

Ministère de l'enseignement supérieur et de la
recherche scientifique



Université Abderrahmane mira Bejaia

Faculté de la technologie

Département Génie Électrique



Mémoire de fin d'étude

En vue de l'obtention du diplôme Master en Automatique

Spécialité : Automatique et Informatique Industrielle

Thème

***Automatisation d'alimentation de
trémies d'une ligne de production
de sucre avec supervision IHM***

Réalisé par :

M. BOUROUIS Mohammed Amine

M. CHIBOUTI Billal

Encadré par :

Mme. BELLAHSENE

Membres de jury :

M. HADDAR

M. YAHIAOUI

Promotion2020

Remerciements

Nous tenons tout d'abord à remercier Dieu Tout Puissant Mais aussi nous remercions notre encadreur Mme.Bellahsene pour sa disponibilité et ses efforts incessants pour rendre ce travail complet.

Nos profonds remerciements aux membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre travail.

Nous tenons à exprimer toutes nos reconnaissances à tous ceux qui ont contribués de prêt ou de loin à la réalisation de ce modeste travail.

Dédicaces

Nous dédions ce modeste travail :

À nos très chers parents pour leurs efforts et sacrifices afin que nous puissions aujourd'hui réaliser notre rêve en travaillant sur ce projet remarquable et instructif.

À nos frères et sœurs, et à l'ensemble des membres de nos familles qui nous ont encouragés pour fournir plus d'efforts et le meilleur de nous-même.

À nos ami(e)s et camarades pour avoir passés de très bons moments lors des périodes de travaux et pendant le cursus universitaire.

Liste des figures

FIGURE I.1 : Dévouteur à fond Plat.....	6
FIGURE I.2 : Etapes de transfert	7
FIGURE I.3 : Schéma global d'un capteur	7
FIGURE I.4 : Capteur à levier et à tige souple	8
FIGURE I.5 : Capteur de pression	8
FIGURE I.6 : Vérin pneumatique simple effet [13].....	9
FIGURE I.7 : Vérin pneumatique à double effet [13].....	9
FIGURE I.8 : Vanne TOR.....	10
FIGURE I.9 : Vanne de régulation.....	10
FIGURE II.1: Automate Programmable	14
FIGURE II.2: Architecture interne d'un API [12]	15
FIGURE II.3 : Exemple d'un programme avec Le GRAFCET.....	21
FIGURE II.4 : Exemple d'un programme LADDER	21
FIGURE II.5 : Exemple d'un comparateur programme LADDER	22
FIGURE II.6 : Exemple d'un programme en Fonction Bloc.	22
FIGURE II.7 : Présentation de la gamme de SIMATIC.	24
FIGURE II.8 : Automate S7-200.	24
FIGURE II.9 : Automate S7-300.	25
FIGURE II.10 : Automate S7-400.	25
FIGURE II.11 : La gamme SIMATIC C7.....	26
FIGURE II.12 : La gamme SIMATIC M7.....	27
FIGURE II.13 : Assistant nouveau projet	30
FIGURE II.14 : Choix de la CPU et du bloc d'organisation.....	30
FIGURE II.15 : Nom et création du projet.....	31
FIGURE II.16 : Station SIMATIC S7-300.	31
FIGURE II.17 : Configuration matériel.	32
FIGURE II.18 : Interface de simulation PLCSIM	33

Liste des figures

FIGURE II.19 : L'interface homme/machine un processus automatisé	34
FIGURE II.20 : Aperçu d'une fenêtre WinCC	35
FIGURE II.21 : Intégration projet WinCC à STEP 7	36
FIGURE III.1 : Partie gauche.....	48
FIGURE III.2 : Partie droite.....	49
FIGURE III.3 : Table des conditions	50
FIGURE III.4 : Table des étapes	51
FIGURE III.5 : Réseau d'activation des étapes 101-103	52
FIGURE III.6 : Réseau de variation du niveau de la Trémie 1	52
FIGURE III.7 : Réseau de commande d'ouverture de la vanne.....	53
FIGURE III.8 : Réseau de conduite du produit.....	53
FIGURE III.9 : Réseau d'alarme de pression cuve	54
FIGURE IV.1. : Fenêtre de sélection d'un pupitre.....	55
FIGURE IV.2. : Table des variables WinCC	56
FIGURE IV.3 : Vue d'accueil.....	56
FIGURE IV.4 : Vue du processus	57
FIGURE IV.5 : Vue d'alarmes.....	58

Liste des tableaux

TABLEAU I.1 : Instrumentation du système.....	5
TABLEAU II.1 : Différents types de variables contenues dans le STEP7	29
TABLEAU III.1 : Description des symboles	38

Liste des abréviations

API : Automate Programmable Industriel.

PID : Proportionnel Intégral Dérivé

CPU : Central Processing Unit (Unité Centrale de Traitement).

RAM : Random Access Memory (Mémoire à Accès Aléatoire).

ROM : Read-Only Memory (Mémoire Morte ou Mémoire de Lecture).

EPROM : Erasable Programmable Read-Only Memory (Mémoire Morte Reprogrammable).

TOR : Tout Ou Rien.

A/N : Analogique Numérique.

N/A : Numérique Analogique.

LD : Ladder Diagram (Schéma à Relais).

FBD : Function Block Diagram (Schéma Par Bloc).

ST : Structured Text (Texte Structuré).

IL : Instruction List (Liste d'Instructions).

MPI : Interface Multipoint.

HMI : Interface Homme/Machine.

PC : Partie Commande.

PO : Partie Opérative.

DOS : Disk Operating System (Système D'exploitation).

OB : Bloc D'Organisation.

DB : Bloc de Données.

FB : Bloc Fonctionnelle.

FC : Fonction.

SFC : Fonction Système.

SFB : Bloc Fonctionnelle Système.

CDC : Cahier De Charge.

CDCF : Cahier De Charge Fonctionnel.

Table des matières

Introduction générale.....	1
Présentation de l'entreprise CEVITAL	2
I.1 Introduction.....	5
I.2. Description globale de processus [1]	5
I.3. Instrumentation du système	5
I.3.1. Trémie	5
I.3.2. Dévouteur.....	6
I.3.3. SAS de transfert	6
I.3.4. Aiguillage rotatif.....	7
I.3.5. Capteurs [13].....	7
I.3.5.1. Les différents types de Capteurs	7
I.3.5.1.1. Capteurs de niveau	7
I.3.5.1.2. Capteurs de position.....	8
I.3.5.1.3. Capteurs de pression	8
I.3.5.1.4. Capteurs de proximité	8
I.3.6. Actionneurs	9
I.3.6.1. Vérins	9
I.3.6.2. Electrovanne	10
I.4. Conclusion	11
II.1. Introduction	13
II.2. Historique	13
II.3. Définition.....	13
II.4. Le rôle de l'automate	14
II.5. Architecture générale des API [9]	15
II.5.1. Module d'alimentation	15
II.5.2. Unité centrale.....	16
II.5.2.1. Le Processeur [5]	16

Table des matières

II.5.2.2. Les mémoires [5]	17
II.5.3. Modules d'entrées/sorties (E/S).....	17
II.5.4. Les liaisons de communication.....	19
II.5.5. Eléments auxiliaires.....	19
II.6. Critères de choix d'un automate [2] [9]	19
II.7. Langages de programmation [16].....	20
II.7.1. Les langages graphiques [16] [9].....	20
II.7.1.1. Le GRAFCET	20
II.7.1.2. Diagramme Ladder	21
II.7.1.3. Bloc de Fonction.....	22
II.7.2. Les langages textuels	22
II.7.2.1. Texte Structuré	23
II.7.2.2. Liste d'instructions	23
II.8. Présentation de la gamme SIMATIC de SIEMENS [4] [12]	23
II.8.1. Les différentes variantes dans la gamme SIMATIC.....	24
II.8.1.1. SIMATIC S7.....	24
II.8.1.2. SIMATIC C7	26
II.8.1.3. SIMATIC M7	26
II.9. Logiciel de programmation STEP 7 [7]	27
II.9.1. Structure du programme STEP 7.....	27
II.9.2. Blocs existant sur STEP 7 [12] [7]	28
II.9.3. Mémentos	28
II.9.4. Mnémoniques	29
II.9.5. Les différents types de variables.....	29
II.9.6. Démarrage du logiciel STEP 7	30
II.9.7. Création d'un projet STEP 7	30
II.9.8. Configuration matérielle.....	31

Table des matières

II.9.9. Simulation du programme sous STEP 7	32
II.9.9.1. PLCSIM [6] [12]	33
II.9.9.2. Configuration du PLCSIM [6].....	33
II.9.10. Les logiciels de conception des interfaces homme/machine Siemens :	33
II.9.10.1. Description du logiciel WinCC Flexible	34
II.9.10.2. Elément du Win CC Flexible.....	34
II.9.10.3. Intégration de WinCC flexible à STEP 7 :	35
II.10. Conclusion	36
III.1. Introduction	37
III.2. Système à automatiser	37
III.2.1. Table symboles.....	37
III.2.2. Rôle du cahier de charges.....	39
III.2.3. Cahier de charges fonctionnel du système	39
III.3. Description de l'organigramme.....	40
III.3.1. Organigramme partie gauche	48
III.3.2. Organigramme partie droite	49
III.4. Programmation	50
III.4.1. Table des mnémoniques	50
III.4.1.1. Entrées	50
III.4.1.2. Mémentos	51
III.4.2. Création du bloc OB [4]	51
III.4.3. Création du bloc FC [4].....	51
III.5. Conclusion.....	54
IV.1. Introduction	55
IV.2. Création du projet.....	55
IV.3. Table des variables.....	56
IV.4. Présentation des vues du système	56

Table des matières

IV.5. Conclusion	58
Conclusion générale	59

Introduction

Générale

Introduction générale

La complexité de certains procédés industriels, et la nécessité d'une intervention rapide et exacte pour parer à d'éventuels dangers dans les grandes installations électriques. Cela laisse la manipulation humaine très délicate et souvent entachée d'erreurs. Les erreurs dans de tels procédés peuvent être très néfastes voir dangereuses pour le matériel et les personnes. L'avenue de l'automatisation et des automates programmables a su d'une manière très efficace d'éliminer de tels risques de manœuvre en remplaçant l'homme par la machine, La machine assiste à travers des programmes élaborés par les ingénieurs à la totalité des processus industriels dans ses moindres détails et dans toutes ses étapes.

L'automatisation est la priorité absolue dans les industries modernes, le débat sur ce phénomène est toujours sujet à controverse vu que la machine à tendance à remplacer l'homme dans plusieurs tâches, mais il reste tout de même nécessaire de souligner que l'homme par sa grandeur de concepteur ne peut hélas assurer les tâches que fait la machine dans plusieurs domaines.

Avant l'automatisation, il faut d'abord identifier l'ensemble du système et fixer les limites de l'intervention, dans le présent travail, le système à automatiser est une ligne de transfert de sucre vers plusieurs trémies de production.

L'analyse du fonctionnement du procédé débouchera sur un schéma logique du fonctionnement du système, pour fixer les moyens matériels à utiliser, et traduire les étapes de fonctionnement en séquence programmables.

L'élaboration du programme sera présentée sous le logiciel SIMATIC Manager de Siemens. La simulation et le test du programme se fera sous S7-PLCSIM, la supervision quant à elle se fera sous WinCC Flexible.

Ce Travail est réparti en quatre chapitres, le premier concernera l'identification des équipements, le deuxième pour présentation des API et logiciels de travail (STEP 7 et WinCC), le troisième pour l'analyse fonctionnelle et enfin un quatrième consacré à la supervision du système.

Présentation

De

L'entreprise CEVITAL

Présentation de l'entreprise CEVITAL

Présentation de l'entreprise CEVITAL

Historique

CEVITAL SPA, est parmi les entreprises algériennes qui ont vu le jour dès l'entrée de notre pays en économie du marché. Elle a été créée par des fonds privés en 1998. Son complexe de production se situe dans le port de Béjaïa et s'étend sur une superficie de 45000m².

Le complexe contribue largement au développement de l'industrie agroalimentaire nationale, il vise à satisfaire les besoins du marché national et à exporter le surplus, en offrant une large gamme de produits de qualité.

Situation géographique

CEVITAL est implanté au niveau du nouveau quai du port de Béjaïa **annexe 01**, à trois kilomètres cette ville, à proximité de la RN 26. Cette situation géographique de l'entreprise lui a beaucoup profité étant donné qu'elle lui confère l'avantage de proximité économique. En effet elle se trouve proche du port et de l'aéroport.

Activités de CEVITAL [1]

L'ensemble des activités de Cevital est concentré sur la production et la commercialisation des huiles végétales, de margarine et de sucre et se présente comme suit :

- Conditionnement de Sucre (192 tonnes/jour) ;
- Raffinage de sucre (1600 tonnes/jour) ;
- Production de margarine (600 tonnes/jour) ;
- Stockage des céréales (120000 tonnes/jour) ;
- Conditionnement d'huile (1400 tonnes/jour) ;
- Raffinage des huiles (1800 tonnes/jour) ;
- Fabrication d'emballage (PET) ; Poly-Ethylène-Téréphtalate (9600 unités/heure).

Missions et objectifs [1]

La mission principale de l'entreprise est de développer la production et d'assurer la qualité à des prix nettement plus compétitifs pour satisfaire les clients.

Les objectifs visés par CEVITAL sont :

- L'extension de ses produits sur tout le territoire national ;
- L'importation de graines oléagineuses 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 de graines oléagineuses ;

Présentation de l'entreprise CEVITAL

- 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.

Différents organes constituant le complexe CEVITAL

L'organigramme de l'**annexe 01** donne une vue générale sur les différents organes constituant le complexe CEVITAL [1].

Unité de conditionnement de sucre CEVITAL

Le Bâtiment de conditionnement de sucre a une superficie de 1131.9 m² et effectif de 430 Employés des différents services production ; maintenance et expéditions.

Le conditionnement est un emballage qui a pour rôle de :

- Protéger et conserver le produit ;
- Protéger le consommateur d'éventuel risques chimiques ;
- Faciliter la reconnaissance du produit dans les stands de ventes ;
- Attirer l'attention des acheteurs.

La production

Différentes dimensions et types de sac sont utilisés pour emballer le sucre produit par les machines qui sont :

Sac normal de 1kg

La production du paquet 1kg se fait par vingt-huit conditionneuses installées sur sept lignes indépendantes à une cadence nominale de 70 sacs/minute, dotées de doseur et d'une trieuse pondérale à la sortie de chaque conditionneuse qui consiste à contrôler le poids de chaque paquet, compteur de production et faire le tri sous les conditions suivantes :

- Sac supérieur à 1015 grammes : sac rejeté ;
- Sac inférieur à 995 grammes : sac rejeté ;
- Sac entre 995 et 1015 grammes : sac accepté.

Sac normal de 5kg

La production du sac 5Kg est assurée par une seule conditionneuse a doubles tube de remplissage à une cadence nominale de 25 sacs par tube dotée d'une trieuse pondérale OCS telle que la conditionneuse 1Kg, qui fait le tri suivant les conditions suivantes :

- Sac supérieur à 5150 grammes : sac rejeté ;
- Sac inférieur à 4925 grammes : sac rejeté ;
- Sac entre 4925 et 5150 grammes : sac accepté.

Présentation de l'entreprise CEVITAL

Sac verseur de 1kg

La production du sac verseur 1kg se fait par deux conditionneuses à une cadence nominale de 80 sachets verseur/minute, telle que :

- Sac verseur supérieur à 1015 grammes : sac rejeté ;
- Sac verseur inférieur à 985 grammes : sac rejeté ;
- Sac verseur entre 1015 et 985 grammes : sac accepté.

Sucre morceaux boîte de 750g

La production du sucre en morceaux, boîte de 750g est assurée par deux conditionneuses à une cadence nominale de 23 boîtes/minute, trié sous les conditions suivantes :

- Boîte supérieure à 765 grammes : boîte rejetée ;
- Boîte inférieure à 735 grammes : boîte rejetée ;
- Boîte entre 735 et 765 grammes : boîte acceptée.

CHAPITRE I

Généralités

I.1 Introduction

La diversité des systèmes automatisés et les exigences demandées sur ce dernier à provoquer une diversité de matériel et d'instrumentations du système pour faciliter la tâche de plus en plus, pour cela on voit une large gamme de matériels automatique, on va présenter quelques-uns dans ce chapitre.

I.2. Description globale de processus [1]

Le but de ce système est de transférer le sucre d'une trémie de stockage vers des trémies de réception des conditionneuses situées à une longue distance. Le système se compose :

Une unité Maxflo (SAS de transfert) composée d'une vanne de remplissage Inplatek et un aiguillage rotatif. Le tout est raccordé à des conduites de diamètre 100 millimètres. Au-dessus de l'unité de transfert se trouve une petite trémie tampon, équipée d'une sonde de niveau bas. La trémie stockage au-dessus de la trémie tampon est équipée de dévouteur. Le reste du système se compose de deux trémies de réception produit.

I.3. Instrumentation du système

Le tableau suivant nous indique l'instrumentation utilisé dans le système.

TABLEAU I.1 : Instrumentation du système

Type de Matériel	Unité	Quantité
Moteur Triphasé	U	1
Electrovanne	U	9
Capteurs	U	18

I.3.1. Trémie

Cette partie du système est destinée à recevoir le produit en poudre (sucre), transporté du silo de stockage jusqu'au point d'utilisation. Sa forme, ses dimensions et ses accessoires varient selon le type d'utilisation. Dans la plupart des cas, elle est constituée d'une partie supérieure cylindrique de diamètre variable selon sa capacité, d'une partie inférieure conique pour faciliter le déchargement du produit et d'une vanne ou plusieurs pour le déchargement manuel.

La trémie peut assurer les fonctions suivantes :

- Pesage et dosage (dans ce cas la trémie est accompagnée d'une ou plusieurs cellules de charge) ;
- Alimentation de mélangeurs, de pétrisseuses en continu, de conditionneuses (dans ce cas, la trémie est dotée de sondes de niveau maximum et minimum) ;

- Stockage intermédiaire avec sonde de niveau ;
- Séparation entre produit et air de transport à l'aide d'une toile filtrante interne, ou d'un filtre à manches, avec nettoyage en contre-pression, situé au sommet de la trémie.

I.3.2. Dévouteur

Le Dévouteur à fond plat est destiné à l'extraction du produit, sa fonction est d'assurer un brassage mécanique du produit pour éviter la prise en masse lors des phases de stockage ou après une étape de broyage.

Il comporte une pale tournante actionné par un réducteur qui déplace le produit jusqu'au point d'alimentation en évitant la formations des voûtes.

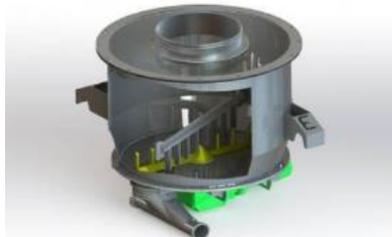


FIGURE I.1 : Dévouteur à fond Plat

I.3.3. SAS de transfert

C'est un système de transfert pneumatique particulièrement adapté aux produits très abrasifs.

Dans ce type de transfert pneumatique, la vanne vient couper le flux de produit au-dessus du réservoir de transfert. Ce réservoir ou sas de transfert est équipé d'une double enveloppe et d'un coude de sortie spécial permettant d'envoyer le produit en petite quantité dans la tuyauterie de transfert pneumatique.

Les actions exécutées sont les suivantes :

- Insertion du produit dans le sas ;
- La vanne coupe l'arrivé du produit ;
- Introduction contrôlée de l'air ;
- L'opération se renouvelle jusqu'à annulation.



FIGURE I.2 : Etapes de transfert

I.3.4. Aiguillage rotatif

Permet la distribution du produit d'une source unique vers plusieurs destinations dans le but de simplifier le système et de réduire l'encombrement.

I.3.5. Capteurs [13]

Les capteurs sont des composants de la chaîne d'acquisition dans une chaîne fonctionnelle. Les capteurs prélèvent une information sur le comportement de la partie opérative et la transforment en une information exploitable par la partie commande.

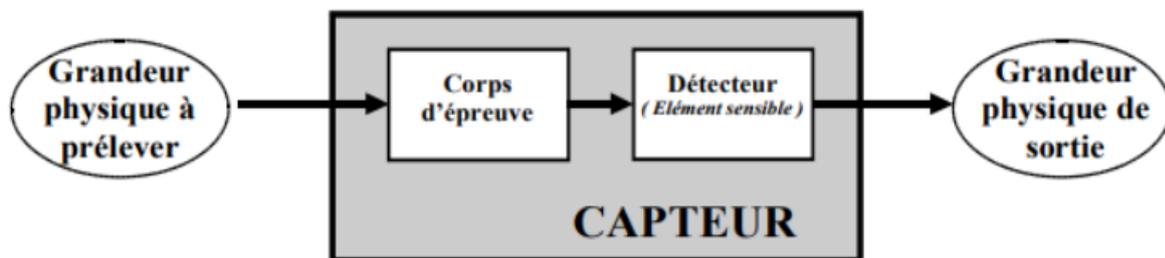


FIGURE I.3 : Schéma global d'un capteur

I.3.5.1. Les différents types de Capteurs

Par ailleurs il existe plusieurs types de capteurs utilisables selon les exigences des systèmes réalisés.

I.3.5.1.1. Capteurs de niveau

Un capteur de niveau est un dispositif électronique qui permet de mesurer la hauteur du matériau, en général du liquide, dans un réservoir ou un autre récipient.

Généralement, ce type de détecteur de niveau fonctionne comme une alarme haute pour signaler une condition de débordement ou en tant qu'indicateur pour une condition d'alarme basse. [15]

I.3.5.1.2. Capteurs de position

Les capteurs de position sont des capteurs de contact. Ils peuvent être équipés d'un galet, d'une tige souple, d'une bille. L'information donnée par ce type de capteur est de type **TOR** et peut être électrique ou pneumatique.

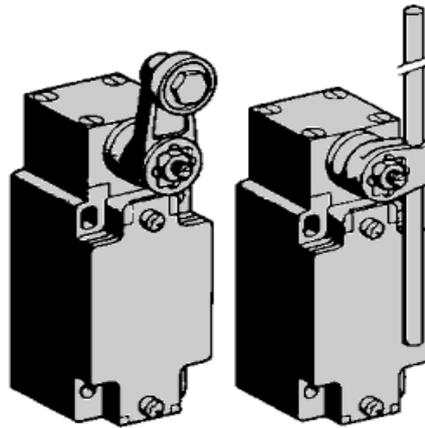


FIGURE I.4 : Capteur à levier et à tige

I.3.5.1.3. Capteurs de pression

Dans un capteur de pression on mesure la force qui s'exerce sur la surface d'un corps d'épreuve.

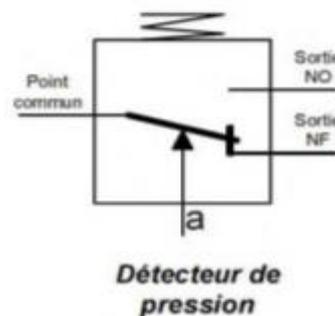


FIGURE I.5 : Capteur de pression

L'entrée de pression « a » pousse la lame du contact pour faire commuter le contact du détecteur lorsque la pression dépasse un certain seuil. Ce seuil mécanique est ajusté en modifiant la tension du ressort dans une vis. On peut utiliser ces détecteurs pour détecter qu'un vacuum est suffisant.

I.3.5.1.4. Capteurs de proximité

Servent à détecter la position du piston des vérins pneumatiques. Ils sont fixés directement sur le vérin. L'aimant torique du piston est détecté à travers la paroi non magnétisée du vérin.

I.3.6. Actionneurs

Un actionneur est un convertisseur électromécanique conçu pour mettre en mouvement un système mécanique à partir d'une commande électrique ou pour convertir une énergie en une autre.

I.3.6.1. Vérins

Les vérins permettent d'obtenir un mouvement rectiligne avec plus ou moins de force. Ils peuvent fonctionner soit avec de l'air comprimé (vérin pneumatique), soit avec de l'huile hydraulique (vérin hydraulique).

Dans notre cas nous avons utilisé les vérins pneumatiques dans lesquelles on trouve deux sortes de vérins qui sont :

- **Le vérin pneumatique à simple effet [13]** : Ce sont des vérins qui effectuent un travail dans un seul sens. Ils permettent soit de pousser soit de tirer une charge, exclusivement. Seules les positions extrêmes sont utilisées avec ce type de vérin. Un vérin pneumatique à simple effet n'a qu'une seule entrée d'air sous pression et ne développe un effort que dans une seule direction. La course de retour à vide est réalisée par la détente d'un ressort de rappel incorporé dans le corps du vérin.

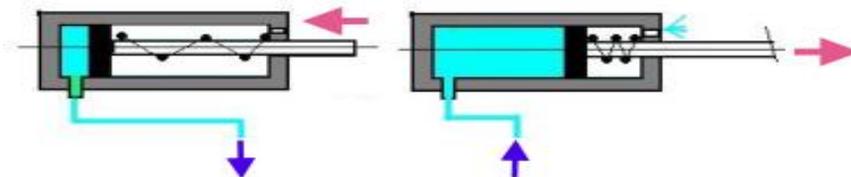


FIGURE I.6 : Vérin pneumatique simple effet [13]

- **Le vérin pneumatique à double effet** : Contrairement à la version à simple effet, ce type de vérin développe une force disponible à l'aller comme au retour pour produire un travail.

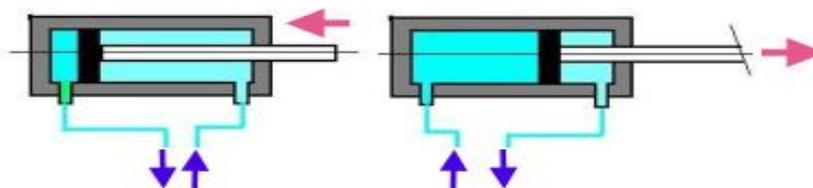


FIGURE I.7 : Vérin pneumatique à double effet [13]

I.3.6.2. Electrovannes

Une électrovanne est composée de deux parties :

- Une tête magnétique constituée principalement d'une bobine, tube, culasse, bague de déphasage, ressort(s).
- Un corps, comprenant des orifices de raccordement, obturés par clapet, membrane, piston, etc. selon le type de technologie employée. L'ouverture et la fermeture de l'électrovanne est liée à la position du noyau mobile qui est déplacé sous l'effet du champ magnétique engendré par la mise sous tension de la bobine.

Nous avons utilisé deux types d'électrovannes :

- **Vanne « TOR »** : La vanne tout ou rien, est soit complètement fermée ou ouverte selon l'information envoyée par l'API.



FIGURE I.8 : Vanne TOR

- **Vanne régulatrice** : La commande de la vanne contrôle le débit du fluide qui la traverse.

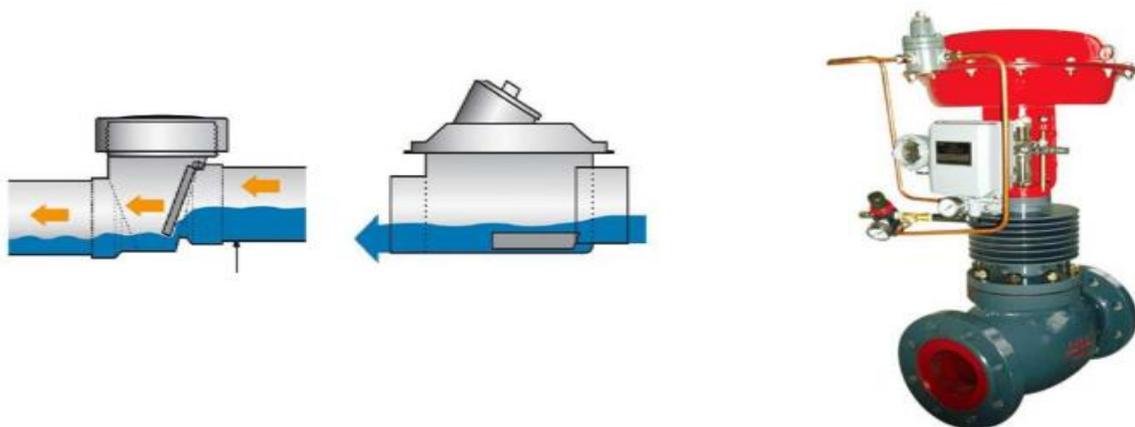


FIGURE I.9 : Vanne de régulation

I.4. Conclusion

Dans ce chapitre nous avons expliqué brièvement le fonctionnement du système automatisé, de l'instrumentation ainsi que les différents rôles et la description de chaque matériel utilisé tel que les capteurs, électrovanne, vérins ...etc.

CHAPITRE II

Présentation des API,

Step7, WinCC

II.1. Introduction

Un Automate Programmable Industriel (**API**) dit Programmable Logic Controller (**PLC**) dans le langage industriel, est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires. Il exécute une suite d'instruction introduite dans ses mémoires sous forme de programme, et s'apparente par conséquent aux machines de traitement de l'information.

Trois caractéristiques fondamentales le distinguent des outils informatiques tels que les ordinateurs utilisés dans les entreprises et le tertiaire :

- Il peut être directement connecté aux capteurs et pré actionneurs grâce à ses entrées/sorties industrielles ;
- Il est conçu pour fonctionner dans des ambiances industrielles sévères, (Température, vibration, microcoupures de la tension d'alimentation, Parasite, etc.) ;
- Enfin, sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme facilite son exploitation et sa mise en œuvre.

II.2. Historique

Les **Automates Programmables Industriels (API)** sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués. [2]

II.3. Définition

Selon la norme française EN 61131-1, un automate programmable est un :

« Système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour le stockage interne des instructions orientées utilisateurs aux fins de mise en œuvre des fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage, et de calcul arithmétique, pour commander au moyen d'entrées et de sorties tout ou rien ou analogiques divers types de machines ou de processus. L'automate programmable et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues. ».



FIGURE II.1: Automate Programmable

II.4. Le rôle de l'automate

L'automate programmable industriel est l'élément du système de contrôle qui commande directement les processus de fabrication en fonction du programme mis en mémoire, l'API reçoit les données des dispositifs d'entrées communiquant avec lui et les utilisent pour surveiller le système de contrôle.

Lorsque le programme demande l'exécution d'une opération, l'API envoie des données aux dispositifs de sortie pour leur ordonner cette opération.

L'API peut être utilisé pour contrôler une tâche simple et répétitive ou il peut se connecter à d'autres API ou à un ordinateur pour intégrer le contrôle d'un processus complexe.

Un API est le cœur de la partie commande et il est en relation avec les autres parties du système grâce à son interface d'entrée-sortie. Il effectue des fonctions d'automatisme programmées telles que :

- Logique combinatoire ;
- Logique séquentielle ;
- Temporisation ;
- Comptage ;
- Calculs numériques ;
- Asservissement et régulation.

II.5. Architecture générale des API [9]

En général, un automate programmable se compose essentiellement d'une unité centrale, un module d'entrées/sorties, un module d'alimentation, un module de stockage et de liaisons et des auxiliaires. [8]

L'architecture décrite ci-dessus est représentée dans la figure suivante :

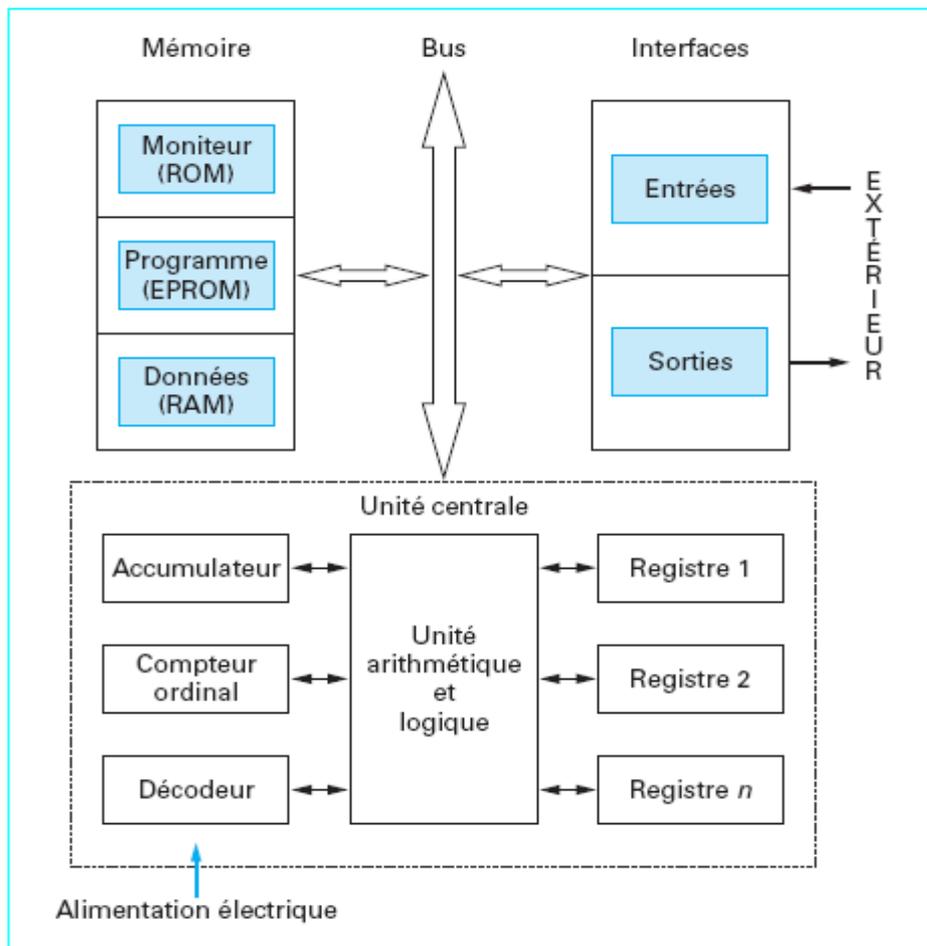


FIGURE II.2: Architecture interne d'un API [12]

II.5.1. Module d'alimentation

Ce module permet l'alimentation en tension continue nécessaire au bon fonctionnement de l'automate programmable ainsi que le circuit de charge. Il convertit la tension du réseau (AC 220) en tension de service (DC 24V, 15V ou 5V). Ce module doit posséder de bonnes performances face aux microcoupures du réseau, ainsi qu'un transformateur d'isolement pour lutter contre les perturbations du même réseau. [5]

II.5.2. Unité centrale

L'unité centrale (CPU) est l'élément le plus important dans l'automate programmable, elle peut être considérée comme le cerveau du système. Elle est constituée de deux composants principaux :

- Le Processeur ;
- Les mémoires.

II.5.2.1. Le Processeur [5]

Le processeur a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement de l'application. Mais en dehors de cette tâche de base, il réalise également d'autres fonctions telle que :

- Calcul des entrées/sorties ;
- Surveillance et diagnostique de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement ;
- Dialogue avec le terminal programmation aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages (contrôle) ou des vérifications de données (supervision).

Le processeur est organisé autour d'un certain nombre de registres, ce sont des mémoires rapides permettant la manipulation des informations qu'elles retiennent, ou leur combinaison avec des informations extérieures.

Les principaux registres existants dans un processeur sont :

➤ **L'accumulateur** C'est le registre où s'effectuent les opérations du jeu d'instruction, les résultats sont contenus dans ce registre spécial.

➤ **Le registre d'instruction** Il reçoit l'instruction à exécuter et décode le code opération. Cette instruction est désignée par le pointeur.

➤ **Le registre d'adresse** Ce registre reçoit, parallèlement au registre d'instruction, la partie opérande de l'instruction. Il désigne le chemin par lequel circulera l'information lorsque le registre d'instruction validera le sens et ordonnera le transfert.

➤ **Le registre d'état** C'est un ensemble de positions binaires décrivant, à chaque instant, la situation dans laquelle se trouve précisément la machine.

➤ **Les piles** Une organisation spéciale de registres constitue une pile, ces mémoires sont utilisées pour contenir le résultat de chaque instruction après exécution.

Ce résultat sera utilisé ensuite par d'autres instructions, et cela pour faire place à la nouvelle information dans l'accumulateur.

II.5.2.2. Les mémoires [5]

Un système à processeur est toujours accompagné d'un ou de plusieurs types de mémoires.

Les automates programmables industriels possèdent pour la plupart les mémoires suivantes :

➤ **Mémoire de travail** La mémoire de travail contient les parties du programme significatif pour son exécution. Le traitement du programme a lieu exclusivement dans la mémoire de travail et dans la mémoire système.

➤ **Mémoire système** La mémoire système contient les éléments de mémoire que chaque CPU met à la disposition du programme utilisateur comme, par exemple, mémoire image des entrées, mémoire image des sorties, mementos, temporisation et compteurs. La mémoire système contient, en outre, la pile des blocs et la pile des interruptions.

Elle fournit aussi la mémoire temporaire allouée au programme (pille des données locales).

➤ **Mémoire de chargement** La mémoire de chargement sert à l'enregistrement du programme utilisateurs sans affectation de mnémoniques ni de commentaires (ces derniers restent dans la mémoire de la console de programmation).

La mémoire de chargement peut être soit une mémoire vive (RAM), soit une mémoire EPROM.

➤ **Mémoire RAM non volatile** Zone de mémoire configurable pour sauvegarder des données en cas de défaut d'alimentation.

➤ **Mémoire ROM** se sont des mémoires mortes dont l'utilisateur ne peut que lire le contenu, dans le cas des automates programmables, elle contient le système d'exploitation qui gère la CPU.

II.5.3. Modules d'entrées/sorties (E/S)

Ils traduisent les signaux industriels en informations API réciproquement, appelées aussi coupleurs.

Beaucoup d'automates assurent cet interfaçage par des modules amovibles ou modulaires.

D'autres automates ont une structure monobloc, avec des modules intégrés dans un châssis de base, (cas des automates de télémécanique TSX17 et SIMATIC S7-314 IFM).

Le nombre total de modules est évidemment limité, pour des problèmes physiques :

- Alimentation électrique ;
- Gestion informatique ;
- Taille du châssis.

Différents types de modules sont disponibles sur le marché selon l'utilisation souhaitée, les plus répandus sont :

➤ **Entrées/sorties TOR (Tout ou Rien)**

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.

Leur nombre est en général de 8,16, 24 ou 32 entrées/sorties, qui peuvent fonctionner :

- En continu : 24V, 48V ;
- En alternatif : 24V, 48V, 100/120V.

➤ **Entrées/Sorties analogiques**

Elles permettent l'acquisition de mesures (entrées analogique), et la commande (sorties analogiques). Ces modules comportent un ou plusieurs convertisseurs Analogique/Numérique (A/N) pour les entrées et Numérique / Analogique (N/A) pour les sorties dont la résolution est de 8 à 16 bits.

Les standards les plus utilisés sont : $\pm 10V$, 0-10V, $\pm 20mA$, 0-20mA et 4-20mA.

Ces modules sont en général multiplexés en entrée pour n'utiliser qu'un seul convertisseur A/N, alors que les sorties exigent un convertisseur N/A par voie pour pouvoir garder la commande durant le cycle de l'API.

➤ **Modules spécialisés**

Ils assurent non seulement une liaison avec le monde extérieur, mais aussi une partie du traitement pour soulager le processeur et donc améliorer les performances.

Ces modules peuvent posséder un processeur embarqué ou une électronique spécialisée.

On peut citer :

- **Les cartes de comptage rapide** Elles permettent de saisir des événements plus courts que la durée du cycle travaillant à des fréquences qui peuvent dépasser 10kHz ;
- **Les Entrées/Sorties déportées** Leur intérêt est de diminuer le câblage en réalisant la liaison avec détecteur, ou actionneurs au plus près de ceux-ci, ce qui a pour effet d'améliorer la précision de mesure.

La liaison entre le boîtier déporté et l'unité centrale s'effectue par le biais d'un réseau de terrain suivant des protocoles bien définis. L'utilisation de la fibre optique permet de porter la distance à plusieurs kilomètres.

II.5.4. Les liaisons de communication

Elles s'effectuent :

- Avec l'extérieur par des **borniers**, sur lesquels arrivent des câbles transportant les signaux électriques, ou par liaison avec un bus de terrain ;
- Avec l'intérieur par des **bus**, liaisons parallèles entre les divers éléments. Il existe plusieurs types de bus, car on doit transmettre des données, des états, des adresses.

II.5.5. Eléments auxiliaires

Il s'agit principalement :

- **D'un ventilateur** qui est en général indispensable dans les châssis comportant de nombreux modules ou dans le cas où la température ambiante est susceptible de devenir assez élevée (plus de 40 °C) ;
- **De support mécanique** Il peut s'agir d'un rack (structure métallique) ;
- **Des indicateurs d'état** concernant la présence de tension, la charge de la batterie, le bon fonctionnement de l'automate, etc.

II.6. Critères de choix d'un automate [2] [9]

Le choix d'un automate programmable est généralement basé sur :

- **Nombre d'entrées / sorties** : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé ;
- **Type de processeur** : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue ;
- **Fonctions ou modules spéciaux** : certaines cartes (commande d'axe, pesage ...) permettront de soulager le processeur et devront offrir les caractéristiques souhaitées (résolution, ...) ;
- **Fonctions de communication** : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ...).

II.7. Langages de programmation [16]

La norme IEC 61131-3 (la Commission Électrotechnique Internationale) définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels.

Ces langages peuvent être divisés en deux grandes catégories.

- **Langages graphiques :**
 - **SFC** « Sequential Function Chart » ou GRAFCET ;
 - **LD** « Ladder Diagram » ou Schéma à Relais ;
 - **FBD** « Function Block Diagram » ou Schéma Par Bloc.
- **Langages textuels :**
 - **ST** « Structured Text » ou Texte Structuré ;
 - **IL** « Instruction List » ou Liste d'Instructions.

II.7.1. Les langages graphiques [16] [9]

C'est un langage de programmation visuelle qui utilise des schémas et des figures spécifiques qui font partie de la syntaxe du langage même.

II.7.1.1. Le GRAFCET

L'acronyme « **GRAFCET** » signifie : graphe Fonctionnel de Commande Etape Transition « **SFC** ». C'est une méthode de représentation graphique permettant de décrire le cahier de charge d'un automatisme. Il est adapté aux systèmes à évolution séquentielle, il est défini par un ensemble d'éléments graphiques de base traduisant le comportement de la partie commande vis-à-vis de ses entrées et de ses sorties.

Un programme « **GRAFCET** » décrit un procédé comme une suite d'étapes, reliées entre elles par des transitions. À chaque transition est associée une réceptivité, celle-ci est une condition logique qui doit être vraie pour franchir la transition et passer à l'étape suivante. Des actions sont associées aux étapes du programme. Le format graphique d'un programme « **GRAFCET** » est le suivant :

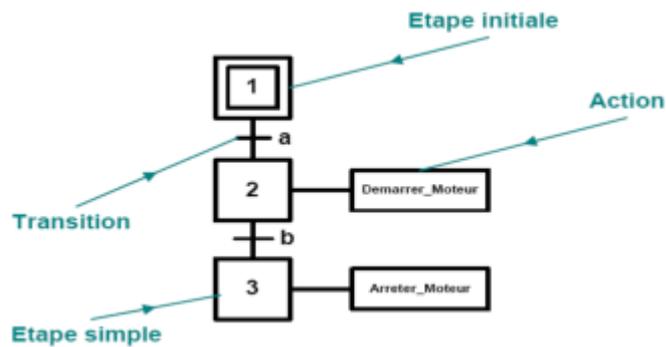


FIGURE II.3 : Exemple d'un programme avec Le GRAFCET.

Une étape représentée par un carré qui a un numéro identificateur et les actions associées sont indiquées dans un rectangle relié à la partie droite du carré ; (l'étape initiale est représentée par un carré double).

Une liaison orientée représentée par une ligne, parcourue par défaut de haut en bas ou de gauche à droite.

Une transition entre deux étapes à qui est associée une réceptivité inscrite à sa droite, est représentée par une barre perpendiculaire aux liaisons orientées qui relient ces étapes.

II.7.1.2. Diagramme Ladder

Le « LD » est une représentation graphique qui traduit directement des équations booléennes en un circuit électrique en combinant des contacts et des relais à l'aide des connexions horizontales et verticales, les contacts représentent les entrées (contact normalement ouvert, contact normalement fermé,) et les relais représentent les sorties. (Relais directs, relais inversés,), les diagrammes « LD » sont limités sur la gauche par une barre d'alimentation et par la masse sur la droite.

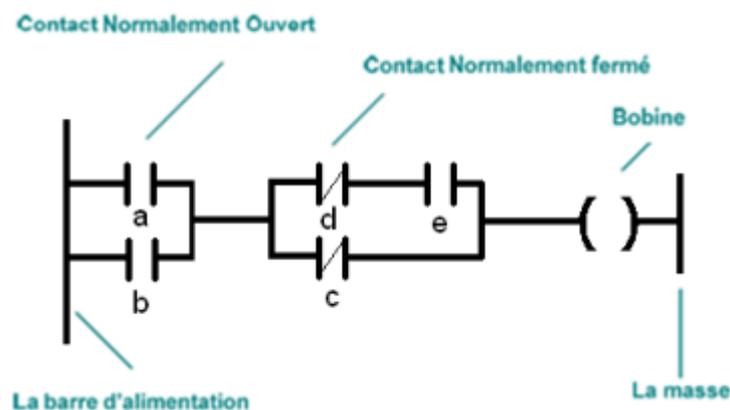


FIGURE II.4 : Exemple d'un programme LADDER

Le langage « LD » propose d'autre type de fonction tel que les fonctions de comptages et de temporisations, les fonctions arithmétiques et logiques, les fonctions de comparaison et de transfert.

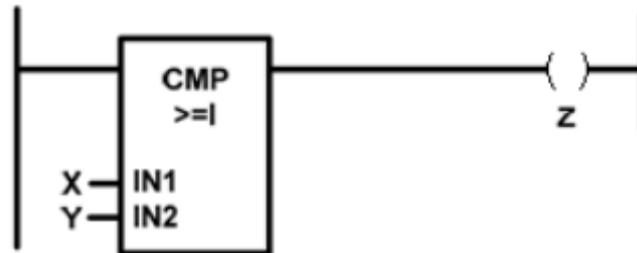


FIGURE II.5 : Exemple d'un comparateur programme LADDER

II.7.1.3. Bloc de Fonction

C'est un langage graphique qui permet la construction d'équations complexes à partir des opérateurs standard, ou de blocs fonctionnels, il se compose de réseaux de fonctions préprogrammées ou non, représentées par des rectangles qui sont connectés entre eux par des lignes.

La programmation avec le « FBD » est très souple et facile à apprendre, la plupart des fonctions nécessaires (les fonctions arithmétique et logique, les fonctions de temporisation, des blocs fonctionnels PID...) sont déjà disponible dans la bibliothèque, il suffit juste de les connectées et bien paramétrer les entrées et les sorties, c'est-à-dire respecter le type de variables lors de la connexion.

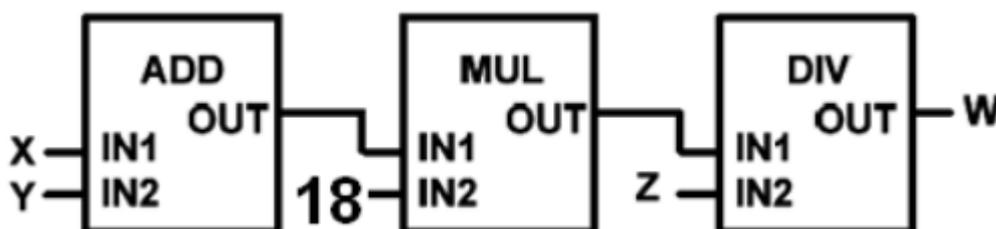


FIGURE II.6 : Exemple d'un programme en Fonction Bloc.

II.7.2. Les langages textuels

C'est un langage de programmation orienté vers l'écriture du programme via des expressions uniques qui respecte la syntaxe du langage.

II.7.2.1. Texte Structuré

Le langage « **ST** » (Structured Text) est un langage de programmation textuel de haut niveau dédié aux applications d'automatisation, il est utilisé principalement pour décrire les procédures complexes et difficilement modélisables avec les langages graphiques, il peut aussi être utilisé en tant que sous-programme avec d'autre langage de programmation.

Il utilise les mêmes énoncés que les langages de programmation de haut niveau (Pascal, C, C++...) comme: les assignations, les appels de fonction, les énoncés de contrôle (IF, THEN, ELSE, CASE) ou d'itération (FOR, WHILE, REPEAT) en plus des opérations arithmétiques et logiques.

II.7.2.2. Liste d'instructions

Le langage « **IL** » est un langage textuel de bas niveau (proche du langage machine), qui utilise un jeu d'instruction simple, il trouve sa puissance dans les applications de petites tailles, et dans la création de sous-programme ou procédure, car il permet un contrôle total et une optimisation parfaite du code, par contre dans les grandes applications il est très difficile de programmer avec le « **IL** », les programmes dans ce langage peuvent être traduit ou déduit des autres langages.

Le « **IL** » a la même structure que l'assembleur, il utilise un ou plusieurs registres de travail.

Les valeurs intermédiaires nécessaires pour l'exécution d'une instruction donnée seront mémorisées dans ces registres le temps de leurs utilisations et il possède un jeu d'instruction assez riche pour décrire toutes les opérations arithmétiques et logiques, les opérations de comptage et temporisation, la comparaison et le transfert...

II.8. Présentation de la gamme SIMATIC de SIEMENS [4] [12]

Siemens propose 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.

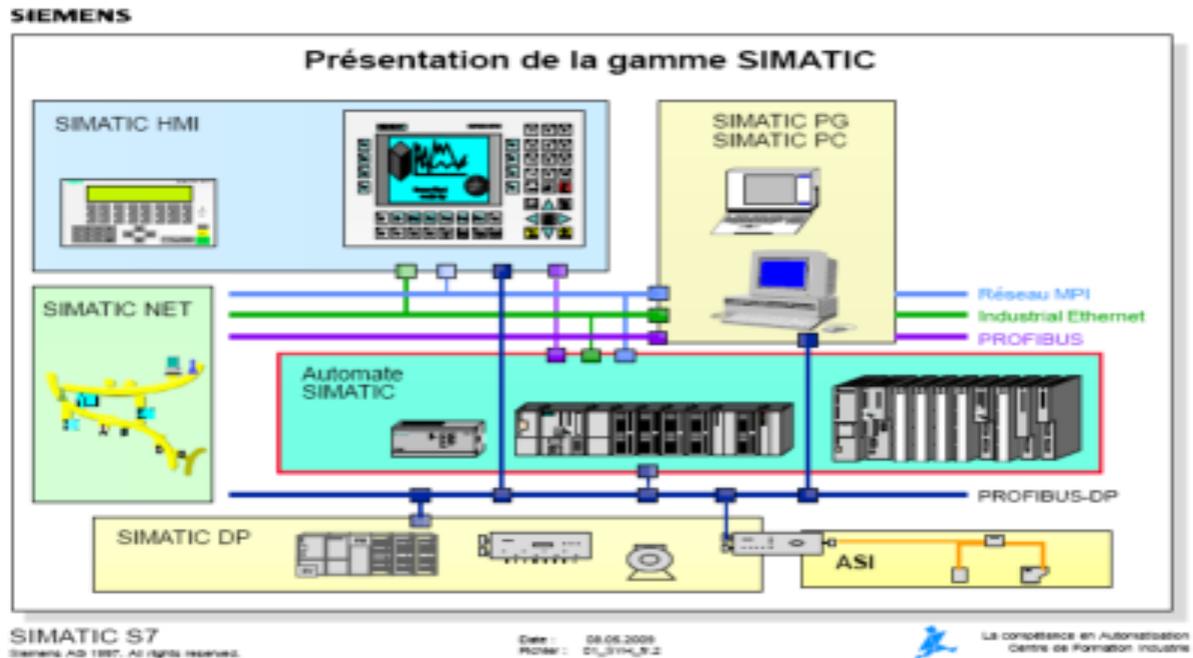


FIGURE II.7 : Présentation de la gamme de SIMATIC.

II.8.1. Les différentes variantes dans la gamme SIMATIC

La gamme SIMATIC ne se résume pas qu'aux automates programmables industriels S7 elle propose aussi des calculateurs tel que les modèles M7.

II.8.1.1. SIMATIC S7

Cette gamme d'automates comporte trois familles :

- **S7-200** : qui est un 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.8 : Automate S7-200.

- **S7-300** : est un automate modulaire 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.9 : Automate S7-300.

- **S7-400** : est un automate de haute performance 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 Industriel Ethernet.



FIGURE II.10 : Automate S7-400.

II.8.1.2. SIMATIC C7

Le SIMATIC C7 combine automate programmable et panneau opérateur dans une seule unité. L'automate compte la CPU, les modules d'entrées/sorties, et le panneau opérateur qui est utilisé comme une interface Homme/Machine HMI. Le C7 permet la visualisation des états de fonctionnement, des valeurs actuelles du processus et des anomalies.



FIGURE II.11 : La gamme SIMATIC C7.

II.8.1.3. 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 réel, 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.



FIGURE II.12 : La gamme SIMATIC M7.

II.9. Logiciel de programmation STEP 7 [7]

STEP 7 est le logiciel de base pour la configuration et la programmation des systèmes SIMATIC par exemple (S7-300).

Il fait partie de l'industrie logicielle SIMATIC. Les tâches de base qu'il offre à son utilisateur lors de la création d'une solution d'automatisation sont :

- La création et gestion de projet ;
- La configuration et le paramétrage du matériel et de la communication ;
- La gestion des mnémoniques ;
- La création des programmes ;
- Le test de l'installation d'automatisation.

Il s'exécute sous les systèmes d'exploitation de Microsoft à partir de la version Windows 95. Par conséquent, il s'adapte à l'organisation graphique orientée objet qu'offrent ces systèmes d'exploitation.

II.9.1. Structure du programme STEP 7

La programmation structurée permet la rédaction claire et transparente de programmes. Elle permet la construction d'un programme complet à l'aide de modules qui peuvent être échangés et/ou modifiés à volonté. Pour permettre une programmation structurée confortable, il faut prévoir plusieurs types de modules : (OB) Bloc d'organisation, (FB) Bloc fonctionnel, (FC) Fonction, (SFB) Bloc fonctionnel système, (SFC) Fonction système, (DB) Bloc de données. [11]

II.9.2. Blocs existant sur STEP 7 [12] [7]

Le système d'automatisation utilise différents types de blocs dans lesquels peuvent être mémorisés le programme utilisateur et les données correspondantes. Selon les exigences du processus, le programme peut être structuré en différents blocs.

- **Bloc d'organisation** : Les blocs d'organisation (**OB**) constituent l'interface entre le système d'exploitation et le programme utilisateur ;
- **Fonction** : Une fonction (FC) assure une fonctionnalité spécifique du programme. Les fonctions peuvent être paramétrables ;
- **Bloc fonctionnel** : Du point de vue du programme, les blocs fonctionnels s'apparentent aux fonctions FB et SFB, mais ils disposent en plus de zones mémoires spécifiques, sous forme de blocs de données d'instance. Les blocs fonctionnels conviennent pour la programmation de fonctionnalités récurrentes encore plus complexes (e.g., assurer des tâches de régulation) ;
- **Blocs de données** : Les blocs de données (DB) sont des zones de données du programme. DB utilisateur, mise à la disposition d'espace mémoire pour les variables de types données. Il existe deux types blocs de données :
 - **Bloc de données globales** : Ce bloc peut être accéder par tous les blocs (FB, FC et OB) et il peut être créer indépendamment des autres blocs et aucune donnée locale statique ne peut être déclarée. L'insertion, la supervision et la modification des variables se fait dans des blocs de données. Dans le bloc de données globale on peut aussi modifier les valeurs initiales et les valeurs en cours. La structure de ce bloc est définie librement.
 - **Bloc de données d'instance** : Ce bloc est associé à un FB et il ne peut pas être créer que si le FB associé est présent dans le programme. Les données locales et statiques peuvent être déclarées dans le FB, ces derniers ne sont pas perdues après la fin de l'exécution du bloc et sont conservé sur plusieurs cycles jusqu'à ce qu'elles soient à nouveau écrites.

II.9.3. Mémentos

Des mémentos sont utilisés pour le fonctionnement interne de l'automate pour lesquelles l'émission d'un signal n'est pas nécessaire. Les mémentos sont des éléments électroniques bistables servent à mémoriser les états logiques "0" et "1". Chaque automate programmable dispose d'une grande quantité de mémentos. On programme ces derniers comme des sorties.

II.9.4. Mnémoniques

Les mnémoniques sont les noms que l'on attribue aux variables globales de l'API. L'emploi des mnémoniques à la place des adresses absolues améliore considérablement la lisibilité et la clarté d'un programme et aide à isoler des défauts éventuels. Les mnémoniques ainsi définies sont utilisables dans tout le programme utilisateur d'un module programmable.

II.9.5. Les différents types de variables

Dans l'environnement de STEP 7, on utilise plusieurs types de variables qu'on doit déclarer au préalable. Le tableau ci-dessous résume les types de variables utilisés.

TABLEAU II.1 : Différents types de variables contenues

Groupe	Types de données	Signification
Type de données binaires	BOOL BYTE WORD DWORD	Les données de ce type Occupent 1 bit, 8bits, 16bits où 32bits.
Type de données sur caractère	CHAR	Les données de ce type Occupent Du jeu de caractère ASCII.
Type de données numériques	INT DINT REAL	Les données de ce type Permettent de traiter des valeurs Numériques.
Type de données temporelles	TIME DATE TIME OF-DAY S5TIME	Les données de ce type Représentent les diverses valeurs De durée et de date dans le STEP 7.

II.9.6. Démarrage du logiciel STEP 7

Pour lancer le logiciel STEP7, on localise l'icône SIMATIC Manager  sur l'écran de l'ordinateur puis avec un double clic sur cette icône, on se permet d'ouvrir sa fenêtre fonctionnelle.

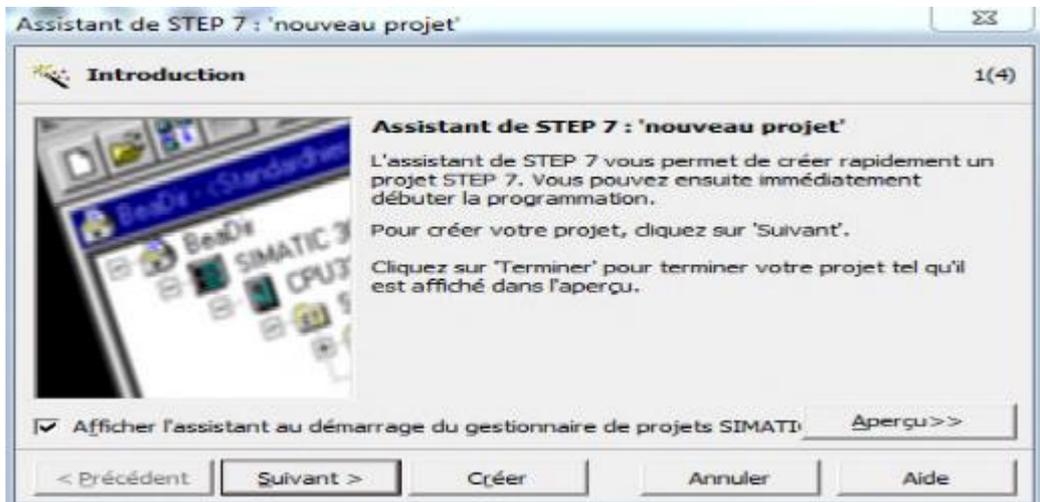


FIGURE II.13 : Assistant nouveau projet

II.9.7. Création d'un projet STEP 7

Le logiciel SIMATIC Manager étant maintenant ouvert puis assistant nouveau projet, après la sélection du type de la CPU et l'insertion du bloc d'organisation et le langage à liste,

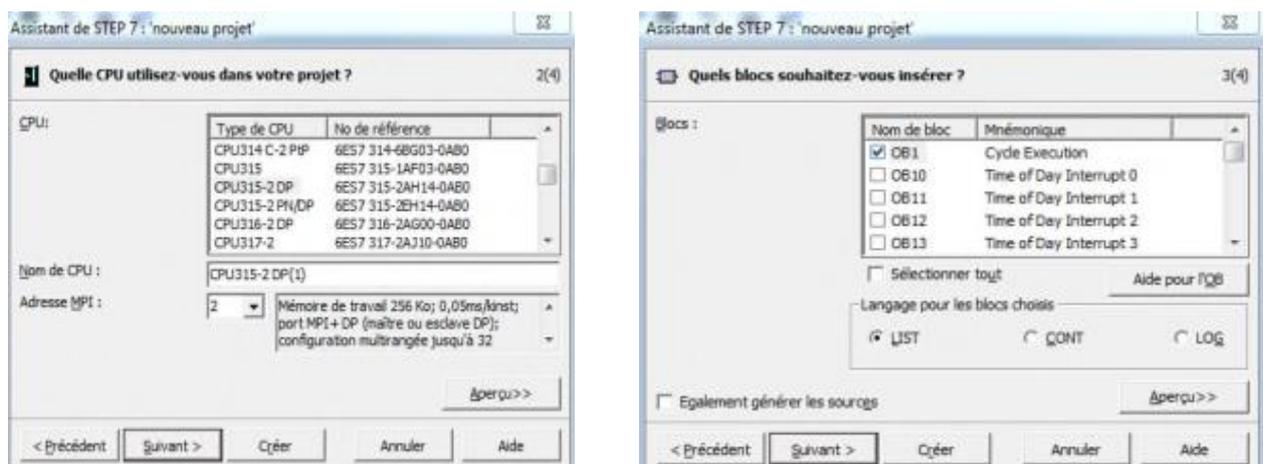


FIGURE II.14 : Choix de la CPU et du bloc

Une fenêtre s'ouvre pour donner un nom de projet et on clique sur « **Créer** » pour finaliser la création de notre projet.



FIGURE II.15 : Nom et création du projet

II.9.8. Configuration matérielle

C'est une étape importante qui correspond à l'agencement des châssis et des modules. Ces derniers sont fournis avec des paramètres définis par défaut en usine. [7]

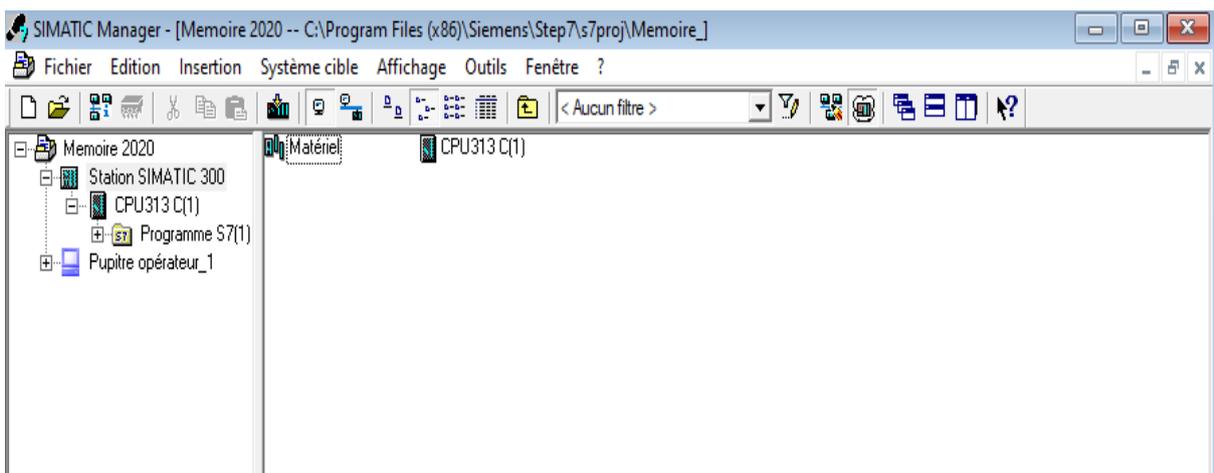


FIGURE II.16 : Station SIMATIC S7-300.

Par exemple le choix du matériels SIMATIC S7-300 avec une CPU313-C, nous conduit à introduire la hiérarchie suivante :

- On commence par le choix du châssis selon la station choisie auparavant. Pour la station300, on aura « RACK-300 » et on le glisse dans le châssis ;
- Dans l’emplacement N°1 on choisit le type d’alimentation, pour cela on a choisi l’alimentation PS 307 10A
- On glisse la CPU313-C dans l’emplacement N°2 ;
- L’emplacement N°3 réservés comme adresse logique pour un coupleur dans une configuration multi-châssis ;
- Dans les emplacements N° 4 et 5 on va ajouter des modules d’entrées et de sorties, dans l’emplacement N°4 on ajoute les modules d’entrées Digital Input DI16Xdc24V et pour l’emplacement N°5 on insère les modules de sorties Digital Output DO16Xdc24V/0.5A.
- Les autres emplacements sont réservés pour les modules qui se trouvent dans le fichier SM-300. L’automate se compose de deux modules d’entrée digitales et un pour les sorties et de trois modules d’entrées analogiques et enfin deux pour les sorties ;

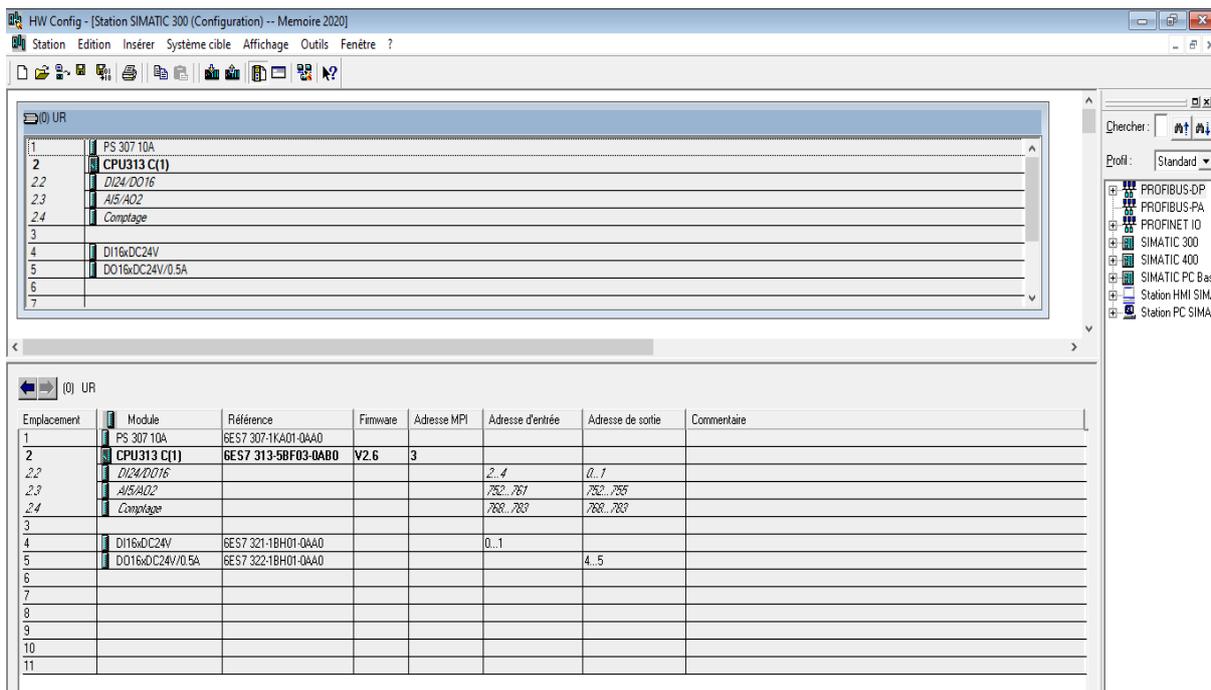


FIGURE II.17 : Configuration matériel.

II.9.9. Simulation du programme sous STEP 7

C’est l’exécution du programme élaboré sans avoir besoin d’un quelconque matériel c’est-à-dire seulement via la machine.

II.9.9.1. PLCSIM [6] [12]

L'application de simulation de modules S7 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 STEP 7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux). L'API S7 de simulation permet de tester des programmes destinés aux CPU S7-300 et aux 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 (**Variables**) utilisés par le programme. Tout en simulant l'exécution du programme en temps réel.

II.9.9.2. Configuration du PLCSIM [6]

Dans l'application S7-PLCSIM, on crée de nouvelles fenêtres pour visualiser les informations provenant de l'automate programmable de simulation et pour créer les diverses fenêtres, on procède comme suite :

- Créer la fenêtre permettant de modifier l'état des entrées intervenant dans le programme ;
 - Choisir la commande « insertion entrée » ou à partir de la barre d'outils ;
- Créer une fenêtre permettant de modifier l'état des sorties intervenant dans le programme ;
 - Choisir la commande « insertion sortie » ou à partir de la barre d'outils.

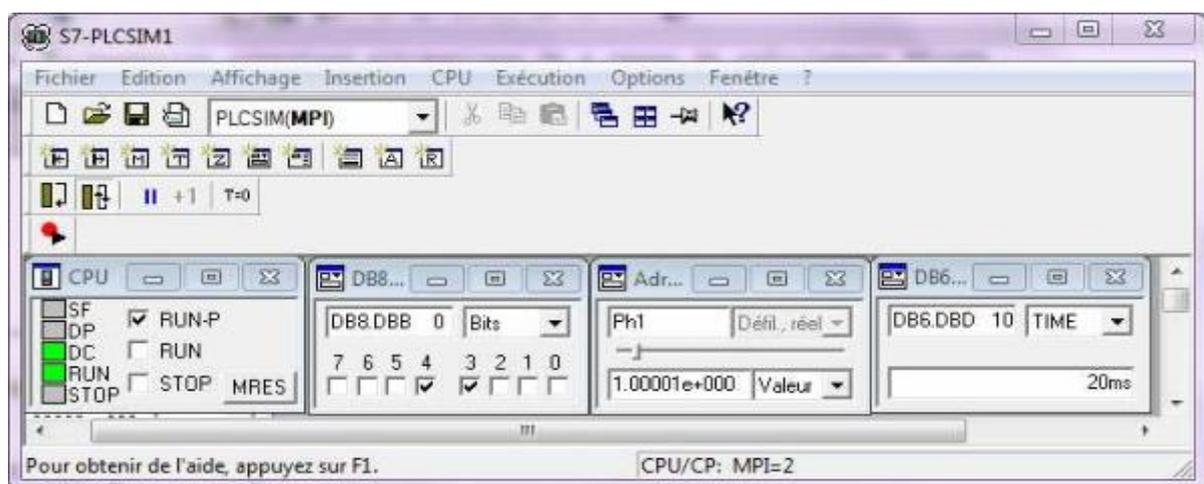


FIGURE II.18 : Interface de simulation PLCSIM

II.9.10. Les logiciels de conception des interfaces homme/machine Siemens :

Suite à l'automatisation industrielle, l'opérateur humain a été contraint de conduire ou de superviser des machines automatisées, en réduisant les prises d'information et les actions

directs sur le processus, ce qui conduit à l'élaboration d'interfaces d'interaction Homme /Machine (**IHM**) flexible et aussi lisible pour un simple opérateur. Le dialogue est d'autant plus facile car l'écran comporte des images avec des synoptiques, des graphes...etc. [3]

II.9.10.1. Description du logiciel WinCC Flexible

WinCC Flexible est le logiciel phare de Siemens permettant de créer des interfaces homme-machine sur pupitre tactile (**IHM**) ou sur écran. Il est un logiciel compatible avec l'environnement STEP 7, et propose pour la configuration de divers pupitres operateurs, une famille de système d'ingénierie évolutifs adaptes aux tâches de configuration.

Créer l'interface graphique et affecté les variables dans le programme STEP 7, 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 la communication entre l'IHM et l'Automate. [11]

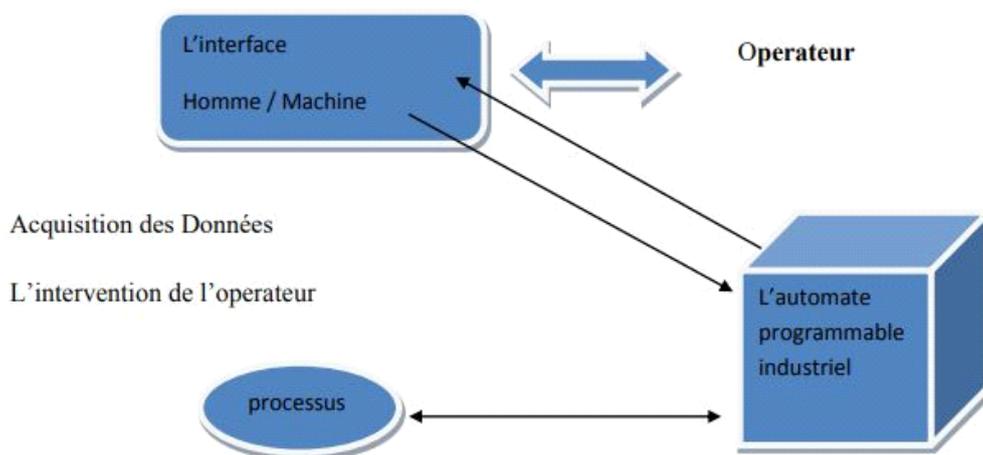


FIGURE II.19 : L'interface homme/machine un processus automatisé

II.9.10.2. Elément du Win CC Flexible

L'environnement de travail de WinCC Flexible se compose de plusieurs éléments. Certains de ces éléments sont liés à des éditeurs particuliers et uniquement visibles lorsque cet éditeur est actif il met à disposition un éditeur spécifique pour chaque tâche de configuration.

On peut configurer par exemple l'interface utilisateur graphique d'un pupitre operateur avec l'éditeur « vues ». Pour la configuration d'alarmes, on utilise par exemple l'éditeur « Alarme TOR ». [10]

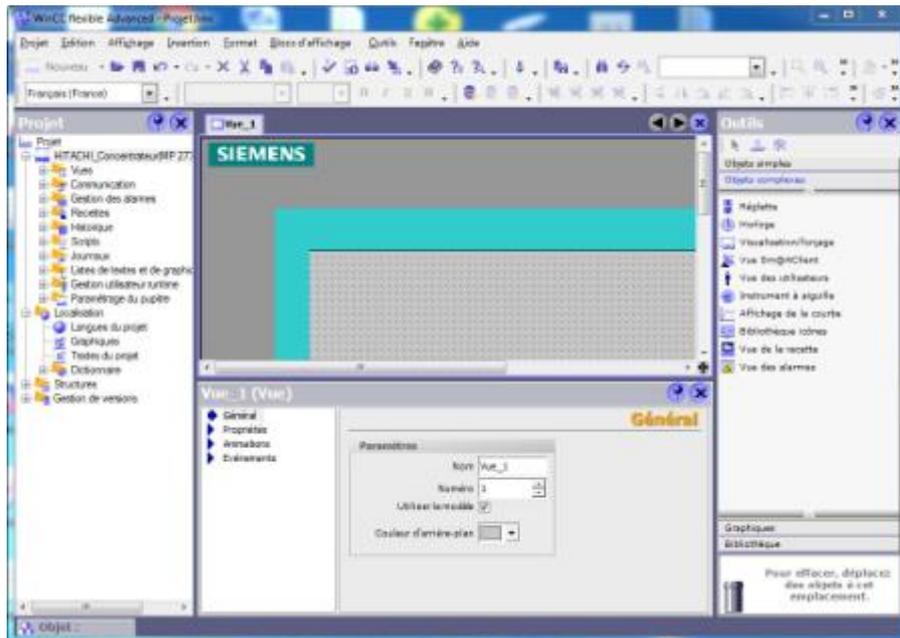


FIGURE II.20 : Aperçu d'une fenêtre WinCC

- **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 ;
- **Barre d'outils** : La barre d'outils permet d'afficher tout dont le programmeur à basion ;
- **Zone de travail** : La zone de travail sert à configurer des vues, de façon qu'il soit le plus compréhensible par l'utilisateur et très facile à manipuler et consulter les résultats ;
- **Boîte à outils** : La fenêtre des outils propose un choix simple ou complexe, par exemple des objets graphiques et élément de commande ;
- **Fenêtre des propriétés** : L'affichage des propriétés de l'objet sélectionné.

II.9.10.3. Intégration de WinCC flexible à STEP 7 :

Pour l'intégration d'un projet WinCC à STEP7 on procède comme suit :

- Ouvrir la configuration WinCC flexible ;
- Sélectionner le menu « Projet > Intégrer dans le projet STEP 7... » Le dialogue « Intégrer dans les projets STEP 7 » s'ouvre ;
- Sélectionner dans le dialogue le projet STEP 7 correspondant ;
- Après la sélection du projet STEP 7, l'intégration est exécutée.

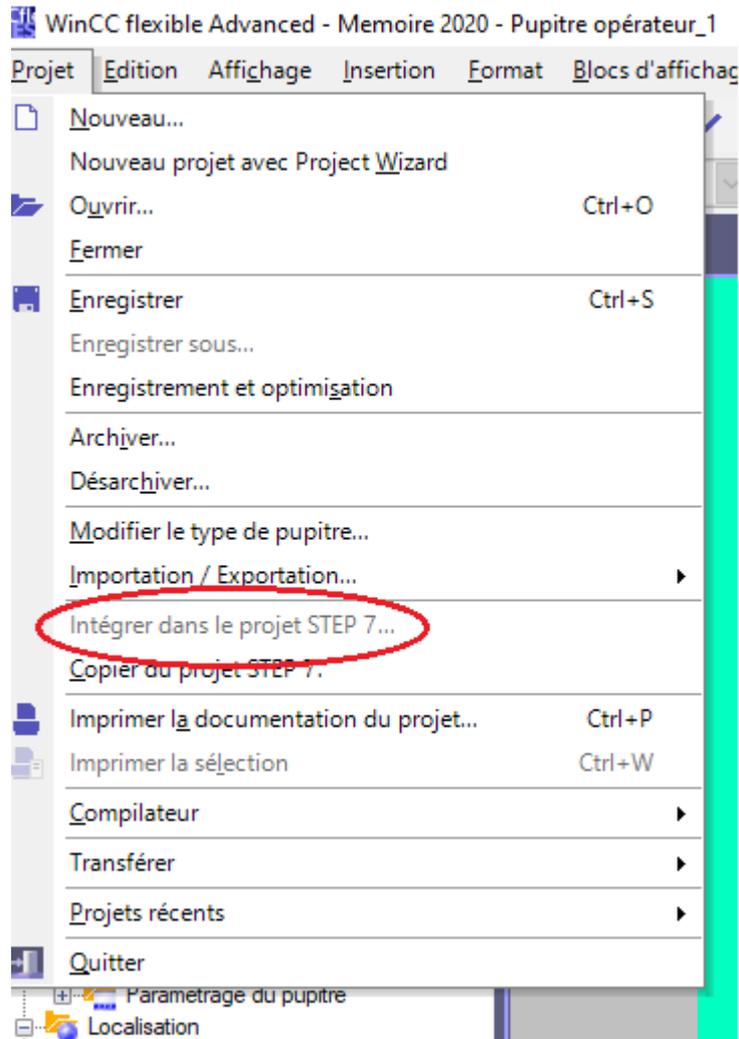


FIGURE II.21 : Intégration projet WinCC à STEP 7

II.10. Conclusion

La plupart des grands constructeurs d'automates programmables proposent une large gamme de produits pour l'automatisation industrielle, et fournissent des logiciels de configuration et de programmation munis des langages SFC, LD, FBD, ST et IL.

Cependant, le choix d'un langage s'appuie sur la complexité de l'application et de la tâche de commande.

PROFIBUS est une solution appropriée pour la connexion de l'automate à son environnement voisin, vue son architecture de communication ouverte et bien adaptée aux domaines d'application des réseaux locaux industriels. Il offre une mise en œuvre aisée et un paramétrage flexible, ce qui explique, en grande partie, sa grande utilisation dans l'industrie.

CHAPITRE III

Analyse Fonctionnelle

III.1. Introduction

Après les définitions des logiciels utilisés dans le chapitre précédent, Nous allons voir les différents blocs de programmation sous STEP 7. Notre objectif étant d'automatiser toute la ligne de production jusqu'aux trémies de réception après avoir présenté l'organigramme associé à cela.

III.2. Système à automatiser

Notre système est composé de trois parties principales qui sont :

- Trémie Tampon ;
- Sas de Transfert ;
- Trémies de Réception.

Ce système étant déjà existant notre rôle est de trouver une solution pour l'automatisation du processus tout en respectant son cahier de charges.

III.2.1. Table symboles

Voici un tableau qui décrit la représentation de chaque symbole du système de transfert de sucre jusqu'aux trémies.

TABLEAU III.1 : Description des symboles

Symboles	Description
S1	Bouton de sélection de la trémie une
S2	Bouton de sélection de la trémie deux
MR1004	Moteur dévouteur
PX1101	Capteur de proximité : ouverture de la vanne de remplissage
PX9101	Capteur de position : l'aiguillage est à la position de la trémie de réception 1
PX9102	Capteur de position : l'aiguillage est à la position de la trémie de réception 2
PX9106	Capteur de proximité : indique que l'actionneur est entré
PX9107	Capteur de proximité : indique que l'actionneur est déployé
PS3101	Capteur de pression : confirme que la vanne de la trémie de réception 1 est ouverte
PS3106	Capteur de pression : confirme que la vanne de la trémie de réception 2 est ouverte
PSH1102	Capteur de pression : indique la disponibilité de suffisamment d'air de transfert
PSH1202	Capteur de pression : indique le bon état des joints gonflants
PSH1103	Capteur de pression : indique la disponibilité de suffisamment d'air de commande
PSH1203	Capteur de pression : indique le bon état des joints
PSH9201	Capteur de pression : confirme le gonflage des joints
PSL1101	Capteur de pression : indique que la cuve n'est pas sous pression
LPH1101	Capteur de niveau bas de la trémie tampon : la présence de produit à amener
LPH1102	Capteur de niveau haut de la trémie de réception 1
LPH1103	Capteur de niveau haut de la trémie de réception 2
LSL3204	Capteur de niveau bas de la trémie de réception 1
LSL3209	Capteur de niveau bas de la trémie de réception 2
SOL1101	Electrovanne : excitée pour ouvrir les vannes d'air de transfert
SOL1102	Electrovanne : désexcitée pour la fermeture de la vanne de remplissage
SOL1103	Electrovanne : excitée pour le gonflage de la vanne de remplissage et la fermeture de la vanne de mise à l'air
SOL9101	Electrovanne : excité pour déployer l'actionneur
SOL9102	Electrovanne : désexcitée pour dégonfler les joints de l'aiguillage
XV3101	Electrovanne : désexcitée pour ouvrir la vanne de la trémie de réception 1
XV3105	Electrovanne : excitée pour ouvrir la vanne de la trémie de réception 1
XV3106	Electrovanne : désexcitée pour ouvrir la vanne de la trémie de réception 2
XV3110	Electrovanne : excitée pour ouvrir la vanne à guillotine de la trémie de réception 2

III.2.2. Rôle du cahier de charges

Le cahier des charges (CDC) est un document contractuel à respecter par le maître d'œuvre lors de la conception d'un projet. Il décrit précisément les besoins de ce dernier.

Il se constitue de plusieurs parties. Notre travail consiste à respecter un de ces parties qui est la partie fonctionnelle

III.2.3. Cahier de charges fonctionnel du système

Le respect des consignes du cahier de charges fonctionnel est obligatoire, ces derniers sont les suivantes :

- Sélectionner la trémie dans laquelle le produit sera déversé ;
- Le dévouteur MR1004 se met en fonctionnement jusqu'au signal du capteur de niveau de la trémie tampon LPH-1101 ;
- Dégonfler les joints de l'aiguillage en désexcitant l'électrovanne SOL-9102 ;
- L'électrovanne SOL-9101 est excitée pour déployer l'actionneur (le capteur de proximité PX-9107 indique que l'actionneur est déployé), Puis l'électrovanne SOL-9101 est désexcité pour faire entrer l'actionneur (le capteur de proximité PX-9106 indique que l'actionneur est entré). L'aiguillage s'indexe dans une position (si cette position ne correspond pas à la trémie choisie, il y a un nouveau cycle d'indexation) ;
- Lorsque le capteur de proximité PX-9106 indique que l'actionneur est entré et que le capteur de position PX-9101 indique que l'aiguillage est à la bonne position, l'électrovanne SOL-9102 est excitée pour gonfler les joints de l'aiguillage ;
- Le capteur de pression (PSH-9201) confirme le gonflage des joints ;
- L'électrovanne XV-3101 (XV-3106) est désexcitée pour ouvrir la vanne à manchon VM3101 (VM-3106) ;
- Après une courte temporisation et la confirmation que la vanne à manchon est ouverte par le capteur de pression PS-3101 (PS-3106), l'électrovanne SOL-1102 est excitée et la vanne de remplissage s'ouvre, permettant au produit d'entrer dans la cuve. (Le capteur de proximité PX-1101 de la vanne de remplissage se déclenche) ;
- L'électrovanne SOL-1103 est désexcitée pour dégonfler la vanne de remplissage et ouvrir la mise à l'air ;
- Une fois écoulé le temps de remplissage prédéfini (défini lors de la mise en service), l'électrovanne SOL-1102 est désexcitée et la vanne de remplissage se referme ;

- L'électrovanne SOL-1103 est excitée pour gonfler la vanne de remplissage et fermer la vanne de mise à l'air ;
- Après une courte temporisation, l'électrovanne SOL-1101 est excitée pour ouvrir les vannes d'air de transfert ;
- La pression monte dans la cuve en 3 à 5s, grâce à quoi le produit commence à sortir de la cuve pour entrer dans le circuit d'alimentation ;
- Il est prévu une temporisation de soufflage pour permettre à la pression de l'air de transfert de se stabiliser. Elle se règle généralement à 15s ;
- Une fois la temporisation de soufflage écoulée, l'électrovanne SOL-1101 est désexcitée pour fermer les vannes d'air de transfert, et une fois que le capteur de niveau bas de la trémie de réception signale la présence de suffisamment de produit, l'électrovanne XV-3105 (XV-3110) est excitée pour ouvrir la vanne à guillotine. L'ensachage peut commencer (CEVITAL).

III.3. Description de l'organigramme

Etape 01:

Ouvrir la vanne SOL-1103 pour le gonflage de la vanne de remplissage et la fermeture de la vanne de mise à l'air, puis on désexcite la vanne SOL-9102 pour dégonfler les joints de l'aiguillage on va désexciter l'électrovanne XV-3101 pour ouvrir la vanne à manchon (VM-3101) de la trémie de réception 1 puis on désexcite l'électrovanne XV-3106 pour ouvrir la vanne à manchon (VM-3106) de la trémie de réception 2, puis on va exciter l'électrovanne XV-3105 pour ouvrir la vanne de guillotine de la trémie de réception 1, finalement on excite l'électrovanne XV-3110 pour ouvrir la vanne à guillotine de la trémie de réception 2.

Après la première étape on aura deux exécutions du système, les étapes des deux exécutions sont montrées en parallèle, ici on va expliquer la partie gauche de l'organigramme puis la partie droite:

Partie Gauche

Condition 01.1:

On vérifie si le bouton S1 est appuyé, le capteur LPH1102 n'est pas activée, le capteur de pression PSH-9201 est activée, et confirmer si les capteurs PS-3101 et PS-3106 sont activées.

Etape 101:

L'exécution de cette étape nécessite le même traitement que l'étape précédente (Etape 01).

Condition 02.1:

Si le produit à amener est présent on saute directement vers l'étape 103, sinon on passe par l'étape 102.

Etape 102:

On met en marche le moteur de dévouteur plus les actions de l'étape précédente.

Condition 03.1:

On vérifie si le produit à amener est présent on passe à l'étape 103.

Etape 103 :

L'exécution de cette étape nécessite le même traitement que l'étape précédente mais sans la mise en marche le moteur de dévouteur.

Condition 04.1:

Si le capteur de proximité PX-9101 est activé on saute directement vers l'étape 108, sinon on passe vers l'étape 104.

Etape 104:

L'exécution de cette étape nécessite le même traitement que l'étape précédente sauf la désexcitation pour dégonfler les joints de l'aiguillage.

Condition 05.1:

On confirme si le capteur de position PX-9101 n'est pas activé.

Etape 105:

Dans cette étape l'exécution nécessite le même traitement que l'étape précédente mais cette fois on excite l'électrovanne SOL-9101 pour déployer l'actionneur.

Condition 06.1:

On confirme si le capteur de proximité PX-9107 est activé.

Etape 106:

Si l'actionneur est déployé les mêmes actions de l'étape 104 seront répétées dans cette étape.

Condition 07.1:

Si le capteur de proximité PX-9106 est activée et le capteur de position PX-9101 n'est pas activée on doit revenir (retour d'état) à l'étape 105. Et si le capteur de position PX-9101 est activé on passe à l'étape 107.

Etape 107:

Les actions de cette étape sont les même que les actions de l'étape 01.

Condition 08.1:

On confirme si le capteur de pression PSH-9201 est activé, si oui on passe à l'étape 108.

Etape 108:

Les actions de cette étape sont les même que les actions de l'étape précédente mais sans désexceptions de l'électrovanne XV3101.

Condition 09.1:

On confirme que le capteur de pression PS-3101 est activé.

Etape 109:

Les actions de cette étape sont les mêmes que les actions de l'étape précédente avec la désexcitation de l'électrovanne SOL-1102 pour la fermeture de la vanne de remplissage.

Condition 10.1:

On vérifie si le capteur de proximité PX-1101 est activé.

Etape 110:

Dans cette étape les changements effectués sur les actions par rapport à l'étape précédente c'est l'ajout d'une temporisation T1 de 5 seconds et l'absence de l'action d'excitation sur l'électrovanne SOL-1103.

Condition 11.1:

Une temporisation T1 de 5 secondes.

Etape 111:

Dans cette étape on a éliminé l'action d'excitation sur l'électrovanne SOL-1102, Et qu'il n'y a aucune temporisation.

Condition 12.1:

On vérifie si le capteur de proximité PX-1101 n'est pas activé.

Etape 112:

Cette étape c'est presque là même avec l'étape 110 mais avec une temporisation T2 de 2 seconds et avec l'action de l'excitation de l'électrovanne SOL-1103 au lieu de l'électrovanne SOL-1102.

Condition 13.1:

Une temporisation T2 de 2 secondes.

Etape 113:

Cette étape c'est presque pareil à l'étape précédente mais avec une temporisation T3 de 15 seconds plus l'action de l'excitation de l'électrovanne SOL-1101.

Condition 14.1:

Une temporisation T3 de 15 secondes.

Etape 114:

L'exécution de cette étape nécessite le même traitement que l'étape 112 mais avec aucune temporisation.

Après l'exécution de l'étape 114 on aura 4 conditions à tester et chaque condition mène vers une différente étape.

Condition 15.1:

- Si le Capteur LPH1102 n'est pas activé et on clique sur le bouton S1: on fait un retour d'état vers l'étape 101 ;
- Si le Capteur LPH1102 est activé et le bouton S1 est appuyé et le bouton S2 n'est pas appuyé : on fait un retour d'état vers l'étape 1 ;
- Si le bouton S1 n'est pas appuyé : on fait un retour d'état vers l'étape 1 ;
- Si le Capteur LPH1102 est activé et les boutons S1 et S2 sont appuyées et le capteur LPH1103 n'est pas activée : on fait un retour d'état vers l'étape 201.

On a terminé avec la partie gauche du grafcet, on passe à la deuxième partie

Partie Droite

Condition 01.2:

On vérifie si le bouton S1 n'est pas appuyé et le bouton S2 est appuyé, le capteur LPH1103 n'est pas activée, le capteur de pression PSH-9201 est activée, et confirmer si les capteurs PS-3101 et PS-3106 sont activées.

Etape 201:

L'exécution de cette étape nécessite le même traitement que l'étape précédente (Etape 01).

Condition 02.2:

Si le produit à amener est présent on saute directement vers l'étape 203, sinon on passe par l'étape 202.

Etape 202:

On met en marche le moteur de dévouteur plus les actions de l'étape précédente.

Condition 03.2:

On vérifie si le produit à amener est présent on passe à l'étape 203.

Etape 203:

L'exécution de cette étape nécessite le même traitement que l'étape précédente mais sans la mise en marche le moteur de dévouteur.

Condition 04.2:

Si le capteur de proximité PX-9102 est activé on saute directement vers l'étape 208, sinon on passe vers l'étape 204.

Etape 204:

L'exécution de cette étape nécessite le même traitement que l'étape précédente sauf la désexcitation pour dégonfler les joints de l'aiguillage.

Condition 05.2:

On confirme si le capteur PX-9101 n'est pas activé.

Etape 205:

Dans cette étape l'exécution nécessite le même traitement que l'étape précédente mais cette fois on excite l'électrovanne SOL-9101 pour déployer l'actionneur.

Condition 06.2:

On confirme si le capteur PX-9107 est activé.

Etape 206:

Si l'actionneur est déployé les mêmes actions de l'étape 204 seront répétées dans cette étape.

Condition 07.2:

Si le capteur de proximité PX-9106 est activée et le capteur de position PX-9102 n'est pas activée on doit revenir (retour d'état) à l'étape 205. Et si le capteur de position PX-9102 est activée on passe à l'étape 207.

Etape 207:

Les actions de cette étape sont les mêmes que les actions de l'étape 01.

Condition 08.2:

On vérifie si le capteur de pression PSH-9201 est activée, si oui on passe à l'étape 208.

Etape 208:

Les actions de cette étape sont les mêmes que les actions de l'étape précédente mais sans désexcitations de l'électrovanne XV3106.

Condition 09.2:

On confirme que le capteur de pression PS-3106 est activé.

Etape 209:

Les actions de cette étape sont les mêmes que les actions de l'étape précédente avec la désexcitation de l'électrovanne SOL-1102 pour la fermeture de la vanne de remplissage

Condition 10.2:

On vérifie si le capteur de proximité PX-1101 est activé.

Etape 210:

Dans cette étape les changements effectués sur les actions par rapport à l'étape précédente c'est l'ajout d'une temporisation T4 de 5 seconds et l'absence de l'action d'excitation sur l'électrovanne SOL-1103.

Condition 11.2:

Une temporisation T4 de 5 secondes.

Etape 211:

Dans cette étape on a éliminé l'action d'excitation sur l'électrovanne SOL-1102, Et qu'il n'y a aucune temporisation.

Condition 12.2:

On vérifie si le capteur de proximité PX-1101 n'est pas activé.

Etape 212:

Cette étape c'est presque là même avec l'étape 110 mais avec une temporisation T5 de 2 seconds et avec l'action de l'excitation de l'électrovanne SOL-1103 au lieu de l'électrovanne SOL-1102.

Condition 13.2:

Une temporisation T5 de 2 secondes.

Etape 213:

Cette étape c'est presque pareil à l'étape précédente mais avec une temporisation T6 de 15 seconds plus l'action de l'excitation de l'électrovanne SOL-1101.

Condition 14.2:

Une temporisation T6 de 15 secondes.

Etape 214:

L'exécution de cette étape nécessite le même traitement que l'étape 212 mais avec aucune temporisation.

Après l'exécution de l'étape 214 on aura 4 conditions à tester et chaque condition mène vers une autre étape.

Condition 15.2:

- Si le Capteur LPH1103 n'est pas activé et on clique sur le bouton S2 : on fait un retour d'état vers l'étape 201 ;
- Si le bouton S2 n'est pas appuyé : on fait un retour d'état vers l'étape 1 ;
- Si le Capteur LPH1103 est activé et le boutons S1 n'est pas appuyé et le bouton S2 est : appuyé : on fait un retour d'état vers l'étape 1 ;
- Si le Capteur LPH1103 est activé et le boutons S1 est appuyé et le bouton S2 n'est pas appuyé on passe à l'étape 215.

Etape 215:

Les actions de cette étape sont les mêmes que les actions de l'étape précédente plus l'action de désexcitations de l'électrovanne XV3106.

Après l'exécution de l'étape 214 on aura 3 conditions à tester et chaque condition mène vers une autre étape.

Condition 16.2:

- Si le bouton S1 ou S2 ne sont pas appuyées : on fait un retour d'état vers l'étape 1 ;
- Si le Capteur LPH1102 n'est pas activé et qu'on actionne le bouton S1 : on fait un retour d'état vers l'étape 101 ;
- Si le Capteur LPH1103 n'est pas activé, le bouton S2 est appuyé et le Capteur LPH1102 est activée : on fait un retour d'état vers l'étape 201.

III.3.1. Organigramme partie gauche

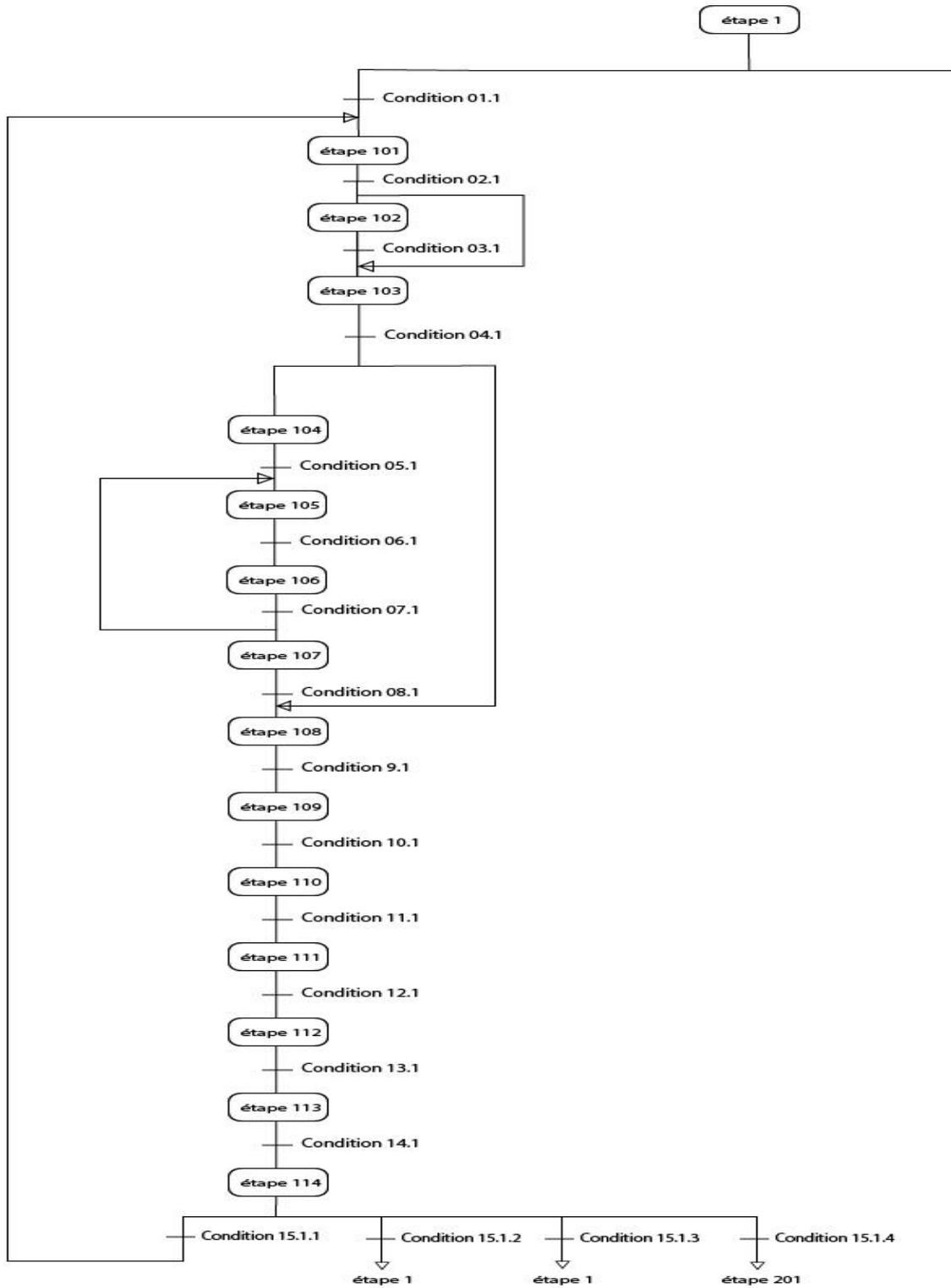


FIGURE III.1 : Partie gauche

III.3.2. Organigramme partie droite

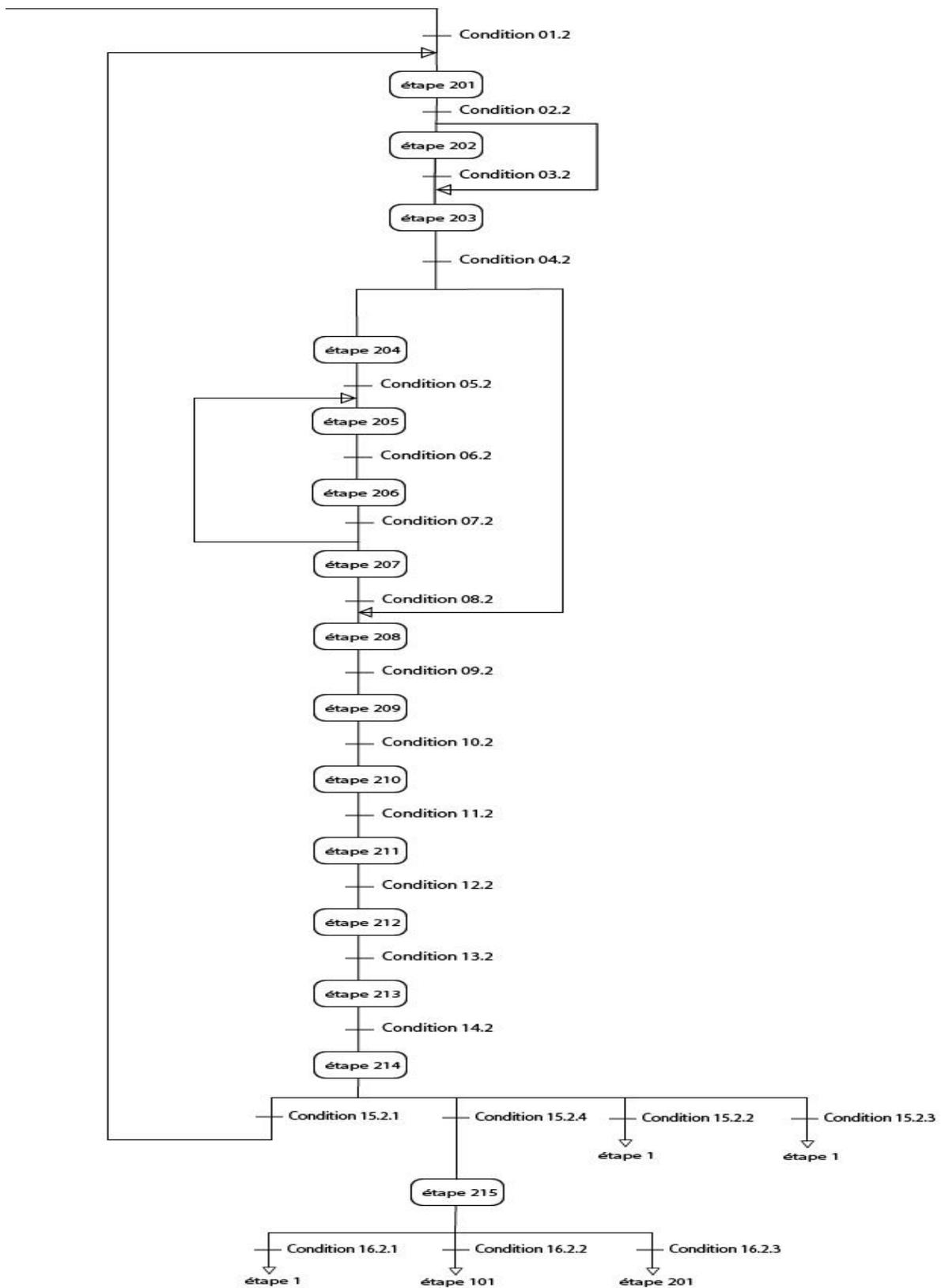


FIGURE III.2 : Partie droite

III.4. Programmation

Dans cette partie nous allons présenter le programme correspondant à l’organigramme élaboré pour gérer l’automatisation du transfert du sucre vers les trémies.

III.4.1. Table des mnémoniques

Lors de l’élaboration d’un programme nous avons besoin d’une table de variables à utiliser afin que le programme puisse reconnaître les différentes entrées/sorties utilisées.

III.4.1.1. Entrées

	Etat	Mnémonique	Opéran /	Type de do	Commentaire
3		condition 1.1	E 0.0	BOOL	Si S1 appuyé, LPH1102 non activée,PSH-9201, PS-3101 et PS-3106 activées.
4		condition 2.1	E 0.1	BOOL	On vérifie si le produit à amener est présent
5		condition 3.1	E 0.2	BOOL	On met en marche le moteur de dévouteur
6		condition 4.1	E 0.3	BOOL	On vérifie si le capteur de proximité PX-9101 est activé
7		condition 5.1	E 0.4	BOOL	On confirme si le capteur de position PX-9101 n'est pas activé.
8		condition 6.1	E 0.5	BOOL	On confirme si le capteur de proximité PX-9107 est activé.
9		condition 7.1	E 0.6	BOOL	Si PX-9106 est activée et PX-9101 n'est pas activée ou si PX-9101 est activé
10		condition 8.1	E 0.7	BOOL	On vérifie si le capteur de pression PSH-9201 est activé,
11		condition 9.1	E 1.0	BOOL	On confirme que le capteur de pression PS-3101 est activé.
12		condition 10.1	E 1.1	BOOL	On vérifie si le capteur de proximité PX-1101 est activé.
13		condition 11.1	E 1.2	BOOL	Une temporisation T1 de 5 secondes.
14		condition 12.1	E 1.3	BOOL	On vérifie si le capteur de proximité PX-1101 n'est pas activé.
15		condition 13.1	E 1.4	BOOL	Une temporisation T2 de 2 secondes.
16		condition 14.1	E 1.5	BOOL	Une temporisation T3 de 15 secondes.
17		condition 15.1.1	E 1.6	BOOL	On vérifie si le Capteur LPH1102 n'est pas activé et on clique sur le bouton S1
18		condition 15.1.2	E 1.7	BOOL	On vérifie si LPH1102 est activé et S1 est appuyé et S2 n'est pas appuyé
19		condition 15.1.3	E 2.0	BOOL	On vérifie si le bouton S1 n'est pas appuyé
20		condition 15.1.4	E 2.1	BOOL	Si LPH1102 est activé et S1 et S2 sont appuyées et LPH1103 n'est pas activée
21		condition 1.2	E 2.2	BOOL	Si S2 est appuyé et S2 non,PSH-9201 PS-3101,PS-3106 sont activées et LPH1103 non
22		condition 2.2	E 2.3	BOOL	On vérifie si le produit à amener est présent
23		condition 3.2	E 2.4	BOOL	On vérifie si le produit à amener est présent
24		condition 4.2	E 2.5	BOOL	On vérifie si le capteur de proximité PX-9102 est
25		condition 5.2	E 2.6	BOOL	On confirme si le capteur PX-9101 n'est pas activé.
26		condition 6.2	E 2.7	BOOL	On confirme si le capteur PX-9107 est activé.
27		condition 7.2	E 3.0	BOOL	Si PX-9106 est activée et PX-9102 n'est pas activée ou si PX-9102 est activé
28		condition 8.2	E 3.1	BOOL	On vérifie si le capteur de pression PSH-9201 est activée.
29		condition 9.2	E 3.2	BOOL	On confirme que le capteur de pression PS-3106 est activé.
30		condition 10.2	E 3.3	BOOL	On vérifie si le capteur de proximité PX-1101 est activé.
31		condition 11.2	E 3.4	BOOL	Une temporisation T4 de 5 secondes.
32		condition 12.2	E 3.5	BOOL	On vérifie si le capteur de proximité PX-1101 n'est pas activé.
33		condition 13.2	E 3.6	BOOL	Une temporisation T5 de 2 secondes.
34		condition 14.2	E 3.7	BOOL	Une temporisation T6 de 15 secondes.
35		condition 15.2.1	E 4.0	BOOL	On vérifie si le Capteur LPH1103 n'est pas activé et on clique sur le bouton
36		condition 15.2.2	E 4.1	BOOL	On vérifie si le bouton S2 n'est pas appuyé
37		condition 15.2.3	E 4.2	BOOL	On vérifie si LPH1103 est activé et S1 n'est pas appuyé et S2 est appuyé
38		condition 15.2.4	E 4.3	BOOL	On vérifie si LPH1103 est activé et S1 est appuyé et S2 n'est pas appuyé
39		condition 16.2.1	E 4.4	BOOL	On confirme si le bouton S1 ou S2 ne sont pas appuyées
40		condition 16.2.2	E 4.5	BOOL	On vérifie si le Capteur LPH1102 n'est pas activé et qu'on actionne le bouton S1
41		condition 16.2.3	E 4.6	BOOL	On confirme si LPH1103 n'est pas activé, S2 est appuyé et LPH1102 est activée

FIGURE III.3 : Table des conditions

III.4.1.2. Mémentos

40		étape 0	M	0.0	BOOL	
41		étape 1	M	0.1	BOOL	
42		étape 101	M	0.2	BOOL	
43		étape 102	M	0.3	BOOL	
44		étape 103	M	0.4	BOOL	
45		étape 104	M	0.5	BOOL	
46		étape 105	M	0.6	BOOL	
47		étape 106	M	0.7	BOOL	
48		étape 107	M	1.0	BOOL	
49		étape 108	M	1.1	BOOL	
50		étape 109	M	1.2	BOOL	
51		étape 110	M	1.3	BOOL	
52		étape 111	M	1.4	BOOL	
53		étape 112	M	1.5	BOOL	
54		étape 113	M	1.6	BOOL	
55		étape 114	M	1.7	BOOL	
56		étape 201	M	2.0	BOOL	
57		étape 202	M	2.1	BOOL	
58		étape 203	M	2.2	BOOL	
59		étape 204	M	2.3	BOOL	
60		étape 205	M	2.4	BOOL	
61		étape 206	M	2.5	BOOL	
62		étape 207	M	2.6	BOOL	
63		étape 208	M	2.7	BOOL	
64		étape 209	M	3.0	BOOL	
65		étape 210	M	3.1	BOOL	
66		étape 211	M	3.2	BOOL	
67		étape 212	M	3.3	BOOL	
68		étape 213	M	3.4	BOOL	
69		étape 214	M	3.5	BOOL	
70		étape 215	M	3.6	BOOL	

FIGURE III.4 : Table des étapes

III.4.2. Création du bloc OB [4]

Le bloc d'organisation (**OB1**) est généré automatiquement lors de la création d'un projet, il constitue l'interface entre le système d'exploitation et le programme qu'on a élaboré.

III.4.3. Création du bloc FC [4]

Le bloc (**FC**) contient des routines pour les fonctions fréquemment utilisées. Il est sans mémoire et sauvegarde ses variables temporaires dans la pile de données locales. Cependant il peut faire appel à des blocs de données globaux pour la sauvegarde de ses données.

Notre programme est élaboré via la fonction (**FC**), puis nous avons effectué un appel de la fonction dans le bloc (**OB1**).

Notre programme se constitue de plusieurs réseaux, la figure suivante représente la structure des réseaux utilisés, le reste du programme se trouve dans l'annexe 02.

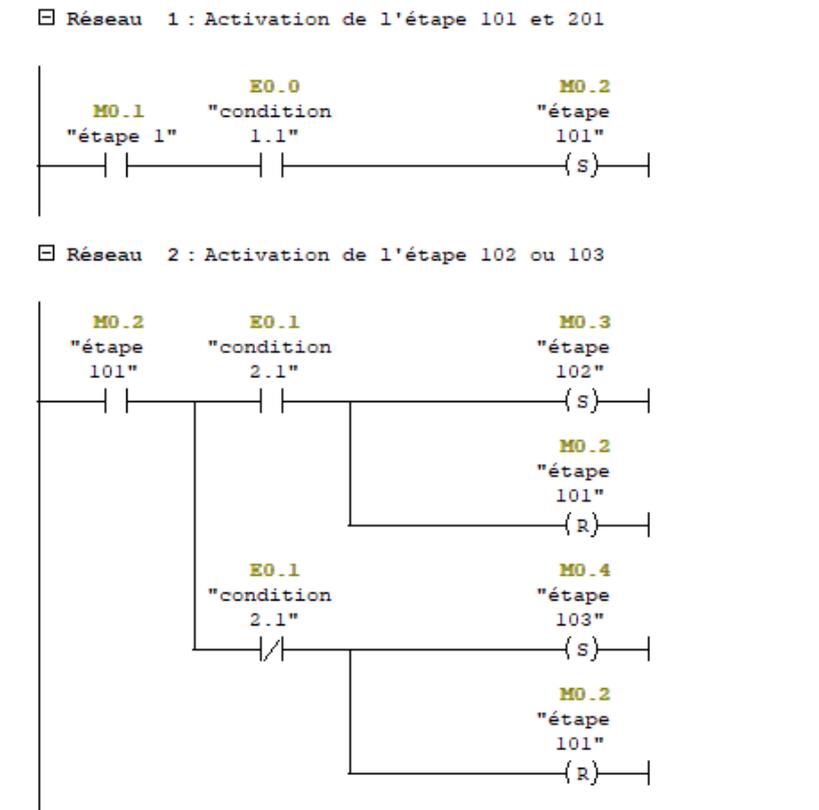


FIGURE III.5 : Réseau d'activation des étapes 101-103

Afin de varier le niveau de la cuve pour notre simulation nous avons utilisé le bloc « Scale FC105 » qui nous permet de prendre le contrôle sur le niveau du produit contenu à l'intérieur.

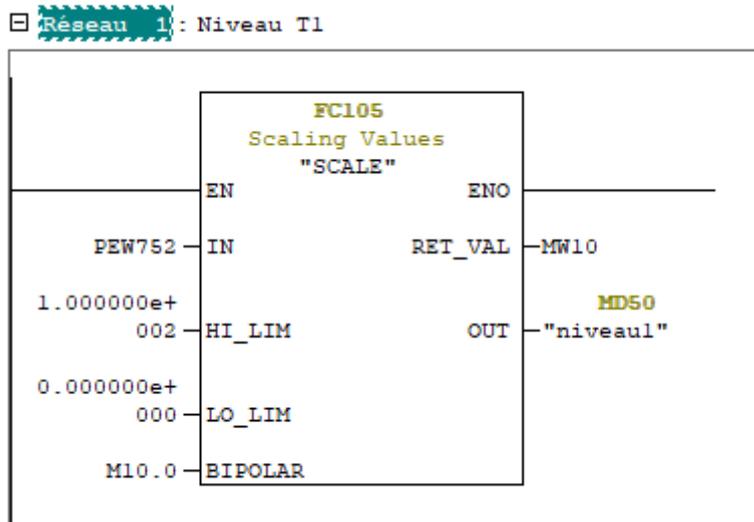


FIGURE III.6 : Réseau de variation du niveau de la Trémie 1

Afin de commander le degré d'ouverture de la vanne on a utilisé le bloc « Unscale FC106 » qui nous permet d'entrer le pourcentage désiré pour l'ouverture de l'électrovanne.

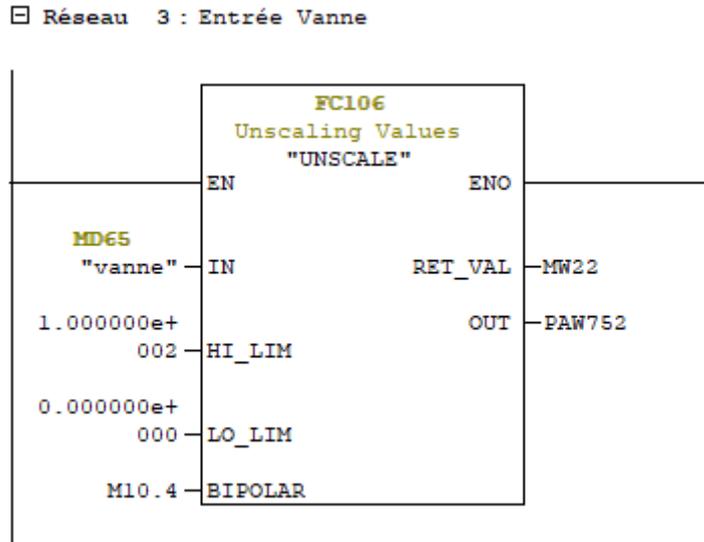


FIGURE III.7 : Réseau de commande d'ouverture de la vanne

Afin de montrer que le produit est en phase de transfert dans la conduite lors de la simulation du programme, nous avons utilisé deux blocs « CMP » relié à une bascule « SR ».

Les conditions suivantes sont respectées :

- Vanne ouverte => conduite allumée ;
- Vanne fermée => conduite éteinte.

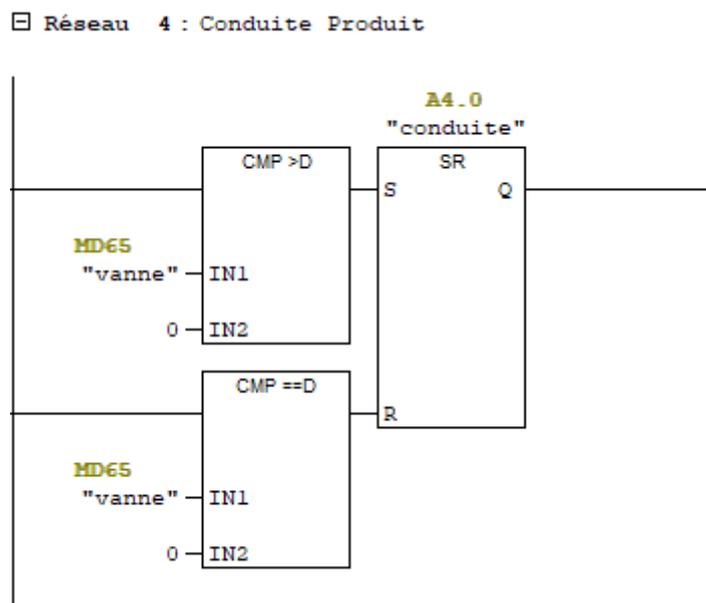


FIGURE III.8 : Réseau de conduite du produit

Pour le déclenchement de l'alarme de pression de cuve, une bascule « SR » est utilisée, la sortie de celle-ci est reliée à notre interface de supervision afin de s'assurer que le niveau de pression de notre cuve est respecté avant l'arrivée du produit.

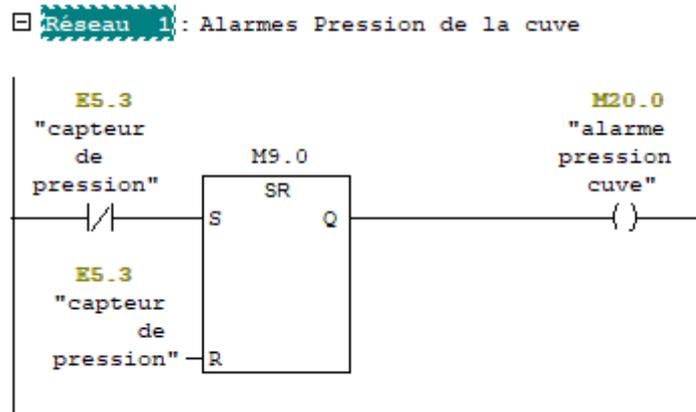


FIGURE III.9 : Réseau d'alarme de pression cuve

III.5. Conclusion

Dans ce chapitre nous avons élaboré l'organigramme correspondant à notre cahier de charges. Nous avons présenté ainsi, les étapes nécessaires à l'élaboration du programme dont nous avons illustré par quelques réseaux. Ce programme permettra de piloter le système de transfert du sucre vers les trémies. L'interface de supervision qui sera reliée à ce software sera donnée au chapitre suivant.

CHAPITRE IV

Supervision

Du Système

IV.1. Introduction

La supervision est une étape importante dans le fonctionnement d'un système industriel. Elle permet via une interface (IHM) de contrôler, commander et de diagnostiquer en temps réel le processus.

Dans ce chapitre, nous allons élaborer les différentes vues de notre interface de commande et de supervision. Pour pouvoir faire fonctionner cette interface en faisant appel au programme réalisé, il faudra faire appel aux variables (mnémoniques) qui représentent les états et les actions du système en réalisant une fonction appelée 'intégration'

Les résultats de la simulation seront donnés introduisant quelques valeurs des niveaux aux trémies pour pouvoir suivre le transfert du sucre vers les trémies.

IV.2. Création du projet

La création du projet passe par le choix du pupitre (FIGURE IV.1), dans notre cas on a choisi la « MP 270 10 pouce touch ».

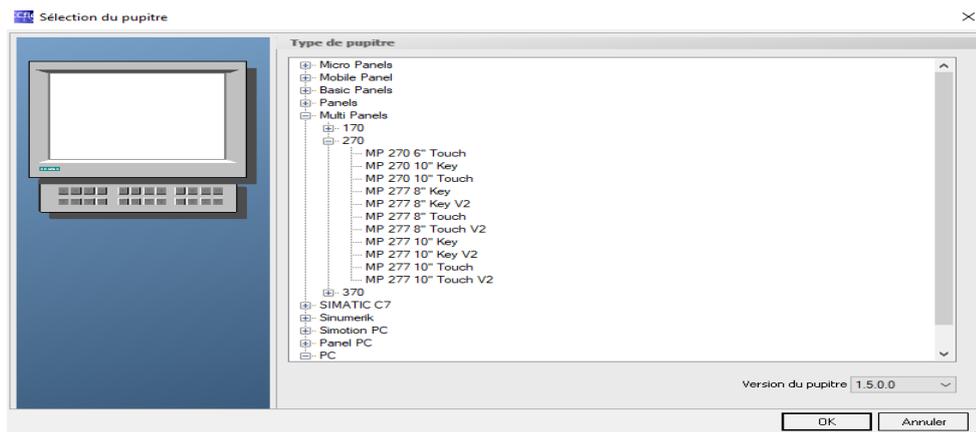


FIGURE IV.1. : Fenêtre de sélection d'un pupitre

Ensuite, il faut passer à la configuration du projet créé.

- Créer une nouvelle vue et naviguer entre les vues déjà existantes ;
- Editer la table des variables ;
- Configurer les alarmes analogiques et TOR ;
- Modifier les paramètres du pupitre ;
- Changer la langue du projet ;
- Ajouter différentes polices d'écriture.

IV.3. Table des variables

Il faut intégrer notre pupitre à un projet STEP 7 Afin d’avoir une supervision dynamique, et pouvoir partager les mêmes variables utilisées dans notre programme STEP 7.

Nom	Connexion	Type de données	Mnémonique	Adresse	Éléments du ta...	Cycle d'acqui...	Commentaire
alarme	Liaison_2	Word	alarme	MW 25	1	100 ms	
alarme pression	Liaison_2	Word	alarme pression	MW 20	1	100 ms	
alarme pressio...	Liaison_2	Bool	alarme pression cuve	M 20.0	1	100 ms	
arrêtUrgence	Liaison_2	Bool	arrêtUrgence	I 5.1	1	100 ms	
arrêtManuel	Liaison_2	Bool	arrêtManuel	M 9.2	1	100 ms	
arrêtManuel_0	Liaison_2	Bool	arrêtManuel	M 9.2	1	100 ms	
arrêtUrgence	Liaison_2	Bool	arrêtUrgence	M 25.0	1	100 ms	
capteur de pre...	Liaison_2	Bool	capteur de pression	I 5.3	1	100 ms	
comutateur	Liaison_2	Bool	comutateur	M 9.1	1	100 ms	
conduite	Liaison_2	Bool	conduite	Q 4.0	1	100 ms	
marche	Liaison_2	Bool	marche	M 9.3	1	100 ms	
marcheManuel	Liaison_2	Bool	marcheManuel	I 5.0	1	100 ms	
moteur de dev...	Liaison_2	Bool	moteur de devouteur	Q 0.0	1	100 ms	
moteur 1	Liaison_2	Bool	moteur1	M 9.4	1	100 ms	
niveau1	Liaison_2	Real	niveau1	MD 50	1	100 ms	
niveau2	Liaison_2	Real	niveau2	MD 55	1	100 ms	
niveau3	Liaison_2	Real	niveau3	MD 60	1	100 ms	
ouverture	Liaison_2	Bool	ouverture	I 5.2	1	100 ms	
vanne	Liaison_2	Real	vanne	MD 65	1	100 ms	

FIGURE IV.2. : Table des variables WinCC

IV.4. Présentation des vues du système

Nous avons créé trois vues pour notre projet.

- Vue d’accueil (FIGURE IV.3) ;
- Vue du processus (FIGURE IV.4) ;
- Vue des alarmes. (FIGURE IV.5).



FIGURE IV.3 : Vue d’accueil

La supervision s'effectue sur la vue du processus là où les différentes cuves, les actionneurs et capteurs de notre système sont représentés.

La LED devant le moteur dévouteur nous informe sur l'état de marche de notre système, après l'appui sur « Marche » la LED tourne au vert et le produit commence à circuler vers la Trémie Tampon qui est équipé d'un capteur niveau bas pour vérifier la présence du produit à l'intérieur, d'un capteur de pression pour s'assurer que les conditions de pressions sont respectées et enfin d'une électrovanne d'ont l'ouverture est commandé via un pourcentage voulu.

Les Trémies de réception (T1 et T2) ont deux capteurs de niveau (haut et bas), niveau bas nous indique la présence du produit à l'intérieur de la trémie, le niveau haut nous indique le seuil de remplissage de la trémie.

Le commutateur nous permet de sélectionner la trémie à remplir entre T1 et T2.

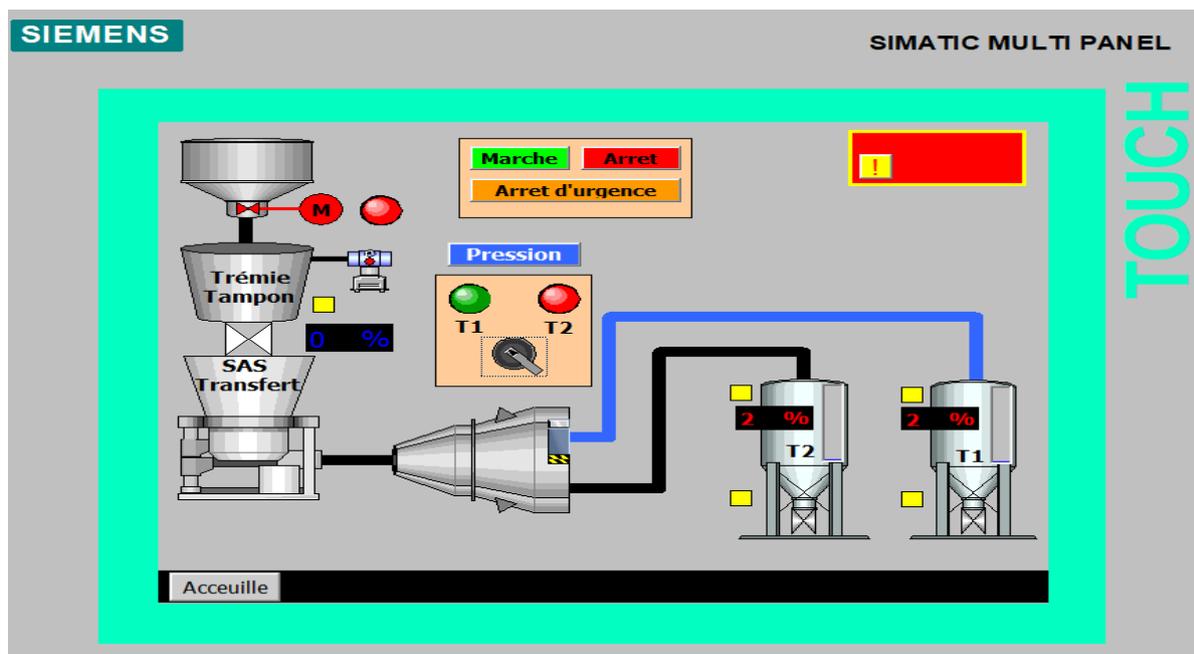


FIGURE IV.4 : Vue du processus

Un témoin clignotant s'affiche indiquant le déclenchement d'une alarme, on accède à la vue d'alarme via le bouton d'exclamation en jaune ou via le bouton « alarmes » situé sur la vue « accueil ».

La vue d'alarme nous indique le type d'alarme déclenché et nous permet son acquittement via un bouton, une fois le problème réglé on actionne le bouton et la signalisation d'alarme disparaît.

Une fois l'alarme acquitté on retourne à la vue précédente via le bouton « Back ».

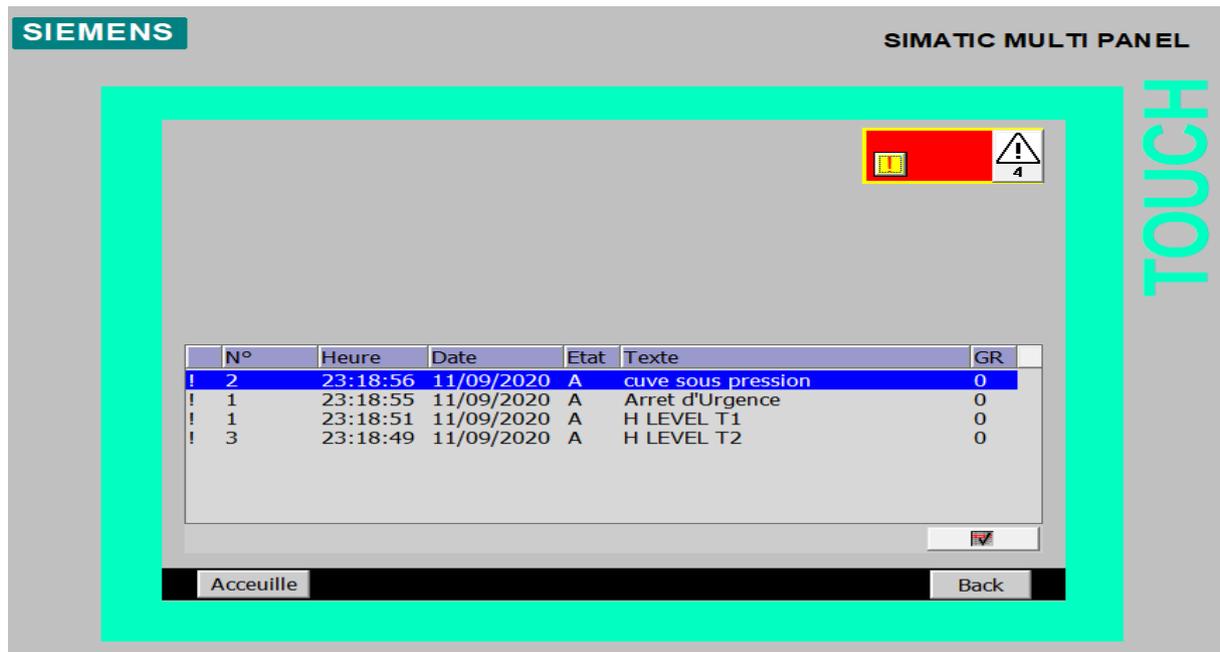


FIGURE IV.5 : Vue d'alarmes

IV.5. Conclusion

Dans ce chapitre nous avons présenté les vues de notre interface de la supervision élaborées afin de montrer le cycle d'exécution de la ligne de production de sucre et son transfert vers les trémies. Les résultats de simulation ont montré que l'intégration vers le programme a pu se faire convenablement et le projet est prêt à être implémenté sur un automate et une interface au niveau de l'entreprise.

Conclusion

Générale

Conclusion générale

L'objectif de ce travail était l'automatisation d'alimentation de trémies d'une ligne de production de sucre, en utilisant la programmation sous STEP 7, et de superviser ensuite le programme avec le WinCC flexible, et pour cela nous avons choisis un automate programmable Siemens S7-300, à causes des possibilités de simulation et de supervision qu'il offre.

Au début on a présenté l'entreprise CEVITAL, puis on a identifié les équipements et les différents appareils et matériels constituant l'installation, puis on a donné des généralités sur les systèmes automatisés et les API et aussi on a pris le temps de présenter les logiciels utilisés (STEP 7 et WinCC) tout en expliquant leurs fonctionnements, en débutant par la création du projet et en passant à la configuration et en finalisant par la simulation.

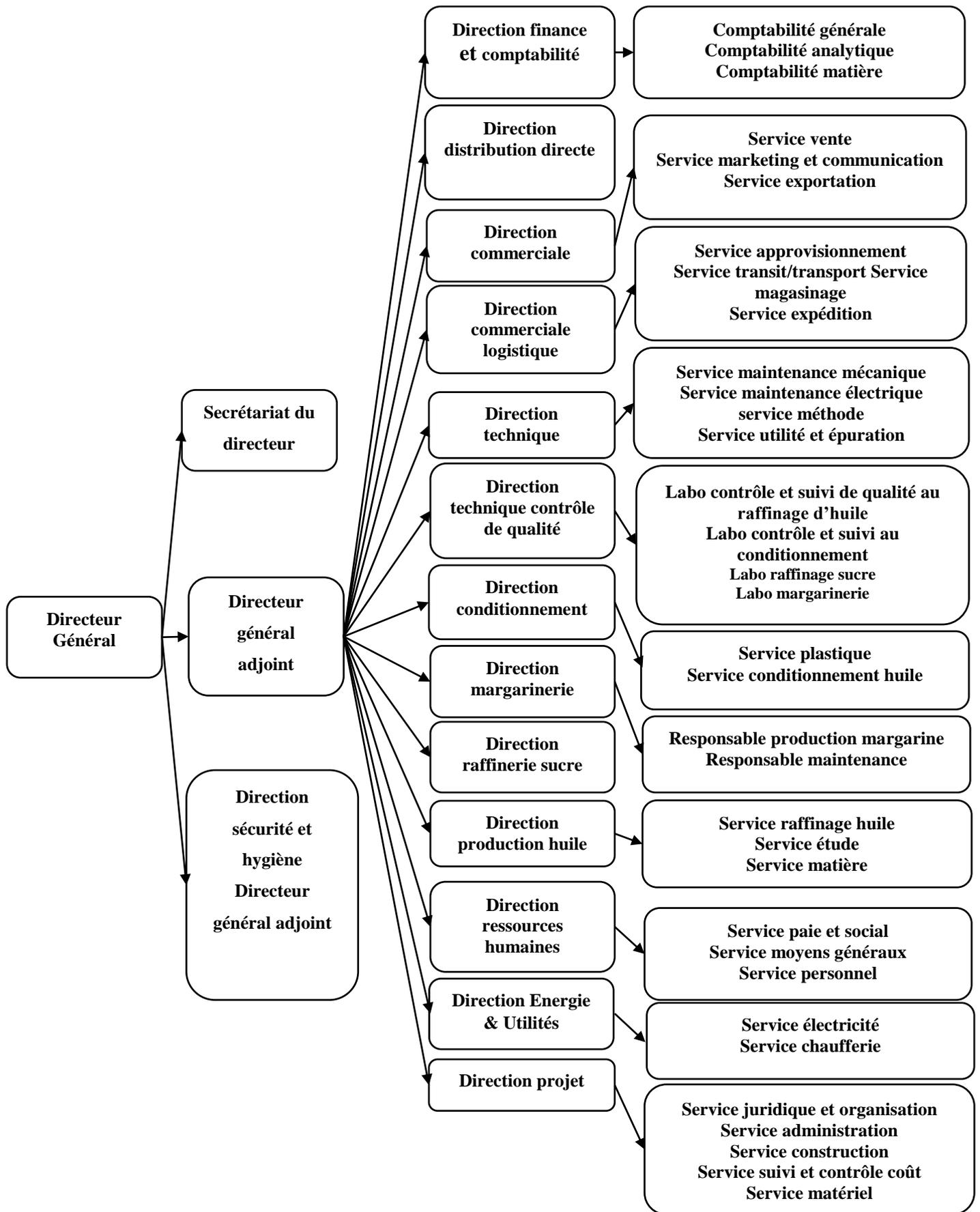
Et par la suite, on a programmé notre système après avoir présenté son cahier de charge et bien expliqué son fonctionnement, à la fin on a supervisé notre programme sur le logiciel WinCC afin de connaître l'état d'avancement du processus en temps réel et d'intervenir directement sur le pupitre de commande depuis la salle de contrôle.

Ce travail nous a permis d'acquérir une expérience personnelle et professionnelle très bénéfique. Ce fut une occasion pour se familiariser avec l'environnement du travail de la vie professionnelle, d'élargir et d'approfondir dans les connaissances et les apprécier aux diverses réalités du terrain.

Bibliographie

- [1] Documentation Cevital.
- [2] Les Automates Programmables Industriels, Alain GONZAGA.
- [3] WinCC : informations générale et installation, 02/2017.
- [4] Manuel Siemens, Programmation avec STEP7, 2000.
- [5] SIMATIC S7-300 Système d'automatisation S7-300 Caractéristiques des modules, Version : 06/2017
- [6] Siemens, SIMATIC S7 PLCSIM V5.4 Guide de l'utilisateur.
- [7] Documentation du logiciel SIMATIC Step7, version 5.6, Siemens.
- [8] André SIMON, Automates programmables, programmation, automatisation, et logique programmée, Edition l'ELAN, 1983.
- [9] M. BERTRAND « Automates Programmables Industriels ». Technique de l'Ingénieur S8015, 2010.
- [10] Manuel SIEMENS, SIMATIC HMI Win CC flexible 2008 Runtime, Manuel D'utilisation, édition Siemens AG, 2007.
- [11] SIMATIC STEP 7 Basic/Professional V16 et SIMATIC WinCC V16, Version : 11/2019
- [12] Formation Totally integrated Automation (TIA)''Programmation d'automates avec le STEP 7, Commandes de programmation de base en CONT /LIST/LOG de STEP 7 et Simulation d'automates avec S7-PLCSIM, 02/2001.
- [13] http://philippe.berger2.free.fr/automatique/cours/cpt/les_capteurs.htm , consulté le 15 aout 2020.
- [14] <http://support.automation.siemens.com> , consulté le 10 mai 2020.
- [15] <https://www.omega.fr/prodinfo/mesure-de-niveau.html> , consulté le 07 mai 2020.
- [16]<https://www.techniques-ingenieur.fr/base-documentaire/automatique-robotique-th16/supervision-des-systemes-industriels-42396210/langages-de-programmation-pour-systemes-automatisees-norme-cei-61131-3-s8030/>, consulté le 20 juillet 2020

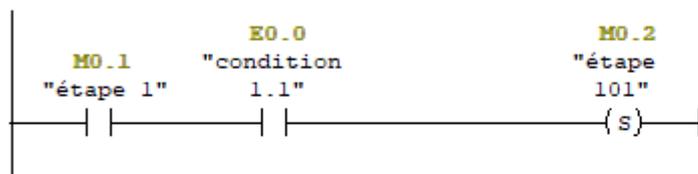
Annexes



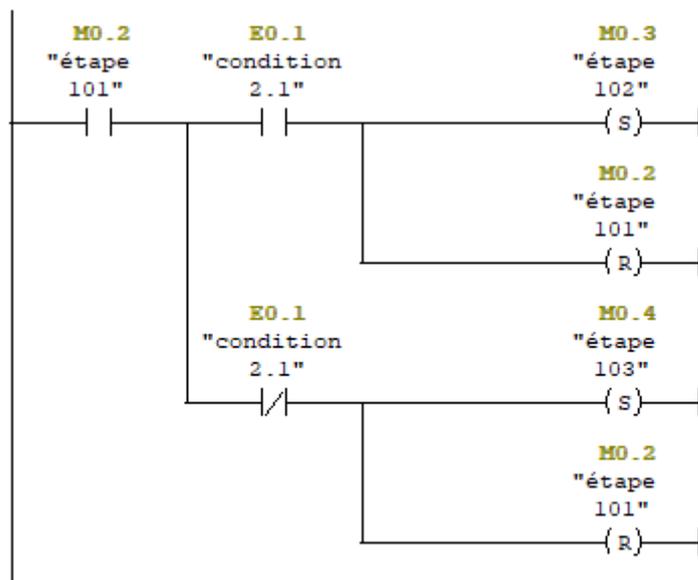
Organigramme du complexe CEVITAL [1]

PROGRAMME

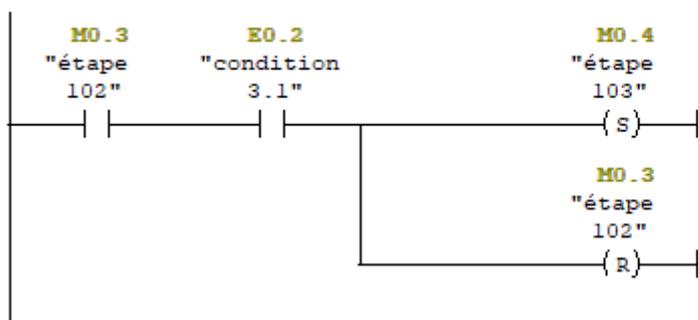
☐ Réseau 1 : Activation de l'étape 101 et 201



☐ Réseau 2 : Activation de l'étape 102 ou 103

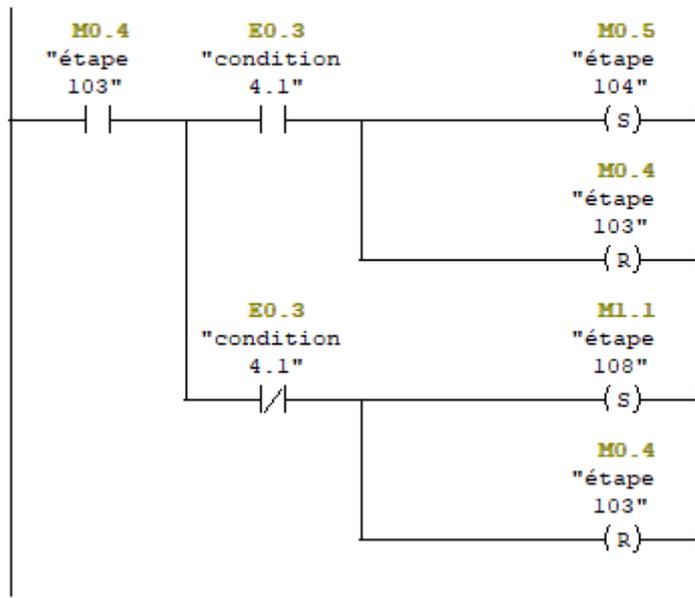


☐ Réseau 3 : Activation de l'étape 103

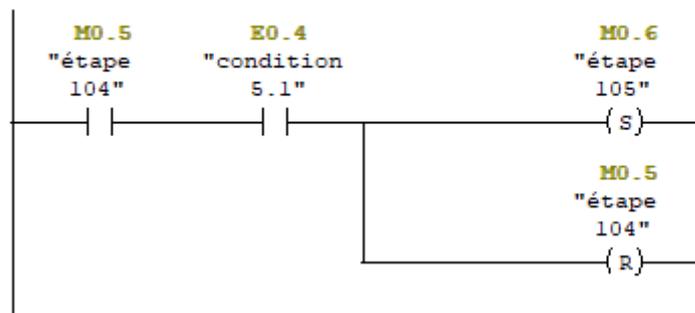


Annexe 02

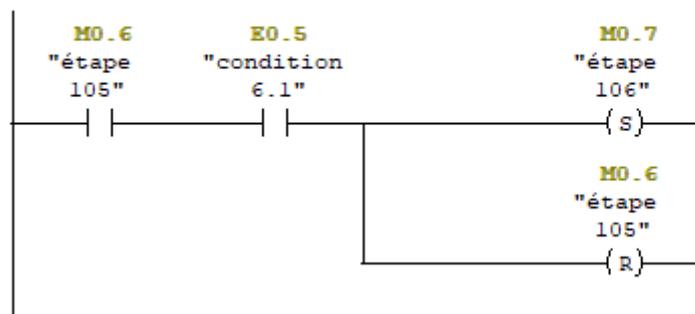
▣ Réseau 4 : Activation de l'étape 104



▣ Réseau 5 : Activation de l'étape 105

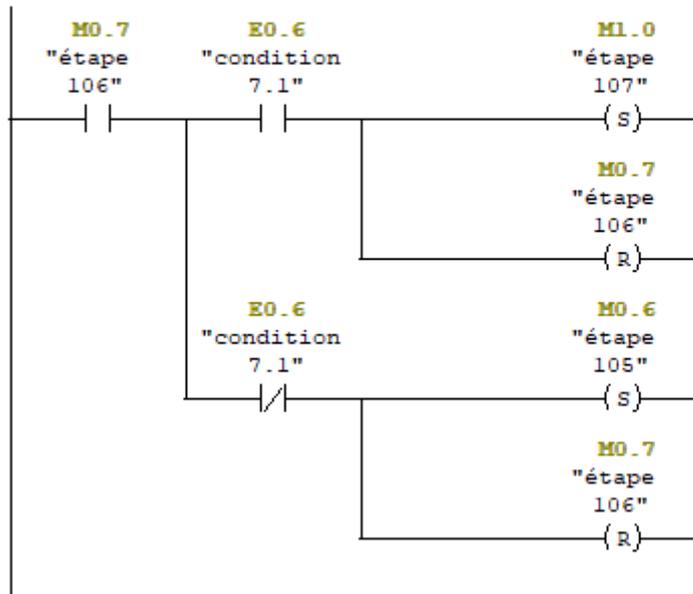


▣ Réseau 6 : Activation de l'étape 106

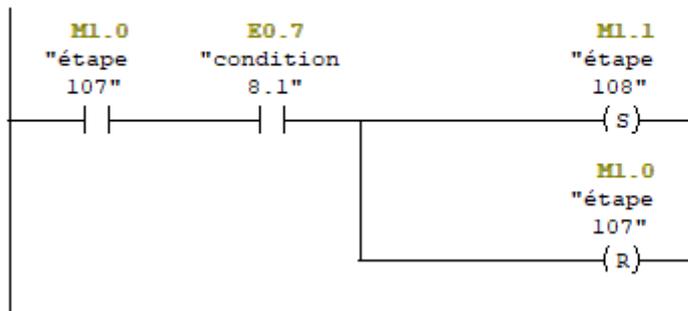


Annexe 02

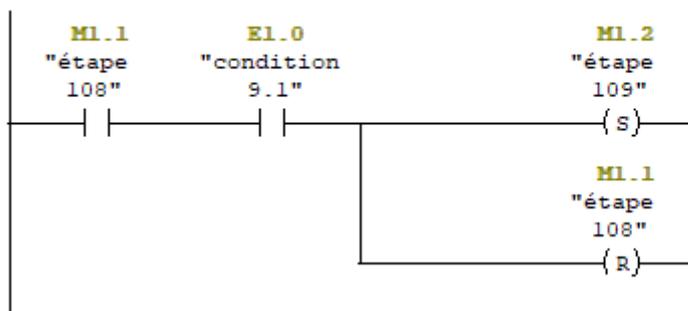
▣ Réseau 7 : Activation de l'étape 107



▣ Réseau 8 : Activation de l'étape 108

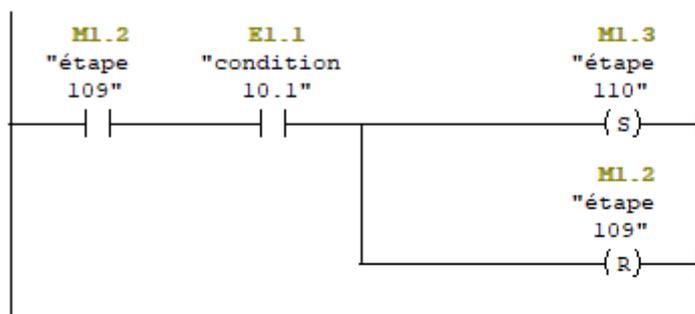


▣ Réseau 9 : Activation de l'étape 109

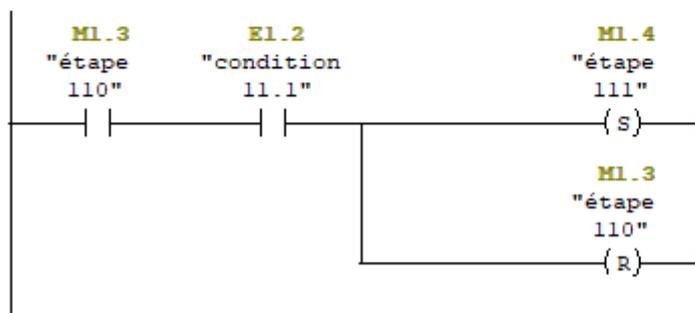


Annexe 02

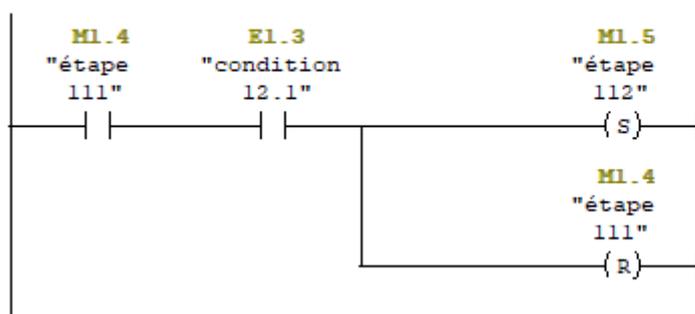
☐ Réseau 10 : Activation de l'étape 110



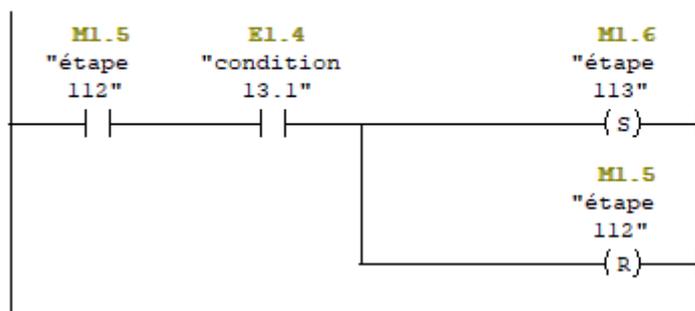
☐ Réseau 11 : Activation de l'étape 111



☐ Réseau 12 : Activation de l'étape 112

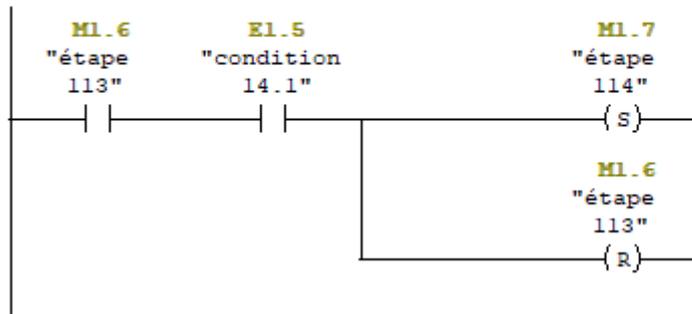


☐ Réseau 13 : Activation de l'étape 113

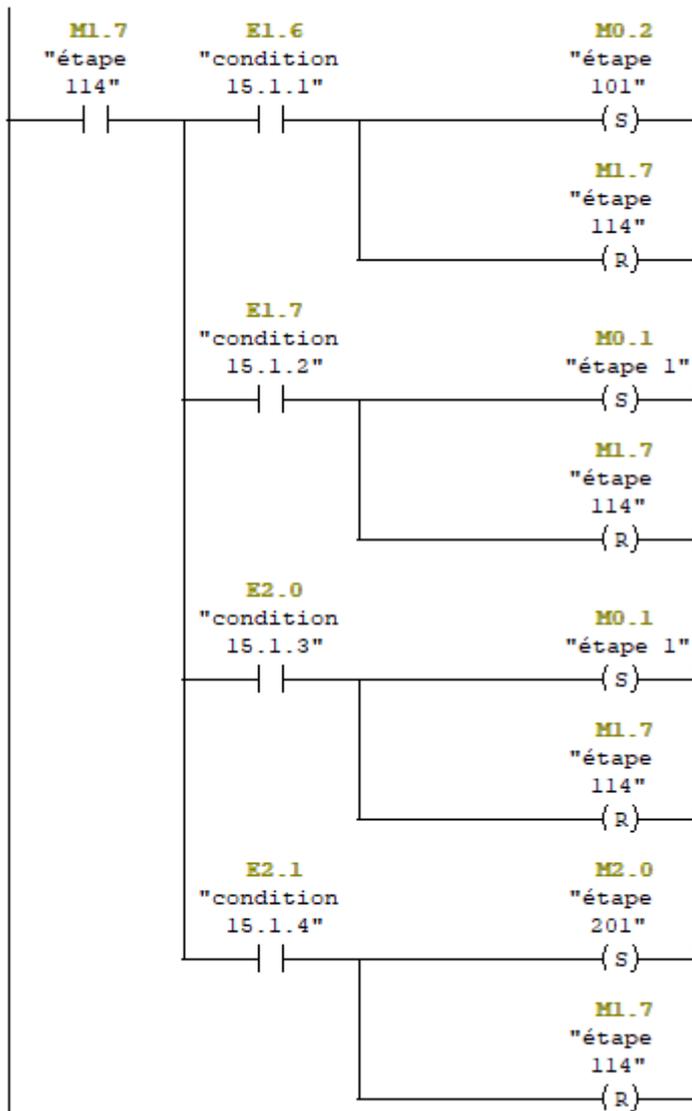


Annexe 02

▣ Réseau 14 : Activation de l'étape 114

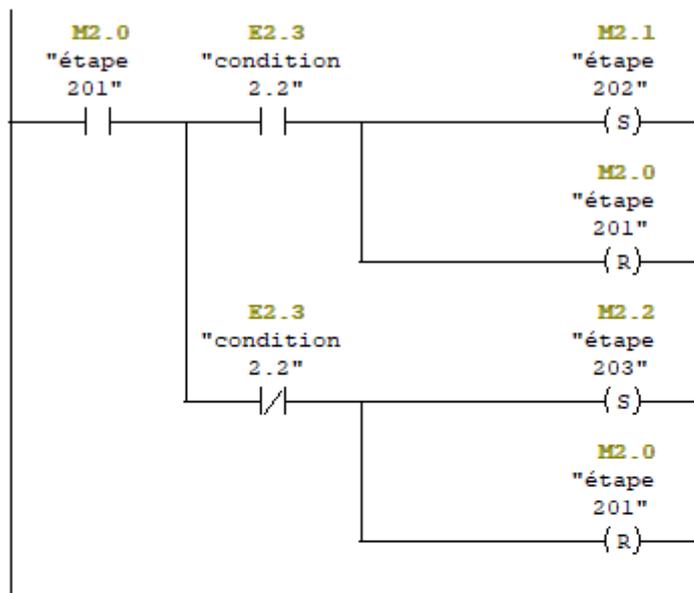


▣ Réseau 15 : Activation de l'étape 1 ou 201 à partir de l'étape 114

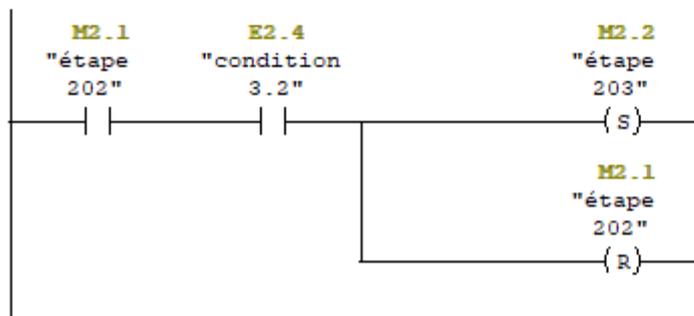


Annexe 02

▣ Réseau 16 : Activation de l'étape 202

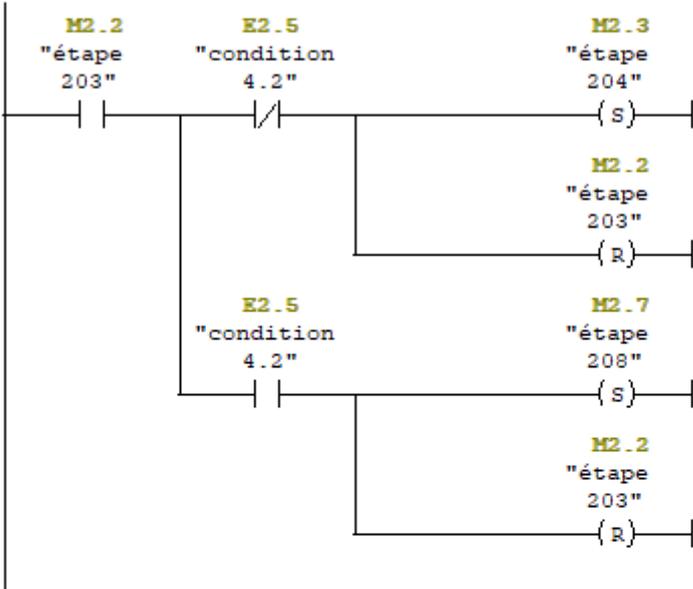


▣ Réseau 17 : Activation de l'étape 203

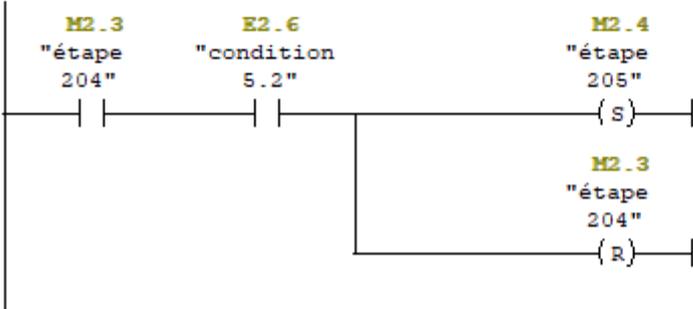


Annexe 02

☐ Réseau 18 : Activation de l'étape 204

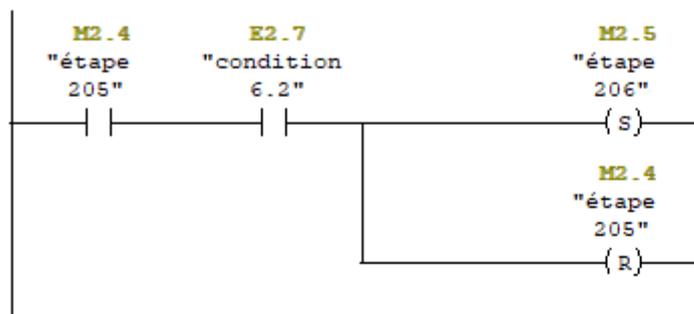


☐ Réseau 19 : Activation de l'étape 205

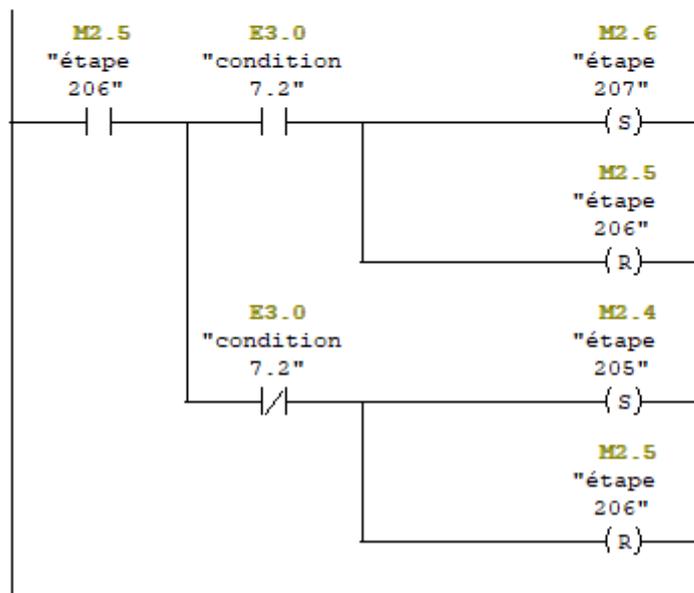


Annexe 02

☐ Réseau 20 : Activation de l'étape 206

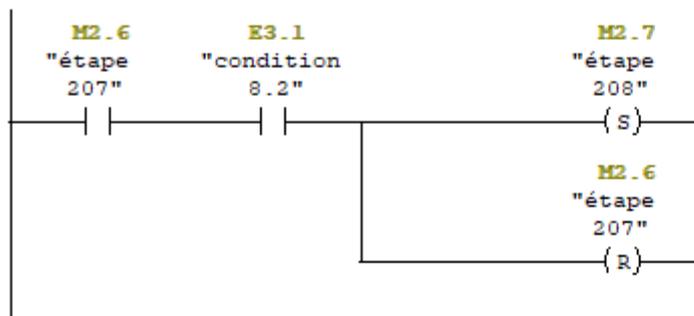


☐ Réseau 21 : Activation de l'étape 207

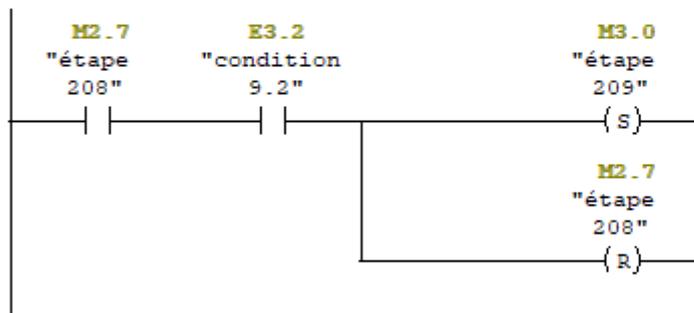


Annexe 02

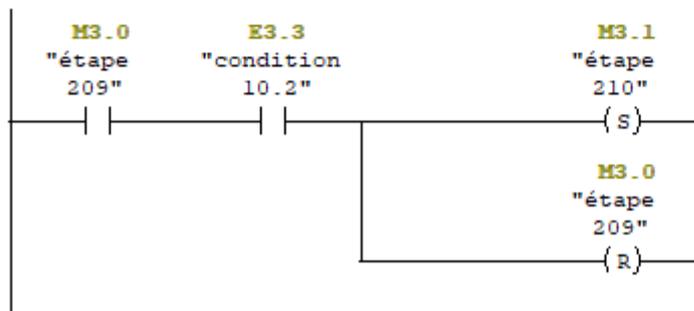
☐ Réseau 22 : Activation de l'étape 208



☐ Réseau 23 : Activation de l'étape 209

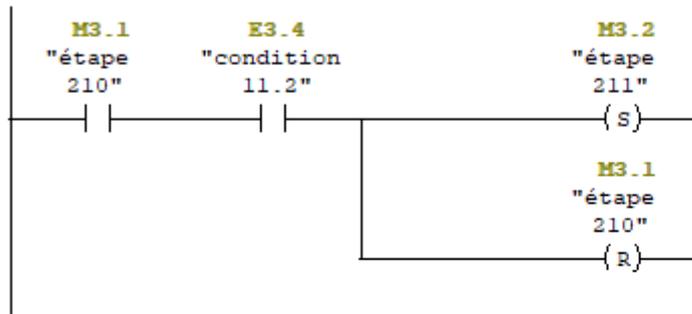


☐ Réseau 24 : Activation de l'étape 210

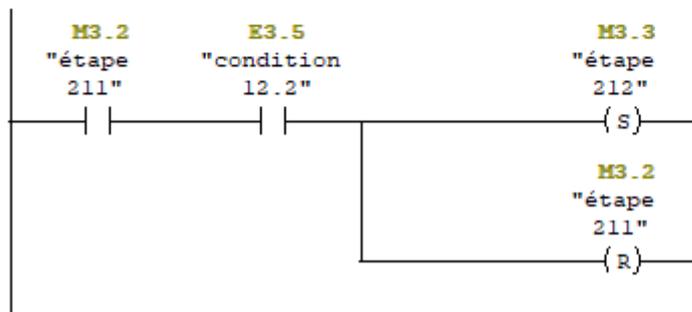


Annexe 02

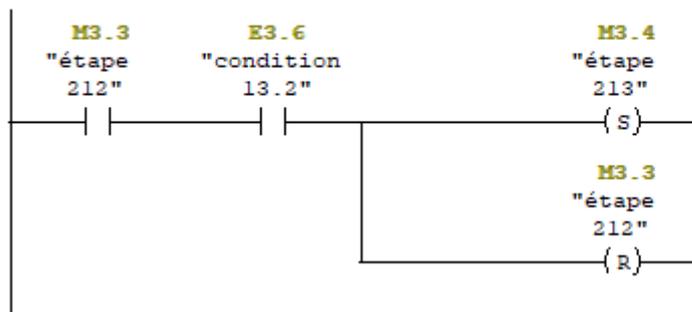
☐ Réseau 25 : Activation de l'étape 211



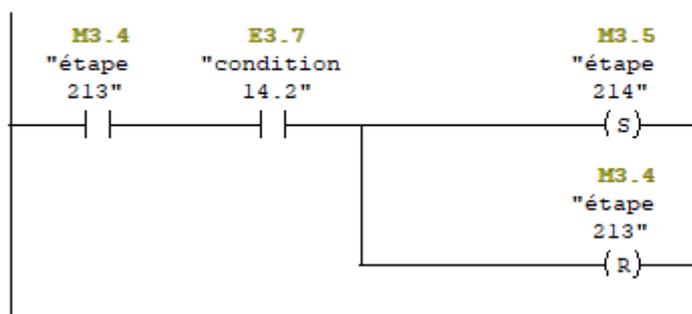
☐ Réseau 26 : Activation de l'étape 212



☐ Réseau 27 : Activation de l'étape 213

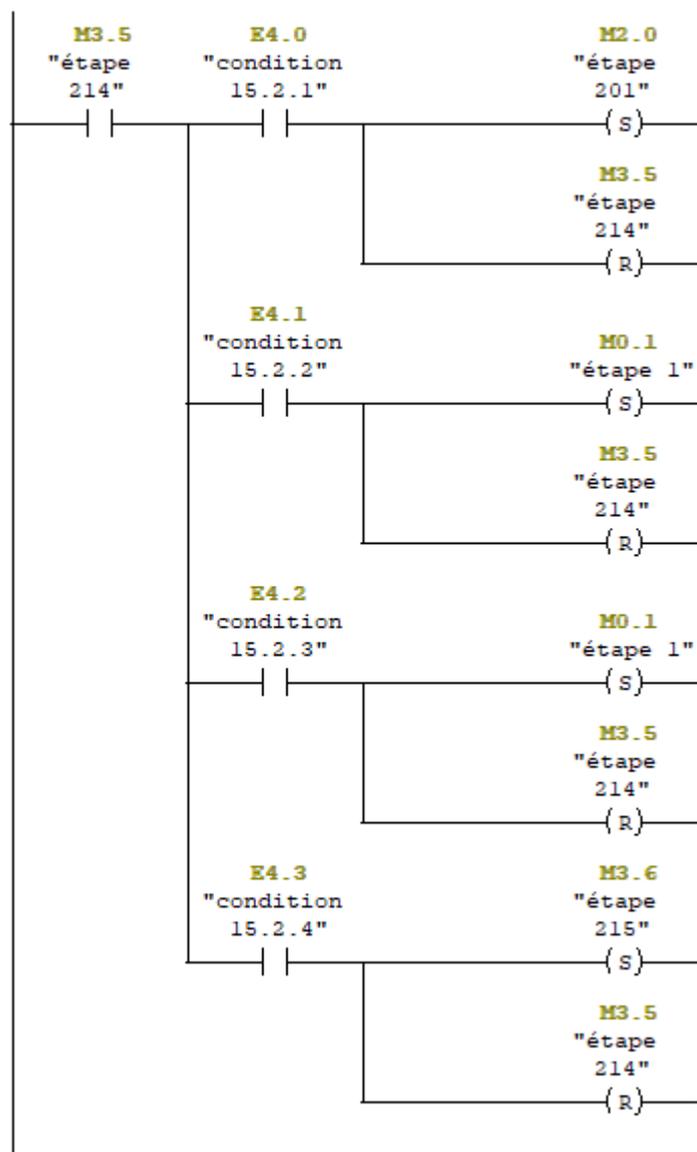


☐ Réseau 28 : Activation de l'étape 214



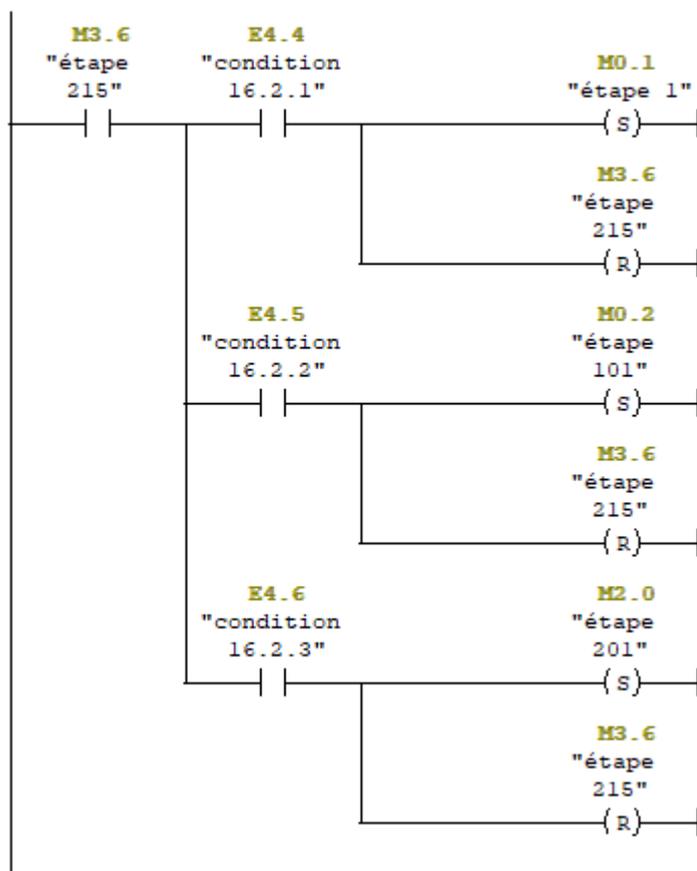
Annexe 02

☐ Réseau 29 : Activation de l'étape 1 ou 202 ou 215 à partir de l'étape 214



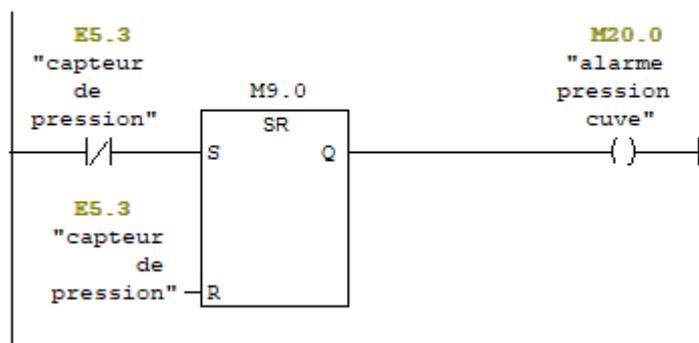
Annexe 02

☐ Réseau 30 : Activation de l'étape 1 ou 101 ou 201 à partir de l'étape 215



Annexe 02

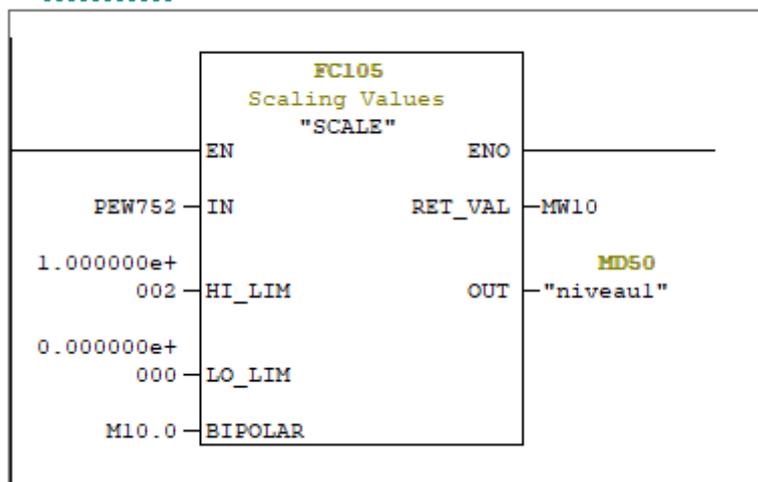
▣ Réseau 1: Alarmes Pression de la cuve



▣ Réseau 2: Alarme d'arrêt d'urgence

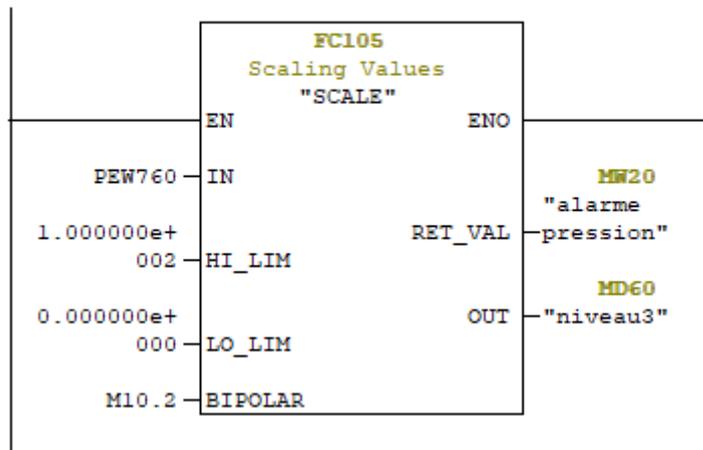


▣ Réseau 1: Niveau T1

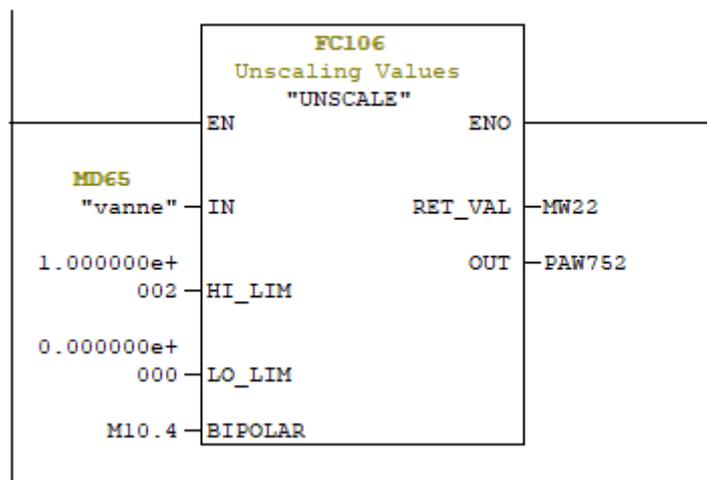


Annexe 02

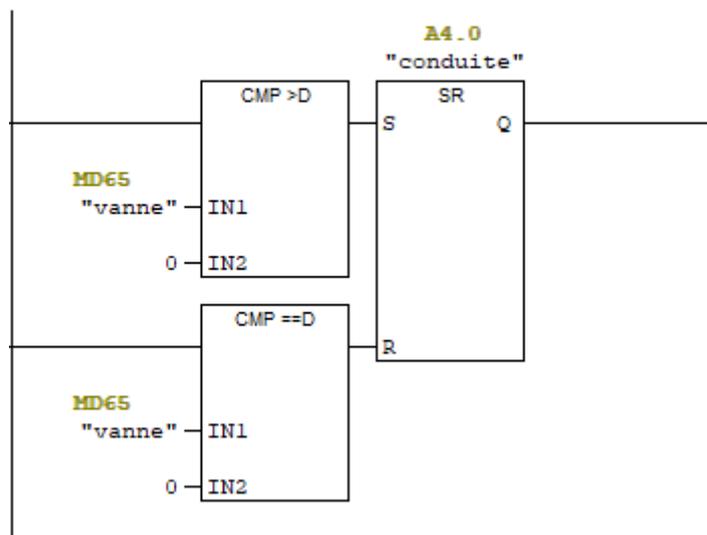
☐ Réseau 2 : Niveau T2



☐ Réseau 3 : Entrée Vanne

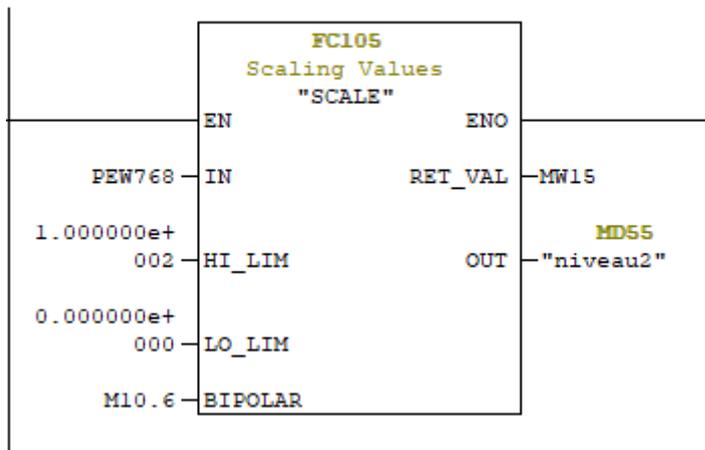


☐ Réseau 4 : Conduite Produit

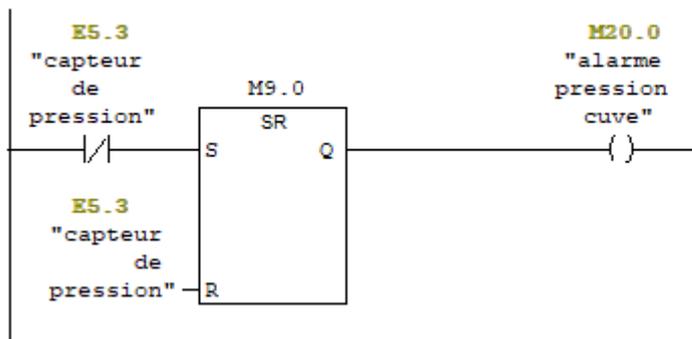


Annexe 02

▣ Réseau 5 : Niveau Présence Produit TT

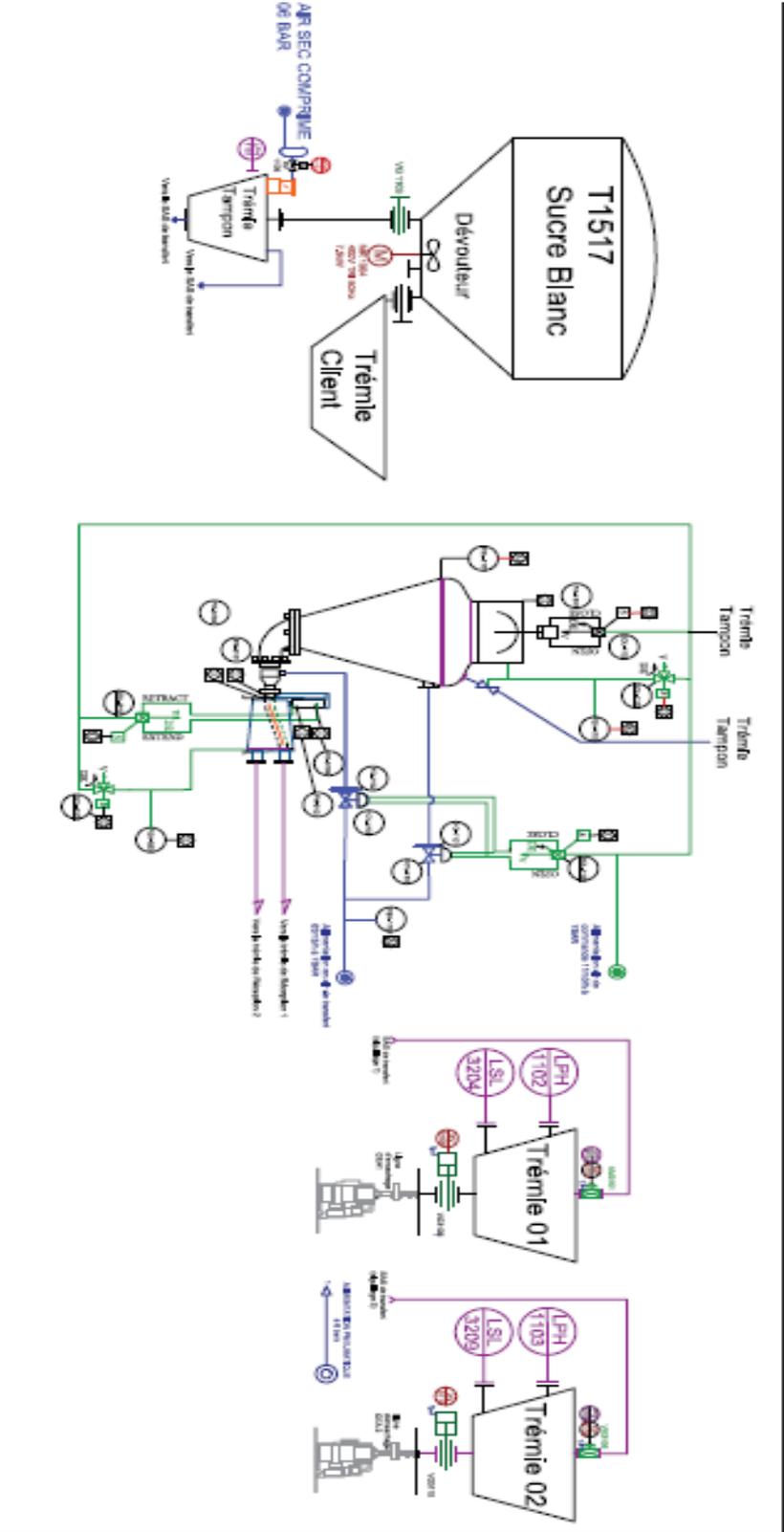


▣ Réseau 1: Alarmes Pression de la cuve



▣ Réseau 2 : Alarme d'arrêt d'urgence





Résumé :

L'objectif de ce projet de fin d'étude concerne l'étude de l'automatisation d'une ligne de production de sucre avec supervision IHM. Notre approche a été de faire une introduction sur les systèmes automatisés et les Automates Programmable Industriel (API). Puis, on a identifié les équipements matériels constituant l'installation puis on a présenté les logiciels à utiliser dans ce travail. Enfin, on a conclu par l'élaboration du programme Step7 permettant l'exécution du système puis on la superviser sur le logiciel WinCC.

Mots-clés : WinCC, Step7, API, Automatisation.

Abstract :

The objective of this end-of-study project concerns the study of the automation of a sugar production line with HMI supervision. Our approach has been to give an introduction to automated systems and Automate Programmable Industriel (API). Then, we identified the hardware equipment constituting the installation and then we presented the software used in this work. Finally, we concluded by developing the Step7 program for running the system and then supervising it on the WinCC software.

Keywords : WinCC, Step7, API, Automation.

ملخص:

الهدف من هذا المشروع المتعلق بدراسة خط إنتاج السكر بإشراف HMI. كان نهجنا هو إعطاء مقدمة للأنظمة المؤتمتة وأجهزة التحكم المنطقية القابلة للبرمجة الصناعية (API). بعد ذلك، حددنا الأجهزة و الآلات المكونة لعملية التثبيت ثم قدمنا البرامج المستخدمة في هذا العمل. أخيرًا، اختتمنا بتطوير برنامج Step7 لتشغيل النظام ومن ثم الإشراف عليه على برنامج WinCC.

الكلمات الأساسية: WinCC, API, Step7, اتتمتة.