

République Algérienne Démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Abderrahmane MIRA de BEJAIA
Faculté de Technologie
Département de Génie Electrique

Mémoire de fin de cycle

En vue de l'obtention du Diplôme de Master en ELECTRONIQUE

Option : AUTOMATIQUE

Thème

***COMMANDE ET SUPERVISION D'UN PROCESSUS
DE FABRICATION DE MARGARINE VIA UN AUTOMATE
PROGRAMMABLE (CHAÎNE PILOTE)***

Présenté par :

M^r : CHERCHOUR Hamza

M^r : CHAHBOUNE Mohamed Lamine

Encadré par :

M^r : MENDIL Boubekeur

M^r : GOUDJIL Boubekeur

Examiné par :

M^r : CHARIKH

M^{me} : ARRAR

Promotion : 2014/2015

Remerciements

Nous remercions, Dieu, le tout puissant pour nous avoir donné la foi qui nous a guidé jusqu'à la réalisation et l'aboutissement de ce projet.

Nous tenons à exprimer nos remerciements les plus distingués :

*A notre promoteur **Mr MENDIL B.**, et à notre encadreur **Mr GOUDJIL B.** De nous avoir fait l'honneur d'assurer l'encadrement de notre travail, nous vous sommes très reconnaissants d'avoir veillé à son élaboration en ne ménageant aucunement votre temps et vos conseils.*

Nous tenons à remercier vivement messieurs les membres du jury d'avoir consacré de leur temps à la lecture de ce manuscrit, et d'accepter de juger et d'évaluer ce travail.

*Nous tenons à remercier vivement l'ensemble du personnel de groupe **CEVITAL** en particulier : **Mr GOUDJIL B.** et **Mr MANCER D.** qui nous ont permis d'effectuer notre stage dans les meilleurs conditions.*

Par le biais de ce travail, nous exprimons notre profonde gratitude à toutes les personnes qui, de près ou de loin, nous ont aidées et accompagnées dans notre travail.

Nous voudrions remercier nos chers parents et nos Familles qui nous ont Soutenus dans nos études.

Hamza et Lamine



Dédicaces

A tous ceux qui nous sont chères ...

Hamza, Lamine



Liste des figures

Numéro	Désignation	Page
I.1.	Situation géographique du complexe CEVITAL.	4
I.2.	L'organigramme de CEVITAL.	5
I.3.	La chaîne pilote.	7
II.1.	Schéma de processus de fabrication de margarine (station pilote).	10
II.2.	Transmetteur de température.	11
II.3.	Structure d'un moteur asynchrone.	13
II.4.	Pompe à piston.	13
II.5.	Pompe centrifuge.	14
II.6.	Electrovanne.	14
II.7.	Manomètre	15
II.8.	variateurs de vitesse	16
II.9.	Régulateur de pression.	17
II.10.	Régulateur de pression électronique.	17
III.1	structure d'un système automatisé.	23
III.2	Automate compact (Allen-Bradley)	25
III.3	Automate Modulaire (Modicon).	25
III.4	Automate modulaire (Siemens).	25
III.5	Structure interne d'un API.	26
III.6	Cycle d'un API.	28
III.7	Constitution d'un API S7-300	30
III.8	CPU 315-2DP S7-300	30
III.9	présentation d'un schéma logique (LOG).	33
III.10	Présentation d'un schéma CONT	33
III.11	Présentation du langage LIST.	34
III.12	Blocs de programme.	34
III.13	Assistant nouveau projet	38
III.14	Choix de la CPU et du bloc d'organisation.	38
III.15	Nom et création du projet.	39
III.16	Station SIMATIC S7-300.	39

III.17	Configuration matériels.	40
III.18	Une partie de la table des mnémoniques du projet.	41
III.19	les blocs de programme.	42
III.20	programme de la pompe de bac de chauffe PW01.	43
III.21	Programme de régulateur continu de pasteurisateur.	44
III.22	Bloc de donnée d'instance de FB41	44
III.23	OB1 station pilote.	45
IV.1	Fenêtre principale du logiciel WinCC flexible.	49
IV.2	Choix du pupitre et de l'automate.	49
IV.3	Fenêtre d'une vue dans le winCC flexible.	50
IV.4	La vue modèle du de la station.	51
IV.5	Création et configuration de la vue combineur/cristaliseur.	52
IV.6	Création de la vue de régulateur de combineur K1.	53
IV.7	Intégration du projet IHM dans le projet STEP7.	53
IV.8	Configuration du réseau MPI.	54
IV.9	Configuration de la liaison entre le pupitre et l'automate.	55
IV.10	Configuration du S7-PLCSIM.	56
IV.11	Chargement du programme dans la CPU	56
IV.12	Lancement de la simulation.	57
IV.13	Visualisation du programme de la pompe PW01.	58
IV.14	Visualisation du programme de régulateur de pasteurisateur.	58
IV.15	Fenêtre de menu principale sur RUNTIME.	59
IV.16	Visualisation des actionneurs combineur/cristaliseur.	60
IV.17	Visualisation de régulateur de pasteurisateur.	61

Liste des tableaux

Numéro	Désignation	Page
III.1	Différents types de variables contenues dans le STEP7	37

Liste des abréviations et des symboles

T/J : Tonne/jour

PO : Partie Opérative

PR : Partie Relation

TOR : Tout Ou Rien

API : Automate Programmable Industriel

CPU : Computer Procès Unit

N/A : Numérique /Analogique

MPI : Multi Point Interface

CC : Courant Continu

IHM : Interface Homme-Machine

GRAFCET : GRAF de Commande Etapes-Transitions

LED : Light-Emitting Diode

PG : Console de programmation

CONT : Schéma à contact

LIST : Liste d'instruction

LOG : Logigramme

PC : Personally Computer

PROFIBUS : Protocole de communication

WINPLC7 : Logiciel de programmation et de simulation

STEP7 : Logiciel de programmation et de simulation

OB : Bloc d'Organisation

IHM : Interface Homme Machine

Win CC Flexible : Logiciel de la supervision

E/S : Entrées/Sorties

Liste des figures

Liste des tableaux

Liste des abréviations et des symboles

Sommaire

Introduction générale..... 1

Chapitre I : Généralités

I.1.Introduction 3

I.2.Présentation de l'entreprise CEVITAL 3

I.2.1. Historique du complexe 3

I.2.2. Situation géographique 3

I.2.3. Activité de CEVITAL 4

I.2.4.Commercialisation 5

I.2.5.Organisation de l'entreprise 5

I.2.6. Missions et objectifs 6

I.3. Unité de Production des margarines 6

I.4. Présentation de la chaine pilote 7

1.5. Conclusion 8

Chapitre II : Description de la chaine pilote pour la production de margarine

II.1. Introduction 9

II.2.Principe de fonctionnement de la station pilote 9

II.3. Description des équipements du système 10

II.3.1. Bacs d'émulsions 10

II.3.2. Capteurs	11
II.3.3. Pasteurisateur	12
II.3.4. Combinateur	12
II.3.5. Cristalliseur	12
II.3.6. Moteurs électriques asynchrones	12
II.3.7. Pompe à piston	13
II.3.8. Pompe centrifuge	14
II.3.9. Electrovanne	14
II.3.10. Manomètre	15
II.3.11. Partie électrique	15
II.3.12. Régulateur de pression	17
II. 4. Régulation via API S7-300	18
II.4.1. Régulation par action « Tout Ou Rien »	18
II.4.2. Régulateur PID	19
II.4.3. Le Bloc FB 41 « CONT_C »	20
II.5. Conclusion	20

Chapitre III : Automates et leur programmation : Cas de la station pilote

III.1. Introduction	21
III.2. Systèmes automatisés	21
III.2.1. Structure des systèmes automatisés	21
III.3. Généralités sur les API	22
III.3.1. Définition d'un automate programmable industriel (API)	22
III.3.2. Architecteur des automates	23
III.3.3. Aspect extérieur des API	23
III.3.4. structure interne d'un automate	25

III.3.5. Description des éléments d'un API	25
III.3.6. Cycle d'un API	27
III.3.7. Critères de choix d'un API	28
III.3.8. Présentation de l'automate utilisé S7-300	28
III.4. Programmation via STEP 7	31
III.4.1. Définition du logiciel STEP7	31
III.4.2. Langages de programmation sous STEP7	32
III.4.3. Structure du programme STEP7	33
III.4.4. Blocs	34
III.4.5. Mémentos	35
III.4.6. Mnémoniques	35
III.4.7. Les différents types de variables	35
III.5. Développement du programme pour la station pilote	36
III.5.1. Démarrage du logiciel STEP7	36
III.5.2. Création d'un projet STEP7	37
III.5.3. Configuration matériel	38
III.5.4. Table des mnémoniques	40
III.5.5. Création du programme	40
III.5.5.1. Programme de FC2	41
III.5.5.2. Programme de FC38	42
III.6. Conclusion	44

Chapitre IV : Supervision et simulation

IV.1. Introduction	45
IV.2. Supervision	45
IV.2.1. Définition de la supervision	45
IV.2.2. Architecture d'un réseau de supervision	46

IV.2.3. Avantage de la supervision	46
IV.3. Présentation du logiciel WinCC flexible	46
IV. 3.1. Création du projet avec WinCC flexible	47
IV.3.2. Choix du pupitre et de l'automate	48
IV.3.3. La mise en route du WinCC flexible	49
IV.3.4. Configuration des vues via WinCC flexible	49
IV.3.5. Création des vues du projet de la station pilote	50
IV.3.6. Intégration du projet WinCC flexible dans le projet step7	52
IV.3.7. Configuration du réseau	53
IV.3.8. Interface multipoint MPI	53
IV.3.9. Activer la liaison	53
IV.4. Simulation du projet	54
IV.4.1. Simulation sous STEP7	54
IV.4.1.1. Ouverture et configuration du S7-PLCSIM	54
IV.4.1.2. Chargement du programme et la mise en ligne	55
IV.4.1.3. Le lancement de la simulation	56
IV.4.1.4. Visualisation du programme de la pompe de bac de chauffe PW01	56
IV.4.1.5. Visualisation du programme du régulateur de pasteurisateur	57
IV.4.2. Simulation avec WinCC flexible	58
IV.4.2.1. Activer la simulation	58
IV.4.2.2. Visualisation du processus dans la partie combineur/cristaliseur	59
IV.4.2.3. Visualisation de régulateur de pasteurisateur	59
IV.5. Conclusion	60
Conclusion générale	61
Références bibliographiques	62
Annexes	

Introduction générale

Introduction générale

Au cours des dernières décennies, les procédés industriels ont connu un essor vertigineux, grâce à l'automatisme. Ainsi, la mondialisation des économies et l'accentuation de la concurrence poussent davantage les entreprises à automatiser leurs processus de production, afin d'assurer leur pérennité en s'offrant la compétitivité, tout en améliorant les conditions de travail de leur personnel ; en supprimant les tâches pénibles et répétitives.

Les automates programmables industriels apportent la solution sur mesure pour les besoins d'adaptation et de flexibilité de nombreuses activités économiques actuelles. Ils sont devenus aujourd'hui les constituants les plus répandus des installations automatisées.

Depuis plus d'une décennie, les entreprises Algériennes ont commencé à renouveler et à automatiser leurs installations. CEVITAL a fait un grand pas dans ce domaine. La modernisation de ses installations et le travail selon les normes internationales fondent son image. Ceci lui a permis de réaliser une productivité optimale et lui offre une meilleure combinaison : prix, qualité, quantité et délai.

Le fond de notre travail est effectué au sein de de l'entreprise CEVITAL. L'objectif est d'installer un panel de supervision d'un processus automatisé pour la fabrication de margarine dans une chaîne pilote. Une étude générale des différents instruments présents sur la station, en particulier l'API utilisé a été effectuée et le programme qui gère son fonctionnement a été analysé. Cette panel, fonctionnant avec un ancien système (*Protool*), a été inséré et reprogrammée avec le logiciel *WinCC flexible*, pour se conformer au reste de l'installation et, en particulier, la salle de contrôle. Après la vérification de la procédure d'installation matérielle, un projet WinCC a été créé pour assurer le bon fonctionnement de la panel et une meilleure supervision de la station.

Pour une meilleure présentation de notre travail, le mémoire a été organisé en quatre chapitres. Le premier est consacré à la présentation de l'entreprise et l'unité de margarinerie. Le chapitre deux expose le principe et le fonctionnement de la station, ainsi que l'identification des différents équipements et la définition des types de régulation utilisés.

Le troisième chapitre donne un bref aperçu sur la structure des systèmes automatisés ainsi que les automates, en particulier l'automate S7-300. La deuxième partie est consacrée aux différents aspects de la programmation. Le cas de la station pilote, qui fait l'objet de notre étude, est abordé dans la dernière partie du chapitre.

L'objectif du dernier chapitre est l'étude d'un système de supervision avec le logiciel WinCC flexible pour visualiser l'état de fonctionnement de la station pilote afin de surveiller et de détecter les anomalies qui peuvent survenir au cours de fonctionnement du procédé. Après un aperçu sur le logiciel de supervision (WinCC flexible), les deux simulations (sous STEP7 et Win CC) ont été présentées, ainsi que les détails de l'implémentation pratique sur la station (configuration matérielle, installation du projet WinCC sur la panel). Enfin, une conclusion générale a été présentée.

Chapitre I

Généralités

I.1. Introduction

Dans ce chapitre, nous évoquons en premier lieu l'évolution historique du complexe agroalimentaire CEVITAL, sa situation géographique, ses différentes activités industrielles, Ses divers objectifs et l'organigramme de ses différentes directions, ensuite, nous présenterons l'unité de margarinerie et la chaîne pilote.

I.2. Présentation de l'entreprise CEVITAL

I.2.1. Historique du complexe

CEVITAL est parmi les entreprises algériennes qui ont vu le jour dès l'entrée de notre pays en économie de marché. Elle a été créée par des fonds privés en 1998. Son complexe de production se situe dans le port de Bejaia et s'étend sur une superficie de 4500m².

CEVITAL contribue largement au développement de l'industrie agroalimentaire nationale. Elle vise à satisfaire le marché national et à exporter le surplus en offrant une large gamme de produits de qualité. En effet, les besoins du marché national sont de 600 T/J de margarine. Les capacités actuelles de CEVITAL sont de 3000 T/J [1].

Les nouvelles données économiques nationales dans le marché de l'agroalimentaire font que les meilleurs sont ceux qui maîtrisent d'une façon efficace et optimale les coûts et les charges et ceux qui offrent le meilleur rapport qualité/prix. Ceci est nécessaire pour s'imposer sur le marché que CEVITAL négocie avec les grandes sociétés commerciales internationales. Ses produits se vendent dans des différentes villes africaines (Lagos, Niamey, Bamako, Tunis, Tripoli...) [1].

I.2.2. Situation géographique

CEVITAL est implanté au niveau du quai du port de Bejaia, à 3km du sud-ouest de 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 (Figure I.1).

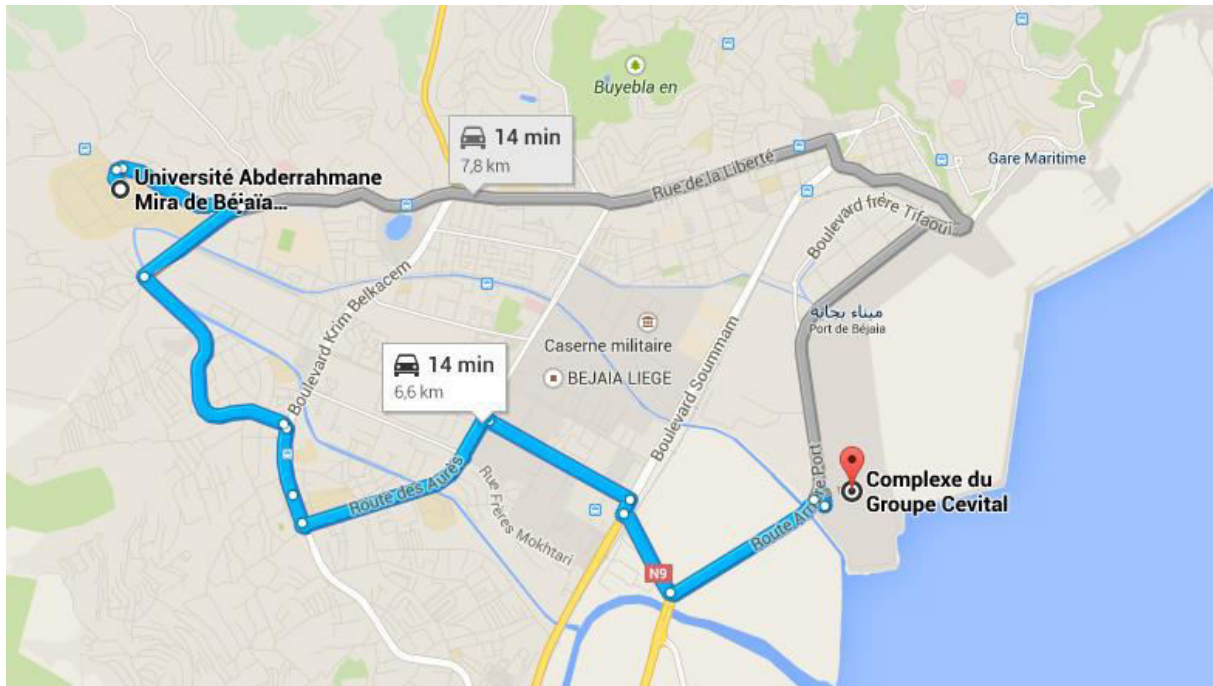


Figure .I.1 Situation géographique du complexe CEVITAL.

I.2.3. Activité de CEVITAL

Lancé en mai 1998, le complexe CEVITAL a débuté son activité par le conditionnement d'huile en décembre 1998. En février 1999, les travaux de génie civil de la raffinerie ont débuté. Cette dernière est devenue fonctionnelle en août 1999.

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 :

- ✓ Raffinage des huiles (1800 tonnes/jour).
- ✓ Conditionnement d'huile (1400 tonnes/jour).
- ✓ Production de margarine (600 tonnes/jour).
- ✓ Fabrication d'emballage (PET) ; Poly-Ethylène-Téréphtalate (9600 unités/heure).
- ✓ Raffinage de sucre (1600 tonnes/jour).
- ✓ Stockage des céréales (120000 tonnes/jour).
- ✓ Minoterie et savonnerie en cours d'étude.

I.2.4. Commercialisation

Un immense budget est consacré à la publicité pour amener un nombre considérable de commerçants dans les différentes wilayas du pays à se faire agréer par CEVITAL et pour gagner la confiance du consommateur qui devient de plus en plus exigeant. CEVITAL présente et rapproche le maximum de ses produits aux consommateurs dans toutes les régions du pays en offrant le meilleur rapport qualité/prix.

I.2.5. Organisation de l'entreprise

Le complexe compte à son effectif plus de 3600 employés (permanents et contractuels), répartis sur les différentes structures. Son organigramme est présenté comme suit :

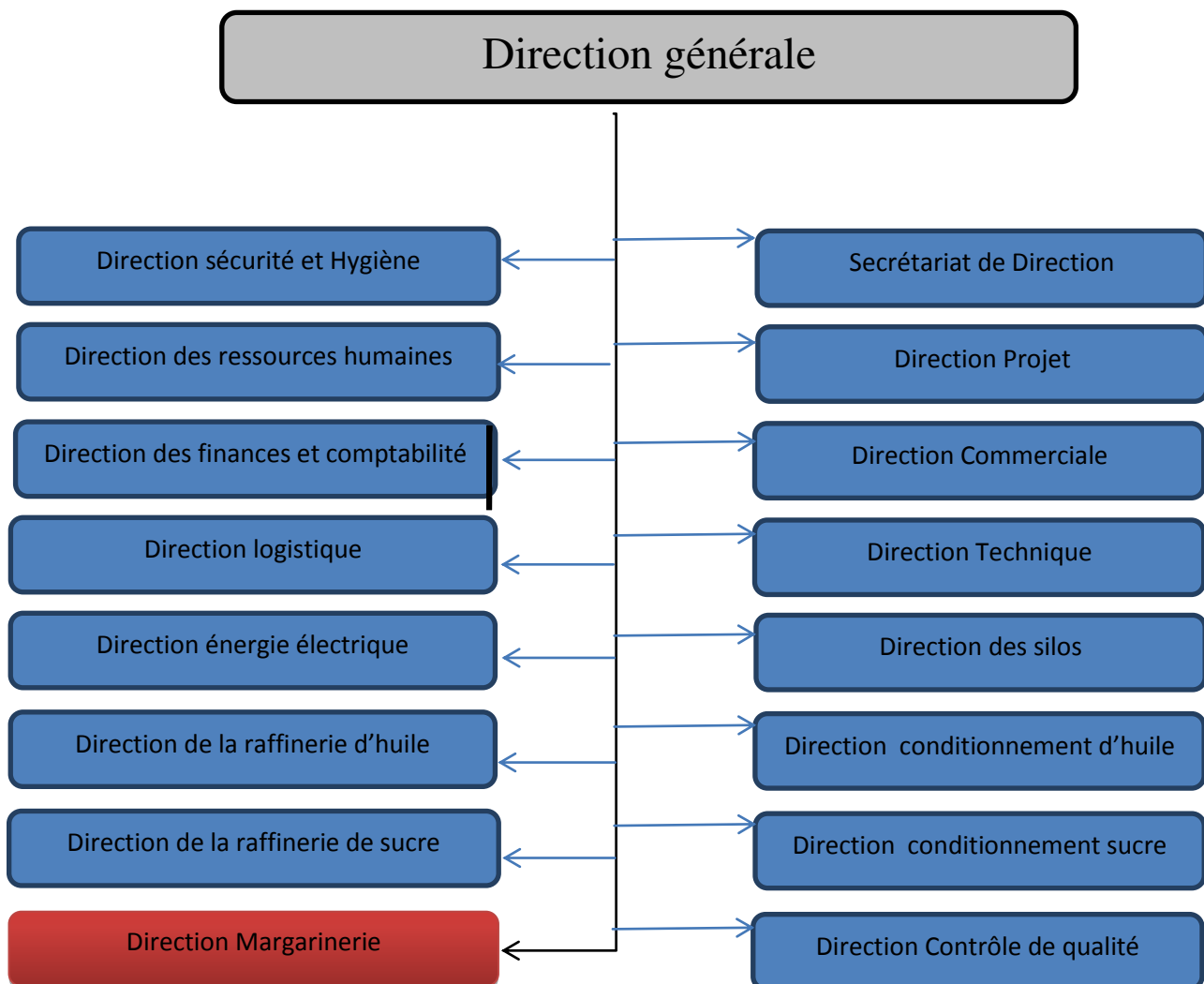


Figure I.2: L'organigramme de CEVITAL.

I.2.6. Missions et objectifs

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

Les objectifs visés par CEVITAL peuvent se présenter comme suit :

- ✓ L'extension de ses produits sur tout le territoire national.
- ✓ L'importation des graines oléagineuses pour l'extraction directe des huiles brutes.
- ✓ L'optimisation de ses offres d'emplois sur le marché du travail.
- ✓ L'encouragement des agriculteurs par des aides financières pour la production locale des graines oléagineuses.
- ✓ La modernisation de ses installations en termes de machines et techniques pour augmenter le volume de sa production.
- ✓ Positionner ses produits sur le marché étranger par ses exportations.

I.3. Unité de Production de margarine

Cette unité est spécialisée dans la production de la margarine et les graisses finies ou semi finies. Elle a une capacité de production de 600 tonnes/jour. Elle produit essentiellement **MATINA, FLEURIAL, Elio2**, Margarine de Feuilletage, **SMEN** et les **Shortening**.

L'atelier de production de l'unité margarinerie possède cinq (05) lignes de production. Chaque ligne est composée d'un certain ensemble de ressources connectées entre elles. Chacune est assignée à produire un (des) article (s) bien spécifique.

- la ligne (L1) pour la production des barquettes de 400 grammes de Mâtina, barquettes de 500 grammes Elio2, barquette de 500 grammes de Smen.
- la ligne (L2) pour la production des plaquettes de 250 grammes de Fleurial, plaquettes de 500 grammes de Feuilletage.
- la ligne (L3) pour la production des plaquettes de 500 grammes de Feuilletage.
- la ligne (L4) pour la production des barquettes de 1.8 kg de Smen.
- la ligne (L5) pour la production des cartons de 20 kg de Shortening des points de fusion (31-32, 35-37 et 38-40)

I.4. Présentation de la chaine pilote

La chaine pilote est une machine de production à petite échelle qui permet de traiter des petits échantillons des nouvelles recettes des produits, exactement dans les mêmes conditions que celles générées dans les installations de production à grande échelle. Elle est créée par le constructeur allemand **Schröder** pour tester et développer de nouvelles recettes de margarines, sans perturber la production à grande échelle.

Cette station est composée de :

- ✓ Deux bacs d'émulsion ;
- ✓ Deux pompes (pompe à piston et pompe centrifuge) ;
- ✓ Un pasteurisateur (réchauffeur) ;
- ✓ Un refroidisseur ;
- ✓ Deux combineurs ;
- ✓ Deux cristalliseurs ;
- ✓ Un tube de repos.



Figure I.3 : La chaine pilote.

1.5. Conclusion

L'acquisition de grandes parts de marché ne peut avoir lieu que si l'entreprise satisfait le plus grand nombre possible de sa clientèle. Pour cela, elle essaie d'attirer l'attention des clients par un bon conditionnement des produits et gagner leur confiance par la bonne qualité, ainsi qu'un bon rapport (qualité/prix).

Parmi les objectifs envisageables par l'entreprise, on cite :

- L'encouragement de l'agriculture locale pour la production des graines oléagineuses en proposant des subventions (aides) ;
- L'incorporation des nouvelles méthodes de gestion et des technologies récentes dans les ateliers de production pour augmenter la productivité ;
- Acquérir d'autres parts du marché national en couvrant toutes les régions possibles ;
- Commercialiser ses produits sur les marchés étrangers.

Chapitre II

Description de la chaine pilote pour la production de margarine

II.1. Introduction

Le système de contrôle du processus a besoin de connaître, à chaque instant, un grand nombre d'informations internes nécessaire à leur fonctionnement. Ses informations sont obtenues par mesure directe ou par déduction (technique d'observation). La chaîne d'instrumentation doit donc prendre en compte les spécificités du processus physique mesuré.

Dans ce chapitre, nous allons d'abord décrire le processus de la station pilote et identifier ces équipements (actionneurs, variateurs, régulateurs). Ensuite, nous allons présenter l'état actuel de la station et la problématique.

II.2. Principe de fonctionnement de la station pilote

La margarine est constituée de divers produits : l'huile, lait, vitamines, eau, sel, arôme, conservateur, ...etc. Le processus de préparation est constitué des étapes suivantes (Figure II.1) :

- Après émulsion de tous ces ingrédients dans le bac avec un mélangeur et un agitateur et après l'obtention d'une température désirée du mélange (cette partie est indépendante de l'automate) ;
- Après l'ouverture de la vanne manuellement, la pompe pousse le produit vers le pasteurisateur (réchauffeur). Ce dernier augmente la température de l'émulsion à l'aide d'une vapeur saturante pour éliminer les bactéries dans le produit. Ensuite, le produit passera par un refroidisseur ;
- Une fois pasteurisée et refroidie, l'émulsion passe à travers des combineurs (échangeurs de chaleur à surface raclée). Ceux-ci permettent le refroidissement du produit à la température requise du processus. Le refroidissement est assuré par un liquide frigorigène (l'ammoniac) qui circule dans la chemise réfrigérante. Un échange de chaleur s'opère entre celle-ci et le produit. A l'intérieur du tube combineur, les lames de l'axe raclent la paroi intérieure du tube cylindrique. Elles empêchent ainsi la cristallisation du produit sur celle-ci et assurent un transfert de chaleur optimal. L'intensité de raclage et de refroidissement au cours du processus dépend du produit (type de margarine par exemple) ;
- Le produit sortant de combineur est travaillé dans les cristalliseurs qui permettent un traitement intermédiaire du produit pendant la phase de cristallisation. A l'intérieur du cristalliseur, les tiges mélangent le produit pour en améliorer sa structure. Ce dernier assure

un refroidissement suffisant de l'émulsion pour donner une forme au produit et lui assurer une bonne cristallisation au niveau du tube à repos ;

- Le produit passe, enfin, au tube dit de repos. Celui-ci est obligatoire pour atteindre la consistance et la performance souhaitées pour le produit fini.

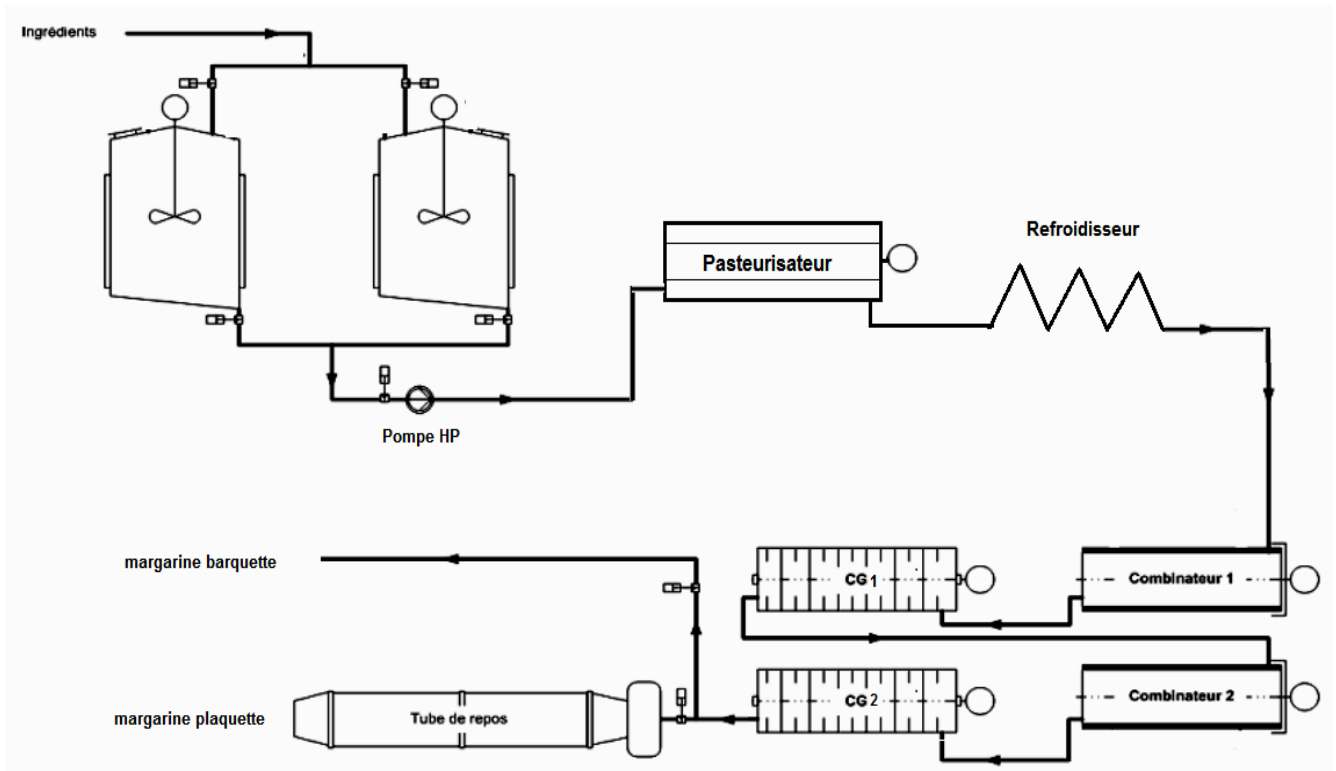


Figure II.1 : Schéma de processus de fabrication de margarine (station pilote).

II.3. Description des équipements du système

La chaîne pilote est conçue pour une large gamme de margarines. Elle offre une bonne cristallisation à ces produits. Les différentes unités qui la composent permettent un développement de produits optimal. La station comporte les éléments suivants:

II.3.1. Bacs d'émulsions

Il en existe deux pour tester deux solutions une après l'autre. Les deux bacs sont des cylindres en acier inoxydable à double parois munis d'un agitateur et d'un mélangeur.

II.3.2. Capteurs

Un capteur est un dispositif qui transforme une grandeur physique en une grandeur normée, généralement électrique, qui peut être interprétée par un dispositif de contrôle /commande. Il donne une information qui permet au système automatisé de déclencher des actions. Les capteurs se divisent en deux groupes : les capteurs logiques, et les capteurs analogiques.

A- Les capteurs logiques : Ce type de capteur permet de détecter un événement ou un objet lié au fonctionnement du système. On les appelle aussi les capteurs « tout ou rien » (TOR) car le signal électrique fourni est de type logique (0 ou 1).

Exemple du capteur de niveau : Ce capteur, de type logique (TOR), est conçu pour la détection de niveau de liquides de toutes natures (ammoniac dans notre système). Son rôle est de détecter l'existence ou non d'un certain niveau du liquide.

B- Les capteurs analogiques : La sortie du capteur analogique peut prendre une infinité de valeurs dans un intervalle donné. Chaque valeur de la grandeur physique doit correspondre à une valeur en sortie. Une variation de la grandeur physique d'entrée du capteur produit une variation de la caractéristique électrique du capteur (courant, tension, fréquence,...).

Exemple de capteur de température PT100 : La sonde PT100 est un capteur de température qui est utilisé dans le domaine industriel (agroalimentaire, chimie, raffinerie...). Il est constitué d'une résistance en Platine de valeur 100 Ohm à 0°C qui augmente avec la température selon un coefficient caractéristique du matériau de la résistance. La valeur de ce coefficient est $\alpha = 3.85 \cdot 10^{-3} \Omega/\Omega/^\circ\text{C}$ calculé entre 0 et 100°C[1]. La sonde est reliée à un transmetteur de température qui transforme la résistivité en intensité (signal normalisé 4-20mA) qui va permettre d'afficher numériquement la température.



Figure II.2 : Transmetteur de température.

II.3.3. Pasteurisateur

L'unité de pasteurisation, aussi appelée débactérisation thermo-contrôlée, est un procédé de conservation des aliments par lequel ceux-ci sont chauffés à une température définie pendant une durée. Cette étape présente plusieurs avantages. Elle assure l'inhibition de bactéries, la croissance et le développement d'autres micro-organismes. Ce qui améliore la microbiologie du produit.

II.3.4. Combinateur

Le combinateur est un échangeur de chaleur à surface raclée. Ceci permet le refroidissement du produit à la température requise. Il se compose d'un cylindre à double paroi dans lequel tourne un axe équipé de lames rotatoires. La surface des tubes en contact avec l'émulsion est en acier inoxydable, tandis que la surface externe est en acier carbone soutenue par une couche de chrome dur. Le refroidissement est assuré par un liquide frigorigène (l'ammoniac) qui circule dans la chemise réfrigérante. A l'intérieur du tube refroidisseur, les lames de l'axe raclent la paroi intérieure du tube cylindrique. Elles empêchent ainsi la cristallisation du produit sur celle-ci et assurent un transfert de chaleur optimal. L'intensité de raclage et de refroidissement au cours du processus dépend du produit.

II.3.5. Cristalliseur

Le cristalliseur permet un traitement intermédiaire du produit pendant la phase de cristallisation. Il se compose d'un cylindre à double paroi équipé de tige, dans lequel circule un autre porte-tige. A l'intérieur du cristalliseur, la tige mélange le produit pour en améliorer la structure. Le cristalliseur assure un refroidissement suffisant de l'émulsion pour donner une forme au produit et lui assurer une bonne cristallisation au niveau du tube à repos.

II.3.6. Moteurs électriques asynchrones

Les moteurs asynchrones, appelés aussi moteurs à induction, constituent plus de 60% des machines tournantes qui assurent la conversion de l'énergie électrique en énergie mécanique. Le moteur asynchrone est robuste et d'un prix de revient relativement insignifiant. Ce qui fait de lui le plus utilisé dans l'industrie, surtout avec le progrès de l'électronique de puissance qui a permis une meilleure régulation à vitesse variable.

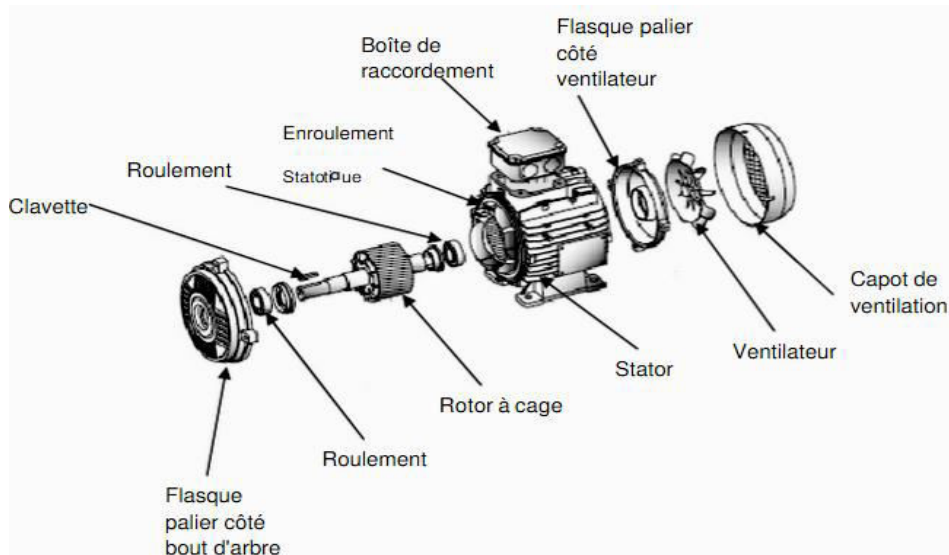


Figure II.3 : Structure d'un moteur asynchrone.

Le moteur asynchrone est constitué d'un inducteur fixe, appelé stator et d'une partie mobile (induit), appelé rotor. Le stator est constitué d'une couronne de tôles d'aciers, encochées à l'intérieur et empilées formant la carcasse du moteur. Elle porte à l'intérieur des encoches d'un bobinage triphasé. Le rotor comporte un bobinage en court-circuit. Il peut être constitué aussi d'un cylindre massif en matériau conducteur. Dans les deux cas, le bobinage rotorique forme un circuit fermé sur lui-même.

II.3.7. Pompe à piston

La pompe à piston constitue l'un des plus anciens types de pompes et demeure parmi les plus répandues. C'est une pompe volumétrique travaillant couramment à des pressions supérieures à plusieurs centaines de bars. Comme son nom l'indique, elle utilise les variations de volume occasionnées par le déplacement des pistons.



Figure II.4 : Pompe à piston.

II.3.8. Pompe centrifuge

Une pompe centrifuge est une machine rotative qui pompe un liquide en le forçant à travers une roue à aube ou une hélice appelée impulseur (souvent nommée improprement turbine). C'est le type de pompe industrielle le plus commun. Par l'effet de la rotation de l'impulseur, le fluide pompé est aspiré axialement dans la pompe, puis accéléré radialement, et enfin refoulé tangentiellement [9].



Figure II.5 : Pompe centrifuge.

II.3.9. Electrovanne

Une électrovanne est un dispositif commandé électriquement, permettant d'autoriser ou d'interrompre, par une action mécanique, la circulation d'un fluide dans un circuit. C'est un actionneur électromagnétique TOR, appelé souvent bobine ou solénoïde. Une électrovanne est constituée principalement d'un corps de vanne où circule le fluide et d'une bobine alimentée électriquement qui fournit une force magnétique déplaçant le noyau mobile qui agit sur l'orifice de passage permettant ainsi la circulation ou non circulation du fluide. La bobine doit être alimentée d'une manière continue pour maintenir le noyau attiré.



Figure II.6 : Electrovanne.

II.3.10. Manomètre

Le manomètre est utilisé pour la mesure de pression. Cet instrument est utilisé pour pression relative, absolue et différentielle. Il contient un élément sensible qui se déforme dès qu'il est sollicité par la pression du fluide. Les éléments peuvent être au ressort tubulaire, à la membrane, à la capsule et au soufflet. Les matériaux avec lesquels est construit peuvent être le ligue de cuivre, l'acier dans les différentes exécutions ou, dans les cas spéciaux, des matériaux spéciaux.

La pression est toujours mesurable en relation à une pression de référence. La pression de référence est la "Pression Atmosphérique" et donc le manomètre indique tout ce qu'il est supérieur ou inférieure à la pression atmosphérique. La pression est indiquée sur un cadran par un index, dans les champs normalisés de mesure, sur un angle de 270°. Dans les cas de vibrations ou de hautes charges dynamiques, la caisse des instruments est remplie de liquide. Dans les installations industrielles d'automation, il est possible de prévoir des contacts électriques pour l'indication de la pression moindre ou maximale ou transmetteurs qui peuvent fournir un signal électrique proportionnel à la pression.



Figure II.7 : Manomètre

II.3.11. Parties électriques

A. Variateur de vitesse

Du point de vue électrique : un variateur de vitesse est un dispositif grâce auquel la vitesse d'un moteur électrique peut être changée. En mécanique, il est défini comme un dispositif composé de courroies et de poulies grâce auquel la démultiplication des moteurs peut être changée. La figure II.8 montre la composition interne d'un variateur de vitesse.

Depuis la venue de la technologie des semi-conducteurs, la variation de vitesse électronique des moteurs électriques a pris le dessus sur les anciens systèmes.

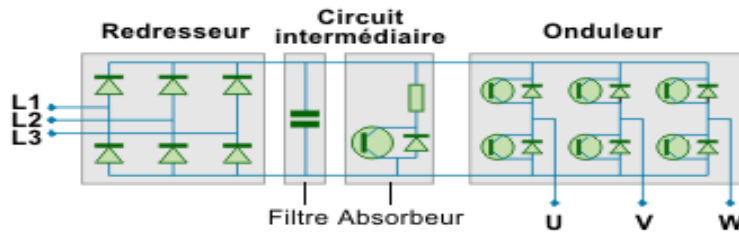


Figure II.8 : variateurs de vitesse. [8]

Le variateur de vitesse est composé essentiellement :

- ✓ D'un redresseur qui, connecté à une alimentation triphasée (le réseau), génère une tension continue à ondulation résiduelle (le signal n'est pas parfaitement continu). Le redresseur peut être de type commandé ou pas ;
- ✓ D'un circuit intermédiaire agissant principalement sur le "lissage" de la tension de sortie du redresseur (améliore la composante continue). Le circuit intermédiaire peut aussi servir de dissipateur d'énergie lorsque le moteur devient générateur ;
- ✓ D'un onduleur qui engendre le signal de puissance à tension et/ou fréquence variables ;
- ✓ D'une électronique de commande pilotant (transmission et réception des signaux) le redresseur, le circuit intermédiaire et l'onduleur [8].

B. Un contacteur

Un contacteur est un appareil de connexion à bobine commandé qui est généralement maintenu fermé par un courant permanent (réduit) circulant dans la bobine. Il est conçu pour effectuer un nombre important de cycle de manœuvre (fermé/ouvert) [1].

C. Un disjoncteur

C'est un appareil mécanique de connexion, conçu pour établir, supporter et interrompre un courant dans un circuit électrique, ainsi il protège l'installation des surcharges et des courts circuits [1].

D. Un Fusible

Le principe de la protection par fusible repose sur la fusion contrôlée d'un élément fusible, fusion qui intervient après un temps donné pour un courant donné [1].

II.3.10. Régulateur de pression

Le régulateur de pression, ou détendeur, permet de garantir une pression de travail (pression du secondaire) aussi régulière que possible tant que la pression d'alimentation (pression du primaire) est supérieure à la pression demandée. Le réglage de la pression souhaitée se fait manuellement.

Afin de garantir une disponibilité optimale de la pression nécessaire avec un air le plus pur possible, chaque équipement industriel est équipé d'une unité de conditionnement d'air comportant, au minimum, un filtre à air et un mano-détendeur. Dans certains cas, on adjoint un huileur (ou lubrificateur) pour lubrifier l'air à l'entrée des équipements industriels.



Figure II.9 : Régulateur de pression.

Exemple de régulateur de pression électronique : Le Régulateur de pression électronique permet de faire varier la valeur du débit et la pression. Ce type de régulateur est appelé aussi régulateur proportionnel à commande électronique. Pour cette partie électronique, le contrôle de la valeur de pression se fait au moyen d'un signal en tension (0-10 Volts) ou en courant (4-20 mA). Un écran de visualisation de la valeur en cours permet d'avoir une sortie analogique avec un signal en tension ou en courant de la valeur de la pression relevée dans le circuit.

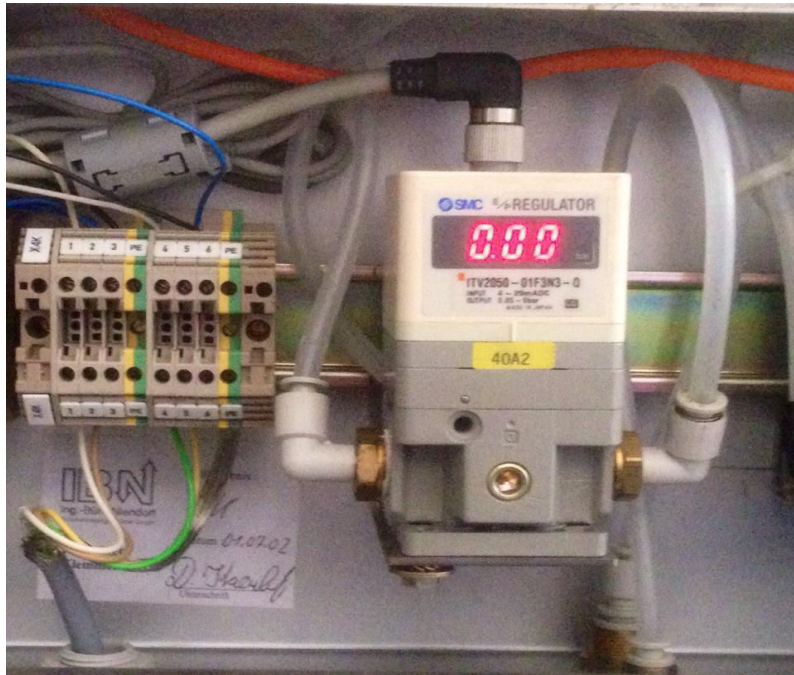


Figure II.10 : Régulateur de pression électronique.

II. 4. Régulation via API S7-300

La régulation automatique est la technique de l'ingénieur offrant les méthodes/outils nécessaires à la prise de contrôle d'un système physique (installation de production, robot, alimentation électronique stabilisée,...) en vue d'en imposer le comportement. Cette prise de contrôle s'effectue par l'intermédiaire de certains signaux (grandeurs physiques) qu'il est alors nécessaire de mesurer afin de déterminer l'action à entreprendre sur le système.

Le comportement des grandeurs contrôlées peut/doit en général satisfaire plusieurs critères:

- Une certaine grandeur physique (vitesse, courant, température) donne une valeur moyenne en régime permanent.
- Cette même grandeur physique doit passer d'une valeur à une autre en un temps donné, voire avec un profil de variation imposé [9] [10].

Les méthodes de l'automatique offrent la possibilité de modifier le comportement statique et dynamique d'une grandeur physique, afin qu'elle évolue conformément aux exigences de l'application.

Le concepteur d'un système de commande doit réaliser les opérations suivantes :

- ✓ Instrumenter le système : choisir les capteurs et actionneurs en fonction des besoins physiques, de coût et de performances demandées au système ;
 - ✓ Déterminer les relations entrées-sorties du système, des capteurs et des actionneurs ;
 - ✓ Modéliser quand on s'attachera à déterminer la structure mathématique de ces relations ;
 - ✓ Identifier quand on s'intéressera à calculer les coefficients du modèle ;
 - ✓ Synthétiser une loi de commande (un correcteur) afin d'obtenir un système performant (i.e., précis, rapide et stable), tout en s'affranchissant des influences néfastes des perturbations.
- Plusieurs types de régulations peuvent être utilisés.

II.4.1. Régulation par action « Tout Ou Rien »

C'est la plus simple et la plus économique des régulations. Elle est utilisée pour des installations ayant une grande inertie et n'ayant pas besoin d'une grande précision : maintien d'une température d'un four, d'un niveau dans une cuve, d'une température d'eau dans un circuit, etc.

Fonctionnement d'un régulateur TOR: Dans ce type de régulateur, la commande « U » du correcteur agit sur un relais électromécanique à contact. Dans le cas simple, lorsque $U=1$, une bobine est excitée et ferme le contact du relais pour alimenter la résistance de chauffe (bac de chauffe dans notre procès) et désexcitée lorsque $U=0$ (le contact s'ouvre alors).

II.4.2. Régulateur PID

Les correcteurs, usuellement utilisés en pratique, sont les régulateurs à effet proportionnel, intégral et dérivé (P.I.D.). Les correcteurs P.I.D. permettent d'engendrer à partir de la sortie du comparateur (c'est-à-dire l'écart existant entre la consigne et la grandeur à réguler) un signal proportionnel à l'erreur et à sa dérivée, d'une part, et à son intégrale, d'autre part [11].

Fonctionnement : Le FB41 de l'AP SIMATIC S7-300 est utilisé en tant que régulateur continu numérique. Son rôle est de calculer une valeur d'ajustement y en fonction de l'erreur (différence entre la mesure et la consigne) selon la commande PID, et de livrer cette grandeur d'ajustement y sur sa sortie analogique.

Le paramétrage du régulateur PID se fait grâce à l'outil : « **Paramétrage de la régulation PID** ». Les paramètres choisis seront sauvegardés dans le DB d'instance local associé à l'appel du FB 41. Pour cela, il est nécessaire de définir les paramètres de régulation suivants : KP (gain proportionnel), TN (temps d'intégration) et TV (temps de dérivation).

II.4.3. Le Bloc FB 41 « CONT_C »

Le bloc FB 41 « CONT_C » (continuous Controller) sert à réguler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage vous permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système régulé [11].

Vous pouvez utiliser le régulateur comme régulateur PID de maintien autonome mais aussi comme régulateur en cascade, de mélange ou de rapport dans des régulations à plusieurs boucles. Sa méthode de travail se base sur l'algorithme PID du régulateur à échantillonnage à sortie analogique, complété le cas échéant par un étage conformateur d'impulsions assurant la formation des signaux de sortie à modulation de largeur d'impulsions pour régulations à deux ou trois échelons avec organes de réglage proportionnels [9].

Description de la situation actuelle et de la problématique :

La chaine pilote se compose principalement des actionneurs (moteurs, pompes, électrovannes) et des capteurs (détecteur de niveau, transmetteurs de température) et d'un automate pour programmer le fonctionnement des équipements de cette station. La chaine pilote est commandée par un logiciel de supervision *proTool* qui est un outil d'analyse des appareils de commande pour visualiser et commander des processus. Le *protool* et le système installé dans le panel de contrôle de la station. Il est fiable, mais il a quelque manque. Car, c'est un ancien système de siemens.

Solution proposé au problème :

La production de la margarine à grand échelle de CEVITAL est commandée et supervisée à partir de la salle de contrôle avec le logiciel *wincc* qui est plus adapté, moins cher et fiable. Car, il répond aux exigences de l'entreprise. Notre travail consiste à installer ce dernier dans le pupitre et de créer un projet pour superviser le processus de la chaine pilote et, enfin, de l'injecté dans le panel de contrôle.

II.5. Conclusion

L'objectif de ce chapitre est de présenter l'environnement de notre travail, à savoir la chaine pilote pour produire de la margarine au sein de l'entreprise CEVITAL. Le principe de fonctionnement ainsi que les différents constituants ont été exposés. L'accent a été mis sur la partie régulation qui fait l'objet de notre étude.

Chapitre III

Automates et leur programmation : Cas de la station pilote

III.1. Introduction

L'automate programmable industriel API(ou Programmable Logic Controller PLC) est l'un des appareils de commande des systèmes de production et d'automatisme les plus utilisés dans l'industrie. Son apparition pour la première fois fut aux Etats-Unis dans le secteur de l'industrie automobile. Dans ce chapitre, nous donnons un bref aperçu sur la structure des systèmes automatisés ainsi que les automates. La deuxième partie est consacrée aux différents aspects de leur programmation. Le cas de la station pilote, qui fait l'objet de notre étude, est abordé dans la dernière partie du chapitre.

III.2. Systèmes automatisés

L'automatisation d'un système consiste à transformer l'ensemble des tâches de commande et de surveillance, réalisées par des opérateurs humains, dans un ensemble d'objets techniques appelés partie commande. Cette dernière mémorise le savoir-faire des opérateurs, pour obtenir l'ensemble des actions à effectuer sur la matière d'œuvre, afin d'élaborer le produit final [13]. Parmi les objectifs de l'automatisation on peut citer : réalisation des tâches répétitives, sécurité, économie des matières premières et l'énergie, augmentation de la productivité et plus d'adaptation à des contextes particuliers (flexibilité).

III.2.1. Structure des systèmes automatisés

Un système automatisé est toujours composé d'une partie commande (PC) et une partie opérative (PO). Pour faire fonctionner ce système, l'opérateur va donner des consignes à la PC. Celle-ci va traduire ces consignes en ordres qui vont être exécutées par la PO. Une fois les ordres accomplis, la PO va le signaler à la PC, par un retour d'information, qui va à son tour le signaler à l'opérateur, ce dernier pourra donc dire que le travail a bien été réalisé. [4]

Partie commande :

C'est la partie qui gère le fonctionnement du système automatisé. Elle est, en général, composée d'un ordinateur qui contient dans sa mémoire un programme. Elle transmet les ordres aux actionneurs de la partie opérative à partir :

- Du programme qu'elle contient ;
- Des informations reçues par les capteurs ;

- Des consignes données par l'utilisateur ou par l'opérateur.

Partie opérative :

Elle consomme de l'énergie électrique, pneumatique ou hydraulique. Elle comporte, en général, un boîtier (appelé bâti) contenant :

- Des actionneurs (transforment l'énergie reçue en énergie utile : moteur, vérin, pompe) ;
- Des capteurs (transforment les variations des grandeurs physiques liées au fonctionnement de l'automatisme en signaux électriques : capteur de position, de température, bouton poussoir).

Interface :

Elle relie la partie opérative (PO) et la partie commande (PC). C'est un système de traduction d'informations entre la PC et la PO.

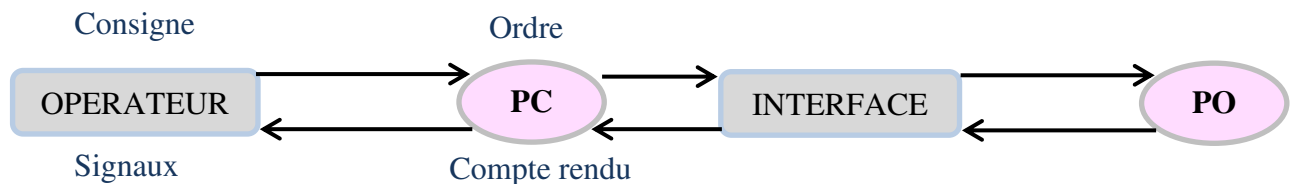


Figure III.1 : structure d'un système automatisé.

III.3. Généralités sur les API

III.3.1. Définition d'un automate programmable industriel (API)

Un automate programmable est un système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel. Il utilise une mémoire programme pour le stockage interne des instructions utilisées aux fins de la mise en œuvre des fonctions spécifiques, telles que : des fonctions logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander, au moyen des entrées/sorties (de type tout/rien ou analogiques), de 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 utilisé dans toutes leurs fonctions prévues.

Un API a trois caractéristiques fondamentales :

- Il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses E/S industrielles ;

- Il est conçu pour fonctionner dans des ambiances industrielles sévères ;
- Enfin, sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme facilite son exploitation et sa mise en œuvre.

III.3.2. Architecteur des automates

L'architecture du processeur d'un automate programmable est fondamentalement la même que celle d'un ordinateur à usage général. Néanmoins, il existe certaines caractéristiques importantes qui les distinguent. Tout d'abord, contrairement aux ordinateurs, les automates programmables sont conçus pour résister aux conditions difficiles de l'environnement industriel. Un automate bien conçu peut être placé dans une zone avec d'importantes quantités: de bruit électrique, des interférences électromagnétiques, des vibrations et d'humidité sans condensation. Une deuxième distinction des automates est que leurs matériels et logiciels sont conçus pour une utilisation facile par les électriciens et les techniciens. Les interfaces matérielles pour la connexion d'appareils de terrain sont en réalité partie de l'automate lui-même et se connectent facilement. Les circuits d'interface modulaires et autodiagnostic sont en mesure d'identifier les dysfonctionnements et, d'ailleurs, ils sont facilement enlevés et remplacés. En outre, la programmation du logiciel utilise des symboles traditionnels relais d'échelle, ou d'autres langues apprises facilement, qui sont familières au personnel de l'usine. Alors que les ordinateurs sont des machines informatiques complexes capables d'exécuter plusieurs programmes ou tâches simultanément et dans n'importe quel ordre. La norme PLC exécute un programme unique dans un mode séquentiel ordonné de la première à la dernière instruction.

III.3.3. Aspect extérieur des API

Les automates peuvent être de type compact ou modulaire.

- **De type compact** : on distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouz,...) des micros automates. Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, ajout d'entrées/sorties analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de Petits automatismes.

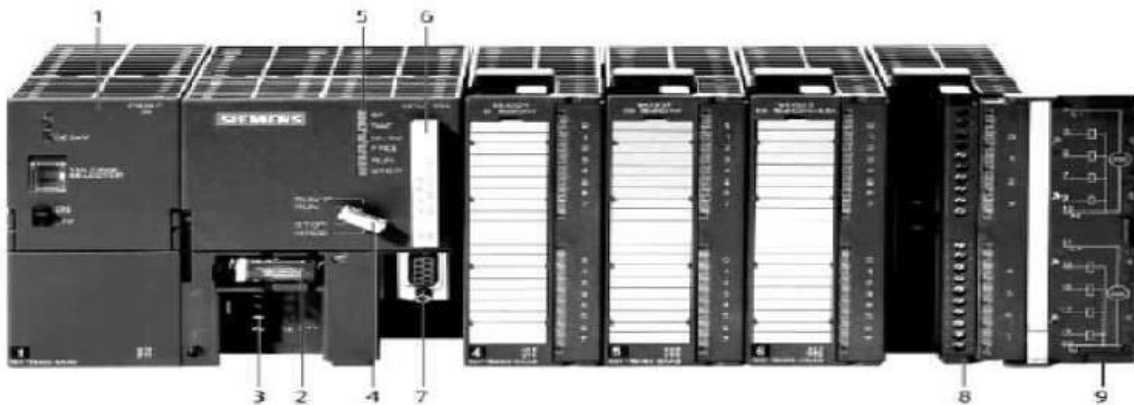
- **De type modulaire** : le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "Fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où puissants, où la capacité de traitement et flexibilité sont nécessaires.



Figure III.2 : Automate compact (Allen-Bradley)



Figure III.3 : Automate Modulaire (Modicon).



- | | |
|--|-------------------------------|
| 1. Module d'alimentation | 6. Carte mémoire |
| 2. Pile de sauvegarde | 7. Interface multipoint (MPI) |
| 3. Connexion au 24V cc | 8. Connecteur frontal |
| 4. Commutateur de mode (à clé) | 9. Volet en face avant |
| 5. LED de signalisation d'état et de défauts | |

Figure III.4 : Automate modulaire (Siemens).

III.3.4. structure interne d'un automate

Les API comportent quatre parties principales : une mémoire, un processeur, des interfaces d'entrées/sorties et d'une alimentation ($240V_{ac}$, $24V_{cc}$).

Ces quatre parties sont reliées entre elles par des bus (ensemble de câbles autorisant le passage de l'information entre ces 4 secteurs de l'API) [6].

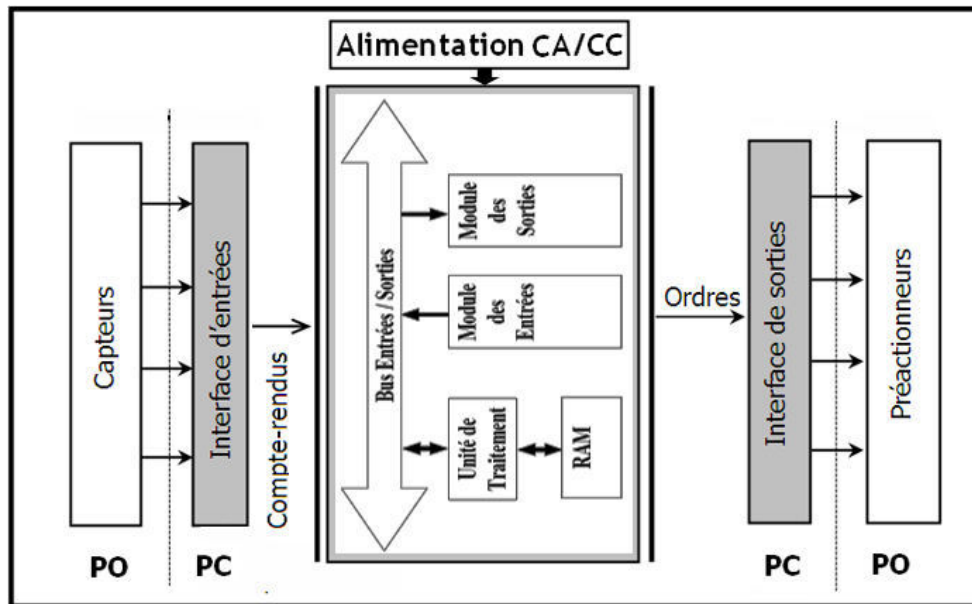


Figure III.5 : Structure interne d'un API.

III.3.5. Description des éléments d'un API

A- L'alimentation :

L'alimentation du système joue un rôle majeur dans le fonctionnement total du système. En fait, il peut être considéré comme le "gestionnaire de premier niveau" de la fiabilité et l'intégrité du système. Sa fonction n'est pas seulement de fournir des tensions continues internes pour les composants du système, mais aussi pour surveiller et réguler les tensions fournies et prévenir la CPU si quelque chose ne va pas. Le bloc d'alimentation a pour fonction de fournir une puissance bien régulée et de protection pour les autres composants du système.

B- Unité centrale ou CPU :

L'unité centrale commande l'interprétation et l'exécution des instructions programmées. Elle est aussi chargée de détecter les pannes de communication, ainsi que d'autres défaillances qui peuvent survenir pendant le fonctionnement du système. Il doit alerter l'opérateur ou le système en cas de dysfonctionnement. A base de microprocesseur, l'unité centrale réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...). Ce module se compose essentiellement de :

Microprocesseur : Il constitue le cœur de la CPU. Son rôle consiste, d'une part, à organiser les différentes relations entre la zone mémoire et les interfaces d'E/S et, d'autre part, à gérer les instructions du programme.

Bus : C'est un ensemble de pistes conductrices (pistes en cuivre) par lesquelles s'achemine une information binaire (suite de 0 ou 1), c'est-à-dire ensemble de fils autorisant le passage des informations entre les quatre secteurs (l'alimentation, la mémoire, le processeur et l'interface E/S) de l'automate. L'unité centrale dispose de trois bus : bus de données, bus d'adresses et bus de commandes.

Mémoire : Elle est conçue pour recevoir, gérer et stocker des informations issues des différents secteurs du système qui sont :

- Le terminal de programmation.
- Le processeur, qui lui gère et exécute le programme.

Elle reçoit également des informations en provenance des capteurs.

Interfaces d'entrées / sorties : Les interfaces d'entrées / sorties permettent à l'unité centrale de communiquer avec l'environnement ou les périphériques.

C- Modules entrées/sorties

Les modules d'E/S assurent le rôle d'interface entre le procédé à commander et la CPU. On distingue deux types :

Module entrées/sorties TOR (Tout ou Rien): La gestion de ce type de variables, constituant le point de départ des API, reste l'une de leurs activités majeures. Leur nombre est en générale de 8, 16, 24 ou 32 entrées/sorties, qui peuvent fonctionner:

- en continue 24V, 48V.
- en alternative 24V, 48V, 100V/120V, 200V/240V.

Module entrées/sorties analogique: Elles permettent l'acquisition de mesures (entrées analogiques) et générer des signaux de commande (sorties analogiques). Ces modules comportent un ou plusieurs convertisseurs analogique/numériques (A/N) pour les entrées et numériques/analogiques (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 utiliser 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.

III.3.6. Cycle d'un API [7]

Tous les automates fonctionnent selon le même mode opératoire :

- **Traitement interne :** l'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN/STOP....) ;
- **Lecture des entrées :** l'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées ;
- **Exécution du programme :** l'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties ;
- **Ecriture des sorties :** l'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

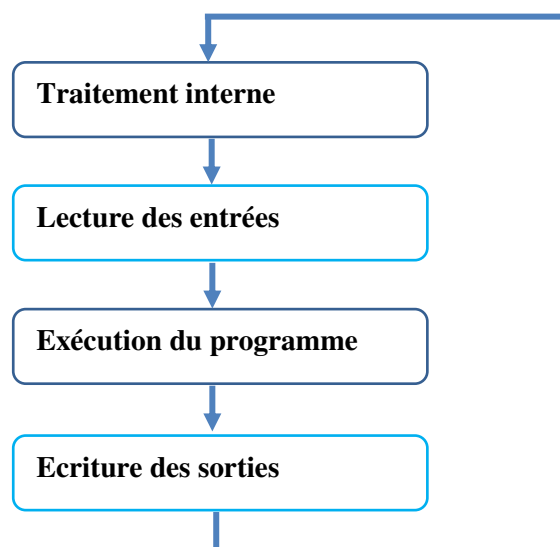


Figure III.6 : Cycle d'un API.

III.3.7. Critères de choix d'un API

Le choix d'un automate programmable est, en premier lieu, le choix d'une société ou d'un groupe où les contacts commerciaux et expériences vécues sont déjà un point de départ. Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir "se retourner" en cas de "perte de vitesse" de l'une d'entre elles.

Le personnel de maintenance doit toutefois être formé sur ces matériels et une très grande diversité des matériels peut avoir de graves répercussions. Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions. La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables.

Il faut ensuite quantifier les besoins :

- 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 ;
- 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).

III.3.8. Présentation de l'automate utilisé S7-300

La famille S7-300 est constituée d'automates programmables, de conception modulaire, utilisés pour des automatismes de gamme et peuvent être connectés entre eux au moyen d'un câble-bus PROFIBUS.

Un automate S7-300 consiste en une CPU, un module d'alimentation PS, un module de signaux FM, un module de signaux SM et un processeur de communication.

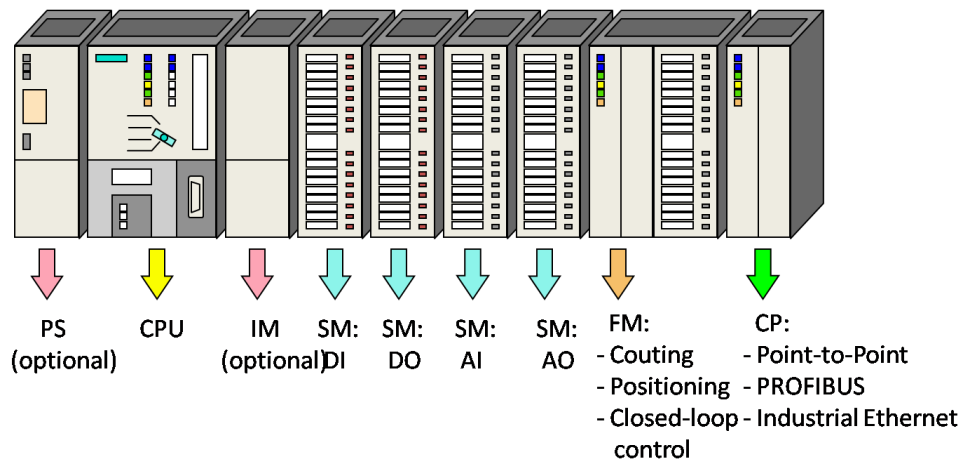


Figure III.7 : Constitution d'un API S7-300.[8]

Module de fonction (FM) : Il a pour rôle, l'exécution de tâche de traitement des signaux du processus à temps critique et nécessitant une importante capacité de mémoire comme le comptage, positionnement et régulation.

Module de communication (CP) : Ils permettent d'établir des liaisons homme-machine qui sont effectuées par les interfaces de communications : Point à point, Profibus et l'Ethernet Industriel;

Caractéristiques de la CPU : La gamme S7-300 offre une grande variété de CPU tels que la CPU312, 314M, 315 ,315-2P, etc. Chacune possède certaines caractéristiques différentes des autres. Par conséquent, le choix de la CPU, pour un problème d'automatisation donné, est conditionné par les caractéristiques offertes par la CPU choisie.

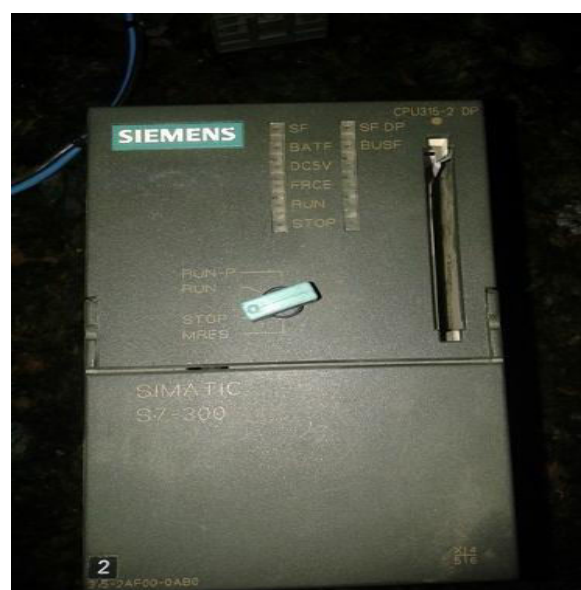


Figure III.8: CPU 315-2DP S7-300.

Interface (MPI) : Une liaison MPI (Multi Point Interface) est nécessaire pour programmer un SIMATIC S7 300 depuis le PC. Elle est une interface de communication utilisée pour la programmation, le contrôle-commande avec HMI et l'échange de données entre CPU CIMATIC S7 jusqu'à 32 nœuds maximum. Chaque CPU du SIMATIC S7-300 est équipée d'une interface MPI intégré.

Signification d'état : La CPU comporte des LED de signalisation suivante:

- **SF (rouge) :** signalisation groupée de défauts qui s'allume si on a de défauts matériels et en cas d'erreurs de programmation, de paramétrage, de calcul, etc.
- **BATF (rouge) :** défaut pile qui s'allume si elle est défectueuse, absente ou déchargée.
- **DC5V (verte) :** alimentation 5Vcc pour la CPU et le bus S7-300 qui s'allume si les 5V sont présente et elle clignote s'il y a surcharge de courant.
- **FRCE (jaune) :** forçage permanent qui s'allume en cas de forçage permanent.
- **RUN (verte) :** état de fonctionnement RUN qui clignote en cas de démarrage de la CPU.
- **STOP (jaune) :** état de fonctionnement STOP qui s'allume si la CPU ne traite aucun programme utilisateur et clignote en cas ou la CPU demande un effacement général.

Les LED de signalisation de défauts **SF-DP** et **BUSF** ne se rencontrent que dans le cas de la CPU314 relative à la configuration maitre-esclave du S7-300.

Commutateur de mode :

- **RUN-P (mode de fonctionnement RUN programme):** la CPU traite le programme utilisateur et la clé ne peut être retirée. Il est possible de lire le programme de la CPU avec une PG (CPU vers PG) et de transférer des programmes dans la CPU (PG vers CPU)
- **RUN (mode de fonctionnement RUN) :** la CPU traite le programme de l'utilisateur. dans cette position, la clé peut être retirée pour éviter qu'une personne non habilitée change le mode de fonctionnement.
- **STOP (mode de fonctionnement STOP) :** la CPU ne traite aucun programme utilisateur, la clé peut être retirée pour éviter le changement de mode inattendue mais on peut lire et écrire dans la CPU.

- **MRES** : c'est position instable du commutateur de mode de fonctionnement, en vue de l'effacement générale de la CPU. Le contenu de la mémoire de chargement rémanente intégré reste inchangé après un effacement général.

La pile : Elle permet de sauvegarder le contenu de la RAM en cas de coupure du courant.

Application pratique : Chaque processus industriel de fabrication ou de transformation se compose d'un ensemble de machines destinées à réaliser la fabrication ou la transformation considérée. Chaque machine ou partie opérative comprend un ensemble de moteurs, vannes et autres dispositifs qui lui permet de fonctionner. Ces actionneurs sont pilotés par un automate (partie commande) à travers les pré-actionneurs.

La partie commande reçoit des informations transmises par un opérateur en fonctionnement normal, ou un dépanneur en cas de réglage ou de mauvais fonctionnement de la partie commande ou de la partie opérative ainsi que par les capteurs. Entre la partie commande et l'homme se trouve la partie dialogue qui permet à ce dernier de transmettre au moyen de dispositifs adaptés (bouton poussoirs, commutateurs,...etc). De même, la partie commande renvoie vers l'homme les informations sous des formes compréhensibles (voyant, afficheurs, cadrans,...etc).

Ainsi, entre l'homme et la partie opérative, s'instaure un dialogue homme machine dont l'importance naguère sous-estimé et aujourd'hui reconnue, et qui est actuellement l'objet de nombreuses études.

III.4.Programmation via STEP 7

III.4.1. Définition du logiciel STEP7

STEP 7 est le logiciel de base pour la configuration et la programmation des systèmes SIMATIC (S7-300). Il fait partie de l'industrie logicielle SIMATIC. Les tâches de bases 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 [9].

III.4.2. Langages de programmation sous STEP7

Pour créer un programme sous STEP7, on dispose de trois langages de programmation CONT, LIST ou LOG qui peuvent être combinés dans le même programme. Selon la spécialité du programmeur, il choisira le modèle qui lui convient.

A- Programmation à schéma logique (LOG) : Le logigramme (LOG) est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les blocs logiques[10].



Figure III. 9 : présentation d'un schéma logique (LOG).

B- Programmation à schéma contact (CONT) : C'est un langage de programmation graphique. La syntaxe des instructions sont très semblable aux schémas de circuits électriques. Le langage a CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts, les éléments complexes et les bobines[10].

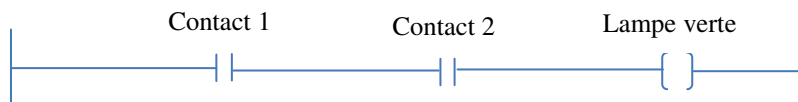


Figure III.10 : Présentation d'un schéma CONT.

C- Programmation à liste (LIST) : C'est un langage de programmation textuel proche de la machine. Dans cette programmation, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme. Pour faciliter la programmation, LIST a été complété par quelques structures de langage évolué (paramètres de blocs et accès structurés aux données) [10].

U	Commutateur 1
U	Commutateur 2
=	Lampe verte

Figure III.11 : Présentation du langage LIST.

III.4.3. Structure du programme STEP7

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].

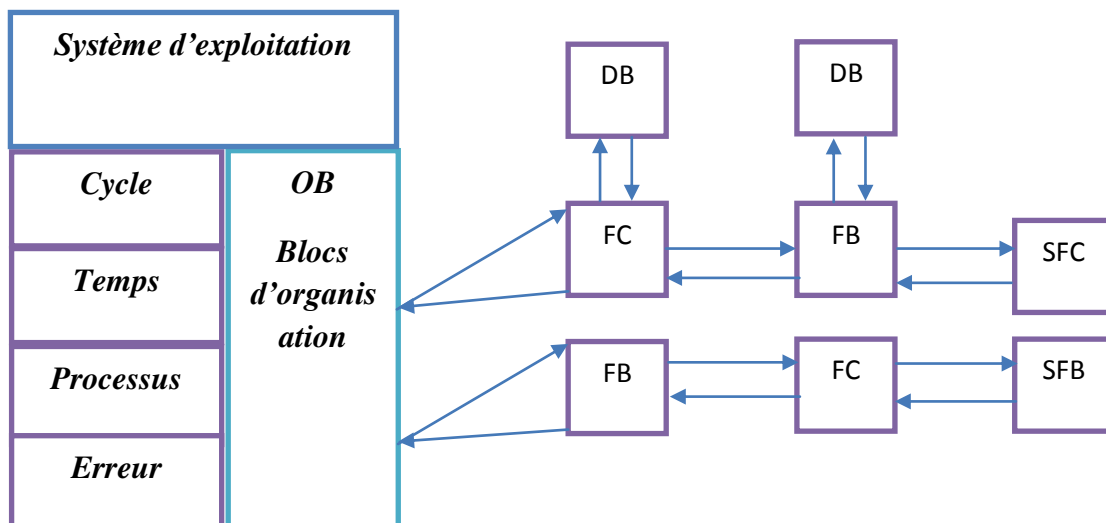


Figure III.12: Blocs de programme.

III.4.4.Blocs

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.

A- Bloc d'organisation : Les blocs d'organisation (**OB**) constituent l'interface entre le système **OB** d'exploitation et le programme utilisateur. L'ensemble du programme peut être concaténé dans un seul bloc OB1 (programme linéaire) appelé de manière cyclique par le système d'exploitation, ou être structuré dans plusieurs blocs (programme structuré) [11].

B- Fonction : Une fonction (**FC**) assure une fonctionnalité spécifique du programme. Les fonctions peuvent être paramétrables. Dans ce cas, des paramètres sont transmis à la fonction lorsqu'elle est appelée. Les fonctions conviennent pour la programmation de fonctionnalités récurrentes et complexes (*e.g.*, effectuer des calculs). Les fonctions système (**SFC**) sont des fonctions paramétrables, intégrées au système d'exploitation de la CPU, dont le numéro et la fonctionnalité sont définis de manière fixe. Pour de plus amples informations, se reporter à l'aide en ligne [10].

C- 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).

D- Blocs fonctionnels système (SFB) : Ce sont des blocs fonctionnels paramétrables, intégrés au système d'exploitation de la CPU, dont le numéro et la fonctionnalité sont définis de manière fixe. Pour de plus amples informations, se reporter à l'aide en ligne.

E- 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 de blocs de données :

- *Bloc de données globales :* Contrairement aux blocs de code, les blocs de données ne contiennent pas d'instruction STEP7. Ils servent à l'enregistrement des données du

programme utilisateur pouvant être utilisées par tous les autres blocs. Ils contiennent des données variables que le programme utilisateur utilise [4] ;

- *Bloc de données d'instance* : Un bloc de donnée d'instance est associé à chaque appel de bloc fonctionnel transmettant des paramètres. Ces blocs contiennent les paramètres effectifs et les données statiques du **FB**. Les variables déclarées dans le **FB** déterminent la structure du bloc de donnée d'instance. L'instance est l'appel d'un bloc fonctionnel.

III.4.5. 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.

III.4.6. 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 [9].

III.4.7. 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 III.1 : Différents types de variables contenues dans le STEP7 [11].

Groupe	Types de données	signification
Type de données binaires	BOOL BYTE WORD DWORD	Les données de ce type occupent 1 bit, 8bits, 16bits ou 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.

III.5. Développement du programme pour la station pilote

III.5.1. Démarrage du logiciel STEP7

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.

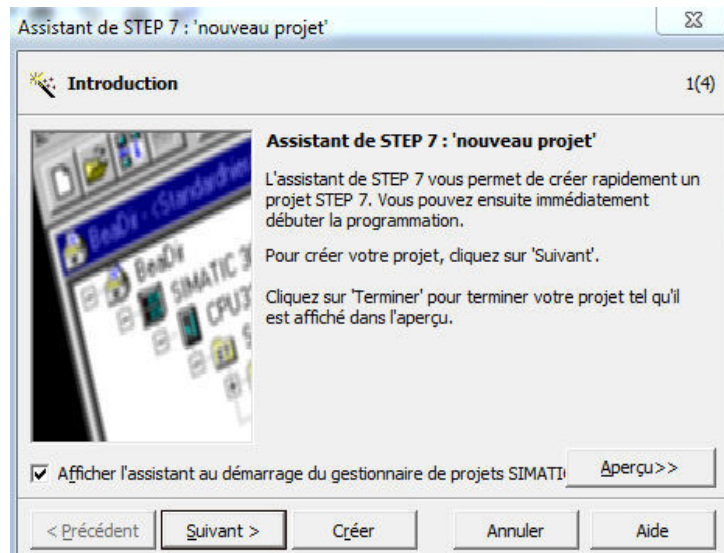


Figure III.13 : Assistant nouveau projet

III.5.2. Création d'un projet STEP7

Le logiciel SIMATIC Manager étant maintenant ouvert, on clic sur l'item fichier puis assistant nouveau projet. Après la sélection du type de la CPU (pour notre projet, on a choisi une **CPU315-2 DP**) et l'insertion du bloc d'organisation et le langage à liste, une fenêtre s'ouvre pour donner un nom au projet. Pour notre cas (Station pilote) et on clic sur créer.

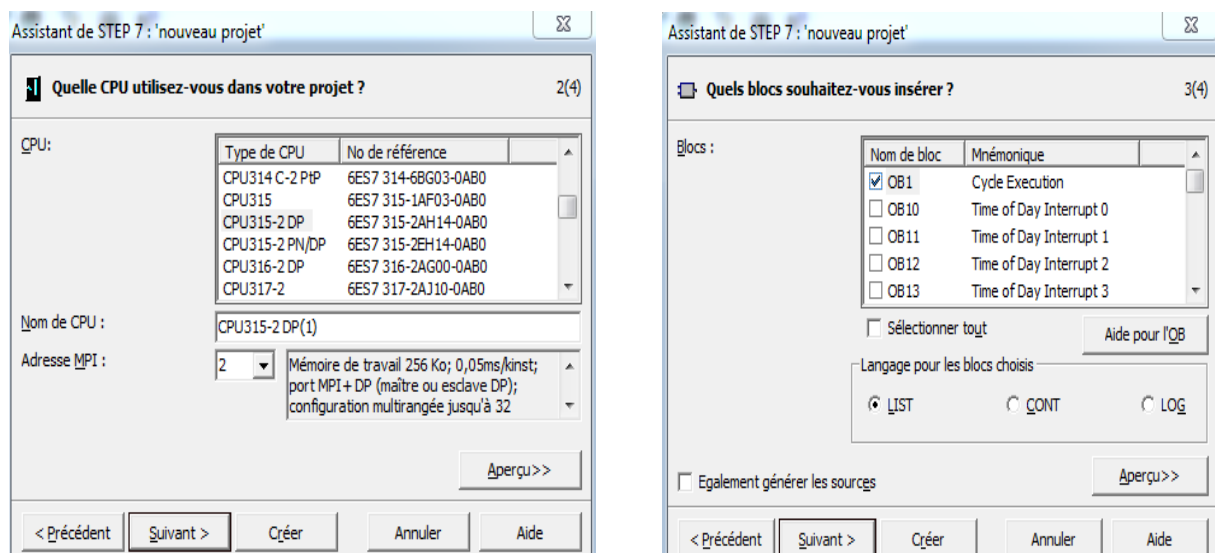


Figure III.14 : Choix de la CPU et du bloc d'organisation.

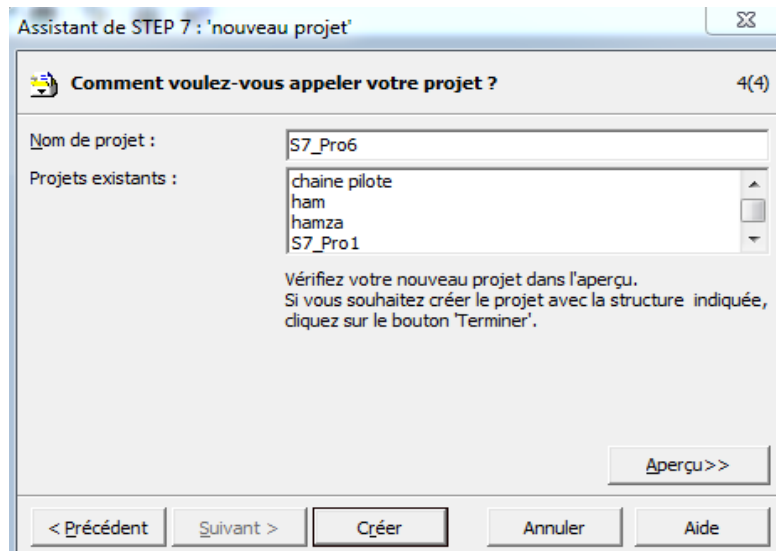


Figure III.15 : Nom et création du projet.

III.5.3. Configuration matériel

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.

Une configuration matériels est nécessaire pour :

- Les paramètres ou les adresse préréglé d'un module.
- Configurer les liaisons de communication.

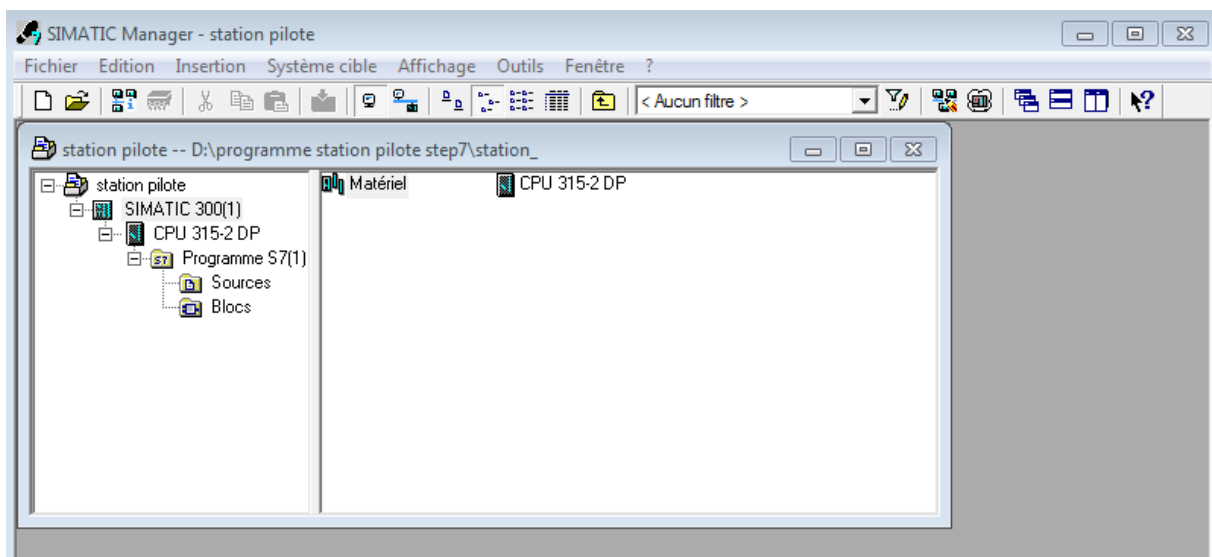


Figure III.16: Station SIMATIC S7-300.

Le choix du matériels SIMATIC S7-300 avec une CPU315-2DP, nous conduit à introduire la hiérarchie suivante :

- On commence par le choix du châssis selon la station choisie auparavant. Pour la station SIMATIC S7-300, on aura « RACK-300 » et on le glisse dans le châssis ;
- Après avoir choisie le RACK, on lui glisse la CPU315-2DP dans l'emplacement N°2 ;
- L'emplacement N°1 est réservé pour l'alimentation et le N°3 réservé comme adresse logique pour un coupleur dans une configuration multi-châssis ;
- 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 deux pour les sorties ;
- Après la configuration des modules, on va créer un réseau maître DP pour la configuration des variateurs de vitesse des actionneurs (moteurs, pompes) qui sont reliés au profibus de l'automate.

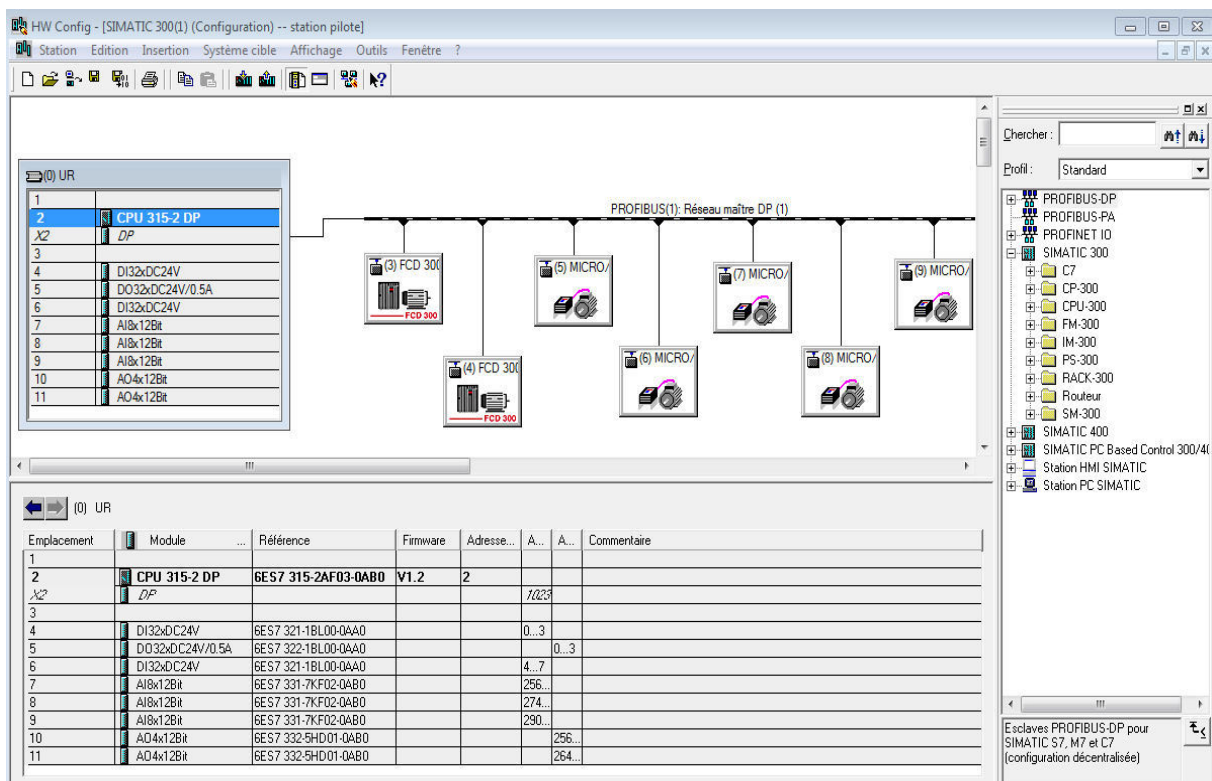
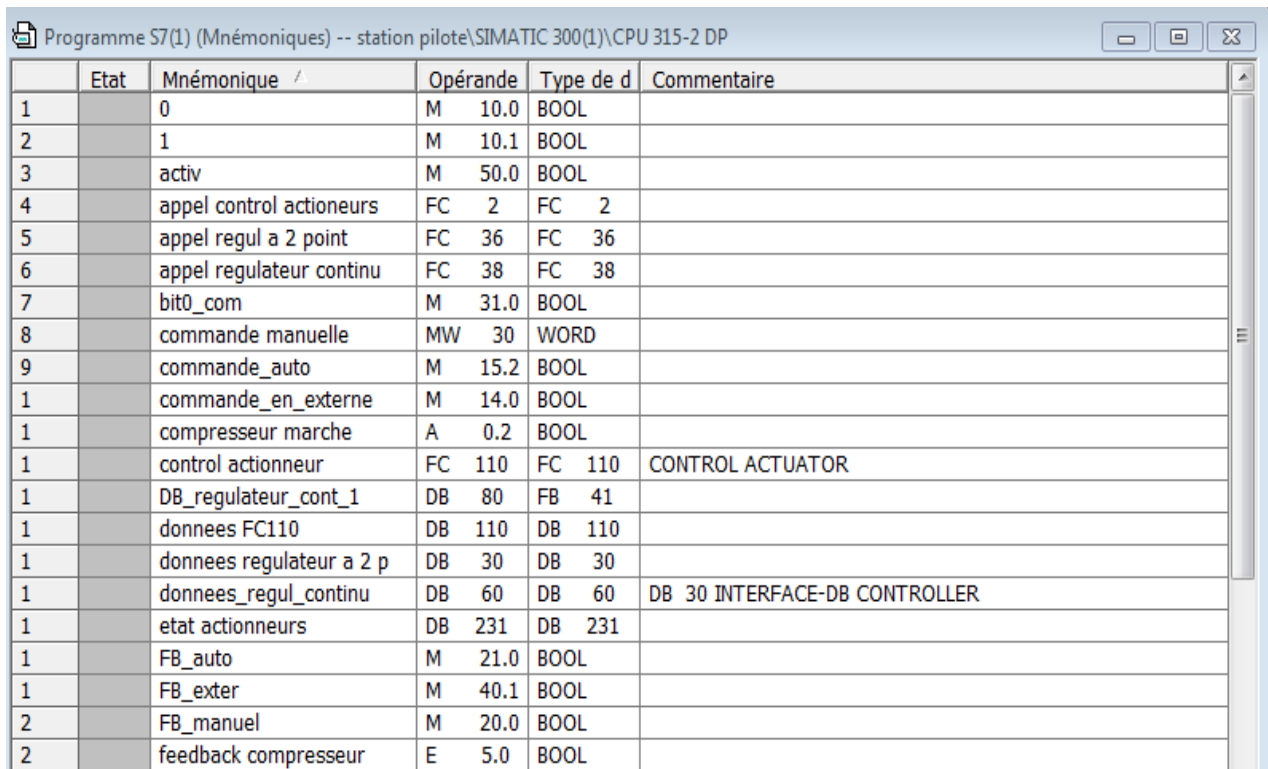


Figure III.17 : Configuration matériels.

III.5.4. Table des mnémoniques

Une table des mnémoniques vide est automatiquement générée lors de la création d'un programme STEP7. Elle se trouve dans le menu <programme>table des mnémonique. En programmant sur STEP7, nous travaillons avec des opérands tels que les E/S, mémentos, compteurs, temporisations, blocs de données et fonctions. Nous pouvons les adresser de manière absolue dans le programme. Mais, nous pouvons aussi améliorer considérablement la lisibilité et la clarté d'un programme en utilisant des mnémoniques à la place des adresses absolues.



	Etat	Mnémonique /	Opérande	Type de d	Commentaire
1		0	M 10.0	BOOL	
2		1	M 10.1	BOOL	
3		activ	M 50.0	BOOL	
4		appel control actioneurs	FC 2	FC 2	
5		appel regul a 2 point	FC 36	FC 36	
6		appel regulateur continu	FC 38	FC 38	
7		bit0_com	M 31.0	BOOL	
8		commande manuelle	MW 30	WORD	
9		commande_auto	M 15.2	BOOL	
1		commande_en_externe	M 14.0	BOOL	
1		compresseur marche	A 0.2	BOOL	
1		control actionneur	FC 110	FC 110	CONTROL ACTUATOR
1		DB_regulateur_cont_1	DB 80	FB 41	
1		donnees FC110	DB 110	DB 110	
1		donnees regulateur a 2 p	DB 30	DB 30	
1		donnees_regul_continu	DB 60	DB 60	DB 30 INTERFACE-DB CONTROLLER
1		etat actionneurs	DB 231	DB 231	
1		FB_auto	M 21.0	BOOL	
1		FB_exter	M 40.1	BOOL	
2		FB_manuel	M 20.0	BOOL	
2		feedback compresseur	E 5.0	BOOL	

Figure III.18 : Une partie de la table des mnémoniques du projet.

III.5.5. Création du programme

Notre programme se compose d'un bloc d'organisation OB1, de fonctions FC, de blocs de données globales et d'instance DB, et d'un bloc fonctionnel FB41.

Nom de l'objet	Nom symbolique	Langage de création	Taille dans la mémoire...	Type	Version (en-tête)
Données système	SDB	...
OB1		CONT	70	Bloc d'organisation	0.1
FB41	regulateur PID	SCL	1462	Bloc fonctionnel	1.5
FC2	appel control actioneurs	LIST	266	Fonction	0.1
FC10	fonction impulsions 1s	LIST	64	Fonction	0.1
FC36	appel regul a 2 point	LIST	176	Fonction	0.1
FC38	appel regulateur continu	LIST	226	Fonction	0.1
FC110	control actionneur	LIST	620	Fonction	0.0
FC176	regulateur a 2 points	LIST	886	Fonction	0.0
FC178	regulateur_continu	LIST	1290	Fonction	0.0
DB30	donnees regulateur a 2 p	DB	244	Bloc de données	0.1
DB40	valeur analogique proces	DB	58	Bloc de données	0.1
DB60	donnees_regul_continu	DB	244	Bloc de données	0.0
DB80	DB_regulateur_cont_1	DB	162	DB d'instance du FB...	0.1
DB110	donnees FC110	DB	44	Bloc de données	0.1
DB111	temps FC 110	DB	44	Bloc de données	0.1
DB230	manuel activation	DB	44	Bloc de données	0.1
DB231	etat actionneurs	DB	44	Bloc de données	0.1

Figure III.19 : les blocs de programme.

III.5.5.1. Programme de FC2

Après avoir validé notre projet et configuré le matériel, on va créer des fonctions qui sont programmées en langage LISTE vu que certaines instructions du programme qu'on a utilisées ne sont pas acceptées en langage CONT.

Dans notre programme on a 2 fonctions qui sont programmées pour le contrôle des 27 actionneurs de la station, FC110 « contrôle actionneur » qui est appelé dans la fonction FC2 « appel contrôle actionneur » pour le fonctionnement de chaque actionneur.

On a choisi comme exemple le programme de la pompe de bac de chauffe PW01 :

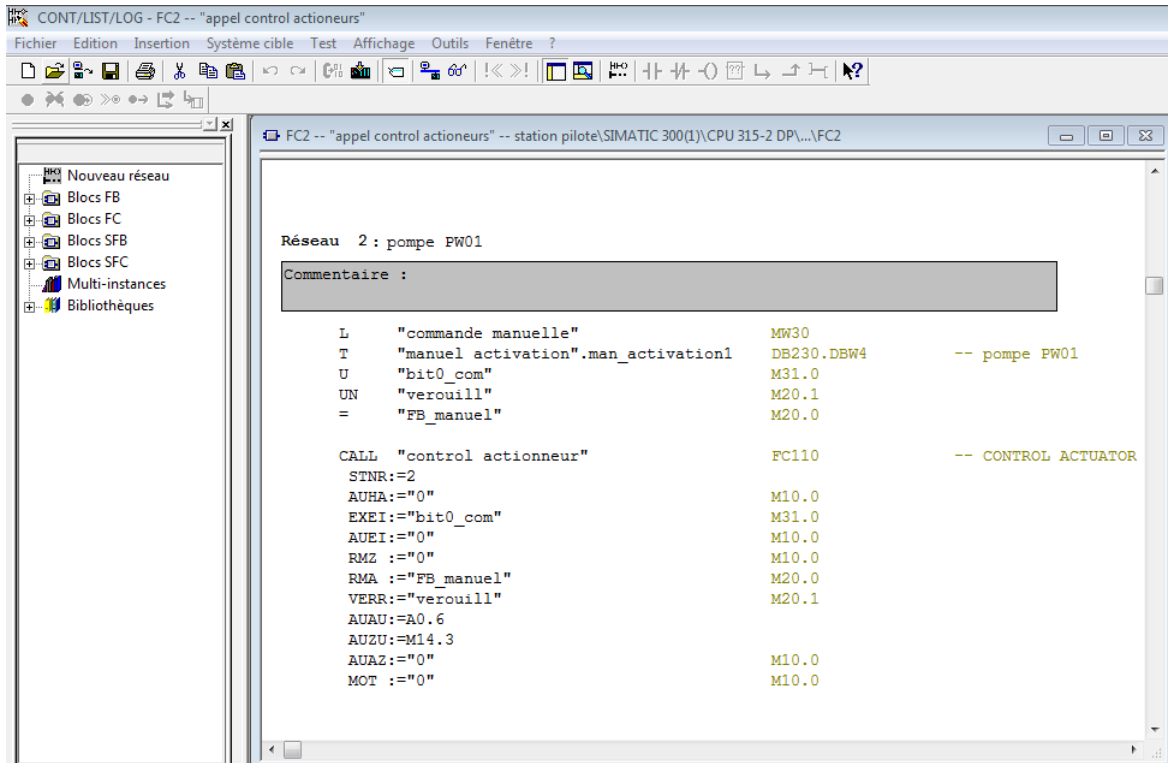


Figure III.20 : programme de la pompe de bac de chauffe PW01.

Signification des paramètres de FC110 :

- STNR : Adresse ou numéros de l'actionneur.
- AUHA : Mode de fonctionnement de l'actionneur, 1=AUTO/ 0=manuel.
- EXEI : Mode de fonctionnement externe.
- AUEI : Entré de mode d'activation automatique.
- RMZ : Entré de feedback fermé.
- RMA : Entré de feedback ouvert.
- VERR : Verrouillage.
- AUAU : Sortie d'activation de l'actionneur.
- AUZU : Sortie d'activation fermée.
- AUAZ : Entré d'addition fermée.
- MOT : Entré d'entrainement de l'actionneur ouvert et fermé.

III.5.5.2. Programme de FC38

La fonction FC178 « régulateur continu » qui est programmé pour le contrôle des 3 régulateurs continu est appelée dans la fonction FC38 « appel régulateur continu » pour programmer le fonctionnement de chaque régulateur.

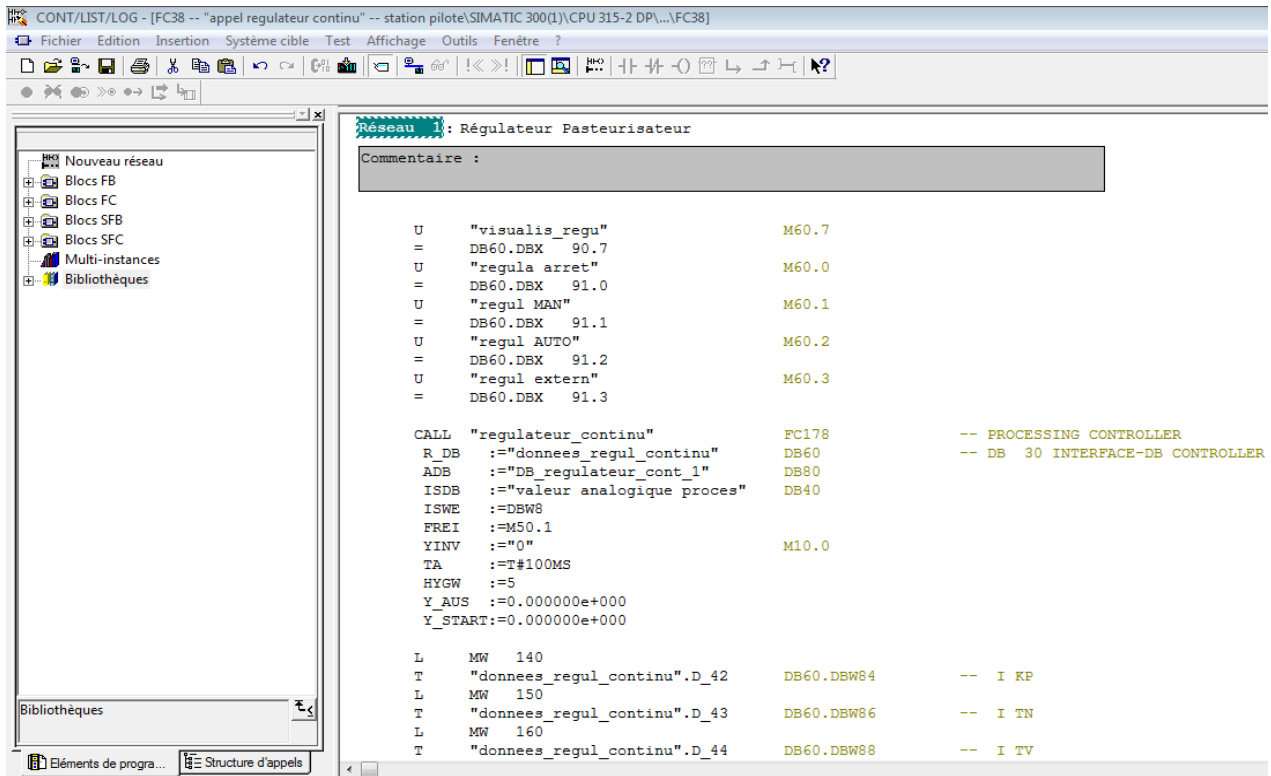


Figure III.21 : Programme de régulateur continu de pasteurisateur.

La fonction FC38 utilise un bloc fonctionnel FB41 qui se compose des paramètres de régulation PID qui sont enregistré dans son bloc de donnée d’instance.

The screenshot shows the 'DB-Param - DB80' window in SIMATIC Manager, displaying the instance data for block FB41. The table lists various parameters with their addresses, data types, and initial values.

Adresse	Décl.	Nom	Type	Valeur initiale	Valeur en cou	Commentaire	
1	0.0	in	COM_RST	BOOL	FALSE	FALSE	complete restart
2	0.1	in	MAN_ON	BOOL	TRUE	TRUE	manual value on
3	0.2	in	PVPER_...	BOOL	FALSE	FALSE	process variable peripherie on
4	0.3	in	P_SEL	BOOL	TRUE	TRUE	proportional action on
5	0.4	in	I_SEL	BOOL	TRUE	TRUE	integral action on
6	0.5	in	INT_HOLD	BOOL	FALSE	FALSE	integral action hold
7	0.6	in	I_ITL_ON	BOOL	FALSE	FALSE	initialization of the integral action
8	0.7	in	D_SEL	BOOL	FALSE	TRUE	derivative action on
9	2.0	in	CYCLE	TIME	T#1S	T#100MS	sample time
10	6.0	in	SP_INT	REAL	0.000000e+...	8.000000e+...	internal setpoint
11	10.0	in	PV_IN	REAL	0.000000e+...	2.050000e+...	process variable in
12	14.0	in	PV_PER	WORD	W#16#0	W#16#0	process variable peripherie
13	16.0	in	MAN	REAL	0.000000e+...	0.000000e+...	manual value
14	20.0	in	GAIN	REAL	2.000000e+...	1.000000e+...	proportional gain
15	24.0	in	TI	TIME	T#20S	T#8S	reset time
16	28.0	in	TD	TIME	T#10S	T#4S	derivative time
17	32.0	in	TM_LAG	TIME	T#2S	T#2S	time lag of the derivative action
18	36.0	in	DEADB_W	REAL	0.000000e+...	0.000000e+...	dead band width
19	40.0	in	LMN_HLM	REAL	1.000000e+...	1.000000e+...	manipulated value high limit
20	44.0	in	LMN_LLM	REAL	0.000000e+...	0.000000e+...	manipulated value low limit
21	48.0	in	PV_FAC	REAL	1.000000e+...	1.000000e+...	process variable factor
22	52.0	in	PV_OFF	REAL	0.000000e+...	0.000000e+...	process variable offset
23	56.0	in	LMN_FAC	REAL	1.000000e+...	1.000000e+...	manipulated value factor
24	60.0	in	LMN_OFF	REAL	0.000000e+...	0.000000e+...	manipulated value offset
25	64.0	in	I_ITLVAL	REAL	0.000000e+...	0.000000e+...	initialization value of the integral action
26	68.0	in	DISV	REAL	0.000000e+...	0.000000e+...	disturbance variable
27	72.0	out	LMN	REAL	0.000000e+...	0.000000e+...	manipulated value
28	76.0	out	LMN_PER	WORD	W#16#0	W#16#0	manipulated value peripherie
29	78.0	out	QLMN_...	BOOL	FALSE	FALSE	high limit of manipulated value reached
30	79.1	out	QLMN_L	BOOL	FALSE	TRUE	low limit of manipulated value reached

Figure III.22 : Bloc de donnée d’instance de FB41

Avant de charger et de compiler le programme de la station il faut appeler toutes les fonctions dans le bloc d'organisation OB1.

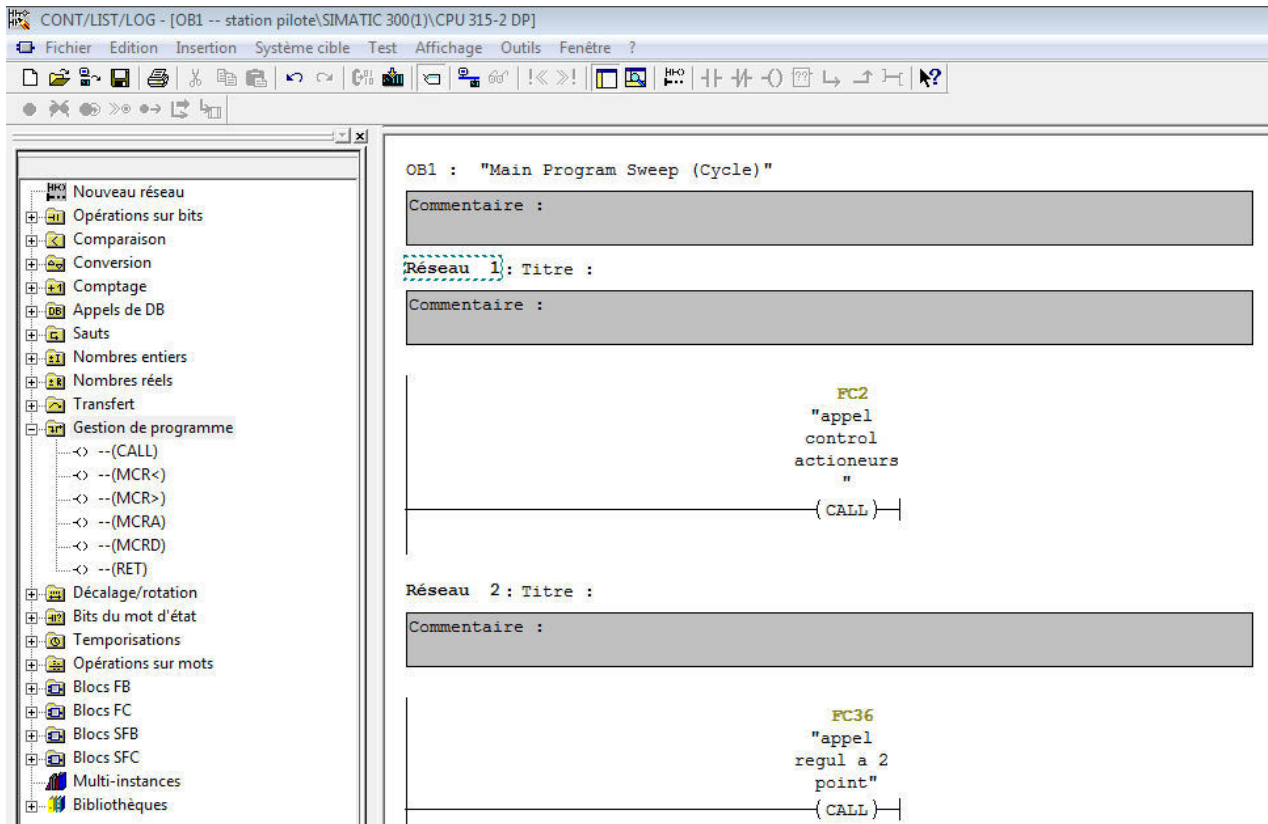


Figure III.23 : OB1 station pilote.

Le reste du programme réaliser est présenté dans l'**annexe 2**.

III.6. Conclusion

Dans ce chapitre, on a présenté un aperçu sur les systèmes automatisés et les API. Notre travail est basé sur l'automate S7-300 de SIEMENS ainsi sur le logiciel associé STEP7. La dernière partie est consacrée à la configuration du matériel de la station pilote et création du programme permettant de gérer le fonctionnement de ses actionneurs. Les tests et la simulation et les résultats seront dans le chapitre IV.

Chapitre IV

Supervision et simulation

IV.1. Introduction

Le maximum de transparence est essentiel pour l'opérateur qui travaille dans un environnement où les processus sont de plus en plus complexes. La supervision est une technique industrielle qui consiste à représenter, surveiller et diagnostiquer l'état de fonctionnement d'un procédé automatisé, pour obtenir un fonctionnement optimal. Le but est de disposer, en temps réel, d'une visualisation de l'état d'évolution des paramètres du processus. Ce qui permet à l'opérateur de prendre rapidement des décisions appropriées à ses objectifs, telles que : la cadence de production, la qualité des produits et la sécurité des biens des personnes [12].

L'objectif de ce chapitre est l'étude d'un système de supervision avec le logiciel WinCC flexible pour visualiser l'état de fonctionnement de la station pilote afin de surveiller et de détecter les anomalies qui peuvent survenir au cours de fonctionnement du procédé.

Une autre partie est consacrée pour la simulation du programme élaboré pour la station, ainsi que son système de supervision en utilisant l'application « S7-PLCSIM » de STEP7 et « Runtime » de logiciel WinCC flexible.

IV.2. Supervision

IV.2.1. Définition de la supervision

La supervision est une forme évoluée de dialogue Homme-Machine. Elle sert à représenter et à surveiller l'état de fonctionnement d'un procédé. Le système assure aussi un rôle du gestionnaire des alarmes, d'archivage pour la maintenance, le traçage des courbes, l'enregistrement des historiques des défauts et le suivi de la production. Les fonctions de la supervision sont nombreuses, on peut citer :

- Elle répond à des besoins nécessitant, en général, une puissance de traitement importante ;
- Elle assure la communication entre les équipements d'automatisme et les outils informatiques de gestion de la production ;

- Coordonne le fonctionnement d'un ensemble de machines enchaînées constituant une ligne de production, en assurant l'exécution d'ordres communs (marche, arrêt,...) et des tâches telles que la synchronisation ;
- L'opérateur assiste dans les opérations de diagnostic et de maintenance.

IV.2.2. Architecture d'un réseau de supervision

En vue de la réalisation d'une communication entre un API et un PC, des protocoles d'échange ont été développés dans ce sens, pour assurer l'échange de données entre le PC de supervision et un automate programmable. Le choix d'un réseau de communication dépend principalement des besoins en termes de couverture géographique, de qualité de données et de nombre d'abonnés. Le PC de supervision n'échange pas directement les données avec les capteurs ou les actionneurs du procédé à superviser, mais à travers l'API qui gère l'ensemble du processus.

Un réseau de supervision est souvent constitué de :

- Un PC utilisé comme poste opérateur qui permet l'acquisition des données, l'affichage des synoptiques et la conduite de l'unité ;
- Un PC comme poste ingénieur, dédié à l'administration du système et au paramétrage de l'application ;
- Un réseau d'acquisition de type MPI, reliant les postes opérateur de l'automate.

IV.2.3. Avantage de la supervision

Un système de supervision donne de l'aide à l'opérateur dans la conduite du processus. Son but est de présenter à l'opérateur des résultats expliqués et interprétés. Ses avantages principaux sont : la surveillance du processus à distance, la détection des défauts et le diagnostic et le traitement des alarmes.

IV.3. Présentation du logiciel WinCC flexible

Lorsque la complexité des processus augmente et que les machines et les installations doivent répondre à des spécifications de fonctionnalité toujours plus sévères, l'opérateur a besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'interface homme-machine (IHM) [12].

L'IHM constitue l'interface entre l'homme (opérateur) et le processus (Machine/Installation). Le contrôle proprement dit du processus est assuré par le système d'automatisation. Il existe par conséquent une interface entre l'opérateur et le WinCC (Windows Contrôle Center) flexible et entre le WinCC flexible et le système d'automatisation. L'IHM se charge des tâches suivantes :

- **Visualisation du processus:** le processus est visualisé sur le pupitre opérateur lorsque l'état de ce dernier évolue.
- **Conduite du processus :** l'opérateur peut contrôler le processus au moyen de l'interface utilisateur graphique. Il peut définir une valeur de consigne pour un automate ou démarrer un moteur par exemple.
- **Affichage des alarmes :** lorsqu'une anomalie survient, cela déclenche automatiquement une alarme.
- **Archivage des valeurs du processus et d'alarmes :** Le système IHM peut identifier les alarmes et les valeurs du processus. Cette fonction permet d'enregistrer des séquences de traitement et d'extraire des données de production antérieures.

Le WinCC flexible est le logiciel IHM pour la réalisation, par des moyens d'ingénierie simples et efficaces, des concepts d'automatisation évolutifs au niveau machine. Il réunit différents avantages, simplicité, ouverture et flexibilité.

IV. 3.1. Création du projet avec WinCC flexible

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

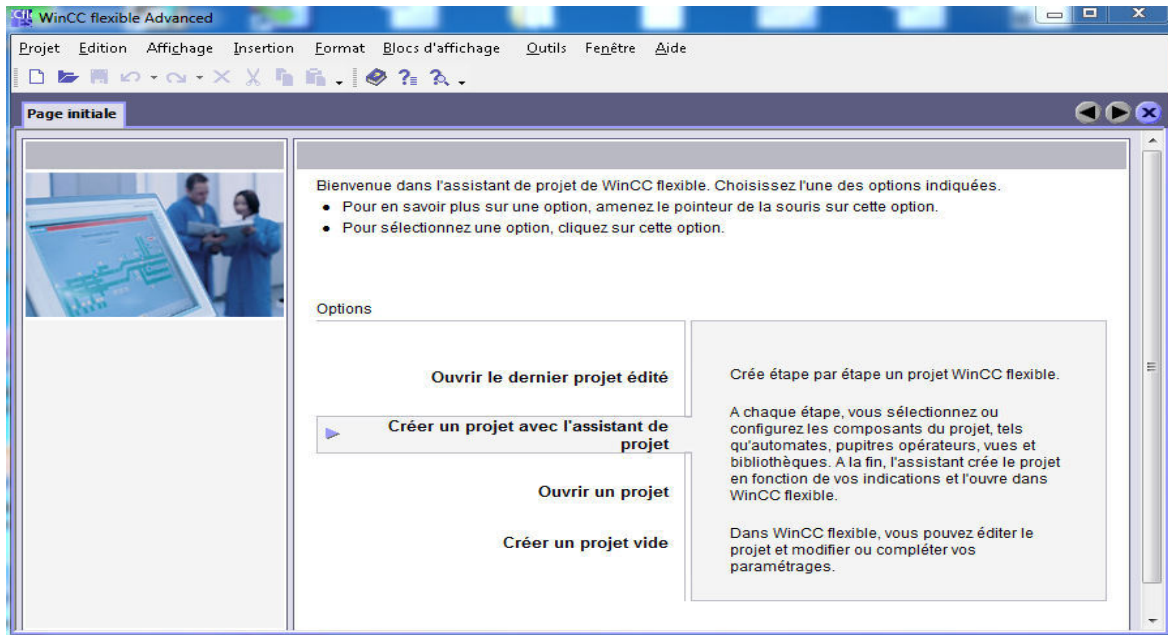


Figure IV.1 : Fenêtre principale du logiciel WinCC flexible.

IV.3.2. Choix du pupitre et de l'automate

Pour notre projet, le choix est porté sur le panel TP 270 10" et l'automate S7 300.

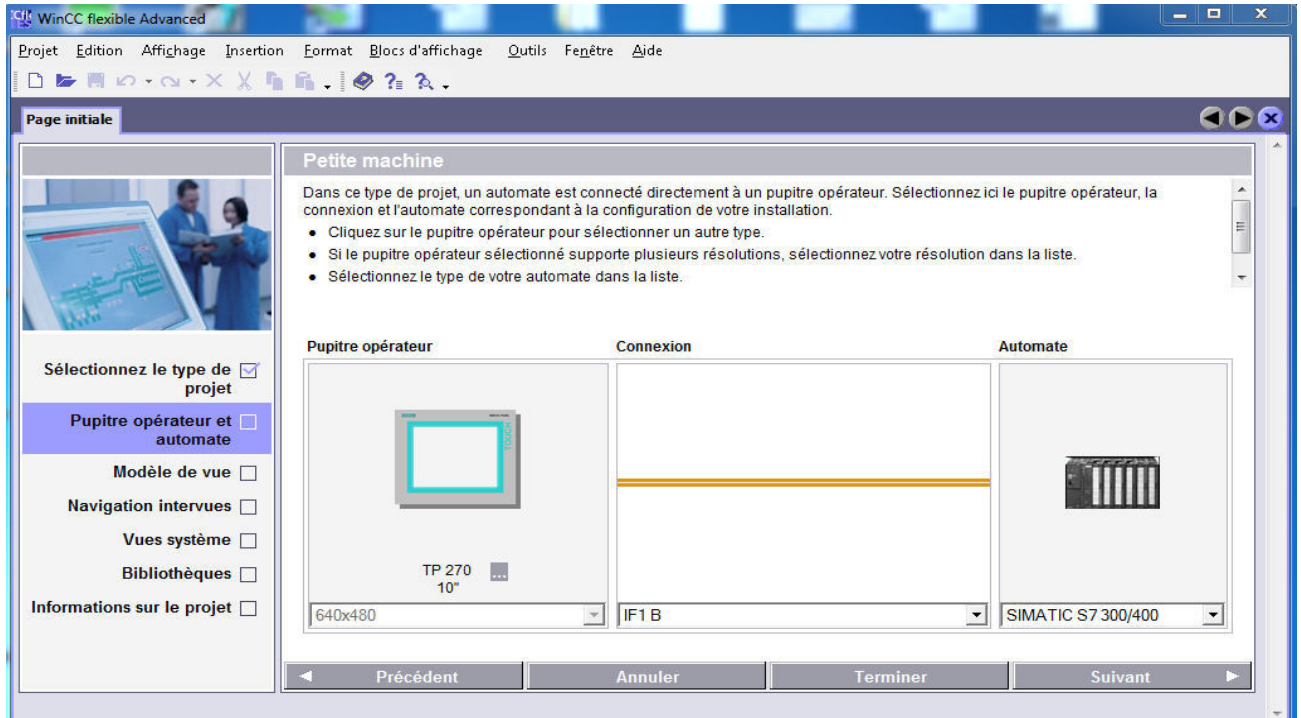


Figure IV.2 : Choix du pupitre et de l'automate.

IV.3.3. La mise en route du WinCC flexible

Après avoir lancé et configuré le logiciel WinCC flexible, ce dernier mettra à disposition une boîte d'outils qui contient les différents éléments pour la réalisation d'un projet, une bibliothèque, une zone de travail et une fenêtre de projet contenant l'ensemble des vues (accueil, paramètres, alarmes et courbes).

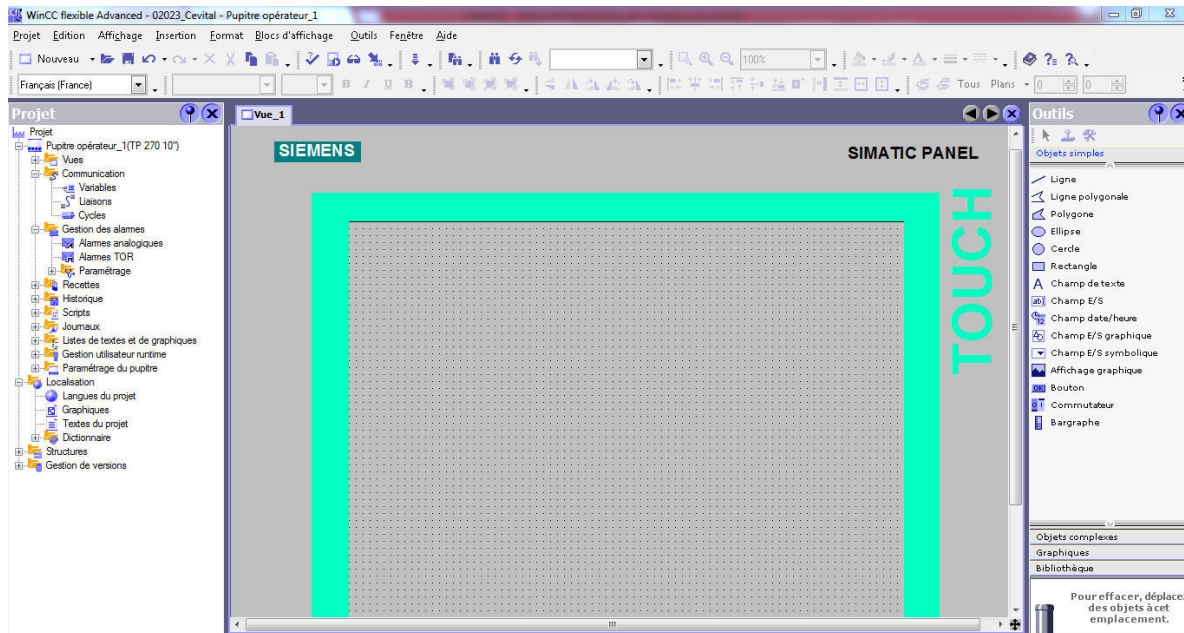


Figure IV.3 : Fenêtre d'une vue dans le winCC flexible.

IV.3.4. Configuration des vues via WinCC flexible

Pour la configuration des vues, nous disposons de fonctions conviviales telles que l'agrandissement/réduction, la rotation et l'orientation. Le WinCC flexible nous offre la possibilité d'adapter l'environnement de travail à nos besoins. Lors de l'ingénierie, un environnement contextuel, adapté à la tâche de configuration considérée, s'affiche sur l'écran de l'ordinateur de configuration. Nous trouverons tous ce que nous avons besoin pour travailler avec le maximum de confort :

- *La fenêtre de projet* : pour la représentation de la structure du projet et sa gestion ;
- *La boîte à outils* : avec différents objets et l'accès à la bibliothèque des objets ;
- *La fenêtre d'objets* : permettant la sélection d'objets déjà créés (et leur copie dans l'image par glisser-lâcher) ;

- *La zone de travail* : dans laquelle il est possible de créer des vues (graphiques et animations) ;
- *La fenêtre des propriétés* : pour le paramétrage des objets dans la zone de travail. Pour l'animation d'objet, une configuration est appliquée dans la fenêtre d'objet qui est composée de quatre éléments : <<Générale, Propriétés, Animation, Evénement>>.

IV.3.5. Création des vues du projet de la station pilote

Pour ce projet, nous avons créé une vue de modèle pour que les autres vues ressemblent à celle-ci, pour nous faciliter la tâche et ajouter des objets de supervision à ces vues. Figure IV.4 illustre le modèle qu'on a créé.

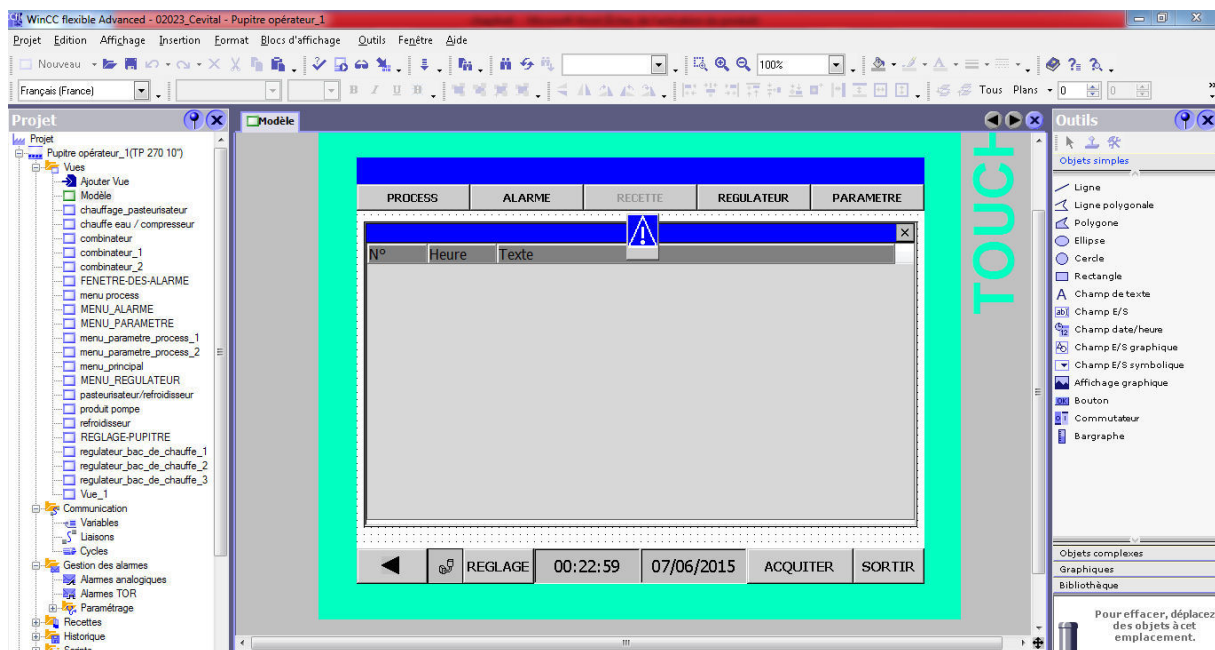


Figure IV.4 : La vue modèle de la station.

Dans ce modèle, on trouve des boutons pour le réglage du pupitre (contraste, date et heures,...), acquitter des alarmes, sortir pour revenir au menu principal. Le bouton *Process* fait afficher un menu des 4 parties de la station, pour le contrôle et la supervision du processus :

- *Produit pompe* : on a montré les deux bacs d'émulsion qui ne sont pas reliés à l'automate, plus les deux pompes (centrifuge et à piston) qui sont contrôlées à partir de cette vue.

- *Pasteurisateur/Refroidisseur* : Dans cette vue, on a créé un refroidisseur et pasteurisateur avec son moteur et un champ entrée/sortie pour la vitesse de rotation réelle et la consigne ainsi la limite de courant pour ce dernier (même chose pour tous les moteurs de la station).
- *Combinateur/Cristaliseur* : Pour cette vue, on a créé deux combinateurs avec leurs moteurs et leurs vannes d'entrée et de sortie de l'ammoniac. A la sortie du combinateur, la margarine passe dans un cristaliseur qu'on a créé dans cette vue aussi.
- *Chauffe-eau/compresseur* : La station à 3 bacs de chauffe avec résistances et pompes, on a créé ces 3 bacs avec 3 champ entrées/sorties pour les températures réelles et la consigne de l'eau. On a aussi créé un compresseur ou on peut juste voir l'état du moteur (marche, arrêt, alarme, verrouillage).

La Figure IV.5 montre la vue de combinateur/cristaliseur de la station. Les autres parties de la station sont présenté dans l'**annexe 2**.

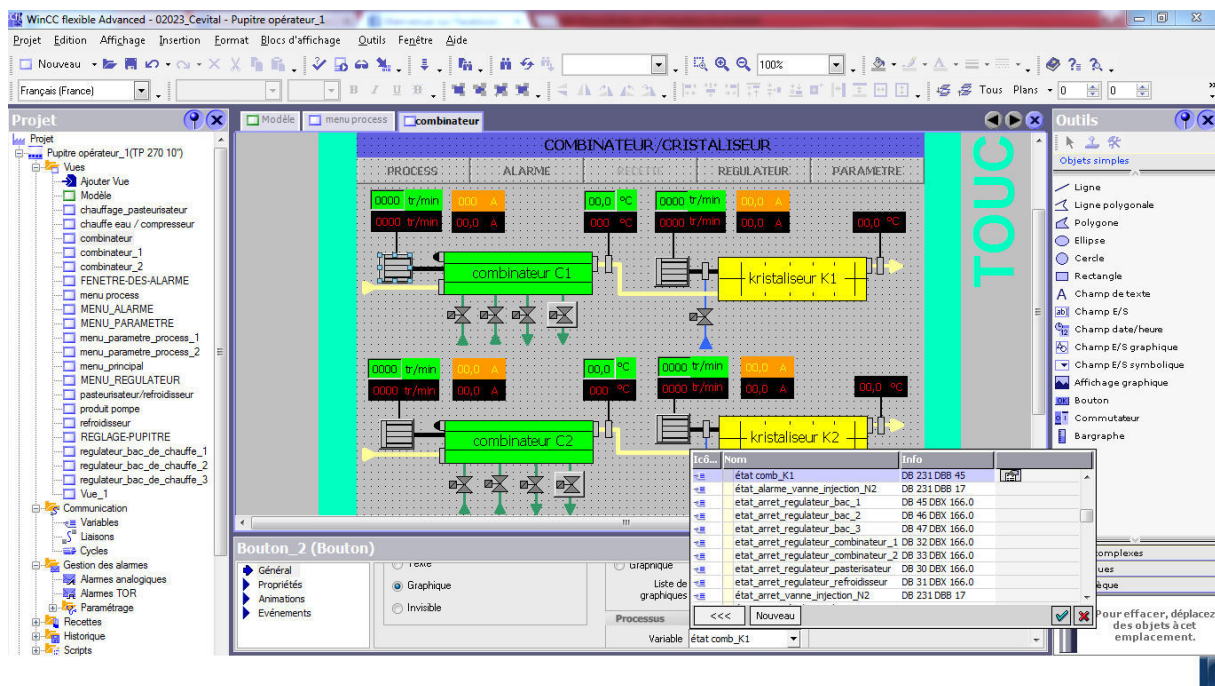


Figure IV.5 : Création et configuration de la vue combinateur/cristaliseur.

Pour les autres boutons de la vue modèle, on a ALARME pour afficher l'historique des alarmes. Le bouton REGULATEUR pour afficher les 7 régulateurs de la station avec leurs courbes. Le bouton PARAMETRE affiche les deux menus paramètres processus pour donner les limites de courant, de la pression et du temps à ne pas dépasser. Figure IV.6 montre la création et la configuration du régulateur de combinateur K1.

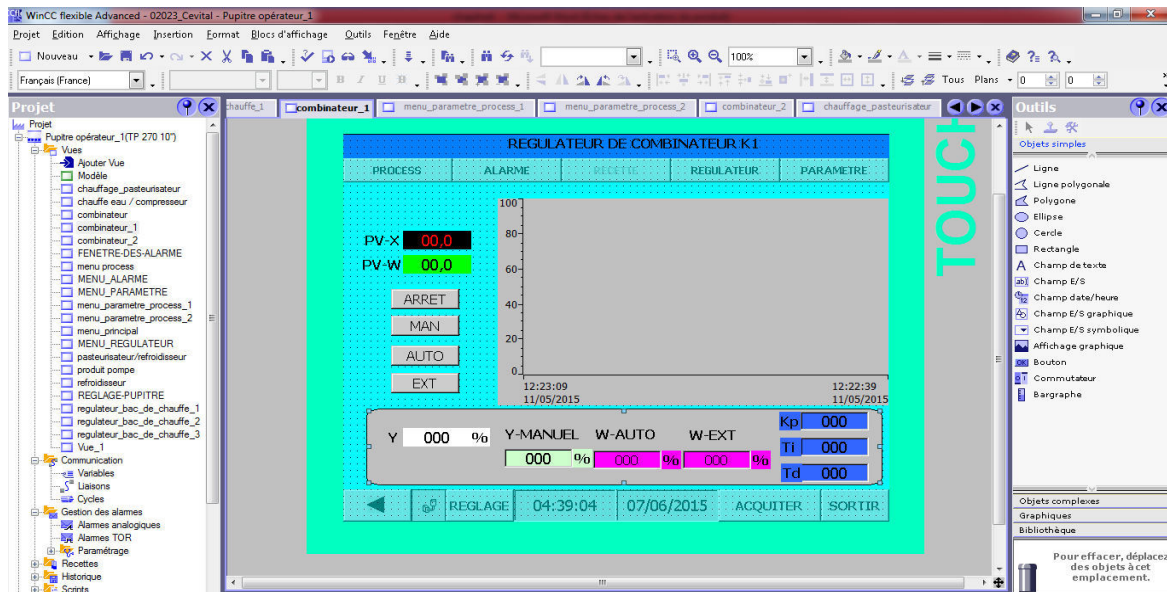


Figure IV.6 : Création de la vue de régulateur de combinatueur K1.

IV.3.6. Intégration du projet WinCC flexible dans le projet step7

Une solution d'automatisation complète est composée d'une IHM, telle que le WinCC flexible, d'un système d'automatisation, d'un bus système et d'une périphérie. Dans la barre des menus du projet WinCC flexible, on clic sur projet<<intégré dans le projet step7>>. Ensuite, on sélectionne notre projet STEP7, dans lequel on veut intégrer notre projet IHM (Figure IV.7).

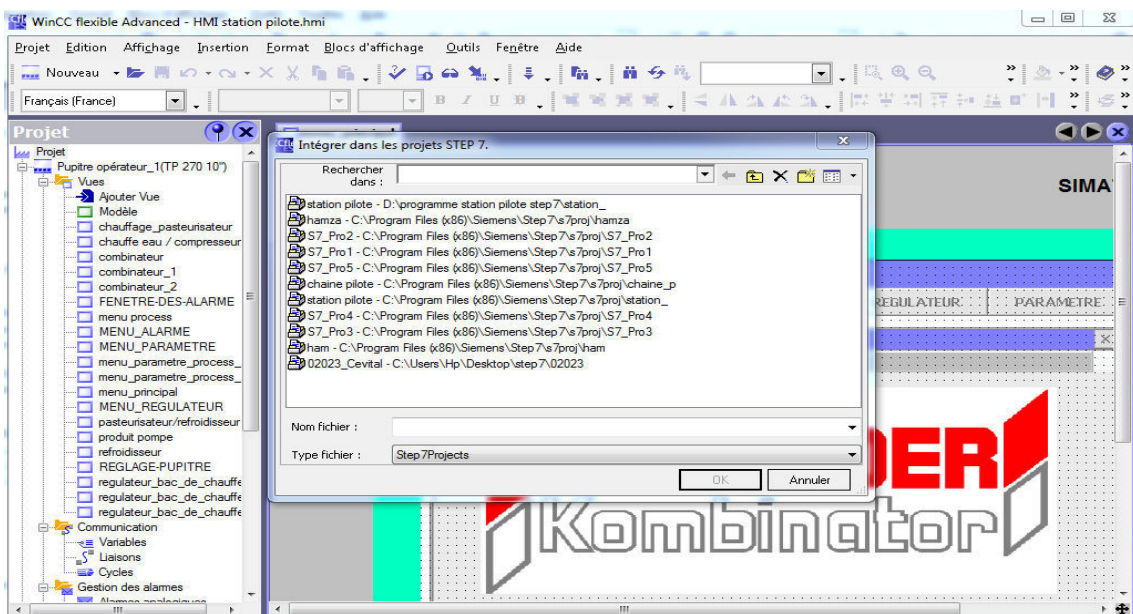



Figure IV.7 : Intégration du projet IHM dans le projet STEP7.

IV.3.7. Configuration du réseau

Dans la barre d'outils de STEP7, on clic sur l'icône  NETPRO pour relié le réseau MPI (la ligne rouge) avec SIMATIC 300 et SMATIC WinCC (Pupitre operateur).

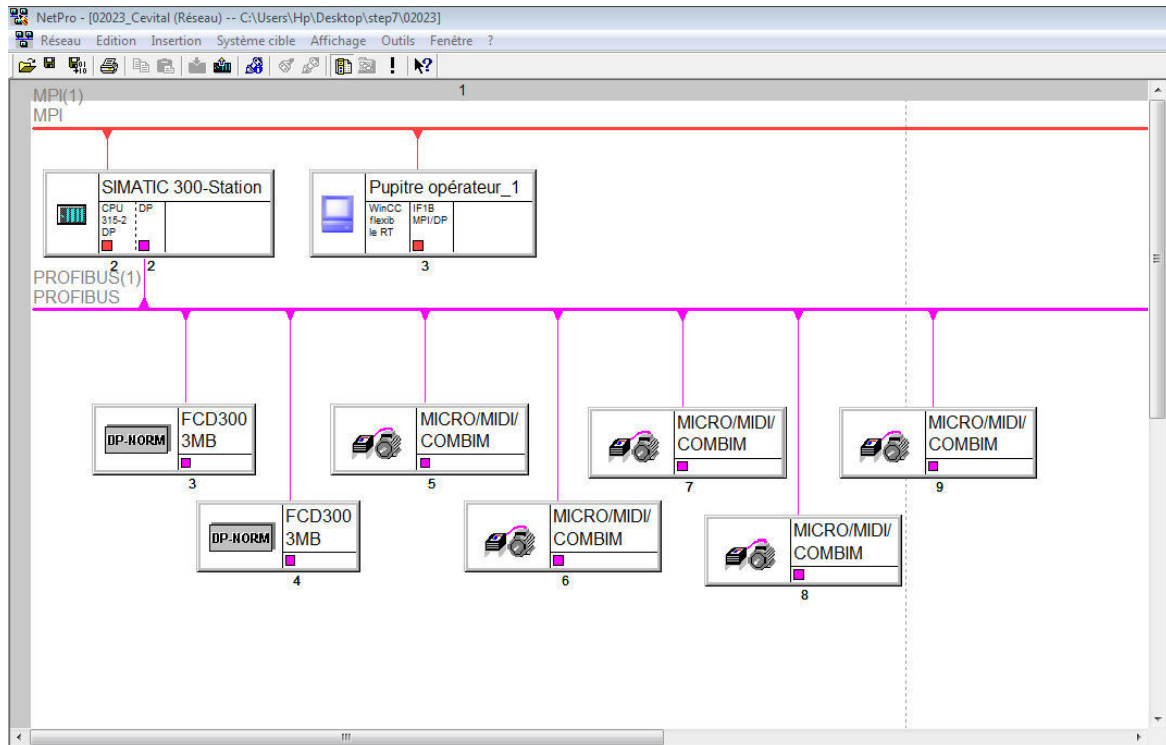


Figure IV.8 : Configuration du réseau MPI.

IV.3.8. Interface multipoint MPI

L'interface multipoint MPI est une interface propriétaire des automates programmables industriels SIMATIC S7 de la compagnie SIEMENS. Elle est utilisée pour le raccordement des stations de programmation (PC), des pupitres opérateurs, ainsi que d'autres appareils appartenant à la famille SIMATIC avec l'automate programmable.

IV.3.9. Activer la liaison

Pour activer la liaison entre le pupitre et l'automate, il faut aller dans WinCC flexible RT>> communication >> liaison (choisir l'automate, le projet step7 et activé la liaison).

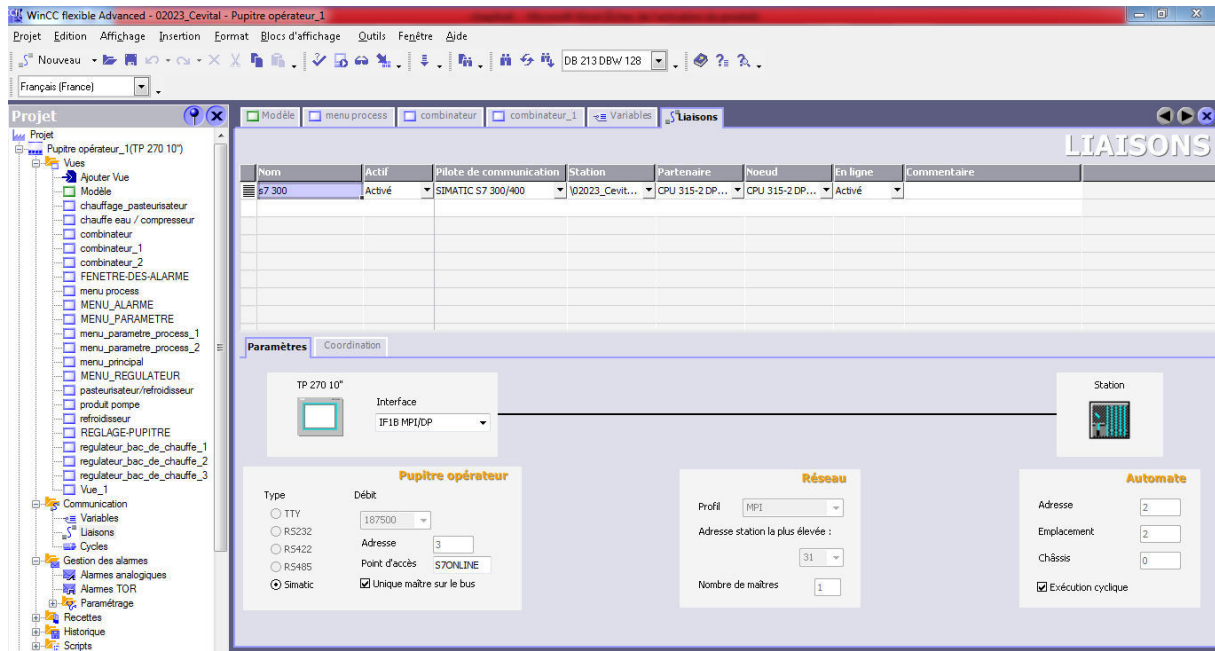



Figure IV.9 : Configuration de la liaison entre le pupitre et l'automate.

IV.4. Simulation du projet

IV.4.1. Simulation sous STEP7

Dans le logiciel STEP7, on trouve l'application **S7-PLCSIM** qui permet de tester et d'exécuter le programme utilisateur STEP7 via un automate programmable simulé (virtuel). Le S7-PLCSIM permet aussi de détecter des erreurs logiques de configurations ou de programmations.

IV.4.1.1. Ouverture et configuration du S7-PLCSIM

Pour ouvrir et activer le S7-PLCSIM, on clic sur l'icone  qui se trouve dans la barre d'outils de STEP7. Ensuite, on va créer de nouvelles fenêtres à partir de la barre d'outils, permettant de modifier l'état des entrées/sorties intervenants dans le programme.

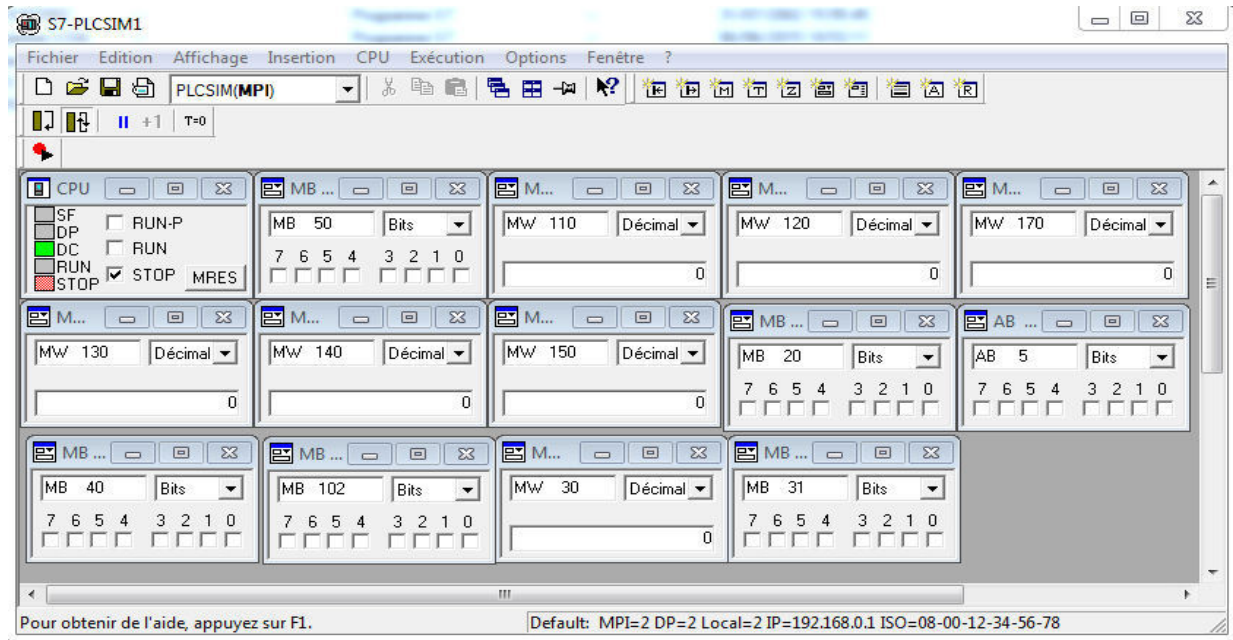




Figure IV.10 : Configuration du S7-PLCSIM.

IV.4.1.2. Chargement du programme et la mise en ligne

Avant le chargement, on doit sélectionner le bloc qui contient le programme. Le chargement se fait, en appuyant sur l’icone  qui se trouve dans la barre d’outils de STEP7. Cette action permet de charger le programme dans la CPU pour son exécution.

Pour la mise en ligne du programme on clic sur l’icone  qui se trouve dans la barre d’outils qui permet d’afficher le programme dans l’automate.

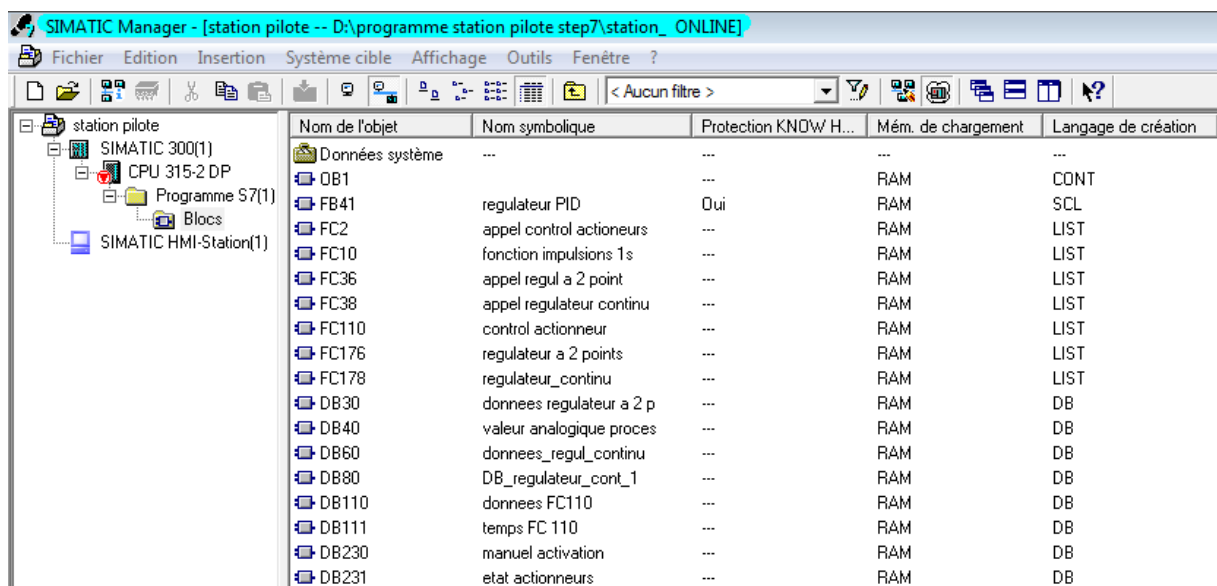


Figure IV.11 : Chargement du programme dans la CPU

IV.4.1.3. Le lancement de la simulation

Pour lancer la simulation, il faut d'abord appuyer sur RUN-P du commutateur de mode de fonctionnement de la CPU qui se trouve dans le S7-PLCSIM.

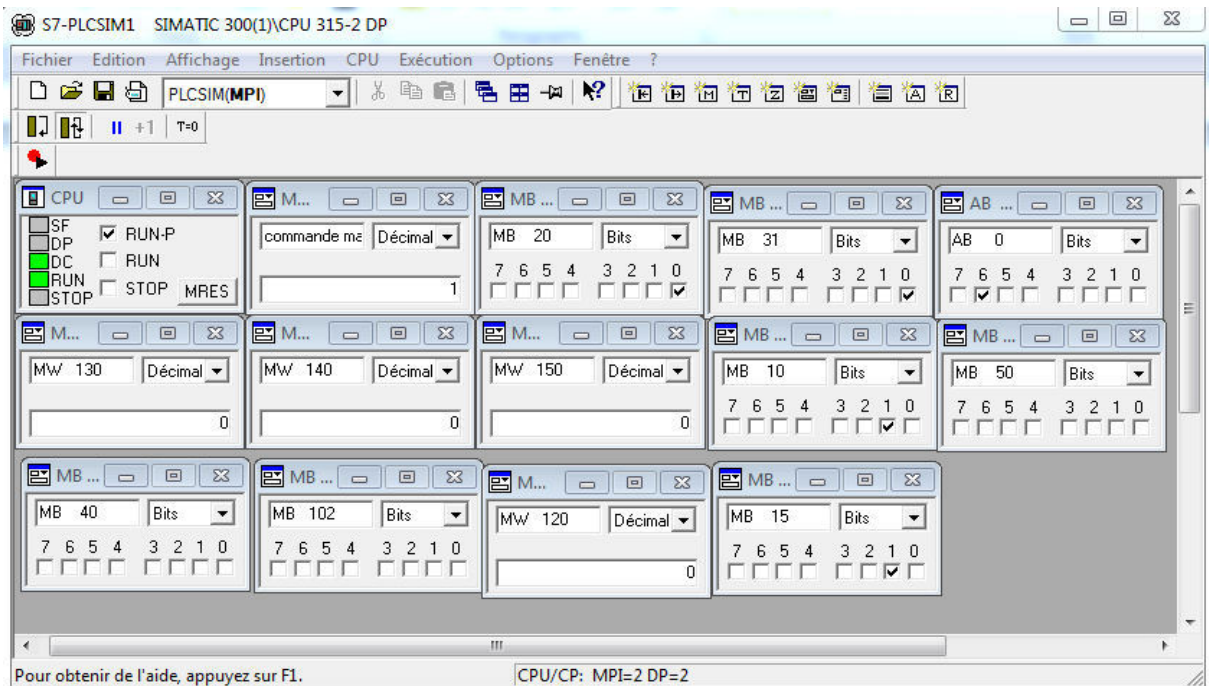



Figure IV.12 : Lancement de la simulation.

IV.4.1.4. Visualisation du programme de la pompe de bac de chauffe PW01

Après le chargement du programme et l'activation de la simulation, on visualise l'état de la pompe de bac de chauffe PW01, grâce à l'icone  qui se trouve dans la barre d'outils de la fonction FC2 « appel contrôle actionneurs ». Dans Figure IV.13, on voit que la pompe est en car (A0.6=1). Dans la barre des détails, on la voit en état MARCHE car (DB231.DBW 4=257).

Remarque :

Les 27 actionneurs de la chaîne pilote ont 4 états qui sont enregistrés dans le DB231.

Voici la signification des chiffres :

- 256= ARRET ;
- 257= MARCHE ;
- 258= ALARME ;
- 259= VERROUILLAGE.

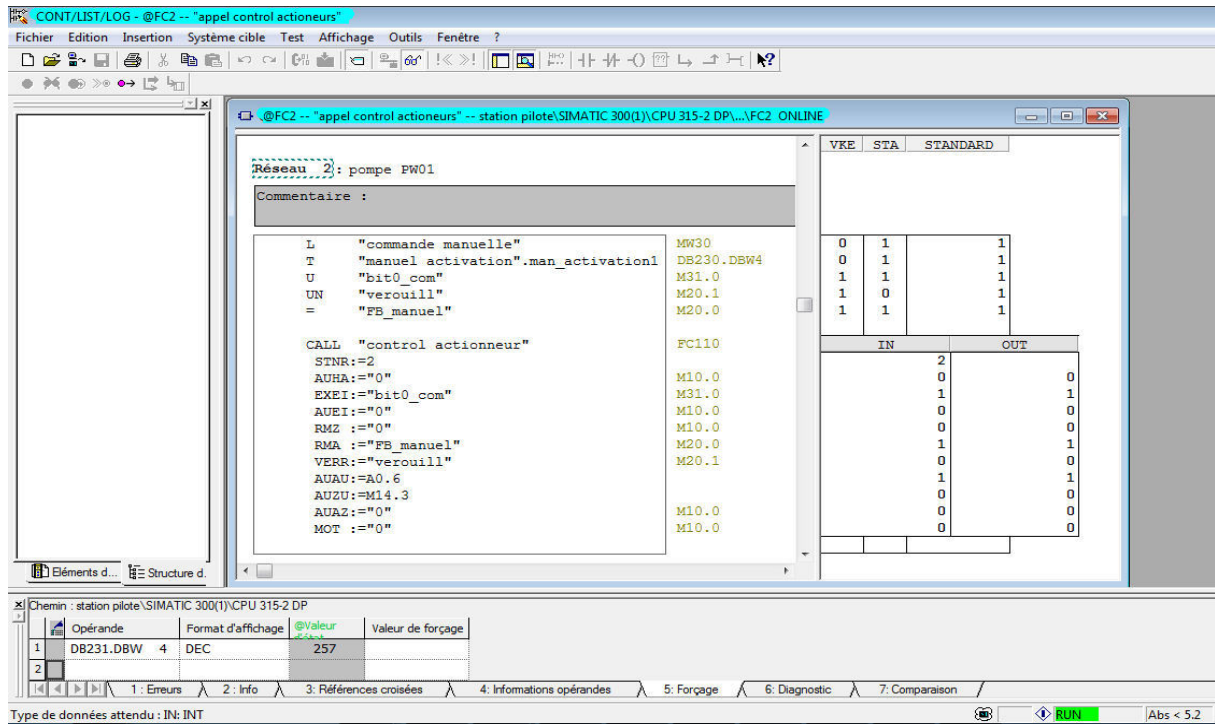



Figure IV.13 : Visualisation du programme de la pompe PW01.

IV.4.1.5. Visualisation du programme du régulateur de pasteurisateur

Dans la fonction FC38 « appel régulateur continu », on trouve l'icone  qui nous permet de visualiser l'activation et l'état (ouverture en %) de la vanne régulatrice de pasteurisateur.

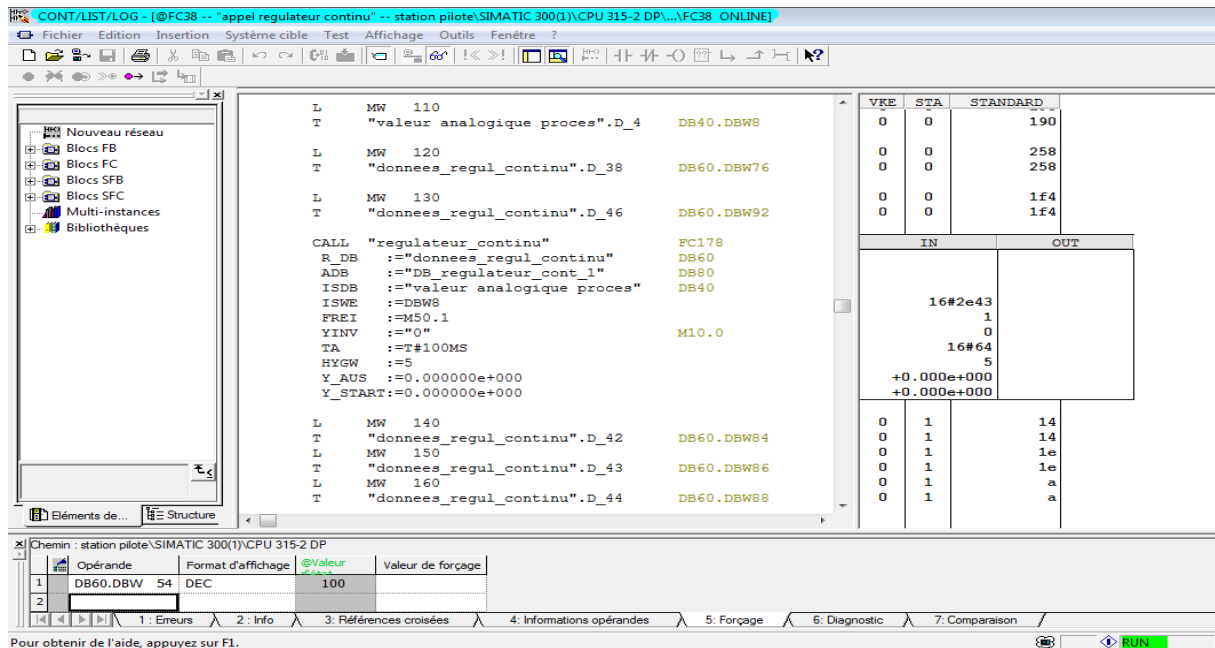



Figure IV.14 : Visualisation du programme de régulateur de pasteurisateur.

Dans la figure, la vanne est en mode automatique. Donc, on lui donne juste la consigne et elle s'ouvre en % automatiquement pour laisser passer de la vapeur dans le pasteurisateur. Sur la figure, la vanne est ouverte à 100%, car la température consigne est plus grande que la température réelle. Une fois arrivée à la température désirée, la vanne régulatrice se referme.

IV.4.2. Simulation avec WinCC flexible

Une fois que la liaison entre le pupitre et l'automate soit activée et le lancement de la simulation sous STEP7, on peut visualiser le processus de la station.

IV.4.2.1. Activer la simulation

Pour activer la simulation, on clic sur l'icone « RUNTIME »  qui se trouve dans la barre d'outils de WinCC flexible. On aura la fenêtre de menu principal dans le pupitre (Figure IV.17).

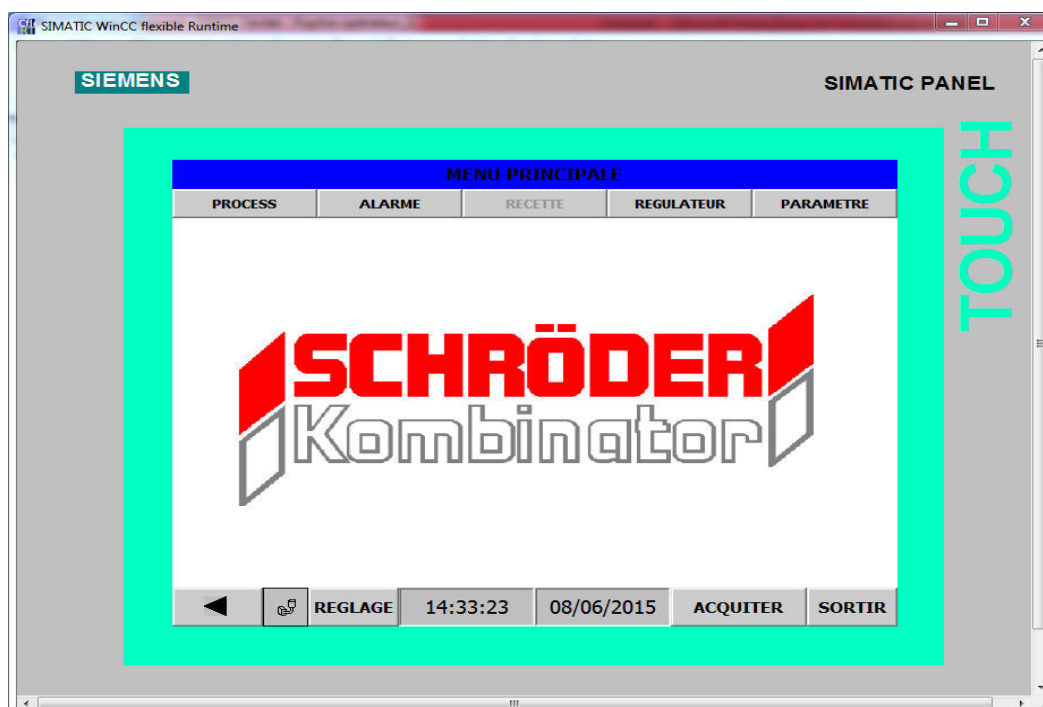


Figure IV.15 : Fenêtre de menu principale sur RUNTIME.

IV.4.2.2. Visualisation du processus dans la partie combineur/cristaliseur

Dans le menu « process », on trouve le bouton « combineur/cristaliseur ». On clic dessus et on actionne les moteurs en donnant les consigne de la vitesse de rotation et de la température souhaitées à la sortie des combineurs.

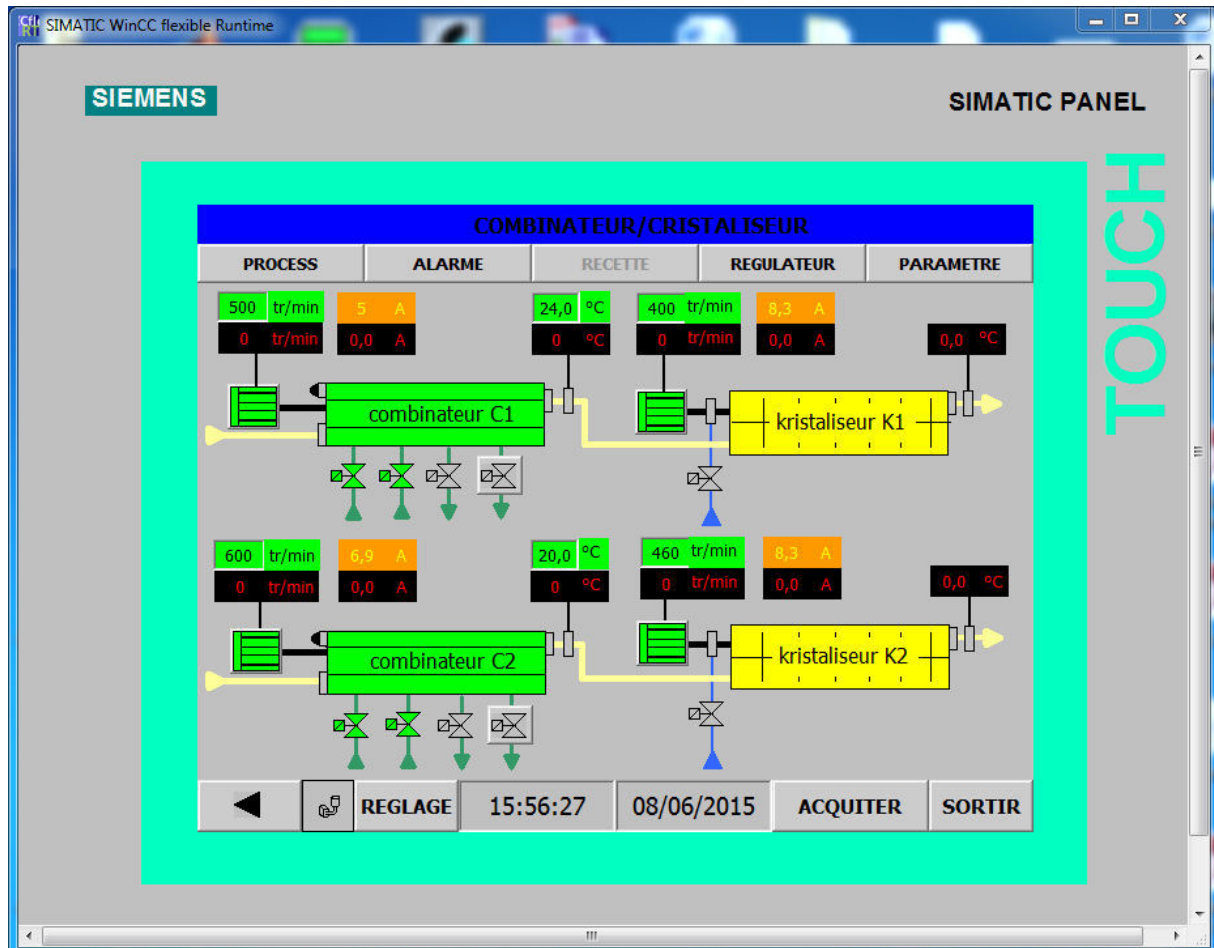


Figure IV.16 : Visualisation des actionneurs combineur/cristaliseur.

IV.4.2.3. Visualisation de régulateur de pasteurisateur

Dans le menu principal, en appuyant sur « REGULATEUR », on choisit Pasteurisateur et on active le mode AUTO pour l'ouverture automatique de la vanne de pasteurisateur en lui donnant la consigne AUTO de la température.

Dans Figure IV.19, on voit les 3 courbes de régulateur de pasteurisateur :

- La courbe de la température réelle « PV-X » est présentée en jaune ;
- La courbe de la température consigne « SP-W » est présentée en blanc ;
- La courbe de pourcentage d'ouverture de la vanne « Y » est présentée en rose.

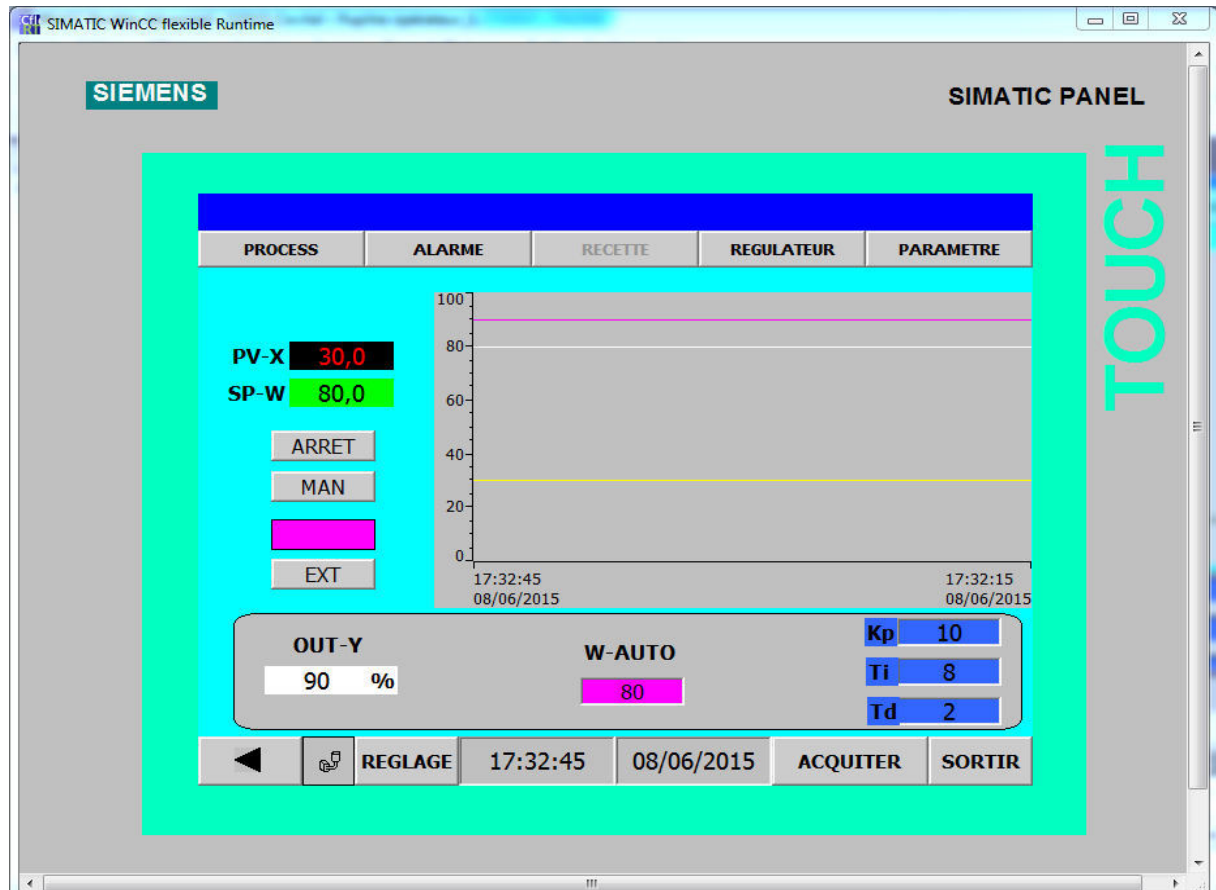


Figure IV.17 : Visualisation de régulateur de pasteurisateur.

IV.5. Conclusion

Ce chapitre est abordé par une présentation générale de la supervision et du logiciel WinCC flexible qui permet de commander et de superviser les processus en temps-réel. Dans une seconde partie, nous avons procédé à la simulation du projet de la station pilote. On a commencé par le STEP7 ou on a simulé le programme de la pompe PW01. Ensuite, on a simulé, avec le WinCC, la partie combineur/cristaliseur.

La simulation nous a permis de tester le programme et la supervision de notre projet, avant de passer à l'implantation du programme dans l'automate et le projet dans le panel.

Conclusion générale

Conclusion générale

Notre travail consistait à étudier l'insertion d'une panel de supervision dans une chaîne pilote de production de margarine. Pour cela, il nous fallait étudier le fonctionnement de la station pour pouvoir raccorder la partie commande qui remplacerait entièrement l'opérateur humain, à la partie opérative déjà fonctionnelle.

Notre connaissance du logiciel STEP7 nous a permis d'implémenter quelques tâches d'automatisation sur l'automate S7-300 ; qui a la particularité d'intégrer des modules d'entrées/ sorties de type tout ou rien (TOR) et analogique.

Après la programmation de la station, on a récupéré l'état des variables qui nous intéressaient pour créer l'interface homme-machine. Pour la conception du projet de supervision du système, nous avons exploité les performances de WinCC flexible qui est un logiciel permettant de gérer les interfaces graphique avec des visualisations et des animations actualisées.

En dernier lieu, on a étudié, en simulation, le programme de la station ainsi que son système de supervision, sous WinCC flexible, permettant à l'opérateur de connaître l'état d'avancement du processus en temps réel et d'intervenir directement sur le pupitre de commande implanté sur l'armoire électrique de la station.

Suite au travail effectué au niveau de l'unité de margarinerie de CEVITAL, on estime que les résultats obtenus sont très concluants. En effet, nous avons pu réaliser le projet qui nous a été confié et tester avec succès l'installation du pupitre de la chaîne pilote.

Ce projet était une occasion d'approfondir nos connaissances acquises durant notre formation et de les confronter en étude de simulation à un problème d'industrie réel. Cela nous a permis d'acquérir une expérience dans le domaine de la pratique.

Références bibliographiques

Références bibliographiques

- [1] Catalogue technique de Cevital: «armoires d'huile finie», 2007.
- [2] Catalogue technique de Cevital : « *CEVITAL installation pilote* », 2000.
- [3] J .Perrain, F .Binet, J.-J .Doumer, C .Merlaud, J. -P .Trichard. « *Automatique et informatique industrielle,*» Edition Nathan, 2006.
- [4] A. Simon, « *Automates programmables industriels Niveau 1,*» Edition l'Elan-Liège, 1991.
- [5] M. Bertrand, « *Technique de l'ingénieur, automates programmables industriels,*» Edition, 2001.
- [6] R. Azizi et R. Chemali, « *Commande et supervision de l'unité de conditionnement d'huile (Cevital),*» Mémoire de Master 2, université de Béjaia, promotion 2010.
- [7] Document Siemens, « *Information et formation, automatisation et entraînements, programmation niveau A,* » Edition Siemens AG, 2003.
- [8] Document Siemens, « *Automate programmable S7-300, caractéristiques électriques et techniques des CPU SIMATIC,*» Edition Siemens, 2001.
- [9] Document de formation pour une solution complète d'automatisation T I A MODULE B5 A&D SCE « *Automatisierung- und Antriebstechnik, Siemens A&D Cooperates with Education,* » Edition, 2005.
- [10] Document de formation pour une solution complète d'automatisation T I A. MODULE A4 A&D SCE « *Automatisierung- und Antriebstechnik, Siemens A&D Cooperates with Education,*» Edition, 2004.
- [11] J.-P. Thomesse, « *Technique de l'Ingénieur, Ingénierie des systèmes homme-machine,* » Edition, 2004.

- [12] L. ANDJOUH et R. TOUATI « Automatisation et supervision de la fosse de relevage de la raffinerie d'huile (Cevital) » Mémoire Master 2, Université de Boumerdes, promotion 2013.
- [13] Logiciel STEP7 « Manuel, » Edition, 2004.

Annexes

Annexes 1 : Programme

Annexes 1 : Supervision

Bloc d'organisation OB1

Appel de toutes les fonctions utilisé dans le programme.

OB1 : "Main Program Sweep (Cycle)"

Commentaire :

Réseau 1: Titre :

Commentaire :

FC10
"fonction
impulsions
1s"
— (CALL) —

Réseau 2: Titre :

Commentaire :

FC2
"appel
control
actioneurs
"
— (CALL) —

Réseau 3: Titre :

Commentaire :

FC36
"appel
regul a 2
point"
— (CALL) —

Réseau 4: Titre :

Commentaire :

FC38
"appel
regulateur
continu"
— (CALL) —

Réseau 5: toujours1

Commentaire :

M10.1 "1" — (S) — M10.1 "1"

Réseau 6: toujours0

Commentaire :

M10.0 "0" — (R) — M10.0 "0"

Programme FC2

La fonction FC2 a été créée pour appeler la fonction FC110 « contrôle actionneur » qui gère le fonctionnement de chaque actionneurs.

FC2 : Titre :

Contrôle actionneurs

Réseau 1: compresseur P-V01

actionneur01

Commentaire :

```

U   "commande_auto"           M15.2
S   "ordre_demmar_auto"       M40.0
U   "ordre_demmar_auto"       M40.0
UN  "verouill"                M20.1
=   "FB compresseur"         M21.2
CALL "control actionneur"     FC110      -- CONTROL ACTUATOR
  STNR:=1
  AUHA:="1"                   M10.1
  EXEI:="commande externe"    M14.1
  AUEI:="commande_auto"       M15.2
  RMZ :="0"                    M10.0
  RMA :="FB compresseur"      M21.2
  VERR:="verouill"            M20.1
  AUAU:="compresseur marche"  A0.2
  AUZU:=M14.3
  AUAZ:="0"                    M10.0
  MOT :="0"                    M10.0

```

Figure A.1.1 : programme de compresseur P-V01.

Réseau 3: Résistance bac1 WWH P-W01.E1

Actionneur 03

Commentaire :

```

U   "commande_auto_res"       M15.3
S   "ordre demmar_auto"       M40.3
U   "ordre demmar_auto"       M40.3
UN  "verouill"                M20.1
=   "FB_résistance_bac1"     M21.3
CALL "control actionneur"     FC110      -- CONTROL ACTUATOR
  STNR:=3
  AUHA:="1"                   M10.1
  EXEI:="commande externe_résist" M14.2
  AUEI:="commande_auto_res"    M15.3
  RMZ :="0"                    M10.0
  RMA :="FB_résistance_bac1"  M21.3
  VERR:="verouill"            M20.1
  AUAU:="résistancel_marche"  A0.3
  AUZU:=M14.3
  AUAZ:="0"                    M10.0
  MOT :="0"                    M10.0

```

Figure A.1.2 : programme de la résistance WWH P-W01.E1.

Réseau 6 : Vanne remplissage C-1 PK1-M101

Commentaire :

```
U   "commande_auto_PK1-M101"      M15.5
S   "ordre_demmar_auto"           M40.0
U   "ordre_demmar_auto"           M40.0
UN  "verouill"                    M20.1
=   "FB_vanne_PK1-M101"           M21.6

CALL "control actionneur"          FC110      -- CONTROL ACTUATOR
STNR:=9
AUHA:="1"                          M10.1
EXEI:="commande_externerésist"     M14.2
AUEI:="commande_auto_PK1-M101"     M15.5
RMZ := "0"                          M10.0
RMA := "FB_vanne_PK1-M101"         M21.6
VERR:="verouill"                   M20.1
AUAU:="vanne_marche"               A0.7
AUZU:=M14.3
AUAZ:="0"                          M10.0
MOT := "0"                          M10.0
```

Figure A.1.3 : Programme de la vanne PK-M101.

Programme FC36

La fonction FC36 gère le fonctionnement des régulateurs à 2 point on utilisant la fonction FC176 « contrôle régulateur à 2 point ».

Réseau 1: regulateur de bac de chauffe n 1

Commentaire :

```

U   "visualis_regu"           M60.7
=   DB30.DBX  90.7
U   "visualis_regu"           M60.7
U   "regul MAN"               M60.1
U   M  102.0
S   DB30.DBX  166.7
U   "visualis_regu"           M60.7
U   "regul MAN"               M60.1
U   M  102.1
R   DB30.DBX  166.7
U   "regula arret"            M60.0
=   DB30.DBX  91.0
U   "regul MAN"               M60.1
=   DB30.DBX  91.1
U   "regul AUTO"              M60.2
=   DB30.DBX  91.2
U   "regul extern"            M60.3
=   DB30.DBX  91.3

L   "T° reelle"                MW80
T   "valeur analogique proces".D_1  DB40.DBW2

CALL "regulateur a 2 points"      FC176      -- 2-POINT CONTROLLER
DB_travail      := "donnees regulateur a 2 p"  DB30
DB_valeur_actuelle := "valeur analogique proces"  DB40
temperature     := DBW2
activation      := "activ"                    M50.0
hysterisis     := 5
AUSG           := A5.0

```

Figure A.1.4 : programme du régulateur a 2 point de bac de chauffe 1.

Visualisation sur le pupitre de la station pilote

Ces figures présentent les différentes parties du processus de supervision pour la station pilote.

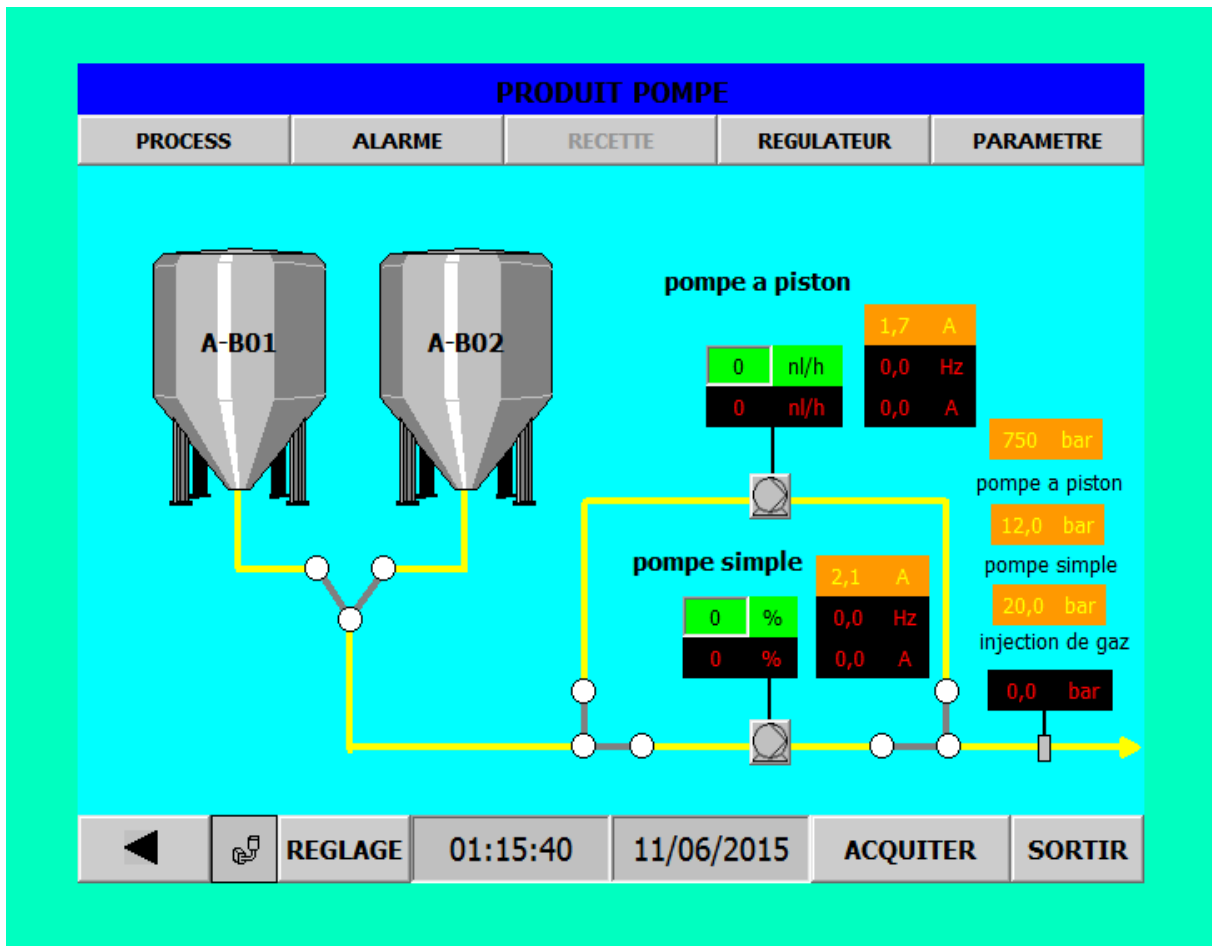


Figure A.2.1 : la partie produit pompe.

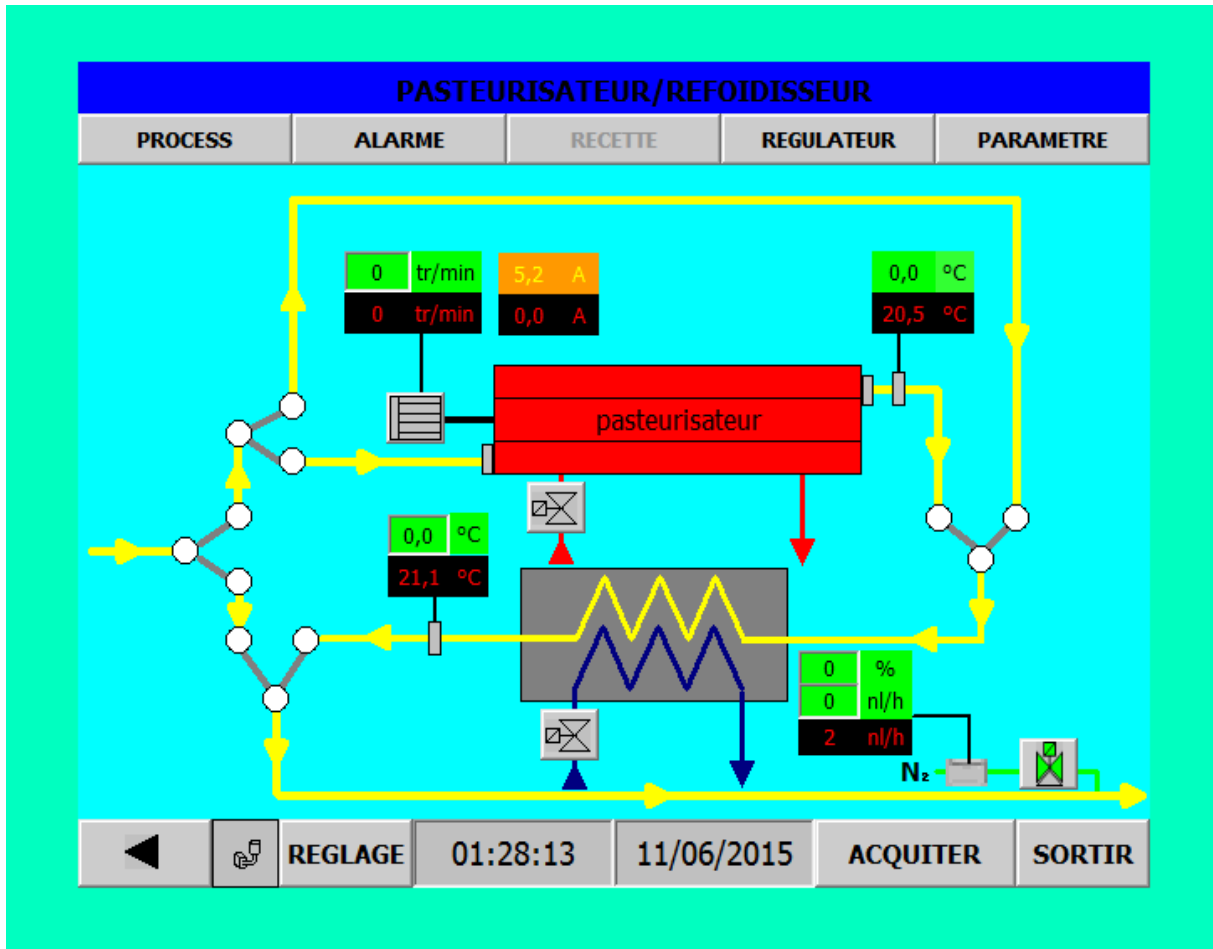


Figure A.2.2 : La partie pasteurisateur/refroidisseur.

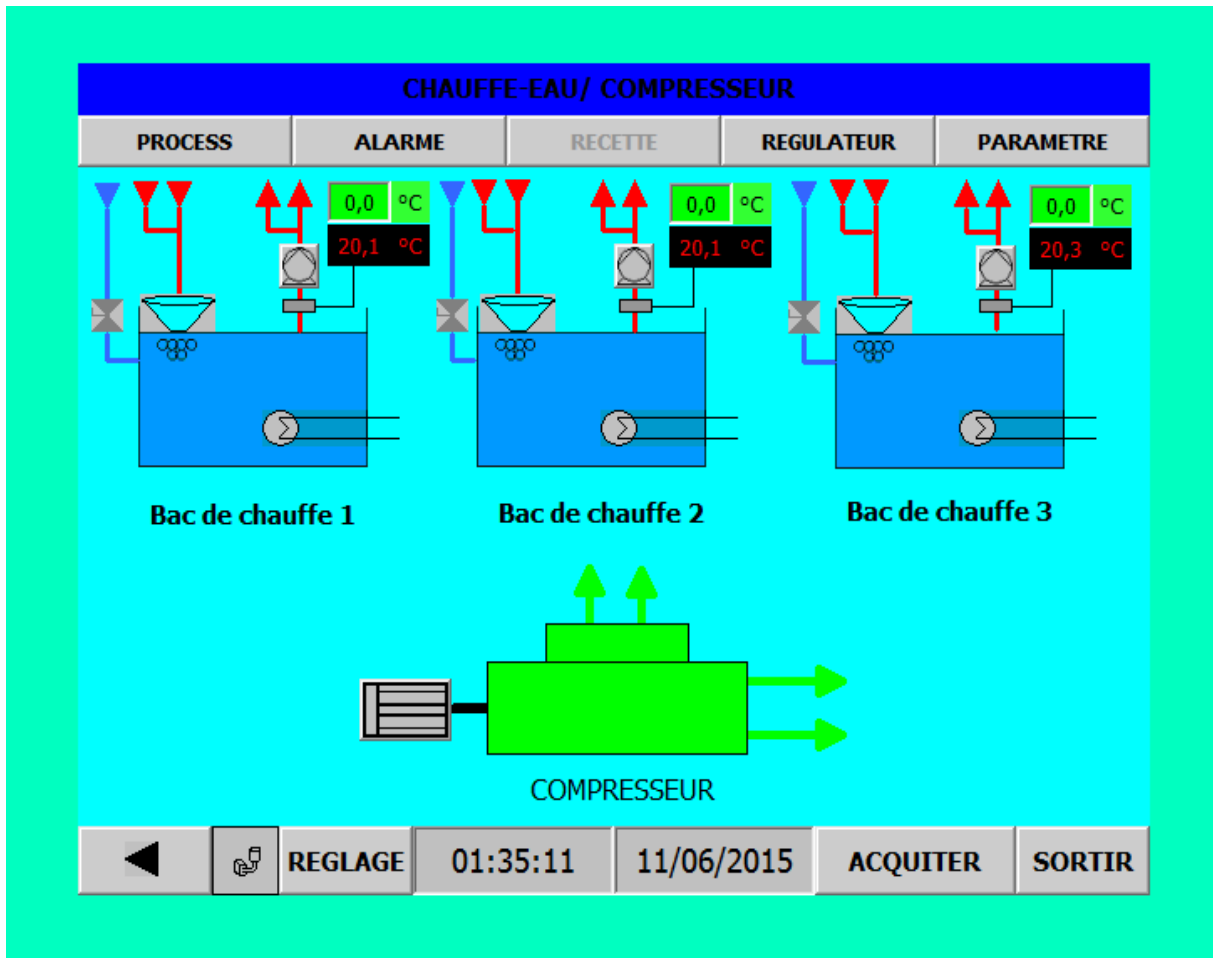


Figure A.2.3 : La partie chauffe-eau/compresseur.

Résumé

Ce mémoire présente une méthodologie générale sur l'automatisation et la supervision d'un système industriel. Il a été question d'une étude détaillée sur une chaîne pilote pour la fabrication de la margarine, le programme élaboré avec le logiciel STEP7 permet, une fois transféré dans l'automate S7-300 de gérer le fonctionnement de la station.

Une interface homme-machine à base du logiciel WinCC flexible a été créée, celle-ci permet entre autres, la récupération et l'affichage en ligne des variables caractéristique du processus.

Vous trouverez également une description détaillée sur les automates programmables industriels et plus précisément le S7-300 de la firme SIEMENS, ainsi sur le logiciel STEP7 en mettant en avant les étapes pour la création d'un projet d'automatisation.

Une simulation du projet a été faite sous le logiciel STEP7 et le logiciel WinCC.

Abstract

The present dissertation introduces a general methodology for automation and monitoring of an industrial system. It was about a detailed study on a pilot chain aiming at margarine production. The elaborated STEP7 program allows, once transferred on the S7-300 automation, the station operation management.

A homme-machine interface, based on WinCC software, has been developed, especially, this proceeds a visual display of the interacting variables of the process.

You will also find a detailed description on industrial programmable automations more precisely the S7-300 of SIEMENS and on STEP7 software highlighting the steps to automation project creation.

A project simulation has been done under the STEP7 and WinCC software.