 Exécution du programme utilisateur**

Après avoir acquis les informations d'entrée, exécuté le système d'exploitation, la CPU passe à l'exécution de programme utilisateur, qui contient la liste d'instructions du programme de fonctionnement du processus. Il est composé essentiellement de blocs de données de code et blocs d'organisation.

### **III-3-4 Commande de processus**

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

## **III-4 Nature des informations traitées par l'API**

Les informations peuvent être de type :

- Tout Ou Rien : l'information peut prendre l'état 0 ou 1.

- Analogique : l'information varie à l'intérieur d'une plage prédéfinie. C'est le cas la température, pression ...
- Numérique : l'information est contenue dans des mots codés sous forme binaire ou hexadécimale. C'est le type d'information délivrée par un ordinateur.

#### **IV- Programmation**

Le STEP 7 est le progiciel de base qui permet de configurer et de programmer des problèmes d'automatisation. Il fait partie de l'industrie logicielle de SIMATIC.

STEP 7 s'exécute sous les systèmes d'exploitation de MICROSOFT à partir de la version Windows 95. Le progiciel de base STEP 7 existe sous plusieurs versions.

- STEP 7-Micro/DOS et STEP 7-Micro/Win pour des applications autonomes simples sur SIMATIC S7-200.
- STEP 7 pour des applications sur SIMATIC S7-300/400, SIMATIC M7-300/400 et SIMATIC C7.

Lors de la création d'une solution d'automatisation, STEP 7 permet :

- La création et la gestion des projets.
- La création et le paramétrage du matériel et de la communication.
- La gestion des mnémoniques.
- La création des programmes.
- Le chargement du programme dans les systèmes cibles.
- Le test de l'installation d'automatisation.
- Le diagnostic lors de la perturbation dans l'installation.

#### **IV-1 Création d'une solution d'automatisation dans STEP 7**

Pour concevoir un projet avec STEP 7, il existe deux approches, qui sont résumées par le plan donné par la figure III-4 :

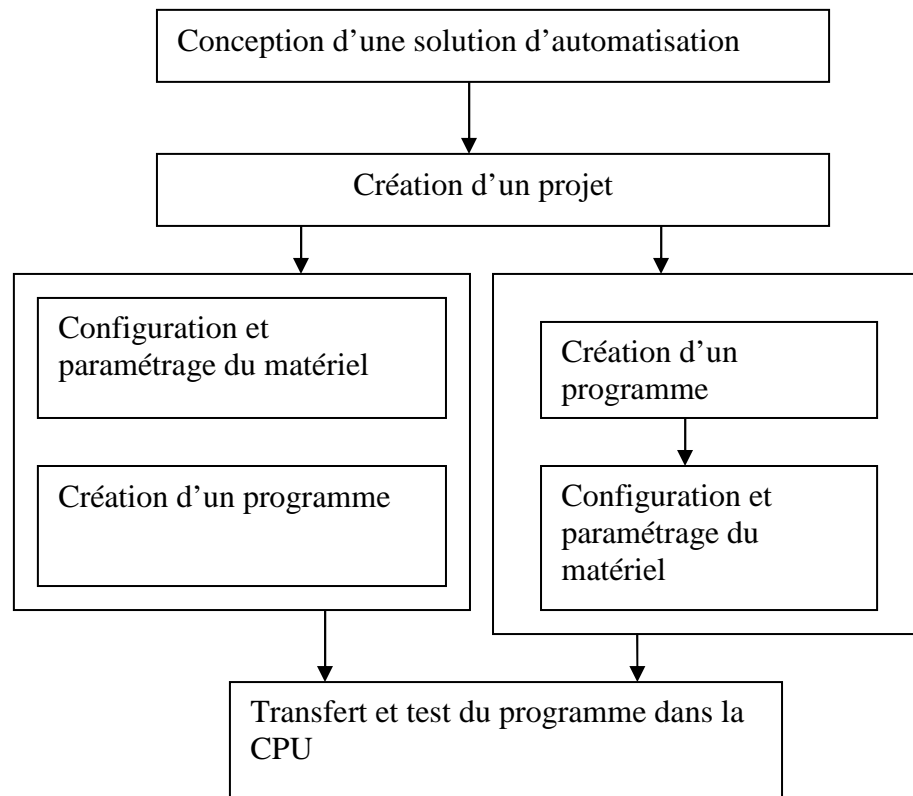


Figure III-4 : Démarche de programmation avec STEP 7

## VI-2 Les langages de programmation

Les langages de base proposés dans l'éditeur du programme du logiciel STEP 7 sont : le CONT, le LIST et le LOG.

### VI-2-1 Le langage contact (CONT)

Le langage CONT (langage contact) ou LADDER est un langage dont la logique est inspirée des réseaux électriques.

Le CONT est un langage graphique, dont la syntaxe des instructions est issue des schémas à relais. CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts (à fermeture et à ouverture), les éléments complexes et les bobines.

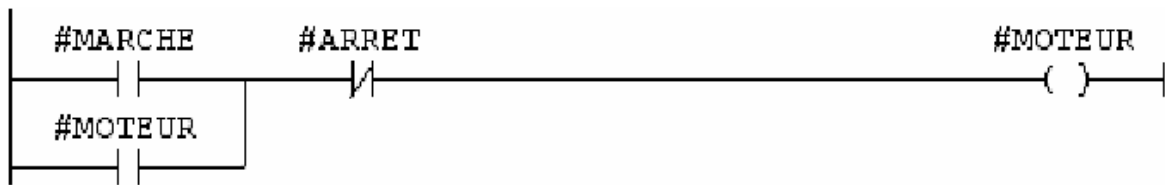


Figure III-5 : exemple de programme CONT

#### IV.2.2. Logigramme (LOG)

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

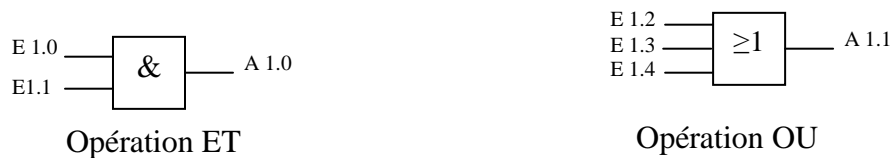


Figure III-6: Représentation des opérations ET, OU par LOG

..

#### IV.2.3. Liste d'instructions (LIST)

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

<b>U E 1.0</b>	<b>O E 1.2</b>
<b>U E 1.1</b>	<b>O E 1.3</b>
<b>= A 1.0</b>	<b>ON E 1.4</b>
	<b>= A 1.1</b>

Figure III-7: Représentation des opérations ET, OU par LIST

Les programmes d'automatisation programmés en CONT ou LOG sont en principe toujours traduisibles en LIST.

### **IV-3 L'opération de temporisation**

Les temporisations sont utilisées pour retarder le déroulement d'un événement, ou bien pour maintenir un état durant un temps déterminé.

### **IV.4 Création d'un programme utilisateur**

Un programme utilisateur comprend toutes les instructions et déclarations, ainsi que les données nécessaires au traitement de signaux de commande d'une installation ou d'un processus.

Il est affecté à la CPU, et peut être structuré en entités plus petites appelées blocs.

Un programme utilisateur devant être exécuté dans une CPU S7 est essentiellement constitué de blocs. Il contient, en outre, des informations supplémentaires, telles que les données destinées à la configuration ou à la mise en réseau du système. En fonction de l'application, nous pouvons donc créer dans le programme utilisateur, les blocs de types suivants :

- Blocs d'organisations (OB)
- Blocs fonctionnels (FB)
- Fonctions (FC)
- Blocs de données (DB)

Des blocs souvent utilisés en relevant du système tel que les blocs fonctionnels système (SFB) et les fonctions systèmes (SFC) sont intégrées au logiciel.

#### **IV.4.1. Lancement du logiciel**

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

#### **IV.4.2. Création du projet**

L'assistant étant lancé, cette première fenêtre s'ouvre à l'écran. C'est la fenêtre d'introduction de l'assistant.

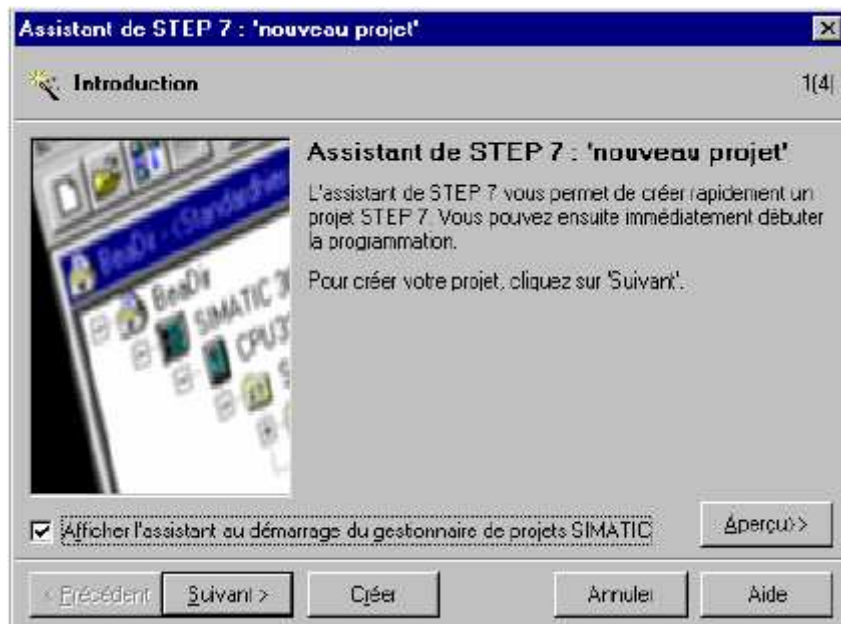


Figure III-8: Fenêtre introduction.

Nous cliquons sur *Suivant*.

Cette seconde fenêtre prend la place de la fenêtre précédente, elle nous permet de choisir la CPU, pour notre projet on choisira la CPU314. La figure IV-8 illustre le choix de la CPU.

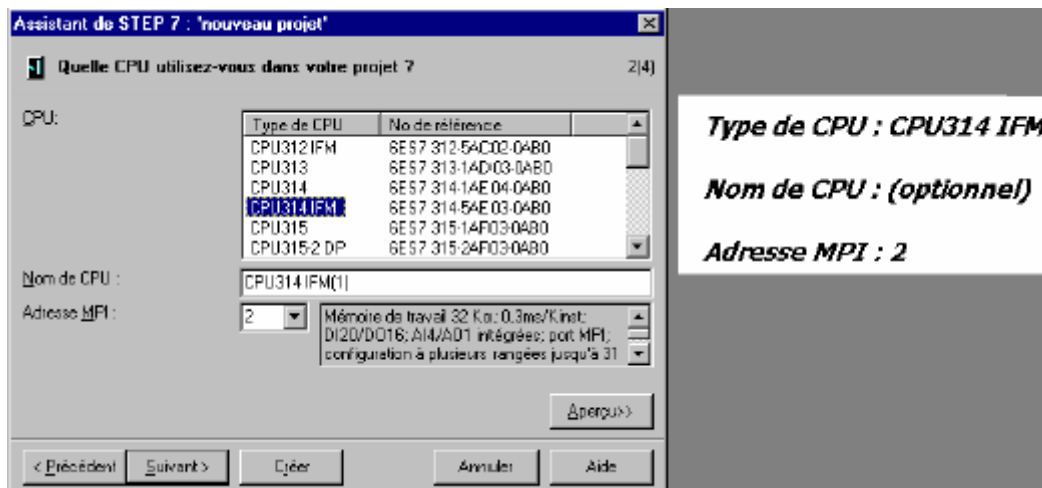


Figure III-9: Choix de la CPU.

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

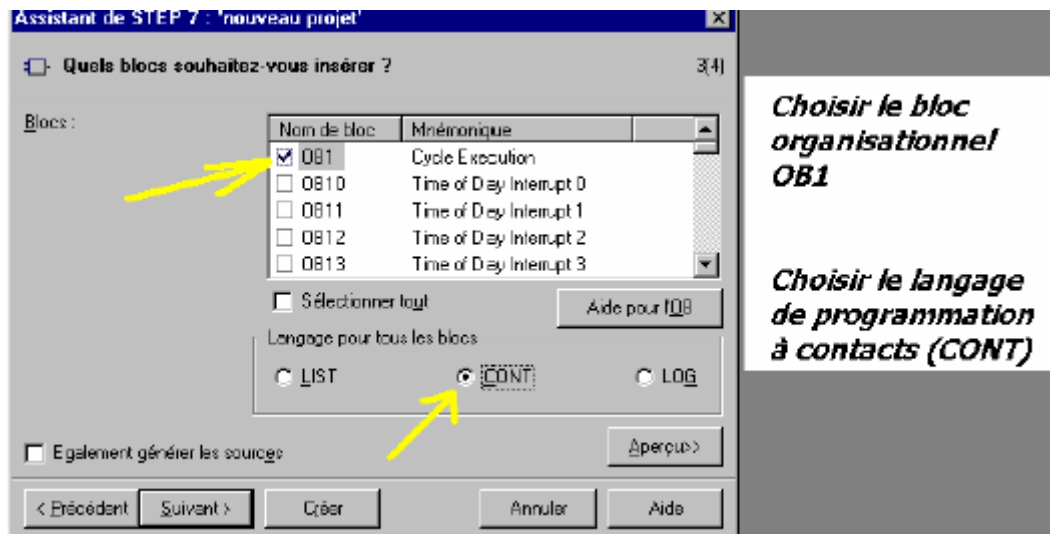


Figure III-10: Choix du bloc et du langage de programmation.

Dans notre projet, nous utiliserons le bloc OB1. Ce bloc contient le programme qui sera continuellement exécuté par l'automate et le langage à contact.

En cliquant sur suivant, la création de projet apparaît pour le nommer comme illustré dans la figure III-11.

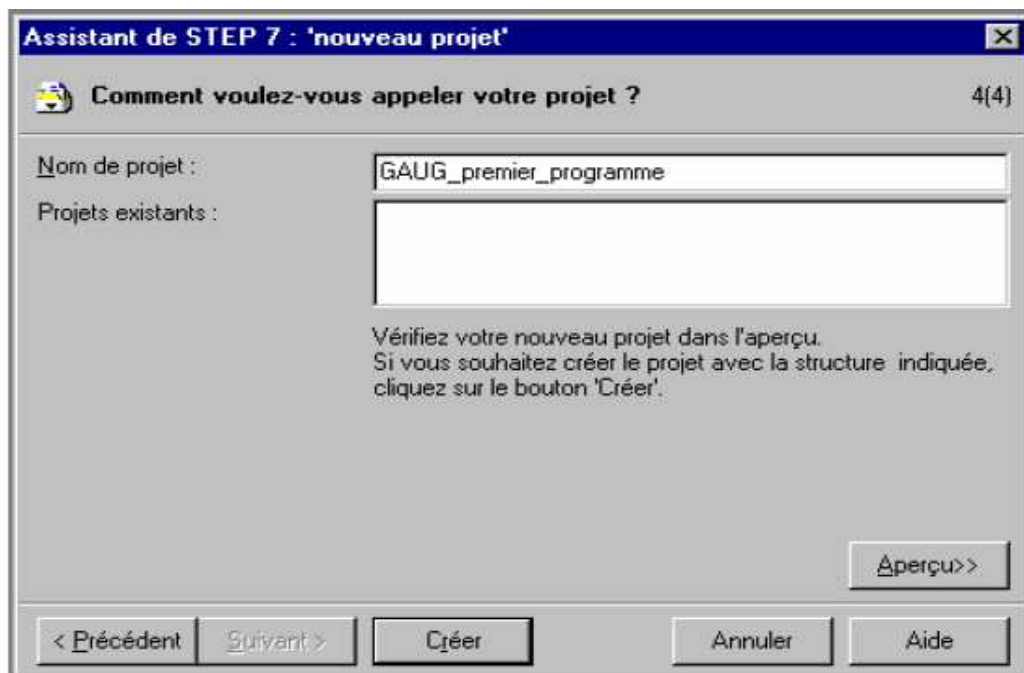


Figure III-11: Nommer projet.

#### IV.4.3. Création de la table des mnémoniques

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

L'objet "mnémoniques" (table des mnémoniques) est automatiquement créé sous un programme S7.

Pour pouvoir remplacer les données globales par des mnémoniques dans un programme, nous devons les affecter dans la table des mnémoniques. On procède de la manière suivante :

1. Nous cliquons deux fois sur le programme S7 dans la fenêtre du projet afin que l'objet "mnémoniques" s'affiche dans la partie droite de la fenêtre.
2. Si la table des mnémoniques a été effacée, nous pouvons insérer une nouvelle en choisissant la commande **insertion>Table des mnémoniques**.
3. Nous ouvrons l'objet "mnémoniques", par exemple nous cliquons deux fois dessus.

Dans la fenêtre qui s'ouvre, nous pouvons éditer la table des mnémoniques.

#### V Configuration et paramétrage du matériel

##### ➤ Configuration :

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

STEP 7 affecte automatiquement une adresse à chaque module dans la table de configuration.

##### Paramétrage :

Par "paramétrage", nous entendons dans ce qui suit :

- Le réglage des paramètres des modules paramétrables pour la configuration centralisée et pour un réseau. Exemple : une CPU est un module paramétrable. La surveillance du temps de cycle est un paramètre qu'on peut définir.
- La définition des paramètres de BUS, des maîtres et d'esclaves pour un réseau maître.

Ces paramètres sont chargés dans la CPU qui les transmet aux modules correspondants. Il est très facile de remplacer des modules car les paramètres définis avec STEP 7 sont automatiquement chargés dans le nouveau module à la mise en route.

Nous donnons dans la figure III-12 suivante un exemple de configuration de l'automate S7-300.

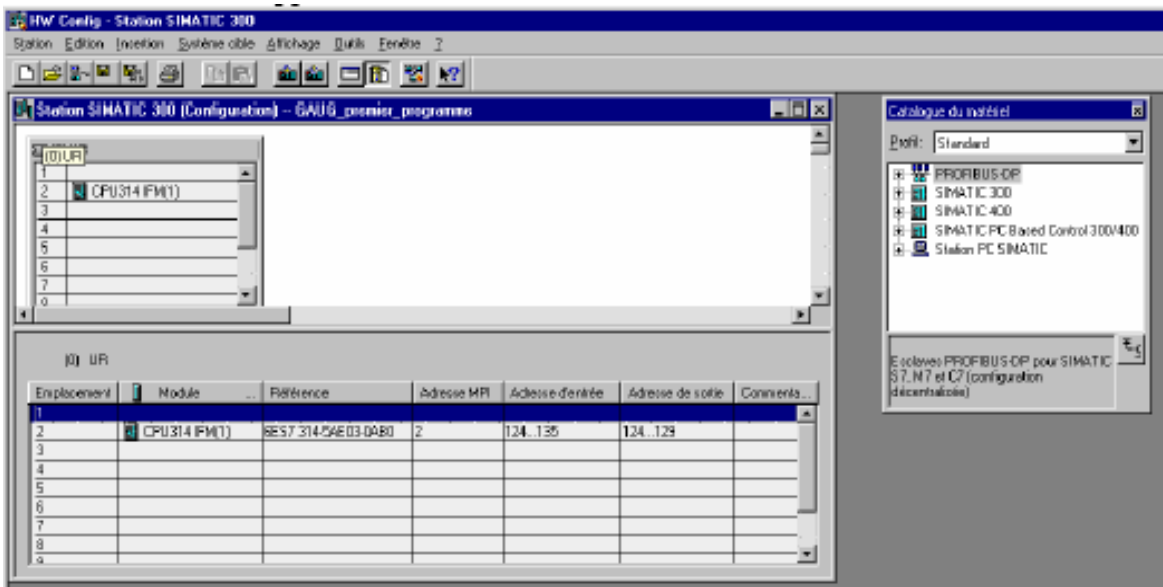
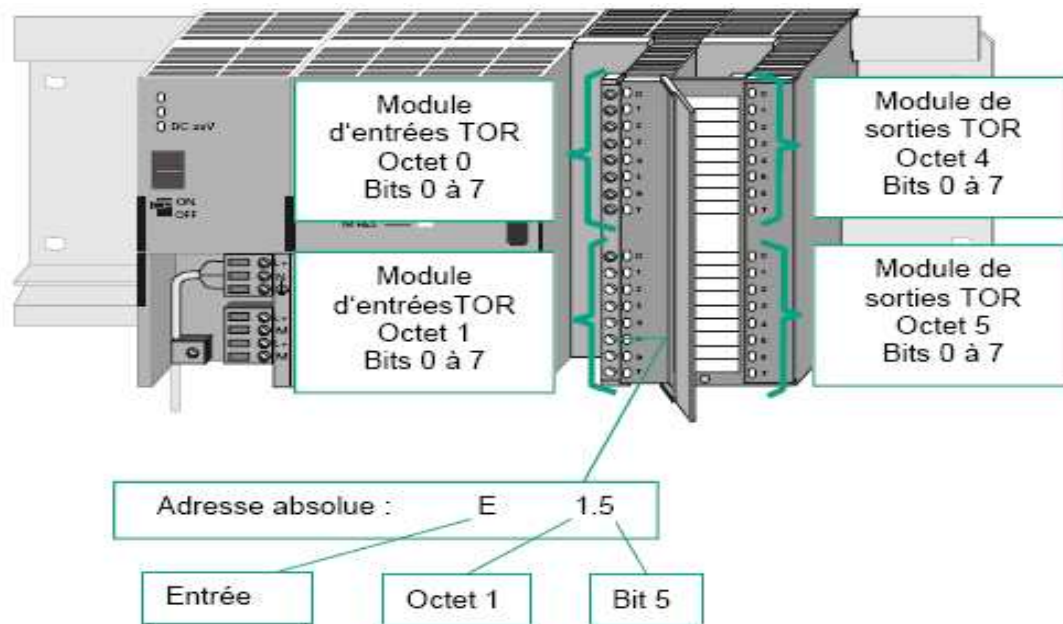


Figure III-12: configuration de l'automate S7-300.

### V.1 Adressage des signaux d'entrées/sorties

La déclaration d'une entrée ou sortie donnée à l'intérieur du programme s'appelle l'adressage. Les entrées/sorties des automates sont regroupés en groupe de huit entrées ou sorties TOR. Cette unité de huit entrées ou sorties est appelée un octet. Chaque groupe reçoit un numéro que l'on appelle l'adresse octet.

Afin de permettre l'adressage d'une entrée ou sortie à l'intérieur d'un octet, il est divisé en huit bits. Ces derniers sont numérotés de 0 à 7. Nous obtenons ainsi l'adresse du bit.



FigureIII-13: Adressage d'entrées/sorties (TOR)

## V.2 Mémentos

Les mémentos sont utilisés pour les opérations internes à l'automate pour lesquelles l'émission d'un signal n'est pas nécessaire. Chaque automate programmable dispose d'une grande quantité de mémentos (S7-315 dispose de 2048 bites de mémentos).

## V.3 Traitement du programme par l'automate

Le traitement du programme dans l'automate est cyclique, il se déroule comme suit :

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

## **VI. Principe de conception d'une structure de programme**

- Au cours de l'exécution du programme dans la CPU, deux programmes différents s'exécutent, le système d'exploitation et le programme utilisateur.

### **VI.1. Système d'exploitation**

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

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

### **VI.2. Programme utilisateur**

Après avoir créé le programme utilisateur (contient toutes les fonctions nécessaires au traitement de la tâche d'automatisation spécifique) et le charger dans la CPU. Il doit :

- Déterminer les conditions pour le démarrage et le redémarrage de la CPU (par exemple, initialiser des signaux).
- Traiter des données de processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques, définir des signaux binaires pour la sortie, écrire des valeurs analogiques).
- Traiter les perturbations dans le déroulement normal du programme.

### VI.3 Blocs dans le programme utilisateur

Le logiciel de programmation STEP 7 nous permet de subdiviser le procédé à automatiser en différentes parties autonomes. Il en résulte les avantages suivants :

- Ecriture de programmes importants et clairs.
- Simplification de l'organisation du programme.
- Modification facile du programme.
- Simplification du test du programme, car nous pouvons l'exécuter section par section.
- Facilité de la mise en service.

#### VI.3.1. Bloc utilisateur

Le STEP 7 offre les blocs utilisateur suivants pour la programmation structurée :

- **Bloc d'organisation (OB) :** Un OB est appelé cycliquement par le système d'exploitation et constitue donc l'interface entre le programme utilisateur et le système d'exploitation. L'OB contient des instructions d'appels de bloc indiquant à l'unité de commande de l'automate l'ordre dans lequel il doit traiter les blocs.
- **Bloc fonctionnel (FB) :** Le FB dispose d'une zone de mémoire qui lui est affectée en propre. Il est possible d'affecter un bloc de données (DB) au FB à l'appel du bloc. Il est possible d'accéder aux données du DB d'instance via les appels contenus dans le FB. Nous pouvons affecter plusieurs DB à un FB. Il est possible d'appeler d'autres FB et FC dans un bloc fonctionnel via des instructions d'appels de bloc.
- **Fonction (FC) :** Une FC ne possède pas une zone mémoire propre. Les données locales d'une fonction sont perdues après l'exécution de la fonction. Il est également possible d'appeler d'autres FB et FC dans une fonction via des instructions d'appels de blocs.
- **Blocs de données (DB) :** les DB sont utilisés pour la mise à disposition d'espace mémoire pour les types de variables de données. Il existe deux types de blocs de données. Les DB globaux dans lesquels tous les OB, FB et FC peuvent lire les données enregistrées ou écrire des données et les DB d'instance qui sont affectées à un FB donné.

### VI.3.2. Blocs système pour fonctions standard et fonction système

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

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

## VII. Traitement du programme par la CPU

Deux traitements du programme sont possibles en fonction de l'unité de commande et de la programmation. Le traitement linéaire, le traitement structuré.

### VII.1. Traitement linéaire du programme

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

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

### VII.2 Traitement structuré du programme

Dans le cas d'un procédé complexe possédant plusieurs fonctions (postes), il est recommandé de partager le programme utilisateur en fonctions principales que l'on programme à l'aide de blocs de codes (OB, FB et FC), d'une manière à faciliter la programmation et le test des parties du programme.

Dans ce cas la CPU exécute le programme principal dans OB1 et fait appel là et quand il le faut aux autres blocs (FC, FB). A la fin de l'exécution du bloc appelé, la CPU revient pour poursuivre le programme appelant.

### **VIII. Différents blocs d'organisation des S7-300**

Les blocs d'organisations (OB) constituent l'interface entre le système d'exploitation de la CPU et le programme utilisateur. Ils nous permettent de déclencher l'exécution de certaines parties de programme telle que la mise en route de la CPU ou bien quand une erreur intervient.

Les différents blocs de l'organisation sont:

#### **a) Bloc d'organisation (OB1) :**

Le système d'exploitation de la CPU S7 exécute l'OB1 de manière cyclique quand la mise en route de la CPU est terminée.

#### **b) Bloc d'organisation pour traitement des erreurs :**

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

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

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

#### **d) Blocs d'organisation pour le programme de mise en route :**

Lors de la mise en route, le système d'exploitation appelle l'OB de mise en route correspondant, à savoir :

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

### **IX. Implantation du GRAFCET dans le S7-300**

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

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

- Analyse et validation du grafcet de notre machine.
- Détermination des conditions d'activation et de désactivation des étapes du grafcet.

- Définition des exigences en matière de sécurité (les alarmes, AUD, aud,...etc.)
- Affectation des mémentos aux variables intermédiaires, et des adresses effectives et formelles aux entrées et sorties.
- Ecriture des équations des étapes.
- Raccordement de toutes les entrées, sorties et la PG à l'automate.
- Après avoir choisi la configuration matérielle et simuler le programme, nous procédons au chargement du projet dans la CPU physique.
- Après validation du projet, nous pouvons commencer la production.

## **IIIX. conclusion**

Nous avons fait une étude de l'automate programmable industriel s7-300 vu qu'il était choisi pour réaliser notre tâche d'automatisation.

Nous avons présenté les différents modules de l'automate et ses langages de programmation.

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

C'est grâce au STEP7 que le lien entre l'utilisateur et l'automate programmable S7-300 existe car cet automate ne peut gerer ses fonctionnalités sans un programme approprié. Mais avant d'élaborer le programme il faut d'abord modéliser notre procédé et cela sera notre tâche.

## **Chapitre VI**

# **PROGRAMMATION ET SIMULATION DU MODELE GRAFCET DE LA MACHINE**

## I- Introduction

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

Pour cela nous avons utilisé le logiciel S7-PLCSIM qui est un logiciel optionnel de STEP7. Cette application de simulation nous permet d'exécuter et de tester notre programme dans un automate programmable.

L'objectif de ce logiciel est le test des programmes STEP 7 pour les automates S7-300 et S7-400 qu'on ne peut pas tester immédiatement sur le matériel et ceci pour différentes raisons. L'application est critique, car elle peut occasionner des dommages matériels ou blessures corporelles en cas d'erreurs de programmation, mais la simulation permet de corriger ces erreurs pendant le test.

## II- Présentation de S7-PLCSIM

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

Le S7-PLCSIM dispose d'une interface simple nous permettant de visualiser et de modifier les différents paramètres utilisés par le programme (comme par exemple d'activer ou de désactiver des entrées). Pendant l'exécution du programme dans la CPU simulée, on a la possibilité de mettre en œuvre les différentes applications du logiciel STEP 7, comme par exemple le test de bloc, afin de visualiser les variables d'entrées et de sorties.

## III- Commande de la CPU

### III-1- La fenêtre CPU



Figure IV-1 : mise en marche de la CPU


- SF : erreur dans le système.
- DP : état de communication avec les E / S décentralisées.
- DC : CPU mise sous tension.
- RUN : CPU en mode de visualisation.
- STOP : visualisation arrêtée.


### III-2- les modes de fonctionnement de la CPU

- Mode STOP : dans ce mode, la CPU n'exécute pas de programme et pour pouvoir changer un programme, la CPU doit être obligatoirement dans ce mode.
- Mode RUN : dans ce mode, la CPU exécute le programme mais on ne peut ni changer de nouveaux programmes ni forcer les variables.
- Mode RUN-P : le programme est exécuté dans la CPU, et il est possible de charger de nouveaux programmes ou blocs et de forcer des variables en utilisant les fonctions du STEP7.
- MRES : Avant de charger le programme dans l'API de simulation, il est recommandé d'effectuer un effacement général de la CPU afin de s'assurer qu'elle ne contient plus d'anciens blocs. L'effacement général doit s'effectuer dans le mode de marche « STOP ».

### III-3- Mise en route du logiciel S7-PLCSIM




Le mode de simulation est disponible à partir du gestionnaire de projets SIMATIC à condition qu'aucune liaison à des API réels ne soit établie. La procédure est la suivante :

- Cliquer sur  ou sélectionner la commande **Outils / Simulation de modules**. Cela lance l'application « S7-PLCSIM » et ouvre une fenêtre « CPU ».

- Dans le gestionnaire de projets SIMATIC, cliquer sur  ou choisir la commande **Système cible / Charger** pour charger les Blocs dans l'API de simulation.

Après avoir exécuté le logiciel « S7-PLCSIM » et chargé les blocs dans le simulateur, on crée des fenêtres d'affectation d'entrées et celles de visualisation comme suit :

- Choisir le menu « CPU » dans S7-PLCSIM et vérifier qu'un point noir (•) apparaît à côté de « **Mettre sous tension** ».
- Choisir la commande **Exécution / Mode d'exécution** et vérifier qu'un point noir (•) apparaît à côté de **Cycle continu**.
- Mettre la CPU de simulation en marche en cliquant sur l'une des cases à cocher « RUN » ou « RUN-P ».

- Cliquer sur  ou sélectionner la commande « **Insertion / Entrée** » pour afficher une fenêtre le module d'entrée «**EB0** » (d'octet 0). Sachant qu'on peut modifier le numéro d'octet « **EB124** » par exemple.
- Cliquer sur  ou sélectionner la commande « **Insertion / Sortie** » pour afficher une seconde fenêtre de module de sortie «**AB0**» (octet de sortie 0). Sachant qu'on peut modifier le numéro d'octet «**AB125** » par exemple.
- Cliquer sur  ou sélectionner la commande **Insertion Temporisation** pour afficher une fenêtre de temporisations « **T0** ».

Choisir le menu CPU dans la fenêtre du S7\_PLCSIM et vérifier que la commande mettre sous tension est activée (figure IV.2).

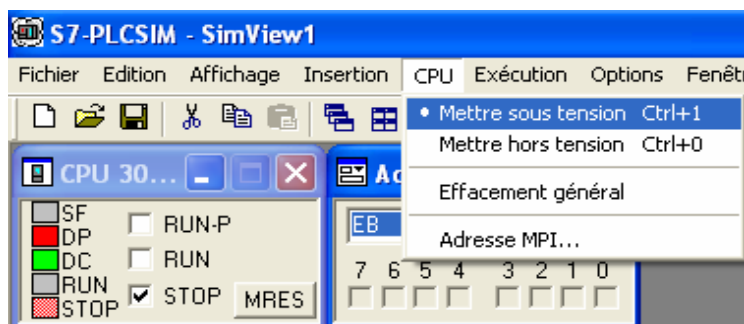


Figure IV.2 : Mise sous tension de la CPU.

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

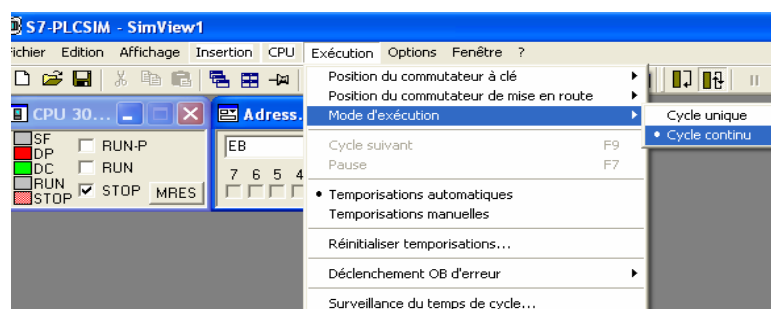


Figure IV.3 : Choix du cycle continu.

Mettre la CPU de simulation en marche en cliquant sur l'une des cases à cocher RUN ou RUN-P (figure IV.4).



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

Une fois toutes les fenêtres d'entrées et de sorties sont prêtes, nous activons les entrées voulues pour lire l'état des sorties (Figure IV.5).

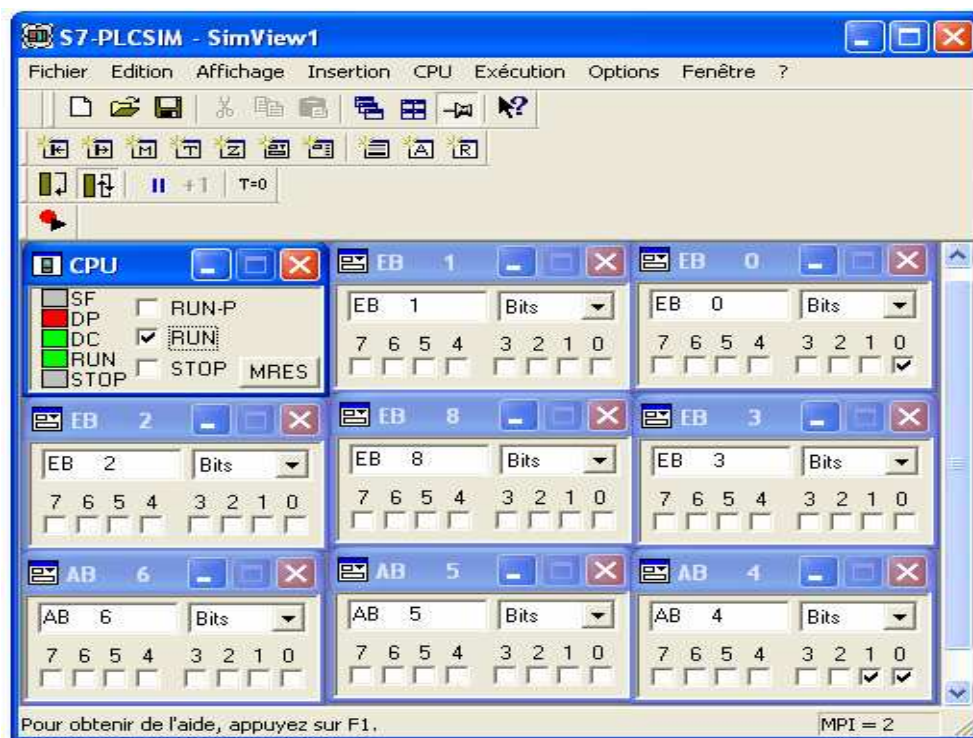



Figure IV.5 : Simulateur du S7-PLCSIM.

#### IV-1 Visualisation de l'état du programme

Après le chargement du programme dans la CPU du simulateur et la mise de cette dernière en mode « RUN » le STEP7 nous permet de visualiser l'état du programme soit en cliquant sur l'icône  ou en sélectionnant la commande **Test/Visualiser**.

## **IV-2 Simulation du programme de l'affûteuse**

La simulation du programme de la machine est faite en deux étapes :

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

## **V Conclusion**

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

Les actions de chaque cycle de fonctionnement de la machine sont programmées dans un FC dans le but de repérer et rendre facile les modifications à apporter.

## Conclusion générale

---

L'étude que nous avons fait rentrer dans le cadre d'un plan de modernisations des automatismes au sein de la Société Nationale des Véhicules Industriels (SNVI), elle a été dédiée pour l'insertion d'un automate programmable en vue de l'automatisation d'une machine à affûter les fraises « WALTER AMS 1000 ». En terme de ce travail, nous avons retenu beaucoup de choses, et cela sur plusieurs volets.

Nous avons pu comprendre la commande réalisé en logique câblée et qui existe sur la machine, et remarquer ces limites et inconvénients.

On a constaté la puissance de l'outil de la modélisation qu'on a choisi pour faire notre tâche et qui est le GRAFCET. Et on a remarque la simplicité de passer d'un modèle ressorti à partir de ce dernier à n'importe quel langage de programmation.

On a acquis une maîtrise du logiciel de programmation des automates programmables industriels en particulier le S7-300 du constructeur SIEMENS.

On a programmé le modèle ressorti du GRAFCET avec le langage STEP7 en faisant appel à un langage évolué qui est le langage CONTACT, avec une configuration matérielle reflétant l'environnement réel.

On s'est permis une confirmation que notre modèle satisfait le cahier des charges du fonctionnement, et cela grâce au logiciel de simulation S7 PLCsim.

Ce stage au niveau de la SNVI nous a permis de nous approcher du milieu professionnel, et d'avoir une idée sur ce monde qui est notre prochaine destination après un long parcours d'études.

Espérant que notre objectif est atteint et que ce modeste travail apportera un plus au domaine et servira de support aux promotions à venir.

**OB1 - <offline>**

"Cycle Execution"

**Nom :**  
**Auteur :**  
**Horodatage Code :**  
**Interface :**  
**Longueur (bloc/code /données locales) :** 00900 00782 00024

**Famille :**  
**Version :** 0.1  
**Version de bloc :** 2

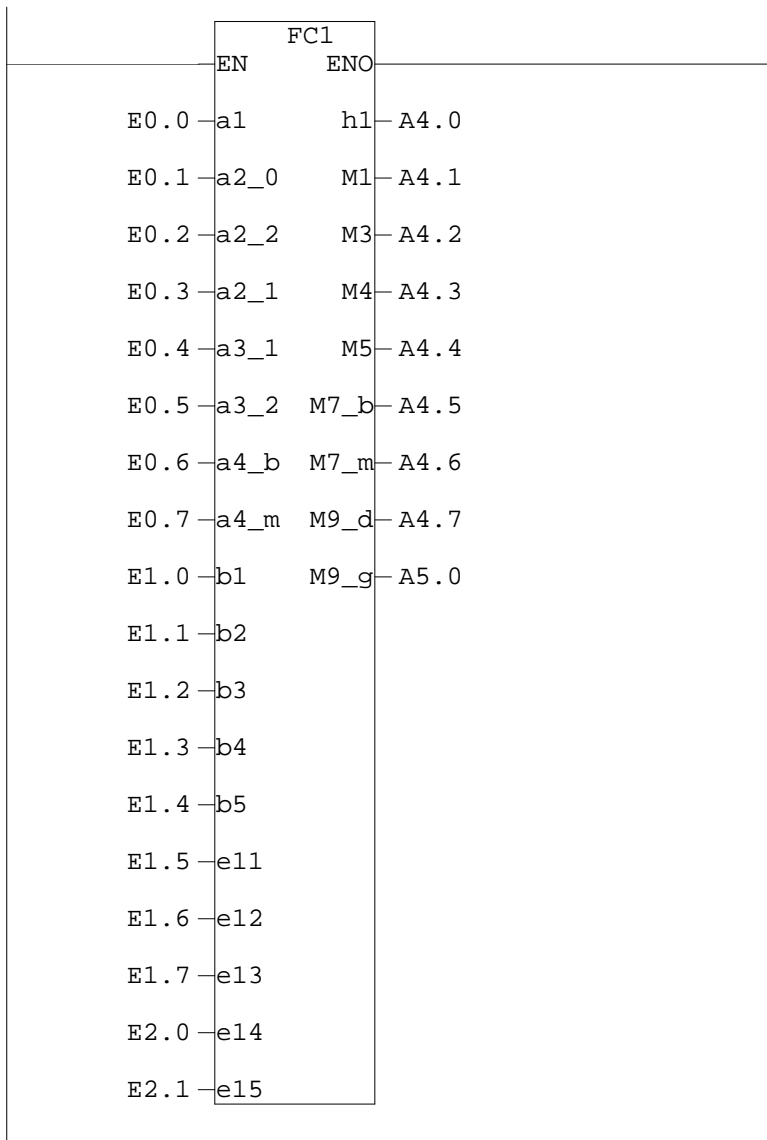
28/06/2009 09:50:51

15/02/1996 16:51:12

Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

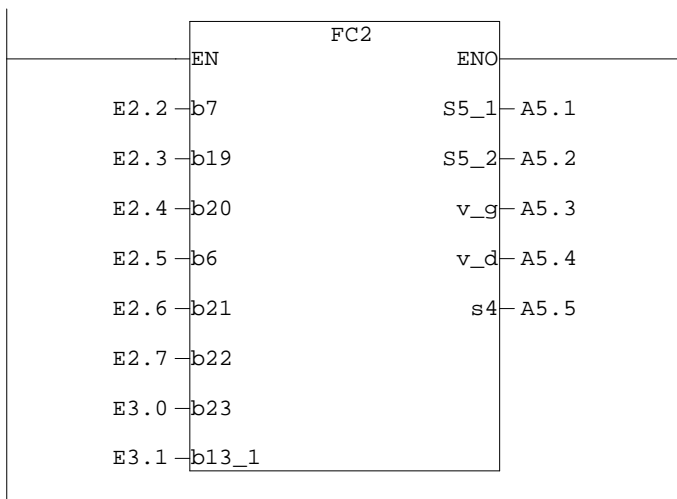
<b>Bloc : OB1 "Main Program Sweep (Cycle)"</b>
--

Réseau : 1

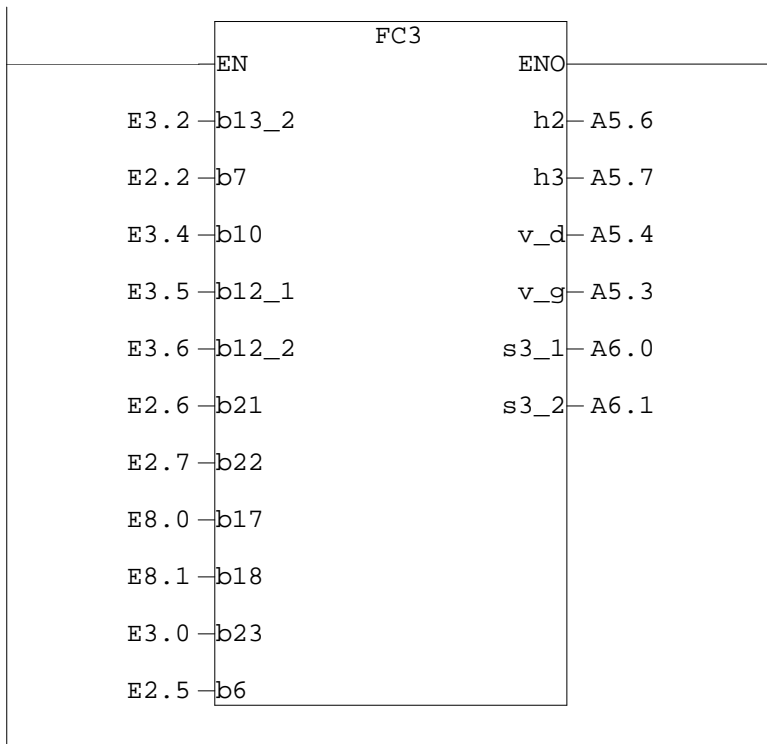


Réseau : 2

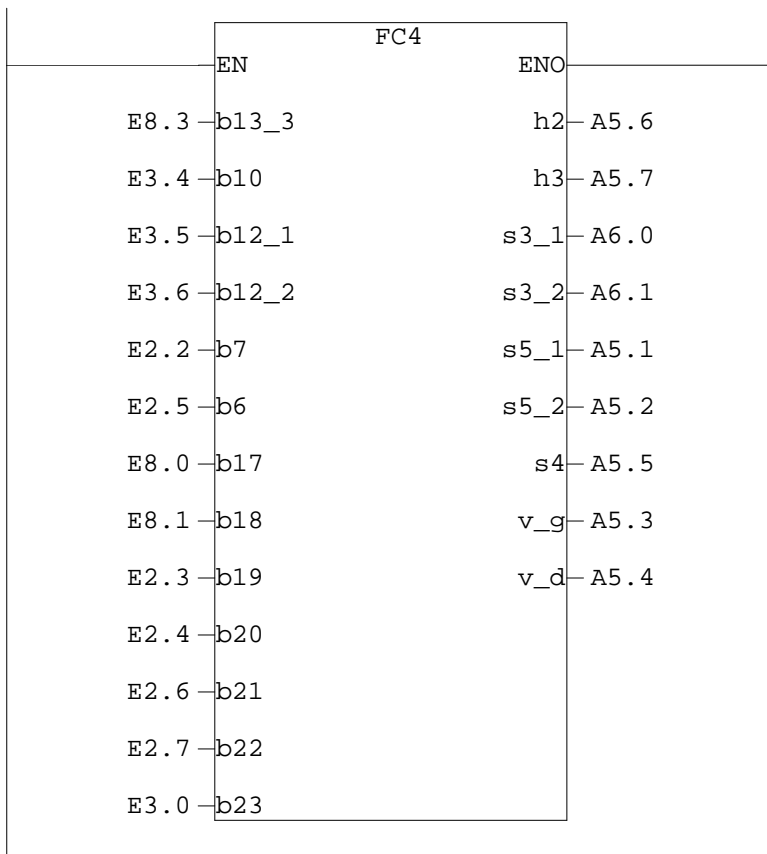
Affûtage périphérique avec va-et-vient et basculement automatique



Réseau : 3



Réseau : 4



# ***ANNEXE***

**FC1 - <offline>**

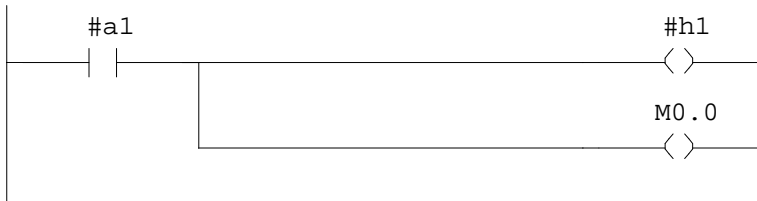
" "

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 29/06/2009 01:13:21  
**Interface :** 26/06/2009 22:42:04  
**Longueur (bloc/code /données locales) :** 00422 00260 00002

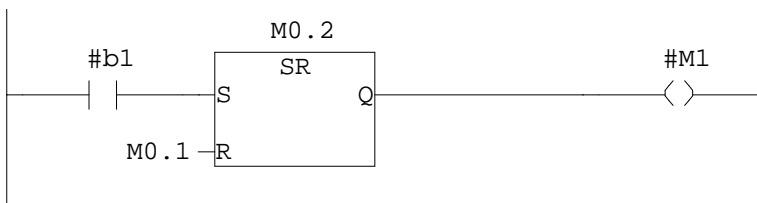
Nom	Type de données	Adresse	Commentaire
IN		0.0	
a1	Bool	0.0	
a2_0	Bool	0.1	
a2_2	Bool	0.2	
a2_1	Bool	0.3	
a3_1	Bool	0.4	
a3_2	Bool	0.5	
a4_b	Bool	0.6	
a4_m	Bool	0.7	
b1	Bool	1.0	
b2	Bool	1.1	
b3	Bool	1.2	
b4	Bool	1.3	
b5	Bool	1.4	
e11	Bool	1.5	
e12	Bool	1.6	
e13	Bool	1.7	
e14	Bool	2.0	
e15	Bool	2.1	
OUT		0.0	
h1	Bool	4.0	
M1	Bool	4.1	
M3	Bool	4.2	
M4	Bool	4.3	
M5	Bool	4.4	
M7_b	Bool	4.5	
M7_m	Bool	4.6	
M9_d	Bool	4.7	
M9_g	Bool	5.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1 cycle de sécurité

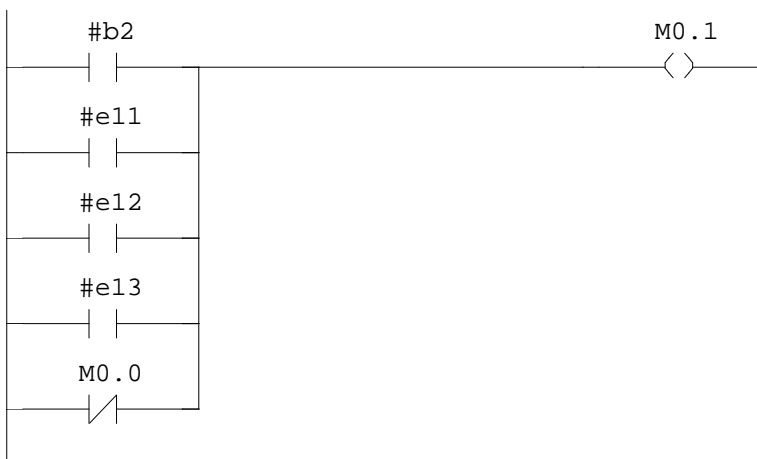
Réseau : 1 mise sous tension de la machine



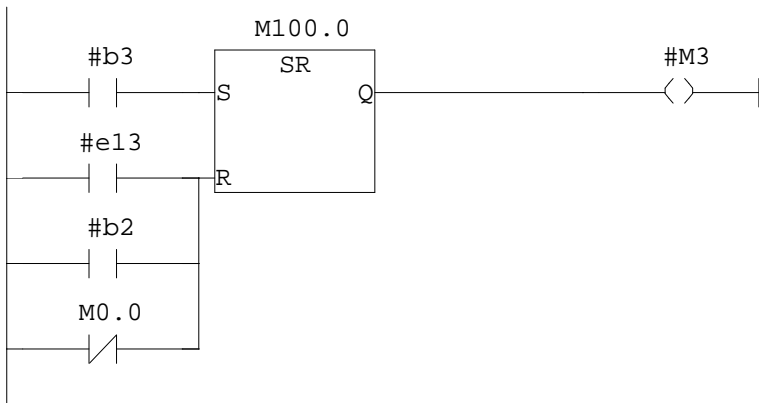
Réseau : 2 mise en marche de la pompe hydraulique



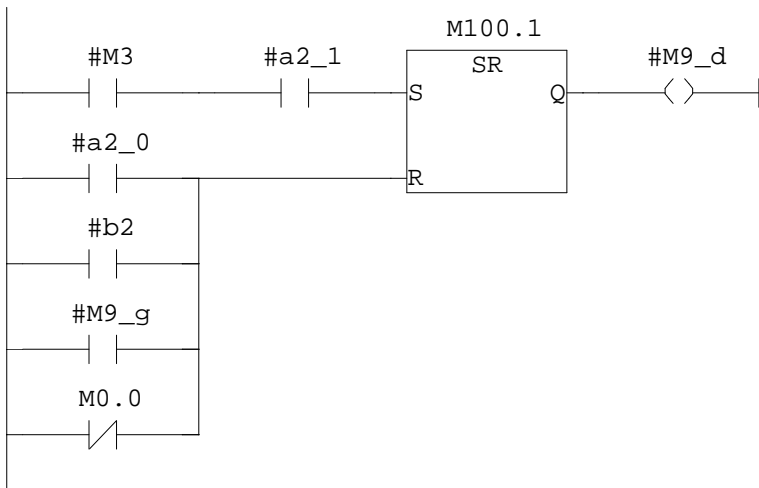
Réseau : 3 conditions d'arrêt de la pompe hydraulique



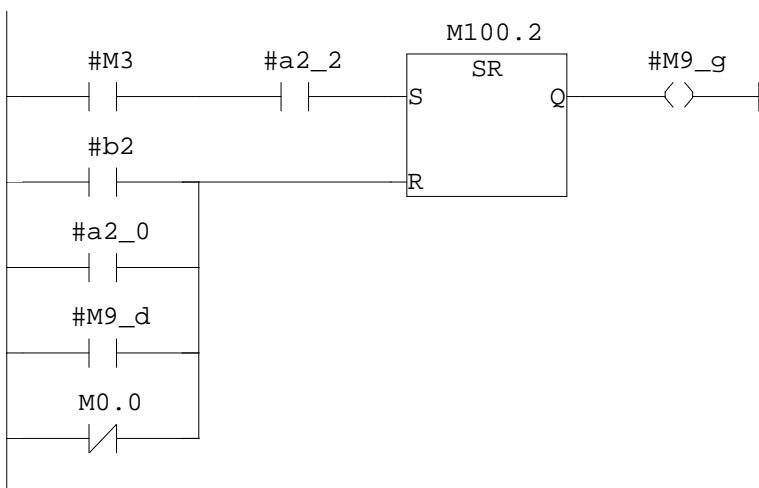
Réseau : 4 mise en marche du groupe léonard



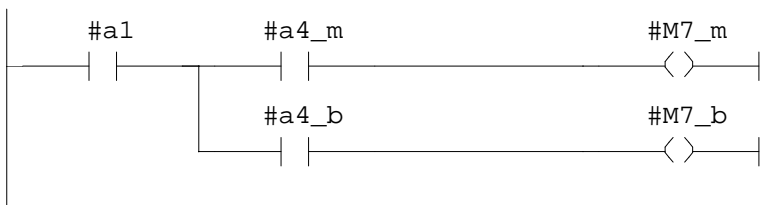
Réseau : 5 rotation de la de la meule à droite



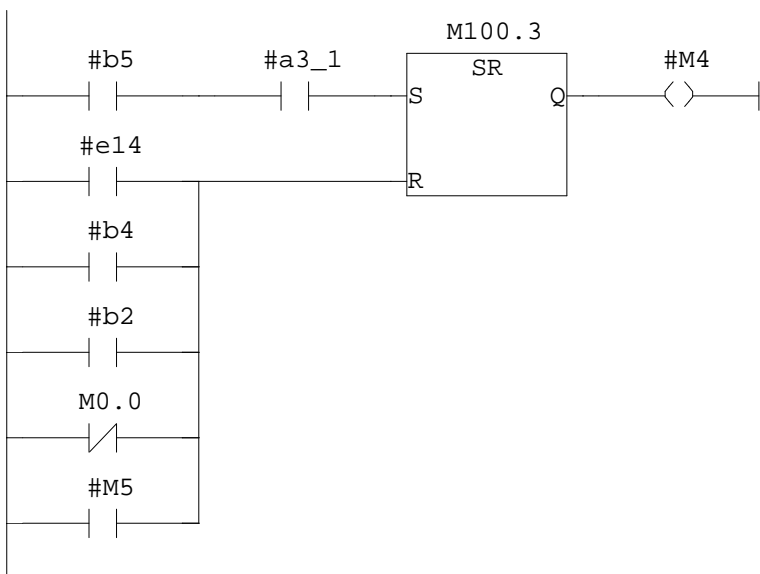
Réseau : 6 rotation de la meule à gauche



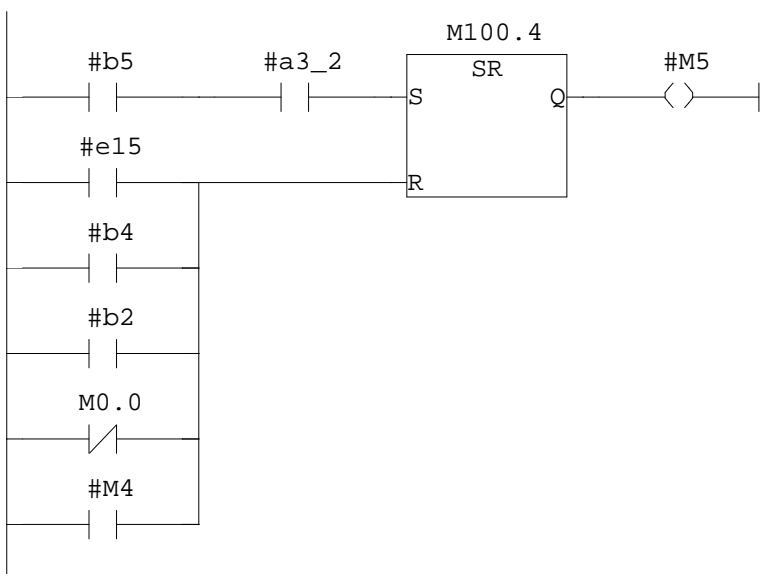
Réseau : 7 mise en marche du moteur monte et baisse



Réseau : 8 Aspiration de la poussière



Réseau : 9 Rotation de la pompe d'arrosage



**FC2 - <offline>**

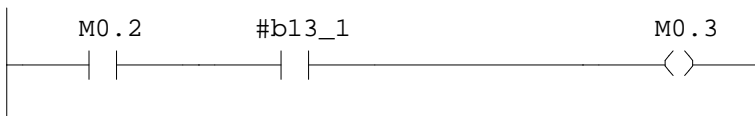
" "

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 29/06/2009 02:34:15  
**Interface :** 26/06/2009 22:51:06  
**Longueur (bloc/code /données locales) :** 00478 00336 00000

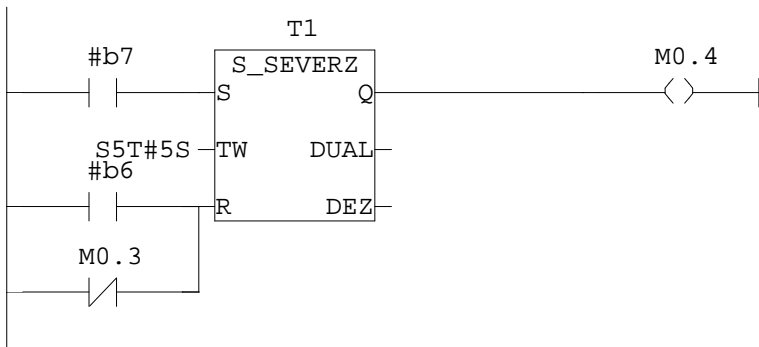
Nom	Type de données	Adresse	Commentaire
IN		0.0	
b7	Bool	0.0	
b19	Bool	0.1	
b20	Bool	0.2	
b6	Bool	0.3	
b21	Bool	0.4	
b22	Bool	0.5	
b23	Bool	0.6	
b13_1	Bool	0.7	
OUT		0.0	
S5_1	Bool	2.0	
S5_2	Bool	2.1	
v_g	Bool	2.2	
v_d	Bool	2.3	
s4	Bool	2.4	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

**Bloc : FC2 affûtage périphérique avec va-et-vient et basculement automatique**

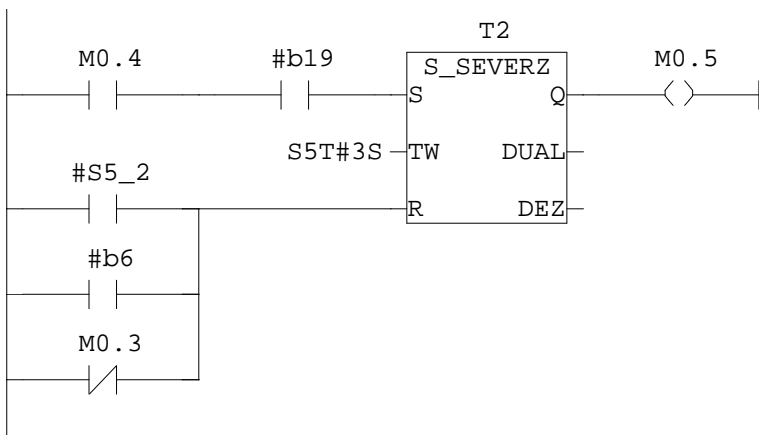
Réseau : 1      Sélecteur en position 1



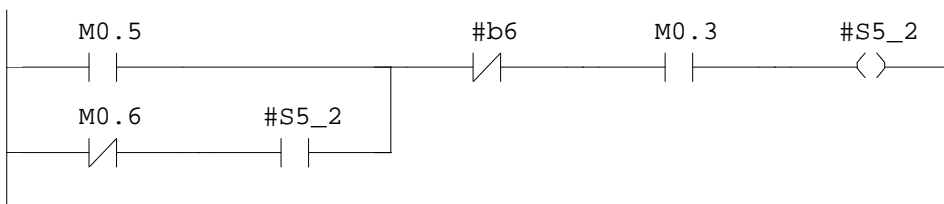
Réseau : 2      temporisation avant le mouvement du basculement automatique



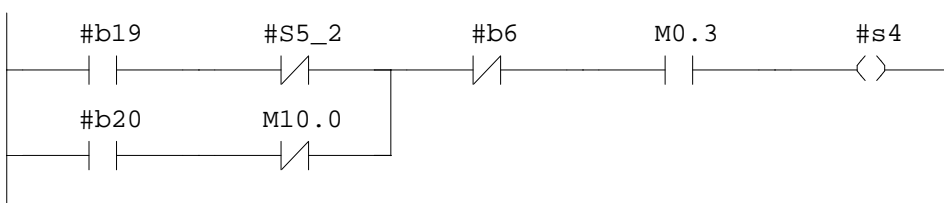
Réseau : 3      temporisation avant le basculement à gauche



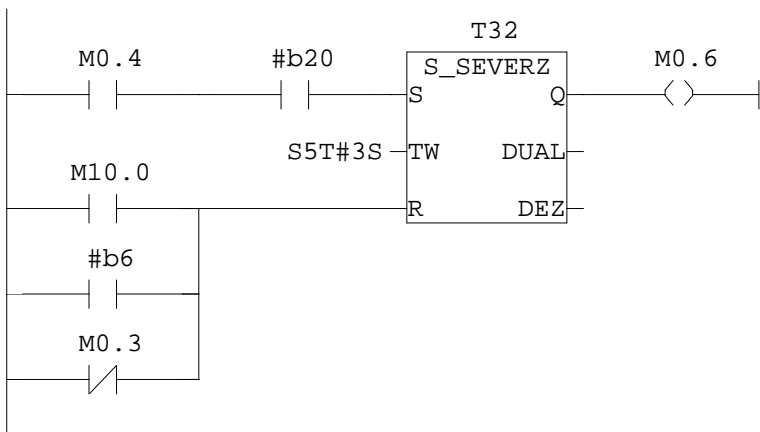
Réseau : 4      basculement à gauche



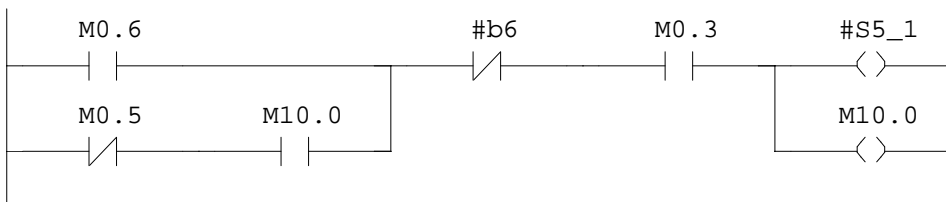
Réseau : 5      raletissement



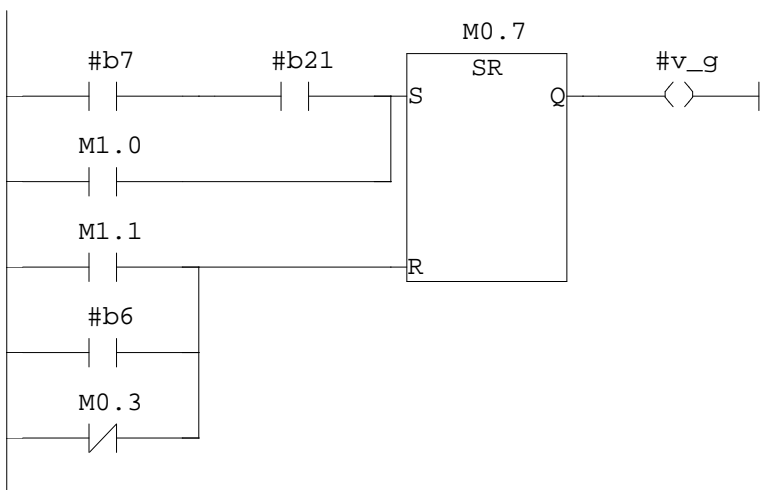
Réseau : 6      temporisation avant le basculement à droite



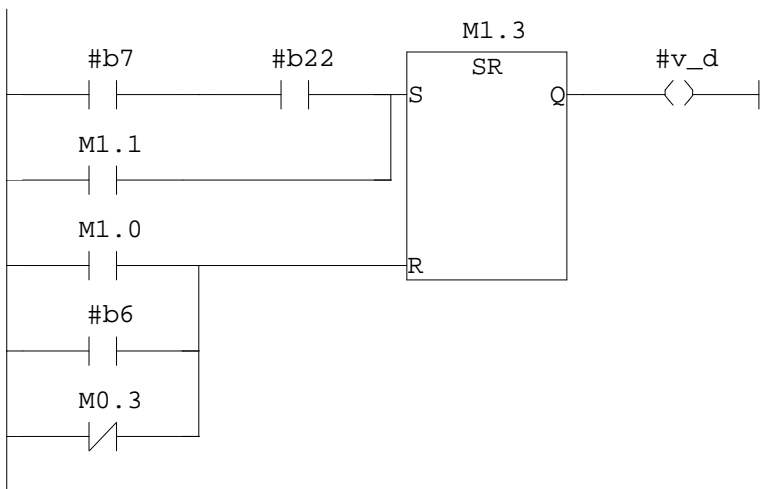
Réseau : 7      basculement à droite



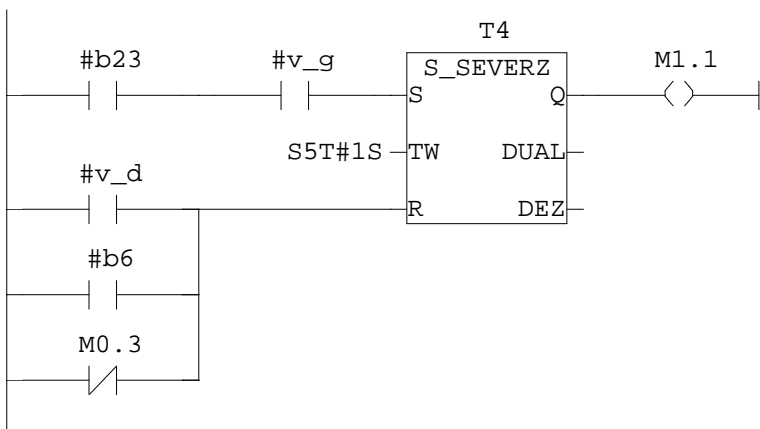
Réseau : 8      vérin à gauche



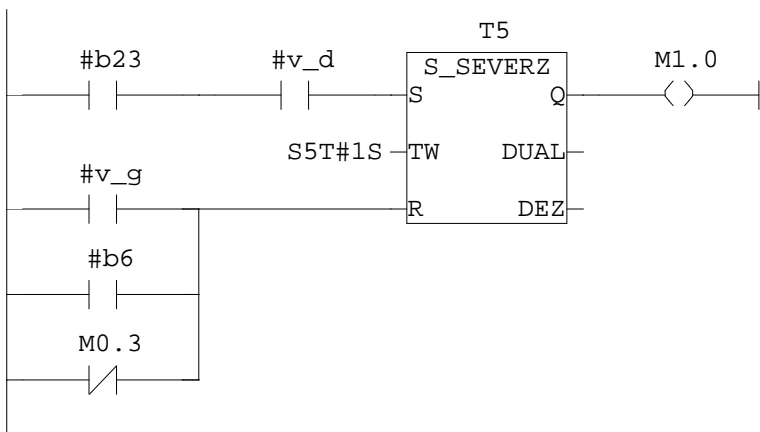
Réseau : 9 vérin à droite



Réseau : 10 commutation automatique



Réseau : 11 commutation automatique



**FC3 - <offline>**

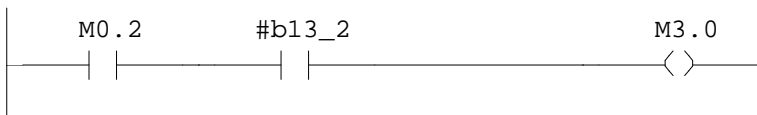
" "

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 27/06/2009 18:21:31  
**Interface :** 26/06/2009 22:20:24  
**Longueur (bloc/code /données locales) :** 00356 00214 00000

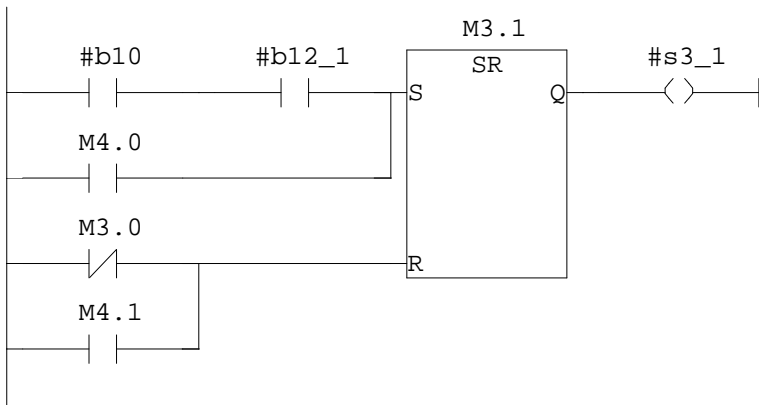
Nom	Type de données	Adresse	Commentaire
IN		0.0	
b13_2	Bool	0.0	
b7	Bool	0.1	
b10	Bool	0.2	
b12_1	Bool	0.3	
b12_2	Bool	0.4	
b21	Bool	0.5	
b22	Bool	0.6	
b17	Bool	0.7	
b18	Bool	1.0	
b23	Bool	1.1	
b6	Bool	1.2	
OUT		0.0	
h2	Bool	2.0	
h3	Bool	2.1	
v_d	Bool	2.2	
v_g	Bool	2.3	
s3_1	Bool	2.4	
s3_2	Bool	2.5	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

**Bloc : FC3 affûtage dent par dent**

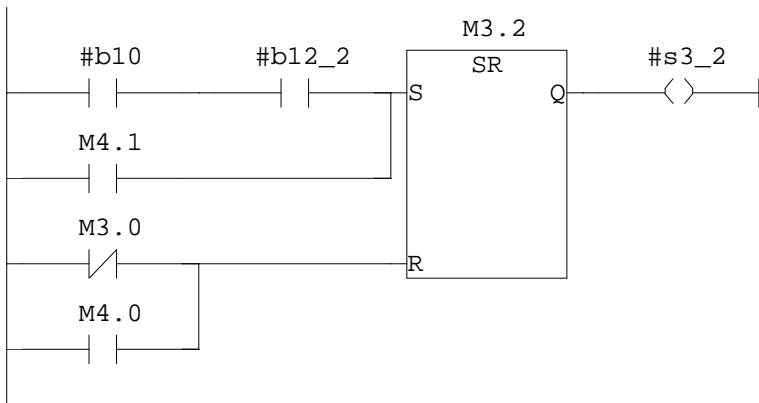
**Réseau : 1 sélecteur en position 2**



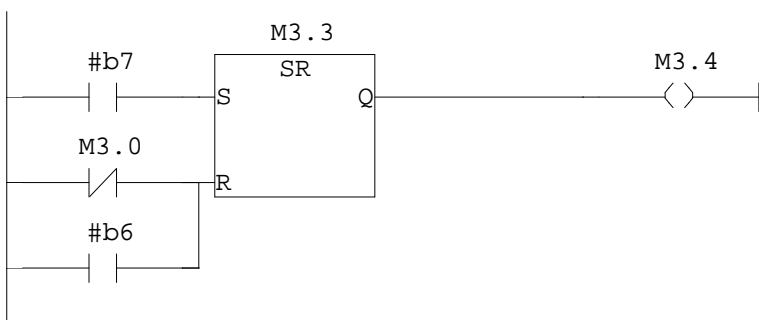
Réseau : 2      rotation de la fraise à droite



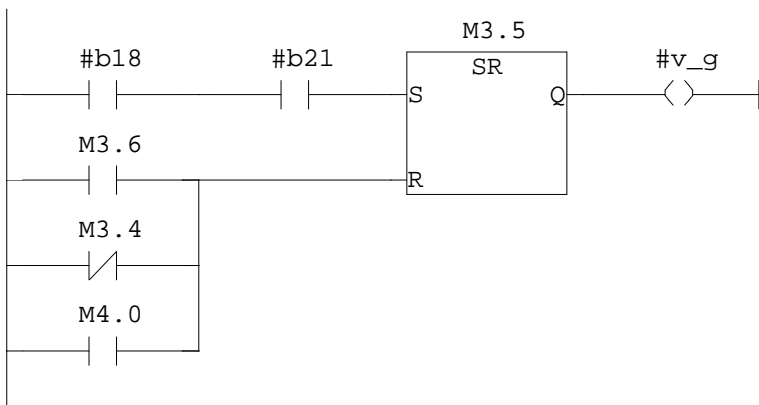
Réseau : 3      rotation de la fraise à gauche



Réseau : 4      bouton va et vient



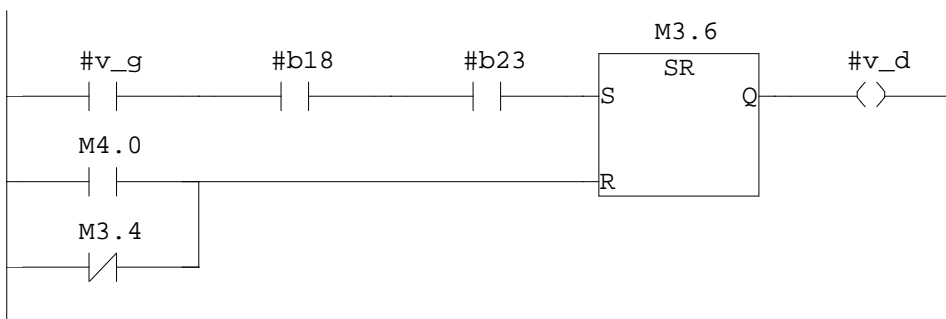
Réseau : 5      déplacement du chariot à gauche



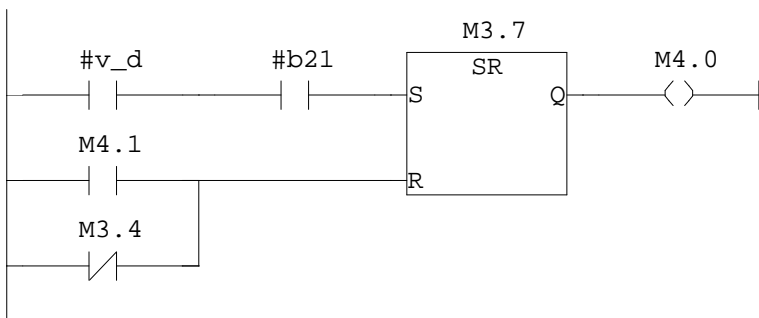
Réseau : 6      allumer le voyant h2



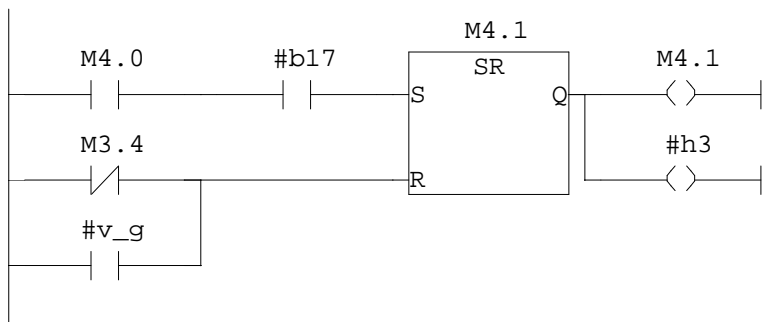
Réseau : 7      retour du chariot



Réseau : 8      rotation de la fraise à droite



Réseau : 9      rotation de la fraise à gauche



**FC4 - <offline>**

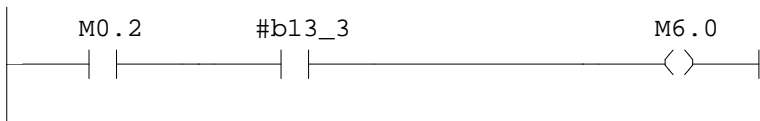
" "

**Nom :** **Famille :**  
**Auteur :** **Version :** 0.1  
**Version de bloc :** 2  
**Horodatage Code :** 28/06/2009 23:52:41  
**Interface :** 28/06/2009 00:17:08  
**Longueur (bloc/code /données locales) :** 00470 00308 00000

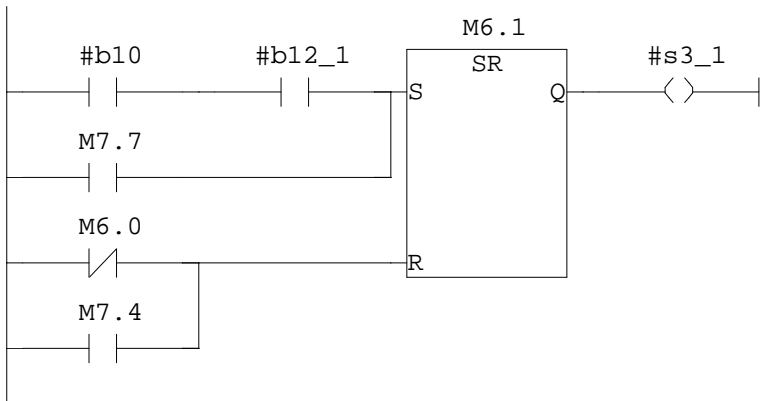
Nom	Type de données	Adresse	Commentaire
IN		0.0	
b13_3	Bool	0.0	
b10	Bool	0.1	
b12_1	Bool	0.2	
b12_2	Bool	0.3	
b7	Bool	0.4	
b6	Bool	0.5	
b17	Bool	0.6	
b18	Bool	0.7	
b19	Bool	1.0	
b20	Bool	1.1	
b21	Bool	1.2	
b22	Bool	1.3	
b23	Bool	1.4	
OUT		0.0	
h2	Bool	2.0	
h3	Bool	2.1	
s3_1	Bool	2.2	
s3_2	Bool	2.3	
s5_1	Bool	2.4	
s5_2	Bool	2.5	
s4	Bool	2.6	
v_g	Bool	2.7	
v_d	Bool	3.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

<b>Bloc : FC4 affûtage dent par dent avec basculement</b>
---

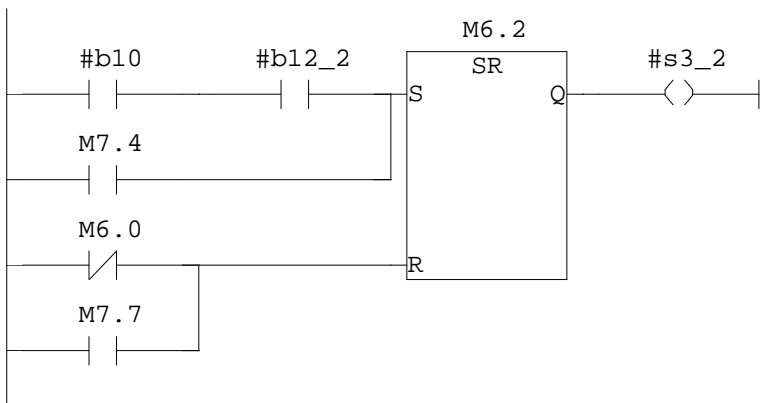
Réseau : 1      sélecteur en position 3



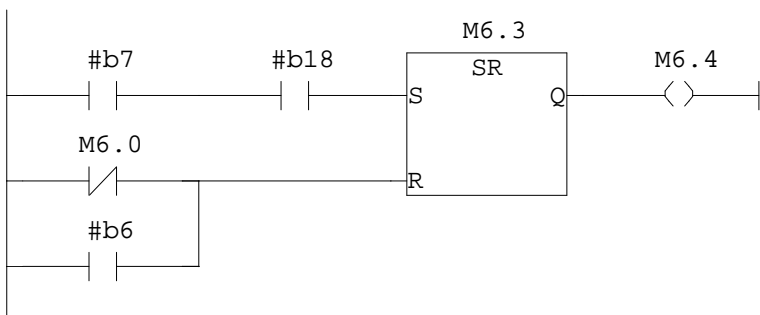
Réseau : 2      rotation de la fraise à droite



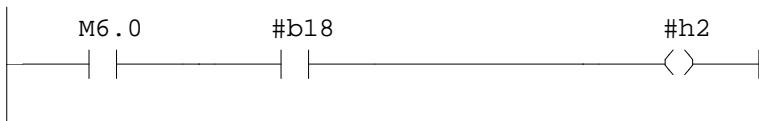
Réseau : 3      rotation de la fraise à gauche



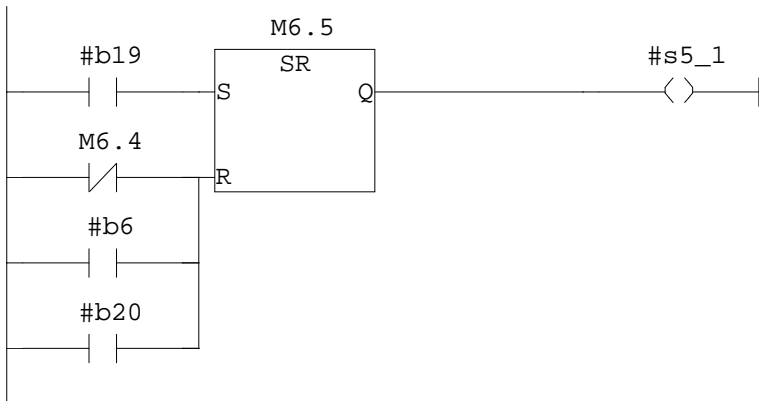
Réseau : 4      bouton va et vient



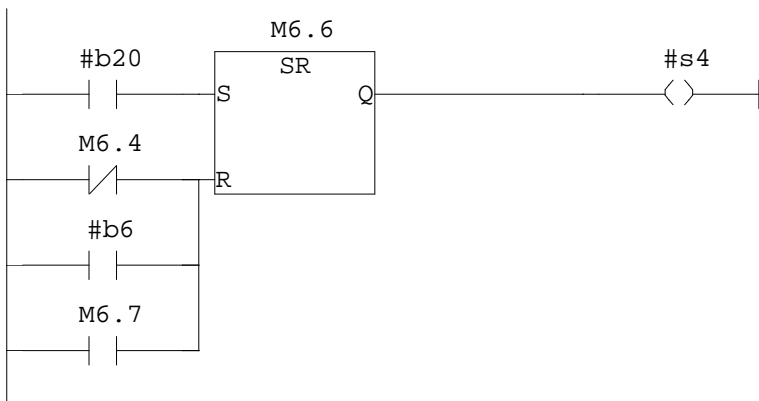
Réseau : 5 allumage de h2



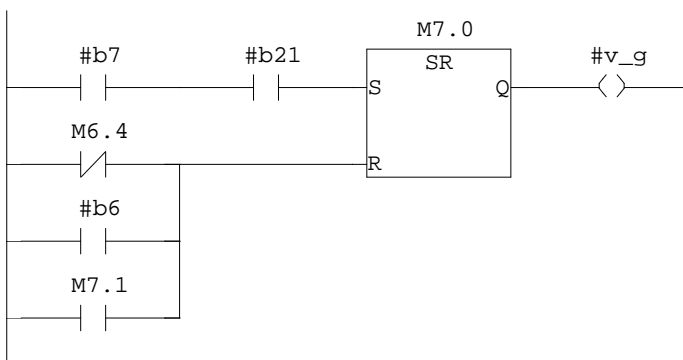
Réseau : 6 basculement automatique à gauche



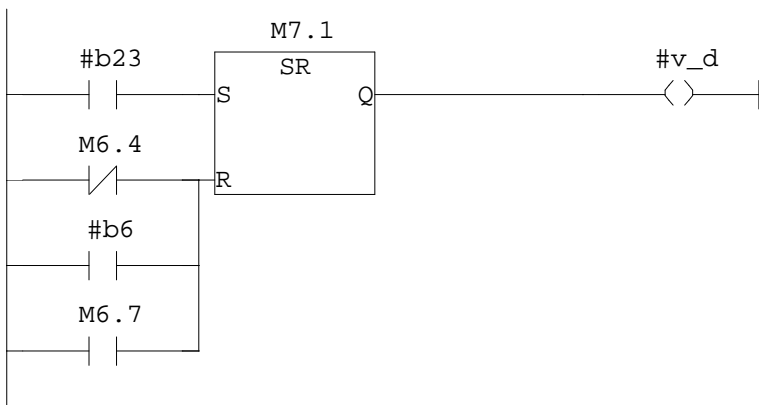
Réseau : 7 ralentissement



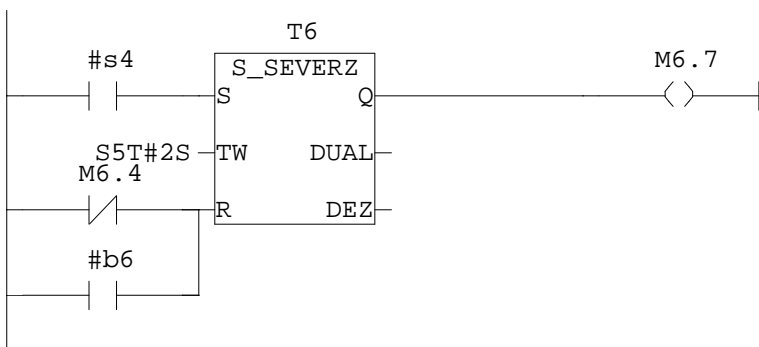
Réseau : 8 vérin à gauche



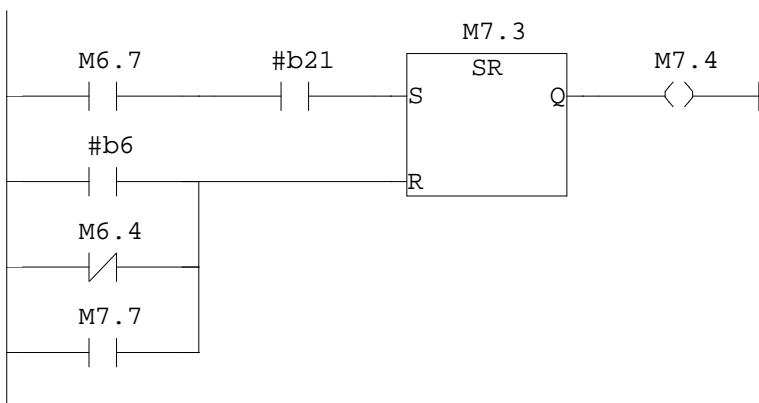
Réseau : 9 vérin à droite



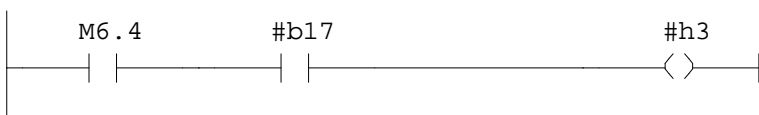
Réseau : 10 enclancher la division



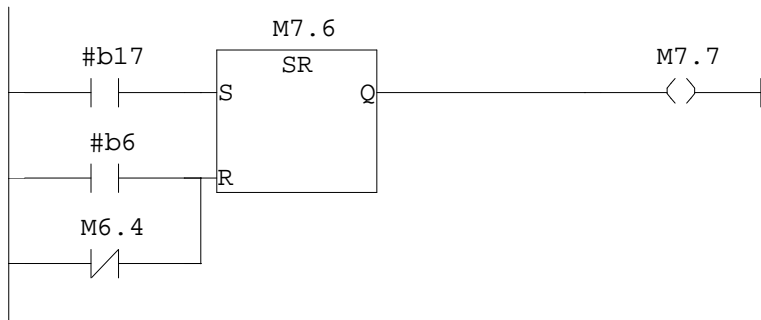
Réseau : 11 inverser le sens de rotation de la fraise



Réseau : 12 allumage h2



Réseau : 13      reinverser le sens de rotatioon de la fraise



## **Reuves et catalogues**

- Manuel de fonctionnement de l'affûteuse
- Schémas électriques et hydrauliques de l'affûteuse
- Documentation SIEMENS
- Documentation sur le GRAFCET
- Manuel d'utilisation du logiciel SIMATIC
- Les Automates Programmables Industriels d'Alain GONZAGA
- Introduction à l'automate programmable SIEMENS S300 et au logiciel Step7 du  
Département de Génie Electrique, Collège Montmorency
- Fundamentals of industrial instrumentation and process control de WILLIAM  
C.DUNN

## **Les Thèses**

- N-SLIMI et S-BOUNI, Etude et conception en vue de moderniser l'unité commande d'une machine-outil « GSP » de type aloseuse, Thèse d'ingénieur, Université de Boumerdes, département génie mécanique, option maintenance industrielle, 2008/2009.

## **Chapitre II**

# **MODELISATION DU FONCTIONNEMENT DE LA MACHINE**

## **Chapitre III**

# **AUTOMATISATION DE LA MACHINE PAR L'AUTOMATE PROGRAMMABLE INDUSTRIEL S7-300**

## **Chapitre VI**

# **PROGRAMMATION ET SIMULATION DU MODELE GRAFCET DE LA MACHINE**

# **Chapitre I**

## **DESCRIPTION ET FONCTION DE LA MACHINE**

# ***ANNEXE***

# ***ANNEXE***

# ***ANNEXE***

# **Chapitre I**

## **DESCRIPTION ET FONCTION DE LA MACHINE**