

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université 8 Mai 1945 – Guelma  
Faculté des Sciences et de la Technologie  
Département de Génie Electrotechnique et Automatique

Réf:...../2018



## MEMOIRE

Présenté pour l'obtention du **diplôme de MASTER Académique**

**Domaine:** Sciences et Technologie

**Filière:** Automatique

**Spécialité:** Automatique et informatique industrielle

**Par:** (Oulediaf Samir et Hamzaoui Youssef)

### Thème

**Etude d'un processus Industriel  
par automate programmable « S7- 300 »:Convoyeur rapide**

Soutenu publiquement, le **24/06 /2018**, devant le jury composé de:

M .MOUSSAOUI Abdelkrim	Professeur	Univ. Guelma	Président
M.BOUDJHEM Badreddine	MCA	Univ. Guelma	Encadreur
Mme BOUCERREDJ Leila	MCB	Univ. Guelma	Examineur
Mme LOUCIF Fatiha	MAA	Univ. Guelma	Examineur

**Année Universitaire: 2017/2018**



## *Remerciement*

*Le plus grande merci revient à dieu qui seul nous guide dans le bon sens durant notre vie et qui nous aide à réaliser ce modeste travail.*

*Nous tenons à remercier beaucoup notre encadreur Dr. : Boudjhem . B pour ses précieux conseils*

*Nous remercions tous ceux qui ont contribué près ou de loin à la réalisation de ce travail.*

*Qu'ils trouvent l'expression de nos reconnaissances et meilleure gratitude*

## Dédicaces

*L'imagination dispose de tout elle fait la beauté, la justice et le bonheur qui est le tout du monde. En témoignage de ma connaissance éternelle pour leurs encouragements et sacrifices je dédie ce modeste travail :*

- *à ma très chère maman*
- *à notre oncle Dr. Boucifhem. B*
- *à tout la famille Hamxaoui et Ouleddiaf*
- *Ma mère harraga louiza et mon père rabah*
- *A tout mes amies*
- *A tout ce qui m'aime e tous ce que j'aime*

## Sommaire

### Introduction générale

## Chapitre I : les automates programmables

I-1 Définition d'un automate programmable (API) .....	2
I-2 Architecture matériel d'un API .....	2
I-2-1 Structure externe .....	2
I-2-2 Structure interne .....	3
I-3 Architecture logiciel d'un API .....	4
I-3-1 Système d'exploitation.....	5
I-3-2 Programme utilisateur.....	5
I-4 Nature des informations traitées par l'automate .....	5
I-5 Les périphériques et auxiliaires des automates.....	6
I.6 Le module d'alimentation :.....	7
I.7 domaine d'utilisation des A.P.I :.....	/
I-8 L'automatisme.....	9
I-8-1 Définition d'un automatisme :.....	9
I.8.2 La LOGIQUE CABLEE (HARDWARE: LOGIQUE).....	9
I.8.2.1 Exemple de la commande câblée :.....	10
I.8.3 La logique programmée :(SOFTWARE).....	10
I.8.4 Structure d'un processus automatique.....	10
I.8.4.1 La partie commande ('PC') .....	10
I.8.4.2 L'automate :.....	10
I.8.4.3 La Partie opérative.....	11
I.9 Conclusion.....	13

## ChapitreII:Grafcet-STEP7-Langae contacte

II.1. Introduction :.....	14
II.2 GRAFCET.....	14
II.2.1.Description du GRAFCET .....	14
II.2.2. Les Règles d'évolution du GRAFCET.....	18
II.2.3. GRAFCET de niveau 1 et 2.....	18
II.2.3.1. GRAFCET de niveau 1.....	18
II.2.3.2. GRAFCET de niveau 2.....	18
II.3 le STEP 7.....	20
II.3.1. Introduction :.....	20
II.3.2. programme utilisateur :.....	20
II.3.3. Structure de programme :.....	20
II.3.3.1 Programme linéaire :.....	20
II.3.3.2 Programme segmenté :.....	20
II.3.3.3 Programme structuré :.....	21
II.3.4 Blocs du programme utilisateur :.....	21
II.3.4.1 Bloc d'organisation (OB) :.....	21
II.3.4.2 Bloc fonctionnel (FB) :.....	21
II.3.4.3 Fonction (FC) :.....	21
II.3.4.4 Bloc de données (DB) :.....	22

II.3.5	Présentation générale de l'automate S7-300 :	22
II.3.5.1	Caractéristiques de l'automate S7-300 :	22
II.3.5.2	Programmation de l'API S7-300.	23
II.4	langage contacte.....	24
II.4.1	Introduction.....	24
II.4.2	Les éléments graphiques de base :	24
II.4.2.1	Les éléments de teste :	24
II.4.2.2	Contact à fermeture.....	24
II.4.2.3	Contact à ouverture .....	25
II.4.2.4	Connexion de contacts en série.....	26
II.4.2.5	Utilisation de « Contact à fermeture » en série.....	27
II.4.2.6	Connexion de contacts en parallèle.....	28
II.4.3	Elément de liaison :	30
II.4.4	Eléments d'action:.....	30

### Chapitre III Application : de convoyeur rapide

III.1	Description de l'installation :	31
III.1.1	Schéma cinétique :	31
III.1.2	Description du fonctionnement :	31
III.2	Critiques sur l'installation actuelle.....	32
III.2.1	Inconvénient de la commande :	32
III.2.2	Solution proposée :	32
III.2.3	Avantage à apporté pour l'automatisation du système :	33
III.3	Solution du problème :	33
III.3.1	l'ravail demandé :	33
III.3.2	Description du fonctionnement convoyeur rapide :	33
III.3.3	Le système convoyeur rapide :	36
III.3.3.1	Description de la partie opérative :	36
III.3.3.2	Description de la partie commande(PC) :	39
III.4	Grafcet de système.....	42
III.4.1	Grafcet niveau 1 bob 1.....	42
III.4.2	Grafcet niveau 2 bob 1.....	44
III.4.3	Grafcet niveau 1 bob 2 :	46
III.4.4	Grafcet niveau 2 bob 2 .....	46
III.4.5	Grafcet niveau 1 bob 3 :	46
III.4.6	Grafcet niveau 1 bob 3 .....	46
III.5	Recensement des entrées.....	46
III.6	Ressensement des sorties.....	50
III.7	Programme.....	54
III.7.1	Tableau mnémonique.....	54
III.7.2	SIMULATION GRAPHIQUE.....	55
III.7.3	blocs FB1 FB2 OB1 .....	60
	Conclusion général.....	61

# Introduction général

L'automatisation des processus est devenue actuellement un élément essentiel dont les industries moderne nécessitent pour l'augmentation du rendement de production et l'amélioration de l'économie. Dans un système industriel, le système de commande constitue la partie la plus important dont les chercheurs n'ont pas cessé d'améliorer et perfectionner passant de logique câblée à la logique programmée. Cette dernière ouvre des grandes perspectives pour les systèmes automatisées.

Les premiers automates programmables en étaient introduits aux états unis vers les années 60 dans le domaine des Industries automobiles. Ce succès est dû en grande partie à la facilité de leur utilisation et leur prix de mise en œuvre.

Le but de notre travail est l'étude d'un processus industriel. Il s'agit d'un convoyeur rapide de l'unité Laminoir à Chaud.

Ce mémoire est organisé comme suit :

Le premier chapitre présente une description générale sur les automates programmables industriels

Le deuxième chapitre, présente la modélisation par grafcet, logiciel STEP7, et langage contacte

Finalement, le dernier chapitre consacre à la mise en place de l'automatisation du système du convoyeur rapide. Nous décrivons son fonctionnement, le grafcet qui correspond ainsi que sa mise en œuvre avec langage contact via STEP 7.

# **Chapitre I :**

## **Automates programmables**

## I-1) Définition d'un automate programmable (API) :

Un API (ou PLC Programmable Logic Controller) est un appareil électronique adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de pré actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques.

## I-2) Architecture matériel d'un API :

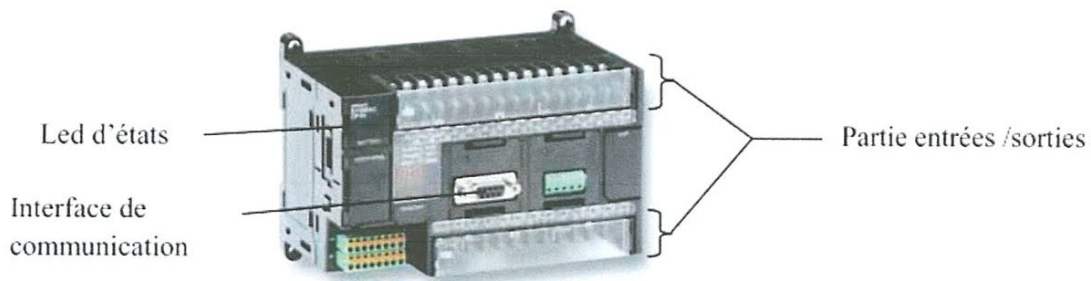
### I-2-1) Structure externe :

Les automates peuvent être de type compact ou modulaire.

Le type compact intègre à la fois le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité.

Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes.

La figure suivante montre un exemple d'automates compacts :

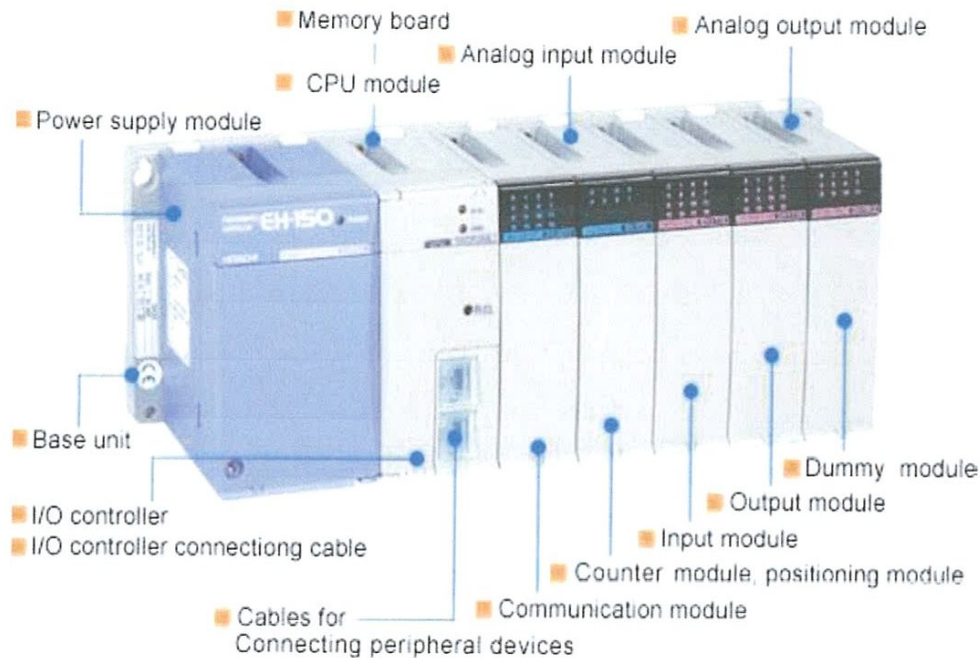


*Figure I. 1:* Automate compacte

Dans le type modulaire, le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant les bus plus les connecteurs.



Ces automates sont utilisés dans les automatismes complexes où puissance, capacité de traitement, flexibilité sont nécessaires. La figure I-2 montre un automate modulaire



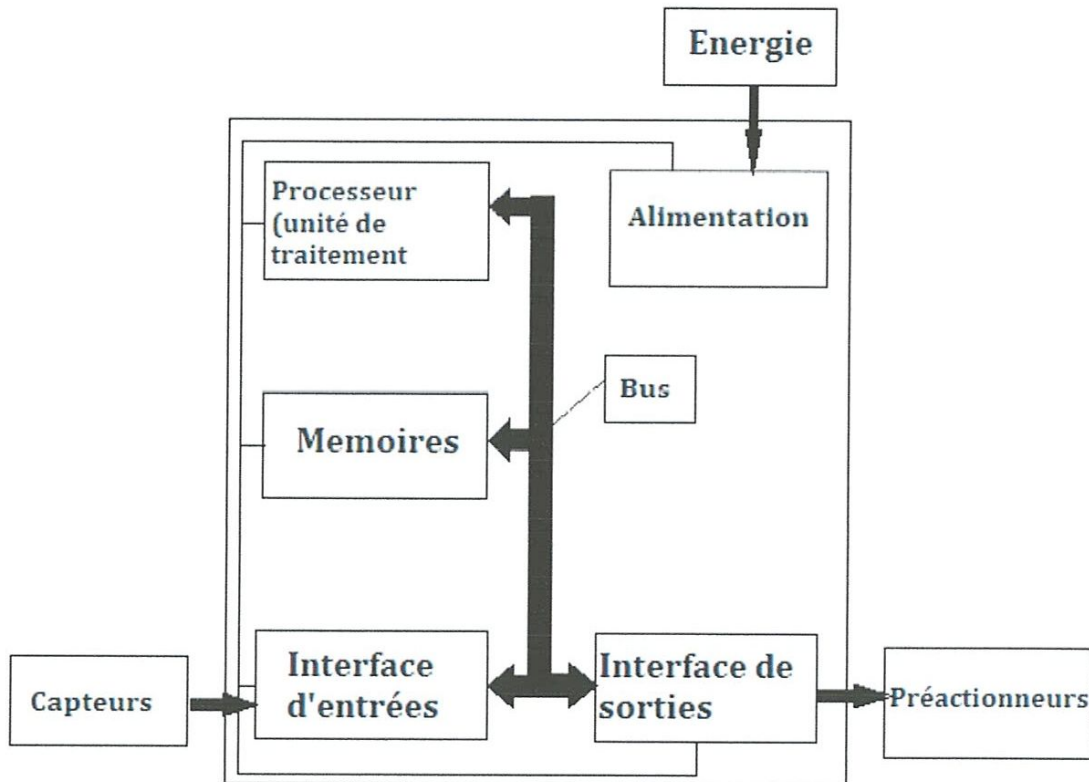
*Figure I. 2:* Automate modulaire

### I-2-2) Structure interne :

La figure I-3 montre la structure interne d'un API :

- **Module d'alimentation** : il assure la distribution d'énergie aux différents modules.
- **Unité de traitement** : à base de microprocesseur, elle réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation...).
- **Le bus interne** : il permet la communication de l'ensemble des blocs de l'automate et des éventuelles extensions.
- **Mémoires** : Elles permettent de stocker le système d'exploitation (ROM ou PROM), le programme (EEPROM) et les données système lors du fonctionnement (RAM).

- **Interface d'entrée** : elle permet de recevoir les informations du Système automatisé ou du pupitre et de mettre en forme (filtrage, ...) ce signal.
- **Interface de sortie** : elle permet de commander les divers prés actionneurs et éléments de signalisation du Système.



*Figure I. 3:* Structure interne d'un automate

### I-3) Architecture logiciel d'un API :

Dans un automate, Deux programmes différents s'exécutent d'une façon parallèle : le système d'exploitation et le programme utilisateur.

**I-3-1) Système d'exploitation :**

Le système d'exploitation, 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 du 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 des zones de mémoire,
- la communication avec des consoles de programmation et d'autres partenaires de communication. [1]

**I-3-2) Programme utilisateur :**

C'est le programme qui doit être créé et chargé dans la CPU. Il contient toutes les fonctions nécessaires au traitement d'une tâche d'automatisation spécifique. Il doit entre autres :

- déterminer les conditions pour le démarrage et le redémarrage de la CPU (par exemple, initialiser des signaux).
- traiter des données du processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques, définir des signaux binaires pour la sortie, écrire des valeurs analogiques).
- réagir aux alarmes. [1]

**I-4) Nature des informations traitées par l'automate :**

Les informations peuvent être du type :

- **Tout ou rien (T.O.R) :** l'information ne peut prendre que deux états (vrai/faux, 0 ou 1). C'est le type d'information délivrée par un détecteur, un bouton poussoir...
- **Analogique :** l'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température).

- **Numérique** : l'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent

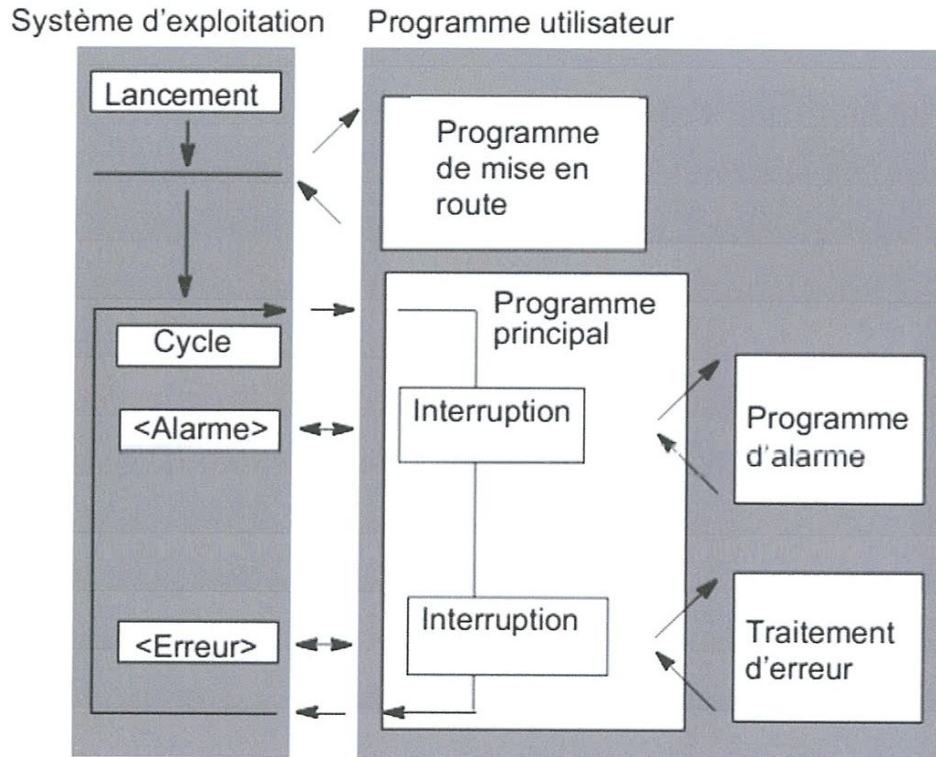


Figure I. 4: Déroulement du programme dans un API

### I-5) Les périphériques et auxiliaires des automates :

On peut regrouper les périphériques en trois classes :

- Les mémoires auxiliaires de masse : disque fixes ou amovibles, disque souples, tambours magnétiques, feuille ; languette ...
- Les unités des communications en entrée et en sortie :  
Lectures de cartes, imprimantes, unité visualisation, traceur de courbes..., Etc.
- Les unités de traitement associées: consolée analogique. Min calculateur, automate.

Les notices des automates programmables mentionnent, sous le qualificatif de périphériques, des systèmes à cassettes, des téléimprimeurs, disques souples ...etc. en réalité, en fonctionnement normale, un automate programmables n'échange généralement de

l'information qu'avec le procédé qu'il contrôle. Ces divers systèmes se sont donc des périphérique de la console de programmation et nom de l'automate lui-même : ce sont les auxiliaires de l' A.P.I cette notion regroupe :

- Le système d'enregistrement /lecteur à cassette.
- Le système de lecture et /ou de perforation, de ruban.
- Le boîtier de simulation.
- L'ensemble de diagnostic d'erreur ...Etc.

### **I.6) module d'alimentation :**

Le module d'alimentation fournit la tension nécessaire au fonctionnement de l'automate à partir de réseau il le module d'alimentation est équipé d'un dispositif de surveillance du secteur lors de la coupure secteur celui-ci déclenché:

- de scrutation du programme utilisateur
- La sauvegardé du contexte du système (contenu de la mémoire interne de processeur).
- La remise à zéro des sorties si l'utilisateur la spécifie dans son programme . Lors de la reprise L'arrête secteur deux cas se présent.
- L'automate effectue une reprise à chaude du fonctionnement c -à- d le redémarrage de l'exécution de programme à l'endroit ou il à été interrompu.
- L'automate effectue une reprise à froide de fonctionnement c.à.d. le redémarrage de l'exécution au début du programme après l'initialisation des données système et de l'application.

Dans tous les cas concepteur de la commande doit pouvoir si l'application le nécessite, impose par le biais du programme que l'installation ne puisse pas redémarrer sans intervention de l'utilisateur.

### **I.7 domaine d'utilisation des A.P.I :**

Pour les raisons qui viennent d'être évoquée, les A.P.I s'adressent à des applications que l'ontrouve dans la plupart des secteurs industriels, d'ores et déjà ces machines de la fonction permanente.

-

- **Métallurgies et sidérurgie :**

Les A.P.I sont utilisés dans les applications manutention en cokerie, changement des hautes fourneaux, Automatisation en fonderie, analyse de gaze control qualité.

- **Mécanique et automobile :**

On rencontre les A.P.I sur les chaînes de fabrications et montages, sur les bancs d'essais de moteurs en liaison avec diverses machines : presse, tous automatique, rectifieuse machines transfert, machines de soudage, machines à Scier automatique

- **L'industrie chimique :**

Actuellement les A.P.I sont utilisés en pilotage d'unité de protection, dosage et mélange de produits, épuration des effluents, des unités de transformation de plastique, en liaison avec certaines machines de l'industrie du caoutchouc.

- **L'industrie pétrolière :**

Les A.P.I peuvent être présentés dans les stations de pompes, à la demande et la surveillance des pipe-lines, la distribution des gazes et des liquides.

- **Industrie agricoles et alimentaire :**

On trouve des A.P.I aux postes de mélanges en ligne de produits et pulvérulents, de séchage et de contrôle des produits et aux systèmes de conditionnement d'aliments ou produits divers.

- **Transport et manutention :**

Nous avons ici toute une gamme de procédés séquentiels dans les quels les A.P.I peuvent rendre des précieux services : tri de paquets, de courrier, gestion mécanisées de parcs de stockage emballage, convoyage, palettisation, ascenseurs, monte-charges, gestion des parkings urbains ...etc.

- **Application diverses :**

Les A.P.I sont utilisés dans différents domaines, exemple :

Découpe automatique, contrôle fibrage dans le textile, cristalleries, sécurité dans le nucléaires...etc.

## I-8 L'automatisme

### I-8-1 Définition d'un automatisme :

On appelle un processus automatique ou un automatisme tout système industriel dont seul processus déroule d'une façon plus au moins autonome des que certaines actions initiales l'on fait démarrer (pression sur un bouton poussoir ou gradeur physique sur un capteur, etc.). De telles installations ont été mises au point en vue d'apporter des solutions à des problèmes à caractère technique, économique, de la sécurité et/ou humains

Un processus automatique doit permettre au moins :

- L'élimination des taches complexes, fatigants, rebutante et dangereuses.
- L'accroissement de la production ainsi que la productivité en générale (industrielle, agricole, alimentaire, etc...).
- La souplesse de changement de fabrication nécessaire, lors de basses répétitives, pour passer d'un type de fabrication à un autre de plus aisément possible.

Simultanément :

- L'amélioration de qualité des produits fabriqués ou conditionnés.
- Une plus grande sécurité des personnels.
- La surveillance et la protection des complexes et des machines industrielle.
- La réalisation des économies d'énergie industrielle et de matières.
- Enfin, la réduction des prix de revient de l'article fabriqué ou conditionné.

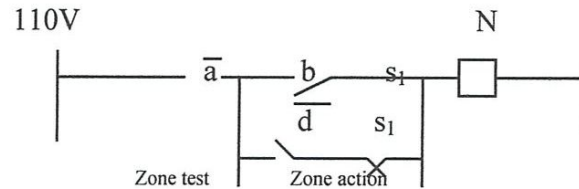
A quelque technologie qu'ils appartiennent, les processus automatique pénètrent chaque jour davantage dans tous les secteurs d'activité industrielle, à tous les niveaux, et jusqu'au secteur tertiaire

### I.8.2 LOGIQUE CABLEE (HARDWARE: LOGIQUE)

C'est toutes les fonctions technologiques à relayage électromécanique, électrique, ou pneumatique .Elles sont câblées entre elles par un câblage qui s'appui sur un schéma ou un diagramme BOOLEEN.

Tout changement de cycle de production ou évolution entrainera le changement ou la modification de l'architecture de la filière d'ou le nom : logique câblée.

### I.8.2.1 Exemple de la commande câblée :



Equation BOOLEENNE:  $S_1 = \bar{a} (b + \bar{d} S_1)$

### I.8.3 La logique programmée :(SOFTWARE)

On substitue au fonctionnement parallèle des outils câblés, une approche séquentielle mettant en œuvre un processus auquel on peut spécifier par des instructions appropriées la fonction à effectuer à un instant donné et les signaux concernés par cette fonction. Le processeur exécute les instructions une par une, et les unes après les autres. Pour fonctionner il a besoin de mémoires qui contiennent :

- des instructions à exécuter (programme)
- les données et les résultats de l'enchaînement de ces instructions.
- l'ensemble des instructions que comprend le processeur s'appelle le langage de programmation du processeur. Une suite particulière d'instruction décrivant un traitement de l'information donnée s'appelle un programme.

### I.8.4 Structure d'un processus automatique

Un processus d'automatisme comprend essentiellement trois parties :

**I.8.4.1 La partie commande (PC) :** elle comprend l'appareillage sensible aux modifications de l'évolution de processus (capteur ou intervention humaine) et transformant toute changement en impulsions électrique. [3] ,[4]

**I.8.4.2 L'automate :** c'est lui qui reçoit toute les informations qu'ils gèrent et qui lui permettant, après conditionnement, de diriger d'autres information, en accord avec le processus à développer, sur les sorties.

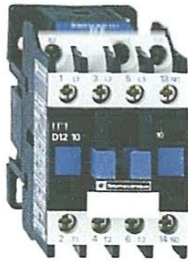


### 2.1. La Partie opérative

- Elle exécute les ordres qu'elle reçoit de la partie commande grâce aux actionneurs
- Elle possède des capteurs qui permettent de recueillir des informations.
- Elle reçoit des messages et envois des consignes vers la partie commande.

Elle comporte les éléments suivants :

- **Pré-actionneur** : est un constituant dont le rôle est de distribuer, sur ordre de la partie commande, l'énergie utile aux actionneurs. Les pré-actionneurs les plus utilisés sont les contacteurs (pour les moteurs électriques) et les distributeurs (pour les vérins pneumatiques).



Contacteur



distributeur électro pneumatique



distributeur pneumatique

*Figure I.5 :* pré-actionneurs

- **Actionneur** (moteur...) : Objet technique qui transforme l'énergie d'entrée qui lui est appliquée en une énergie de sortie (généralement mécanique) utilisable par un Effecteur pour fournir une action définie.



Moteur électrique




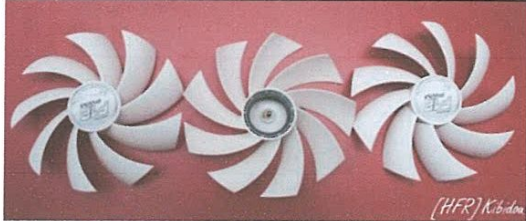
moteur pneumatique



vérin pneumatique

**Figure I.6.:** Actionneurs

- **Effecteur** : qui agissent sur la matière d'oeuvre (pales de ventilateurs...) (tout organe en contact avec la matière d'œuvre).

Système	Effecteur
 <p data-bbox="462 680 597 709">Ventilateur</p>	 <p data-bbox="1000 684 1227 714">pale de ventilateur</p>

*Fig.I.7* : Effecteurs

- **Capteur** : est un élément de prélèvement et de codage d'informations sur un processus ou sur l'environnement du système. Il convertit une grandeur physique (position, vitesse,...) en une information appelée compte-rendu et compréhensible par la Partie Commande.

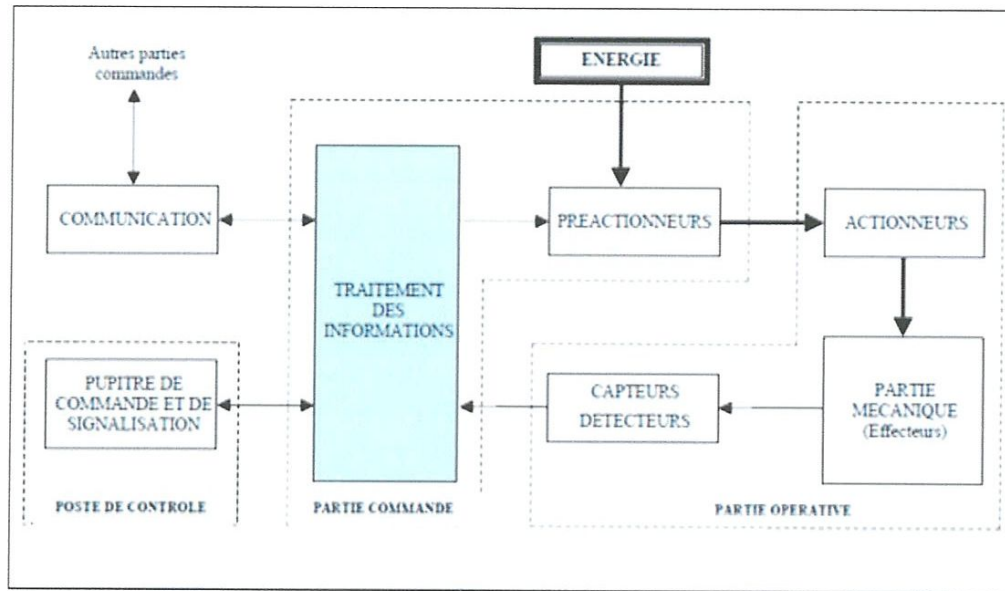


capteur de pression



capteur de position

*Fig.I.8* : Capteurs



*Fig I.9* : Structure d'un système automatisé. [5]

## I.9 Conclusion

L'API est un bon équipement s'il est bien choisi et bien employé.

Le rôle de l'automatisme industriel est prépondérant puisque les systèmes automatisés occupent et contrôlent l'ensemble des secteurs de l'économie, il a comme objectif d'améliorer la productivité, la qualité, la sécurité et autres variables qui peuvent influencés les objectifs de l'entreprise

**CHAPITRE II :**

**GRAFCET-STEP7-**  
**LANGAGE CONTACTE -**

## II.1. Introduction :

Ce chapitre est consacré au Grafcet et sa mise en œuvre par le langage contacte. Ainsi le but de chapitre est :

- Décrire les principaux éléments et les règles d'évaluation du GRAFCET.
- Présenté le langage contact et ces éléments graphique.
- Décrire le STEP 7, qui est le logiciel de base pour programmation et la configuration dans SIMATIC.
- Les bases de wincc flexible et simulation

## II.2 GRAFCET

### II.2.1. Description du GRAFCET

Le Grafcet (**GRA**phe **F**onctionnel de **C**ommande **E**tape **T**ransition) est créé en 1977 par un groupe de travail l'AF CET (Association Française pour la Cybernétique Economique et Technique). C'est un outil graphique de définition pour l'automatisme séquentiel en tout ou rien. C'est un langage universel qui peut se câbler par séquenceur et être programmé sur automate ou sur ordinateur.

Lorsque le mot GRAFCET en lettre capitale est utilisé, il fait référence à l'outil de modélisation. Lorsque le mot grafcet est écrit en minuscule, il fait alors référence à un modèle obtenu à l'aide des règles du GRAFCET présenté sous forme d'organigramme. Son but est la description du fonctionnement de l'automatisme contrôlant le procédé. C'est tout d'abord un outil graphique puissant directement exploitable. Il est aussi un langage pour la plupart des

API existants sur le marché. Il comprend :

- des étapes associées à des actions
- des transitions associées à des réceptivités,
- des liaisons orientées reliant étapes et transitions. [5]
- **Les étapes**

L'étape symbolise un état ou une partie de l'état du système. Elle possède deux états possibles: active représentée par un jeton dans l'étape ou inactive. L'étape *i*, repérée

numériquement, possède ainsi une variable d'état, appelée variable d'étape Xi. Cette variable est une variable booléenne valant 1 si l'étape est active, 0 sinon.

- **Etape initiale :**



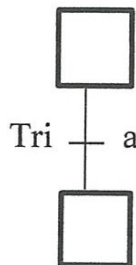
Elle représente le système à l'état de repos initial. Elle est activée au début du cycle.

- ❖ **Etape :**



A chaque étape est associée une action ou plusieurs, c'est à dire un ordre vers la partie opérative ou vers d'autres grafcecs.

- **Les transitions :**



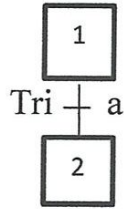
Une transition indique la possibilité d'évolution qui existe entre deux étapes et donc la succession de deux activités dans la partie opérative. Lors de son franchissement, elle va permettre l'évolution du système. A chaque transition est associée une réceptivité qui exprime la condition nécessaire pour passer d'une étape à une autre.

Cette condition est écrite sous forme d'une proposition logique, une fonction combinatoire calculée à partir :

- des variables d'entrées traduisant l'état des capteurs, des boutons poussoirs, etc.
- du temps
- de l'état courant des étapes du grafcec (les Xi).

Si la réceptivité n'est pas précisée, alors cela signifie qu'elle est toujours vraie.

**Règle :** si l'étape  $i$  est inactive,  $X_i = 0$ , la transition  $Tri$  est sans effet. Cependant, attention, valider sans raison une transition peut avoir des conséquences graves, perturbant le cycle dans certains



Cas Si l'étape 1 est active,  $X_i = 1$ , la transition  $a$  est validée, alors :

Si  $a = 0$ , alors attente

Si  $a = 1$ , alors l'étape  $i$  est validée  $X_i = 0$  et l'étape suivante 2 est activée,

$X_{i+1} = 1$ .

- **Les liaisons orientées :**

Une liaison orientée est le lien qui lie une étape à une transition ou l'inverse. Par convention, étapes et transitions sont placées suivant un axe vertical. Les liaisons orientées sont de simples traits verticaux lorsque la liaison est orientée de haut en bas, et sont munis d'une flèche vers le haut lorsque la liaison est orientée vers le haut.

- **Notation des entrées/sorties :**

Lors de l'établissement du grafcet de spécification de niveau 1, on utilise des noms explicites pour les entrées du système modélisé ainsi que pour les sorties. Lors du passage au grafcet de réalisation de niveau 2, on utilise plutôt des noms logiques :  $E_i$  pour les entrées et  $S_i$  pour les sorties.

- **Les actions**

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

- **Action continue :** La ou les sorties correspondant à l'ordre  $a$  sont mises à 1 tant que l'étape associée est active. Lorsque l'étape devient inactive, la ou les sorties sont mises à 0.

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

- Action conditionnelle simple : Type C (Condition)
- Action retardé : Type D (delay)

- Action limitée dans le temps : Type L (limité)
- **Action mémorisée** : On peut ainsi donner l'équation d'un ordre  $a$  en fonction des états des étapes, des conditions éventuelles et du temps.

- **Les réceptivités**

Une réceptivité est associée à chaque transition. C'est une fonction booléenne calculée à partir des entrées du graphe, des états des étapes ou des temporisations. Une réceptivité est donc écrite en utilisant les opérateurs ET, OU, NON et front. Le front montant ou descendant d'une variable permet de situer dans le temps le changement de valeur de ce capteur. On les note respectivement E et  $\bar{E}$ . Les fronts ne sont à 1 que durant un délai, qui correspond au temps de prise en compte de l'évènement, c'est à dire le temps de franchissement d'une transition.

- **Les temporisations**

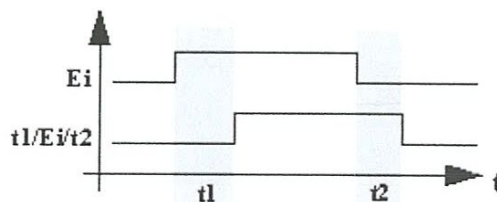
Les temporisations sont des variables booléennes qui permettent une prise en compte du temps. Pour écrire ces temporisations, on fait appel à un opérateur normalisé " $t1/Ei/t2$ " (CEI/IEC 617-12). Cet opérateur sert de base à la notation utilisée en GRAFCET. Les temporisations utilisés en GRAFCET font référence aux variables d'étapes et donc s'écrivent sous la forme " $t1/Ei/t2$ " (alors  $t1$  désigne le retard apporté au changement de l'état inactif à l'état actif de l'étape  $i$ , et  $t2$  désigne le retard apporté au changement de l'état actif à l'état inactif de l'étape  $i$ ).

Il est important de noter que:

L'étape  $i$  doit être active pendant un temps supérieur ou égal à  $t1$  pour que " $t1/Ei/t2$ " puisse passer à l'état vrai.

Si  $t1 = 0$ , on note " $Ei/t2$ ", Si  $t2 = 0$  on note " $t1/Ei$ ".

Il faut préciser l'unité de temps à laquelle on fait référence.





## II.2.2. Les Règles d'évolution du GRAFCET

### *Règle N°1 – Condition initiale*

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

### *Règle N°2 – Franchissement d'une transition*

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

### *Règle N°3 – Evolution des étapes actives*

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

### *Règle N°4 – Franchissement simultané*

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

### *Règle N°5 – Conflit d'activation*

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

## II.2.3. GRAFCET de niveau 1 et 2

### II.2.3.1. GRAFCET de niveau 1

C'est en général la description de l'automatisme seul, c'est à dire l'enchaînement des actions et des transitions permettant de contrôler le procédé. Lorsque l'on aborde l'analyse et la description d'un système, on ne sait pas quelle technologie sera retenue pour les actionneurs, les capteurs et la commande. On décrira dans ce GRAFCET les actions et les évènements en termes généraux.

### II.2.3.2. GRAFCET de niveau 2

C'est la description complète de l'automatisme qui tient compte de toutes les contraintes du procédé. Les points essentiels du GRAFCET de niveau 2 sont :

- la simplification du GRAFCET niveau 1, c'est à dire les parallélismes et les séquences répétées,
- les modes de fonctionnement de l'automatisme, c'est à dire la prise en compte de la marche automatique, de la marche par cycle et de la marche manuelle utile pour la maintenance et les réglages,
- les arrêts d'urgence entraînant la coupure d'alimentation, l'utilisation d'une variable AU (si arrêt d'urgence  $AU = 1$ ) à rajouter dans les conditions logiques pour les différentes actions, une procédure de dégagement si  $AU = 0$ ,
  - les sécurités procédé, c'est à dire l'arrêt de l'automatisme si une condition anormale est détectée, par exemple la détection d'un objet incorrect ou mal positionné,
  - les conditions initiales, c'est à dire les différentes positions au repos (au départ du cycle), l'état de l'alimentation et l'ensemble des tests destinés à vérifier les conditions initiales.

La complexité de cette tâche peut être grande. Le GRAFCET de niveau 2 doit être étudié avec le plus grand soin. Ceci peut éventuellement conduire à modifier le procédé, et à modifier si besoin les capteurs et les actionneurs. Dans certains cas, on peut être amené (après étude) à rechercher d'autres solutions que l'automatisme séquentiel, par exemple la logique floue. Ces cas limites sont :

- une indétermination logique au niveau du procédé,
- une description logique inadaptée,
- une durée de cycle de scrutation ou d'acquisition trop longue,
- la difficulté à définir une séquence de dégagement pour l'arrêt d'urgence.

**Remarque :** En milieu industriel, les essais erreurs peuvent être dangereux ou catastrophiques : d'où l'intérêt de faire une étude du grafcet de niveau 2 très approfondie.

## **II.3 STEP 7**

### **II.3.1. Introduction :**

Le STEP 7 est le logiciel de base pour programmation et la configuration dans SIMATIC. Il est formé d'un ensemble d'application avec lesquelles on peut aisément réaliser des tâches partielles.

Le SIMATIC MANAGER est le gestionnaire de projet SIMATIC, il organise la mise en commun dans un projet de toutes les données et de tous les paramètres requis afin de réaliser une tâche d'automatisation.

A l'aide de STEP 7, on peut scinder une installation en projet. Par conséquent, la création d'un projet ou d'une structure de projet est une condition essentielle pour travailler avec le STEP 7.

### **II.3.2. Programme utilisateur :**

Le programme utilisateur contient toutes les fonctions nécessaires au traitement d'un problème d'automatisation spécifique, il est chargé dans la CPU qui comprend différents blocs avec lesquels avons structuré notre programme, ces blocs peuvent être : blocs d'organisation, de fonction ou des données.

### **II.3.3. Structure de programme :**

Le programme contient trois classes :

#### **II.3.3.1 Programme linéaire :**

L'ensemble du programme se trouve dans le bloc d'organisation OB1, ce modèle rappelle la logique des contacts à relais câblés que les automates programmables ont remplacé, la CPU traite successivement les différentes opérations.

#### **II.3.3.2 Programme segmenté :**

Le programme subdivisé en blocs, chacun d'entre eux contenant uniquement le programme nécessaire pour réaliser la tâche qui lui est assignée. Un bloc peut à son tour se subdiviser en plusieurs réseaux. Lorsque l'on utilise des réseaux similaires, on peut créer des modèles.

Le bloc d'organisation OB1 contient des opérations qui appellent d'autres dans un ordre défini.

### II.3.3.3 Programme structuré :

Un programme structuré contient des blocs avec des paramètres appelé blocs paramétrables. Ces blocs sont conçus pour être universellement utilisables.

Lors de l'appel d'un bloc paramétrable, effectifs (les adresses effectives des entrées et sorties ainsi que les valeurs paramétriques) lui sont transmis.

### II.3.4 Blocs du programme utilisateur :

Le logiciel de programmation STEP7 offre la possibilité de structurer le programme utilisateur, c'est-à-dire le subdiviser en différentes parties, soit autonomes ou dépendantes qui donnent les avantages suivants : [6]

- Ecrire des programmes importants et clairs.
- Standardiser certaines parties du programme.
- Simplification de l'organisation du programme.
- Modification facile du programme.
- Simplifier le test du programme.
- Faciliter la mise en service.

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

#### II.3.4.1 Bloc d'organisation (OB) :

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

#### II.3.4.2 Bloc fonctionnel (FB) :

Les blocs fonctionnels sont des blocs de code qui sauvegardent en permanence leurs valeurs dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement du bloc.

#### II.3.4.3 Fonction (FC) :

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

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

- renvoyer une valeur de fonction au bloc appelant (exemple : fonction mathématique).
- Exécuter une fonction technologique.

#### **II.3.4.4 Bloc de données (DB) :**

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

Tous les FB, FC, OB peuvent lire les données contenues dans un DB global ou écrire des données dans un DB global. Ces données sont conservées dans le bloc de données même lorsqu'on quitte le DB, tandis que les DB locale sont spécifiques à un FB et sont accessible uniquement par celui-ci.

#### **II.3.5 Présentation générale de l'automate S7-300 :**

L'automate S7-300 est un automate modulaire fabriqué par la firme SIEMENS, on peut le composer en fonction de nos besoins à partir d'un vaste éventail de modules. Les automates SIEMENS sont conçus indépendamment d'une tâche précise. Tous les éléments logiques, fonctions de mémoire, temporisations, compteurs...etc., nécessaires à l'automatisation sont prévus par le fabricant et sont intégrés à l'automate. Ils se distinguent principalement par le nombre des :

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

##### **II.3.5.1 Caractéristiques de l'automate S7-300 :**

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

- Gamme diversifiée de la CPU.
- Gamme complète du module.
- Possibilité d'extension jusqu'à 32 modules.
- Bus de fond de panier intégré en module.
- Possibilité de mise en réseaux avec MPI, PROFIBUS ou INDUSTRIAL ETHERNET.

- Liberté de montage au différent emplacement.
- Configuration et paramétrage à l'aide de l'outil configuration matériel.

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

### II.3.5.2 Programmation de l'API S7-300 :

Un API est programmé à l'aide d'outils de développement spécialisés, fournis par son constructeur (exemple : Step7 pour SIEMENS et PL7 pour SCHNEIDER), et utilisables au travers d'une interface (un logiciel sur PC, un pupitre...). Un standard définit quatre langages correspondant aux familles de langages les plus utilisées pour la programmation des API, les outils de développement support au moins un de ces langages, ces langages sont: [6]

- **Langage CONT (LD : Ladder Diagram)** : est un langage de programmation graphique. La représentation est inspirée des schémas de circuits. Les requêtes des signaux binaires sont placées sur les circuits sous forme de contacts.
- **LOG (logigramme)** : est un langage de programmation graphique qui utilise les opérateurs de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes peuvent être représentées directement combinées avec les portes logiques.
- **Langage LIST (IL : Instruction Liste)** : est un langage de programmation textuel proche du langage machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme.

## II.4 LANGAGE CONTACTE

### II.4.1 INTRODUCTION :

Le schéma à contacte CONT est un langage de programmation graphique. La syntaxe de ces instructions s'inspire des schémas à relais. Un programme CONT permet de suivre facilement le flux d'énergie circulant via des entrées, des sorties et des opérations entre les barres d'alimentation. En effet, le langage de programmation CONT met à notre disposition tous les éléments nécessaires à la création d'un programme utilisateur complet. Il dispose d'un jeu d'opérations très important. Nous disposons de diverses opérations de base différentes ainsi que d'une large palette d'opérations et d'adressages. Cela vaut également pour le concept des fonctions et des blocs fonctionnels qui nous permettent de structurer clairement un programme CONT. Pour la famille, des A.P.I, SIMENS, le logiciel de programmation CONT est une partie intégrant d'un logiciel de base STEP7 (il existe d'autres logiciels pour la même famille). Il se compose d'une suite de réseaux exécutée séquentiellement par l'automate. Dessiné entre deux barres de potentiel, un réseau est un ensemble d'éléments graphiques représentant :

- Les entrées/ sorties de l'automate.
- Des fonctions d'automatisme (temporisation, compteur).
- Des opérations arithmétiques et logiques et des opérations de transfert.
- Les variables internes de l'automate (bits, mots).

Les éléments graphiques sont reliés entre eux par des connexions horizontales et verticales, définissant ainsi <<les réseaux contacts>>.

### II.4.2 Les éléments graphiques de base :

#### II.4.2.1 Les éléments de test :

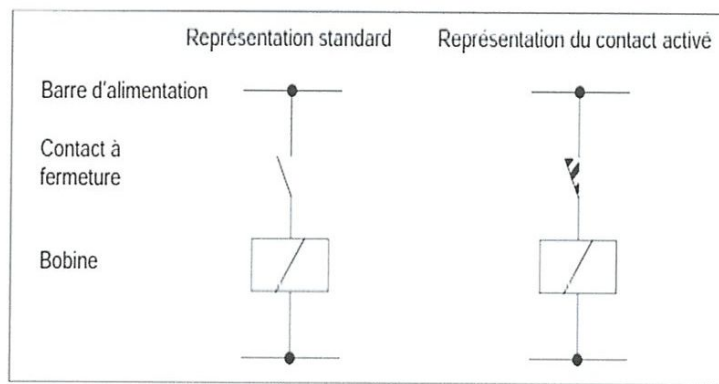
Les éléments de test sont : les contacts à fermeture et les contacts à ouverture. Le principe de ces éléments consiste à interroger l'état du signal d'entrée pour produire un résultat qui doit être copié dans le bit du résultat logique RLG. [7]

#### II.4.2.2 Contact à fermeture

La figure II.1 montre deux états d'un circuit à relais avec un contact entre une barre d'alimentation et une bobine. A l'état normal, ce contact est ouvert. S'il n'est pas activé, il demeure ouvert et son état de signal est 0 (inactivé). Si ce contact reste ouvert, l'énergie de la

barre d'alimentation ne peut pas exciter la bobine au bout du circuit. En revanche, s'il est activé – son état devient 1 –, l'énergie circule jusqu'à la bobine.

Le circuit de gauche de la figure II.1 montre un contact à fermeture tel qu'il est parfois représenté dans les schémas de circuits à relais. Dans cet exemple, le circuit de droite est utilisé pour montrer que le contact a été activé et qu'il est donc fermé.



**Figure II.1** Circuit à relais avec contact à fermeture

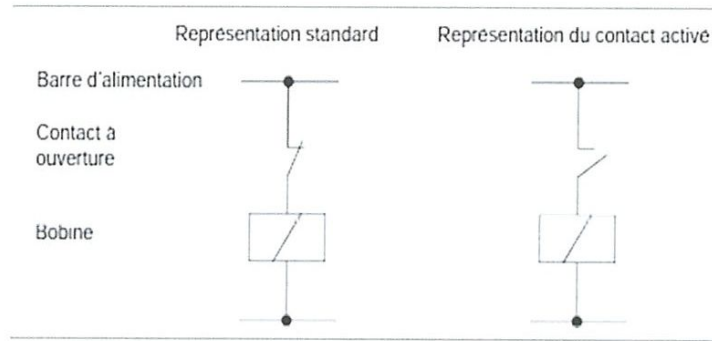
L'opération « Contact à fermeture » vous permet d'interroger l'état de signal d'un contact à fermeture. Elle détermine ainsi si l'énergie peut traverser le contact ou non. Si c'est le cas, l'opération fournit 1 comme résultat ; si ce n'est pas le cas, elle fournit 0 comme résultat. L'opération peut ensuite soit sauvegarder ce résultat, soit le combiner.

#### II.4.2.3 Contact à ouverture

La figure II.2 montre deux états d'un circuit à relais avec un contact entre une barre d'alimentation et une bobine. A l'état normal, ce contact est fermé. S'il n'est pas activé, il demeure fermé et son état de signal est 0 (inactivé). Si ce contact reste fermé, l'énergie de la barre d'alimentation peut traverser le contact et exciter la bobine au bout du circuit. En revanche, s'il est activé – son état devient 1 –, le contact s'ouvre interrompant ainsi le flux d'énergie vers la bobine.



Le circuit de gauche de la figure II.2 montre un contact à ouverture tel qu'il est parfois représenté dans les schémas de circuits à relais. Dans cet exemple, le circuit de droite est utilisé pour montrer que le contact a été activé et qu'il est donc ouvert.



**Figure II.2** Circuit à relais avec contact à ouverture

L'opération « Contact à ouverture » vous permet d'interroger l'état de signal d'un contact à ouverture. Elle détermine ainsi si l'énergie peut traverser le contact ou non. Si c'est le cas, l'opération fournit 1 comme résultat ; si ce n'est pas le cas, elle fournit 0 comme résultat. L'opération peut ensuite soit sauvegarder ce résultat, soit le combiner.

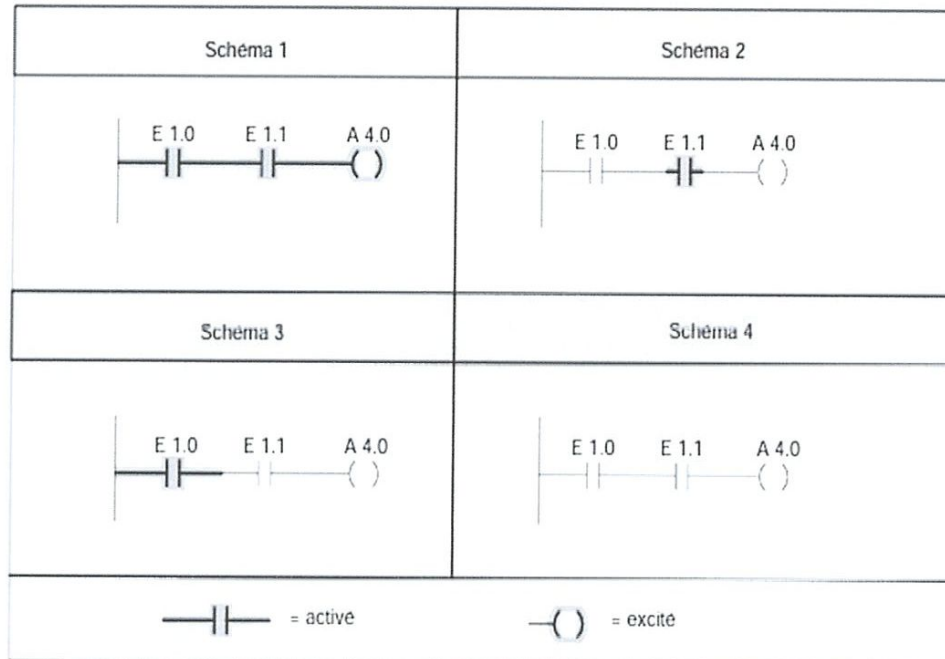
#### II.4.2.4 Connexion de contacts en série

La figure II.3 montre une séquence combinatoire d'opérations CONT qui représente deux contacts à fermeture, connectés en série à une bobine. Les contacts sont identifiés par E pour « entrée » et la bobine par A pour « sortie ». Lorsque les deux contacts de la séquence combinatoire sont activés (c'est-à-dire fermés), l'énergie peut circuler de la barre d'alimentation à travers chaque contact pour exciter la bobine au bout du circuit. Ainsi, l'énergie atteint la bobine lorsque les deux contacts E 1.0 et E 1.1 sont activés.

Les deux contacts sont activés dans le schéma 1. Activer un contact à fermeture ferme ce contact : l'énergie peut circuler de la barre d'alimentation à travers chaque contact fermé afin d'exciter la bobine au bout du circuit.

Dans les schémas 2 et 3, l'énergie ne peut pas circuler jusqu'à la bobine – qui n'est donc pas excitée – car l'un des deux contacts n'est pas activé.

Aucun contact n'est activé dans le schéma 4. Les deux contacts restent ouverts et l'énergie ne peut pas circuler vers la bobine qui n'est donc pas excitée.



*Figure II.3* Programmation de contacts en série avec « Contact à fermeture »

#### II.4.2.5 Utilisation de « Contact à fermeture » en série

La figure II.3 présente un schéma CONT avec lequel vous pouvez programmer deux contacts à fermeture connectés en série à une bobine. La première opération « Contact à fermeture » dans la séquence combinatoire interroge l'état de signal du premier contact dans la série (entrée E 1.0) et fournit le résultat 0 ou 1 en conséquence. 1 comme résultat signifie que le contact est fermé et que l'énergie disponible peut traverser le contact. 0 comme résultat signifie que le contact est ouvert, interrompant le flux d'énergie disponible au contact.

La première opération « Contact à fermeture » copie ce 1 ou ce 0 dans le bit RLG (résultat logique) du mot d'état de l'automate programmable.

La seconde opération « Contact à fermeture » dans la séquence combinatoire interroge l'état de signal du second contact dans la série (E 1.1) et fournit un résultat égal à 1 ou à 0 selon que ce contact est ouvert ou fermé. Puis, elle combine le résultat de l'interrogation de l'état de signal du second contact à la valeur figurant dans le bit RLG. Le résultat de cette combinaison – 0 ou 1 – remplace l'ancienne valeur dans le bit RLG du mot d'état. L'opération « Sortie » affecte cette nouvelle valeur à la bobine (sortie A 4.0).

On peut représenter les résultats possibles d'une telle combinaison dans une table de vérité, 1 signifiant « vrai » et 0 « faux ». Les combinaisons possibles et leurs résultats sont résumés dans le tableau ; « contact fermé » et « flux d'énergie » correspondent à « vrai » et « contact ouvert » et « pas de flux d'énergie » correspondent à « faux ».

**Tableau 1** : Table de vérité ET

Si le résultat fourni par l'interrogation de l'état de signal du contact E 1.0 est	et que le résultat fourni par l'interrogation de l'état de signal du contact E 1.1 est	le résultat de la combinaison présentée à la figure 6-3 est
1 (contact fermé)	1 (contact fermé)	1 (flux d'énergie)
0 (contact ouvert)	1 (contact fermé)	0 (pas de flux d'énergie)
1 (contact fermé)	0 (contact ouvert)	0 (pas de flux d'énergie)
0 (contact ouvert)	0 (contact ouvert)	0 (pas de flux d'énergie)

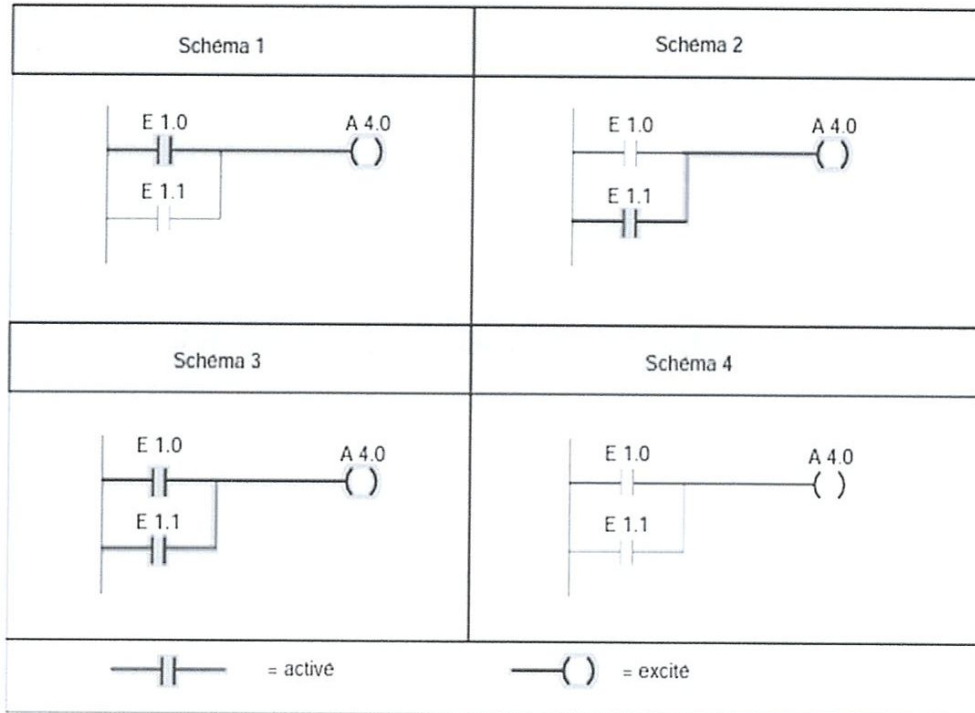
#### II.4.2.6 Connexion de contacts en parallèle

La figure II.4 montre une séquence combinatoire d'opérations CONT dans laquelle deux contacts à fermeture sont connectés en parallèle à une bobine. Les contacts sont identifiés par E pour « entrée » et la bobine par A pour « sortie ». Activer un contact à fermeture ferme ce contact. Si l'un des deux contacts de la séquence combinatoire est activé (c'est-à-dire fermé), l'énergie peut circuler via E 1.0 ou E 1.1 de la barre d'alimentation jusqu'à la bobine au bout du circuit et l'exciter. Si les deux contacts sont fermés, l'énergie parvient également à la bobine qu'elle excite.

Dans les schémas 1 et 2, un contact est activé et l'autre pas. Activer un contact à fermeture ferme ce contact : l'énergie peut circuler de la barre d'alimentation à travers le contact fermé jusqu'à la bobine au bout du circuit. Comme les deux contacts sont connectés en parallèle, il suffit que l'un d'eux soit fermé pour que l'énergie parvienne à la bobine au bout du circuit et l'excite.

Dans le schéma 3, les deux contacts sont activés : l'énergie peut traverser les deux contacts fermés et exciter la bobine au bout du circuit.

Aucun contact n'est activé dans le schéma 4. Ils restent tous deux ouverts et l'énergie ne peut pas circuler vers la bobine qui n'est donc pas excitée.



**Figure II.4** Programmation de contacts en parallèle avec « Contact à fermeture »

La figure II.4 présente un schéma CONT avec lequel vous pouvez programmer deux contacts à fermeture connectés en parallèle à une bobine. La première opération « Contact à fermeture » dans la séquence combinatoire interroge l'état de signal du premier contact (entrée E 1.0) et fournit le résultat 0 ou 1 en conséquence. 1 comme résultat signifie que le contact est fermé et que l'énergie disponible peut traverser le contact. 0 comme résultat signifie que le contact est ouvert, interrompant le flux d'énergie disponible au contact. La première opération « Contact à fermeture » copie ce 1 ou ce 0 dans le bit RLG (résultat logique) du mot d'état de l'automate programmable.

La seconde opération « Contact à fermeture » dans la séquence combinatoire interroge l'état de signal du second contact (E 1.1) et fournit un résultat égal à 1 ou à 0 selon que ce contact est ouvert ou fermé. Puis, elle combine le résultat de l'interrogation de l'état de signal du second contact à la valeur figurant dans le bit RLG. Le résultat de cette combinaison – 0 ou 1

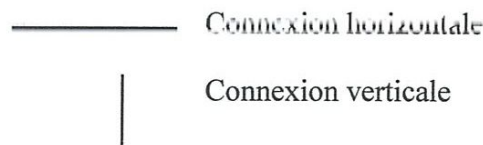
– remplace l’ancienne valeur dans le bit RLG du mot d’état. L’opération « Sortie » affecte cette nouvelle valeur à la bobine (sortie A 4.0).

On peut représenter les résultats possibles d’une telle combinaison dans une table de vérité, 1 signifiant « vrai » et 0 « faux ». Les combinaisons possibles et leurs résultats sont résumés dans le tableau suivant; « contact fermé » et « flux d’énergie » correspondent à « vrai » et « contact ouvert » et « pas de flux d’énergie » correspondent à « faux ».

**Tableau 2: Table de vérité OU**

Si le résultat fourni par l’interrogation de l’état de signal du contact E 1.0 est	et que le résultat fourni par l’interrogation de l’état de signal du contact E 1.1 est	le résultat de la combinaison présentée à la figure 6-4 est
1 (contact fermé)	0 (contact ouvert)	1 (flux d’énergie)
0 (contact ouvert)	1 (contact fermé)	1 (flux d’énergie)
1 (contact fermé)	1 (contact fermé)	1 (flux d’énergie)
0 (contact ouvert)	0 (contact ouvert)	0 (pas de flux d’énergie)

**II.4.3 Elément de liaison :**



**II.4.4 Eléments d’action:**

- ( ) —— Bobine directe.
- (\) —— bobine inverse.
- (S) —— bobine d’élanement(Set)
- (R) —— bobine de déclanchement(Reset)
- (J) bobine à saut aux autre réseaux
- ( )\* } bobines sauvegardées en cas de coupure secteur ;
- (/)\* } Les états des objets bits associés sont sauvegardés
- (S)\* } durant le premier tour de cycle.
- (R)\* } —
- (SI) —— Temporisation sous forme d’impulsion

## **CHAPITRE III**

**Application :**

**Systeme convoyeur rapide**

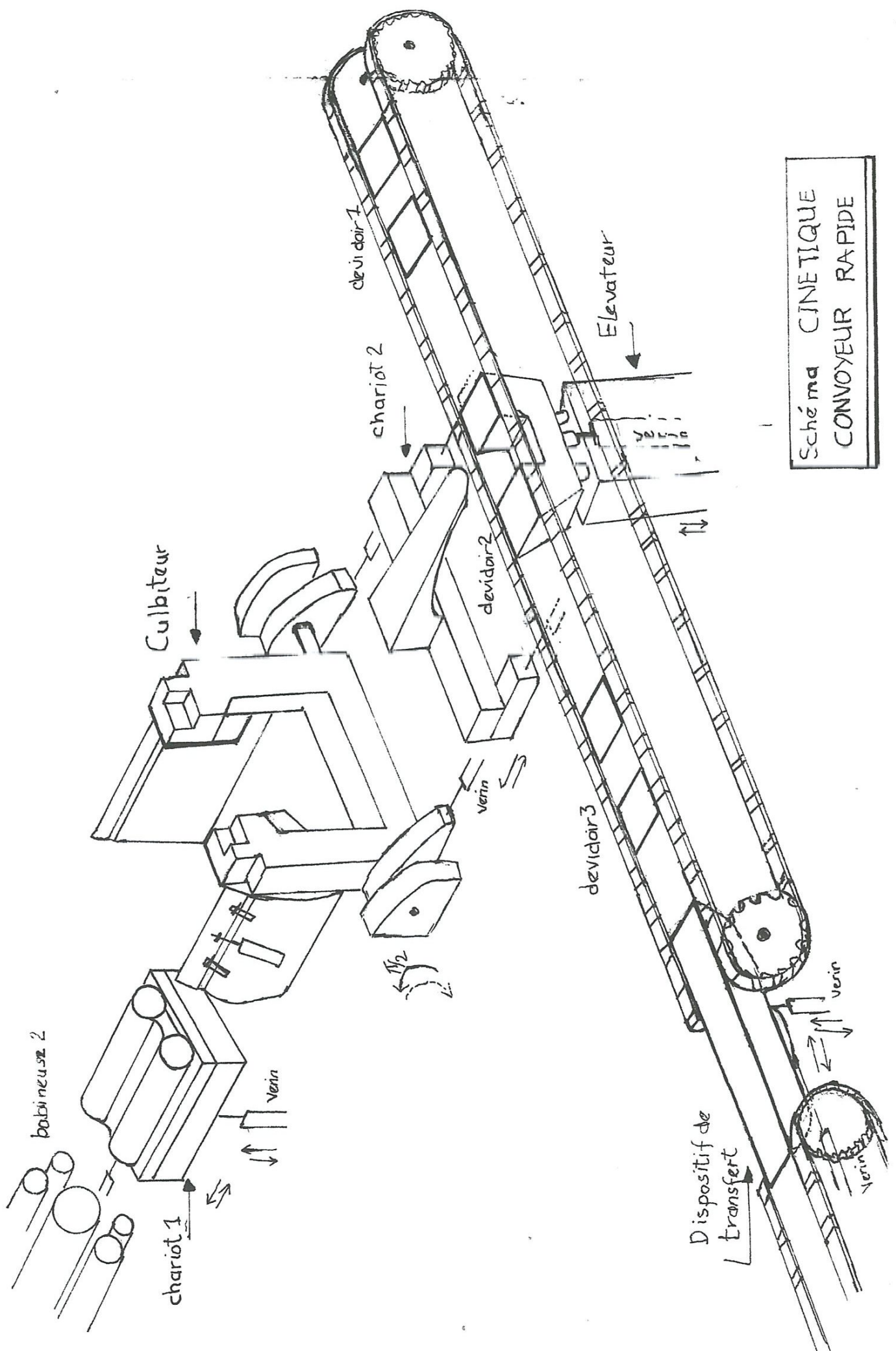


Schéma CINÉTIQUE  
CONVOYEUR RAPIDE

### III.1 - DESCRIPTION DE L'INSTALLATION :

#### III.1.1- SCHEMA CINETIQUE: (figure ci-contre)

Le système se compose de : 3 bobineuses et un convoyeur à chaîne.

Chaque bobineuse possède un système de transfert des bobines jusqu'à la chaîne du convoyeur rapide.

Ce système de transfert est constitué de :

- Un chariot (1) qui transporte la bobine de la bobineuse jusqu'au culbuteur.
- Deux mandrins de serrage solidaires au culbuteur pour le serrage de la bobine .
- Un culbuteur pour mettre la bobine en position verticale en pivotant d'un angle de  $\pi/2$ .
- Un chariot (2) pour transporter la bobine jusqu'à la chaîne.
- Un élévateur pour poser la bobine sur la chaîne du convoyeur (sur le dévidoir).

L'installation possède :

- Une chaîne entraînée par un moteur, pour déplacer les bobines jusqu'au dispositif de transfert.
- Un dispositif de transfert qui réalise le transfert des bobines au convoyeur lent .

#### III.1.2-DESCRIPTION DU FONCTIONNEMENT :

Lorsque l'une des trois bobineuses produit une bobine, cette dernière est d'abord libérée des mandrins, chariot (1) la ramène jusqu'au culbuteur et il revient à sa position initial. Le culbuteur après serrage de la bobine, pivote d'un angle de  $90^\circ$  pour mettre la bobine en position verticale. Le chariot (2) prend la bobine et se déplace en avant jusqu'au l'élévateur, qui à son tour la dépose sur la chaîne du convoyeur qui la transporte jusqu'au dispositif de transfert et après au convoyeur lent.

Tout les mouvements depuis la bobineuse jusqu'à la chaîne sont assurés par des vérins commandés par des électrovannes.

La translation de la chaîne est assurée par un moteur à CC

La commande de l'ensemble se fait par un opérateur en agissant sur des boutons placés sur un pupitre de command



**III.2- CRITIQUES SUR L'INSTALLATION ACTUELLE :****III.2.1- INCONVENIENT DE LA COMMANDE:**

Cette installation est conçue pour une production d'une cadence d'une bobine chaque 3mn, et la vitesse de production dépend de la demande.

Hors actuellement cette installation n'arrive pas à suivre ce rythme à cause des arrêts successifs causés par des pannes (qui peut être des fois au niveau d'autre installation) et aussi la difficulté de la commande.

Au niveau de cabine de commande, là où il y a le pupitre, l'opérateur exécute beaucoup de tâches ; ce qui rend son rôle très difficile.

- Il doit surveiller l'arrivée de la bande issue du train finisseur
- Par conséquent il doit sélectionner la bobineuse et la préparer.
- Il doit commander ensuite tous les mouvements du transfert de la bobine à partir du pupitre qui contient à peu près une vingtaine de boutons, actionneurs sélecteurs,.....etc.
- En même temps il doit surveiller l'arrivée de l'autre bande pour refaire le cycle.
- En plus l'installation est datée des années 1970 donc les circuits de commande sont très anciens par rapport à la technologie actuelle, ce qui rend la maintenance et le dépannage très difficile et parfois impossible ; parce que le constructeur lui-même ne fabrique pas ces anciens équipements.
- Toutes ces difficultés reflètent directement sur la production, par des pannes (des arrêts) et le non-respect du temps d'exécution du programme de production, et par conséquent une perte d'argent.

**III.2.2- SOLUTION PROPOSEE:**

Pour minimiser(le plus possible) ces problèmes, la solution consiste à remplacer la partie commande moderne architecturé autour d'un automate, qui lui-même peut être ouvert sur d'autre automates ou autres systèmes informatiques de processus ou de gestion et supervision, dont le développement Soft-commande est plus facile à développer et maintenir.

**III.2.3- AVANTAGES A APPORTER POUR L'AUTOMATISATION DU SYSTEME :**

Suret  de fonctionnement.

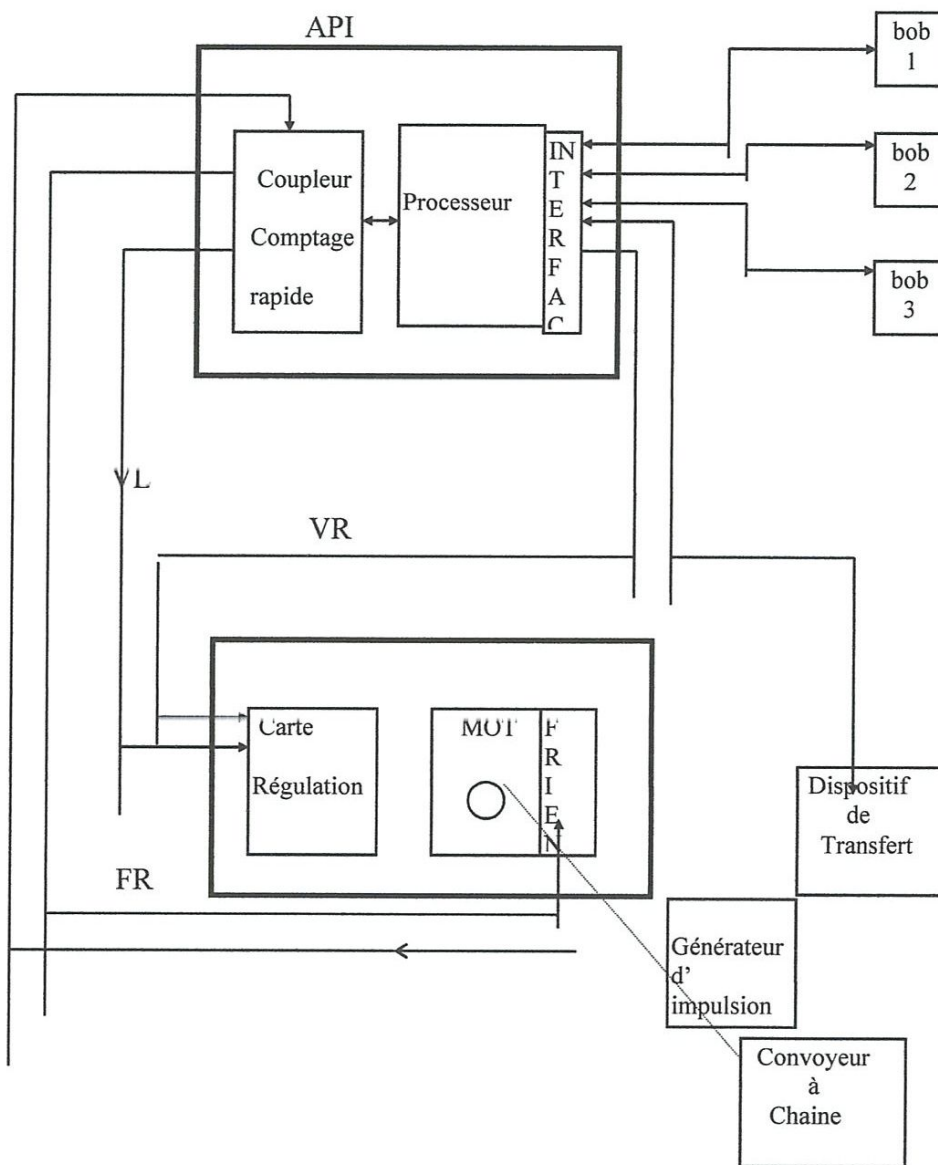
- Economie
- Souplesse d'emploi
- L'entretien est facile pour un personnel form  pour.
- Accessible   tout pour une meilleure maintenance Soft.
- D tection d'anomalies se fait tr s rapidement et par cons quent d tection rapide de l'organe d fectueux.
- Disponibilit  de la pi ce de rechange au niveau des constructeurs.
- Les modifications du fonctionnement se fait tr s rapidement par programme.

**III.3 SOLUTION DU PROBLEME****III.3.1- TRAVAIL DEMANDE :**

Etudier la possibilit  d'une substitution de la logique c bl e ( du syst me convoyeur rapide) par une logique programm e pilot  par un API qui prend en charge tous les op rateurs commandables, avec une sp cification tel que l'utilisation d'une technique de commande de vitesse de la chaine du convoyeur bas e sur le comptage d'impulsion provenant d'un g n rateur li  avec le moteur .

**III.3.2.DESCRPTION DU FONCTIONNEMENT DU CONVOYEUR RAPIDE**  
(solution propos e)

\* Sch ma bloc :



• **Données de l'installation :**

Vitesse du moteur = 725 tr/mn = 12,08 tr/s

Vitesse du convoyeur = 0,55 m/s

Le gain du réducteur est de 82 c.à.d. 82 tour du moteur correspond à un tour de la roue d'entraînement

La fréquence du générateur est de 6,04 kHz

1 impulsion correspond à 0,089 mm

La distance entre le dévidoir 1 et le point d'arrêt sur le dispositif de transfert est fixe, elle est : 17540mm

- Pour le dévidoir 2 elle est de : 10225mm
- Pour le dévidoir 3 elle est de : 2225mm
- Un générateur d'impulsion est connecté au moteur de façon à ce qu'il débute en même temps que le moteur se met en marche.
- Ce générateur donne 500 impulsions par tour du moteur
- Une rotation de la roue à chaîne correspond à 3656 mm voie de transport.
- On calculant le nombre d'impulsion correspondant à chaque dévidoir on trouve les résultats suivants:

dévidoir 1 = 17540mm correspond à  $\sim 197078$  impulsion  $\sim 17539,94$ mm

dévidoir 2 = 10225mm  $\longrightarrow$  114887 impulsion  $\sim 10224,94$ mm

dévidoir 3 = 2225 mm  $\longrightarrow$  25000 impulsion

- Précision du positionnement (centrage de la bobine) =  $\pm 1$  impulsion

1 imp = 0,089 mm

- Donc l'écart  $\Delta x = \pm 0,089$  mm est acceptable parce qu'il est négligeable devant les dimensions de la bobine  $\sim 2000$ mm de diamètre.
- Le moteur utilisé est à CC de puissance 11kw
- **Fonctionnement de l'ensemble:**

- A l'état initial les trois dévidoirs sont vides et le moteur est à l'arrêt.

- Dès qu'il y a détection bobine sur le convoyeur le système commence à fonctionner selon les trois cas suivants:

1er CAS : La bobine est sur le dévidoir 1

- Le convoyeur démarré avec une vitesse max 0,55 m/s
- En même temps le générateur d'impulsion, de débit et un compteur approprié se met en marche .
- On choisit le dévidoir 3 comme point de référence pour le changement de vitesse qui correspond à un nombre d'impulsions calculé selon la distance entre le dévidoir 1 et 3 (197078 - 25000) impulsion.

- Dès que le compteur atteint cette valeur, on lance la vitesse lente qui persiste pendant 25000 impulsions, correspondantes à la distance entre le dévidoir 3 et le point final sur le dispositif de transfert, ce qui provoque l'arrêt du moteur et l'excitation des freins mécaniques, pour l'arrêt absolu de la chaîne.
- Lorsque cette dernière est à l'arrêt, le dispositif de transfert s'occupe du transfert de la bobine au convoyeur lent.

2eme CAS : la bobine est sur le dévidoir 2 :

Le même fonctionnement sauf que le nombre d'impulsion correspondant à la vitesse maxi change, il devient (114887 - 25000)impulsion.

3eme CAS : la bobine est sur le dévidoir 3 :

Dans ce cas le moteur démarré directement avec la vitesse lente pendant un membre d'impulsion de 25000 impulsion, et le reste sera comme les deux Premier cas.

### III.3.3. Système convoyeur rapide:

#### III.3.3.1 Description de la partie opérative.

Le système est composé de :

#### Bobineuse 1 :

- un chariot transporteur ( CH 1,1 ) qui se déplace verticalement et horizontalement sur des railles .
- un culbuteur (CB1) muni d'un dispositif de serrage, et qui pivote d'un angle de  $\pi/2$ .
- un 2<sup>eme</sup> chariot transporteur (CH1, 2) qui se déplace horizontalement sur des railles .
- un élévateur (EV1) qui dépose les bobines sur le dévidoir Dv1.
- Un vérin (VRC 1, 1) pour le déplacement vertical du chariot CH1 1
- Un vérin (VRC1, 1') // // horizontal // //
- Deux vérins VRS1, 1 et VRS1, 1` pour le serrage des bobines.
- Deux vérins VRB1, 2 et VRB1, 1` pour le pivotement du culbuteur CB1
- Deux vérins VRC1, 2 et VRC1, 2` pour le déplacement horizontal du chariot CH1,2.
- Un vérin VRE1 pour l'élévateur 1 (déplacement vertical)

#### **Bobineuse 2:**

- Un chariot transporteur CH1, 2 qui se déplace verticalement et horizontalement sur des railles .

- Un culbuteur CB2 muni d'un dispositif de serrage et qui pivote d'un angle de  $\pi/2$ .
- Un 2eme chariot transporteur CH2,2 qui se déplace horizontalement sur des railles .
- Un élévateur EV2 qui dépose des bobines sur le dévidoir DV2.
- Un vérin VRC2, 1 pour le déplacement vertical du chariot CH2,1.
- Un vérin VRC2, 1' // horizontal // //
- Deux vérins VRS2, 1 et VRS2,1' pour le serrage des bobines .
- Deux vérins VRB2, 1 et VRB2,1' pour le pivotement du culbuteur CB2.
- Deux vérins VRC2, 2 et VRC2,2' pour le déplacement horizontal du chariot CH2,2 .
- Un vérin VRE2 pour l'élévateur 2 (déplacement vertical).

### La bobineuse 3 :

- Un chariot transporteur CH1, 3 qui se déplace verticalement et horizontalement sur les railles .
- Un culbuteur CB3 muni d'un dispositif de serrage et qui pivote d'un angle de  $\pi/2$ .
- Un 2eme chariot transporteur CH2, 3 qui se déplace horizontalement sur des railles.
- Un élévateur qui dépose les bobines sur le dévidoir DV3.
- Un vérin pour le déplacement vertical du chariot CH1, 3.
- // // horizontal // //
- Deux vérins pour le serrage des bobines.
- // // le pivotement du culbuteur CB3.
- Deux vérins pour le déplacement horizontal du chariot CH2, 3.
- Un vérin VRE3 pour l'élévateur 3 (déplacement vertical).

### Le Convoyeur :

- Une chaine mobile CHA entraînée par un moteur.
- Un dispositif de transfert DT entre le convoyeur rapide et le convoyeur lent.
- Un verin VRDT1 pour le déplacement vertical de DT.
- // VRDT1' // // horizontal de DT.
- Un générateur d'impulsion lié à la roue de rotation qui entraine la chaine.
- Une carte comptage rapide des impulsions.
- Une carte de régulation de vitesse du moteur.
- Un dispositif de freinage électromécanique.

Les capteurs :

- Un capteur de fin bobinage (bobine prête) pour la bobineuse 1
- // // // // 2
- // // // // 3
- // de position basse du chariot 1 bobineuse 1
- // // haute // // //
- // // arrière // // //
- // // avant // // //
- // // basse // // // 2
- // // haute // // //
- // // arrière // // //
- // // Avant // // //
- // // Basse // // // 3
- // // haute // // //
- // // arrière // // //
- // // Avant // // //
- deux capteurs de serrage de la bobine culbuteur 1
- // // de serrage // // 1
- // // de serrage // // 2
- // // de serrage // // 2
- Deux capteurs de serrage bobinent culbuteur 3
- // // desserrage // // 3
- // // de position initial de culbuteur 1
- // // de position finale culbuteur 1
- // // de position initiale culbuteur 2
- // // de position finale culbuteur 2
- // // de position initiale culbuteur 3
- // // de position finale culbuteur 3
- un capteur de position arrière chariot 2 bobineuses 1
- // // Avant // // // 1
- // // Arrière // // // 2
- // // Avant // // // 2

•	//	//	Avant	//	//	//	3
•	//	//	Basse		Elévateur		1
•	//	//	haute		//		1
•	//	//	arrière	//	//	//	3
•	//	//	Basse		//		2
•	//	//	haute		//		2
•	//	//	Basse		//		3
•	//	//	haute		//		3
•	//	//	Avant	dispositif de transfert.			
•	//	//	arrière	//	//	//	
•	//	//	Basse	//	//	//	
•	//	//	haute	//	//	//	

**Le pupitre :**

- Un bouton marche/arrêt.
- Un sélecteur mode de marche ( auto - C/C- Manu ) .
- lampe verte -----> fonctionnement normal .
- lampe rouge -----> anomalies au niveau de l'installation.
- Une partie du pupitre contenant des boutons et actionneurs pour chaque dispositif de l'installation pour la marche manuelle.

**III.3.3.2- Description de la partie commande :(PC)**

Elle est équivalente pour les trois bobineuses.

- **Cycle de fonctionnement**
- Si les conditions initiales sont prêtes (chariot 1 en position basse arrière et tous les autre dispositifs en position initiale, le convoyeur est à l'arrêt, et détection bobine prête)
- monte chariot transporteur 1 jusqu'à la position haute.
- libération de la bobine.
- Déplacement avant du chariot transporteur 1 jusqu'à la position finale.
- serrage de la bobine.
- descente charriot 1 jusqu'à position basse (dépôt bobine sur culbuteur).



- pivotement du culbuteur jusqu'à position finale (angle de  $\pi/2$ ), (dépôt bobine sur chariot transporteur 2).
- desserrage de la bobine.
- retour chariot 1 en position initiale.
- déplacement avant du chariot 2 jusqu'à position finale.
- monte élévateur jusqu'à position haute (prise bobine du chariot 2).
- pivotement culbuteur jusqu'à position initiale.
- retour chariot 2 en position initiale.
- descente élévateur jusqu'à position basse (dépôt bobine sur le dévidoir).
- déplacement de la chaîne selon les 3 modes suivants :

**Si la bobine est sur le dévidoir 1 :**

démarrage du convoyeur avec vitesse max, en même temps, le générateur d'impulsions débite, et le comptage se fait par une carte connectée avec l'automate ; qui lance la vitesse lente dès que le nombre d'impulsions atteint (197078- 25000), (correspondante au dévidoir 3).

lorsque le nombre d'impulsions devient 25000 (valeur finale correspondante à la position d'arrêt sur le dispositif de transfert) l'automate provoque l'arrêt du moteur ainsi que le freinage mécanique pour l'arrêt absolu (Remise à zéro du compteur).

**Si la bobine est sur le dévidoir 2 :**

démarrage du convoyeur avec vitesse max, en même temps le générateur d'impulsions débite, le comptage se fait par la carte du comptage connectée avec l'automate, qui lance la vitesse lente dès que le nombre d'impulsions atteint (114887 - 25000) correspond au passage de la bobine sur le dévidoir 3

lorsque le nombre d'impulsions devient 25000 valeurs finales correspondantes à la position d'arrêt, l'automate provoque l'arrêt du moteur, ainsi que le freinage mécanique pour l'arrêt absolu (remise à zéro du compteur).

**Si la bobine est sur le dévidoir 3 :**

démarrage du convoyeur avec vitesse lente, en même temps le générateur d'impulsions débite, le comptage se fait par la carte. Lorsque le nombre d'impulsions devient 25000 (valeur finale correspondante à la position d'arrêt), l'automate provoque l'arrêt du moteur, ainsi que le freinage mécanique pour l'arrêt absolu du convoyeur (RAZ du compteur).

Après un temps de l'arrêt absolu du convoyeur (la bobine est sur le point de transfert sur le dispositif de transfert).

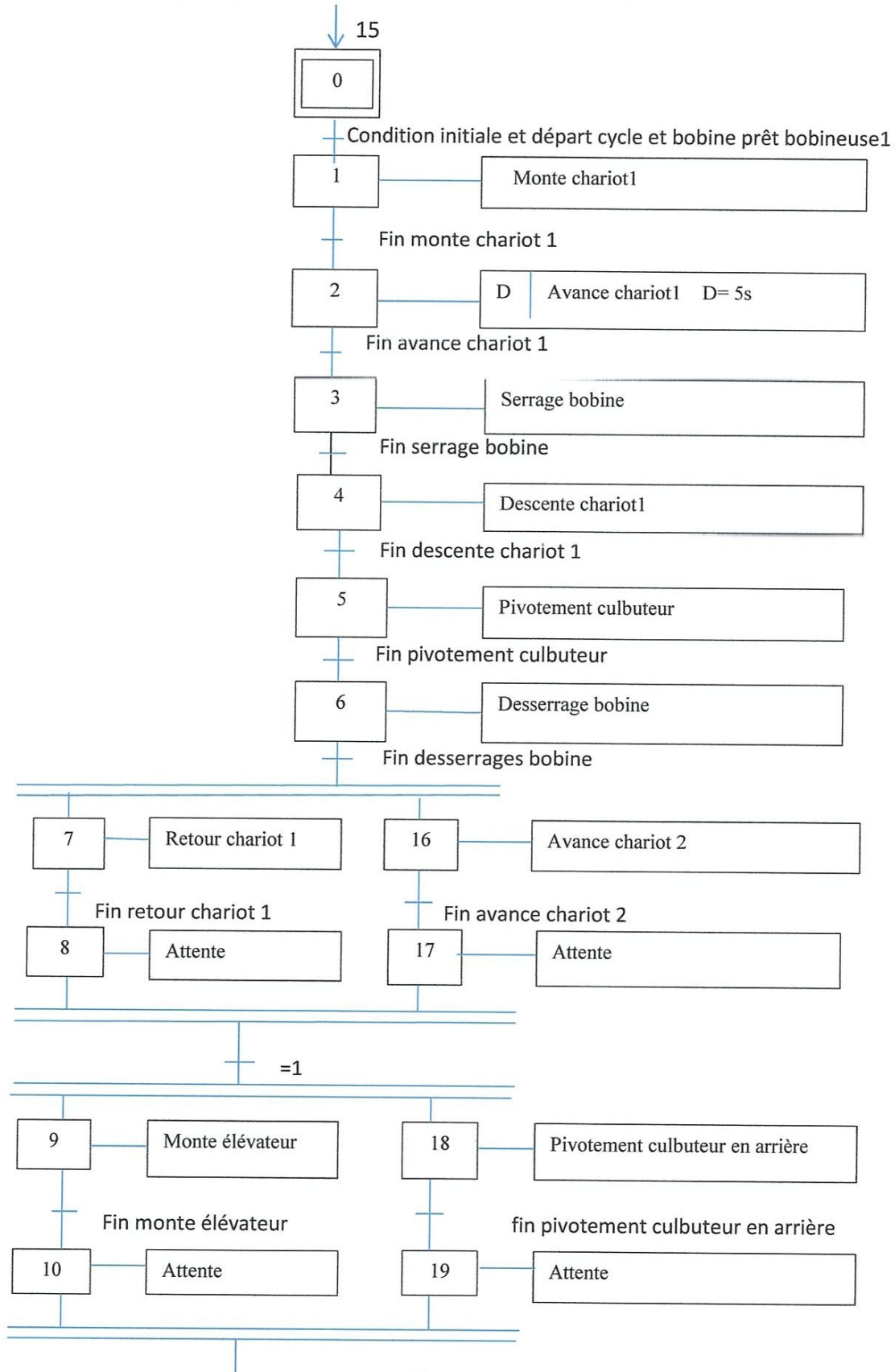
monte du dispositif jusqu'à position haute (prise bobine).

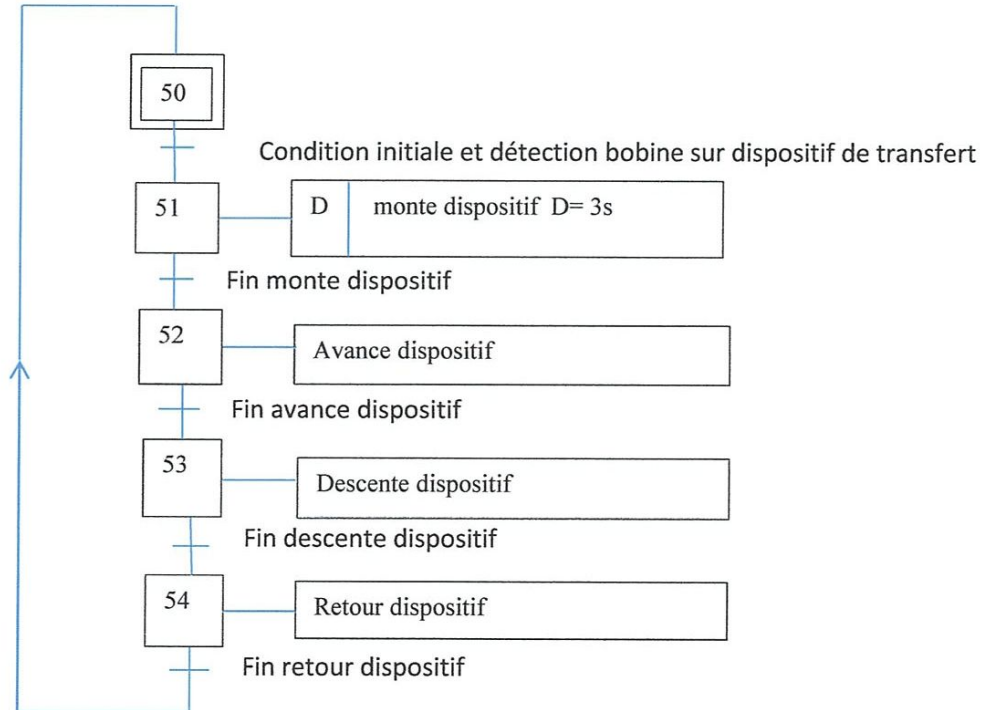
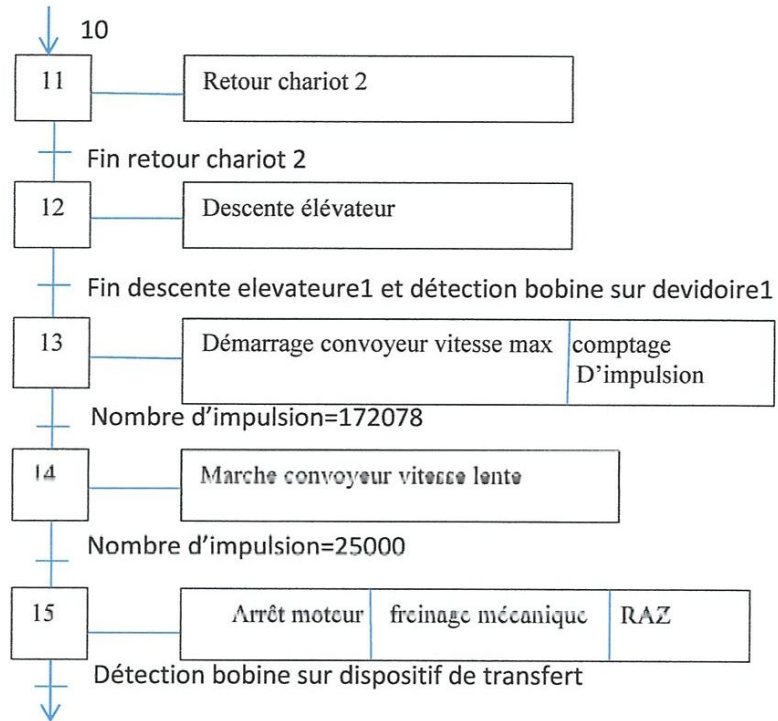
avance du dispositif jusqu'à position finale.

- descente du dispositif jusqu' a position basse (dépôt bobine sur convoyeur lent).
- retour du dispositif a la posait

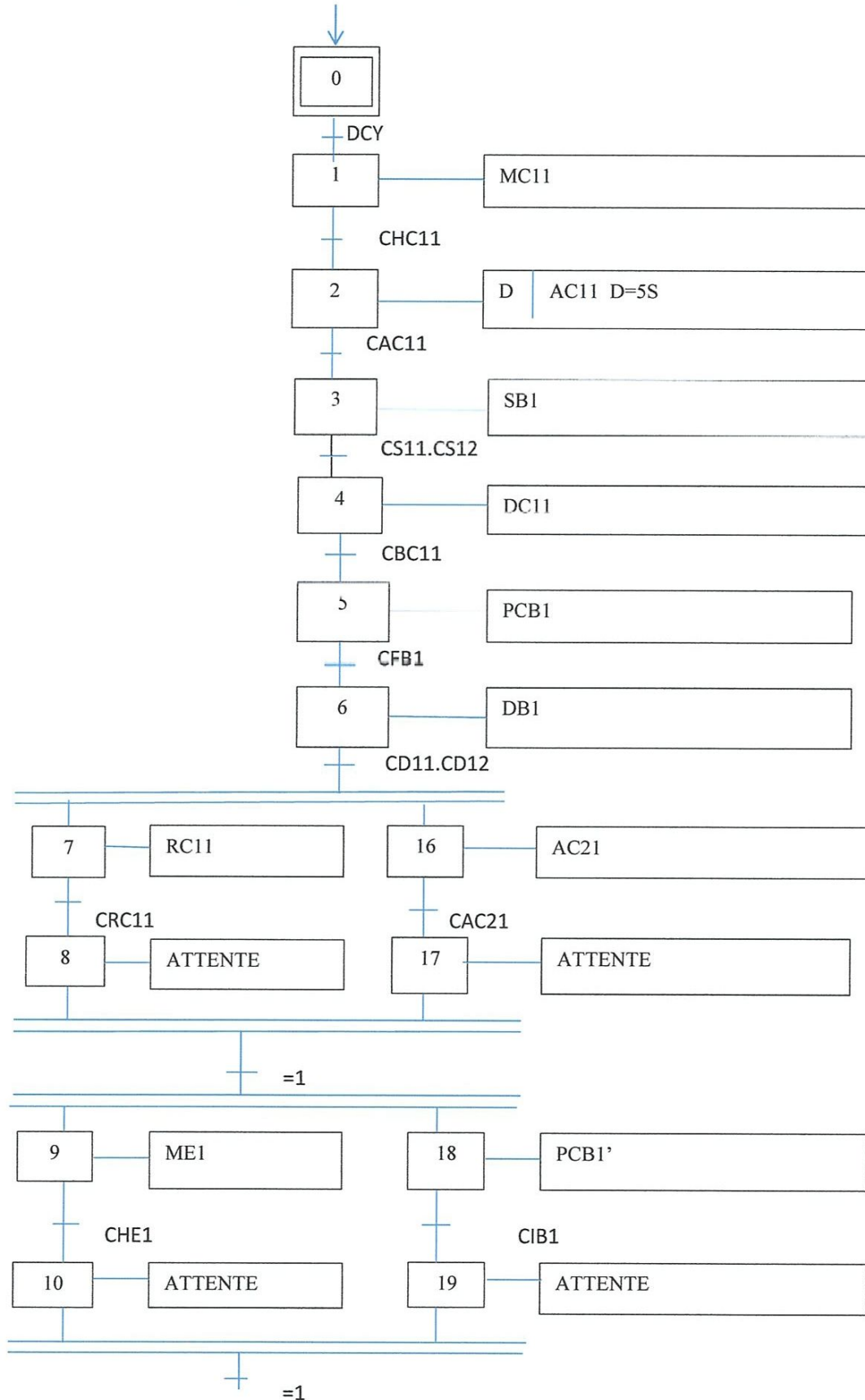
II.4 Grafcet de système

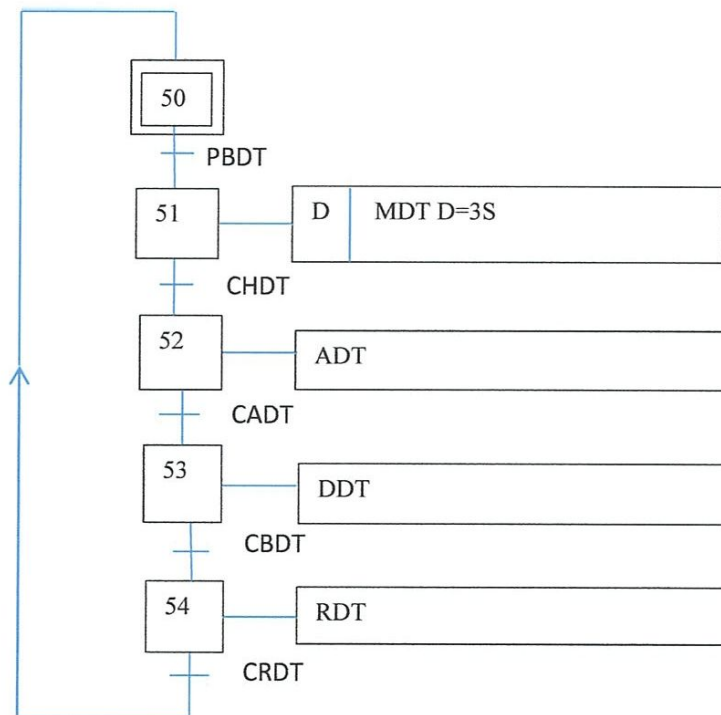
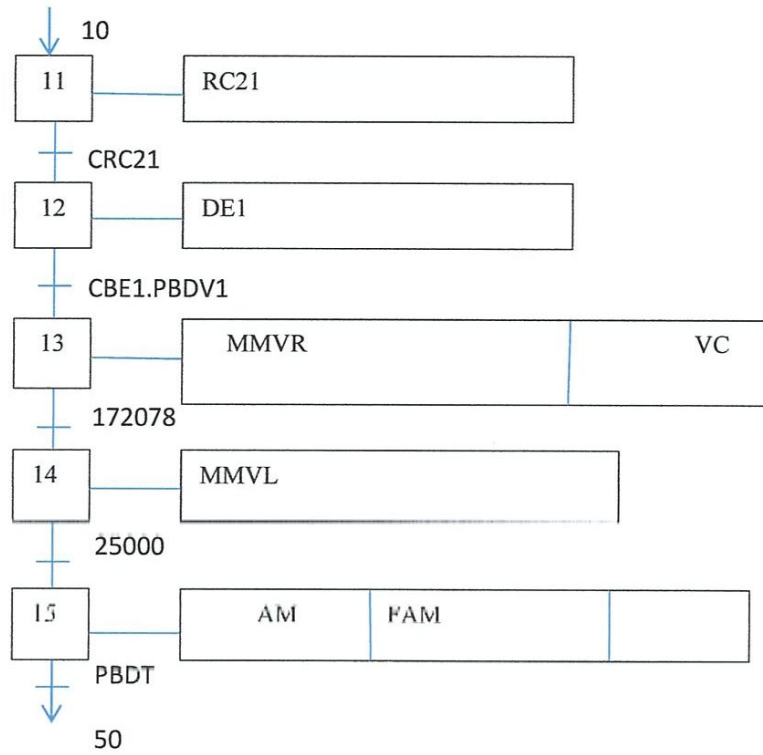
III.4.1 Grafcet niveau 1 bob 1





III.4.2 Grafset niveau 2 bob 1 :





**III.4.3 Grafctet niveau 1 bob 2 :**

Même grafctet de niveau 1 bobine 1

**III.4.4 Grafctet niveau 2 bob 2**

Même grafctet de niveau 2 bobine 1

**III.4.5 Grafctet niveau 1 bob 3 :**

Même grafctet de niveau 1 bobine 1

**III.4.6 Grafctet niveau 1 bob 3 :**

Même grafctet de niveau 2 bobine 1

**III.5 Ressensement des entrees :**

N°	Entrée	Mnémonique	Désignation
01	- un bouton	M/A	Marche / Arrêt
02	un sélecteur mode de marche	*Auto * C/C *Manu	Marche automatique Cycle /cycle Marche manuelle
03	Un capteur fin de course	CBP1	Bobine prête bobineuse 1
04	un capteur fin de course	CBP2	Bobine prête bobineuse 2
05	un capteur fin de course	CBP3	Bobine prête bobineuse 3
06	un capteur fin de course	CBC1, 1	position basse, chariot 1 bobineuse 1
07	un capteur fin de course	CHC1,1	Position haute chariot 1, bobineuse 1
08	un capteur fin de course	CRC1,1	Position arrière

			chariot 1, bobineuse 1
09	un capteur fin de course	CBC1, 2	Position basse chariot 1, bobineuse 2
10	un capteur fin de course	CHC1, 2	Position haute chariot1, bobineuse 2
11	un capteur fin de course	CRC1, 2	Position arrière chariot 1, bobineuse 2
12	un capteur fin de course	CAC1, 2	Position avant chariot 1, bobineuse 2
13	un capteur fin de course	CBC1, 3	Position base chariot 1, bobineuse 3
14	un capteur fin de course	CHC1, 3	Position haute chariot 1, bobineuse 3
15	un capteur fin de course	CAC1, 1	position avant chariot 1, bobineuse 1

N°	entrée	Mnémonique	Description
16	un capteur FC	CRC 1,3	Position arrière chariot 1, bobineuse 3
17	un capteur FC	CAC 1,3	Position avant Chariot 1, bobineuse 3
18	un capteur FC	CS11 et CS12	Serrage bobine culbuteur 1
19	un capteur FC	CD11 et CD12	Desserrage bobine culbuteur 1



20	un capteur FC	CS21 et CS22	Serrage bobine culbuteur 2
21	un capteur FC	CD21 et CD22	Desserrage bobine culbuteur 2
22	un capteur FC	CS31 et CD32	Serrage bobine culbuteur 3
23	un capteur FC	CD31 et CD32	Desserrage bobine culbuteur 3
24	un capteur FC	CIB1	Position initiale culbuteur 1
25	un capteur FC	CFB1	Position finale culbuteur 1
26	un capteur FC	CIB2	Position initiale culbuteur 2
27	un capteur FC	CFB2	Position finale culbuteur 2
28	un capteur FC	CIB3	Position initiale culbuteur 3
29	un capteur FC	CFB3	Position finale culbuteur 3
30	un capteur FC	CRC2, 1	Position arrière chariot 2, bobineuse 1
31	un capteur FC	CAC2, 1	Position avant Chariot 2, bobineuse 1

N°	Entrée	Mnémonique	Désignation
32	un capteur FC	CRC2,2	Position arrière chariot 2, bobineuse 2
33	un capteur FC	CAC2,2	Position avant Chariot 2, bobineuse 2
34	un capteur FC	CRC2,3	Position arrière Chariot 2, bobineuse 3
35	un capteur FC	CAC2,3	Position avant chariot 2, bobineuse 3
36	un capteur FC	CBE1	Position basse élévateur 1
37	un capteur FC	CHE1	Position haute élévateur 1
38	un capteur FC	CBE2	Position basse élévateur 2
39	un capteur FC	CHE2	Position haute élévateur 2
40	un capteur FC	CBE3	Position basse élévateur 3
41	un capteur FC	CHE3	Position haute élévateur 3
42	un capteur FC	CADT	Position avant dispositif de transfert
43	un capteur FC	CRDT	Position arrière dispositif de transfert

44	un capteur FC	CBDT	Position basse dispositif de transfert
45	un capteur FC	CHDT	Position haute dispositif de transfert
46	un capteur FC	PBDV1	présence bobine, dévidoir 1
47	un capteur FC	PBDV2	présence bobine, dévidoir 2
48	un capteur FC	PBDV3	présence bobine, dévidoir 3
49	un capteur FC	PBDT	présence bobine sur dispositif de transfert

**III.6 Resensement des sorties:**

N°	Sortie	Mnémonique	Désignation
01	Lampe verte	LV	Fonction normal
02	Lampe rouge	LR	Défaut de fonctionnement
03	Avance chariot 1 Bobineuse 1	AC1, 1	déplacement horizontal
04	Monte chariot 1 Bobineuse 1	MC1, 1	déplacement vertical
05	Descente chariot 1 Bobineuse 1	DC1, 1	déplacement vertical

06	Retour chariot 1 Bobineuse 1	RC1, 1	Déplacement horizontal
07	Descente chariot 1 Bobineuse 2	DC1, 2	Déplacement vertical
08	Monte chariot 1 Bobineuse 2	MC1, 2	Déplacement vertical
09	Retour chariot 1 Bobineuse 2	RC1, 2	Déplacement horizontal
10	Avance chariot 1 Bobineuse 2	AC1, 2	Déplacement horizontal
11	Descente chariot 1 Bobineuse 3	DC1, 3	déplacement vertical
12	Monte chariot 1 Bobineuse 3	MC1, 3	Déplacement vertical
13	Retour chariot 1 Bobineuse 3	RC1,3	Déplacement horizontal
14	Avance chariot 1 Bobineuse 3	AC1, 3	Déplacement horizontal
N°	Sortie	Mnémorique	Désignation
15	Serrage bobine culbuteur 1	SB1	Serrage bobine
16	Desserrage bobine culbuteur1	DB1	Desserrage bobine
17	Serrage bobine culbuteur 2	SB2	Serrage bobine
18	Desserrage bobine	DB2	Desserrage bobine

	culbuteur2		
19	Serrage bobine culbuteur 3	SB3	Serrage bobine
20	Desserrage bobine culbuteur 3	DB3	Desserrage bobine
21	Pivotement en avant culbuteur 1	PCB1	Pivotement d'un angle $\pi/2$
22	Pivotement dans le sens inverse, culbuteur 1	PCB1'	Pivotement d'un angle $\pi/2$
23	Pivotement en avant culbuteur 2	PCB2	Pivotement d'un angle $\pi/2$
24	pivotement dans le sens inverse, culbuteur 2	PCB2'	Pivotement d'un angle $\pi/2$
25	Pivotement en avant culbuteur 3	PCB3	Pivotement d'un angle $\pi/2$
26	Pivotement en arrière culbuteur 3	PCB3'	Pivotement d'un angle $\pi/2$
27	Retour chariot 2 bobineuse 1	RC2, 1	Déplacement horizontal
28	Avance chariot 2 Bobineuse 1	AC2, 1	Déplacement horizontal
29	Retour chariot 2 bobineuse 2	RC2, 1	Déplacement horizontal
30	Avance chariot 2 bobineuse 2	AC2, 2	Déplacement horizontal
31	Retour chariot 2	RC2, 3	Déplacement horizontal

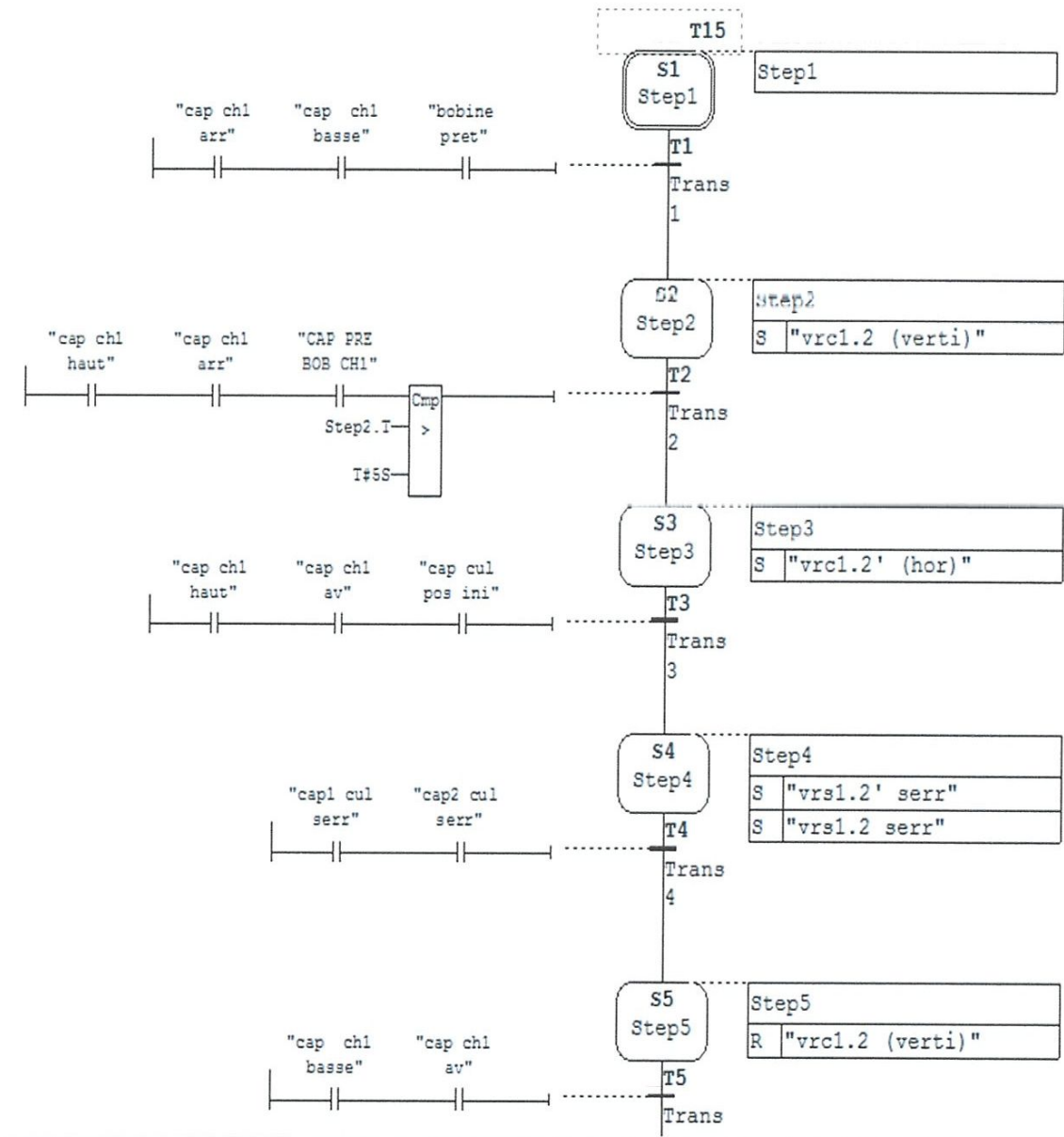
	bobineuse 3		
32	Avance chariot 2	AC2,3	Déplacement horizontal
N°	Sortie	Mnémonique	Désignation
33	monte élévateur 1	ME1	Monte élévateur
34	descente élévateur 1	DE1	Descente élévateur
35	Monte élévateur 2	ME2	Monte élévateur
36	Descente élévateur 2	DE2	Descente élévateur
37	Monte élévateur 3	ME3	Monte élévateur
38	Descente élévateur 3	DE3	Monte élévateur
39	Monte dispositif de transfert	MDT	Monte dispositif
40	Descente dispositif de transfert	DDT	Descente dispositif
41	Avance dispositif de transfert	ADT	Avance dispositif
42	Retour dispositif de transfert	RDT	Retour dispositif
43	Marche moteur vitesse rapide	MMVR	fonction max
44	Marche moteur vitesse lente	MMVL	fonction min
45	Arrêt moteur	AM	L'arrêt du convoyeur
46	Freinage mécanique	FRM	pour l'arrêt absolu
47	libération de la roue	LRM	pour le redémarrage
48	Validation comptage	VC	Début de comptage

## III.7.PROGRAMME :

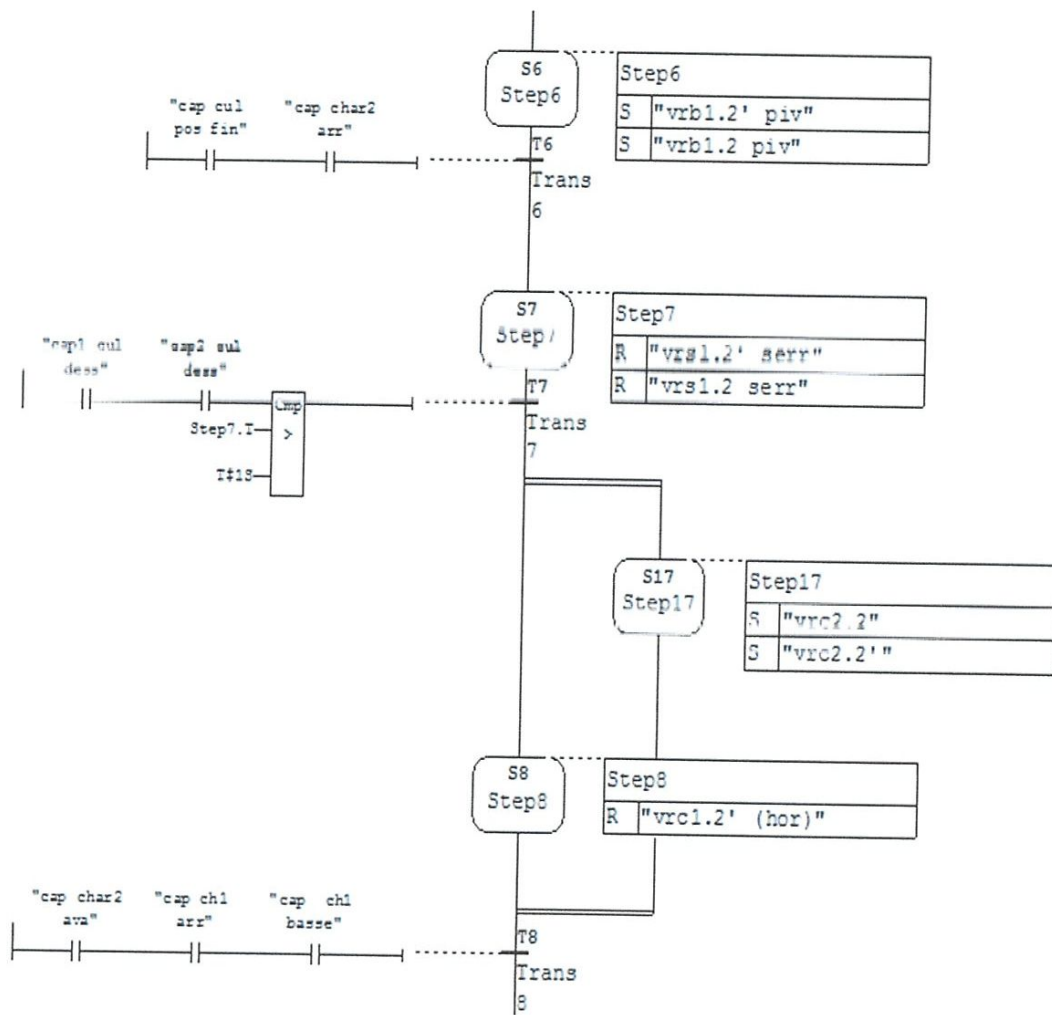
## III.7.1.Tableau mnémorique :

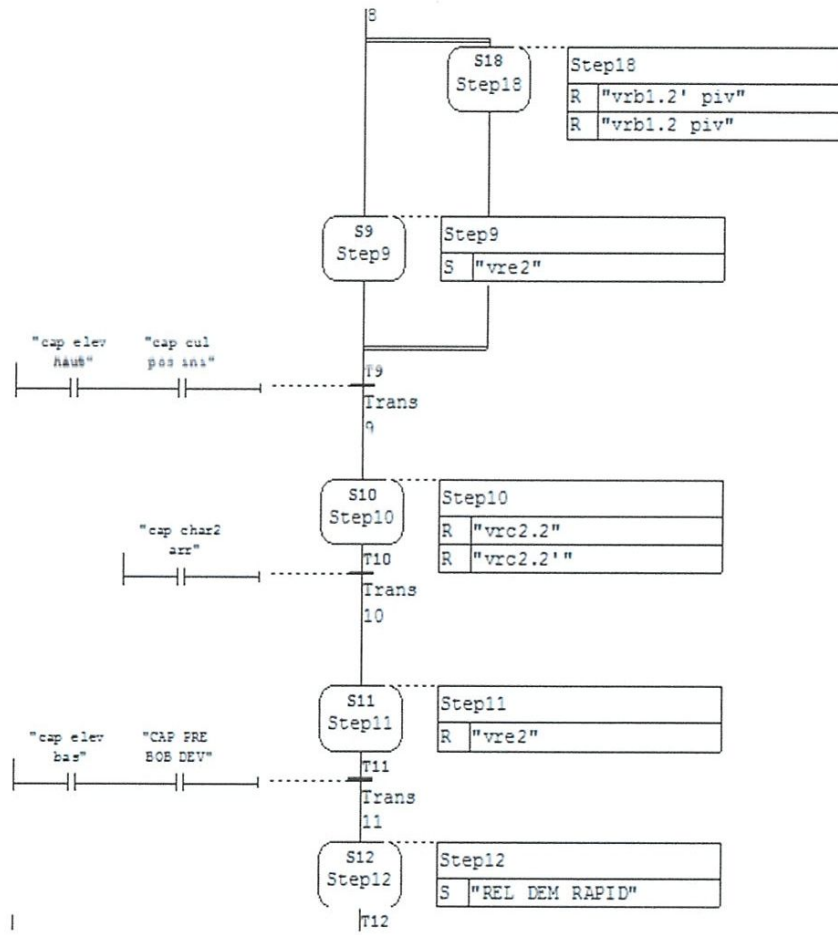
Mnémonique	Opérande	Type de d	Commentaire
bobine pret	E 124.4	BOOL	
cap ch1 basse	E 124.0	BOOL	
cap ch1 arr	E 124.2	BOOL	
cap ch1 av	E 124.3	BOOL	
cap ch1 haut	E 124.1	BOOL	
cap char2 arr	E 125.3	BOOL	
cap char2 ava	E 125.4	BOOL	
cap cul pos fin	E 124.6	BOOL	
cap cul pos ini	E 124.5	BOOL	
cap DT arr	E 126.1	BOOL	
cap DT ava	E 126.2	BOOL	
cap DT bob	E 125.7	BOOL	
cap DT haut	E 126.0	BOOL	
cap elev bas	E 125.5	BOOL	
cap elev haut	E 125.6	BOOL	
cap pre bob DT	E 126.6	BOOL	
CAP PRE BOB CH1	E 126.4	BOOL	
CAP PRE BOB CHAR2	E 126.3	BOOL	
CAP PRF BOB DEV	E 126.5	BOOL	
cap1 cul dess	E 124.7	BOOL	
cap1 cul serr	E 125.1	BOOL	
cap2 cul dess	E 125.0	BOOL	
cap2 cul serr	E 125.2	BOOL	
G7_STD_3	FC 72	FC 72	
KF	A 124.0	BOOL	bobine pour frein
REL DEM LENT	A 125.5	BOOL	
REL DEM RAPID	A 124.1	BOOL	bobine pour contacteur Moteur cc
TIME_TCK	SFC 64	SFC 64	Read the System Time
VDT1	A 124.2	BOOL	vertical DT
VDT1'	A 124.3	BOOL	horizontal DT
vitesse convoyeur	AW 768	WORD	
vrb1.2' piv	A 124.4	BOOL	
vrb1.2 piv	A 124.5	BOOL	
vrc1.2' (hor)	A 124.6	BOOL	horizontal chariot 1
vrc1.2 (verti)	A 124.7	BOOL	vertical chariot 1
vrc2.2	A 125.0	BOOL	horizontal 1 chariot 2
vrc2.2'	A 125.1	BOOL	horizontal 2 chariot 2
vre2	A 125.2	BOOL	deplacement vertical de l'elevateur
vrs1.2' serr	A 125.3	BOOL	
vrs1.2 serr	A 125.4	BOOL	

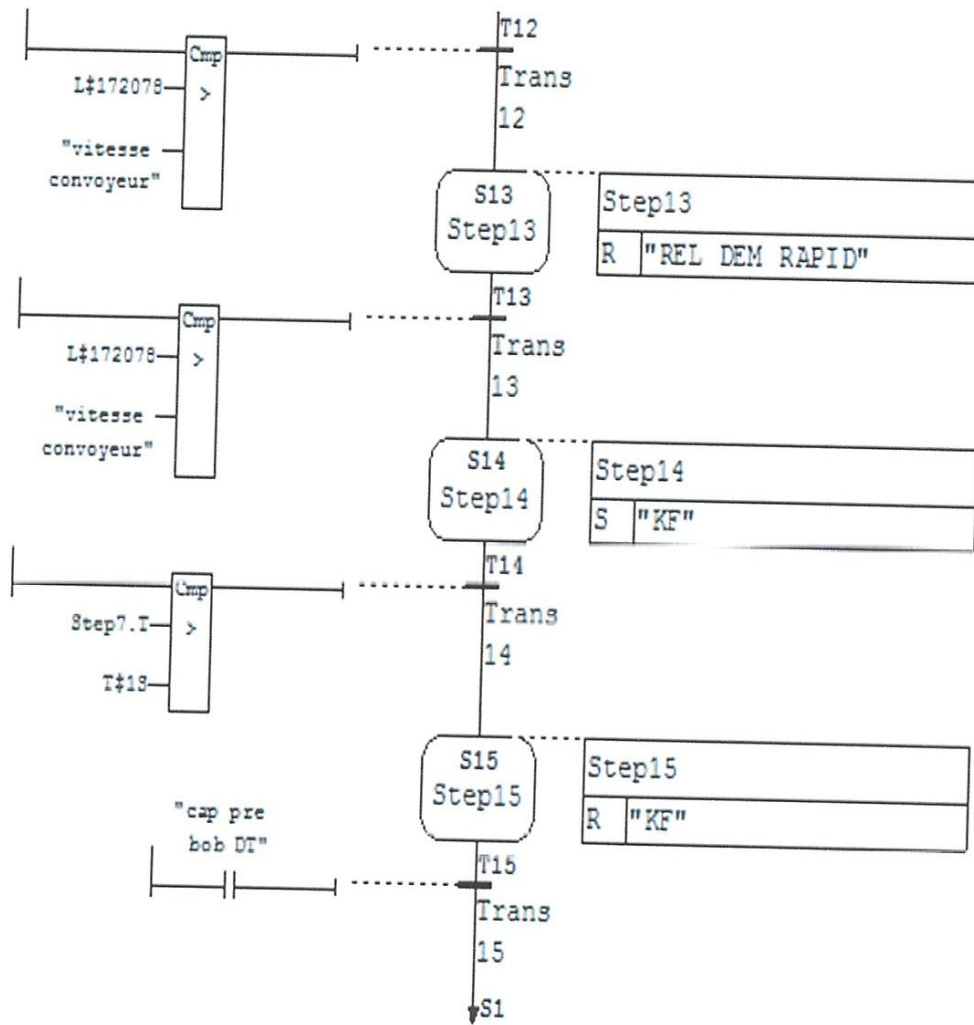
III.7.2.SIMULATION GRAPHIQUE :

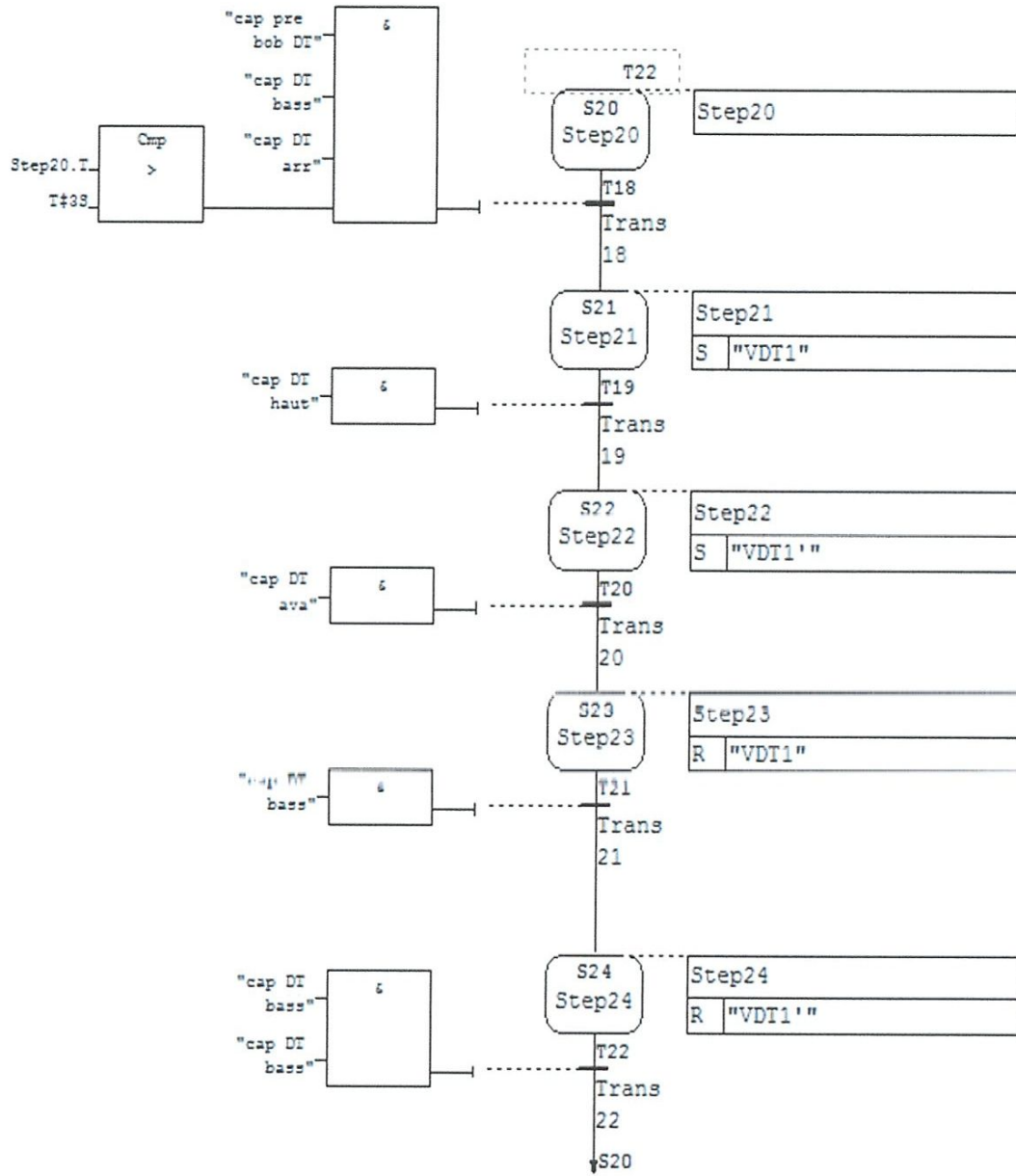






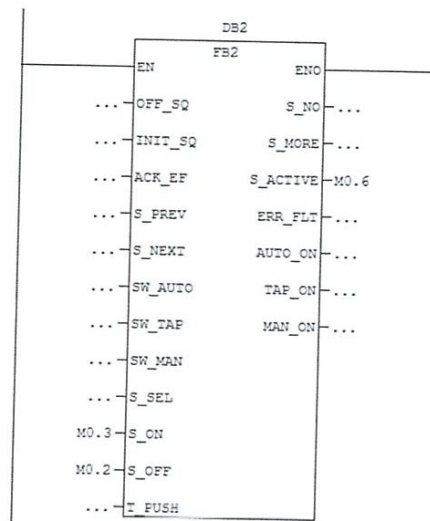




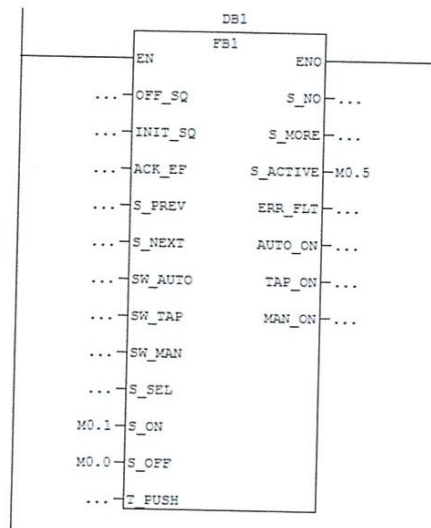


III.7.3.BLOCS FB1 FB2 :

FB1



FB2



## CONCLUSION GENERALE

Cette étude nous a permis de proposer une solution pour une installation au niveau du convoyeur rapide.

Cette solution consiste à la substitution de la logique câblée classique du convoyeur rapide, par une logique programmée en utilisant un API .

On a bénéficié durant cette étude de beaucoup de connaissances de base sur les automates programmables industriels et leurs applications dans les différents domaines de l'industrie. Aussi elle nous a permis de savoir les techniques et les langages de programmations par le logiciel de programmation step7 S 300.

Cette étude ne peut en aucun cas traiter entièrement le problème de l'automate et ses applications , notre attention est seulement de montrer au gens qu'il y a des manières et des techniques de passage d'une commande câblée à une commande programmée et découvrir aussi les possibilités de l'automatisme en utilisant le GRAFCET (une nouvelle méthode de traduction des CACH) .

Souhaitant bien que cette initiative soit un outil commode pour permettre au gens de terminer la 2eme partie qui est l'installation et la mise en marche réelle.

## Bibliographie :

- [1] Automate programmable S7-300 Manuel Système.
- [2] livre : L'informatique en automatisation industrielle, A.Bianciotto, boye.
- [3] L. BERGOUGNOUX, « Automates Programmables Industriels », POLYTECH' Marseille Département de Mécanique Energétique, 2004–2005.
- [4] « L'automate programmable industriel », Collège Montmorency, Département de technologie du génie électrique, 31-08-2006.
- [5] « Structure d'un système automatisé »
- [6] « Grafcet » Livre : Du Grafcet aux réseaux de Pétri "René David et Hassan Alla "
- [7] Manuel SIMATIC (programmer avec Step7).