

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira - Bejaia

Faculté de Technologie

Département de Génie Electrique



PROJET DE FIN D'ETUDES
En vue de l'obtention du diplôme de
Master II en Electrotechnique
Option : *Automatismes Industriels*



Thème

AUTOMATISATION ET
SUPERVISION D'UN SYSTEME
DEVIATEUR DES BOUTEILLES
D'HUILE DE 1.8L

Réalisé par :

Mr : *MEBARKI Amer*

Mr : *ATMANI Adel*

Encadré par :

Mr. : *MELAHI Ahmed*

Mr : *BEN CHAALAL Samir*

Promotion 2017-2018

Remerciements

Nous tenons tout d'abord à remercier Dieu Tout Puissant.

Nous avons l'honneur d'exprimer nos remerciements et notre gratitude à Mr Melahi Ahmed, notre Promoteur pour ses conseils et orientations.

C'est également pour nous une occasion de faire part de toute notre gratitude à tout le personnel de servisse conditionnement d'huile du complexe agroalimentaire CEVITAL en particuliers Mr ben Chaalal Samir et Maouche Yazid pour sa disponibilité et son accueil chaleureux.

Nous adressons nos vifs et sincères remerciements aux membres de jury qui ont accepté de jurer notre travail.

Dédicaces

Que ce travail témoigne de mes respects :

À mes parents :

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont

pu

Créer le climat affectueux et propice à la poursuite de mes études.

Aucune dédicace ne pourrait exprimer mon respect, ma considération

et

Mes profonds sentiments envers eux.

Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils

seront

Toujours fiers de moi.

À mes sœurs et à mes frères.

Ils vont trouver ici l'expression de mes sentiments de respect et de

Reconnaissance pour le soutien qu'ils n'ont cessé de me porter.

A tous mes professeurs :

Leur générosité et leur soutien m'oblige de leurs témoigner mon

profond

respect et ma loyale considération.

A tous mes amis et mes collègues :

Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

AMFR

Dédicace

J'ai le grand plaisir de dédier ce modeste travail :

A ma très chère mère, qui me donne toujours l'espoir de vivre et qui n'a jamais cessé de prier pour moi.

A mon très cher père, pour ses encouragements, son soutien, surtout pour son amour et son sacrifice afin que rien n'entrave le déroulement de mes études.

A toute ma famille sans exception: Mes chères frères et mes sœurs

A tous mes amis

A mon binôme Amer

A Tous ceux que j'aime et je respecte.

ADEL

Sommaire

Remerciements	
Dédicaces	
Sommaire	
Liste des figures	
Introduction général.....	01
Présentation de l'entreprise	
1 Présentation générale de l'entreprise.....	02
2 Historique.....	02
3 Situation géographique	02
4 Activités et unités du complexe CEVITAL.....	03
5. Missions et objectifs	03
6 Différents organes constituant le complexe CEVITAL.....	04
Chapitre I : les systèmes automatisés et GRAFCET	
I.1 Introduction	05
I.2 Les systèmes automatisés.....	05
I.2.1 Définition d'un système automatisé.....	06
I.2.2 Le but de l'automatisme.....	06
I.3 Structure générale d'un système automatisé.....	06
I.3.1 La partie commande.....	06
I.3.1.1 Différent type de commande.....	06
I.3.2 La partie opérative.....	07
I.3.2.1 Les actionneurs	08
I.3.2.2 Les préactionneurs.....	10
I.3.2.3 Capteurs.....	11
I.3.3 La partie relation.....	13
I.4 Architecture des systèmes automatisés	13
I.4.1 Machines automatisés.....	13
I.4.2 Machines associés en ligne.....	14
I.4.3 Cellule de production à commande centralisée.....	14
I.4.4 Cellule à commande décentralisée et coordonnée.....	14
I.4.5 Cellule flexible à commande répartie et hiérarchisé	15
I.5 Graph fonctionnel de commande par étapes et transitions (GRAFCET).....	16
I.5.1 Historique.....	16

I.5.2 Définition du grafcet.....	16
I.5.3 Structure et interprétation du grafcet.....	16
I.5.4 Les éléments graphiques de base du grafcet.....	17
I.6 Conclusion.....	20

Chapitre II : Les automates programmables industriels

II.1 Introduction	21
II.2 Les automates programmables industriels.....	21
II.2.1 Définition générale.....	21
II.2.2 Structure de l'automate programmable.....	21
II.2.2.1 Aspect extérieur	21
II.2.2.2 Structure interne	22
II.2.2.2.1 Première partie de l'API : unité de traitement de processeur.....	23
II.2.2.2.2 Deuxième partie de l'API : mémoires.....	24
II.2.2.2.3 Troisième partie de l'API : modules d'entrée/sortie.....	25
II.2.2.2.4 Quatrième partie de l'API : module d'alimentation.....	26
II.2.3 Les automates programmables industriels siemens.....	26
II.2.3.1 Historique.....	26
II.2.3.2 Présentation de la gamme SIMATIC de SIMENS.....	26
II.2.3.3 Les différentes variantes dans la gamme SIMATIC	27
II.2.3.3 Les différentes variantes dans la gamme SIMATIC	30
II.2.3.4 Les logiciels de programmation et de supervision step7 et wincc flexible.....	30
II.2.3.4.1 Description du step7.....	30
II.2.3.4.2 Gestionnaire de projet SIMATIC	30
II.2.3.4.3 Stratégie pour la conception d'une structure programmable complète et optimisée.....	31
II.2.3.4.4 Surveillance du fonctionnement et diagnostic du matériel	36
II.2.3.5 Progiciel de conception et configuration d'interface wincc flexible	38
II.2.3.5.1 Définition.....	38
II.2.3.5.2 Élément du progiciel wincc flexible	39
II.2.3.5.3 Les différents outils et barres de l'éditeur de vues	40
II.2.3.5.4 Utilisation des variables	41
II.2.3.5.5 Fenêtre des propriétés	42
II.2.3.5.6 Actualisation de valeurs de variables dans les scripts et les fonctions	42
II.2.3.5.7 Compilation et simulation.....	43
II.3 Conclusion	43

Chapitre III : Description de la machine

III. 1 Introduction.....	44
III. 2 Description du déviateur	44
III. 3 Partie mécanique	45
III.3.1 mécanisme de positionnement	45
III.4 Partie capteurs et instrumentation.....	46
III.4.1 Capteur de fin de course	46
III.4.2 Détecteur de proximité.....	47
III. 4.2.1 Définition	47
III.4.2.2 Principe de fonctionnement.....	48
III.4.3 Détecteur photoélectrique	49
III.4.3.1 Définition.....	49
III.4.4 Capteurs de position angulaire (codeurs rotatifs).....	50
III.4.4.1 Codeur optique.....	50
III.4.4.1.1 Définition	50
III.4.4.1.2 Principe de fonctionnement.....	51
III.4.4.1.3 Les types de codeurs optiques	51
III.5 Partie commande	53
III.5.1 Pupitre de commande	53
III.5.2 Contrôleur numérique du moteur de positionnement.....	53
III.5.2.1 Principe fonctions du contrôleur PMC-2.....	54
III.5.3 Variateur de vitesse	55
III.6 Partie pneumatique	57
III.6.1 Distributeurs	58
III.7 Partie actionneurs	59
III.7.1 Moteur asynchrone.....	59
III.7.2 Moteur synchrone	60
III.8 Conclusion.....	61

Chapitre IV : Programmation et supervision du système proposé

IV.1 Introduction	62
IV.2 Problématique.....	62
IV.3 Elaboration des GRAFCETS de déviateur.....	62
IV.3.1 Cahier des charges fonctionnelles.....	62
IV.3.2.1 Table des symboles des grafcet.....	64
IV.3.3 GRAFCET.....	65

IV.3.3.1 Grafctet de défaut.....	65
IV.3.3.2 Grafctet de convoyeur.....	65
IV.3.3.3 Grafctet de marche moteur.....	66
IV.3.3.4 Grafctet des électrovannes.....	66
IV.3.3.5 Grafctet de défaut bouchon.....	67
IV.3.3.6 Grafctet de défaut bouteille.....	67
IV.4 Programmation avec step7.....	68
IV.4.1 Prise en main de l'API.....	68
IV.4.2 Manipulation de l'automate SIMATIC S7-300.....	69
IV.4.2.1 Création d'un projet.....	69
IV.4.2.2 Configuration de l'automate.....	70
IV.4.2.3 Création de la table des mnémoniques.....	74
IV.4.2.3.1 La table mnémonique.....	75
IV.4.2.4 Création d'un bloc de donnée 'DB'.....	76
IV.4.2.5 Création d'un bloc d'organisation 'OB'.....	77
IV.4.2.6 Création d'une fonction 'FC'.....	77
IV.4.2.7 Traitement du programme par la CPU.....	78
IV.4.2.7.1 Programmation linéaire.....	78
IV.4.2.7.2 Programmation structurée (hiérarchisés).....	78
IV.5 Supervision avecWincc.....	79
IV.5.1 Intégration de wincc dans step7.....	79
IV.5.2 Liaison entre station et wincc.....	79
IV.5.3 Table des variables utilisés dans HMI.....	80
IV.5.4 Présentation de l'interface HMI pour le système.....	81
IV.5.4.1 La fenêtre principale.....	81
IV.5.4.2 Vue global.....	82
IV.5.4.3 Vue maintenance.....	82
IV.5.4.4 Vue alarme et les mesures de sécurité.....	83
IV.6 Conclusion.....	84
Conclusion générale.....	85
Bibliographie	
Annexes	

LISTE DES FIGURES

PRESENTATION DE L'ENTREPRISE

Figure 1 : Situation géographique du complexe Cevital.	2
Figure 2 : Organigramme du complexe Cevital.	4

CHAPITRE I

Figure I.1 : Les constituants d'un système automatisé.....	5
Figure I.2 : Eléments de logique câblée	7
Figure I.3 : Description fonctionnelle du pré-actionneur.....	8
Figure I.4 : moteur électrique asynchrone.....	9
Figure I.5 : Le vérin pneumatique.....	9
Figure I.6 : Description fonctionnelle du pré-actionneur.....	10
Figure I.7 : Fonction distributeur	11
Figure I.8 : Schéma d'un capteur (T.O.R)	12
Figure I.9 : Capteurs numériques	12
Figure I.10 : machines autonomes.....	13
Figure I.11 : machines associées en ligne.....	14
Figure I.12 : machines à commande centralisée.....	14
Figure I.13 : machines à commande décentralisé et cordonnée.....	15
Figure I.14 : machines à commande répartie et hiérarchisée.....	15
Figure I.15 : structure et interprétation du grafcet.....	16
Figure I.16 : Les étapes.. ..	17
Figure I.17 : Les actions associées à une étape.....	18
Figure I.18 : les transitions.....	18
Figure I.19 : les liaisons orientées.....	19

CHAPITRE II

Figure II.1 : Les automates de type compact	22
Figure II.2 : Les automates de type modulaire.....	22
Figure II.3 : Structure d'un automate programmable.....	23
Figure II.4 : Liaison entre partie commande et partie opérative.....	25
Figure II.5 : Présentation de la gamme de SIMATIC	27
Figure II.6 : L'API S200.....	27
Figure II.7 : L'API S300.....	28
Figure II.8 : L'API S7 400.....	28
Figure II.9 : La gamme SIMATIC C7.....	29
Figure II.10 : La gamme SIMATIC M7.....	30
Figure II.11 : Organisation structurelle d'un projet.....	31
Figure II.12 : les différentes icônes existant dans STEP 7.....	32
Figure II.13: Configuration matérielle.....	33
Figure II.14: les étapes a' suivre pour crée la table de mnémoniques	33
Figure II.15 : Exemple d'une Table de mnémonique	34
Figure II.16 : mode de représentation des langages basiques de programmation STEP7	35
Figure II.17 : Exemple de programmation avec langage graph	35
Figure II.18: Diagnostic du matériel	37

Figure II.19: logiciel de simulation PLC-SIM.....	37
Figure II.20: L'interface Homme/Machine dans processus automatisé.....	38
Figure II.21 : Etapes pour la conception d'une interface via Win CC flexible	39
Figure II.22 : Vue d'ensemble du logiciel Win CC flexible.....	41
Figure II.23 : éditeur de variable dans le Win CC flexible	42
Figure II.24 : vue d'une fenêtre des propriétés pour les variables.....	42

CHAPITRE III

Figure III.1 : Vue générale de la machine	44
Figure III.2 : Vue de dessus du séparateur de bouteilles	45
Figure III.3 : Détecteur de position et symbole	47
Figure III.4 : Détecteur de proximité inductif et symbole	48
Figure III.5 : Principe de fonctionnement détecteur inductif	48
Figure III.6 : Exemple d'application Détecteur photoélectrique et symbole.....	50
Figure III.7 : Codeur incrémental	51
Figure III.8 : Pupitre de commande	53
Figure III.9 : Principale fonctionnement du PMC-2.....	54
Figure III.10 : Variateur de vitesse	55
Figure III.11 : La partie pneumatique du répartiteur de bouteilles	57
Figure III.12 : Distributeur bistable	58
Figure III.13 : Différentes parties d'un distributeur.....	58
Figure III.14 : Moteur asynchrone triphasé	59
Figure III.15 : Moteur synchrone	60

CHAPITRE IV

Figure IV.1 : Graficets de défaut	65
Figure IV.2 : Graficets de convoyeur	65
Figure IV.3 : Graficets de marche moteur	66
Figure IV.4 : Graficets des électrovannes	66
Figure IV.5 : Graficets de défaut bouchon	67
Figure IV.6 : Graficets de défaut bouteille	67
Figure IV.7 : API SIMATIC S7-300. CPU 314-2DP.....	68
Figure IV.8 : Création d'un nouveau projet.....	69
Figure IV.9 : Insérer de station SIMATIC 300.....	69
Figure IV.10 : Fenêtre générée	70
Figure IV.11 : Panneau de configuration du matériel	70
Figure IV.12 : Affichage de profilé support	71
Figure IV.13 : Insertion des modules dans le tableau de configuration	72
Figure IV.14 : Choix de l'alimentation	72
Figure IV.15 : Choix du CPU.....	73
Figure IV.16 : Configuration finale	73
Figure IV.17 : Fenêtre de création des mnémoniques	74
Figure IV.18 : Table des mnémoniques	75
Figure IV.19 : Fenêtre Création d'un bloc de donnée 'DB'	76
Figure IV.20 : Fenêtre de nom de bloc de donnée.....	76
Figure IV.21 : Fenêtre de Création d'un bloc d'organisation 'OB'.....	77
Figure IV.22 : Fenêtre de Création d'une fonction ' FC '	77

Figure IV.23 : Fenêtre des blocs	78
Figure IV.24 : Intégration de WINCC dans step7.....	79
Figure IV.25: Intégration de WINCC dans step7.....	79
Figure IV.26 : Tableau des variables	80
Figure IV.27 : La fenêtre principale de l'application	81
Figure IV.28 : Vue Global	82
Figure IV.29 : Vue maintenance.....	83
Figure IV.30 : Vue alarmes	84

Liste des abréviations utilisées

PO : Partie Opérative

PC : Partie Commande

PR : Partie Relation

TOR : Tout Ou Rien

GRAF CET : Graphe Fonctionnel de Commande par Etapes et Transitions

API : Automate Programmable Industriel

CPU : Central Processing Unit

ROM : Read Only Memory

RAM : Random Access Memory

EPROM : PROM à Effacement électrique

WINCC : Windows Control Center

MPI : Multi Point Interface

CONT : Contacte

LOG : Logigramme

SAP : Systèmes Automatisés de Production

VAT : Table des Variables

HMI : Interface Homme/Machine

Introduction

générale

Introduction générale

Actuellement, dans le contexte économique mondial, la production automatisée est une nécessité absolue, la cadence de la production doit être très élevée pour rester compétitif sur le marché international. Pour répondre au besoin de la demande du marché et assurer la pérennité de l'entreprise, les industriels sont obligés de faire appel à l'automatisation des machines et des lignes de production qui doivent être flexibles.

La réalisation d'un système automatisé nécessite un cahier des charges donné par le client qui contient les besoins de l'industrie, et le constructeur qui a pour mission de donner le bon choix des éléments à utiliser.

La problématique qui nous a été posée au sein de l'unité de conditionnement d'huile de CEVITAL est de faire l'automatisation d'un système déviateur des bouteilles d'huile de 1.8L, ce dernier est actuellement commandé par une carte électronique et on veut remplacer cette carte par un API S7 300, ce que cause:

- La facilité d'utilisation et la souplesse qu'offre l'A.P.I pour sa programmation, sa connexion et son adaptation aux conditions industrielles
- La carte électronique est un ensemble de circuits intégrés formant un programme prédéterminé par le programmeur. Donc on ne peut pas la modifier et elle est très coûteuse.

Notre travail est organisé en quatre chapitres qui se terminent par une conclusion générale.

Dans le premier chapitre nous présentons les systèmes automatisés en générale et le logiciel

Le deuxième chapitre est consacré à la présentation des généralités sur les automates programmables

Le troisième chapitre est consacré à la présentation de la machine et le système actuel.

Le quatrième chapitre est consacré à la programmation et l'élaboration de la supervision du système proposé et nous terminons notre travail par une conclusion générale.

Préambule

Présentation de l'entreprise

1 Présentation générale de l'entreprise

CEVITAL est un ensemble industriel intégré, concentré en première partie dans le secteur de l'agroalimentaire : raffinage d'huile et de sucre, produits dérivés, négoce de céréales, distribution de produits destinés à l'alimentation humaine et animale. Elle conçoit des produits de qualité supérieure à des prix compétitifs, grâce à ses installations performantes, son savoir-faire, son contrôle strict de qualité et son réseau de distribution. Elle couvre les besoins nationaux et a permis de faire passer l'Algérie du stade d'importateur à celui d'exportateur pour les huiles, les margarines et le sucre. Ses produits se vendent dans plusieurs pays, notamment en Europe, au Maghreb, au Moyen Orient et en Afrique de l'Ouest [1].

2 Historique

Fondé par Mr. Isaad Rebrab, CEVITAL est un groupe familial de plusieurs sociétés bâti sur une histoire, créé par des fonds privés en 1998 à Bejaïa, à l'entrée du pays dans l'économie de marché. Première entreprise privée algérienne à avoir investi dans des secteurs d'activités diversifiés, elle a traversé d'importantes étapes historiques pour atteindre sa taille et sa notoriété actuelle.

3 Situation géographique

CEVITAL Agro-industrie est implanté au niveau du nouveau quai port de BEJAIA à 3 Km du sud-ouest de cette ville, à proximité de la RN 09. Cette situation géographique de l'entreprise lui a beaucoup profité étant donné qu'elle lui donne l'avantage de proximité économique. En effet, elle se trouve proche du port et de l'aéroport [1].

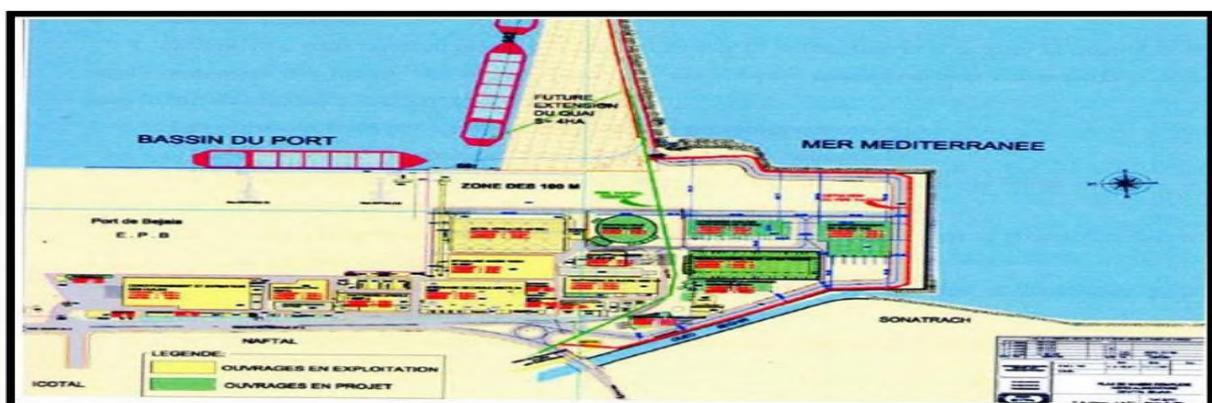


Figure 1 Situation géographique du complexe

Présentation de l'entreprise

4 Activités et unités du complexe CEVITAL

Créée en 1998 et implantée au sein du port de Bejaïa, Cevital Agro-industrie dispose de plusieurs unités de production ultramodernes :

- 2 raffineries de sucre.
- 1 unité de sucre liquide.
- 1 raffinerie d'huile. 1 margarinerie.
- 1 unité de conditionnement d'eau minérale.
- 1 unité de fabrication et de conditionnement de boissons rafraîchissantes.
- 1 conserverie.
- 1 unité de fabrication de chaux calcinée.

Elle possède également des silos portuaires ainsi qu'un terminal de déchargement portuaire d'une capacité de 2000 tonnes/heure ce qui en fait le premier terminal de déchargement portuaire en Méditerranée [1].

5. Missions et objectifs

L'entreprise a pour mission principale de développer la production et d'assurer la qualité et le conditionnement des huiles, des margarines et du sucre à des prix nettement plus compétitifs et cela dans le but de satisfaire le client et le fidéliser.

Les objectifs visés par CEVITAL peuvent se présenter comme suit :

- L'extension de ses produits sur le territoire national.
- L'importation de graines oléagineuse pour l'extraction directe des huiles brutes.
- L'optimisation de ses offres d'emploi sur le marché du travail.
- L'encouragement des agriculteurs par des aides financières pour la production locale des graines oléagineuses.
- La modernisation de ses installations en termes de machine et technique pour augmenter le volume de sa production.
- Le positionnement de ses produits sur le marché étranger par leurs exportations [1].

Présentation de l'entreprise

6 Différents organes constituant le complexe CEVITAL :

L'organigramme suivant donne une vue générale sur les différents organes constituant le complexe CEVITAL [1]:

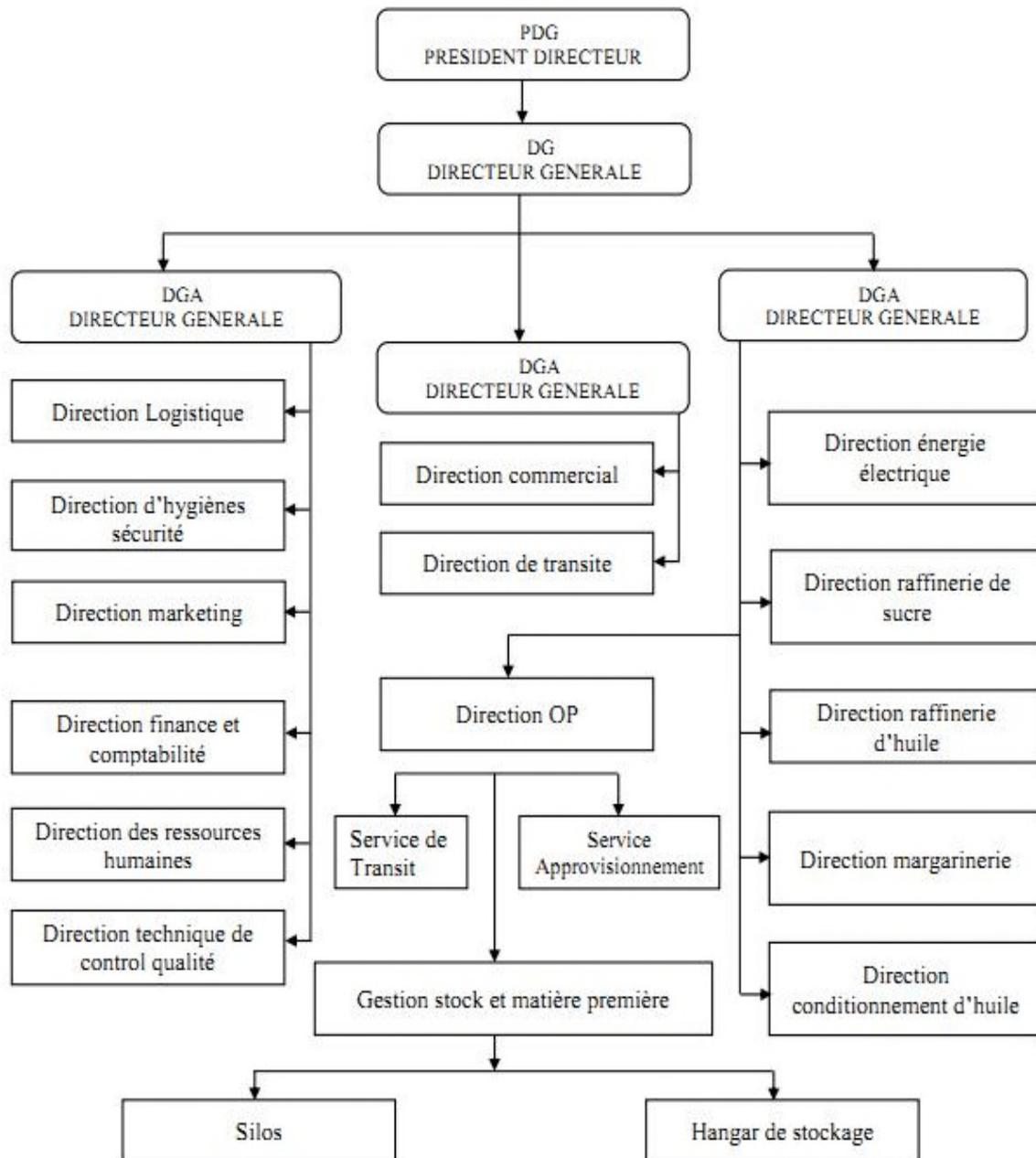


Figure 2 Organigramme du complexe Cevital

Chapitre I :
Les
systemes
automatisés
et GRAFCET

I.1 Introduction

Ce chapitre traite la structure d'un Système Automatisé et définit les différentes parties de ce système et GRAFCET.

L'automatisation consiste à rendre automatique les opérations qui exigeaient auparavant l'intervention humaine. Ce système est un sous-ensemble d'une machine, destinée à remplacer l'action de l'être humain dans des tâches, en générale simples et répétitives, réclamant précision et rigueur. On est passé d'un système dit manuel, à un système mécanisé, puis un système automatisé.

L'automatisation d'une machine nécessite un dialogue entre le client qui définit le cahier des charges et le constructeur qui propose des solutions c'est pour ça l'ADEPA (agence pour le développement de la production appliquée à l'industrie) a créé le GRAFCET.

I.2 Les systèmes automatisés

Simple ou complexes, les systèmes automatisés sont partout dans notre environnement quotidien et connaître leur fonctionnement nous permet aussi de mieux nous adapter avec le secteur de travail.

Les systèmes automatisés, utilisés dans le secteur industriel, possèdent une structure de base identique. Ils sont constitués de plusieurs parties plus ou moins complexes reliées entre elles :

- la partie opérative (PO).
- la partie commande (PC) ou système de contrôle/commande.
- la partie relation (PR) de plus en plus intégrée dans la partie commande.

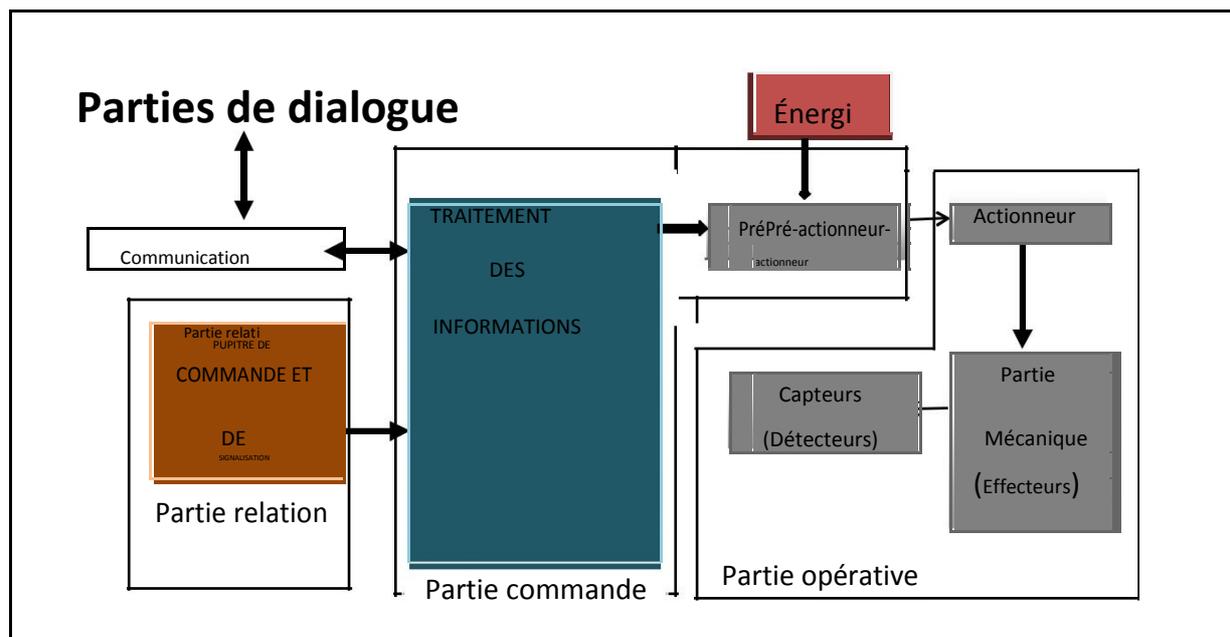


Figure I.1 Les constituants d'un système automatisé [2]

Chapitre I : systèmes automatisés et GRAFCET

I.2.1 Définition d'un Système automatisé

Un système est dit " automatisé " s'il exécute toujours le même cycle de travail après avoir reçu les consignes d'un opérateur. Celui-ci se contente de donner des ordres de départ et si besoin d'arrêt [2].

I.2.2 Le but de l'automatisme

- ✓ effectuer une production qualitative. (Pas d'erreur humaine : Zéro défaut.) ;
- ✓ effectuer une production quantitative. (rapidité) ;
- ✓ suppression des tâches ou actions physiques peu ou pas gratifiantes pour l'homme ;

- ✓ pouvoir accéder à des milieux de travail hostiles. (Chimique, nucléaires ...) ou des sites inaccessibles à l'homme (mer, espace) ;
- ✓ augmenter la sécurité ;
- ✓ superviser les installations et les machines et les processus de production [2].

I.3 Structure générale d'un système automatisé

I.3.1 La partie commande

Elle donne les ordres de fonctionnement à la partie opérative. Les pré-actionneurs permettent de commander les actionneurs ; ils assurent le transfert d'énergie entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs.

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

En fonction de ces consignes et de son programme de gestion des tâches (implanté dans un automate programmable ou réalisé par des relais (on parle de logique câblée)), elle va commander les pré-actionneurs et renvoyer des informations au pupitre de signalisation ou à d'autres systèmes de commande et/ou de supervision en utilisant un réseau et un protocole de communication [3].

I.3.1.1 Différents types de commande

> La logique programmée

L'élément principal s'appelle l'Automate Programmable Industriel ou l'API. La détection est électrique. Le pilotage des actionneurs se fait par l'intermédiaire de relais ou de distributeurs. Il existe sur le marché de nombreuses marques d'automates: Télémécanique, Siemens, Omron, Allen Bradley, Cegetel, etc. [3].

➤ La logique câblée

L'élément principal s'appelle module séquenceur (Figure I.2) et l'association de modules constitue un ensemble appelé séquenceur. La détection est pneumatique, le pilotage des distributeurs se fait par une action de l'air comprimé sur un piston qui fait déplacer le tiroir du distributeur à droite ou à gauche.

L'ensemble, appelé tout pneumatique, est homogène et fiable [3].

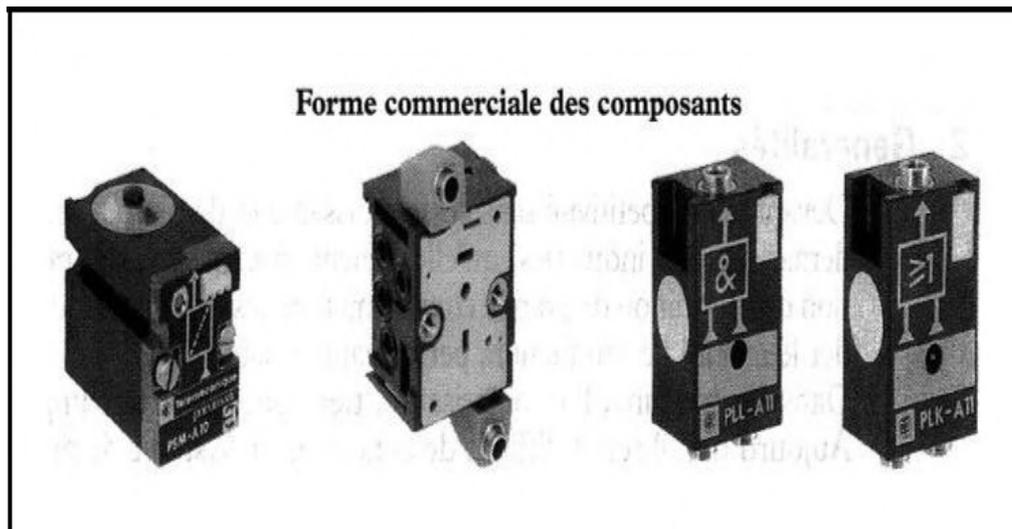


Figure I.2 Eléments de logique câblée

I.3.2 partie opérative

C'est la partie visible du système. Elle comporte les éléments du procédé, c'est à dire :

- des actionneurs (vérins, moteurs, vannes...) qui ont pour rôle d'exécuter les ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie conforme à l'action souhaitée (mécanique, thermique...).
- des pré-actionneurs (distributeurs, contacteurs...) qui reçoivent des ordres de la partie de commande.
- des capteurs qui informent la partie de commande de l'exécution du travail. Par exemple, on va trouver des capteurs mécaniques, pneumatiques et électriques ou magnétiques montés sur les vérins. Le rôle des capteurs est donc de mesurer, surveiller et informer la PC sur l'évolution du système.

I.3.2.1 actionneurs

Un actionneur est un organe de puissance qui convertit une énergie d'entrée (électrique, pneumatique, hydraulique) en une énergie de sortie utilisables pour obtenir une action définie (mouvement de rotation, mouvement de translation, abaissement de température, production d'un son, production d'un rayonnement lumineuse, ...etc.). Il est en générale associé à un pré-actionneur (voir figure I.3) [3].

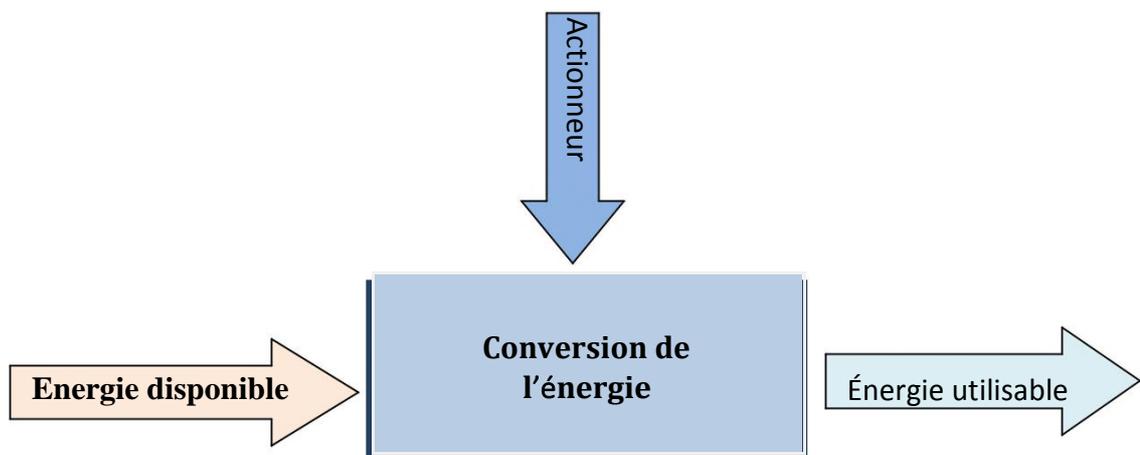


Figure I.3 Description fonctionnelle actionneur

Il existe trois types d'actionneurs [4] :

a. Les actionneurs électriques

Ils utilisent directement l'énergie électrique distribuée sur les machines ils prennent des formes variées :

- Moteur à vitesse constante ou variable.
- Electrovanne de débit.
- Résistance de chauffage.

Les pré-actionneurs associés à ces actionneurs sont principalement les contacteurs ou variateurs de vitesse.

Chapitre I : systèmes automatisés et GRAFCET

Exemple d'un moteur électrique asynchrone :

Le moteur électrique asynchrone transforme une énergie de puissance électrique triphasée en énergie mécanique de rotation (voir figure. I.4).



Figure I.4 Moteur électrique asynchrone

b. Les actionneurs pneumatiques

Ils utilisent directement l'air comprimé distribué sur les machines, Ils sont simples d'emploi et se présentent sous des forme très variées. Les plus utilisés sont les vérins pneumatiques. Les distributeurs sont les pré-actionneurs qui leur sont associés.

Exemple de vérin pneumatique :

Un vérin pneumatique est un actionneur linéaire dans lequel l'énergie de l'air comprimé est transformée en travail mécanique (voir figure. I.5).



Figure I.5 Le vérin pneumatique

c. Les actionneurs hydrauliques

Ils ne sont utilisés que lorsque les actionneurs électriques et pneumatiques ne peuvent pas donner satisfaction, car ils exigent l'installation sur la machine d'un groupe générateur de pression. Ils sont utiles lorsque des efforts à développer sont très importants (pression) ou lorsque des vitesses lentes doivent être contrôlés avec précision [3].

I.3.2.2 Les Pré-actionneurs

Le Pré-actionneur est un constituant de gestion de l'énergie fournie à l'actionneur (voir figure. I.6).

On peut distinguer deux catégories de pré-actionneur [4] :

- Pré-actionneur de distribution ou de commande tout ou rien(TOR). Ce type de pré-actionneur permet d'établir ou de couper un circuit (alimentation des actionneurs).
- Pré-actionneur de conversion/amplification. Ce type de pré-actionneur permet d'adapter les caractéristiques (puissance, pression,...) de son énergie, son signal de commande à celle de l'énergie de puissance nécessaire à l'alimentation des actionneurs.

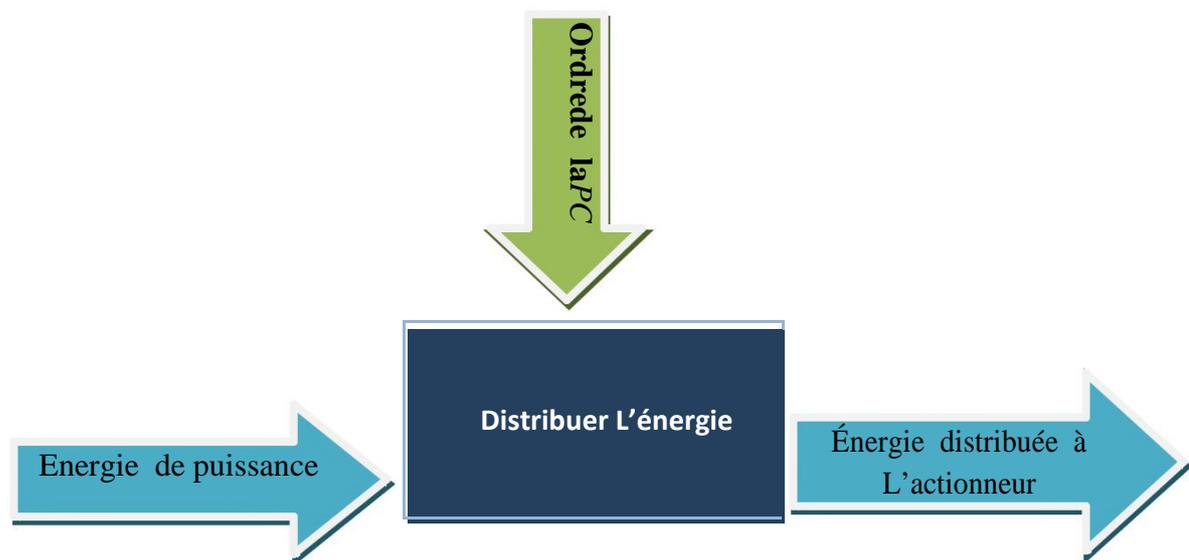


Figure I.6 Description fonctionnelle du pré-actionneur

a. Variateur de fréquence

C'est un pré-actionneur analogique, qui est utilisé pour varier l'énergie distribuée à l'actionneur. Si on a comme actionneur un moteur asynchrone, la fréquence de rotation du moteur est fonction de la fréquence de courant provenant du pré-actionneur. En utilisant un variateur de fréquence, la fréquence alimentant le moteur varie en fonction de la consigne provenant de l'automate programmable [5].

b. Relais contacteur

Ils sont des pré-actionneurs Tout Ou Rien (TOR). Les contacteurs sont des relais conçus pour commuter des courants électriques forts [3].

c. distributeurs (pré-actionneurs pneumatique)

Ils sont utilisés pour distribuer l'air sous pression aux différents orifices des actionneurs pneumatiques (voir Figure I.7)

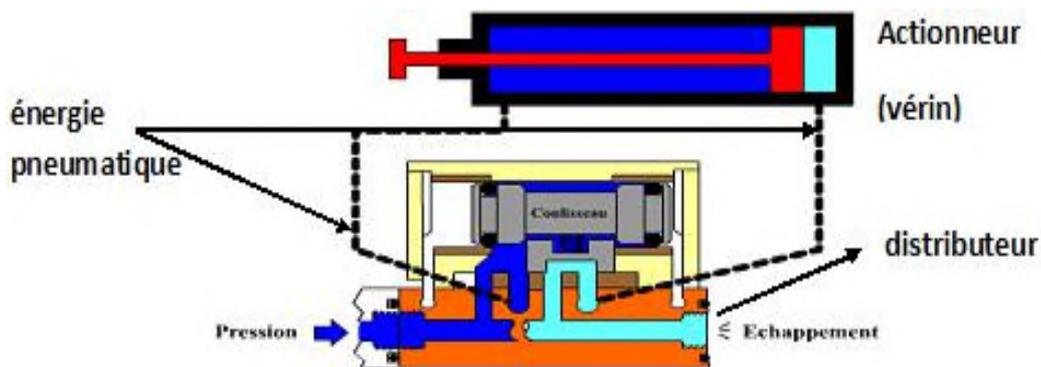


Figure I.7 Fonction distributeur

I.3.2.3 Capteurs

Les capteurs sont des éléments qui transforment une grandeur physique (position, distance, vitesse, température, pression...etc.) D'une machine ou d'un processus en une grandeur normée, généralement électrique, qui peut être interprétée par un dispositif de contrôle commande [4].

➤ capteurs Tous Ou Rien (T.O.R)

Sont des capteurs qui délivrent un signal de sortie logique c'est à dire 0 ou 1, et Ce sont des capteurs les plus répandus en automatisation courante : capteurs à contact, capteur sans contact.

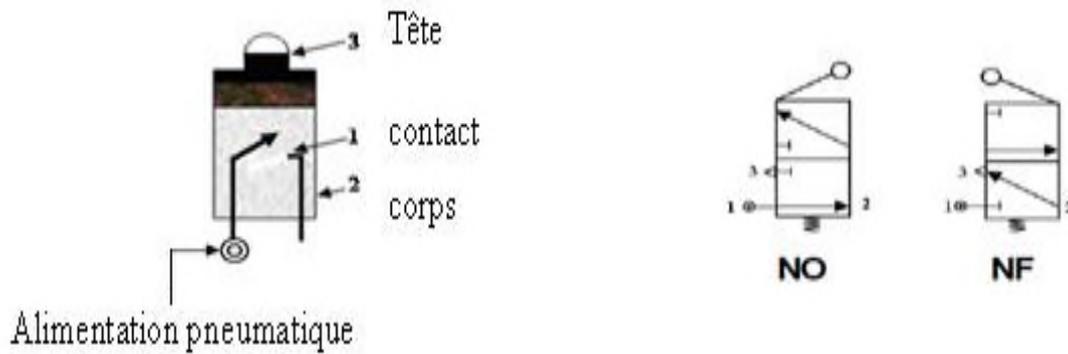


Figure I.8 Schéma d'un capteur (T.O.R)

➤ Les capteurs analogiques

Les capteurs analogiques sont des capteurs qui permettent de délivrer une image d'un phénomène physique sous la forme d'une tension ou d'un courant proportionnels à la grandeur mesurée [3].

Le plus souvent, cette image analogique sera transformée en image numérique par un coupleur pour pouvoir être traitée par un automate

Parmi ces capteurs, on distingue

Les détecteur photoélectrique a' sortie analogique

Les capteurs de vitesse

Les sondes de température

➤ Les capteurs numériques

Ils transmettent des valeurs numériques précisant des positions, des pressions, des températures,... sous forme de combinaison de signaux (0-1). Le signal délivré par ce capteur est une valeur numérique [5].



Figure I.9 Capteurs numériques

I.3.3 La partie relation

Elle représente le pupitre de dialogue homme-machine équipé des organes de commande permettant la mise en/hors énergie de l'installation, la sélection des modes de marche, la commande manuelle des actionneurs, la mise en référence, le départ des cycles ,l'arrêt d'urgence, etc.... et nous permet la visualisation de l'état du processus à tout instant.

Elle doit être sous tension dans le cas d'une technologie câblée et en mode « RUN » du programme en cas d'une technologie programmée [6].

I.4 les différentes structures d'un système automatisé :

I.4.1 Machines autonomes

Chaque machine réalise une étape dans l'élaboration du produit. La manutention entre machine, les chargements et les déchargements sont nombreux, coûteux et longs. Ils sont le plus souvent manuels [5].

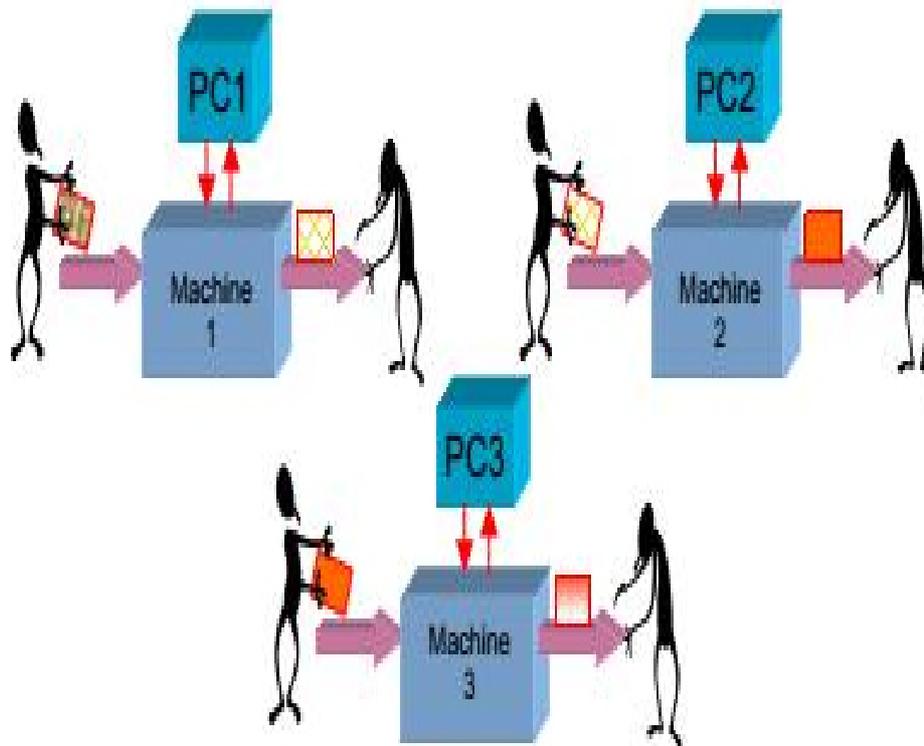


Figure I.10 Machines autonomes

I.4.2 Machines associées en ligne

Le produit passe automatiquement d'une machine à la suivante. Dans ce cas simple, c'est le transfert du produit lui-même qui assure la liaison entre les machines.

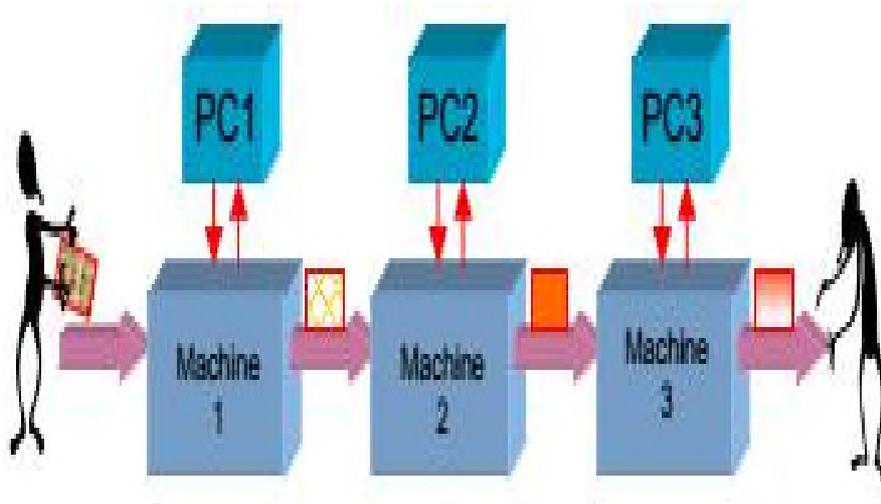


Figure I.11 Machines associées en ligne

I.4.3 Cellule de production à commande centralisée

La nécessité de coordonner l'action des machines a d'abord conduit à centraliser leurs commandes, ce qui par ailleurs a compliqué les interventions locales de réglage et de dépannage [5].

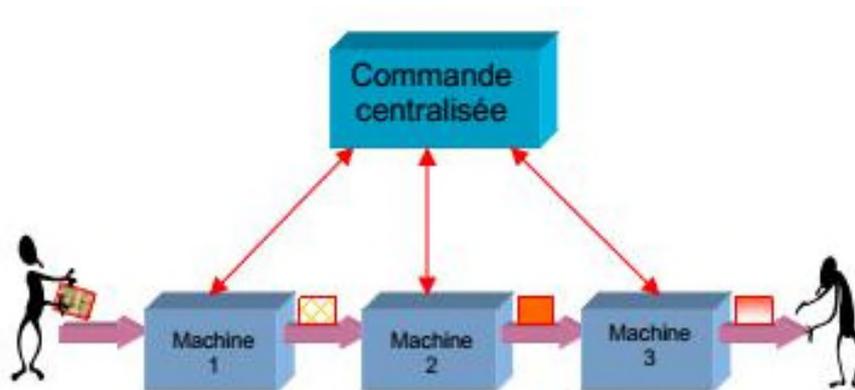


Figure I.12 Machines à commande centralisée

I.4.4 Cellule à commande décentralisée et coordonnée

Un retour aux commandes décentralisées s'est imposé, mais avec une coordination entre machines ici assurée par des liaisons inter niveaux [5].

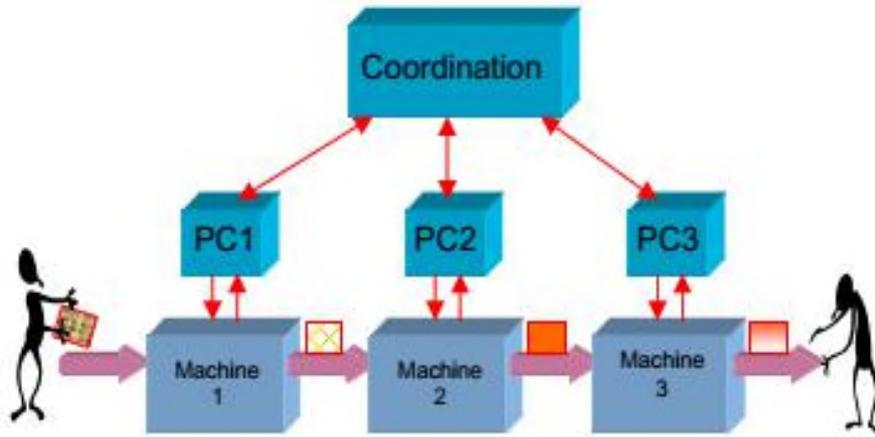


Figure I.13 Machines à commande décentralisée et coordonnée

I.4.5 Cellule flexible à commande répartie et hiérarchisée

Le besoin de flexibilité conduit à prévoir des transferts libres de produit de machine à machine : une machine donnée peut traiter ou non le produit présenté. Les liaisons iso-niveaux assurent la communication avec la supervision [5].

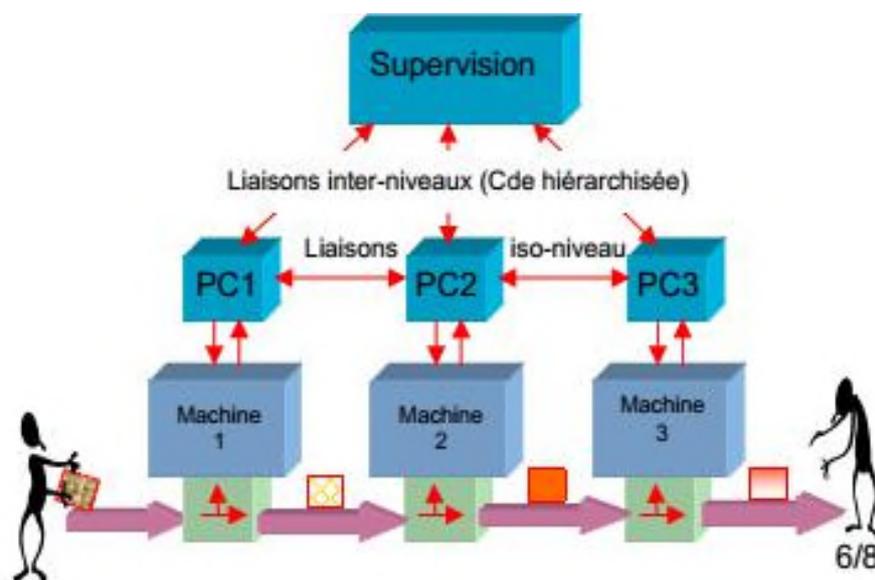


Figure I.14 Machines à commande répartie et hiérarchisée

I.5 GRAPhe Fonctionnel de Commande par Étapes et Transition (GRAFCET)

I.5.1 Historique

Le GRAFCET (Graphe Fonctionnel de Commande par Etapes et Transitions) est né en 1977 des travaux de l'AFCEC (Association Française pour la Cybernétique Economique et Technique), en tant que synthèse théorique des différents outils existant à cette époque (organigramme, organiphase, diagramme de Girard, réseaux de Pétri, etc...). Mis sous forme actuelle par l'ADEPA (Agence nationale pour le Développement de la Production Automatisées) en 1979, normalisé sur le plan français en 1990 (norme NF C03-190), sur le plan européen (EN61131) et sur le plan international en 1992 (norme CEI 1131) [7].

I.5.2 Définition du grafcet

Le GRAFCET est un modèle de représentation graphique des comportements successifs d'un système logique, préalablement défini par ses entrées et ses sorties [8].

I.5.3 Structure et interprétation du grafcet :

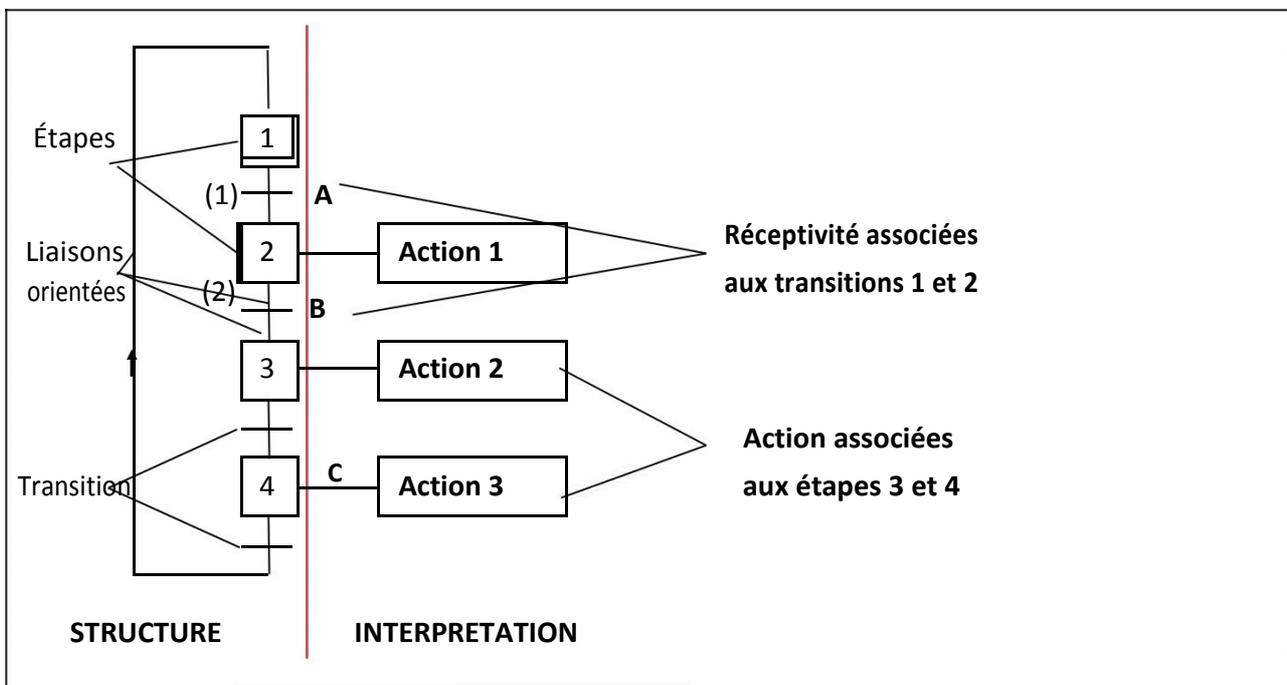


Figure I.15 Structure et interprétation du grafcet

Chapitre I : systèmes automatisés et GRAFCET

Le **GRAFCET** exprime les comportements attendus de systèmes logiques.

- Sa représentation, faite à partir d'éléments graphiques de base, comprend :
 - Des étapes
 - Des transitions
 - Des liaisons orientées.
- Ses évolutions sont définies par :
 - Cinq règles d'évolution.
- Son interprétation se traduit par :
 - Des actions associées aux étapes.
 - Des réceptivités associées aux transitions.

I.5.4 Les éléments graphiques de base du GRAFCET :

a) Les étapes

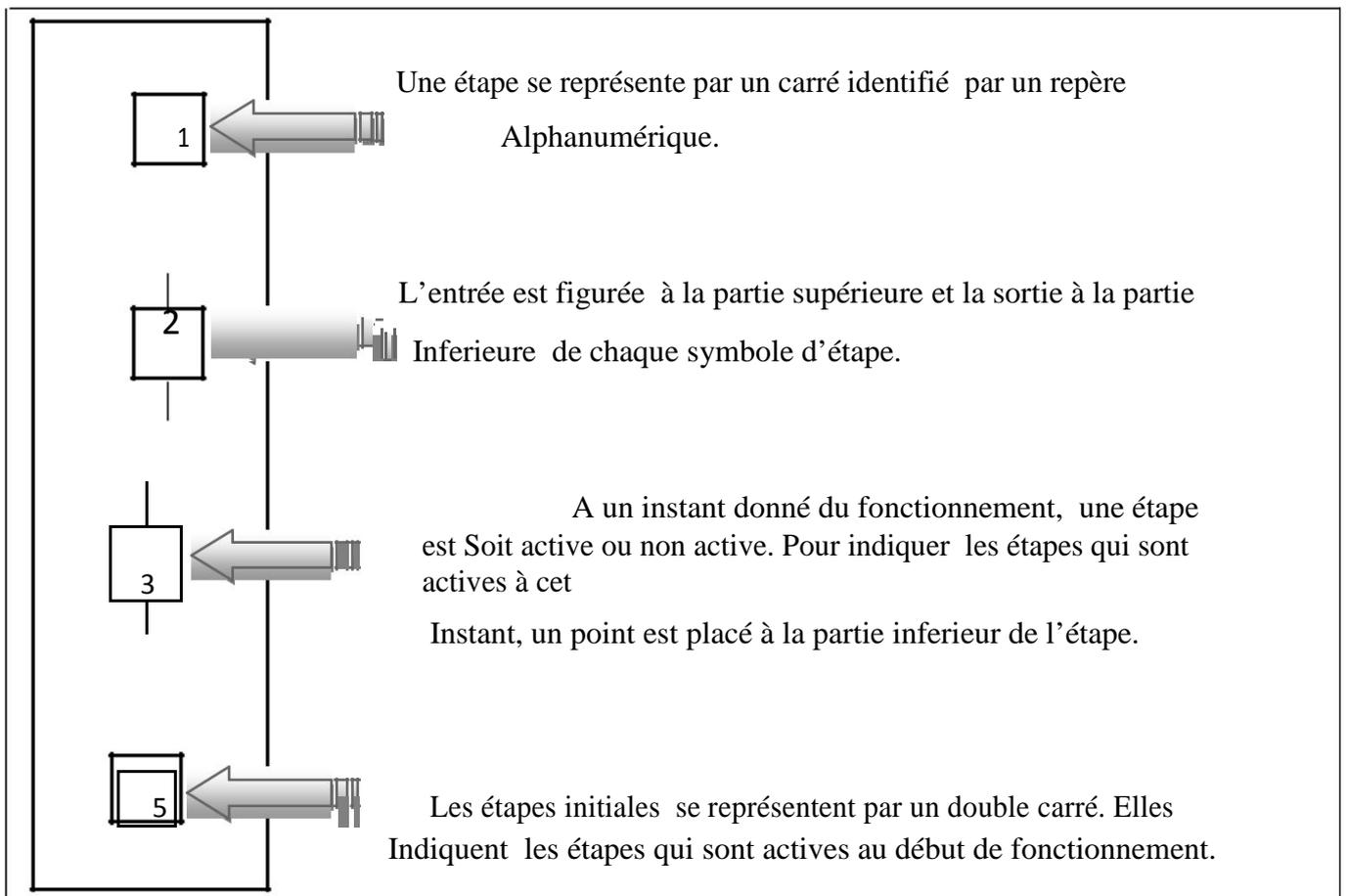


Figure I.16 Les étapes

Chapitre I : systèmes automatisés et GRAFCET

Une ou plusieurs ACTIONS peuvent être associées à une étape. Elles traduisent « ce qui doit être fait » chaque fois que l'étape à laquelle elles sont associées est active.

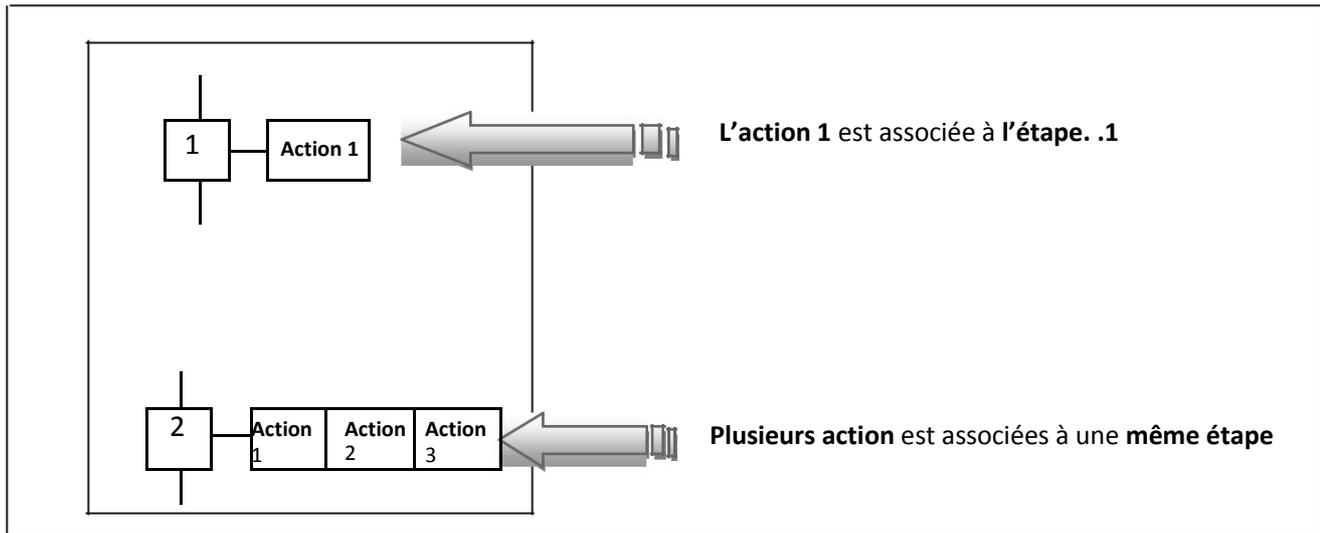


Figure I.17 Les actions associées à une étape

b) Les transitions

Une **transition** indique la possibilité d'évolution entre plusieurs étapes.

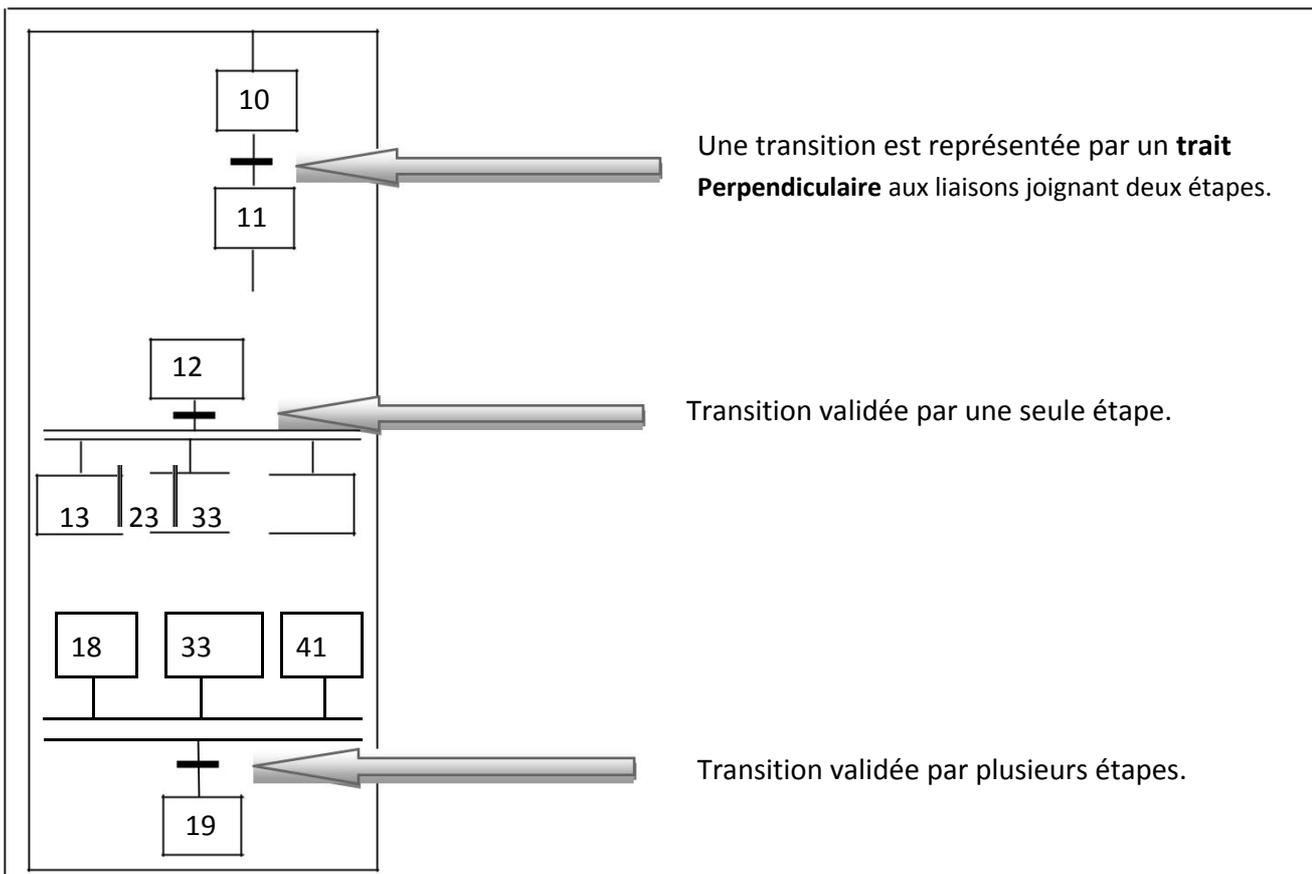


Figure I.18 Les transitions

c) Les liaisons orientées

Les liaisons orientées relient les étapes aux transitions et les transitions aux étapes. Les liaisons orientées se représentent par des lignes Horizontales ou verticales. Par convention, le sens de flèches est de haut vers le bas. Des flèches doivent être utilisées lorsque cette convention n'est pas respectée.

(a) : sens descend , (b) : sens montant

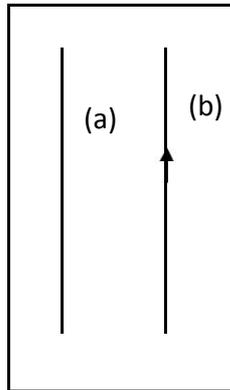


Figure I.19 Les liaisons orientées

d) Les règles du grafcet



Règle d'alternance

L'alternance étape/transition doit être respectée. C'est à dire deux étapes ne peuvent

Pas de suivre ni deux transitions.



Règles d'évolution



Situation initiale (règle 1)

La situation initiale d'un GRAFCET est caractérisée par une étape initiale seule et unique. Cette situation correspond généralement à une position de repos. S'il existe plusieurs GRAFCET, il y aura plusieurs étapes initiales.

Certains automatismes nécessitent le redémarrage à l'étape active au moment, par exemple, d'une coupure secteur. Cette situation doit être clairement définie et explicitée.



Franchissement (règle 2)

L'évolution du GRAFCET d'une étape vers une autre ne peut se faire : que si l'étape est valide, c'est à dire active et que la réceptivité de la transition est vraie.

Lorsque ces deux conditions sont réunies, la transition est obligatoirement franchie



Evolution (règle 3)

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

I.5 Conclusion

Ce chapitre nous a permis de présenter la structure et l'architecture générale d'un système Automatisé. Nous avons décrit la fonction de chacun des constituants de système.

En effet, la conception d'un système automatisé repose essentiellement sur le choix judicieux des capteurs et actionneur interagissant avec la partie commande.

Chapitre II :
Les automates
programmables
industriels

II.1 Introduction

L'automate programmable industriel API est aujourd'hui un élément incontournable pour la commande des systèmes automatisés dans tous les domaines. Il répond aux besoins d'adaptation et de flexibilité dans nombre d'activités économiques actuelles ; les API ont une place majeure due à leurs caractéristiques propres, matérielles et logicielles, leur capacité à s'intégrer dans un ensemble plus large, et donc à répondre aux besoins d'un système automatisé de production SAP.

II.2 automates programmables industriels

II.2.1 Définition générale

Un automate programmable industriel (API) est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires. Il exécute une suite d'instructions introduites dans ses mémoires sous forme de programmes, et s'apparente par conséquent aux machines de traitement de l'information [15].

II.2.2 Structure de l'automate programmable

II.2.2.1 Aspect extérieur

L'aspect des automates change d'un modèle à un autre, mais ils sont placés selon le type compact ou modulaire.

a) Automate de type compact

Les automates de type compact, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes. Ce sont des micros automates ; ils intègrent le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, ils pourront réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogique...) et recevoir des extensions en nombre limité. De type compact (Figure II.1), on distinguera les modules de programmation (LOGO de siemens, ZELIO de Schneider et MILLENIUM de Crouzet) [5].

Chapitre II : Les automates programmables industriels



Figure II.1 Les automates de type compact[5].

b) Automate de type modulaire

Les automates de type modulaire (Figure II.2) sont intégrés dans les automatismes complexes ou de puissance où capacité de traitement et flexibilité sont nécessaires. Dans ces automates le processeur, l'alimentation et les interfaces d'entrée/sortie résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le fond de panier (bus plus connecteur)[5].

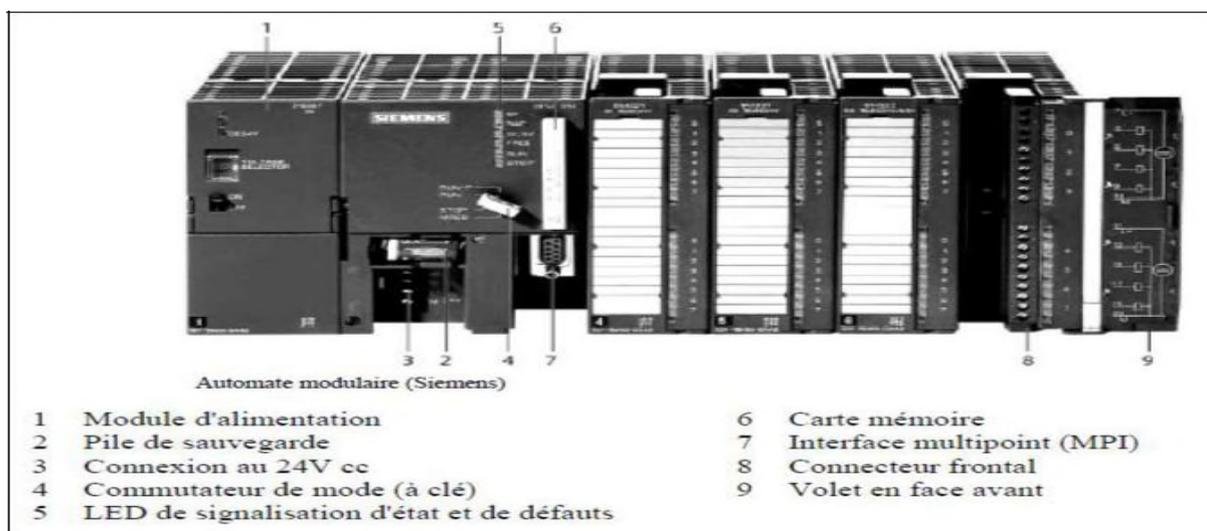


Figure II.2 Les automates de type modulaire [5].

Chapitre II : Les automates programmables industriels

II.2.2.2 Structure interne

De forme compacte ou modulaire, Les sous-ensembles fondamentaux composant un automate programmable sont [9]:

- l'unité centrale ou CPU qui traite les variables en fonction du traitement logique programmé en mémoire et élabore les ordres de commande ;
- la tête de bac qui assure la liaison entre l'unité centrale et les interfaces ;
- les interfaces d'entrée qui reçoivent les données machines provenant des capteurs;
- les interfaces de sortie qui appliquent les processus de commande.

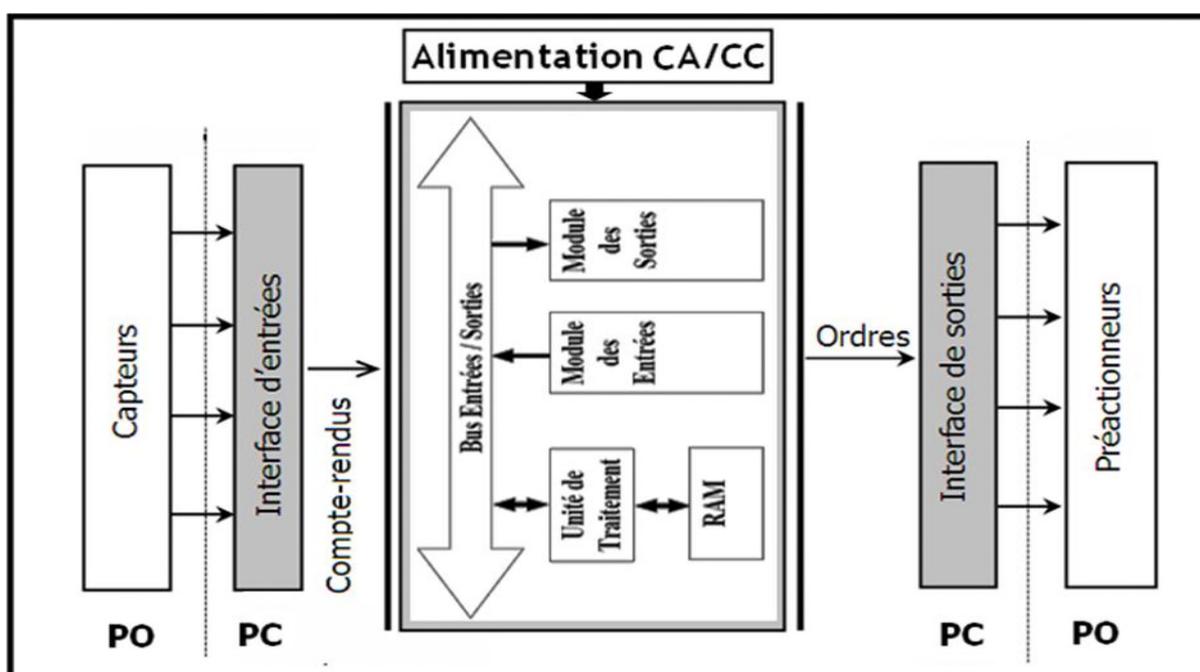


Figure II.3 Structure d'un automate programmable [9]

II.2.2.2.1 Première partie de l'API (Unité de traitement ou processeur) :

C'est le cœur de l'appareil, dans l'unité centrale ; ce n'est pas paradoxalement, le point le plus caractéristique, mais il conditionne tout de même largement les performances. On est passé des processeurs à cycle de scrutation unique (on exécutait en permanence un programme gérant essentiellement des variables binaires) vers des processeurs plus performants, issus du monde de l'informatique.

La principale fonction du processeur est de commander et gouverner les différentes activités du système. Il effectue cette tâche en interprétant et en exécutant un ensemble de programmes système. Ces derniers forment un groupe de programmes superviseurs stockés de façon permanente dans le processeur. Grâce à ces programmes superviseurs le processeur peut ainsi exécuter toutes ses tâches de contrôle, ainsi que divers fonctions. Le processeur gère l'ensemble des échanges d'informations entre les parties de l'API, en assurant la lecture des informations d'entrées, l'exécution des programmes ou les instructions mises en mémoire et la commande ou l'écriture des sorties. Le processeur effectue aussi des opérations logiques sur les bits ou mots binaire, la temporisation et le comptage [10].

II.2.2.2 Deuxième partie de l'API (Mémoires) :

Elles sont étroitement associées à l'unité centrale, La mémoire est l'élément fonctionnel qui peut recevoir, conserver et restituer les informations. C'est un dispositif électronique qui sert à stocker des informations (issues des capteurs d'entrée ou générées par le processeur pour commander les sorties [11].

Nous distinguons selon la possibilité ou non d'écrire et d'effacer ou de lire les différents types de mémoire qu'on peut trouver dans un API [12]:

- R.A.M. (Random Access Memory) : C'est une mémoire vive qui doit être alimentée électriquement pour pouvoir conserver les informations. Avant son exécution, le programme est transféré dans cette mémoire qui permet d'atteindre des vitesses en lecture et écriture très rapides.
- R.O.M. (Read Only Memory) : mémoire à lecture uniquement, appelée également mémoire Morte, elle permet de stocker des informations indéfiniment sans aucune alimentation électrique.
- P.R.O.M. (Programmable Read Only Memory) : mémoire de type ROM qu'on peut programmer une seule fois.
- E.P.R.O.M. (Erasable Programmable Read Only Memory) : mémoire de type ROM mais que l'on peut effacer par exposition du circuit aux rayons ultra-violets.
- E.E.P.R.O.M. (Electrical Erasable Programmable Read Only Memory) : mémoire de type PROM que l'on peut effacer électriquement en écrivant à nouveau sur le contenu de la mémoire. Ce type de mémoire par sa simplicité de mise en œuvre tend à remplacer de plus en plus la mémoire EPROM.

II.2.2.2.3 Troisième partie de l'API (Module d'Entrées/Sorties) :

Les modules d'E/S assurent la liaison entre la partie commande et la partie opérative (figure II.4), ils se situent entre la CPU et le processus.

Pour ce faire ils doivent [10]:

- regrouper les variables de même nature pour diminuer la complexité et le coût.
- assurer le dialogue avec la CPU.
- traduire les signaux industriels en information API et inversement.

Plusieurs types de modules sont disponibles sur les marchés comme :

Modules d'E/S tout ou rien (TOR) Ces modules traitent une information qui ne peut prendre que deux états (vrai ou faux, 0 ou 1), ils constituent l'interface entre l'API et les différents capteurs et pré-actionneurs présents.

Modules d'E/S analogiques Dans ce cas, le signal traité est analogique et prend des valeurs comprises dans une plage bien déterminée. Ces modules sont munis de convertisseur analogique/numérique pour les entrées et respectivement de convertisseur numérique/analogique

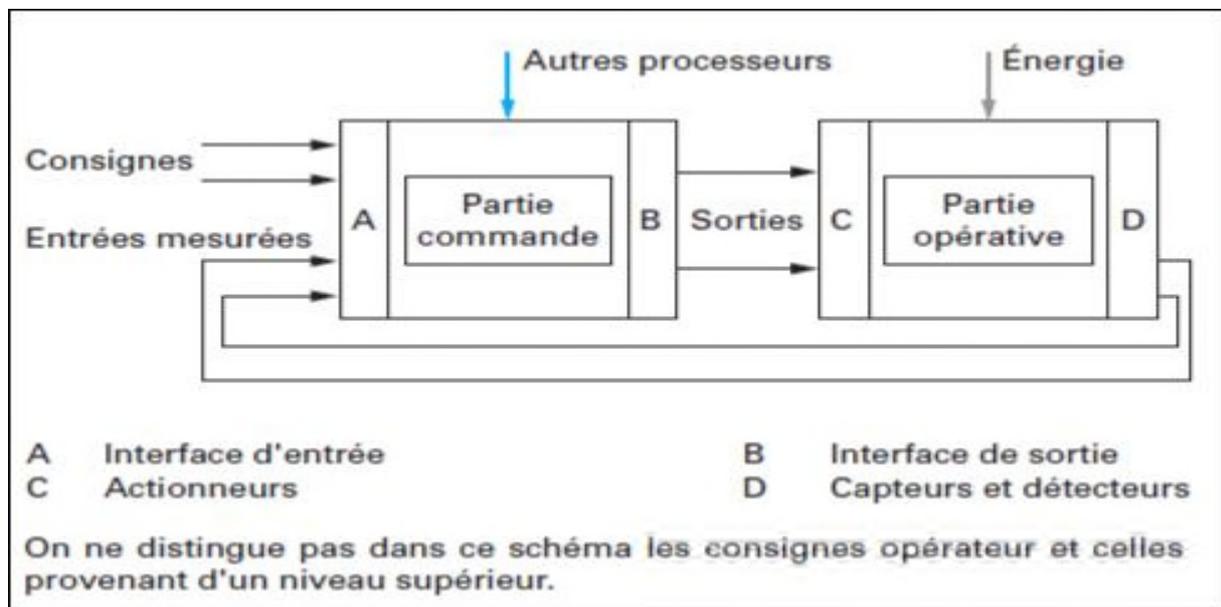


Figure II.4 Liaison entre partie commande et partie opérative [10].

II.2.2.2.4 Quatrième partie de l'API (Module d'alimentation) :

À partir du secteur, plus rarement d'une source de 24 V continu, l'alimentation principale fournit l'énergie nécessaire à l'UC et aux interfaces (sous des tensions de 5, 12, 15, 24 V continu). Elle ne peut que rarement alimenter des sorties, et pas toujours les entrées. Elle doit faire face aux microcoupures du réseau. Un onduleur évite les risques de coupure si celles-ci risquent de dépasser les tolérances admises, 10 ms au plus à 50 Hz avec au moins 1 seconde entre deux coupures. L'alimentation fait l'objet d'une protection avec notamment un disjoncteur magnétothermique. Il est parfois nécessaire, pour lutter contre les perturbations électriques du secteur, d'introduire un transformateur d'isolement. C'est notamment le cas pour les raccordements à un réseau électrique à « neutre flottant ». Il ne faut pas oublier que les châssis d'extension, les E/S déportées, voire certaines E/S directes, exigent aussi une alimentation. Son dimensionnement doit prendre en compte les ajouts possibles de modules et/ou de liaisons fil à fil directes lors de modifications ultérieures de l'installation [10].

II.2.3 automates programmables industriels Siemens

II.2.3.1 Historique

Les premiers automates programmables (Programmable Controller, PC) ont été introduits en 1969 aux Etats unis pour satisfaire aux besoins de l'industrie automobile.

Les années soixante-dix connaissent une explosion des besoins industriels dans le domaine de l'automatique, de la flexibilité et l'évolutivité des Systèmes Automatisés de Production (SAP). Siemens AG est un groupe allemand. Fondé en 1847 par Werner Von Siemens, il réalise des équipements électroniques et électrotechniques. Son siège est à Munich, et c'est l'une des plus grosses entreprises européennes [13].

II.2.3.2 Présentation de la gamme SIMATIC de SIEMENS

Siemens reste le seul à proposer une gamme complète de produits pour l'automatisation industrielle, par le biais de sa gamme SIMATIC. L'intégration globale de tout l'environnement d'automatisation est réalisée grâce à :

- une configuration et une programmation homogène des différentes unités du système.
- une gestion cohérente des données.
- une communication globale entre tous les équipements d'automatisme mis en œuvre.

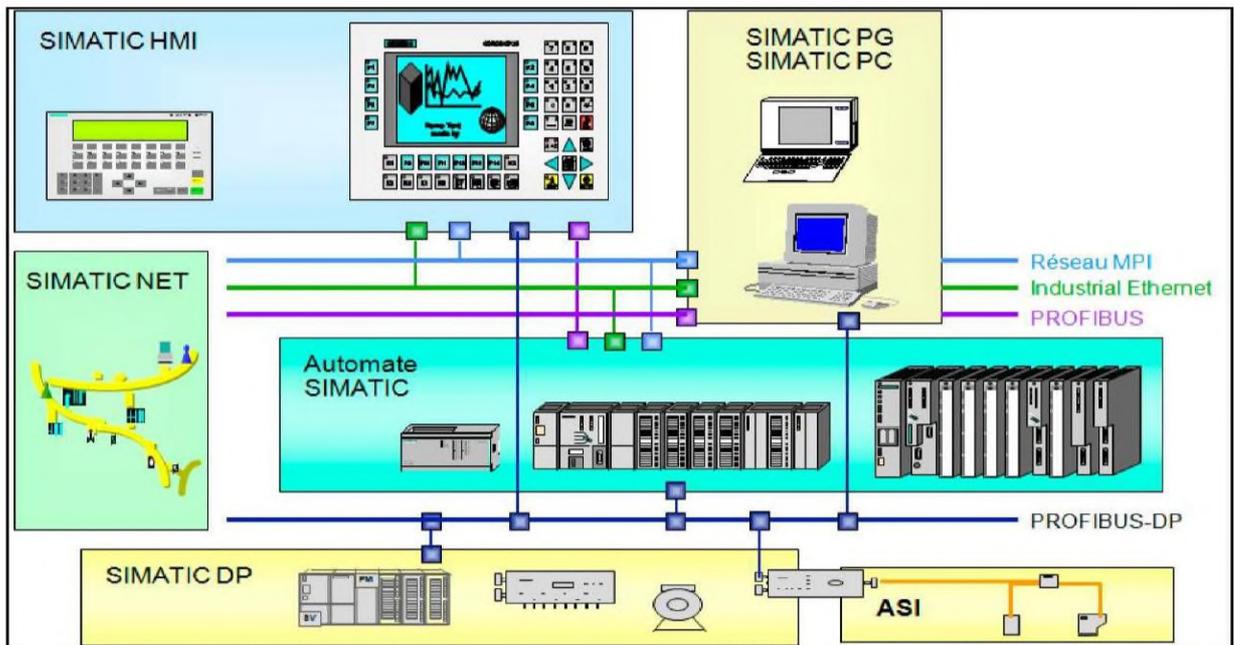


Figure II.5 Présentation de la gamme de SIMATIC [11]

II.2.3.3 Les différentes variantes dans la gamme SIMATIC

❖ SIMATIC S7

Cette gamme d'automates comporte trois familles [11] :

- **S7 200**, qui est un Micro-automate modulaire pour les applications simples, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par l'interface multipoint (MPI) ou PROFIBUS.

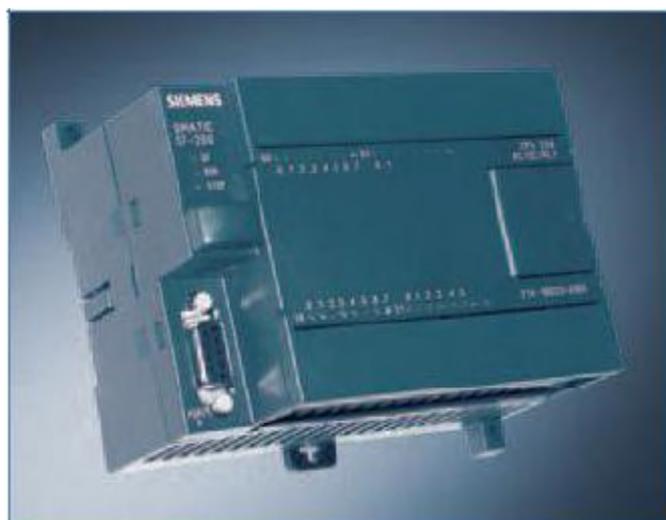


Figure II.6 L'API S7 200

Chapitre II : Les automates programmables industriels

- **S7 300** est un Mini-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI), PROFIBUS et Industriel Ethernet.



Figure II.7 L' API S7 300

- **S7 400** est un automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension a plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (MPI), PROFIBUS ou Industrial Ethernet.



Figure II.8:L' API S7 400

❖ SIMATIC C7

Le SIMATIC C7 combine automate programmable et panneau operateur dans une seule unité, L'automate compte la CPU, les modules d'entrées/sorties, et le panneau operateur qui est utilisé comme une interface Homme/Machine HMI [11].

Le C7 permet la visualisation des états de fonctionnement, des valeurs actuelles du processus et des anomalies.



Figure II.9 La gamme SIMATIC C7

❖ SIMATIC M7

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut être intégré dans un automate S7 300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.

Le M7 300/400 est capable d'effectuer simultanément avec une seule CPU des opérations en temps réels, par exemple des algorithmes complexes de commande, de régulation ainsi que des tâches de visualisation et de traitement informatique. Les logiciels sous DOS ou Windows sont exploitables sur le M7-300. Par ailleurs, avec son architecture normalisée PC, il permet une extension programmable et ouverte de la plate-forme d'automatisation S7[11].



Figure II.10 La gamme SIMATIC M7

II.2.3.4 logiciels, de programmation et de supervision, Step7 et Wincc flexible

II.2.3.4.1 Description du STEP7

STEP 7 est le logiciel de base pour la configuration et la programmation de systèmes d'automatisation SIMATIC. Il fait partie de l'industrie logicielle SIMATIC. Le logiciel de base assiste dans toutes les phases du processus de création de la solution d'automatisation, La conception de l'interface utilisateur du logiciel STEP 7 répond aux connaissances ergonomiques modernes et son apprentissage est très facile [5].

II.2.3.4.2 Gestionnaire de projet SIMATIC

Le gestionnaire de projets SIMATIC (SIMATIC Manager) constitue l'interface d'accès à la configuration et à la programmation.

On peut [5] :

- Configurer et paramétrer le matériel
- ; programmer des modules ;
- Mettre de matériel en réseau ;
- Tester et mettre en œuvre les programmes;

- hors ligne, c'est-à-dire sans qu'un automate soit raccordé ;
- en ligne, c'est-à-dire avec un automate raccordé;

II.2.3.4.3 Stratégie pour la conception d'une structure programme complète et optimisée

La mise en place d'une solution d'automatisation avec STEP 7 : nécessite la réalisation des tâches fondamentales suivantes [5]:

❖ **Création du projet SIMATIC Step7**

➤ **Définition**

Un projet contient la description complète de l'automatisme Il comporte donc deux grandes parties : **la description matérielle** et **la description du fonctionnement**. Le fonctionnement du programme est structuré comme suit :

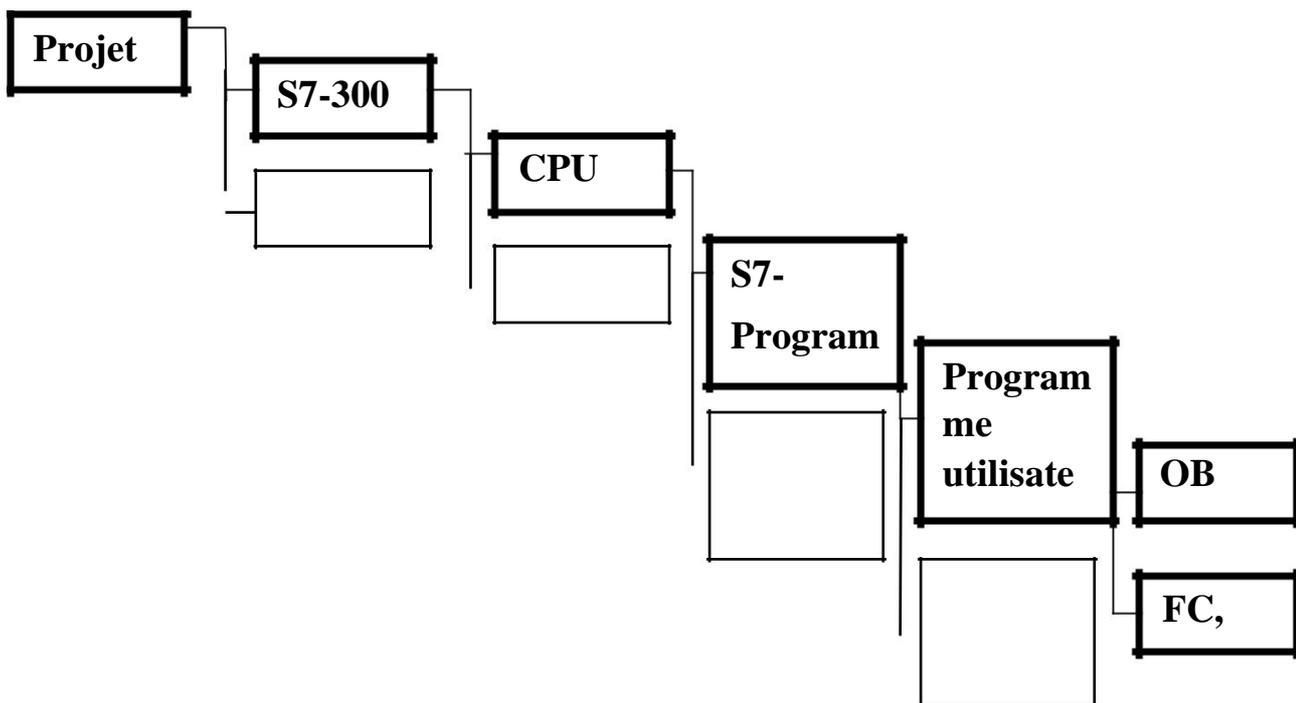


Figure II.11 Organisation structurelle d'un projet

➤ SIMATIC Manager

SIMATIC Manager est l'écran d'accueil de programmation avec STEP7. Il affiche l'arborescence de tous les objets d'un projet et à travers ces objets, Il permet d'accéder à toutes les fonctions qui sont nécessaires pour résoudre le problème d'automatisation. Depuis le SIMATIC Manager, on peut configurer et paramétrer S7-300 et le programmer.

Le tableau ci-après contient les objets de STEP 7 qu'il nous faut connaître Ce tableau montre les icônes affectées aux différents objets et leur signification :

Icône	Objet	Description	Se trouve dans le classeur :
	Projet	Il représente l'ensemble des données et des programmes d'un automatisation.	Situé au sommet dans la hiérarchie
	Station SIMATIC 300	Il représente une configuration d'automate comportant un ou plusieurs modules programmables.	Projet
	Module programmable	Il s'agit d'un module programmable (CPU).	Station
	Programme S7 (hors ligne)	Il contient la table des mnémoniques, le programme utilisateur (hors ligne) et les sources, en mémoire de la PG ou du PC.	Module programmable ou projet
	Programme utilisateur (en ligne)	Il contient les modules exécutables qui sont chargés dans votre S7-300.	Programme S7 (en ligne)
	Bloc (hors ligne) Bloc (en ligne)	Ce sont par exemple : <ul style="list-style-type: none"> des blocs de code (OB et FC). 	Programme utilisateur

Figure II.12 les différentes icônes existant dans STEP 7

❖ Configuration matérielle (Partie Hardware)

Une fois le projet créé, on doit configurer l'API ; pour cela on double clic sur (SIMATIC 300) puis sur icône matériel on obtient une liste on entrant sur SIMATIC 300 puis rack-profil support.

On commence par le choix du châssis selon la station choisie auparavant, Pour la station SIMATIC S300, on aura le châssis RACK-300 qui comprend un rail profile.

Sur ce profile :

L'alimentation préalablement sélectionnée se trouve dans l'emplacement n°1. Parmi celles proposées notre choix s'est porte sur la PS-307 5A .

La CPU 315-2DP est impérativement mise a l'emplacement n°2.

L'emplacement n°3 est réservé comme adresse logique pour un coupleur dans une configuration multi châssis.

Chapitre II : Les automates programmables industriels

A partir de l'emplacement 4, il est possible de monter au choix jusqu'à 8 modules de signaux (SM), processeurs de communication (CP) ou modules fonctionnels (FM). Après cela il ne nous reste qu'à enregistrer et compiler.

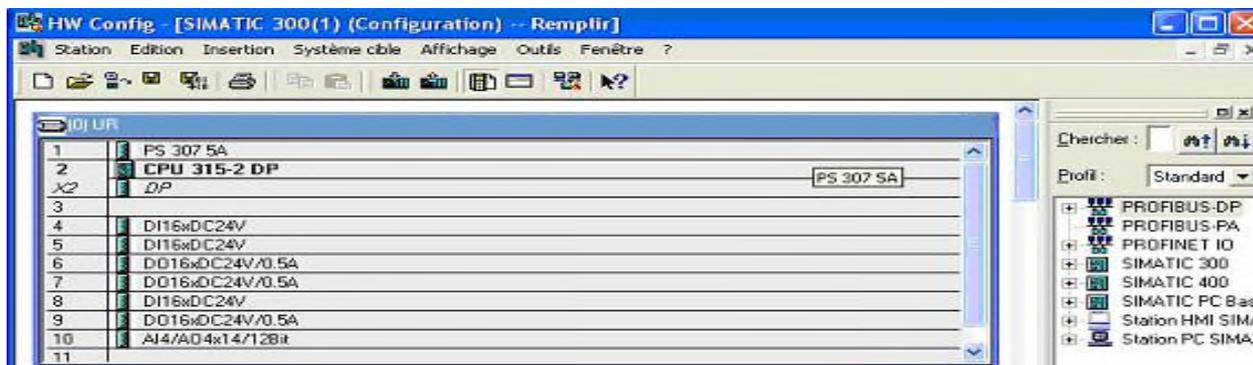


Figure II.13 Configuration matérielle

❖ Création de la table des mnémoniques (Partie Software)

Dans tout programme il faut définir la liste des variables qui vont être utilisées lors de notre programmation. Pour cela la table des mnémoniques est créée. L'utilisation des noms appropriés rend le programme plus compréhensible et plus facile à manipuler. Ce type d'adressage est appelé relatif.

Pour créer cette table, on suit le cheminement suivant :

Développer « SIMATIC 300 », puis « CPU » et « programme S7 ». Double cliquer sur l'icône « mnémonique » qui apparaît dans la partie droite de l'écran :

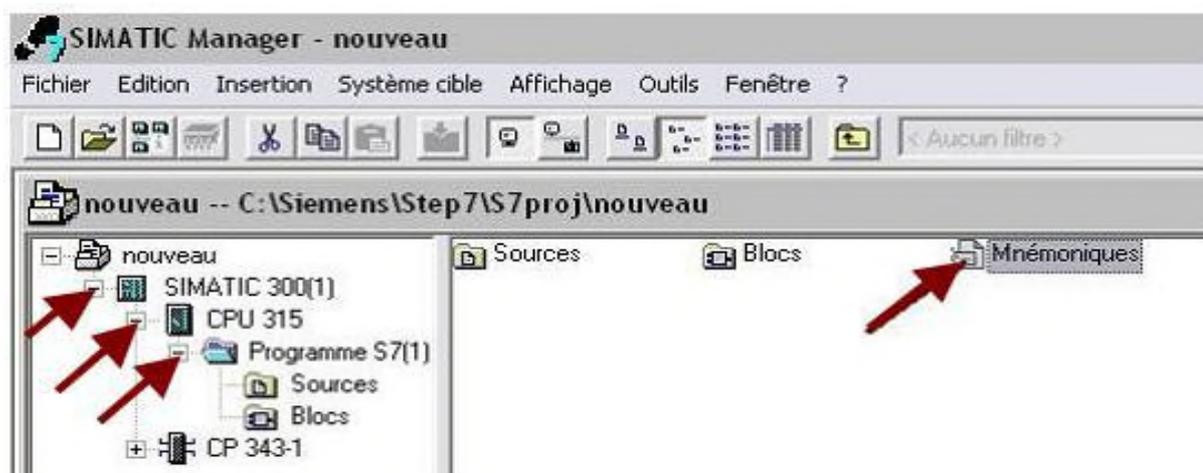
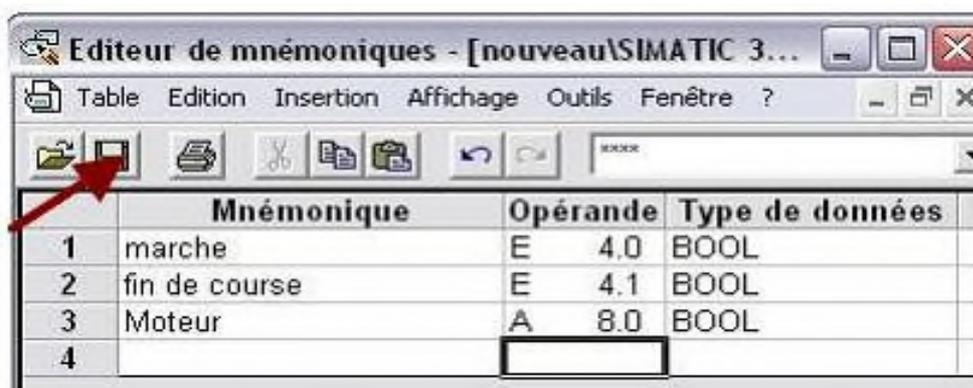


Figure II.14 Les étapes à suivre pour créer la table de mnémoniques

Chapitre II : Les automates programmables industriels

La colonne « mnémorique » doit contenir une description simple de la variable. Indiquer dans la colonne « opérande » l'adresse de la variable (E pour entrer, A pour sortie . . .). Le logiciel donne le type de données dans la colonne suivante (ici « bool » pour booléen, c'est-à-dire binaire), mais celui-ci peut être modifié en cas de besoin. Voici un exemple de table de mnémoniques :



The screenshot shows a software window titled 'Editeur de mnémoniques - [nouveau\SIMATIC 3...'. The window has a menu bar with 'Table', 'Edition', 'Insertion', 'Affichage', 'Outils', and 'Fenêtre'. Below the menu bar is a toolbar with various icons. A red arrow points to the 'Table' icon in the toolbar. The main area of the window contains a table with the following data:

	Mnémorique	Opérande	Type de données
1	marche	E 4.0	BOOL
2	fin de course	E 4.1	BOOL
3	Moteur	A 8.0	BOOL
4			

Figure II.15 Exemple d'une Table de mnémorique

❖ Editeur de programme et langages de programmation

Les langages de programmation CONT, LIST et LOG, font la partie intégrante du logiciel de base d'un API[5]:

- **Le schéma à contacts (CONT)** est un langage de programmation graphique. La Syntaxe des instructions fait penser aux schémas de circuits électriques. Le langage CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts, les éléments complexes et les bobines.
- **La liste d'instructions (LIST)** est un langage de programmation textuel proche de la machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme.
- **Le logigramme (LOG)** est un langage de programmation graphique qui utilise les boites ou fonctions de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boites logiques.

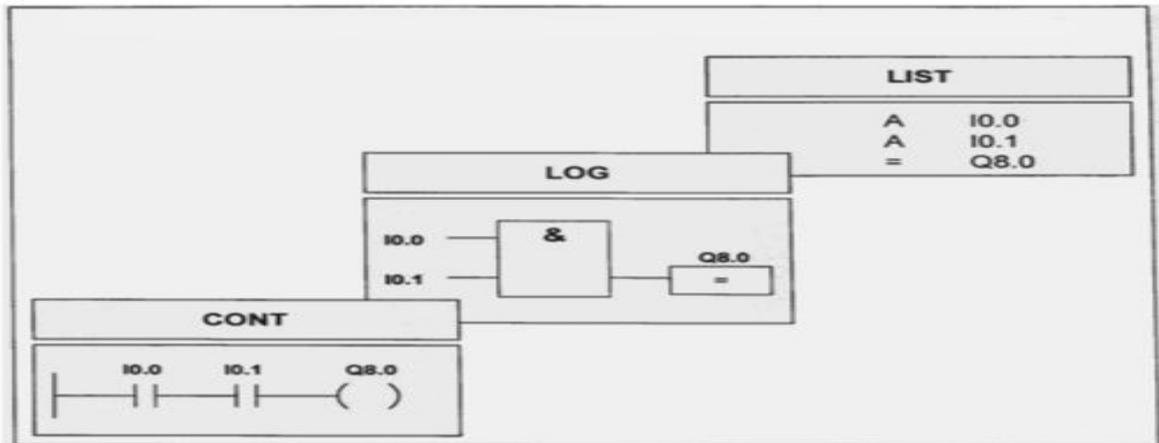


Figure II.16 Mode de représentation des langages basiques de programmation STEP7

On dispose de langages de programmation plus évolués, au détriment de l'optimisation mémoire:

- **GRAPH** est un langage de programmation permettant la description aisée de Commandes séquentielles (programmation de graphes séquentiels). Le déroulement du processus y est subdivisé en étapes. Celles-ci contiennent en particulier des actions pour la commande des sorties. Le passage d'une étape à la suivante est soumis à des conditions de transition.

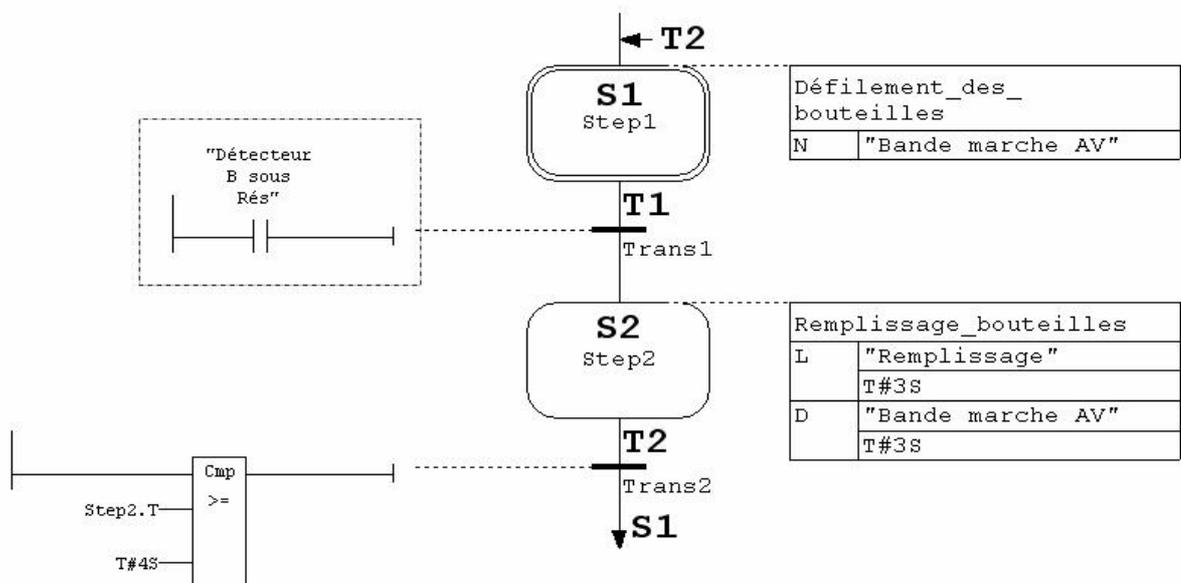


Figure II.17 Exemple de programmation avec langage graph

- **Hi Graph** est un langage de programmation permettant la description aisée de processus asynchrones non séquentiels sous forme de graphes d'état. A cet effet, l'installation est subdivisée en unités fonctionnelles pouvant prendre différents états. Ces unités fonctionnelles peuvent se synchroniser par l'échange de messages.

- **SCL** est un langage évolué textuel. Il comporte des éléments de langage que l'on trouve également sous une forme similaire dans les langages de programmation Pascal et C. SCL convient donc particulièrement aux utilisateurs déjà habitués à se servir d'un langage de programmation évolué.

- **CFC** pour S7 et M7 est un langage de programmation graphique permettant l'interconnexion graphique de fonctions existantes. Ces fonctions couvrent un large éventail allant de combinaisons logiques simples à des régulations et commandes complexes. Un grand nombre de ces fonctions est disponible sous la forme de blocs dans une bibliothèque.

II.2.3.4.4 Surveillance du fonctionnement et diagnostic du matériel

Pour le test du bon fonctionnement du programme, le logiciel optionnel de simulation permet d'exécuter et de tester le programme dans un système d'automatisation qu'on simule dans

L'ordinateur ou dans la console de programmation en y accédant par la fonction  dans la barre d'outils.

Pour la recherche d'erreurs, des icônes de diagnostic permettent de déceler des défauts sur un module et indiquent l'état de ce dernier. Pour une CPU, par exemple, son état de fonctionnement (**RUN, STOP ou RUN-P**).

Les icônes de diagnostic s'affichent dans la vue en ligne de la fenêtre du projet, dans la vue rapide (présélection) ou encore dans la vue de diagnostic lors d'un appel de la fonction "Diagnostic du matériel".

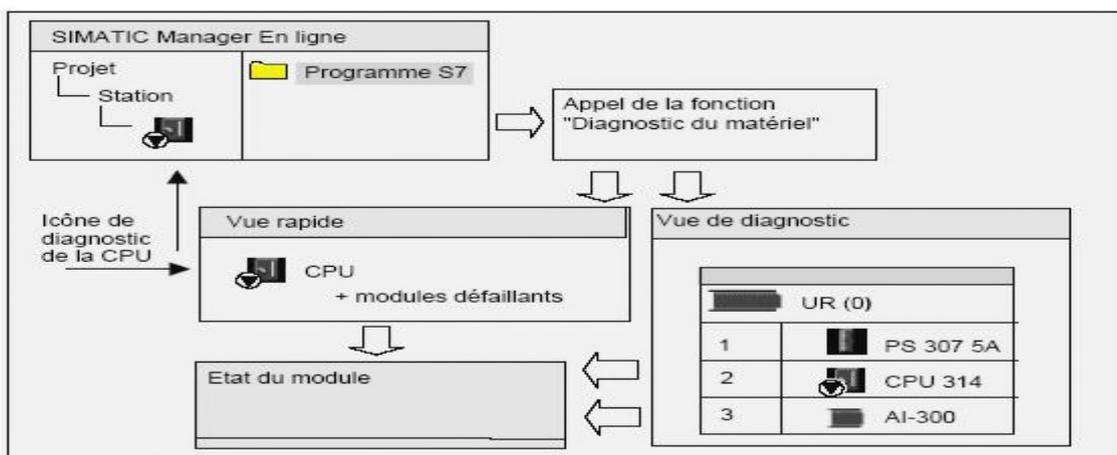


Figure II.18 Diagnostic du matériel

➤ Simulation avec STEP 7

L'application de simulation de modules S7-PLCSIM permet d'exécuter et de tester le programme dans un automate programmable (AP) qu'on simule dans un ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux) L'AP S7 de simulation permet de tester des programmes destinés aux CPU S7 300 et aux CPU S7-400, et de remédier à d'éventuelles erreurs.

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (comme, par exemple, d'activer ou de désactiver des entrées). Tout en exécutant le programme dans l'AP de simulation, on a également la possibilité de mettre en œuvre les diverses applications du logiciel STEP 7 comme, par exemple, la table des variables (VAT) afin d'y visualiser et d'y forcer des variables.

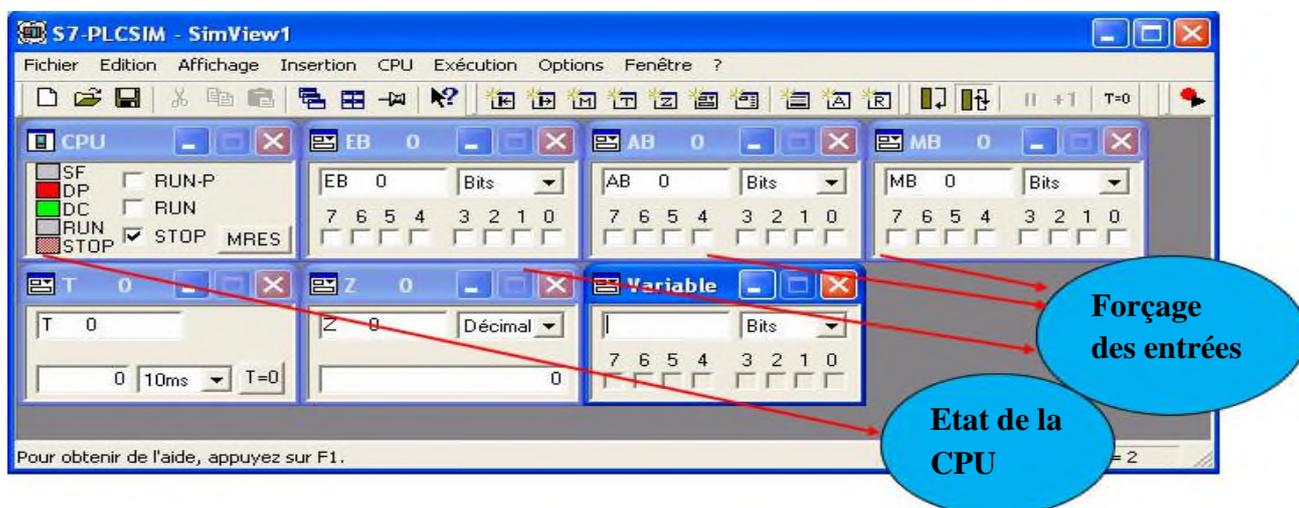


Figure II.19 Logiciel de simulation PLC-SIM

II.2.3.5 Logiciel de conception et configuration d'interface WinCC flexible

II.2.3.5.1 Définition

WinCC Flexible 2008 est l'Interface Homme-Machine (IHM) idéale pour toutes les applications au pied de la machine et du processus dans la construction d'installations, de machines et de machines de série. De par sa conception généraliste, WinCC Flexible permet de disposer d'un logiciel d'ingénierie pour tous les terminaux d'exploitation SIMATIC HMI, du plus petit pupitre Micro jusqu'au Multi Panel.

Créer l'interface graphique et les variables, c'est pouvoir lire les valeurs du processus via l'automate, les afficher pour que l'opérateur puisse les interpréter, et ajuster, éventuellement, le processus, toujours via l'automate [5].

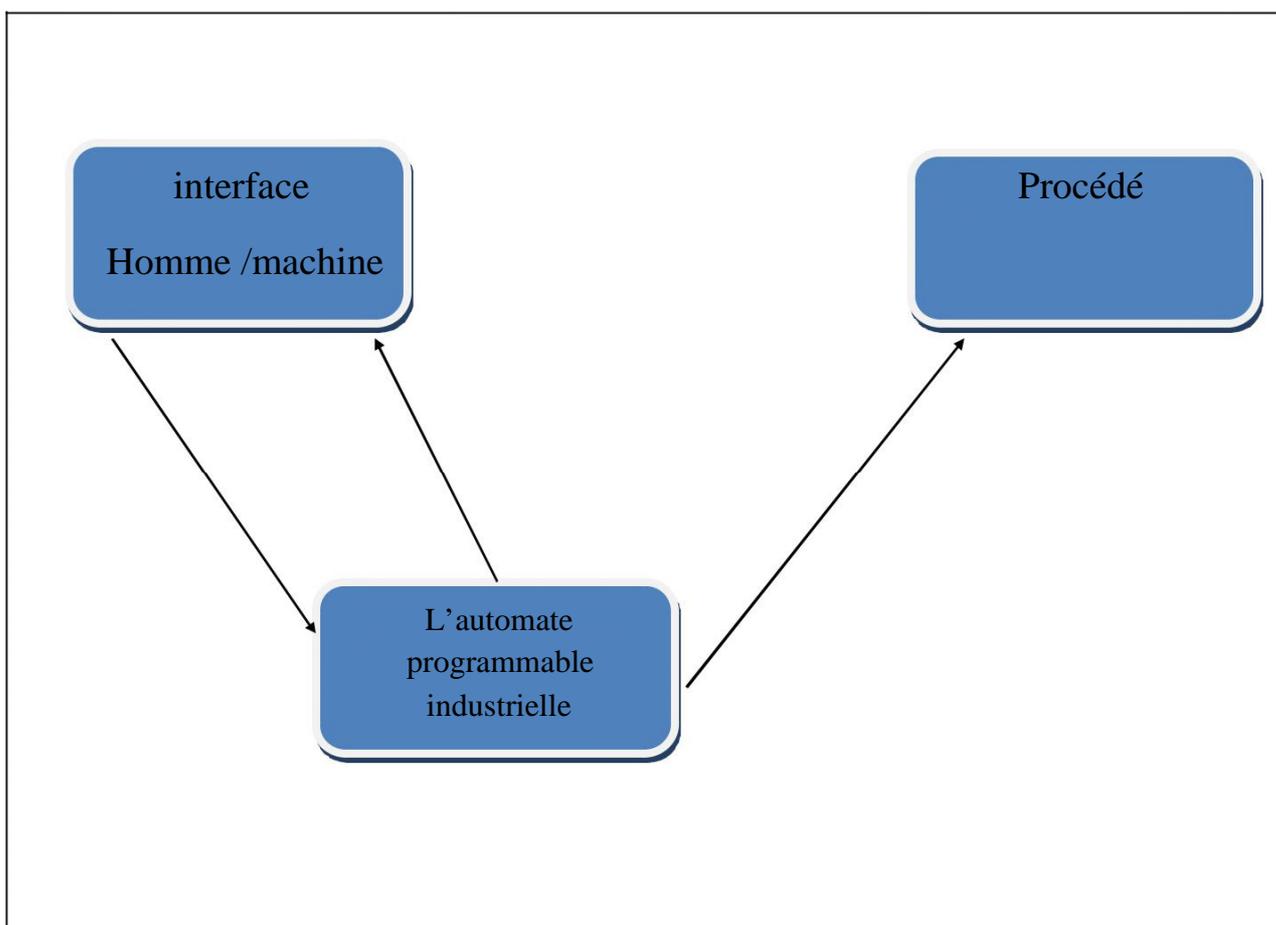


Figure II .20 L'interface Homme/Machine dans processus automatisé

II.2.3.5.2 Elément du progiciel Win CC flexible

L'utilisation de ce logiciel pour la conception d'une interface, passe par plusieurs étapes Résumé dans la figure qui suit [5]:

L'environnement de travail de Win CC flexible se compose de plusieurs éléments.

Certains de ces éléments sont liés à des éditeurs particuliers, visibles uniquement lorsque cet éditeur est actif. Il met à disposition un éditeur spécifique pour chaque tâche de configuration.

On peut configurer l'interface utilisateur graphique d'un pupitre opérateur avec l'éditeur "Vues". Pour la configuration d'alarmes, on peut utiliser l'éditeur "Alarmes TOR" ou autre.

Dans WinCC flexible, chaque projet créé contient principalement des vues que l'on crée pour le contrôle-commande des machines et des installations. Lors de la création des vues, on dispose d'objets prédéfinis permettant d'afficher des procédures et de définir des valeurs du processus.

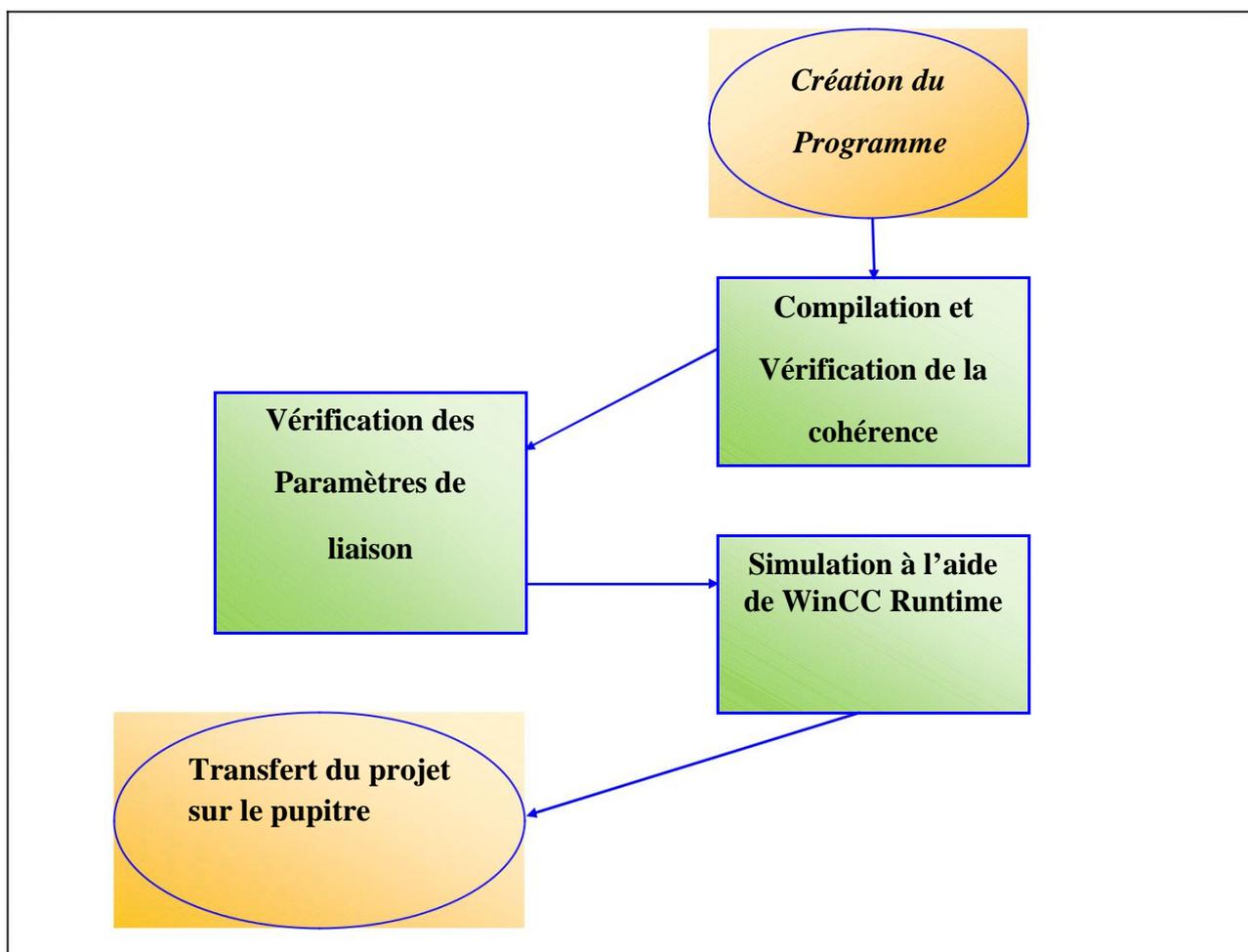


Figure II .21 Etapes pour la conception d'une interface via Win CC flexible

II.2.3.5.3 Les différents outils et barres de l'éditeur de vues

Comme éléments on trouve [5]:



Barre des menus

La barre des menus contient toutes les commandes nécessaires à l'utilisation de WinCC flexible. Les raccourcis disponibles sont indiqués en regard de la commande du menu.



Barres d'outils

La barre d'outils permet d'afficher tout dont le programmeur à besoin.



Boîte à outils

La fenêtre des outils propose un choix d'objets simples ou complexes qu'on insère dans les vues, p. ex. des objets graphiques et éléments de commande.



Zone de travail

La zone de travail affiche toutes les variables sous forme de tableau. on éditez les Attributs des variables dans les cellules du tableau. En cliquant sur le titre d'une colonne, on trie la table suivant les entrées de cette colonne. on peut configurer la sélection de colonne en fonction de vos besoins : Certaines colonnes ne sont pas disponibles sur tous les pupitres. A l'enregistrement du projet, la sélection est enregistrée automatiquement. Elle est liée au nom d'utilisateur qui a été entré à l'ouverture de session dans Microsoft Windows.



Fenêtre des propriétés

Pour configurer les variables, la fenêtre des propriétés fournit les mêmes informations et possibilités de configuration que la table de la zone de travail. La fenêtre des propriétés présente, sur sa partie gauche, une arborescence qui permet de sélectionner toutes les catégories de propriétés. Les champs de configuration de la catégorie sélectionnée sont affichés dans la partie droite de la fenêtre.

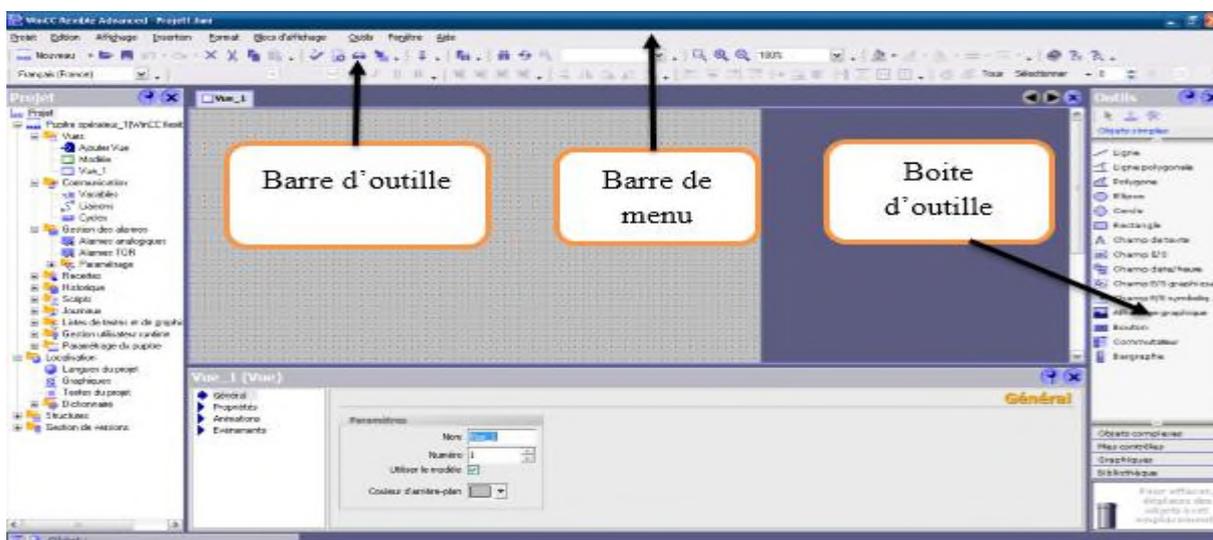


Figure II.22 Vue d'ensemble du progiciel Win CC flexible

II.2.3.5.4 Utilisation des variables

On distingue deux types de variables, les variables externes et les variables internes [5]:

❖ Variable externe :

Les variables externes permettent de communiquer, c.-à-d. d'échanger des données entre les composants d'un procès automatisé, entre un pupitre opérateur et un automate

Une variable externe est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture à cette cellule mémoire est possible aussi bien à partir du pupitre opérateur que de l'automate.

Les variables externes étant une image d'une cellule mémoire de l'automate, les types de données utilisables dépendent de l'automate auquel le pupitre opérateur est connecté. Si on configure STEP 7 ou SIMOTION Scout intégré, on peut accéder pour la création des variables externes directement à toutes les variables qui ont été créées lors de la programmation de l'automate avec STEP 7 ou SIMOTION Scout.

Si on utilise des systèmes distribués, on peut directement accéder aux variables des autres systèmes IHM via l'interface OPC.

❖ Variables internes :

Les variables internes ne possèdent aucun lien avec l'automate, ils sont enregistrées dans la mémoire du pupitre. Le pupitre opérateur en question peut donc accéder en lecture et en écriture aux variables internes. Les variables internes sont créées p. ex. pour exécuter des calculs locaux. on peut utiliser pour les variables internes tous les types de données de base. on trouve une liste détaillées des types de données sous "Types de données de base".



Nom	Liaison	Type de donn...	Adresse	Eléments du tableau	Cycle d'acquisiti...	Comr
Variable_1	Connection_1	Int	VW 0	1	1 s	
Variable_2	Connection_1	Int	VW 2	4	1 s	
Variable_3	<Variable interne>	Int	<Pas d'ad...>	1	1 s	
Variable_4	<Variable interne>	Int	<Pas d'ad...>	1	1 s	

Figure II.23 Editeur de variable dans le Win CC flexible

Chaque variable peut être configurée selon le besoin et cela en accédant à sa fenêtre de priorité après l'avoir sélectionnée.

II.2.3.5.5 Fenêtre des propriétés

Pour configurer les variables, La fenêtre des propriétés fournit les mêmes informations et possibilités de configuration que la table de la zone de travail. La fenêtre des propriétés présente, sur sa partie gauche, une arborescence qui nous permet de sélectionner toutes les catégories de propriétés. Les champs de configuration de la catégorie sélectionnée sont affichés dans la partie droite de la fenêtre [5].

➤ **Fenêtre des propriétés pour les variables**



Figure II. 24 Vue d'une fenêtre des propriétés pour les variables

✓ **Types de données**

Les types de données de base sont disponibles pour toutes les configurations. Pour les variables externes, on dispose de types de données additionnels, spécifiques à l'automate connecté. On trouve une liste détaillée des types de données de base et des types de données spécifiques à une connexion à des automates S7 sous "Types de données en cas de connexion à S7". On trouve des informations sur les types de données disponibles en cas de connexion à d'autres automates dans la documentation sur les pilotes de communication correspondants.

On peut en outre générer types de données en créant des structures.

II.2.3.5.6 Actualisation de valeurs de variables dans les scripts et les fonctions

Les scripts et les fonctions système accèdent à la valeur d'une variable externe qui est stockée dans la mémoire du runtime. Au démarrage du runtime, la valeur actuelle est lue dans l'automate et stockée dans la mémoire du runtime. Après cela, la valeur de la variable est actualisée suivant le temps de cycle réglé. Les scripts et les fonctions accèdent d'abord aux valeurs de variable qui ont été lues dans l'automate au cycle précédent [5].

II.2.3.5.7 Compilation et Simulation

Après avoir créé le projet et terminé la configuration, il est indispensable de vérifier la cohérence du projet, de contrôler la cohérence et de chercher les erreurs, à l'aide de la commande sur la barre du menu contrôle de la cohérence. Après le contrôle de cohérence, le système crée un fichier de projet compilé. La simulation permet de détecter des erreurs logiques de configuration, par exemple, des valeurs limites incorrectes, et cela à l'aide du simulateur Runtime par la commande « Démarrer le système Runtime du simulateur » [5].

II.3 Conclusion

On constate la facilité et la souplesse qu'offre L'A.P.I pour sa programmation, connexion, et adaptation aux conditions industrielles, avec toutes les fonctionnalités indispensables à l'automatisation des processus. La diversité des possibilités mise en œuvre et son coût, le rendent incontournable lors de l'élaboration d'une solution.

Ce chapitre nous a permis de voir les différentes possibilités offertes par les automates programmables : simulation et supervision des systèmes séquentiels.

Chapitre III :

Description

de

la machine

III.1 Introduction

Pour faciliter le travail d'étude et la compréhension du fonctionnement d'un système industriel, il est important de connaître le matériel qui entre dans sa constitution. Ce présent chapitre est consacré à la représentation des différents équipements du déviateur ainsi que leur identification.

III.2 Descriptions du déviateur

C'est un mécanisme industriel destiné à répartir les bouteilles sur les différents couloirs d'une façon homogène pour quelle soient regroupées dans les paquets enveloppés par la suite.

Le déviateur est un système de positionnement à base d'un servomoteur et d'un système de guidage (le bras de positionnement). Ce système offre un contrôle à haut rendement et une grande précision [14].



Figure III.1 Vue générale de la machine.

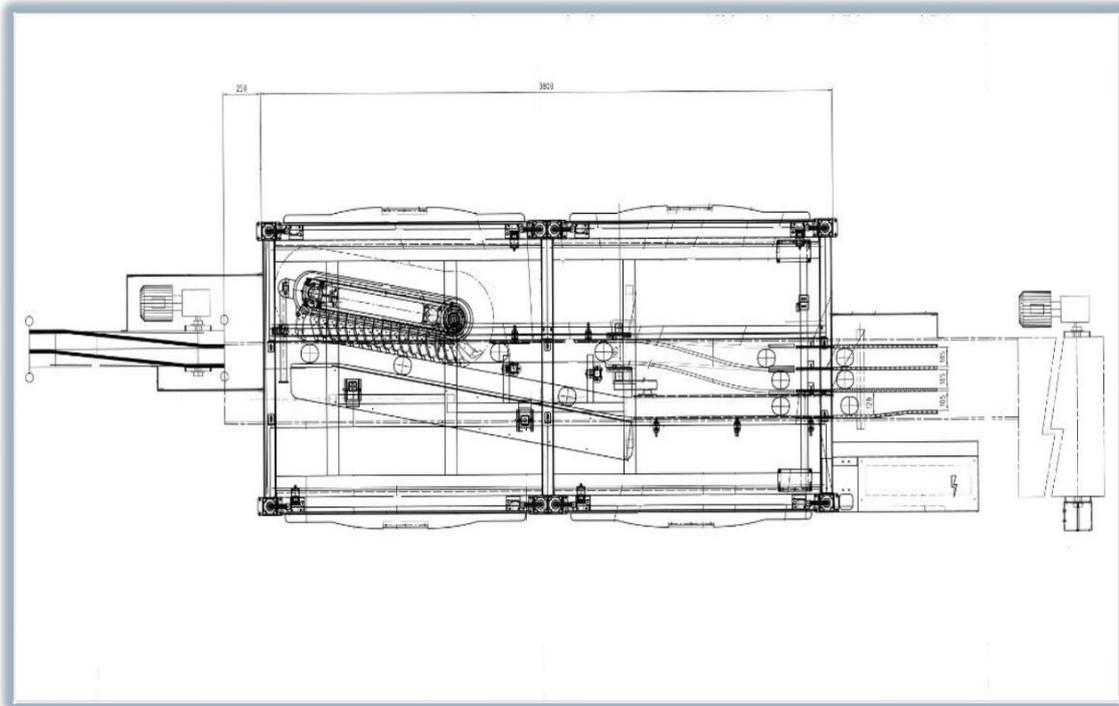


Figure III.2 Vue de dessus du séparateur de bouteilles [15].

III.3 Partie mécanique

III.3.1. Mécanisme de positionnement

C'est la partie essentielle de ce système, elle est constituée de : rail de guidage, chaîne porte câbles, le bras de déviateur, guide sur rail, les courroies dentées, les poulies dentées et arbre de transmission.

Il existe trois rails de guidages avec bande de protection, chaque rail possède trois guides fixés sur les courroies afin d'effectuer un mouvement de translation. Ils sont très rigides et utilisés dans plusieurs domaines qui nécessitent des guidages compacts comme des robots industriels.

Les guides sur rail sont véhiculés à travers de trois courroies. Ces derniers sont souvent utilisés dans plusieurs applications de transmission du mouvement à cause de leur fiabilité et rentabilité. En cas d'une utilisation difficile, de fortes accélérations et décélérations ou un positionnement précis des produits, l'adaptation des courroies dentées est indispensable [14].

Les poulies sont aussi utilisées pour la transmission d'un mouvement de rotation d'un arbre menant à un arbre mené relativement équipé de l'autre, par l'intermédiaire des courroies dentées.

Les chaînes porte câbles sont utilisées pour éviter l'encombrement du câblage et assurent le bon fonctionnement sans interruption des systèmes de commande.

Un autre élément de cette partie, c'est le bras déviateur qui assure le guidage des bouteilles à une entrée désirée du couloir. Il est constitué de deux glissières montées en parallèle, d'une partie fixe sur la

Sortie du tapis qui vient de l'étiqueteuse et d'une autre partie fixe au-dessous du guidage, qui fait le mouvement transversal.

Enfin, un tapis roulant qui est entraîné par des pignons montés sur l'arbre d'entraînement. Un moteur asynchrone équipé d'un réducteur de vitesse à roue et vis sans fin est destiné à mouvoir cet arbre d'entraînement. L'utilisation d'un réducteur à roue et vis sans fin a pour but de réduire l'encombrement, et d'avoir un rapport de réduction important. La manutention des bouteilles au niveau du déviateur et assurée par ce tapis roulant [14].

III.4 Partie capteurs et instrumentation

Un capteur est un objet technique de prélèvement d'information sur un processus. Il réalise la conversion d'une grandeur physique (vitesse, position, température, pression...) en une grandeur électrique ou pneumatique. Cette dernière est exploitée par la partie commande. Les informations de sortie du capteur peuvent être de caractère analogique, logique ou numérique.

Les capteurs possède trois parties différentes : une première partie qui a pour rôle de détecter l'information, une seconde partie qui a pour rôle de transformer l'information en un signal (électrique ou pneumatique) et une dernière partie qui a pour rôle de d'envoyer le signal récupéré au circuit de commande après amplification, linéarisation ; mise en forme et correction[3].

Le système possède quatre types de capteurs :

- capteurs de fin de course ;
- détecteurs de proximité ;
- détecteur photoélectrique ;
- codeurs rotatifs ;

III.4.1 Capteurs fin de course

Dans ce système il existe quatre capteurs de fin de course qui sont utilisés pour la sécurité et comme des limites de déplacements. Ces capteurs sont caractérisés par une tête de mouvement angulaire[3].

Les capteurs mécaniques de position, appelés aussi interrupteurs de position, sont surtout employés dans les systèmes automatisés pour assurer la fonction « détecter les positions ». Ils sont réalisés à la base de microcontacts placés dans un corps de protection et muni d'un système de commande ou tête de commande[3].

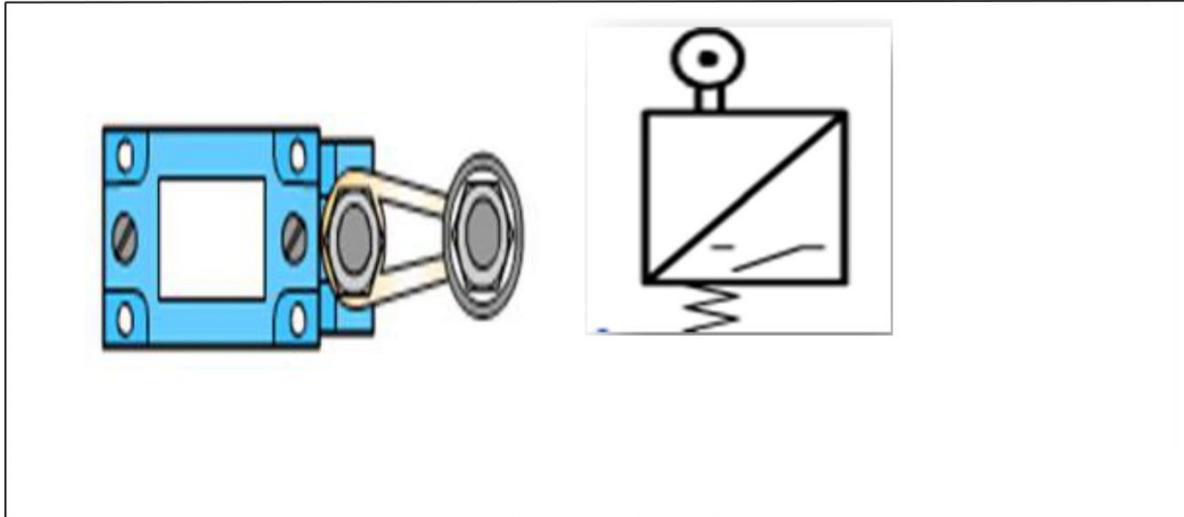


Figure III.3 Détecteur de position et symbole[3].

Ce capteur de position est utilisé dans plusieurs applications à savoir la détection de présence ou de position des objets solides, ainsi que les fins de course.

Ces capteurs possèdent les caractéristiques suivantes[3] :

- sécurité de fonctionnement élevée ;

- mise en œuvre simple, fonctionnement visualisé et grande résistance aux ambiances industriels ;

- bonne fidélité sur les points d'enclenchement ;

- tension d'emploi élevée ;

- bonne aptitude à commuter les courants faibles combinés à une grande endurance électrique ;

- fiabilité des contacts et manœuvre positive d'ouverture.

III.4.2 Détecteur de proximité

Les détecteurs de proximité sont utilisés afin de déterminer le sens de déplacement et d'initialiser le cycle de fonctionnement. Dans ce système il y a deux détecteurs de proximité inductive PX1 et PX2 de type XS1 M12PA37OD[15].

III.4.2.1 Définition

Ces détecteurs fonctionnent grâce à la variation d'un champ électromagnétique perturbé par la proximité d'un objet métallique.

La distance de détection varie de 1 à 60 mm selon le type de capteur, les conditions d'utilisations et de la nature de l'objet à détecter (acier, aluminium, cuivre ...)[15].

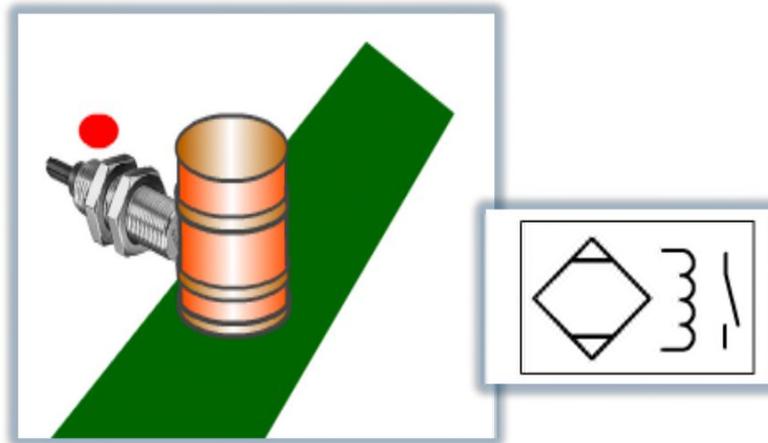


Figure III.4 Détecteur de proximité inductif et symbole [3].

III.4.2.2 Principe de fonctionnement

Un détecteur inductif se compose essentiellement d'un oscillateur dont les bobines constituent la face sensible du capteur. Ainsi, à l'avant du capteur, un champ magnétique alternatif est créé.

Lorsqu'un objet conducteur est placé dans ce champ, il développe à sa surface des courants induits qui contrarient le champ magnétique initial et provoquent l'arrêt des oscillations.

Un circuit électronique placé à l'intérieur du capteur détecte cette modification et délivre alors un signal de sortie [15].

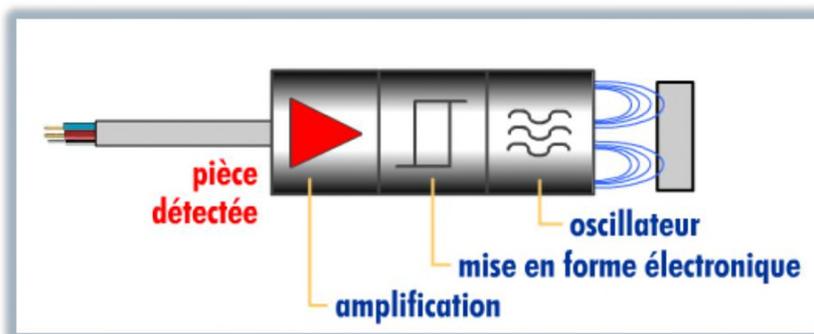


Figure III.5 Principe de fonctionnement[3].

Ce type de détecteurs se rencontre dans les secteurs suivants[15] :

Convoyage, manutention, domaines d'applications de l'usinage, assemblage, agro-alimentaire, robotique.

Les détecteurs de proximité inductifs possèdent les caractéristiques suivantes :

Détecteur statique, pas de pièce en mouvement

détecteur statique, pas de pièce en mouvement.

détecteur statique, pas de pièce en mouvement

produit entièrement encapsulé dans la résine (étanche).

très bonne tenue à l'environnement industriel (atmosphère polluante).

pas d'usure, durée de vie indépendante du nombre de manœuvres.

pas de contact physique avec l'objet détecté : possibilité de détecter des objets

fragiles, fraîchement peints.

III.4.3 Détecteur photoélectrique

Ce détecteur est utilisé pour le comptage des bouteilles par leur présence. Il est placé sur la partie fixe du bras de déviation. Il se compose essentiellement d'un émetteur de lumière associé à un récepteur sensible à la quantité de lumière reçue (phototransistor). Le type de détecteur utilisé est XUM H07[15].

III.4.3.1 Définition

Un détecteur photoélectrique réalise la détection d'une cible, qui peut être un objet ou une Personne, au moyen d'un faisceau lumineux.

Il est constitué d'un émetteur (qui est généralement une diode électroluminescente) associé à un récepteur de lumière (généralement un phototransistor).

Il délivre une information chaque fois que le faisceau lumineux issu de l'émetteur est interrompu par un obstacle (objet opaque). Le récepteur détecte la coupure de ce faisceau.

Si l'émetteur et le récepteur sont dans le même boîtier, le faisceau lumineux doit être renvoyé par un réflecteur ou par l'objet à détecter (objet réfléchissant)[15].

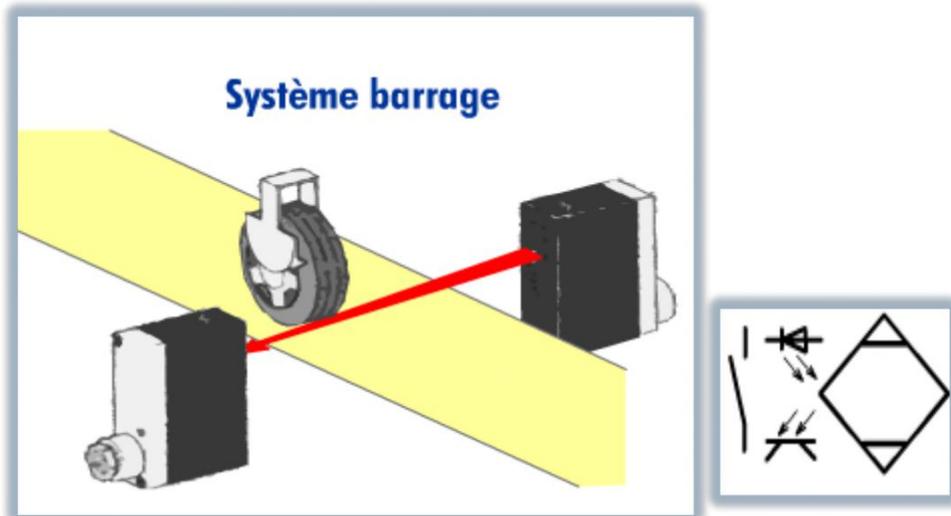


Figure III.6 Exemple d'application Détecteur photoélectrique (système barrage) [15].

Ce type de détecteur est utilisé dans la [15] :

- détection d'objet et de produits dans la manutention et le convoyage.
- détection de pièces dans les secteurs de la robotique.
- détection de personnes, véhicules ou d'animaux dans les secteurs des Ascenseurs et du bâtiment en général.

Ces détecteurs sont caractérisés par [15] :

- détection à très grande distance.
- pas de contact physique avec l'objet détecté.
- généralement en lumière infrarouge invisible, indépendante des conditions d'environnement.
- détection d'objet de toutes formes et de matériaux de toutes natures.

III.4.4 Capteurs de position angulaire (codeurs

rotatifs) III.4.4.1 Codeurs optiques

III.4.4.1.1 Définition

L'évolution des systèmes de production conduit à un besoin croissant d'information de position, de vitesse, ou de déplacement des parties mobiles à tout instant du processus et non uniquement en fin de course.

Les détecteurs de position et de fin de course qui ne délivrent que des informations binaires (tout ou rien) ne peuvent répondre à ces besoins.

Les codeurs permettent de fournir à la partie commande électronique (automate programmable, carte de contrôle...) des informations de position ou déplacement précises.

Le codeur optique est lié mécaniquement a un arbre qui l'entraîne en rotation .Ce sont des informations sur la rotation de cet arbre que l'ont souhaité connaitre (vitesse ou valeur de la rotation, nouvelle position angulaire atteinte...)[15]

III.4.4.1.2 Principe de fonctionnement

Le codeur émet divers signaux électriques à destination de la partie commande chargée de les exploiter et de leur donner un sens (par décodage, comptage, mesure de la fréquence...) [15].

Tous les codeurs optiques exploitent des principes de fonctionnement similaires.

Ils sont constitués d'un disque comportant des zones opaques et des zones translucides. Le nombre de ces zones et leur disposition dépendent de la nature de codeur et du type d'information que l'on souhaite obtenir.

Des diodes électroluminescentes (LED) émettent une lumière qui peut traverser les zones transparentes. Des phototransistors, situés de l'autre côté du disque en regard des LED, Captent cette Lumière lorsqu' ils sont face à une ouverture et délivrent un signal électrique, image de la présence de cette ouverture[15].

III.4.4.1.3 Les types de codeurs optiques

Il existe deux principaux types de codeurs optiques[15] :

Les codeurs incrémentaux

Ils Délivrent une information de déplacement angulaire de disque sous forme d'un train d'impulsions.



Figure III.7 Codeur incrémental.

En général le disque rotatif comporte 3 pistes. Une ou deux pistes extérieures divisées en plusieurs intervalles égaux alternativement opaques et transparents. Un déphasage de 90° permet de déterminer le sens de rotation. Une piste intérieure comporte une seule fenêtre transparente et délivre un seul signal par tour. Ce signal détermine une position de référence et permet le ré initiation à chaque tour.

Chapitre III : Description de la machine

Le comptage ou le décomptage des impulsions, par l'unité de traitement (exemple automate) permet de connaître la position du mobile.

Le codeur incrémental est de conception simple (son disque ne comporte que deux pistes) donc plus fiable.

En cas d'absence prolongé du réseau (sans sauvegarde de l'unité de traitement) l'information de position peut être perdue.

En cas de modification de position (déplacement manuel hors tension du mobile), il y a perte de la position du mobile.

Le codeur absolu

Il a été créé pour pallier aux inconvénients du codeur incrémental. Le disque comporte plusieurs nombre de pistes (ou nombre de bits) et chaque piste a son propre système de lecture (diode émettrice et diode réceptrice). Pour chaque position angulaire de l'axe, le disque fournit un code binaire. Il existe 2 gammes de codeurs absolus :

- ✓ Le codeur absolu simple tour qui donne une position absolue dans chaque tour.
- ✓ Le codeur absolu multi-tours, donne une position absolue dans chaque tour et permet grâce à un système supplémentaire d'axes secondaire d'indiquer le nombre de tours.

Ce codeur est insensible aux coupures du réseau : la position du mobile est détenue dans un code qui est envoyé en parallèle au système de traitement.

L'information de position est donc disponible dès la mise sous tension.

Si le système de traitement une information de position délivrée par le codeur, la position réelle du mobile ne sera pas perdue car elle restera valide à la lecture suivante.

Il est de conception électrique et mécanique plus complexe aussi son coût est plus élevé qu'un codeur incrémental.

Les informations de position sont délivrées <<en parallèle>> ; son utilisation mobilisera donc un nombre important d'entrées du système de traitement (API par exemple).

III.5 Partie commande

La partie commande du système automatisé doit gérer le fonctionnement du système. Elle est chargée d'élaborer les consignes opératives en fonction de ses entrées et des règles de traitement (décrites par des équations logiques, un programme ou un Grafcet).

III.5.1 Pupitre de commande

Le pupitre est l'un des éléments permettant le dialogue entre l'opérateur et la partie commande afin de gérer l'ensemble des modes de marche et d'arrêt du système piloté.

L'opérateur envoie des consignes et reçoit des informations principalement visuelles. Il dispose pour cela de différentes formes de boutons (marche, arrêt, ..), sélecteurs et voyants[15].



Figure III.8 Pupitre de commande[15].

III.5.2 Contrôleur numérique du moteur de positionnement

Le contrôleur numérique du moteur de positionnement PMC-2 est un variateur de vitesse numérique à commande MLI sinusoïdale. Il est destiné à piloter le moteur sans balais (servomoteur) équipé d'un résolveur transmetteur. Cette version du PMC-2 équipé de sa carte option spécifique, permet de contrôler directement la position et la vitesse du moteur à partir d'un résolveur[15].

Le PMC-2 comprend[15] :

- commande de positionnement et de déroulement.
- régulateur de position.
- régulateur de vitesse.
- régulateur de couple.
- circuit intermédiaire à tension constante.
- onduleur triphasé en technique IGBT.

Ce type de contrôleur est utilisé dans toutes les applications qui requièrent un positionnement hautement dynamique, flexible et précis. Par exemple : machine d'industrie alimentaire, emballages et machines textiles...

Chapitre III : Description de la machine

Le contrôleur de moteur de positionnement PMC-2 communique avec [15] :

Une commande par un programme enregistré par transfert des données et sorties.

Une commande par un programme enregistré par les entrées et sorties.

Des appareils de contrôle et d'écran de visualisation.

Le PROFIBUS-DP.



Principe de fonctionnement du contrôleur PMC-2

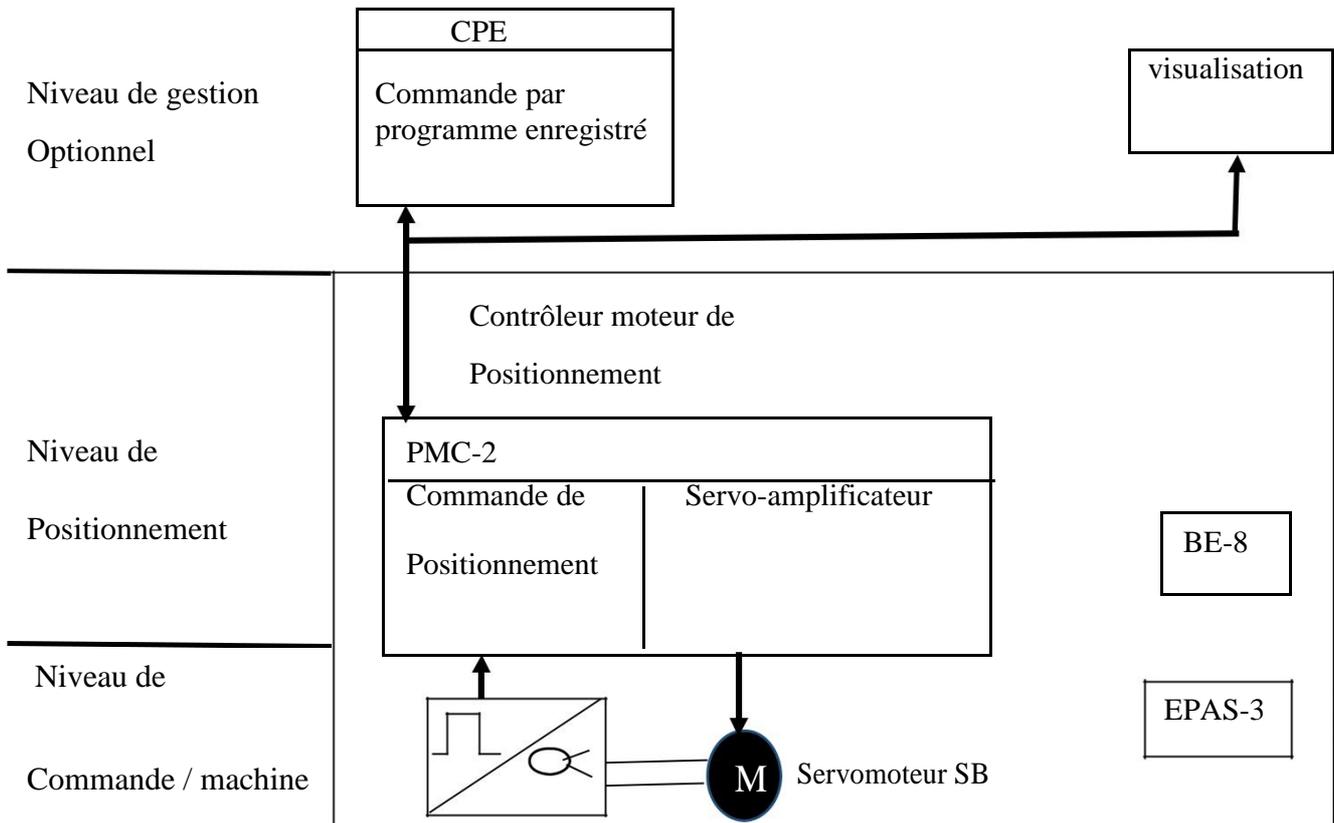


Figure III.9 Principale fonction du PMC-2[15].

Niveau de gestion

C'est un niveau où on trouve la commande par programme enregistré et la visualisation des applications et l'évènement durant le fonctionnement [15].

Niveau de positionnement

Ce niveau représente le contrôleur de positionnement lui-même, on trouve la commande de positionnement et les autres appareils de contrôle[15].

Niveau de commande / machine

Dans ce niveau on trouve l'actionneur (moteur) et l'ensemble des convertisseurs analogique / numérique qui reçoit le signal à partir d'un codeur angulaire[15].

Chapitre III : Description de la machine

De fait de la présence d'un filtre de réseau intégré, le courant de fuite de PMC -2 est supérieur à 3.5 mA ceci entraîne une incompatibilité des dispositifs protecteurs universels de courant de défaut. Cette protection est confirmée par la norme DIN VDE 0160.

L'Interface du contrôleur comprend[15] :

une interface de communication avec les consoles de programmation et d'autres contrôleurs.

une interface pour l'alimentation de moteur.

une interface d'alimentation de contrôleur par une tension alternative (220-240) et une alimentation par une source à courant continu de 24V.

carte mémoire de type EPROM.

carte d'entrées / sorties : utilisée pour recevoir des informations tout ou rien ou commander les divers actionneurs et pré actionneurs.

Le contrôleur possède les spécifications techniques suivantes [15] :

Tension de commande est de 24V utilisée pour l'alimentation des différentes parties logiques.

Tension d'alimentation du PMC-2 est 3~ 360 ... 240V avec une fréquence de 48 ... 62 Hz.

III.5.3 Variateur de vitesse

Le variateur de vitesse (FRS 500) est un convertisseur fréquence, utilisé pour l'alimentation de moteur asynchrone qui entraîne le tapis roulant. Grâce à une génération de transistor de puissance qui bénéficie des fonctions de commandes automatiques qui peut être ajouté le couple sur toute la plage de variation de fréquence pour un fonctionnement plus stable [15].



Figure III.10 Variateur de vitesse[15].

Chapitre III : Description de la machine

L'utilisation ultra simplifiée du FRS 500 le rend idéal pour toute l'application standard.

C'est le bon choix soit pour les tâches simples soit pour celles qui présentent plus de particularité comme[15] :

- Bandes transporteuses
- tapis
- portails
- Broyeurs à grains
- pompe
- moulins
- Convoyeurs à chaîne
- ventilation
- portes coulissants

Le variateur possède les caractéristiques suivantes :



Paramètres d'entrée :

Tension d'entrée 380 – 480 VcA triphasé (-15% +10%).

Fréquence 50 / 60 HZ.

Charge 0.9 à 9.5 KvA.



Paramètres de sortie :

Puissance 0.2 à 3.7 KW.

Courant nominal 1.4 à 7.7 A.

Protection / surcharge Surveillance du courant de sortie.

Tension de sortie triphasé, 0V – tension programmable.

Fréquence de sortie 0.5 à 120 Hz.

Méthode de contrôle scalaire (V / F).

Modulation MLI traditionnel.

Fréquence de découpage 0.7 à 14.5 kHz.

Le système comprend aussi :

Un transformateur de 250 VA 400 / 24 V

Un redresseur

Un fusible en triphasé pour chaque unité de commande, des contacteurs et des disjoncteurs, sélecteur porte fusibles, utilisés pour la protection de système de commande.

III.6 Partie pneumatique

Les systèmes automatisés qui mettent en œuvre des actionneurs pneumatiques sont nombreux dans les secteurs industriels automatisés. L'objet de cette série est de décrire les principaux types de vérins pneumatiques et les éléments de lignes pneumatiques que l'on peut rencontrer sur un système automatisé.

L'énergie pneumatique utilise l'air comprimé pour le transport de l'énergie et sa transformation en énergie mécanique [14].

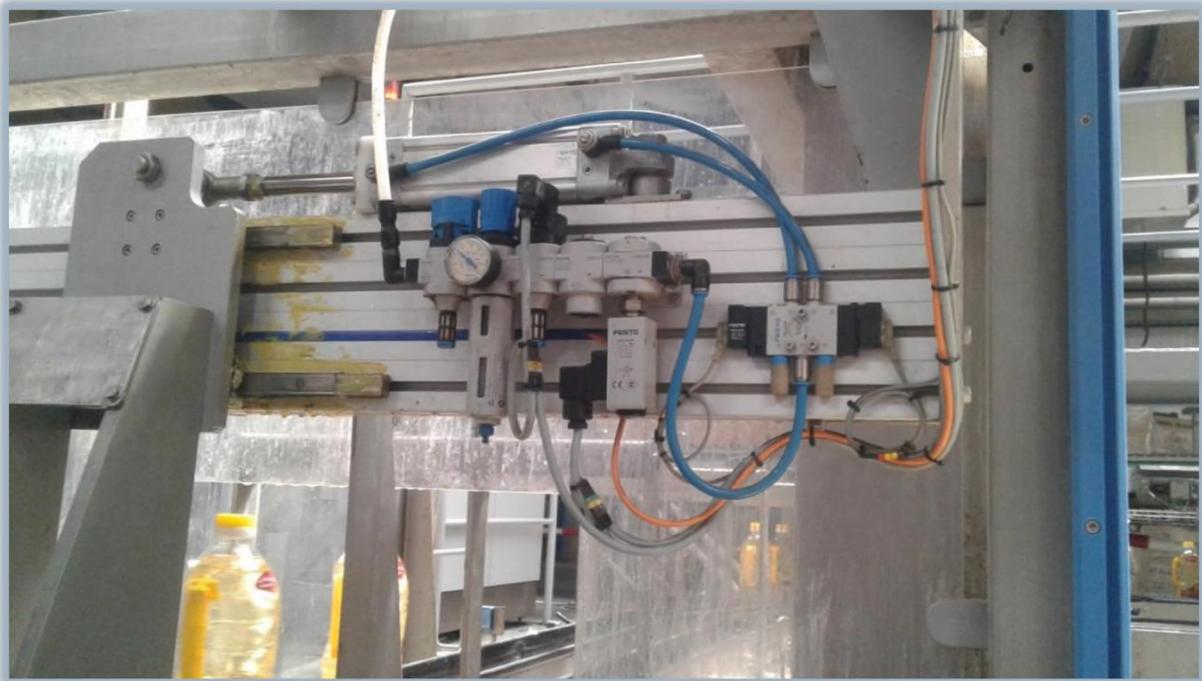


Figure III.11 La partie pneumatique du répartiteur de bouteilles.

Cette partie comporte[15] :

- Un vérin qui assure le mouvement de translation transversal ;
- un conditionneur composé d'un :
 - filtre pour débarrasser l'air de toutes les impuretés ;
 - manomètre qui nous indique la pression de l'air ;
 - détendeur qui garantit une pression de travail régulière ;
 - des limiteurs de débit unidirectionnel fixés sur les vérins ;
 - une vanne ;
 - un pressostat ;

Chapitre III : Description de la machine

- une source d'air ;
- un silencieux pour éviter les bruits d'échappement d'air ;
- des flexibles qui assurent le câblage ;

III.6.1 Distributeurs

Les distributeurs pneumatiques ont pour fonction de distribuer l'air comprimé jusqu'à l'actionneur (vérin). Ils ont le même rôle que les contacteurs[15].

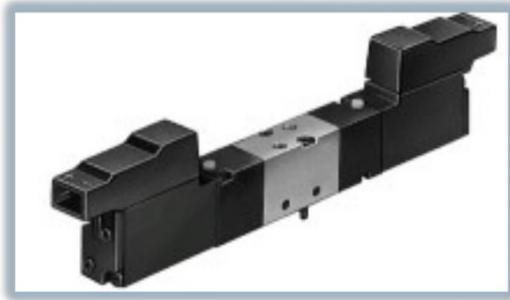


Figure III.12 Distributeur bistable [3].

Le distributeur possède deux pilotages de même nature, il est bistable (ou double pilotage).

Les deux positions sont des positions stables : en absence d'un signal de commande extérieur, le tiroir ne bouge pas et reste dans la position qu'il occupe[15].

Le schéma suivant montre les principales pièces qui constituent un distributeur.

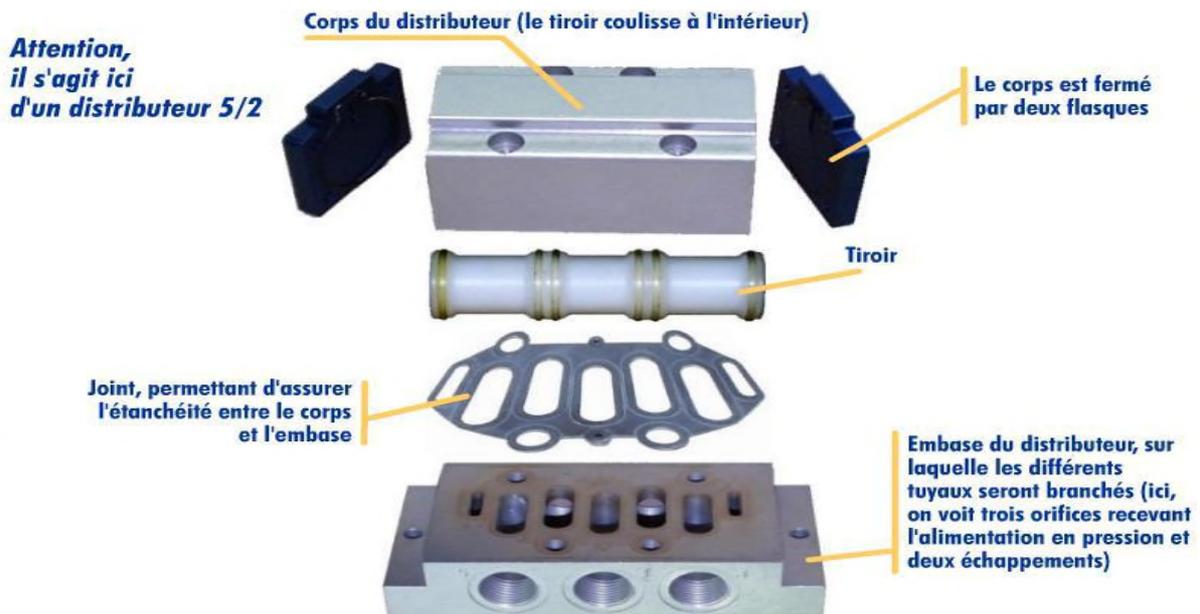


Figure III.13 Différentes parties d'un distributeur [3].

Chapitre III : Description de la machine

Le distributeur est caractérisé par[15] :

Le nombre des orifices : 2, 3,4 ou 5 ;

Le nombre de position : 2 ou 3 ;

Par la technologie de pilotage : pneumatique, électrique ou mécanique ;

Le type de commande de pilotage assurant le changement de position :

Simple pilotage avec rappel de ressort ;

Double pilotage ;

Dans notre cas on a un distributeur 5/2 bistable, à commande électrique (électrovanne EV).

III.7 Partie actionneurs

Les actionneurs Sont des convertisseurs électromécaniques conçus pour mettre en mouvement des systèmes mécaniques à partir de commandes électriques.

III.7.1 Moteur asynchrone

Un moteur de moyenne puissance est utilisé pour l'entraînement du tapis roulant. Il est construit par la firme SEW-USOCME [15].



Figure III.14 Moteur asynchrone triphasé.

Le moteur utilisé possède les caractéristiques suivantes[15] :

Tension : (230 / 240) V

Courant : (3.05 / 1.76) A

Fréquence nominale : 1360 tr/m

Puissance : 0.55 kW

Service : S1

III.7.2 Moteur synchrone

Le moteur effectue le mouvement transversal, est de série SB-105. C'est un servomoteur Brushless à aimant permanents, adapté à la régulation de vitesse et aux asservissements de position.

Il est optimisé pour fournir un couple élevé et des accélérations importantes grâce à la faible inertie de son rotor. Ses applications sont multiples et comprennent la robotique, les machines spéciales, la manutention...etc.

Ce moteur est utilisé pour entraîner le bras déviateur par l'intermédiaire des poulies, courroies et les guides sur rails. Est du type SB 105-40-04, il est constitué de frein, réducteur et un capteur de position angulaire utilisé pour l'autopilotage du moteur [15].



Figure III.15 Moteur synchrone (effectue le mouvement transversal).

Le moteur utilisé possède les caractéristiques suivantes [15] :

couple d'immobilisation du moteur : $M_{om} = 5.25 \text{ Nm}$

couple nominal du moteur : $M_{Nm} = 4.51 \text{ Nm}$

courant : $I_{sm} = 10 \text{ A}$

constant : $K_m = 1.18 \text{ Nm/A}$

vitesse de rotation : $N_n = 4000 \text{ tour/A}$

Chapitre III : Description de la machine

moment d'inertie (moteur avec frein et codeur) : $J_m = 3.4 \text{ kg.cm}$

tension d'alimentation : 230/400 V

la puissance du moteur : $P_{nm} = 1.89 \text{ KW}$

III.8 Conclusion

Ce chapitre est consacré à la description du processus et des principes de fonctionnements de sépareur. Cette partie nous a permis de familiariser avec le système et bien comprendre son fonctionnement, ainsi que le rôle de chaque constituant (capteurs, moteurs...etc.), afin de bien satisfaire au niveau de la réalisation.

Chapitre IV :
Programmation
et supervision
du
système propose

Chapitre IV : Programmation et supervision du système proposé

IV.1 Introduction

Pour commander la machine présentée dans les chapitres précédents, on doit élaborer un programme qui gère les différentes étapes de procès. Ainsi que la supervision de machine et ses différentes composantes.

Dans ce chapitre on va entamer la programmation et la supervision de notre processus. Il contient deux parties : partie programmation et partie supervision. On va détailler notre travail, en présentant les différents blocs de programme et les variables utilisées ainsi que les différentes vues qui ont été conçues pour la supervision.

IV.2 Problématique

La machine est commandée par une carte électronique spécifique à base de microprocesseur. Mais les performances de l'exploitation sont mises en question par les entreprises employant cette machine à cartes électroniques, conséquences de certains inconvénients s'illustrant comme suit :

- ✓ Elles sont très coûteuses.
- ✓ Elles ne sont pas réglables en cas de panne (grillage)
- ✓ La carte électronique est un ensemble de circuits intégrés formant un programme prédéterminé par le programmeur. Donc on ne peut pas la modifier.
- ✓ Elle n'offre pas la possibilité de visualisation des états des systèmes (E/S) à chaque instant. Donc la localisation des pannes ce fait de façon globale.

IV.3 Elaboration des GRAFCETS de déviateur

Avant d'élaborer les GRAFCETS il est important de définir le cahier des charges qui représente les exigences et les conditions de fonctionnement.

IV.3.1 Cahier des charges fonctionnelles

Le système commence à fonctionner lors d'un appuie sur le bouton de mise en marche Bmc, après ça le moteur du convoyeur tourne pour qu'il entraîne ce dernier sur lequel les bouteilles d'huile de 1.8L arrivent .Une nappe portée une coté denté et une autre lisse, gère la déviation et la séparation des bouteilles dans deux voies. Un capteur de position angulaire 12B6 capte la position de la nappe, lorsqu'il détecte le coté lisse ,30 bouteilles passent dans une déviation dérivé lui-même à deux sens et elles seront comptées par un capteur photo

Chapitre IV : Programmation et supervision du système proposé

cellulaires Cb0. Après le passage de ces bouteilles, l'électrovanne 1 se ferme par ordre d'un capteur noté 'elc2' pour orienter les 30 bouteilles suivantes compter par un capteur photo cellulaires Cb2 à l'autre sens .

Les 30 bouteilles suivantes vont passer dans le sens 2, et l'électrovanne 2 se ferme par ordre d'un capteur 'elc1'. Un moteur fait tourner la nappe vers le coté denté pour le passage de 15 bouteille vers l'autre déviation que seront comptés par Cb0.

Pour la sécurité le système s'arrête lorsque l'un de ces cause sera vraie:

- ✓ ouverture d'une porte parmi les quatre ;
- ✓ arrêt d'urgence;
- ✓ pressostat (seuil de pression)
- ✓ défaut de variateur
- ✓ défaut de machine avale ;
- ✓ défaut des bouchons ;
- ✓ défaut des bouteilles tombées;

le système se met en fonctionnement après le réarmement de défaut par le bouton poussoir 11S1.

Pour le défaut bouchon, lorsque la bouteille est détectée par le capteur Cb1, elle sera prête à lui mettre un bouchon ,ce dernier est détecté par un capteur CBh ,il a deux possibilités:

la présence du bouchon fait initialiser le compteur des bouteilles

l'absence du bouchon, et après un comptage de deux bouteilles sans bouchons successives ,le défaut sera affiché après 02 seconds. La réinitialisation du compteur se fait après l'écoulement de 200 ms après l'appuie sur le bouton de reset . Pour le défaut bouteille, c'est le même principe .

Chapitre IV : Programmation et supervision du système proposé

IV.3.2.1 Table des symboles des grafset

Symboles	Commentaires
8SP1	Capteur de la porte 1
8SP2	Capteur de la porte 2
8SP3	Capteur de la porte 3
8SP4	Capteur de la porte 4
AR	Arrête d'urgence
11B5	Pressostat
DV	Défaut variateur
DEMA	Défaut machine aval
DFBOC	Défaut bouchon
ALDBT	Défaut bouteille
11S1	Réarmement
INT	Initialisation
12B6	Capteur de position de la nappe
M.M	Marche moteur
C0,C1,C2,C3,C4	Des compteurs
Bmc	Bouton marche
Cb0 Cb2	des capteurs photo cellulaires
Cb1 Cb3	Des Capteurs de présence de bouteille
elc 1	Capteur d'électrovanne 1
elc 2	Capteur d'électrovanne 2
EL 1	Electrovanne 1
EL 2	Electrovanne 2
CBH	Capteur de bouchon
RS	Reset de compteur C2

Chapitre IV : Programmation et supervision du système proposé

CBT	Capteur bouteille
RS2	Rest de compteur C0
AREE	Arrête
MC	Marche convoyeur
DEF	Défaut
Ar	Bouton arrêt

Tableau 1 : Table des symboles des grafcet.

IV.3.3 GRAFCETS

IV.3.3.1 Grafkets des défauts

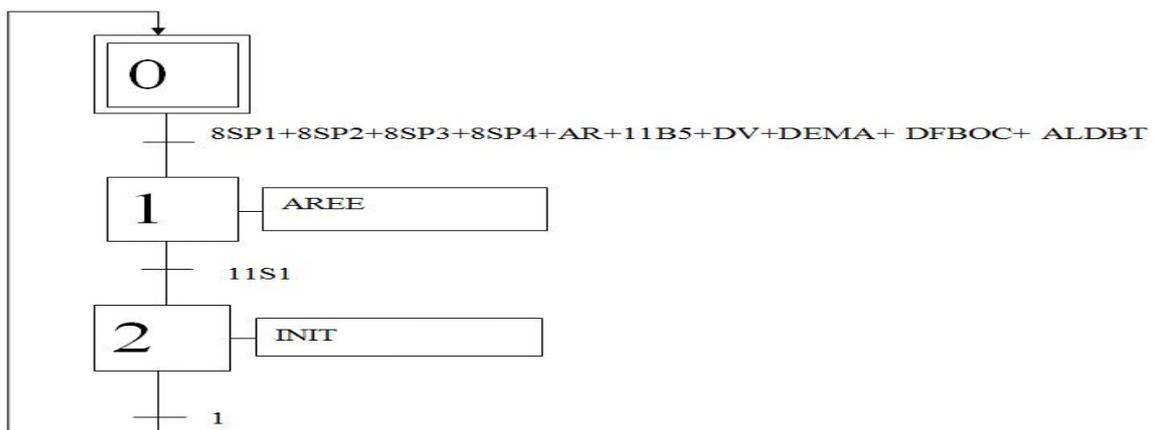


Figure IV.1 Grafkets de défaut.

IV.3.3.3 Grafkets de convoyeur

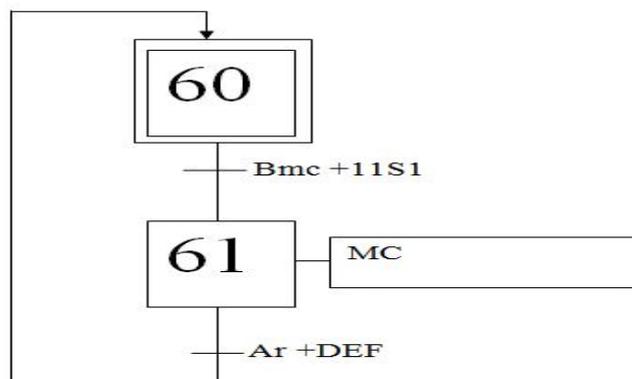


Figure IV.2 Grafkets de convoyeur.

IV.3.3.2 Grafquets de marche moteur

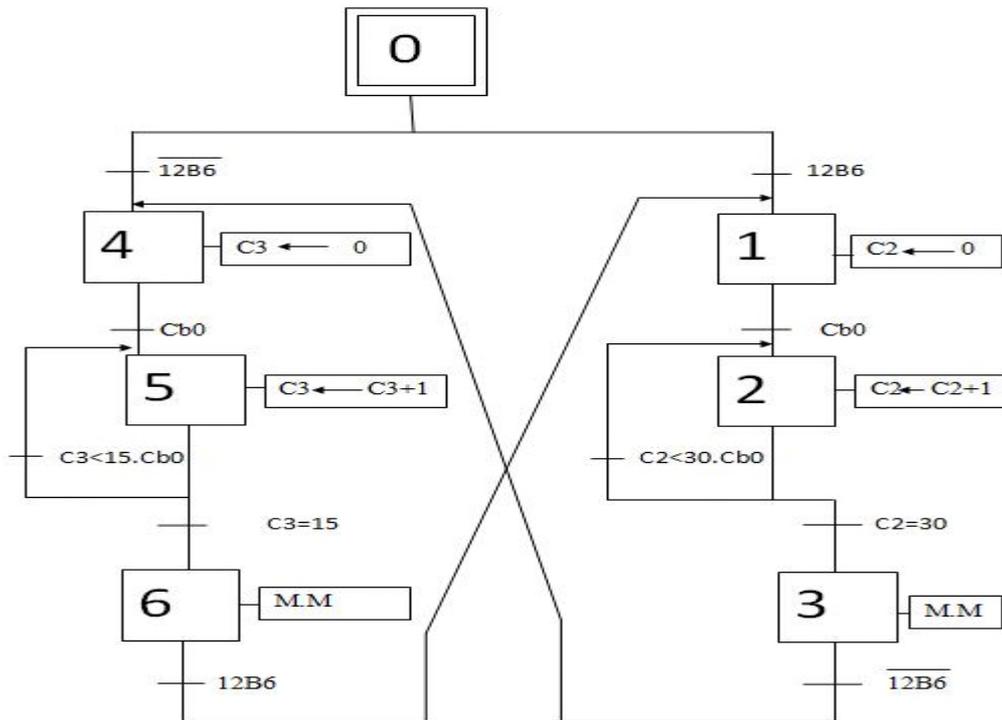


Figure IV.3 Grafquets de marche moteur.

IV.3.3.4 Grafquets des électrovannes

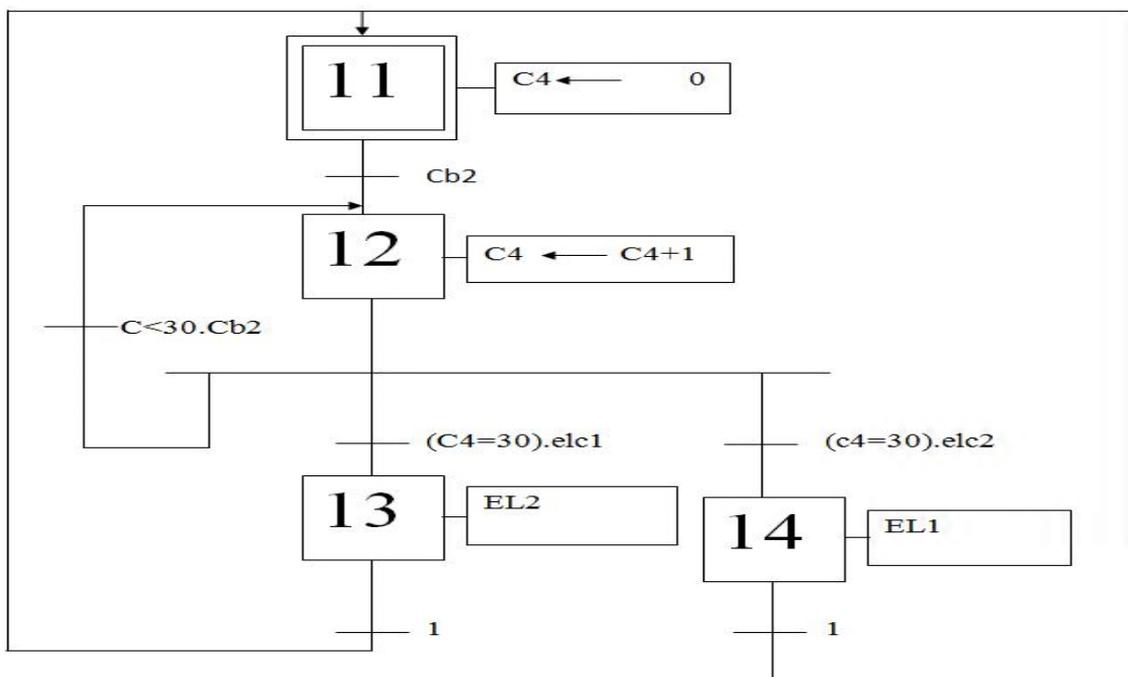


Figure IV.4 Grafquets des électrovannes.

IV.3.3.5 Grafquets de défaut bouchon

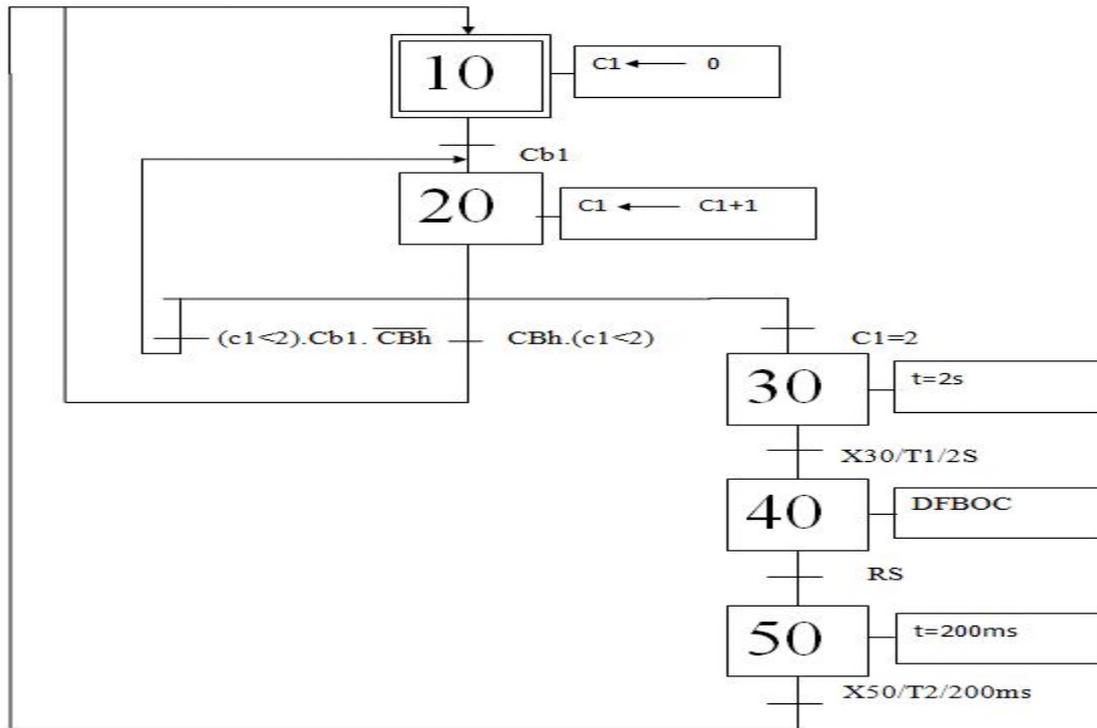


Figure IV.5 Grafquets de défaut bouchon.

IV.3.3.6 Grafquets de défaut bouteille

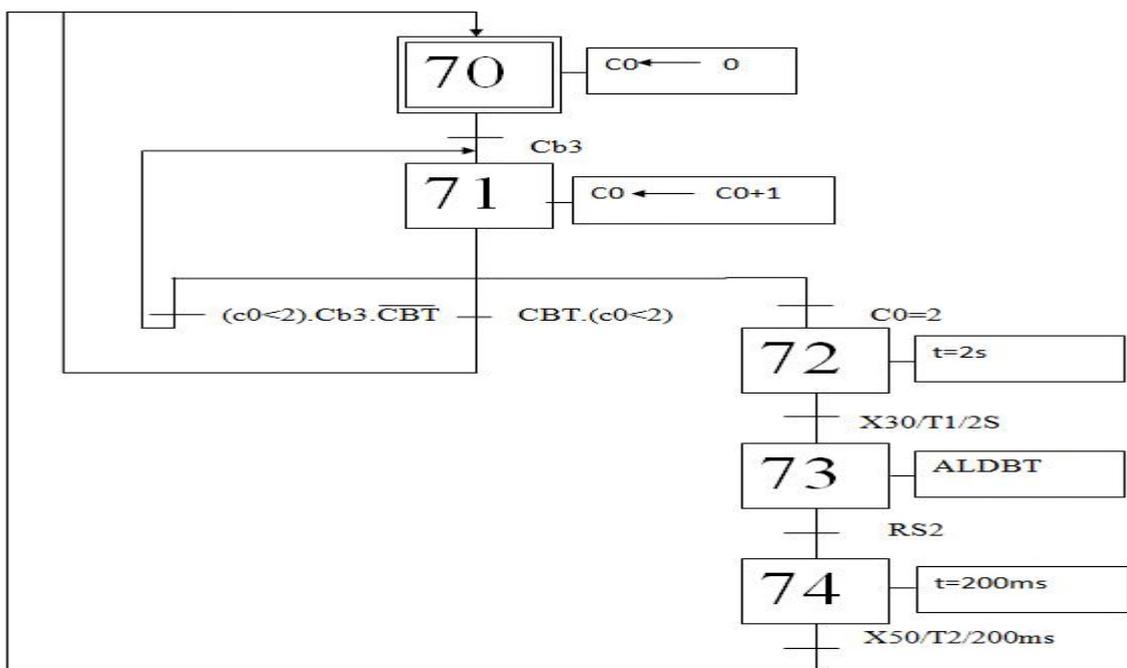


Figure IV.6 Grafquets de défaut bouteille.

IV.4 Programmation avec STEP7

IV.4.1 Prise en main de l'API

L'automate disponible au CEVITAL est un API siemens SIMATIC S7-300 (figure IV.7) dont les caractéristiques sont :

CPU 314 IFM;

6ES7 314-5AE03-0AB0;

mémoire de travail de 32 Ko;

0,3ms/kilo- instructions ; DI20/DO16

; AI4/AO1 intégrées ;

port MPI;

configuration à plusieurs rangées jusqu'à 31

modules; communication S7 (FB/FC chargeables);

firmware V1.2;



FigureIV.7 API SIMATIC S7-300. CPU 314 IFM.

Chapitre IV : Programmation et supervision du système proposé

IV.4.2 Manipulation de l'automate SIMATIC S7-300

L'automate a été livré avec le logiciel de programmation Step7 (version 5.5), accompagné du logiciel de supervision et de commande WINCC flexible.

IV.4.2.1 Création d'un projet

Un double clic sur l'icône SIMATIC Manager  sur le bureau ouvre la page d'accueil du logiciel Step7. Pour la création d'un projet, on clique sur fichier puis nouveau projet (figure IV.8) et on l'enregistre en lui donnant un nom.

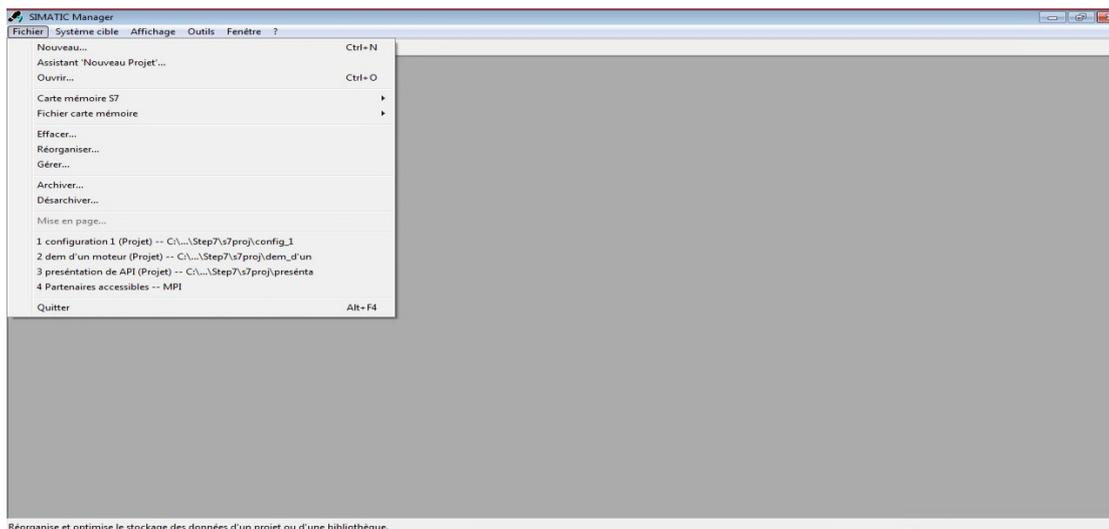


Figure IV.8 Création d'un nouveau projet.

On insère ensuite une station SIMATIC 300. On procède ainsi : clic droit de la souris → insérer un nouvel objet → station SIMATIC 300 (figure IV.9).

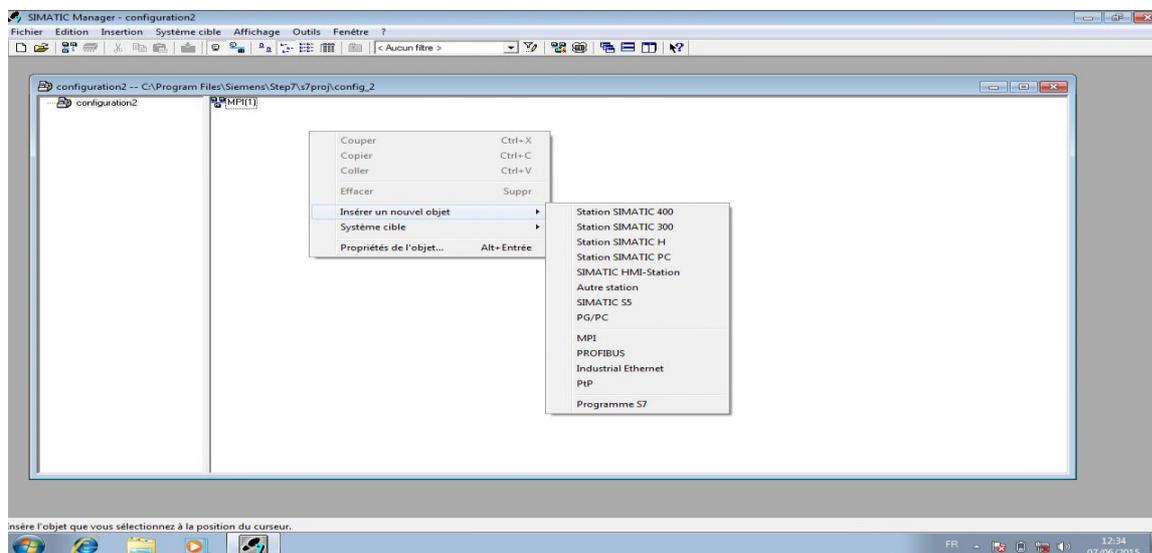


Figure IV.9 Insertion de station SIMATIC 300.

Chapitre IV : Programmation et supervision du système proposé

Après l'ajout de la station SIMATIC 300, on aura la fenêtre de la (figure IV.10) qui s'affiche.

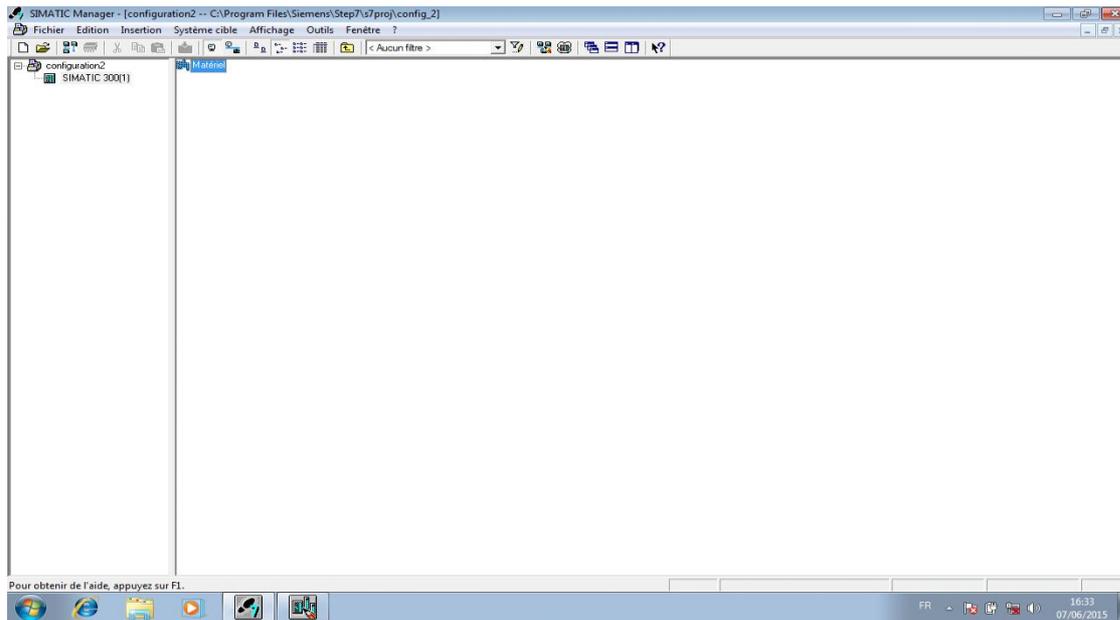


Figure IV.10 Fenêtre générée.

IV.4.2.2 Configuration de l'automate

Une fois le projet créé, on doit configurer l'API ; pour cela on double clique sur "Matériel" pour "Ouvrir" la fenêtre de configuration où le panneau de configuration de la station est vide pour le moment (Figure IV.11), et auquel on peut ajouter le matériel voulu sous forme de racks.

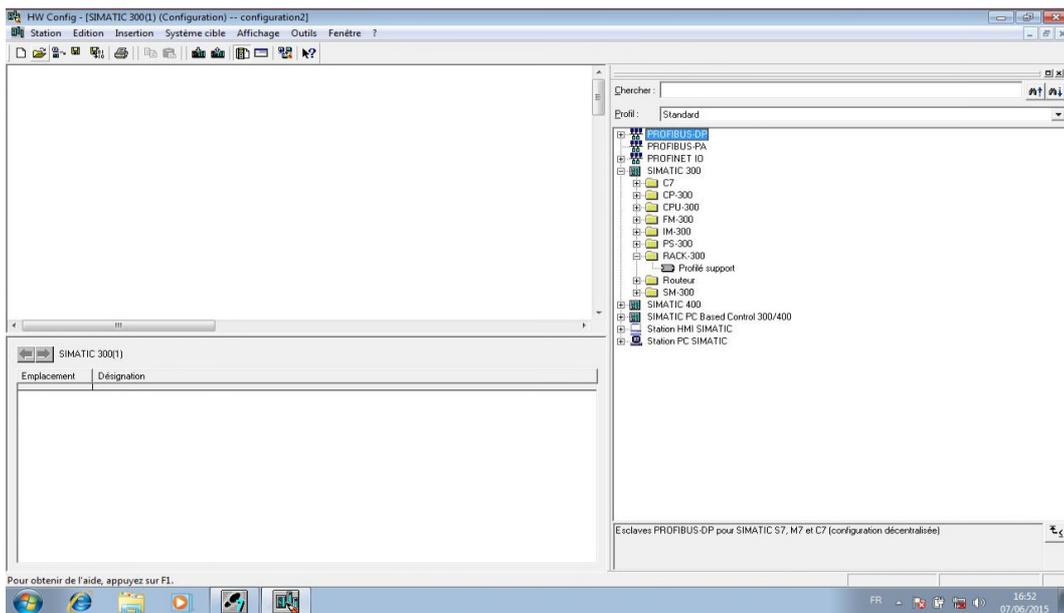


Figure IV.11 Panneau de configuration du matériel.

Chapitre IV : Programmation et supervision du système proposé

On va ajouter notre API à la station. Pour ajouter le rack correspondant à notre API, dans le panneau de configuration, dans la fenêtre catalogue matériel, on clique sur SIMATIC 300→RACK 300→Profilé support (figure IV.12).

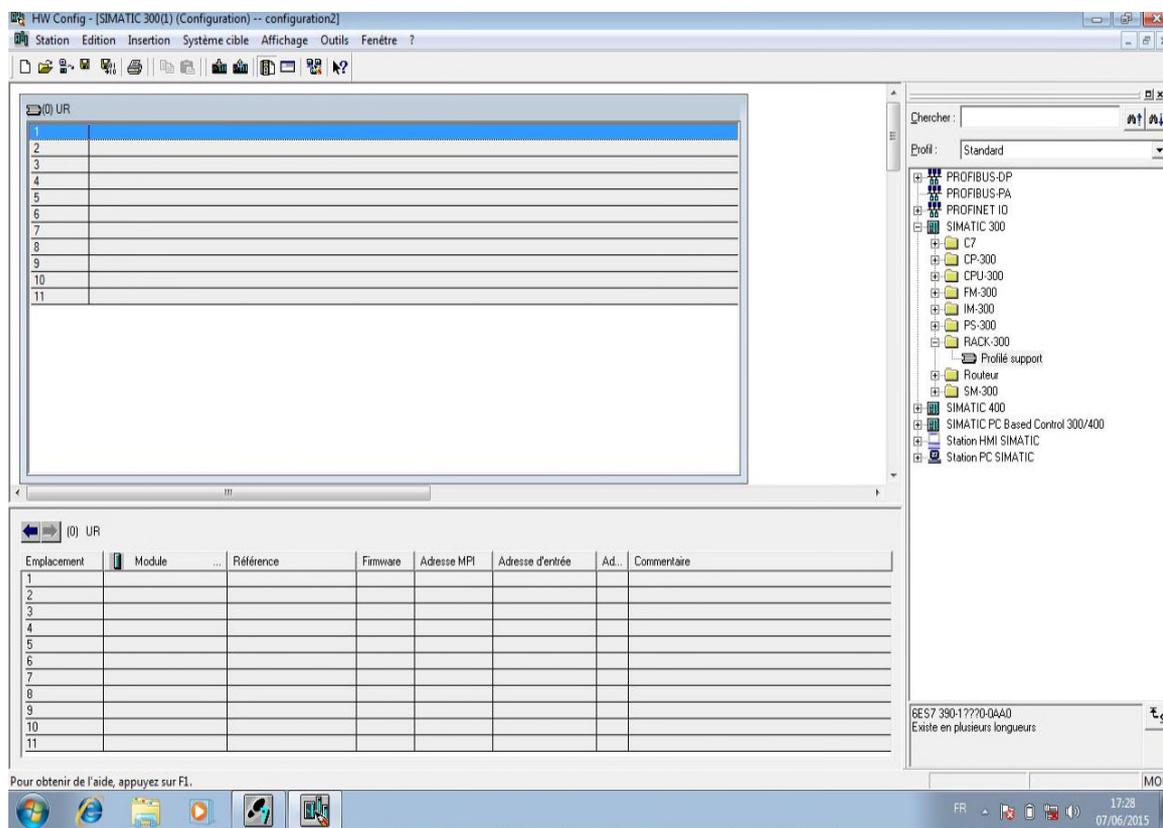


Figure IV.12 Affichage de profilé support.

Depuis le catalogue matériel on peut sélectionner les modules et les insérer dans le tableau de configuration. Ces modules sont effectivement connectés au rack réel. Pour cela il suffit juste de sélectionner la désignation du module et de la glisser dans une ligne du tableau de configuration, ou bien, clic droit dans une ligne du tableau de configuration puis objet (Figure IV.12), Là on va ajouter les modules voulu à leur place. Pour les premières lignes du tableau, on doit respecter l'ordre ; la ligne 1 est réservée à l'alimentation et la ligne 2 est réservée pour la CPU, la ligne 3 reste vide pour les modules de connexion.

Chapitre IV : Programmation et supervision du système proposé

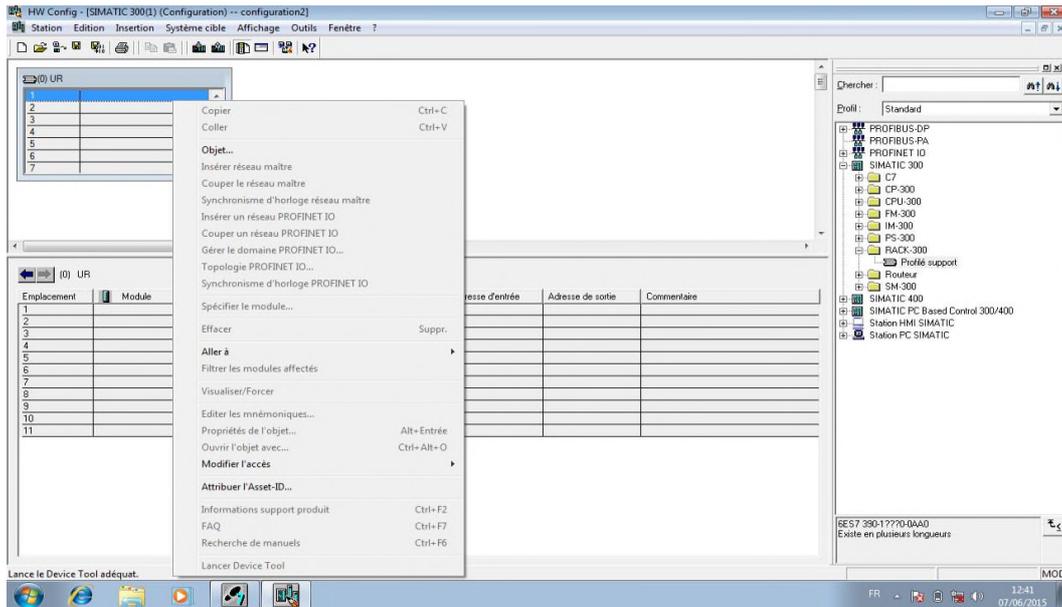


Figure IV.13 Insertion des modules dans le tableau de configuration.

Pour ajouter l'alimentation, clic droit dans la ligne 1 →objet→ SIMATIC 300→ PS-300→ PS 307 5A.

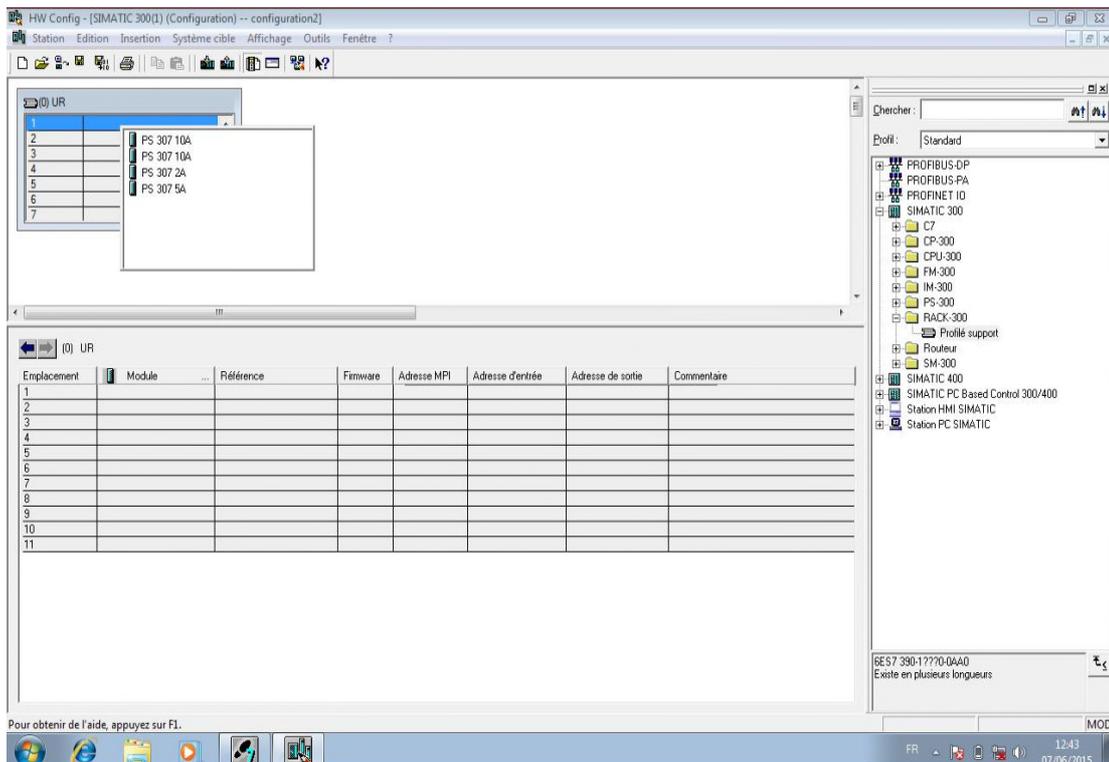


Figure IV.14 Choix de l'alimentation.

Chapitre IV : Programmation et supervision du système proposé

À l'étape suivante, ligne 2, on va ajouter le CPU 314 IFM. avec Clic droit sur deuxième emplacement → objet → CPU 314 IFM → 6ES7 314-5AE03-0AB0 → V1.2 (figure IV.15)

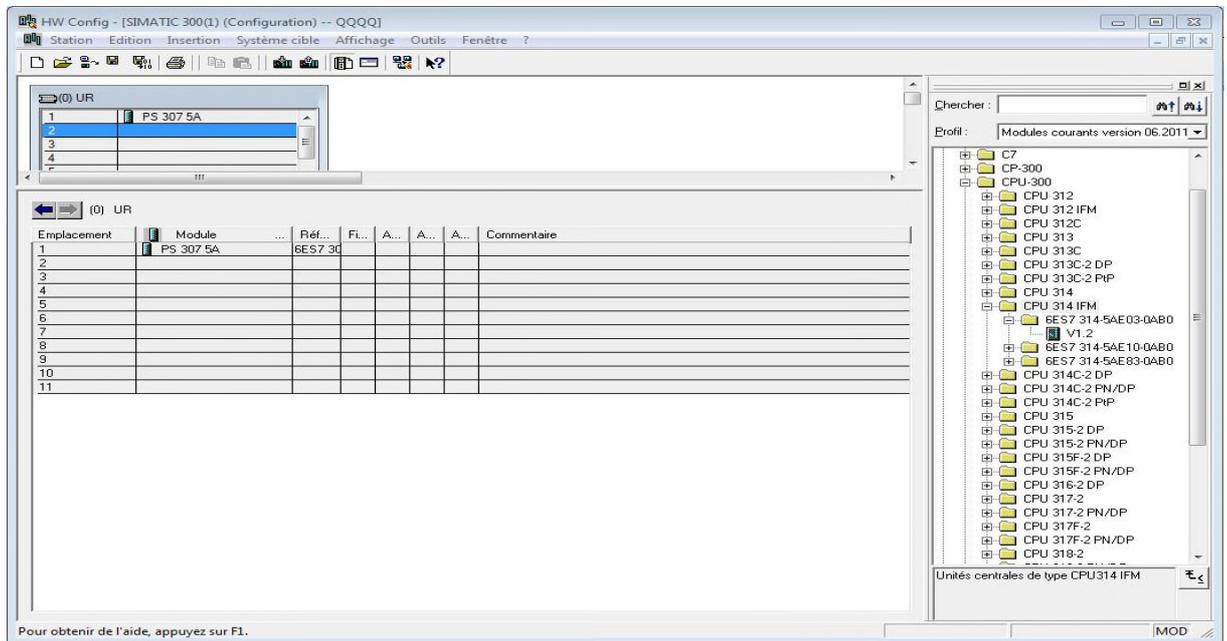


Figure IV.15 Choix du CPU.

Après avoir choisi le CPU, on aura le tableau de configuration suivant (figure IV.16).

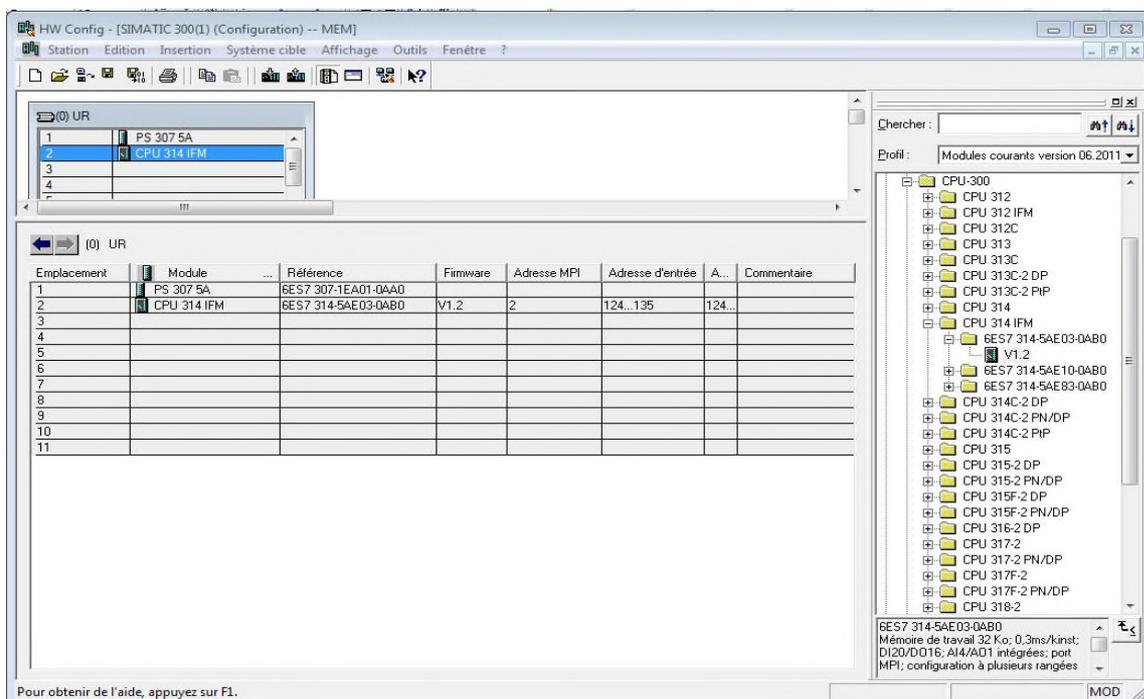


Figure IV.16 Configuration finale.

Chapitre IV : Programmation et supervision du système proposé

IV.4.2.3 Création de la table des mnémoniques

➤ **Mnémonique** : est un nom donné par l'utilisateur qui peut remplacer une variable, un type de donnée ou un bloc dans la programmation.

➤ **Table des mnémoniques** : Il s'agit d'une table qui permet d'affecter des mnémoniques (noms) à des adresses de données globales, accessible à partir de tous les blocs. Ils peuvent être en particulier des mementos (M), des entrées (E), des sorties (A), des temporisateurs, des compteurs ou des éléments de bloc de données (DB). Pour insérer la table des mnémoniques, on clique sur « Programme, Mnémonique » comme le présente la figure suivante:

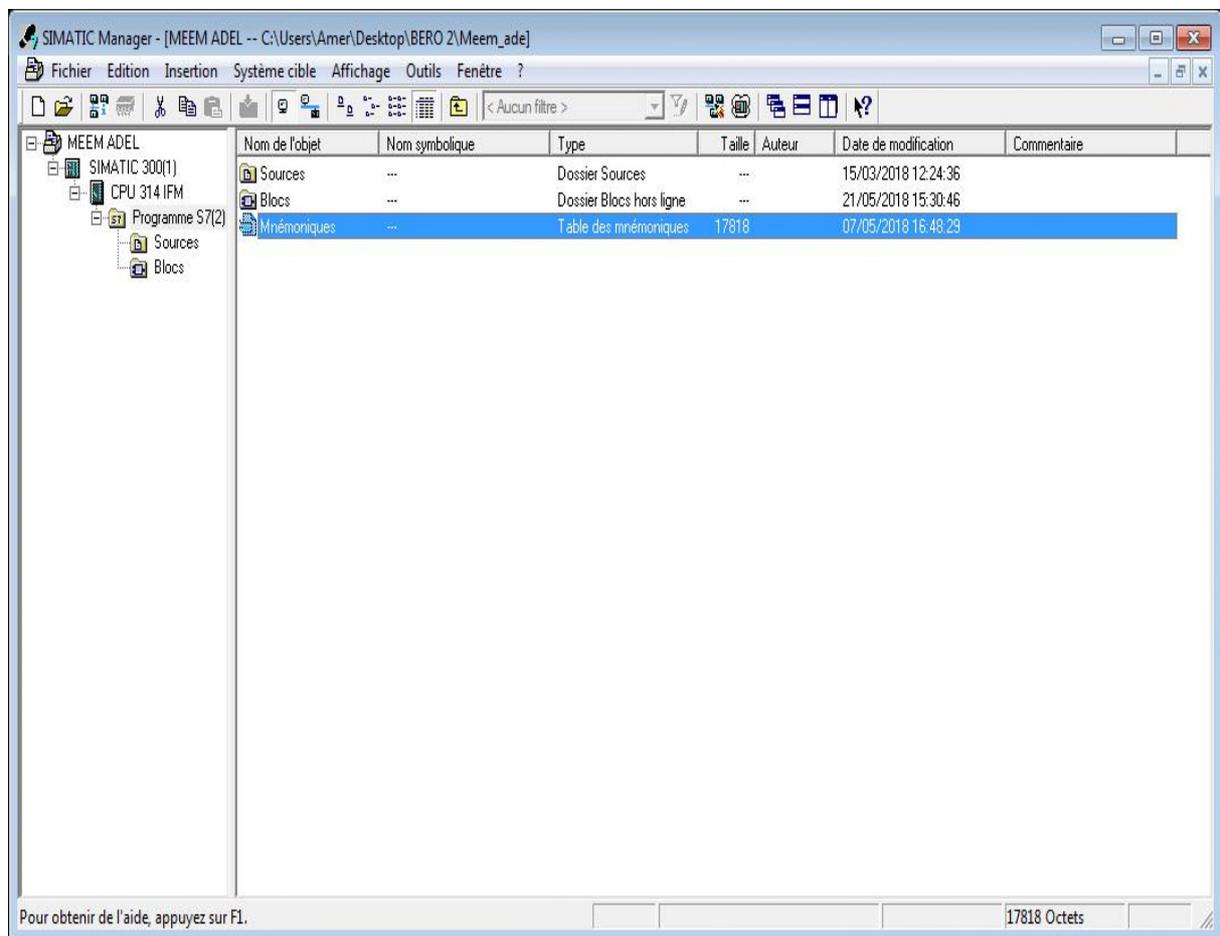


Figure IV.17 Fenêtre de création des mnémoniques.

Chapitre IV : Programmation et supervision du système proposé

IV.4.2.3.1 La table des mnémoniques

La fenêtre de la table des mnémoniques s'affiche comme suit :

	Etat	Mnémonique /	Opérande	Type de d	Commentaire
1		10B1	E 125.3	BOOL	CELLULE COMPTAGE
1		10KM1	A 124.7	BOOL	CONTACTOR
1		11B3	E 124.3	BOOL	CELLULE BOURRAGE SORTIE TRAIN
1		11B5	E 124.5	BOOL	PRESSOSTAT
1		11S1	E 124.0	BOOL	REARMEMENT DEFAULT
1		11S2	E 124.1	BOOL	MARCHE
1		11S3	E 124.2	BOOL	ARRET
1		12B6	E 124.6	BOOL	INDUCTIF PRISE D'ORIGINE
1		13KA1	A 124.0	BOOL	PRET APRES P.O.M
1		13KA2	A 124.1	BOOL	EN DEFAULT
2		14Y2	A 124.5	BOOL	ELECTROVANNE
2		14Y3	A 124.6	BOOL	ELECTROVANNE
2		8SP1	E 124.7	BOOL	PORTE 1
2		8SP2	E 125.0	BOOL	PORTE 2
2		8SP3	E 125.1	BOOL	PORTE 3
2		8SP4	E 125.2	BOOL	PORTE 4
2		AD	M 0.4	BOOL	ARRET D'ORGENCE
2		ALDB	M 10.1	BOOL	DEFAULT BOUCHONS
2		ALDBT	M 1.2	BOOL	DEFAULT BOUTIELLE F
2		AR	E 125.4	BOOL	ARRET D'ORGENCE
3		BLOC DEFAULT B...	FC 3	FC 3	
3		BLOC DE PROGR...	FC 4	FC 4	
3		BLOC DEFAULT B...	FC 2	FC 2	
3		BLOC DES DEFAU...	FC 1	FC 1	
3		C1	E 126.1	BOOL	CAPTEUR BOUTEILLKE
3		C2	E 126.6	BOOL	
3		CBH	E 126.2	BOOL	CAPTEUR DE BOUCHONS
3		CBT	E 126.4	BOOL	CAPTEUR DE BOUTIELLE
3		DEMA	E 125.6	BOOL	DEFAULT MACHINE AVALE
3		DFBOC	M 1.1	BOOL	DEFAULT SANS BOUCHONS
4		DFBOT	M 10.3	BOOL	DEFAULT BOUTIELLE
4		DFV	M 0.6	BOOL	DEFAULT VARIATEUR
4		DMA	M 1.0	BOOL	DEFAULT MACHINE AVAL
4		DV	E 125.5	BOOL	DEFAULT VARIATEUR
4		ELECTROVANNE	E 125.7	BOOL	"14Y3"
4		ELECTROVANNE 2	E 126.0	BOOL	"14Y2"
4		GV	M 10.2	BOOL	
4		M M	A 125.0	BOOL	MARCHE MOTEUR
4		P1	M 0.0	BOOL	PORTE1
4		P2	M 0.1	BOOL	PORTE2
5		P3	M 0.2	BOOL	PORTE3
5		P4	M 0.3	BOOL	PORTE4

Figure IV.18 Table des mnémoniques.

Chapitre IV : Programmation et supervision du système proposé

IV.4.2.4 Création d'un bloc de donnée 'DB'

Un bloc de données s'agit d'une zone de données dans le programme qui contient des données utilisateurs. Ces blocs de données globales peuvent être accés pour tout les bloc de code (OB, FC).

On clique sur le répertoire «bloc », puis avec un clic droit sur cette fenêtre, on choisit « Insérer un nouvel objet, Type de données » comme illustré dans la Figure IV.7 ci dessous :

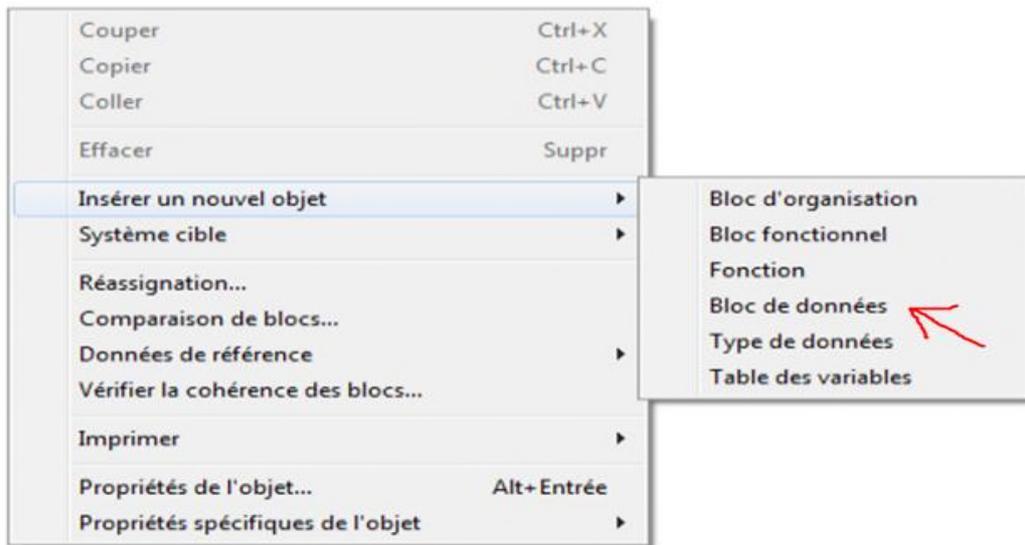


Figure IV.19 Fenêtre Création d'un bloc de donnée 'DB'.

La fenêtre suivante s'ouvre, on donne un nom à ce bloc « donnée de supervision ».

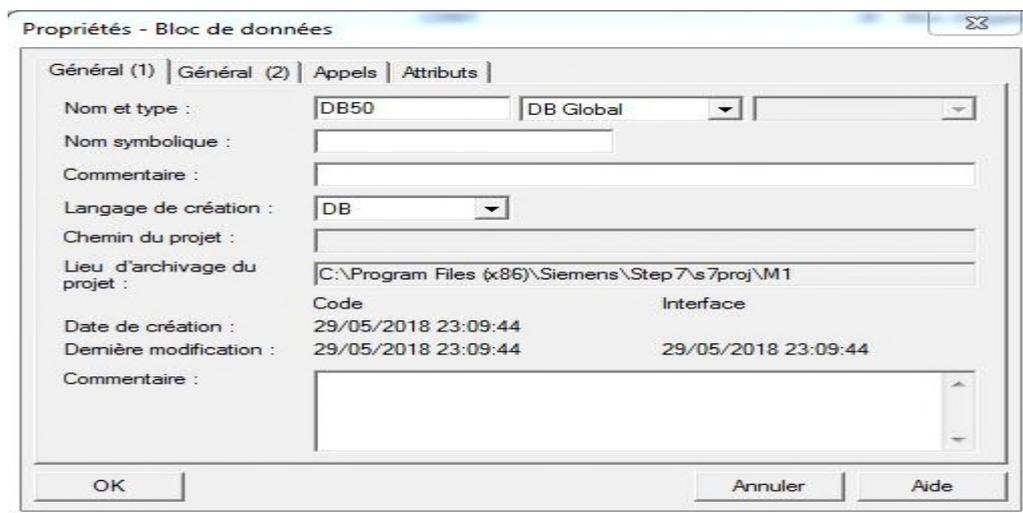


Figure IV.20 Fenêtre de nom de bloc de donnée.

Chapitre IV : Programmation et supervision du système proposé

IV.4.2.5 Création d'un bloc d'organisation 'OB'

Tout bloc doit être appelé avant de pouvoir être exécuté ; on désigne par hiérarchie d'appel, l'ordre, l'imbrication dans un bloc d'organisation. On clic sur le répertoire « blocs », puis avec un clic à droite sur cette fenêtre « Insérer un nouvel objet, Bloc d'organisation » qui est un bloc d'organisation pour le programme, dans lequel on fait appel aux différentes fonctions utilisées dans notre projet

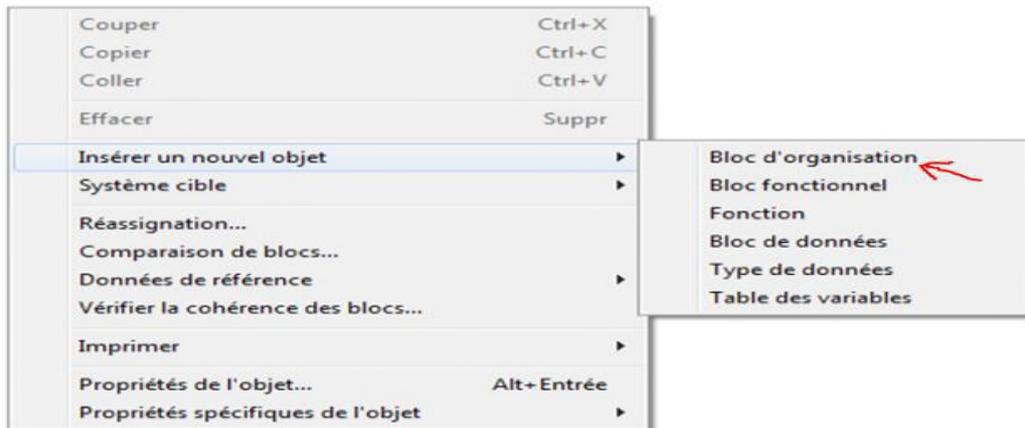


Figure IV.21 Fenêtre de Création d'un bloc d'organisation 'OB'.

IV.4.2.6 Création d'une fonction 'FC' :

C'est une fonction qui contient un programme qui est exécuté quand cette fonction est appelée par un autre bloc, On fait appel a la fonction pour :

- Envoyer une valeur de la fonction au bloc appelant.
- Exécuter une fonction.

On clique sur le répertoire « blocs », puis avec un clic à droite sur cette fenêtre On « Insérer un nouvel objet, Fonction ».

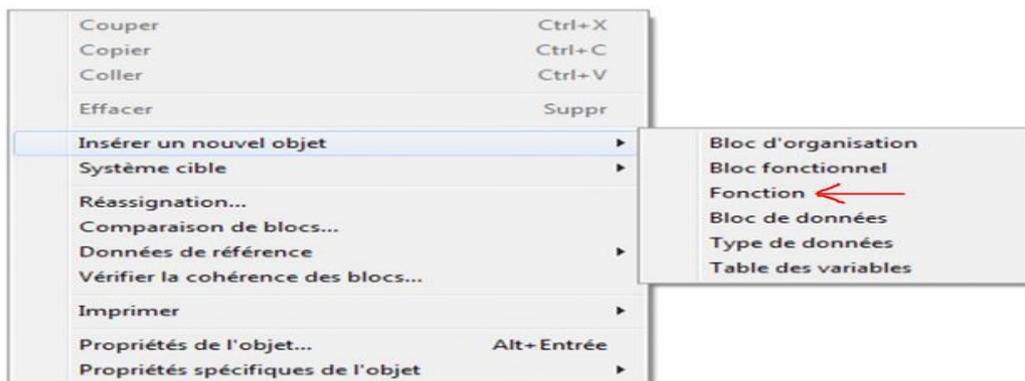


Figure IV.22 Fenêtre de Création d'une fonction 'FC'.

Chapitre IV : Programmation et supervision du système proposé

IV.4.2.7 Traitement du programme par la CPU

On distingue deux types de programmation:

Linéaire

Structurée.

IV.4.2.7.1 Programmation linéaire

La CPU exécute le cycle habituel, en appelant le bloc OB1 dans le programme principal ou les instructions s'exécutent les unes après les autres jusqu'à la fin. Ce type de traitement est utilisé pour des programmes simples.

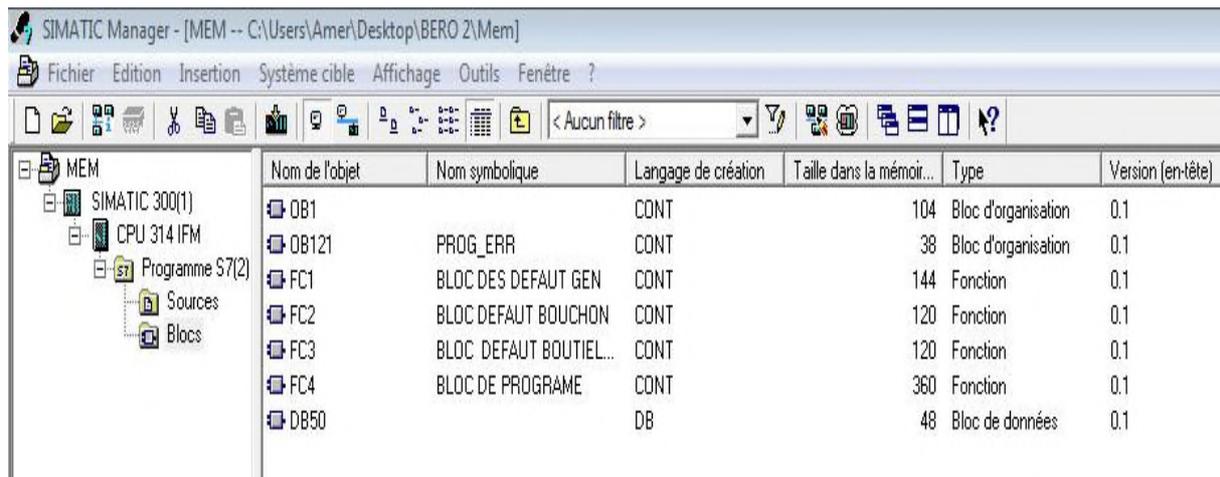
IV.4.2.7.2 Programmation structurée (hiérarchisée)

La programmation structurée consiste à subdiviser un programme complexe en sous-programmes pour exécuter des fonctions spécifiques plus petites et faciles. Le programme principal sera chargé de gérer ces sous-programmes et d'en faire appel autant de fois qu'il est nécessaire.

La programmation structurée sert à faciliter la maintenance et l'analyse fonctionnelle.

Dans le projet on a utilisé la programmation structurée qui contient les blocs suivants :

- OB1 : Le bloc principal.
- FC1 : La fonction des défauts
- FC2 : La fonction de défaut bouchon
- FC3 : La fonction de défaut bouteille
- FC4 : La fonction de système
- DB50: Données de supervision.



The screenshot shows the SIMATIC Manager interface. The main window displays a table of blocks. The table has the following columns: Nom de l'objet, Nom symbolique, Langage de création, Taille dans la mémoire..., Type, and Version (en-tête). The table lists the following blocks:

Nom de l'objet	Nom symbolique	Langage de création	Taille dans la mémoire...	Type	Version (en-tête)
OB1		CONT	104	Bloc d'organisation	0.1
OB121	PROG_ERR	CONT	38	Bloc d'organisation	0.1
FC1	BLOC DES DEF AUT GEN	CONT	144	Fonction	0.1
FC2	BLOC DEF AUT BOUCHON	CONT	120	Fonction	0.1
FC3	BLOC DEF AUT BOUTIEL...	CONT	120	Fonction	0.1
FC4	BLOC DE PROGRAMME	CONT	360	Fonction	0.1
DB50		DB	48	Bloc de données	0.1

Figure IV.23 Fenêtre des blocs.

IV.5 Supervision avec WINCC

Pour la simulation, la commande et la supervision des programmes STEP7 écrits pour notre machine automatique, nous avons réalisé une interface graphique (HMI) interactive à l'aide du logiciel WINCC flexible.

IV.5.1 Intégration de WINCC dans step7

Pour faire l'intégration de WINCC dans step7 en utilise la fenêtre de projet comme la montre la figure IV.24 suivante :

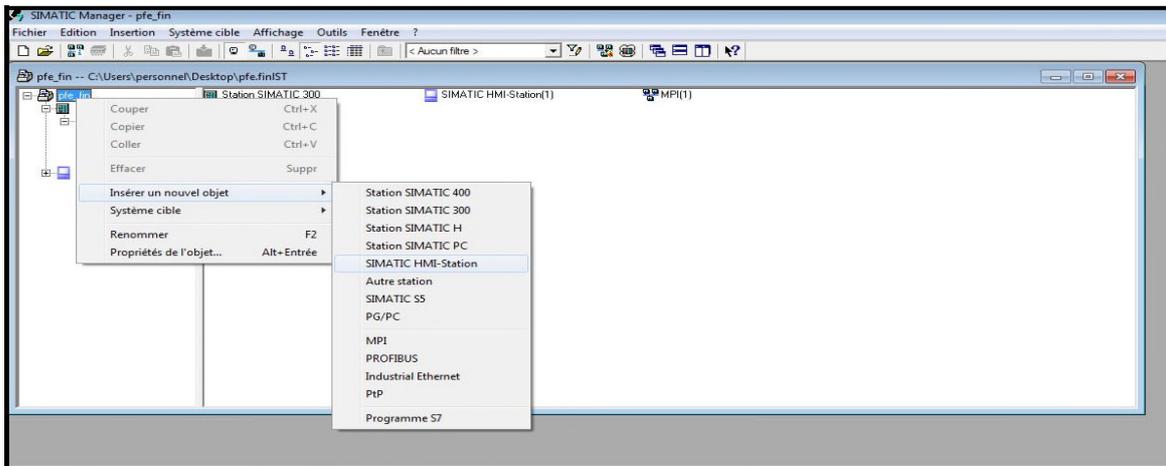


Figure IV.24 Intégration de WINCC dans step7.

IV.5.2 Liaison entre station et WINCC

L'activation de la liaison entre la SIMATIC 300 et l'interface HMI se fait à partir de WINCC :

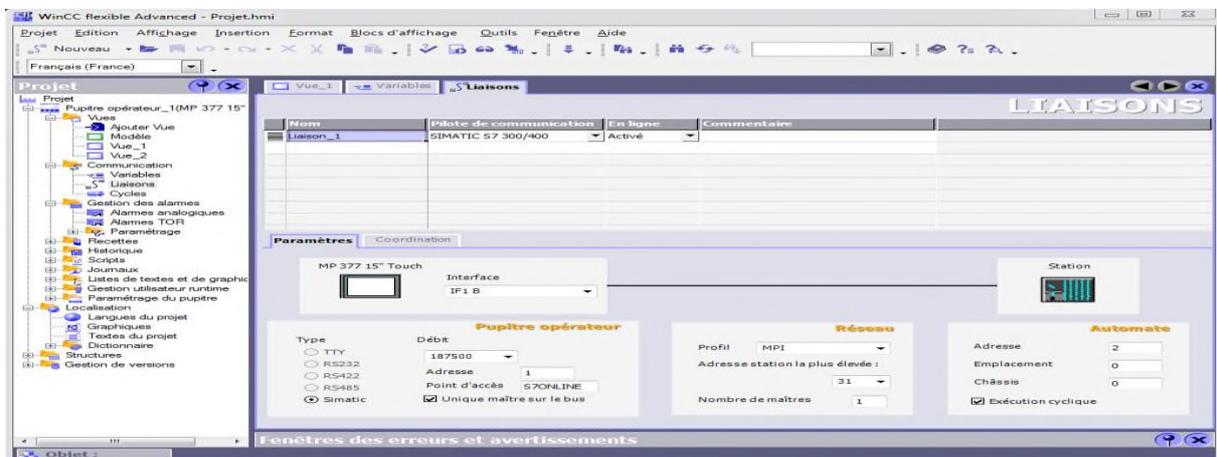
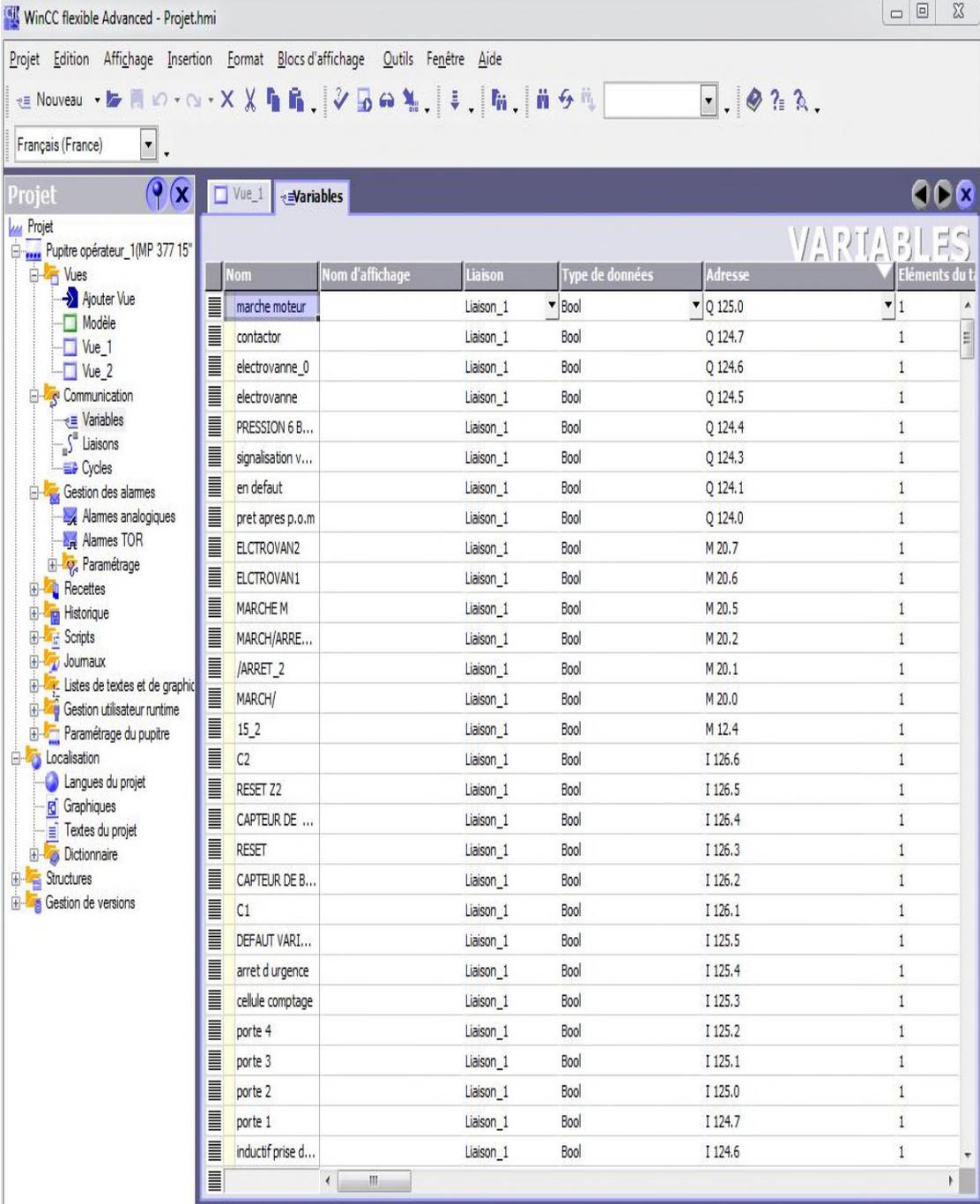


Figure IV.25 Intégration de WINCC dans step7.

Chapitre IV : Programmation et supervision du système proposé

IV.5.3 Table des variables utilisés dans HMI

Les variables déclarées dans la table des variables de WINCC sont en relation avec les variables déclarés dans la table mnémoniques dans Step7, pour créer une liaison entre les programmes et le pupitre de la commande (interface de commande).



Nom	Nom d'affichage	Liaison	Type de données	Adresse	Éléments du t
marche moteur		Liaison_1	Bool	Q 125.0	1
contactor		Liaison_1	Bool	Q 124.7	1
electrovanne_0		Liaison_1	Bool	Q 124.6	1
electrovanne		Liaison_1	Bool	Q 124.5	1
PRESSION 6 B...		Liaison_1	Bool	Q 124.4	1
signalisation v...		Liaison_1	Bool	Q 124.3	1
en default		Liaison_1	Bool	Q 124.1	1
pret apres p.o.m		Liaison_1	Bool	Q 124.0	1
ELCTROVAN2		Liaison_1	Bool	M 20.7	1
ELCTROVAN1		Liaison_1	Bool	M 20.6	1
MARCHE M		Liaison_1	Bool	M 20.5	1
MARCH/ARRE...		Liaison_1	Bool	M 20.2	1
/ARRET_2		Liaison_1	Bool	M 20.1	1
MARCH/		Liaison_1	Bool	M 20.0	1
15_2		Liaison_1	Bool	M 12.4	1
C2		Liaison_1	Bool	I 126.6	1
RESET Z2		Liaison_1	Bool	I 126.5	1
CAPTEUR DE ...		Liaison_1	Bool	I 126.4	1
RESET		Liaison_1	Bool	I 126.3	1
CAPTEUR DE B...		Liaison_1	Bool	I 126.2	1
C1		Liaison_1	Bool	I 126.1	1
DEFAULT VARI...		Liaison_1	Bool	I 125.5	1
arret d urgence		Liaison_1	Bool	I 125.4	1
cellule comptage		Liaison_1	Bool	I 125.3	1
porte 4		Liaison_1	Bool	I 125.2	1
porte 3		Liaison_1	Bool	I 125.1	1
porte 2		Liaison_1	Bool	I 125.0	1
porte 1		Liaison_1	Bool	I 124.7	1
inductif prise d...		Liaison_1	Bool	I 124.6	1

Figure IV.26 Tableau des variables.

IV.5.4 Présentation de l'interface HMI pour le système

IV.5.4.1 La fenêtre principale

La fenêtre principale a trois boutons « ALARMES », « VUE GLOBALE » et « MAINTENANCE ». C'est la fenêtre qui s'affiche au lancement elle est générale, Depuis cette vue l'opérateur peut accéder à la vue global ou à la vue maintenance ou vue alarmes

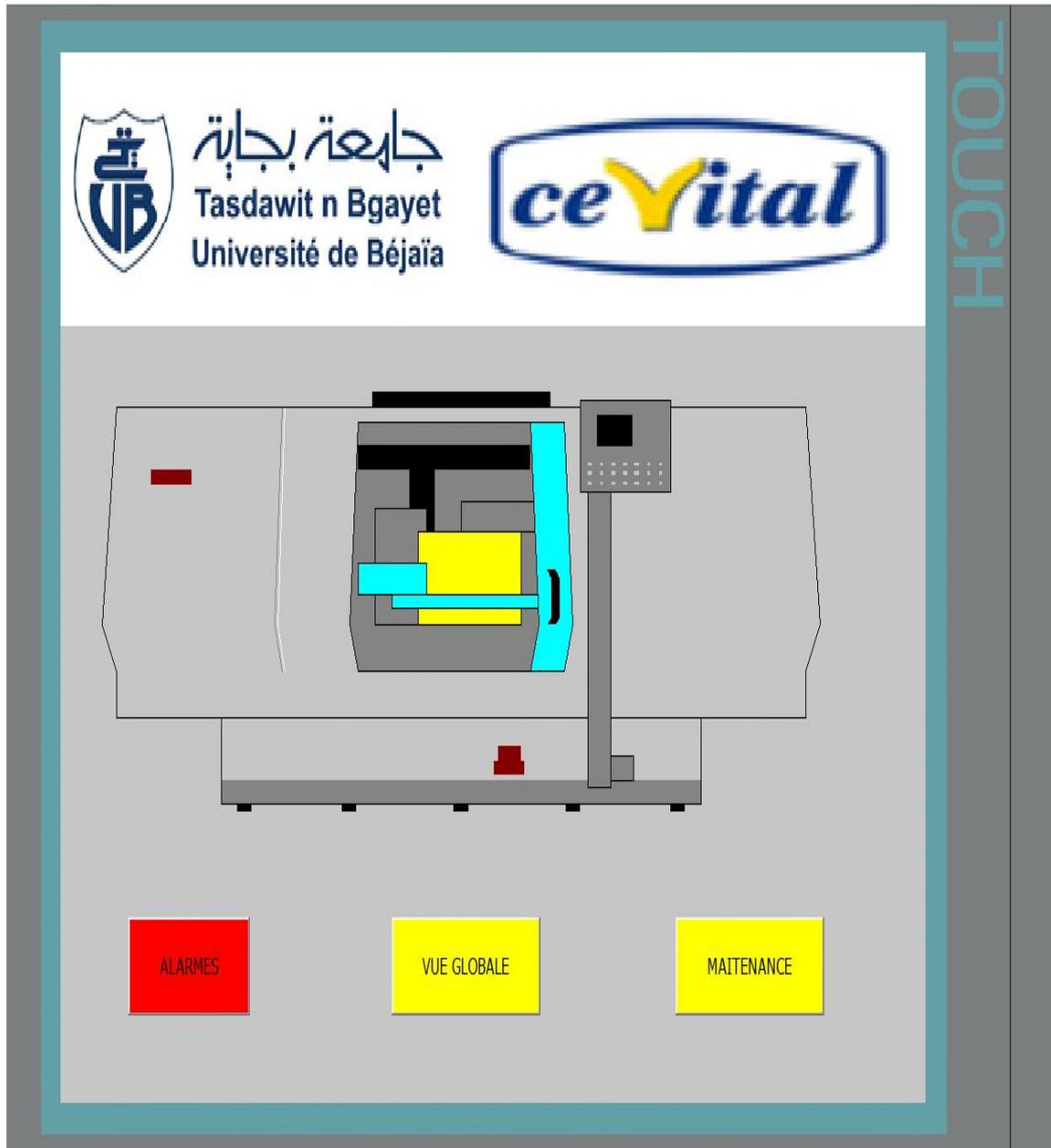


Figure IV.27 La fenêtre principale de l'application.

Chapitre IV : Programmation et supervision du système proposé

IV.5.4.2 Vue global

C'est une vue détaillé elle représente le système de fonctionnement de notre machine et depuis cette vue on peut supervisé et commander notre système .

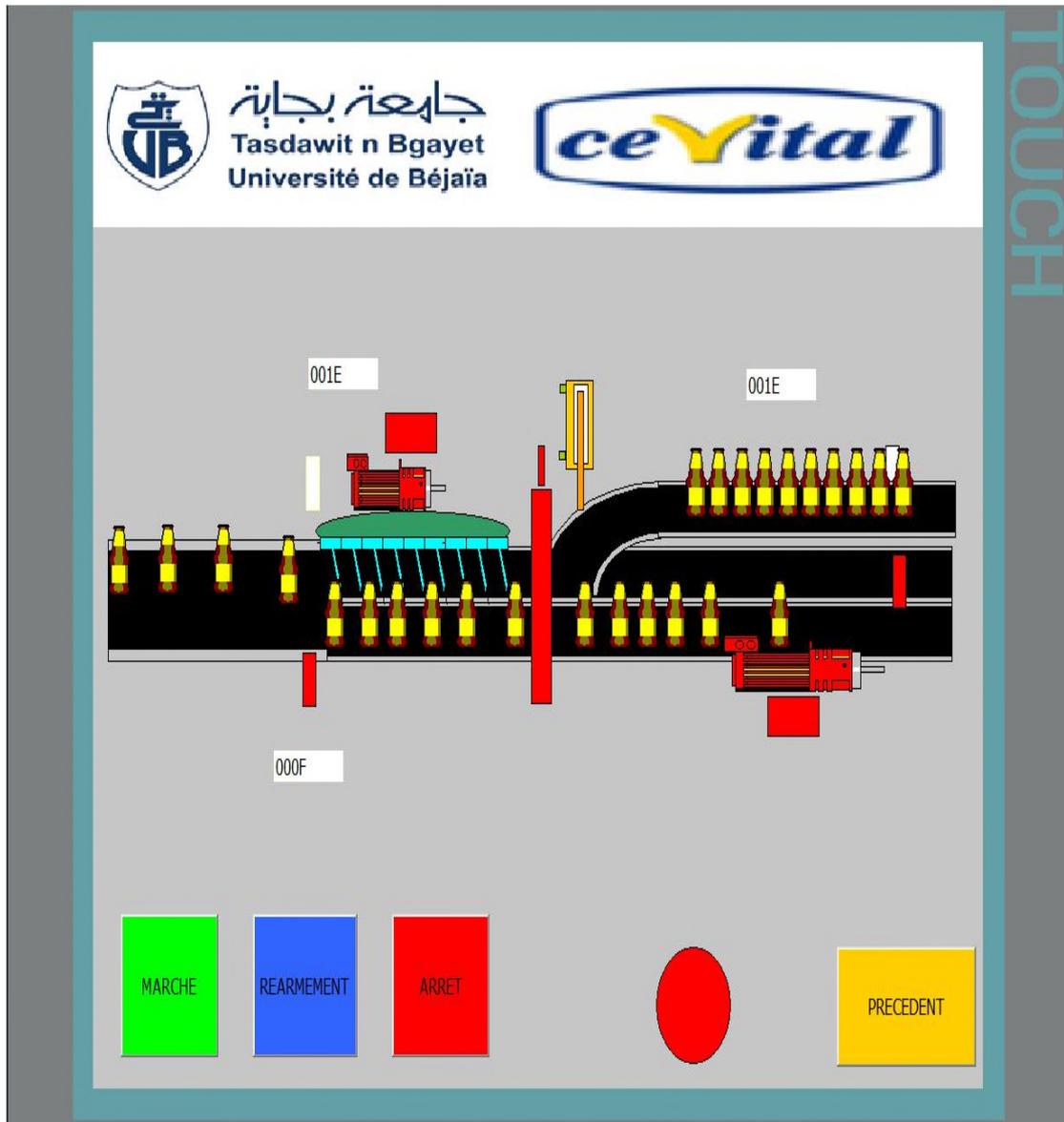


Figure IV.28 Vue Global.

IV.5.4.3 Vue maintenance

En cas de maintenance cet vue nous permet de tester le moteur et les électrovannes

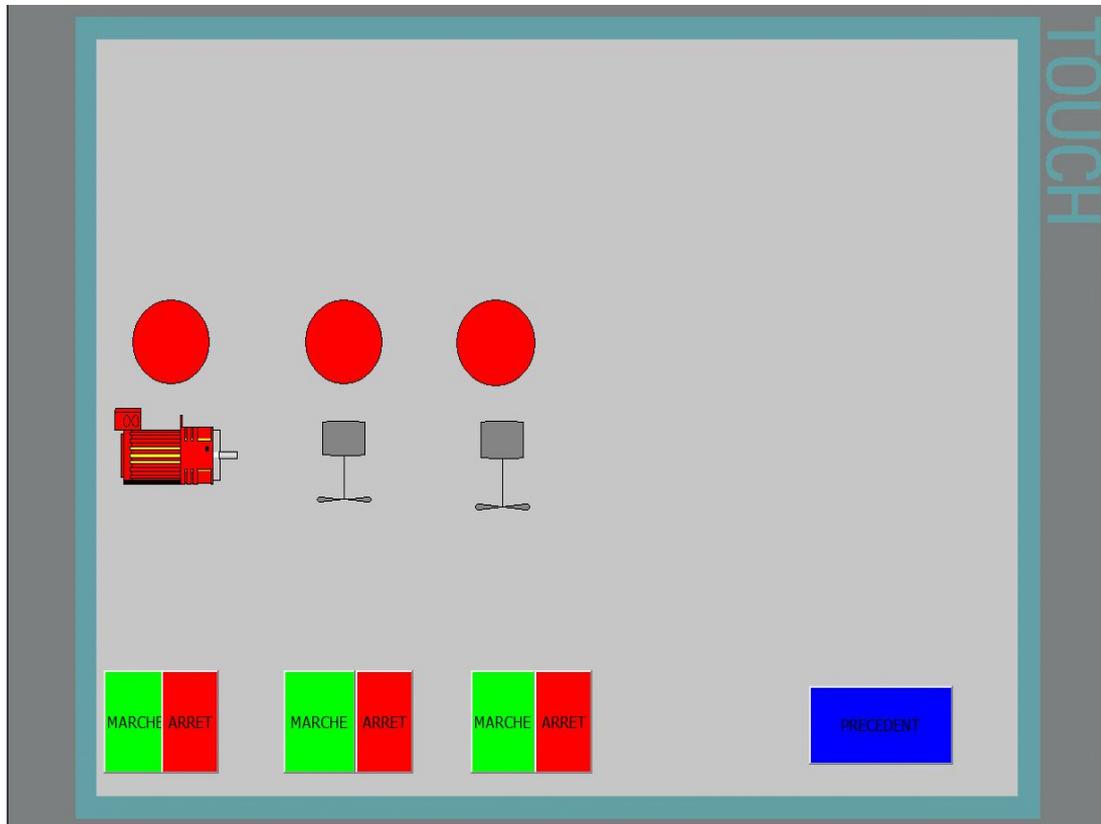
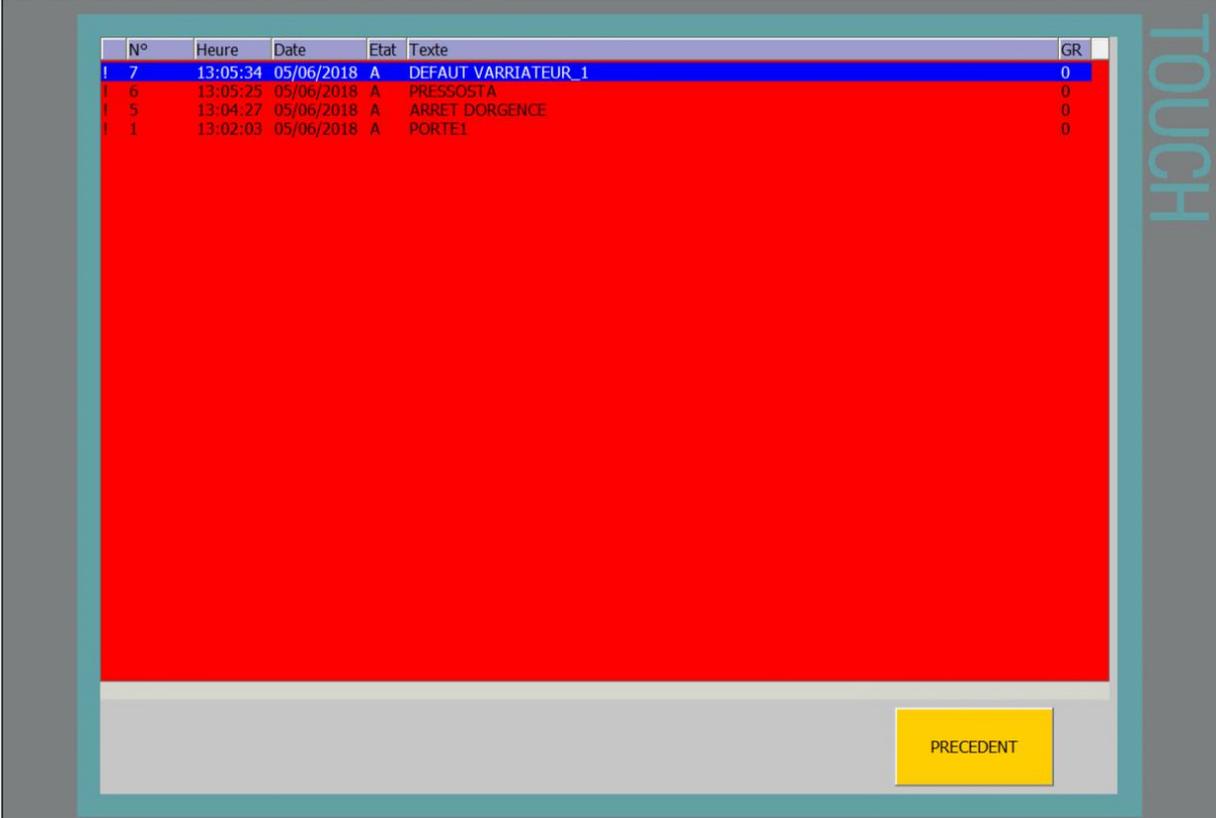


Figure IV.29 Vue maintenance.

IV.5.4.4 Vue alarmes et les mesures de sécurité :

Pour assurer le bon fonctionnement de la machine on a pensé aux mesures de sécurité suivantes :

1. arrêt d'urgence : en cas de problème de fonctionnement ou un risque pour l'utilisateur, on clique sur le bouton d'arrêt d'urgence ce qui va désactiver toutes les étapes.
2. les portes : en cas ou les portes sont ouvertes
3. le défaut variateur
4. le défaut machine avale
5. le défaut bouchon
6. le défaut bouteille



N°	Heure	Date	Etat	Texte	GR
7	13:05:34	05/06/2018	A	DEFAULT VARRIATEUR_1	0
6	13:05:25	05/06/2018	A	PRESSOSTA	0
5	13:04:27	05/06/2018	A	ARRET DORGENCE	0
1	13:02:03	05/06/2018	A	PORTE1	0

Figure IV.30 Vue alarmes.

IV.6 Conclusion

Dans ce chapitre, nous avons décrit les étapes essentielles en vue de la réalisation d'une solution d'automatisation pour un processus industriel avec les logiciels : STEP7 et WinCC-flexible, et de répondre à la problématique qui a été posée depuis le début de notre stage au sein de l'unité conditionnement d'huile du complexe Cevital.

Conclusion générale

Conclusion générale

La facilité d'utilisation et la souplesse qu'offre l'A.P.I pour sa programmation, sa connexion et son adaptation aux conditions industrielles, avec toutes les fonctionnalités indispensables à l'automatisation des processus, la diversité des possibilités mise en œuvre et son coût, le rend incontournable lors de l'élaboration d'une solution d'automatisme.

L'objectif de notre travail est la conception d'un système automatisé d'un déviateur des bouteilles d'huile. Pour cela nous avons étudié la machine et son principe de fonctionnement actuel. Ensuite on a décrit un cahier des charges fonctionnel qui a abouti en premier temps à décrire les processus avec le GRAFCET.

Mais il ne faut pas oublier qu'il est nécessaire d'avoir une bonne analyse du problème à résoudre tout en assurant le respect des règles d'installation.

A l'aide du logiciel performant STEP7 les automates programmables de la série Siemens SIMATIC S7-300 forment des unités de traitement et de commande d'une grande flexibilité. En effet, simple à utiliser et doté d'une interface graphique très intuitive, le logiciel de programmation step7 permet d'exploiter de manière optimale les différents CPU de la gamme.

WinCC est un logiciel performant qui permet la supervision des processus ainsi que l'exploitation des données acquises à partir de ces derniers. Aussi, il nous offre la possibilité d'intégrer nos propres programmes dans le but de piloter les processus.

Bibliographie

Bibliographie

- [1] Site officiel de CEVITAL « www.cevital.com ».
- [2] Philippe GRARE et Imed KACEM, « AUTOMATISME, CE QU'IL FAUT SAVOIR SUR LES AUTOMATISMES », Livre, Paris : Ellipse, 2008
- [3] Thierry shanen, <<GUIDE DES AUTOMATISME>>. [www.guide des automatismes.com](http://www.guide-des-automatismes.com), 2001/2007.
- [4] makhlouf tahar et mebaraki salah << PILOTAGE D'UN TOUR ETD'UNE FRAISEUSE AU TRAVERS D'UN API>> Projet de Fin d'Etude d'Ingénieur d'Etat en Génie Mécanique 2014 / 2015.
- [5] Ourari Hicham et Hachemi Khalid << DEVELOPPEMENT D'INTERFACES DE SUPERVISION : POUR LA COMMANDE DE DEUX PROCESSUS INDUSTRIELS PAR UN API SIEMENS S7-300>> Mémoire de fin d'étude institut de maintenance et sécurité industrielle 2013/2014
- [6] Kali Rahim et Kasmi Bachir, « AUTOMATISATION D'UNE CHAUDIERE LOOS AVEC UN AUTOMATE PROGRAMMABLE S7-300 A L'UNITE CHAUFFERIE AU COMPLEXE CEVITAL », Mémoire de fin d'études Master, Département de Génie Electrique, Université de Bejaia, 2016.
- [7] grafcet avec PGF /TIKZ Papanicola Robert. 11 mai 2011.
- [8] automatisation industriels Dr. SAADOUNE Achour Université Mohamed Khider – Biskra.
- [9] SIEMENS, « STEP 7 V 5.5, Getting started », SIMATIC Manager.
- [10] M.BERTRAND, « Automates programmables industriels », Technique de l'ingénieur, Vol. S 8 015
- [11] Jean Hég, « PRATIQUE DE LA MAINTENANCE PREVENTIVE », Livre, Paris :L'Usine nouvelle, 2011.

Bibliographie

[12] BELKACEM Hamza et RAIS Abdel basset, « SYSTEME DE CONTROLE DISTRIBUE (DCS) AVEC L'EXPLOITATION DE L'AUTOMATE PROGRAMMABLEAC800 F (ABB) », Mémoire de fin d'études Master Automatique, Université Mohamed Khider de Biskra, 2012.

[13]LES AUTOMATES PROGRAMMABLES INDUSTRIELS, Alain GONZOGA

[14] a/ halim degumoune, h/ ait aissa <<ETUDE ET CONCEPTION D'UN SYSTEME DE COMMANDE DE BASE D'UN AUTOMATE PROGRAMMABLE INDUSTRIEL (API) POUR LE DEVIATEUR DE BOUTEILLES (CEVITAL)>>mémoire de fin d'études 2005/2006,option électromécanique.

[15] Documentation technique interne de CEVITAL.

Annexes

Propriétés de la table des mnémoniques

Nom : Mnémoniques
Auteur :
Commentaire :
Date de création : 30/04/2018 17:23:32
Dernière modification : 07/05/2018 16:48:29
Dernier filtre sélectionné : Tous les mnémoniques
Nombre de mnémoniques : 56/56
Dernier tri : Opérande ordre croissant

Etat	Mnémonique	Opérande	Type de données	Commentaire
X				
X				
X				
X				
	13KA1	A 124.0	BOOL	PRET APRES P.O.M
	13KA2	A 124.1	BOOL	EN DEFAULT
X		A 124.2	BOOL	SINGLISATION VOYANT VERT
X		A 124.3	BOOL	SINGLISATION VOYANT ROUGE
X		A 124.4	BOOL	PRESSION 6 BARS OK
	14Y2	A 124.5	BOOL	ELECTROVANNE
	14Y3	A 124.6	BOOL	ELECTROVANNE
	10KM1	A 124.7	BOOL	CONTACTOR
	M M	A 125.0	BOOL	MARCHE MOTEUR
X		A 126.3	BOOL	
	11S1	E 124.0	BOOL	REARMEMENT DEFAULT
	11S2	E 124.1	BOOL	MARCHE
	11S3	E 124.2	BOOL	ARRET
	11B3	E 124.3	BOOL	CELLULE BOURRAGE SORTIE TRAIN
	11B4	E 124.4	BOOL	CELLULE BOURRAGE DANS VOLET
	11B5	E 124.5	BOOL	PRESSOSTAT
	12B6	E 124.6	BOOL	INDUCTIF PRISE D'ORIGINE
	8SP1	E 124.7	BOOL	PORTE 1
	8SP2	E 125.0	BOOL	PORTE 2
	8SP3	E 125.1	BOOL	PORTE 3
	8SP4	E 125.2	BOOL	PORTE 4
	10B1	E 125.3	BOOL	CELLULE COMPTAGE
	AR	E 125.4	BOOL	ARRET D'ORGENCE
	DV	E 125.5	BOOL	DEFAULT VARIATEUR
	DEMA	E 125.6	BOOL	DEFAULT MACHINE AVALE
	ELECTROVANNE	E 125.7	BOOL	"14Y3"
	ELECTROVANNE 2	E 126.0	BOOL	"14Y2"
	C1	E 126.1	BOOL	CAPTEUR BOUTEILLE
	CBH	E 126.2	BOOL	CAPTEUR DE BOUCHONS
	RS	E 126.3	BOOL	RESET
	CBT	E 126.4	BOOL	CAPTEUR DE BOUTIELLE
	RS2	E 126.5	BOOL	RESET Z2
	C2	E 126.6	BOOL	
	BLOC DES DEFAULT GEN	FC 1	FC 1	
	BLOC DEFAULT BOUCHON	FC 2	FC 2	
	BLOC DEFAULT BOUTIELLE	FC 3	FC 3	
	BLOC DE PROGRAMME	FC 4	FC 4	
	P1	M 0.0	BOOL	PORTE1
	P2	M 0.1	BOOL	PORTE2
	P3	M 0.2	BOOL	PORTE3
	P4	M 0.3	BOOL	PORTE4
	AD	M 0.4	BOOL	ARRET D'ORGENCE

Etat	Mnémonique	Opérande	Type de données	Commentaire
	PRE	M 0.5	BOOL	PRESSOSTAT
	DFV	M 0.6	BOOL	DEFAULT VARIATEUR
	DMA	M 1.0	BOOL	DEFAULT MACHINE AVAL
	DFBOC	M 1.1	BOOL	DEFAULT SANS BOUCHONS
	ALDBT	M 1.2	BOOL	DEFAULT BOUTIELLE F
	ALDB	M 10.1	BOOL	DEFAULT BOUCHONS
	GV	M 10.2	BOOL	
	DFBOT	M 10.3	BOOL	DEFAULT BOUTIELLE
X		M 10.5	BOOL	
	PROG_ERR	OB 121	OB 121	Programming Error

OB1 - <offline>

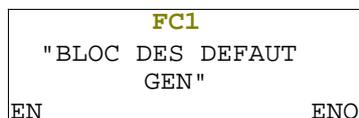
" "

Nom : **Famille :**
Auteur : **Version :** 0.1
Horodatage Code : 08/05/2018 11:59:24
Interface : 15/02/1996 16:51:12
Longueur (bloc/code /données locales) : 00210 00068 00022

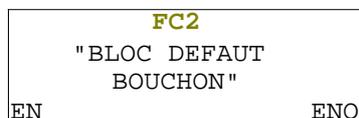
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
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

Bloc : OB1 "Main Program Sweep (Cycle)"

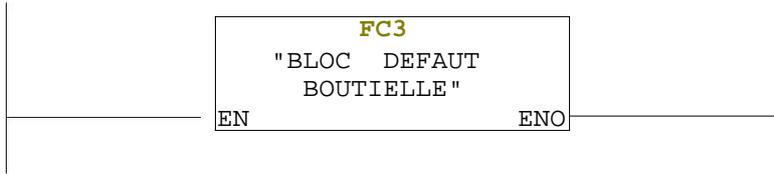
Réseau : 1



Réseau : 2



Réseau : 3



Réseau : 4



Réseau : 5



DB50 - <offline> - Vue des déclarations

""

Bloc de données (DB) global 50

Nom : Famille :
Auteur : Version : 0.1
 Version de bloc : 2
Horodatage Code : 30/04/2018 16:05:37
 Interface : 30/04/2018 16:05:37
Longueur (bloc/code /données locales) : 00132 00012 00000

Bloc : DB50

Adresse	Nom	Type	Valeur initiale	Commentaire
0.0		STRUCT		
+0.0	COMP	WORD	W#16#0	Variable temporaire de réservation
+2.0	PORTE1	BOOL	FALSE	
+2.1	PORTE2	BOOL	FALSE	
+2.2	PORTE3	BOOL	FALSE	
+2.3	PORTE4	BOOL	FALSE	
+2.4	ARRETDORGENCE	BOOL	FALSE	
+2.5	PRES	BOOL	FALSE	
+2.6	DEFAUTVARRIATEUR_1	BOOL	FALSE	
+2.7	DEFAUTMACHINAVAL	BOOL	FALSE	
+4.0	COMTAGE	WORD	W#16#0	
+6.0	COMPAGE2	WORD	W#16#0	
+8.0	FO	WORD	W#16#0	
+10.0	marche1	BOOL	FALSE	
+10.1	arret	BOOL	FALSE	
+10.2	fautmarcharret	BOOL	FALSE	
+10.3	rearmementdefaut	BOOL	FALSE	
=12.0		END_STRUCT		

FC2 - <offline>

"BLOC DEFAUT BOUCHON"

Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) : 00186 00084 00000

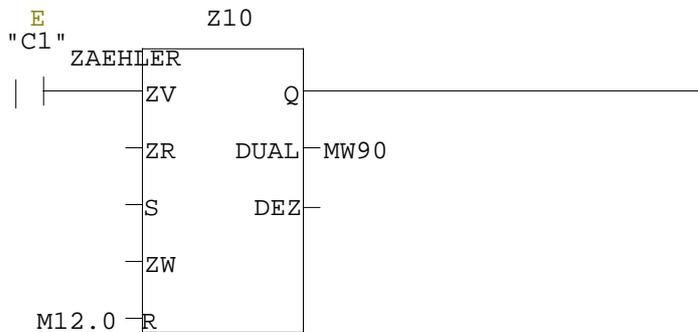
Famille :
Version : 0.1
Version de bloc : 2
 21/05/2018 15:28:11
 24/04/2018 19:23:25

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC2 DEFAUT BOUCHON

Réseau : 1

E126.1
 CAPTEUR
 BOUTEEILLK

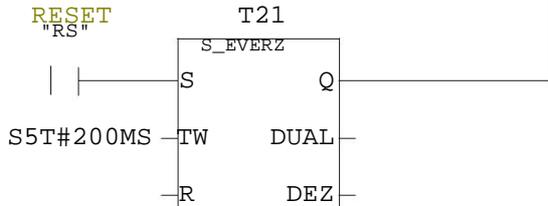


Réseau : 2

E126.2
CAPTEUR
DE
BOUCHONS



E126.3
RESET
"RS"



Réseau : 3

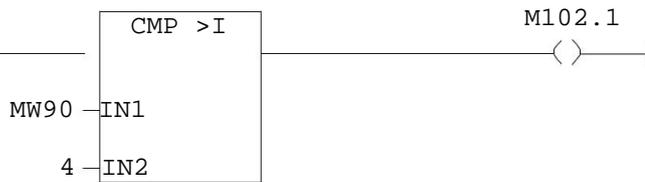
M1.1
DEFAULT
SANS
BOUCHONS

E126.3
RESET
"RS"

M10.2
"GV"

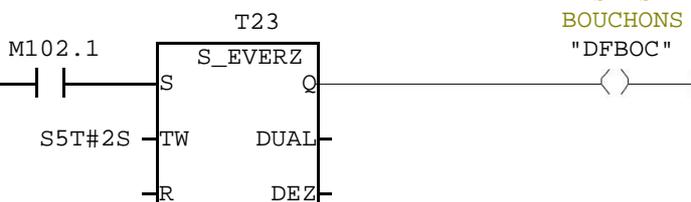


Réseau : 4



Réseau : 5

M1.1
DEFAULT
SANS
BOUCHONS
"DFBOC"



FC3 - <offline>

"BLOC DEFAUT BOUTIELLE"

Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) : 00186 00084 00000

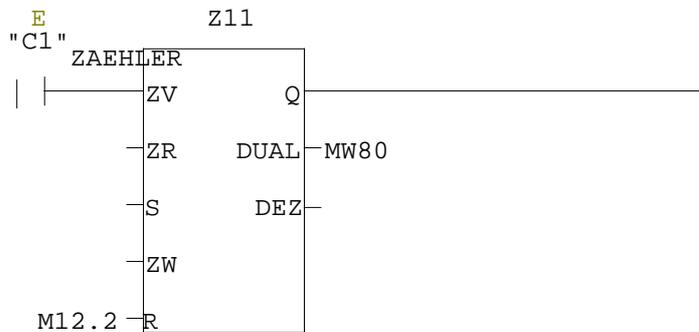
Famille :
Version : 0.1
Version de bloc : 2
 21/05/2018 15:27:06
 24/04/2018 19:23:36

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

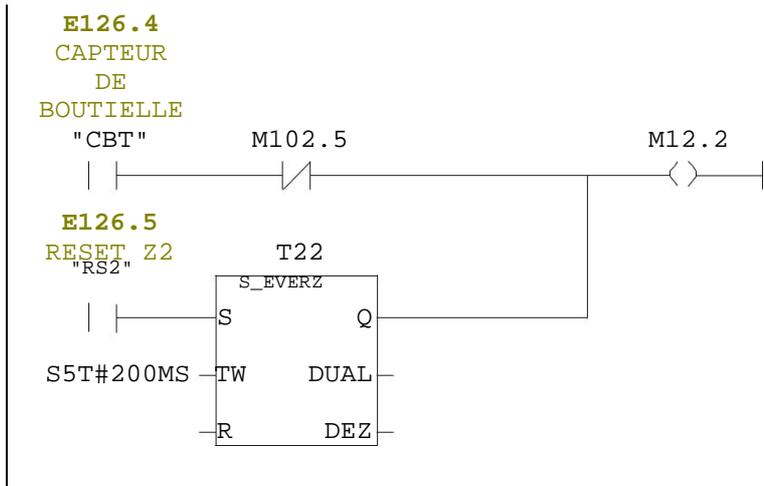
Bloc : FC3 DEFAUT BOUTIELLE

Réseau : 1

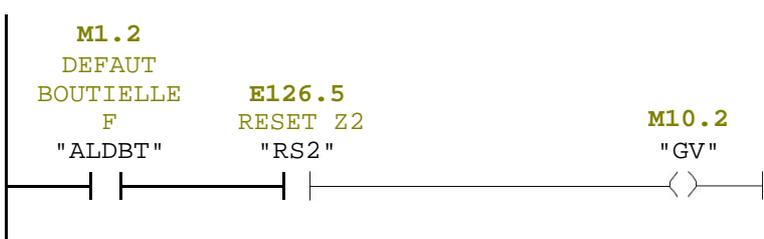
E126.1
 CAPTEUR
 BOUTEEILLK



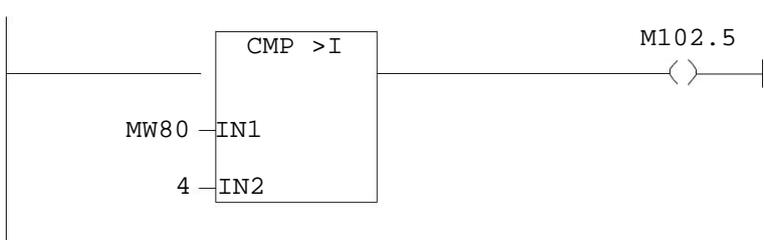
Réseau : 2



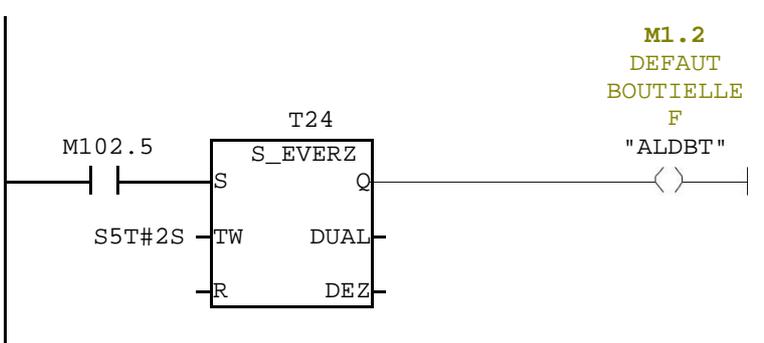
Réseau : 3



Réseau : 4



Réseau : 5



FC4 - <offline>

"BLOC DE PROGRAME"

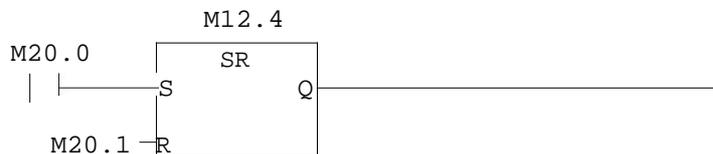
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) : 00468 00336 00000

Famille :
Version : 0.1
Version de bloc : 2
 05/06/2018 23:27:32
 24/04/2018 19:23:46

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC4 fonctionnement systém

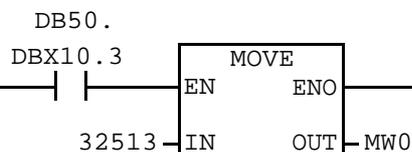
Réseau : 1 marche /arrêt



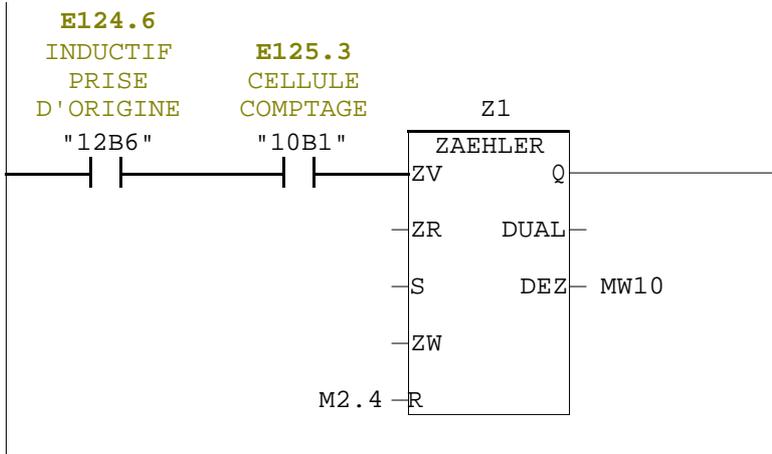
Réseau : 2



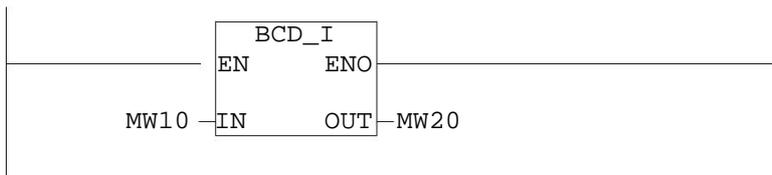
Réseau : 3



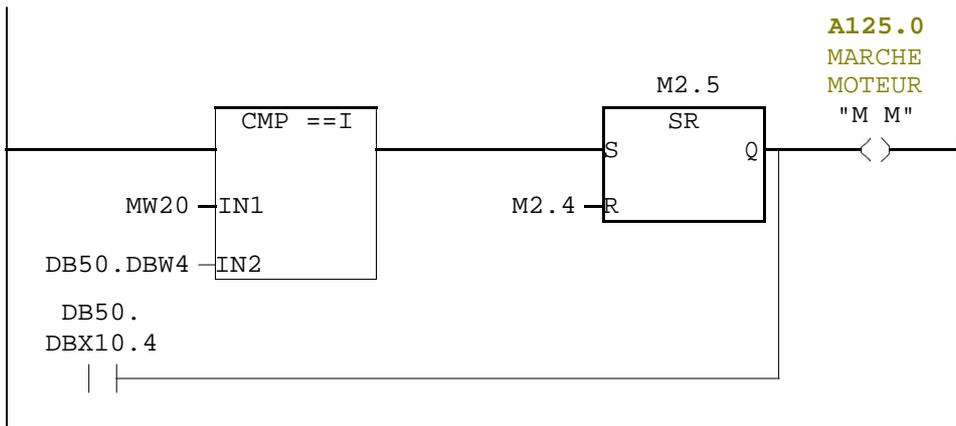
Réseau : 4 marche moteur



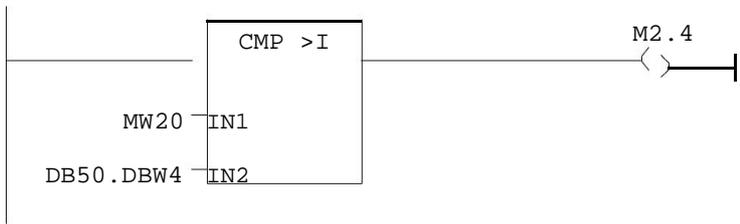
Réseau : 5



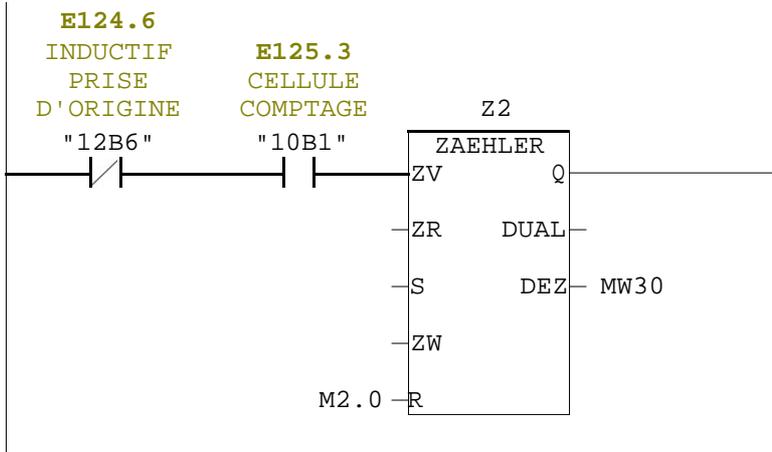
Réseau : 6



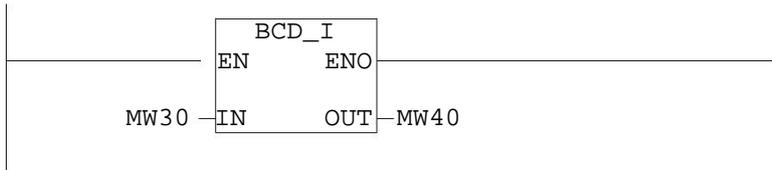
Réseau : 7



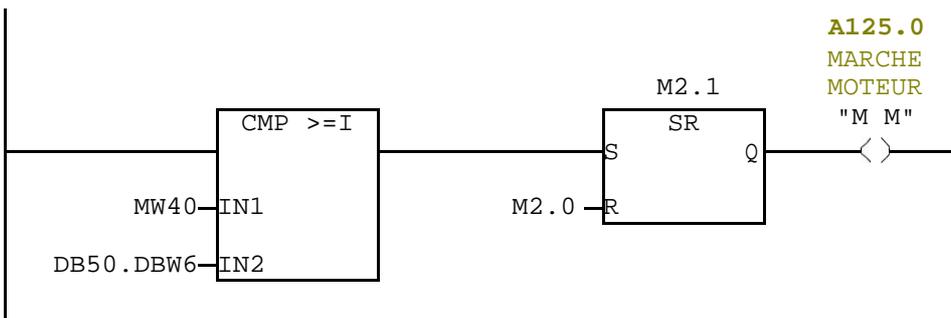
Réseau : 8



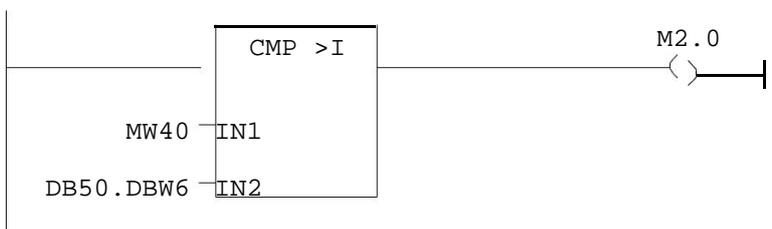
Réseau : 9



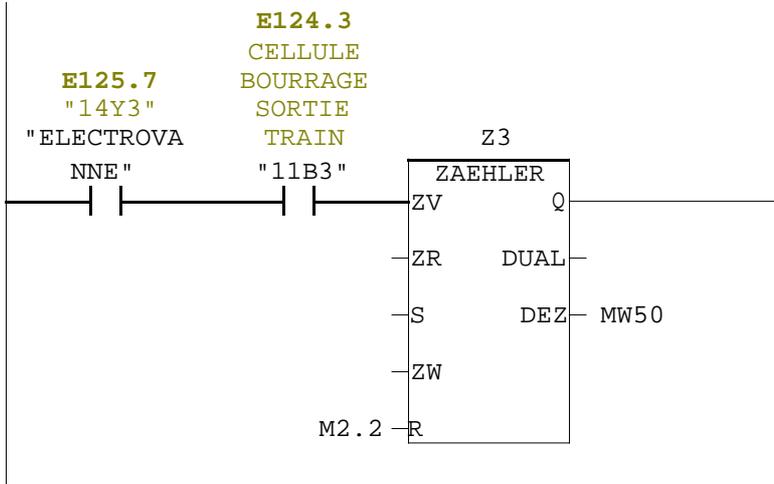
Réseau : 10



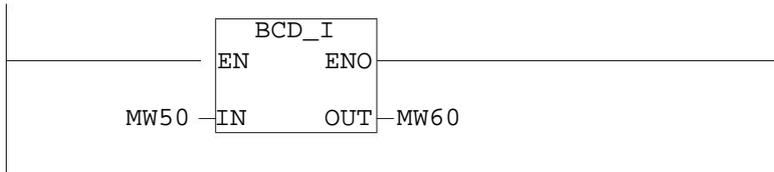
Réseau : 11



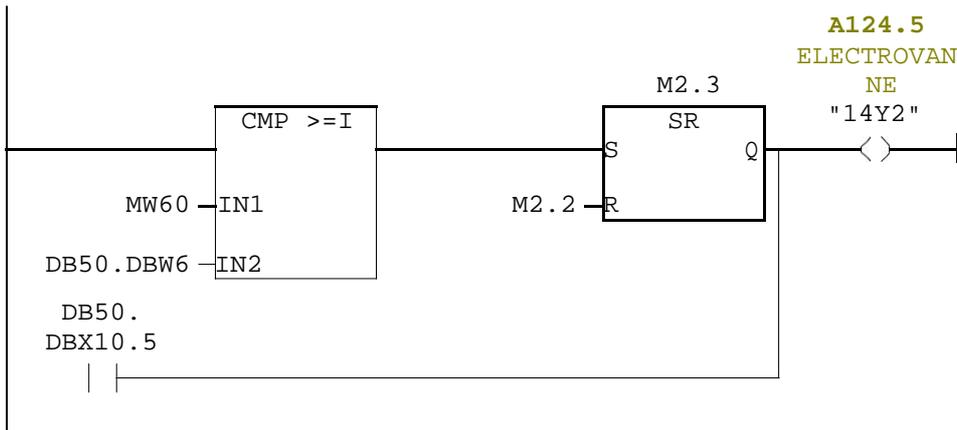
Réseau : 12 lae électrovanne



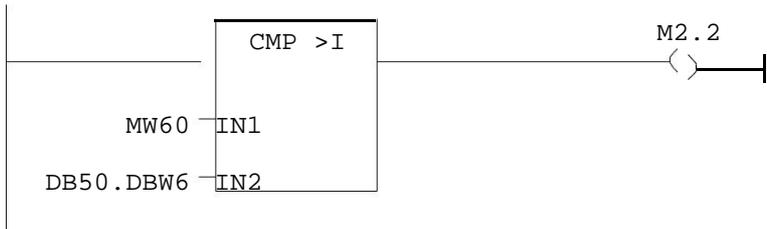
Réseau : 13



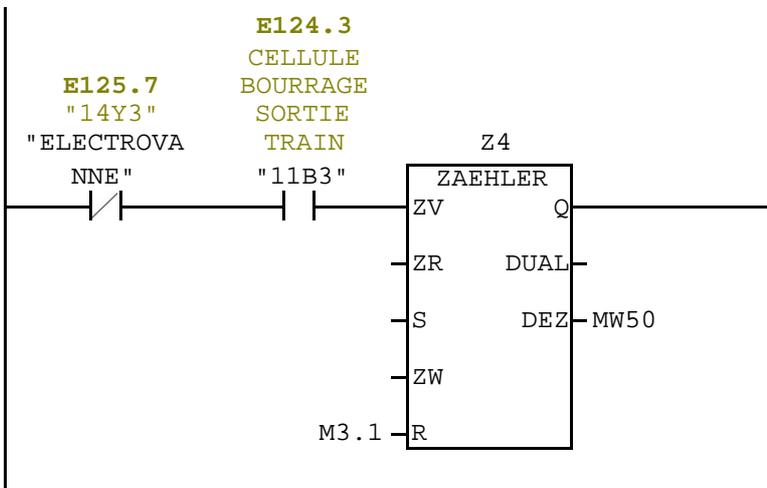
Réseau : 14



Réseau : 15



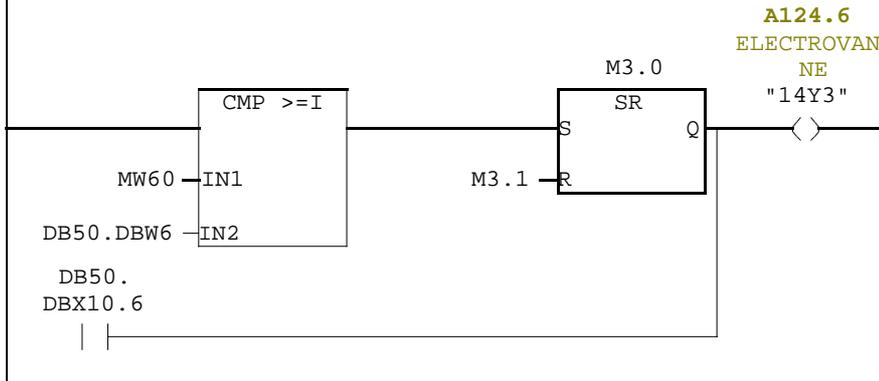
Réseau : 16



Réseau : 17



Réseau : 18



Réseau : 19

