

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DE TECHNOLOGIE  
DEPARTEMENT D'ELECTRONIQUE

N° :ESEM05/2019



DOMAINE : SCIENCES TECHNOLOGIE  
FILIERE : ELECTRONIQUE  
OPTION : ELECTRONIQUE DES  
SYSTÈMES EMBARQUÉS

Mémoire présenté pour l'obtention  
Du diplôme de Master Académique

Par: Mr. LAMINE Elbarka

Mr.HAMMOUALI abdelhamid

Intitulé

**Conception d'une Régulation de Niveau  
avec un Automate Programmable**

Soutenu devant le jury composé de:

Dr.BENAHCENE Madani	Université	M'SILA	Président
Dr.LALAOUI Lahouaoui	Université	M'SILA	Rapporteur
Dr.KHALFA Ali	Université	M'SILA	Examineur

Année universitaire : 2018 /2019

# Dédicace

Je dédie ce modeste travail : A mes très chers parents, pour leur soutien et encouragement le long de mes années d'études, aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.

A mes chères sœurs que dieu les garde pour moi.

A mes chers amis : aucun mot ne saurait exprimer tout l'amour que j'ai pour eux.

A toutes mes chères amis chaqu'un par son nom et particulièrement.

A tous mes enseignants et mes collègues.

Et à tous ceux que ma réussite leur tient à cœur

Qu'ils trouvent ici l'expression de toute ma reconnaissance

**Abdelhamid**

# Dédicace

Je dédie ce travail:

À la lumière de mon voyage, à mes parents qui m'ont donné

La vie, symbole de la tendresse et de mon école d'enfance, qui

On m'a promis toute ma vie de m'encourager à

Assistance et protection psychologique toutes les années  
Études. Dieu les protège et les protège.

À mes frères et sœurs.

Et toute ma famille.

Aux stars de ma vie: mes amis.

Pour chacun des Aziz Ali.

Pour tous ceux qui m'aiment.

Pour tout ce que j'aime.

Pour toutes la promotion électronique 2019.

Pour tous ceux qui ont contribué de près ou de loin à

Réaliser ce travail modeste.

Elbarka

# REMERCIEMENTS

Nous voulons remercier Dieu d'abord.

Nous donner la force et la patience nécessaires pour accomplir cette tâche modeste.

Deuxièmement, nous voudrions remercier nos chers parents qui ont toujours été avec nous et qui ont fourni un excellent exemple de travail acharné et de persévérance.

Nous remercions également notre exécutif

**Mr. LALAOUI LAHOUAOUI** qui a eu la gentillesse de nous superviser pour développer et observer notre projet de remise des diplômes.

Nous souhaitons également remercier les membres du jury qui nous ont honorés pour avoir examiné ce travail et présenté leurs idées et leurs propositions scientifiques.

Nous saisissons cette occasion pour remercier tous les professeurs de l'Université de M'Sila pour tout ce qu'ils nous ont fourni, proches et lointains, avec leur aide généreuse pour développer ce travail.

**Elbarka et Abdelhamid**

# Sommaire

---

Liste des figures.....	
Liste des Tableaux.....	
Liste des abréviations.....	
Introduction générale.....	1

## **Chapitre I : Les automates Programmables Industriels**

I.1. Introduction .....	4
I.2. Systèmes automatisés .....	4
I .2.1. Objectif de l'automatisation .....	4
I.2.2. Les avantages et les inconvénients de l'automatisation .....	5
I.2.2.1. Les avantage .....	5
I.2.2.2. Les inconvénients .....	5
I.2.3. Domaines d'application des systèmes automatisés .....	5
I .2.4. Structure d'un automatisme .....	5
I .2.5. Mise en œuvre d'un automate .....	7
I .2.6. Caractéristiques d'un automatisme .....	7
I .3. Définition d'un API .....	8
I .3.1. Principe générale de fonctionnement d'un API.....	8
I .3.2. Architecture d'un API .....	9
I.3.2.1. Aspect extérieur .....	9
I.3.2.1.1. Compacte .....	9
I.3.2.1.2. Modulaire .....	9
I.3.2.2. Structure interne .....	9
I.3.2.2.1. Un module d'alimentation .....	9
I.3.2.2.2. Une unité de traitement ou processeur .....	10
I.3.2.2.3. Une mémoire programme .....	10
I.3.2.2.4. Une mémoire de données .....	10
I.3.2.2.5. Une interface d'entrée .....	10
I.3.2.2.6. Une interface de sortie .....	11

## Sommaire

---

I.3.3. Critère de choix de l'automate programmable industriel .....	11
I.4. Présentation d'un automate .....	12
I.5. Présentation de quelques gammes SIMATIC .....	13
I.5.1. SIMATIC S7 .....	13
I.5.1.1. SIMATIC S7-200 .....	13
I.5.1.2. SIMATIC S7-300 .....	14
I.5.1.3. SIMATIC S7-400 .....	14
I.5.2. Présentation de l'automate à utiliser S7-300 .....	15
I.5.3. Critère de choix d'un API .....	16
I.5.4. Description de l'Automate S7-300 .....	16
I.5.5. Constitution de l'automate S7-300.....	16
I.6. Présentation de logiciel de programmation STEP7 .....	18
I.7. Langages de programmation .....	18
I.7.1. Liste d'instructions (IL : Instruction List) .....	18
I.7.2. Langage littéral structuré (ST : Structured text) .....	18
I.7.3. Langage à contacts (LD : Ladder Diagram) .....	18
I.8. Présentation de Win CC Runtime Advanced .....	19
I.8.1. Introduction .....	19
I.8.2. Les tâches d'un système IHM.....	19
I.9. Conclusion .....	20

### **Chapitre II : Généralité sur les remplisseuses et régulation**

II.1. Introduction .....	22
II.2. Les remplisseuses .....	22
II.2.1. Les remplisseuses rotatives.....	22
II.2.2. Les remplisseuses linéaires.....	23
II.3. Systèmes de remplissage.....	23
II.3.1. Remplisseuse à niveau .....	24
II.3.2. Remplisseuse à débitmètre .....	24
II.3.3. Remplisseuse pondérale .....	25
II.3.4. Remplisseuse volumétrique .....	26
II.4. Élément pour fonction de détection .....	26
II.4.1. Capteurs .....	26
II.4.1.1. Capteurs passifs .....	26

## Sommaire

---

II.4.1.2. Capteurs actifs .....	27
II.4.2.Principales caractéristiques des capteurs .....	27
II .4.3.Familles des capteurs .....	27
II.4.3.1.Les capteurs TOR .....	27
II.4.3.2.Capteurs analogique.....	27
II.4.3.3.Capteur numérique .....	28
II.5. Mesure et détection de niveau .....	29
II.5.1. Le capteur de niveau .....	29
II.5.2. Principe de fonctionnement .....	30
II-6.La régulation automatique .....	30
II .6.1.Définition de la régulation automatique .....	30
II.6.2.Objectif de la régulation automatique.....	31
II .6.3. Notion de Boucle Ouverte/Fermée .....	31
II .6 .3.1. Système boucle ouverte .....	31
II.6 .3.2.Système boucle fermée .....	32
II .7.Définition de la régulation PID .....	32
II .7.1.Notion de correcteur PID .....	32
II .7.2.Structure des régulateurs électroniques .....	33
II .7.2.1.PID parallèle .....	33
II .7.2.2.PID série .....	34
II .7.2.3.PID mixte .....	34
II.7.3.Utilisation .....	34
II .8.Conclusion .....	34
<b>Chapitre III : Description Logiciel de programmation et simulation et supervision</b>	
III.1. Introduction .....	36
III.2. Logiciel de programmation « TIA Portal V13 » .....	36
III.2.1 .Présentation du logiciel .....	36
III.2.2 .La conception d'un programme avec TIA PORTAL V13 .....	36
III.2.3. Vue du portail .....	37
III.2.4.Vue du projet .....	38
III.2.5.Création du projet dans <i>TIA PORTAL V13</i> .....	38
III.2.6. Configuration matérielle (Partie Hardware) .....	39

## Sommaire

---

III.2.7. Adresse Ethernet de la CPU .....	40
III.2.8. Création du tableau des variables API .....	40
III.2.9. Types de variable utilisées en STEP7 .....	41
III.2.10. Ecriture du programme .....	42
III.3. Ajouter un GRAFCET .....	43
III.4. Structure graphique du Grafcet .....	43
III.4.1. Etape .....	43
III.4.2. Transition .....	44
III.4.3. Branche OU .....	44
III.4.4. Saut .....	44
III.5. Simulation du programme sous step7 avec S7-PLCSIM .....	45
III.5.1 Présentation de S7-PLCSIM .....	45
III.5.2 Caractéristiques de S7-PLCSIM .....	45
III.6. SIMATIC Win CC Runtime Advanced (Portail TIA) .....	46
III. 6.1. Définition de l'interface homme-machine (IHM) .....	46
III.6.2. Avantages .....	46
III.6.3. Configurer une vue IHM .....	47
III. 6.4. Eléments de la vue racine .....	48
III.7. Simulation de la station .....	49
III.7.1. Les Blocs utilisateurs .....	49
III.7.1.1. Bloc d'organisation (OB) .....	49
III.7.1.2. FB (Bloc de fonction).....	51
III.7.1.3. Blocs de données (DB) .....	52
III.8. Réalisation de la supervision .....	52
III.8.1. Etablir une liaison directe .....	52
III.8.2. Création de la table des variables.....	53
III.8.3. Création de vue .....	54
III.8.4. Vue générale .....	54
III.8.5. Compilation et Simulation .....	55
III.8.6. Simulation du programme sous WinCC .....	57
III.9. Conclusion .....	58
<b>Conclusion Générale</b> .....	<b>60</b>
<b>Bibliographie.</b>	
<b>Annexes.</b>	

## Liste des Figures

---

### Liste des Figures

#### Chapitre I

<b>Figure I.1</b> : Représentation d'un automate programmable industriel.....	4
<b>Figure I .2</b> : Structure d'un système automatisé.....	6
<b>Figure I .3</b> : Dialogue entre la partie commande et la partie opérative.....	6
<b>Figure I.4:</b> Fonctionnement d'un API.....	9
<b>Figure I.5</b> : Structure interne des automates.....	11
<b>Figure I.6</b> : Automate Programmable Industriel SIEMENS .....	12
<b>Figure I.7</b> : API SIMENS S7-200.....	13
<b>Figure I.8</b> : API SIMENS S7-300.....	14
<b>Figure I.9</b> : API SIMENS S7-400.....	14
<b>Figure I.10</b> : Constitution d'API S7-300 .....	15
<b>Figure I.11</b> : Constituants de l'automate S7-300.....	17

#### Chapitre II

<b>Figure II.1:</b> Vue générale d'une remplisseuse rotative.....	22
<b>Figure II.2:</b> Technique de soutirage.....	22
<b>Figure II.3:</b> Remplisseuse linéaire.....	23
<b>Figure II.4:</b> Remplisseuse .....	24
<b>Figure II.5:</b> Capteur.....	26
<b>Figure II.6:</b> Capteur tout ou rien.....	27
<b>Figure II.7:</b> Capteur analogique.....	28
<b>Figure II.8:</b> Capteur numérique.....	28
<b>Figure II.9:</b> Capteur de niveau.....	30
<b>Figure II.10</b> : Présentation d'un schéma fonctionnel d'un régulateur.....	31
<b>Figure II.11</b> : Système en BO.....	31
<b>Figure II.12</b> : Système en BF.....	32
<b>Figure II .13:</b> Schéma d'un régulateur PID parallèle.....	33
<b>Figure II .14:</b> schéma d'un régulateur PID série.....	34
<b>Figure II.15:</b> schéma d'un régulateur PID mixte.....	34

#### Chapitre III

<b>Figure III.1</b> : Organisation pour la création d'un projet sous TIA PORTAL .....	37
<b>Figure III.2</b> : Vue détaillée du portail.....	37

## Liste des Figures

---

<b>Figure III.3</b> : Vue détaillée du projet.....	38
<b>Figure III.4</b> : Mise en créer du projet.....	39
<b>Figure III.5</b> : Représentation de notre automate S7-300.....	40
<b>Figure III.6</b> : Adresse Ethernet de la CPU.....	40
<b>Figure III.7</b> : Tableau de variable API.....	41
<b>Figure III.8</b> : Création des blocs et des fonctions.....	42
<b>Figure III.9</b> : Représentation d'un Grafcet .....	43
<b>Figure III.10</b> : Etape.....	43
<b>Figure III.11</b> : Transition.....	44
<b>Figure III.12</b> : Branche OU .....	44
<b>Figure III.13</b> : Saut.....	45
<b>Figure III.14</b> : Configurer une vue IHM.....	47
<b>Figure III.15</b> : Choisir le pupitre.....	47
<b>Figure III.16</b> : Des éléments de la vue racine.....	48
<b>Figure III.17</b> : Fenêtre d'ajout de bloc.....	49
<b>Figure III.18</b> :Réseau 1 dans l'OB 1.....	50
<b>Figure III.19</b> : Réseau 2 dans l'OB 1.....	50
<b>Figure III.20</b> : Réseau 3 dans l'OB 1.....	51
<b>Figure III.21</b> : Bloc fonction [FB1].....	51
<b>Figure III.22</b> : Bloc de donnée DB1.....	52
<b>Figure III.23</b> : Liaison entre la PLC et IHM.....	53
<b>Figure III.24</b> : Table de variable IHM.....	53
<b>Figure III.25</b> : Création de vue.....	54
<b>Figure III.26</b> : Vue générale.....	55
<b>Figure III.27</b> : Etape de compilation PLC.....	56
<b>Figure III.28</b> : Simulateur S7-PLCSIM.....	56
<b>Figure III.29</b> : La vue production.....	57
<b>Figure III.30</b> : Vue du système.....	57

## Liste des Tableaux

---

### Liste des Tableaux

<b>Table II.1</b> : Notion de correcteur PID.....	33
<b>Tableau III.1</b> : Types de variables utilisées en step7 .....	41
<b>Tableau III.2</b> : Tableau d'éléments de la vue racine.....	48

## Liste des abréviations

---

### Liste des abréviations

**A.P.I** : Automate Programmable industriel

**CPU** : Unité centrale de l'automate (Central processing unit).

**DB** Blocs de données globaux

**E/S** Entrée / Sortie.

**EEPROM** : Electric al Erasable Programmable Read Onlay Memory

**EPROM** : Erasable Programmable Read Onlay Memory

**FB** : Bloc de fonction

**FC** : Fonction

**HMI** Human Machine Interface

**IHM** : Interface Homme/Machine

**IP** : Internet Protocol

**LIST** : Le langage de liste d'instructions

**LOG** : Le langage a base de logigramme

**MPI** Protocole de communication (Multi Point Interface).V

**OB** Bloc d'organisation.

**PN/IE** : Profinet/Industriel Ethernet

**IP**: Internet protocol.

**LD**:Ladder Diagram.

**IL**: Instruction List.

**ST**:Structuredtext.

**PC** :Partie Commande.

**PO** :Partie Opérative.

**PID** : Proportionnel-intégral-dérivée

**PLC** : Programmable Logic Controller

**PROM** : Erasable Programmable Read Onlay Memory

**RAM**: Random Access Memory

**ROM**: Read Only Memory

**S7**: Step7

**GRAFCET**: Graf de Commande Etapes-Transition.

**SIMATIC**: Siemens Automatic

**SF** : Modules fonctionnels

**SM** : Gamme des modules E/S des automates de Siemens

## Liste des abréviations

---

**SP** : Sept Point

**TOR** : Tout ou Rien

**TIA Portal**: Totally Integrated Automation Portal

**V13** : Version 13

**WinCC** : Windows Control Center

# **Introduction Générale**

# Introduction générale

---

## Introduction générale

Le contrôleur logique programmable est l'appareil principal de la boucle de contrôle du processus industriel que nous aspirons à concrétiser sur le terrain afin de le contrôler. Sa tâche principale est de collecter des informations sur l'état du système, à partir de divers capteurs à travers les interfaces d'entrée, et de les traiter pour décision, conformément à la logique de fonctionnement mise en surbrillance par le programme stocké en mémoire.

L'automatisation de cette unité est essentielle car elle contribue à accroître la productivité, la flexibilité, la qualité et les conditions de travail. Les systèmes automatisés sont rapides et précis, car ils exécutent des tâches difficiles, voire presque impossibles pour l'homme. Automatisation signifie qualité de production, sécurité et précision.

Nous avons travaillé sur l'automatisation et la supervision de l'usine de production afin de remplir et de réguler le niveau de bouteilles de jus de fruits à l'aide du programme TIA Portal V13. L'étude sera basée sur une précision absolue dans l'analyse de tous les composants de notre usine, ainsi que des besoins industriels.

Un automate de type S7-300 a été utilisé et cette automatisation a été réalisée grâce au programme TIA PORTAL V13 de SIEMENS Pour refléter notre projet; qui est considéré comme l'un des meilleurs programmes d'ingénierie développés par cette société.

L'objectif de notre projet est d'améliorer le travail des services de la station de remplissage en régulant le niveau, en améliorant et en automatisant, en mettant en place le système de supervision et en assurant la continuité de ce travail.

Dans cette note, nous présentons trois grands chapitres décrivant les éléments clés de notre projet de remise des diplômes:

Le premier chapitre est consacré aux généralités sur l'automatisation. Nous avons mentionné les objectifs et la structure du système automatisé, ainsi que le principe de fonctionnement et les différentes unités constituant l'API.

Le deuxième chapitre est consacré à la définition de la régulation automatique et à la définition du remplissage des bouteilles et des types.

# **Introduction générale**

---

Le troisième et dernier chapitre est consacré à l'étude des différents composants de notre installation et au TIA PORTAL V13 avec ses différents composants, afin de préparer le projet, de choisir la CPU et les unités appropriées, ainsi que de différents langages de programmation, spécifications et adoption de solutions nous permettant de contrôler d'interface homme-machine (HMI) et de contrôler notre système. Nous clôturerons notre travail par une conclusion générale.

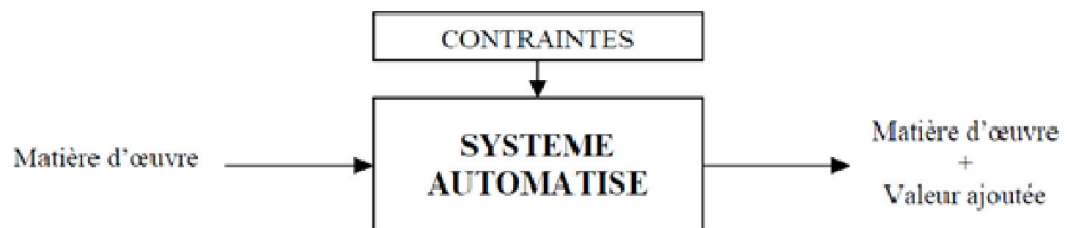
# **Chapitre I : Les automates Programmables Industriels**

## I.1. Introduction :

Les processus d'automatisation créent des opérations qui requièrent une intervention humaine pour automatiser les opérations. L'API vise à remplacer ce travail par des tâches simples, générales et pouvant être répétées nécessitant de la précision. Nous sommes passés d'un système manuel à un système automatisé utilisant des techniques électromécaniques, électroniques, pneumatiques et hydrauliques. L'automatisation est disponible dans tous les domaines (fabrication, alimentation, transport, automobile ...Etc.).

## I.2. Systèmes automatisés :

Un système automatisé est un ensemble d'éléments en interaction, et organisés dans un but précis : agir sur une matière d'œuvre an de lui donner une valeur ajoutée. Le système automatisé est soumis à des contraintes : énergétiques, de configuration, déréglage et d'exploitation qui interviennent dans tous les modes de marche et d'arrêt du système [1].



**Figure I.1 :** Représentation d'un automate programmable industriel.

### I .2.1. Objectif de l'automatisation :

L'objectif de l'automatisation des systèmes est de produire, en ayant recours le moins possible l'homme, des produits de qualité et ce pour un coût le plus faible possible, augmentation de la sécurité, s'adapter à des contextes particulier.

Elle permet de :

- Accroître la productivité de l'entreprise en réduisant les coûts de production.
- Améliorer la flexibilité de la production.
- Perfectionner la qualité du produit.
- S'adapter à des contextes particuliers.
- Augmenter la sécurité.

## **I.2.2. Les avantages et les inconvénients de l'automatisation :**

### **I.2.2.1. Les avantage :**

- La capacité de production accélérée.
- L'aptitude à convenir à tous les milieux de production.
- La souplesse d'utilisation.
- La création de postes d'automaticiens.

### **I.2.2.2. Les inconvénients :**

- Le coût élevé du matériel, principalement avec les systèmes hydrauliques.
- La maintenance doit être structurée.
- La suppression d'emplois

## **I.2.3. Domaines d'application des systèmes automatisés :**

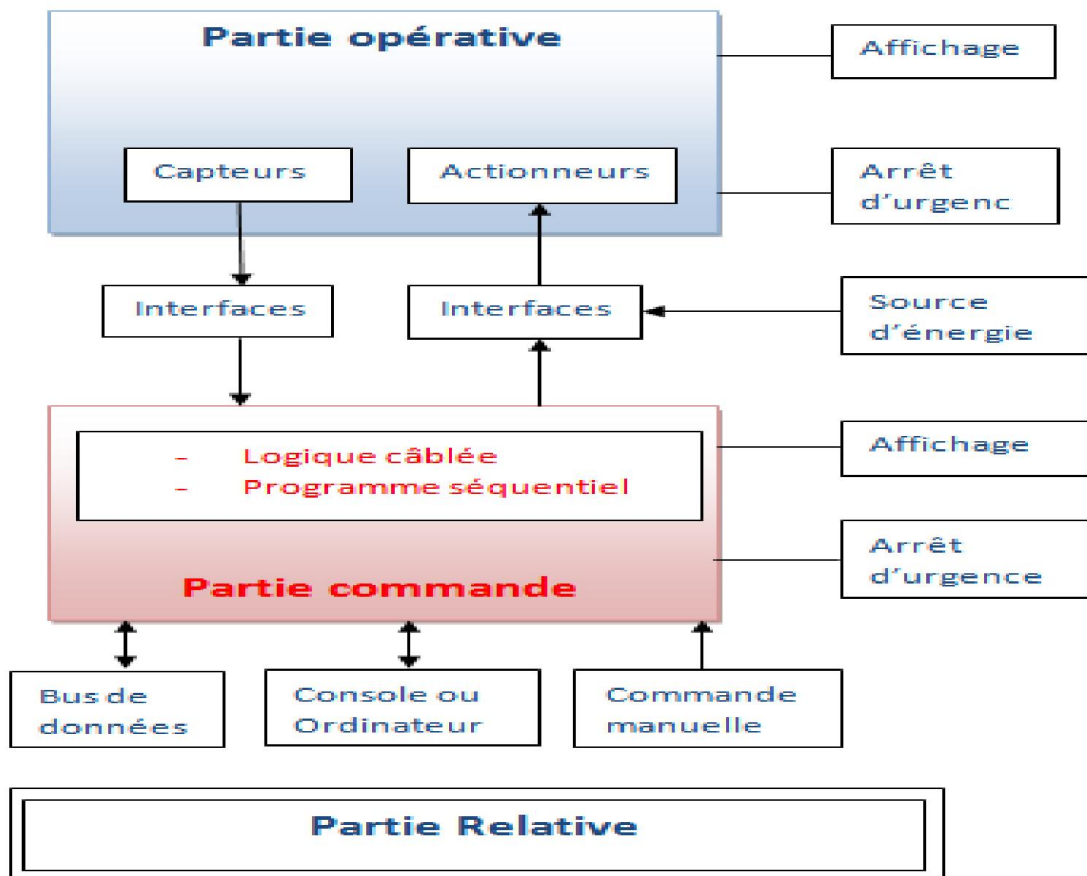
Aujourd'hui, il serait difficile de concevoir un système de production sans avoir recours aux différentes technologies et composants qui forment les systèmes automatisés.

- Automobile
- Aviation
- Industrie
- Médical
- Transport

## **I.2.4. Structure d'un automatisme :**

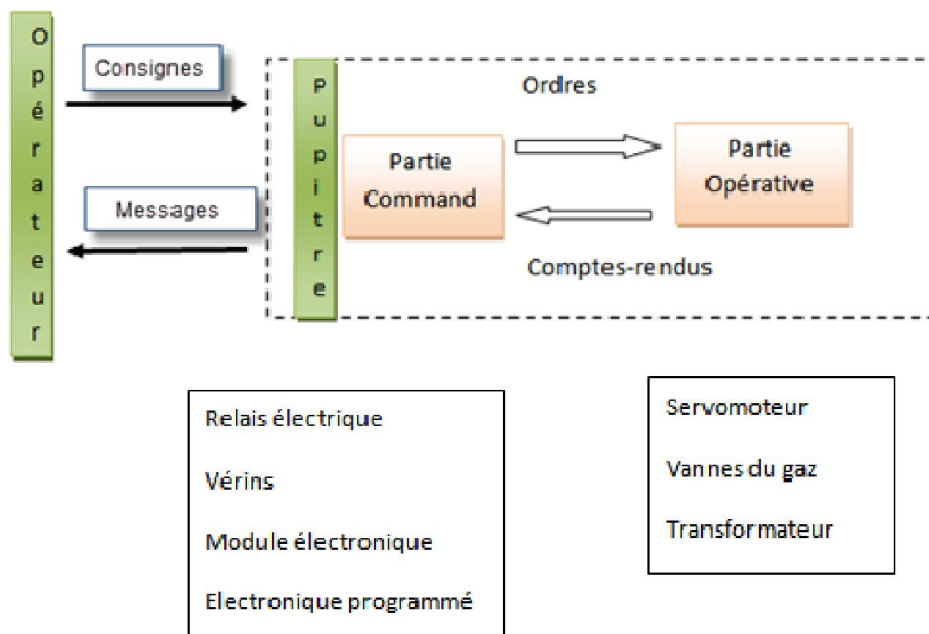
Dès sa conception, un système à automatiser doit être décomposé en trois parties :

- La partie commande ou automate qui élabore les ordres nécessaires à l'exécution du processus, en fonction des consignes qu'elle reçoit à l'entrée et des comptes rendus d'exécution qui lui sont fournis par la partie opérative.
- La partie opérative qui effectue les opérations en exécutant les ordres qui lui sont donnés par la partie commande.
- La partie relation ou dialogue qui représente l'échange des informations entre la partie commande et l'opérateur (pilote, usage, surveillant...) dont elle reçoit des consignes et à qui elle fournit des comptes rendus visuels ou sonores [2].



**Figure I.2 :** Structure d'un système automatisé.

Cette structure permet un dialogue profitable entre le futur utilisateur du système et l'automaticien responsable de la partie commande.



**Figure I.3 :** Dialogue entre la partie commande et la partie opérative

## **I. 2.5. Mise en œuvre d'un automate :**

La mise en œuvre de tout système automatisé implique une série de tâche qui constitue autant d'étapes successives naturellement interdépendantes [2] :

- L'étude préalable.
- L'étude proprement dite et préparation.
- Fabrication et essais
- Mise en route et exploitation.

## **I. 2.6. Caractéristiques d'un automatisme :**

Les automatismes sont des dispositifs qui permettent à des machines ou des installations de fonctionner automatiquement [2].

Un automatisme bien conçu :

- simplifie considérablement le travail de l'homme qui, libéré vis-à-vis de la machine, peut se consacrer à des activités plus noble ;
- réduit les tâches complexes, pénibles ou indésirables en les faisant exécuter par la machine ;
- facilite les changements de fabrication en permettant de passer d'une quantité ou d'un type de production à un autre ;
- améliore la qualité des produits en asservissant la machine à des critères de fabrication et à des tolérances qui seront respectées dans le temps ;
- accroît la production ainsi que la productivité ;
- permet de réaliser des économies de matière et d'énergie ;
- augmente la sécurité du personnel ;
- contrôle et protège les installations et les machines.

## I.3. Définition d'un API :

Un automate programmable industriel (API) est un appareil électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires.

Il exécute une suite d'instructions introduites dans sa mémoire sous forme de programme et s'apparente par conséquent aux machines de traitement d'information.

Trois caractéristiques fondamentales le distinguent totalement des outils d'informatiques tels que les ordinateurs utilisés dans les entreprises et tertiaires :

- Connexion directe aux différents capteurs et actionneurs grâce à ces entrées/sortie ;
- Fonctionnement dans des conditions industrielles sévères (température, vibrations, humidité, microcoupure de l'alimentation en énergie électrique...);
- Son aspect pratique grâce à la possibilité de sa programmation en utilisant un langage spécialement développé pour le traitement de fonctions d'automate (Step7). [3]

### I.3.1. Principe générale de fonctionnement d'un API :

Tous les automates fonctionnent selon le même mode opératoire :

- **Traitement interne** : l'automate effectue des opérations de contrôles et met à jour certains paramètres systèmes (détection des passages en RUN/STOP, mises à jour des valeurs de l'horodateur,...) ;
- **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.

Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique).[4]

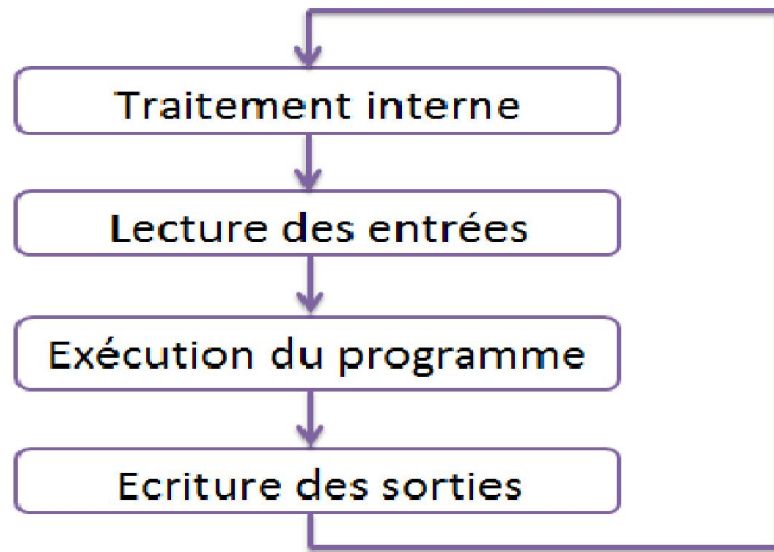


Figure I.4: Fonctionnement d'un API.

### I.3.2. Architecture d'un API :

#### I.3.2.1. Aspect extérieur :

Les automates peuvent être de type compact ou modulaire :

##### I.3.2.1.1. Compacte :

Il intègre le processeur, l'alimentation et les entrées/sorties. Il peut réaliser certaines fonctions supplémentaires et recevoir des extensions limitées. Il est généralement destiné à la commande de petits automatismes.

##### I.3.2.1.2. Modulaire :

Dans ce modèle le processeur, l'alimentation et les interfaces entrées/sorties résident dans des unités séparées (modules). Ces automates sont intégrés dans les automatismes complexes de grande puissance et de capacité de traitement.

#### I.3.2.2. Structure interne :

Un automate programmable industriel est donc constitué de :

##### I.3.2.2.1. Un module d'alimentation :

Le module d'alimentation transforme l'énergie externe provenant du réseau en la mettant en forme afin de fournir aux différents modules de l'API les niveaux de tension nécessaires à leur bon fonctionnement. Plusieurs niveaux de tension peuvent être utilisés par les circuits internes (3V, 5V, 12V, 24V...). Il sera dimensionné en fonction des consommations des différentes parties.

## **I.3.2.2.2. Une unité de traitement ou processeur :**

Le processeur gère l'ensemble des échanges informationnels en assurant :

- La lecture des informations d'entrée.
- L'exécution des instructions du programme mis en mémoire.
- La commande ou l'écriture des sorties.

## **I.3.2.2.3. Une mémoire programme :**

La mémoire programme de type RAM contient les instructions à exécuter par le processeur afin de déterminer les ordres à envoyer aux pré-actionneurs reliés à l'interface de sortie en fonction des informations recueillies par les capteurs reliés à l'interface d'entrée.

## **I.3.2.2.4. Une mémoire de données :**

La mémoire de donnée permet le stockage de :

- L'image des entrées reliées à l'interface d'entrée ;
- L'état des sorties élaborées par le processeur ;
- Les valeurs internes utilisées par le programme (résultats de calculs, états intermédiaires...).
- Les états forcés ou non des E/S.

## **I.3.2.2.5. Une interface d'entrée :**

L'interface d'entrée permet la connexion à l'API d'un multiple de capteurs pouvant être de type :

- TOR (logiques ou Tout Ou Rien) ;
- Numériques ;
- Analogiques.

Ces différentes entrées sont mises en forme par l'interface d'entrée avant d'être stockées dans la mémoire de données.

## I.3.2.2.6. Une interface de sortie :

L'interface de sortie permet la connexion de l'API aux de pré- actionneurs, qui peuvent être de type :

- TOR (logiques ou Tout Ou Rien) ;
- Numériques ;
- Analogiques.

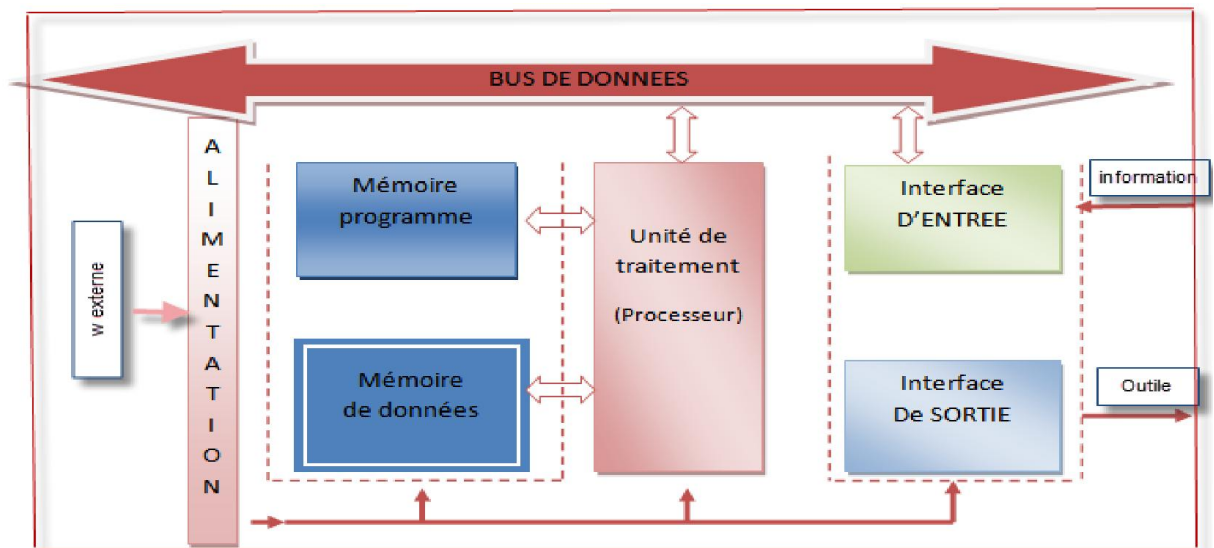


Figure I.5 : Structure interne des automates.

## I.3.3. Critère de choix de l'automate programmable industriel :

Après avoir étudié notre système dans les chapitres précédent, le choix des API revient considérer certains critères important tels que :

- Nombre et la nature des entrées/sorties ;
- Le type du processeur, la taille de la mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur ;
- Fonction ou modules spéciaux : certaine modules permettent de soulager le
- processeur en calcul afin de sécuriser le traitement et la communication avec le procédé ;
- Communication avec d'autre système ;
- La fiabilité et la robustesse ;

# Chapitre I Les automates programmables industriels

- Protection contre les parasites (champs électromagnétiques), baisse et pic de tension.

Pour notre travail, nous avons choisi à l'automate SIEMENS S7-300 et cela vue

- Le nombre d'entrées/sorties (23 /10) ;
- La nature TOR d'entrées/sorties.[5]

## I.4. Présentation d'un automate [6] :

De forme compacte ou modulaire, les automates sont organisés suivant l'architecture suivante :

- Un module d'unité centrale ou CPU, qui assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées/sorties, des mémoires RAM et EEPROM nécessaires pour stocker les programmes et les données.
- Un module d'alimentation qui, à partir d'une tension 220V/50Hz ou dans certains cas de 24V fournit les tensions continues  $\pm 5V$ ,  $\pm 12V$  ou  $\pm 15V$ .
- Un ou plusieurs modules de sorties « tout ou rien » (TOR) ou analogiques pour transmettre à la partie opérative les signaux de commande. Il y a des modules qui intègrent en même temps des entrées et des sorties.
- Un ou plusieurs modules de communication comprenant :
  - Interfaces série utilisant dans la plupart des cas comme support de communication, les liaisons RS-232 ou RS-422/RS-485 ;
  - Interfaces d'accès à un réseau Ethernet ;
  - Interface Profibus ;
  - Interface de type MPI.

La ci-dessous illustre un automate programmable industriel SIEMENS.

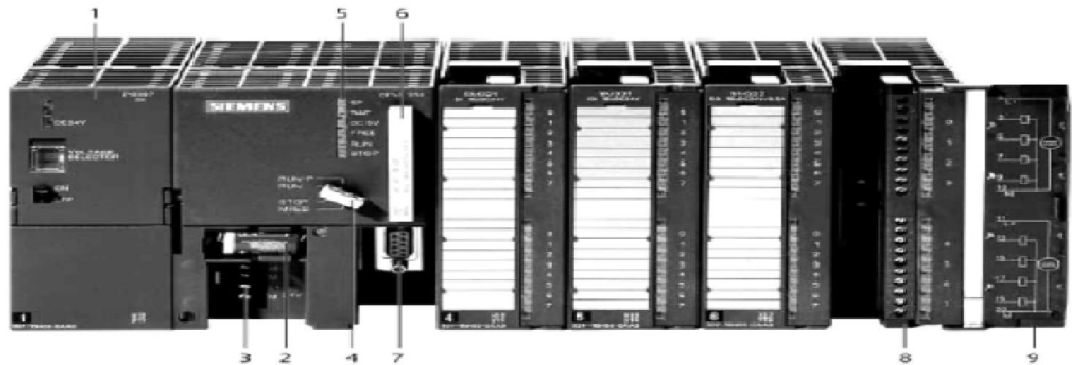


Figure I.6 : Automate Programmable Industriel SIEMENS [13].

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

## I.5. Présentation de quelques gammes SIMATIC :

Il existe plusieurs gammes de SIMATIC. On trouve le SIMATIC S7 et le SIMATIC M7 :

### I.5.1. SIMATIC S7 :

Dans la gamme S7 on distingue trois grandes familles d'automates programmables industriels décrites dans ces paragraphes qui suivent :

#### I.5.1.1. SIMATIC S7-200 :

La famille S7-200 est constituée de micro-automates programmables utilisables dans des applications d'automatisations variées. La Figure ci-dessous présente un micro-automate S7 -200. Son dessin compact, ses possibilités d'expansion, son faible prix et son important jeu d'opérations en font une solution idéale pour la commande de petites applications. On outre, le large choix de tailles et de tensions de CPU offre la souplesse nécessaire pour résoudre un problème d'automatisation.

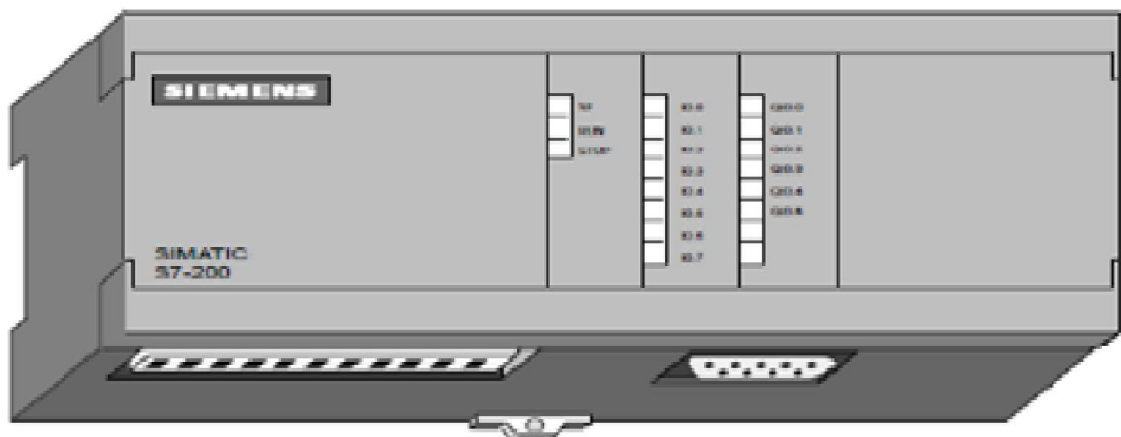


Figure I.7 : API SIMENS S7-200

Un automate programmable S7-200 consiste en une CPU S7-200 seule ou complétée de divers modules d'extension facultatifs connectés à cette dernière a l'aide d'un connecteur de bus fourni avec ce module d'extension[6].

## I.5.1.2. SIMATIC 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. Comme indiqué dans la figure suivante [6].

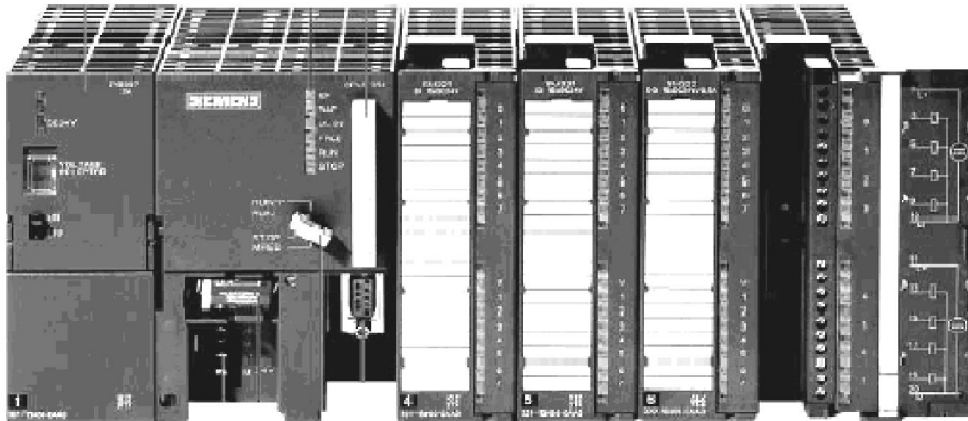


Figure I.8 : API SIMENS S7-300

## I.5.1.3.SIMATIC S7-400 :

La famille S7-400 est aussi constituée d'automates programmables de conception modulaire. Pratiquement chaque tâche d'automatisation peut être résolue par un choix approprié des constituants de S7-400 et avec la possibilité d'expansion de plusieurs modules.

Les modules se présentent sous forme de boîtiers que l'on adapte sur un châssis [6].

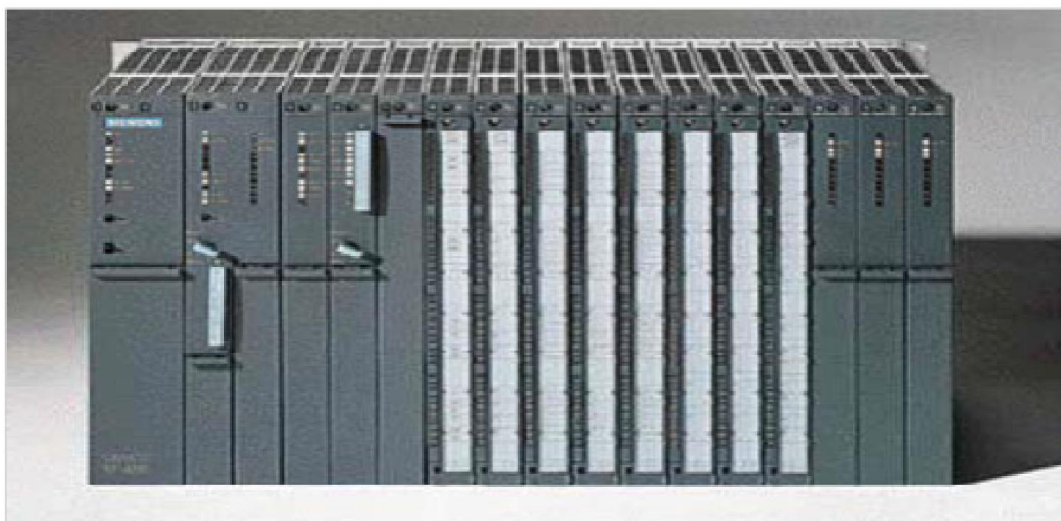


Figure I.9 : API SIMENS S7-400

## I.5.2. Présentation de l'automate à utiliser S7-300 :

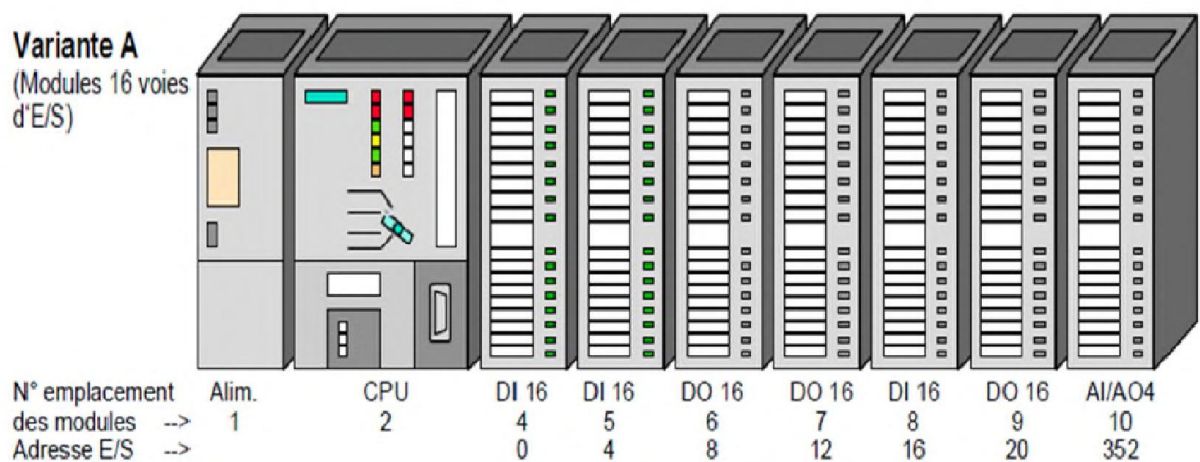
L'automate **S7-300** est un mini automate modulaire, pour les applications d'entrée et de milieu de gamme, fabriqué par la firme SIMENS. On peut le composer en fonction de nos besoins à partir d'un vaste éventail de modules.

La gamme des modules comprend :

- Des CPU de différents niveaux de performance.
- Des modules de signaux pour des entrées/sorties TOR et analogiques.
- Des modules de fonctions pour différentes fonctions technologiques.
- Des processus de communication (CP) pour les tâches de communications.
- Des modules d'alimentation pour le raccordement du S7-300 sur le réseau 230 V

[6].

La figure ci-dessous représente la constitution d'un S7-300.



**Figure I.10 :** Constitution d'API S7-300 [6].

L'automate est équipé des modules ci-dessous :

- Emplacement 1 : alimentation 24V/5A ;
- Emplacement 2 : CPU 314 ;
- Emplacement 3 : module de complémentaire
- Emplacement 4 : entrées TOR 16x24V ;
- Emplacement 5 : entrées TOR 16x24V ;
- Emplacement 6 : sorties TOR 16x24V 0.5A ;
- Emplacement 7 : sorties TOR 16x24V 0.5A ;
- Emplacement 8 : entrées TOR 16x24V ;
- Emplacement 9 : sorties TOR 16x24V 0.5A ;
- Emplacement 10 : module analogique 4AI/4 [6].

## **I.5.3. Critère de choix d'un API :**

Le choix d'un API est fonction de la partie commande à programmer. On doit tenir compte de plusieurs critères :

- Nombre d'entrées / sorties.
- Le temps de traitement.
- La capacité de la mémoire.
- Le nombre d'étapes ou d'instructions.
- Le nombre de temporisateurs

## **I.5.4. Description de l'Automate S7-300 :**

Le S7-300 est un automate modulaire d'une gamme excellente des produits SIMENS, il est synonyme de la nouvelle technologie des automates programmables. Le S7-300 est utilisé dans presque toutes les branches de l'industrie où les applications les plus variées. Sa modularité permet de réaliser des fonctions d'automatisations les plus diverses à partir des différents modules. Ses principales caractéristiques son :

- Sa puissance et sa rapidité ;
- La possibilité d'intégration de nouvelles taches ;
- Haute performance grâce aux nombreuses fonctions intégrées. [7]

## **I.5.5. Constitution de l'automate S7-300 :**

Un système d'automatisation S7-300 est un système modulaire, comprend les composantes indiquées ci-dessous :

### **▪ Module d'alimentation (PS) :**

Transforme la tension secteur en une tension d'alimentation, et délivre sous une tension de 24 V, un courant de sortie assigné de 2A, 5A et 10A.

### **▪ Unité centrale :**

C'est le cerveau de l'automate, exécute le programme utilisateur et commande les sorties, elle comporte les éléments suivants en face avant :

- Des Leds pour la signalisation d'état et de défaut ;
- Commutateur à clé amovible à 4 positions ;
- Raccordement pour tension 24 V DC ;
- Interface multipoint MPI pour console de programmation ou couplage à un autre système d'automatisation ;
- Compartiment pour pile de sauvegarde ;
- Logement pour carte mémoire.

# Chapitre I Les automates programmables industriels

## ▪ Module de signaux(SM) :

C'est des modules E/S sont sélectionnés en fonction de la plage de tension ou de la tension de sortie, utilisés pour les E/S TOR ou analogiques et qui est devisé : de sortie, utilisés pour les E/S TOR ou analogiques et qui est devisé :

## ▪ Modules d'entrée :

Permettent à l'automate de recevoir des informations prévenantes soit de la part des capteurs (entrée logique, analogique ou numérique) ou bien du pupitre de commande ;

## ▪ Modules de sortie :

Permettent de raccorder l'automate avec les différents prés-actionneurs (contacteurs, relais...) ainsi avec les actionneurs (moteurs, pompes...).

## ▪ Module coupleur (IM) :

C'est un coupleur qui permet la configuration multi rangée du S7-300, et assure la liaison entre les châssis et le couplage entre les différentes unités.

## ▪ Module de fonction (FM) :

Assure des taches lourdes en calcul ainsi des fonctions spéciale comme le positionnement, la régulation, le comptage, la commande numérique... etc.

## ▪ Processeur de communication (CP) :

Permet la communication entre plusieurs automates.

## ▪ Le Rais Profilé :

Constitue le châssis de S7-300.

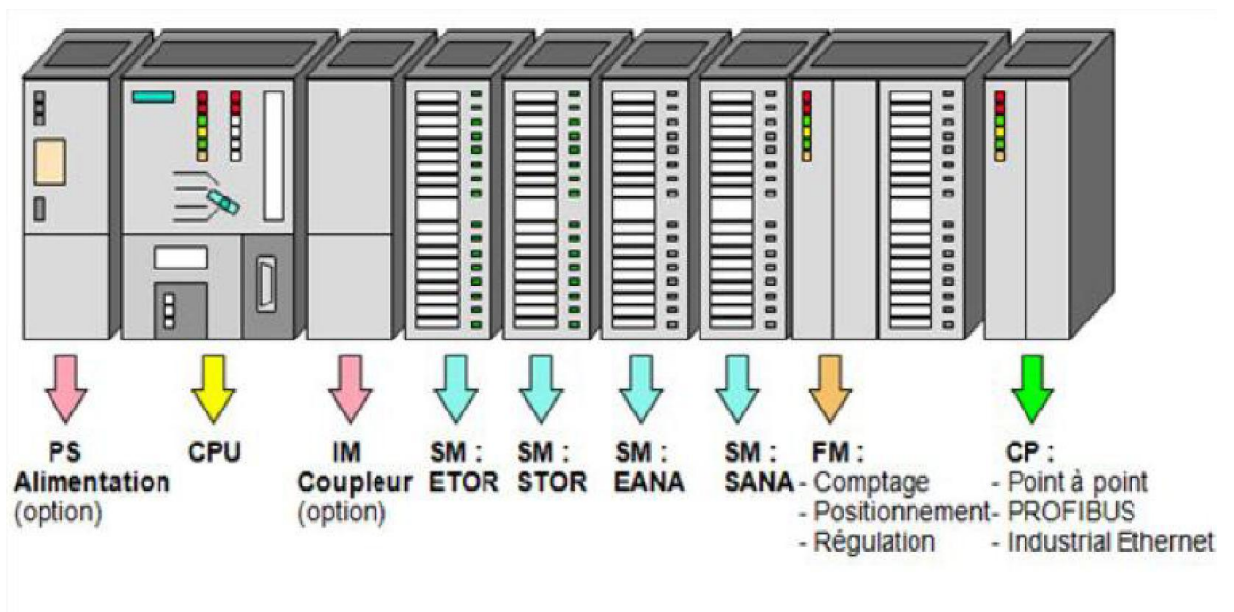


Figure I.11 : Constituants de l'automate S7-300.

## **I.6. Présentation de logiciel de programmation STEP7 :**

STEP7 fait partie de l'industrie logicielle SIMATIC. Le logiciel STEP 7 permet de concevoir, configurer, programmer, tester, mettre en service et maintenir les systèmes d'automatisation 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 projets.
- \_ La configuration et le paramétrage du matériel et de la communication.
- \_ La gestion des mnémoniques.
- \_ La création des programmes.
- \_ Le chargement de programmes dans les systèmes cibles.
- \_ Le test de l'installation d'automatisation.
- \_ Le diagnostic lors des perturbations dans l'installation [8].

## **I.7. Langages de programmation :**

Il existe plusieurs types de langage de programmation, qui font partie intégrante du logiciel de base. Ces langages sont :

### **I.7.1. Liste d'instructions (IL : Instruction List) :**

Langage textuel de même nature que l'assembleur (programmation des microcontrôleurs) ; très peu utilisé par les automaticiens.

### **I.7.2. Langage littéral structuré (ST : Structured text) :**

Issu de l'informatique, il présente des analogies avec le PASCAL ; il offre des possibilités telles :

Si

Alors

Autrement

Fin si

Ce langage facilite donc la mise en œuvre d'algorithme complexe comportant beaucoup de traitement numérique. Il permet aussi des passerelles vers des blocs extérieurs, ainsi que des traitements particuliers. En contrepartie, il est moins commode pour la mise au point de fonctions booléennes.

### **I.7.3. Langage à contacts (LD : Ladder Diagram) :**

Langage graphique développé pour les électriciens. Il utilise les symboles tels que : contacts, relais et blocs fonctionnels et s'organise en réseaux (labels).

L'application est représentée par un ou plusieurs réseaux (x). Un réseau est formé d'éléments graphiques, et éventuellement de blocs fonctionnels, connectés entre eux, partant d'une barre d'alimentation à gauche, et se terminant par une barre à droite. Il correspond au cheminement d'un flux : d'énergie (cas d'une application ou le programme pourrait être remplacé par un ensemble de relais électromagnétiques, avec passage éventuel de courant électrique) ; il circule alors conventionnellement de gauche à droite ; de signaux (cas des blocs fonctionnels, ce qui permet d'introduire des éléments tels que temporisateurs, compteurs, opérateurs arithmétiques).

Au réseau est associée une étiquette, comme en informatique classique.

Dans les cas de traitement booléen, les éléments fondamentaux sont des contacts normalement fermés (NF) ou normalement ouverts (NO), et des bobines (avec possibilité de mémorisations de l'information reçue).

## **I.8. Présentation de Win CC Runtime Advanced :**

### **I.8.1. Introduction :**

Lorsque la complexité des processus augmente et que les machines et 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). Un système 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 Win CC Runtime Advanced (sur le pupitre opérateur) et une interface entre Win CC Runtime Advanced et le système d'automatisation [9].

### **I.8.2. Les taches d'un système IHM :**

- \_ Représentation du processus
- \_ Commande du processus
- \_ Vue des alarmes
- \_ Archivage de valeurs processus et d'alarmes
- \_ Gestion des paramètres de processus et de machine [9].

## **I.9.Conclusion :**

Nous avons présentés dans ce chapitre, l'étude des différentes composantes de l'automate programmable industriel de la gamme S7-300 et leurs caractéristiques. Qui nous a permis d'évaluer leurs importances et de constater leur rôle primordial dans la gestion des processus industriels. Ainsi que le rôle du logicielSTEP7 et le logiciel de supervision Win CC Runtime Advanced que nous avons présentés dans le troisième chapitre qui est l'automate choisit, pour réguler les différentes taches d'installation du Station de remplissage c'est l'objectif de notre travail.

# **Chapitre II : Généralité sur les remplisseuses et régulation**

## II.1. Introduction :

Dans ce chapitre, nous allons présenter les différentes machines de remplissage utilisées de manière industrielle, et permettant de placer dans un récipient (bouteilles, sachets plastiques, flacons, etc.) tout type de contenu, et Il existe différentes machines de remplissage de produits liquides (huiles, sirops, cosmétiques) et les remplisseuses de produits solides (poudres, granulés, gélules,...etc.).

Et nous allons présentées les définitions théoriques, objectives et les caractéristiques sur la régulation dans le domaine industriel ainsi que leur domaine d'application.

## II.2. Les remplisseuses :

### II.2.1. Les remplisseuses rotatives:

Pour les cadences élevées, les remplisseuses sont rotatives en continue avec un carrousel dont le nombre de becs peut varier de 4 à plus de 100[10].



Figure II.1: Vue générale d'une remplisseuse rotative.

### Principe de fonctionnement :

La tulipe de centrage centre la bouteille sous le robinet de soutirage. Le vérin de levage soulève la bouteille contre le robinet de soutirage. Le joint de la tulipe de centrage presse la bouteille hermétiquement contre le robinet [10].



Figure II.2: Technique de soutirage.

### II.2.2. Les remplisseuses linéaires:

Les remplisseuses linéaires sont des machines à un ou plusieurs couloirs pour les cadences faibles à moyennement élevées. Les remplisseuses linéaires ont plusieurs avantages :

- Utilisant peu de pièces de format ;
- Elles se caractérisent par une structure claire ;
- Une large gamme de volumes et de flacons ;
- Un nettoyage facile [10].



Figure II.3: Remplisseuse linéaire.

### II.3. Systèmes de remplissage :

Le remplissage d'une bouteille advient lorsque le liquide est transféré d'une cuve de stockage à un contenant. Ce qui distingue une technique de remplissage par rapport à une autre, c'est la manière dont on détermine la quantité exacte de produit distribué dans chaque contenant. Il existe moult systèmes et techniques à notre disposition. En gros, nous pouvons en distinguer trois catégories: à niveau, à débitmètre et pondérale [11].

Il existe un autre type de remplisseuse en littérature qui appartient à la famille des remplisseuses volumétriques.

Et voici le principe de fonctionnement de ces différentes machines

### II.3.1. Remplisseuse à niveau :

Le soutirage par niveau constant est la technologie la plus utilisée car très robuste et peuchère [10].

#### • Principe de fonctionnement :

Les vannes de remplissage s'ouvrent respectivement plus tard dans la création du projet via la fonction Recette IHM en fonction du temps de remplissage, le remplissage est différent pour le composant, la sortie du mélangeur est réglée sur 1 après 4 secondes la sortie est réinitialisée à 0 et le mélangeur est désactivé, après, nous activons le bloc LIST pour contrôler le tapis et transférer les bouteilles jusqu'au remplissage lorsque la garniture ouvre la vanne concernée pendant 3 secondes pour remplir la bouteille, lors de chaque remplissage le compteur prend en compte le nombre de bouteilles déjà remplies une étape par étape 10 bouteilles maximum La bouteille est transférée remplie vers le poste d'étiquetage une fois celle-ci remplie et sortie du terminal activé, le niveau d'étiquetage est réglé sur 1 au niveau de la station d'étiquetage et est étiqueté avec la date limite apposée sur chaque bouteille. Si la boisson est terminée, la séquence est à nouveau exécutée dès le début. Si le remplissage n'est pas encore terminé, répétez les étapes jusqu'à ce que les 10 bouteilles soient remplies et remplissez-le pour terminer.

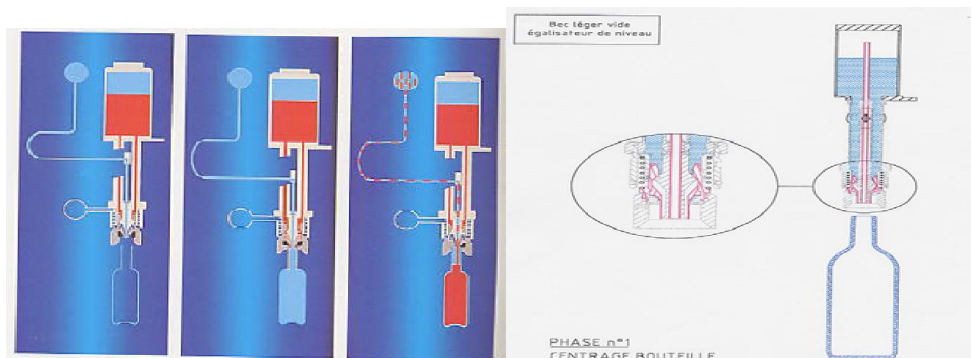


Figure II.4: Remplisseuse

### II.3.2. Remplisseuse à débitmètre :

La machine mesure le volume de produit qui entre dans la bouteille. Pour ce faire, un débitmètre (magnétique ou massique) est prévu sur chaque buse de remplissage. La machine qui utilise cette technologie est appelée remplisseuse électronique [10].

**Avantage :**

- La machine est simplifiée mécaniquement car il n'y a pas de montée de bouteilles avec des sellettes et pas contact avec la bouteille pour éviter des risques de contamination.
- Les becs (ou robinets) sont tout inox sans joint à fermeture magnétique et sans canule pour éviter la recirculation ou la perte de produit.
- Un système électronique de mesure de la dose permet une traçabilité à la bouteille sur ordinateur.

Il existe 2 types de débitmètre :

- Volumique pour les liquides conducteurs électriquement ;
- Massique en cas de non conductivité.

Le massique est plus cher que le volumique, dans les 2 cas, la précision est excellente, de l'ordre de 0,5% de la consigne [10].

**Inconvénients:**

- Cette technologie est souvent un peu plus chère à l'achat que le niveau constant ou le volumétrique.
- L'électronique prend place dans la machine et impose un minimum de connaissances à l'équipe de maintenance. Ceci dit, il en est de même avec les autres machines de la ligne : souffleuse et étiqueteuse [10].

**II.3.3. Remplisseuse pondérale :**

Est une technologie adaptée aux produits à fortes valeur ajoutée car le dosage est très précis, de l'ordre de 0,5% de la consigne. Les becs sont souvent tout inox et sans contact avec la bouteille, ce qui limite les risques d'une contamination. Cette technologie est aussi intéressante pour les produits moussants. Le dosage pondéral est intéressant car il permet de faire un tarage de la bouteille vide avant remplissage pour augmenter la précision, en particulier pour les bouteilles en verre dont la variation de poids due au moulage est réelle.

L'électronique associée aux balances permet d'avoir une traçabilité complète pour les dossiers qualité et de suivi de production. Le logiciel développé pour faire la mesure de poids et contrôler l'ouverture et la fermeture du bec, doit prendre en compte :

- Une correction de la force de Coriolis due à la rotation de la machine qui influe la pesée et donc la précision de dosage.

- Une correction statistique de la « queue de chute », c'est-à-dire la quantité de produit entrain de s'écouler entre le bec et le niveau dans la bouteille et qui finalement pèsera dans la bouteille remplie [10].

**II.3.4. Remplisseuse volumétrique :**

Est une technique simple et toute mécanique. Le principe est celui d'un piston, ou une seringue, qui aspire une dose dans la cuve supérieure de la machine et la réinjecte dans la bouteille par un clapet de sortie.

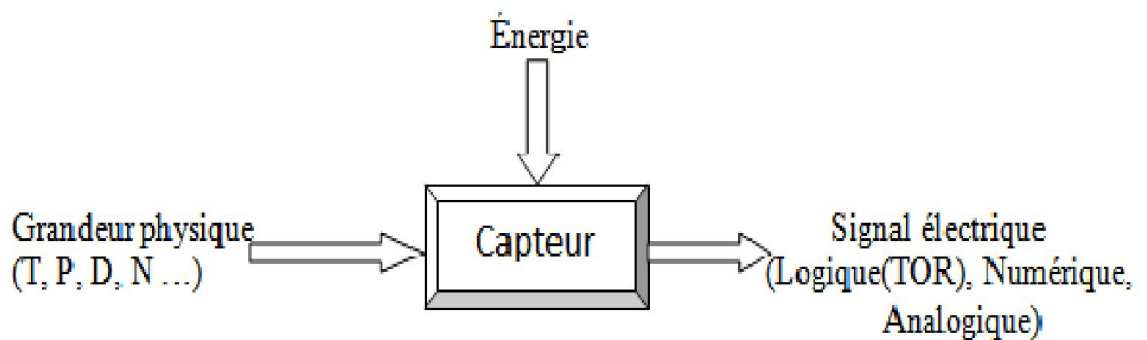
Une came plus ou moins inclinée par des manivelles ou un moteur électrique permet de régler cette dose. La maintenance est importante pour les jeux de galets et les joints de pistons. Le système permet d'avoir des canules pour le remplissage par le fond, en particulier pour les produits visqueux qui peuvent faire des strates dans le récipient rempli et transparent.

Il faut aussi faire attention à la variation de volume avec la température car le volume injecter peut varier par exemple lorsque le processus est froid le matin puis chaud en fin de journée [10].

**II.4.Elément pour fonction de détection :**

**II.4.1.Capteurs :**

Ce sont les éléments placés sur les installations pour détecter les informations locales qui constituent les données de l'automatisation.



**Figure II.5:Capteur.**

**II.4.1.1. Capteurs passifs :**

Ce sont tous les capteurs équivalents à un contact se ouvert ou fermé libre de potentiel (fin de course, bouton poussoirs...) ou aux composants passifs (thermistance, potentiomètre...) [12].

**II.4.1.2. Capteurs actifs :**

Ce sont tous les capteurs qui nécessitent une source de tension. Exemple : détecteur de proximité inductif ou capacitif, cellule de détection photoélectrique [12].

**II.4.2.Principales caractéristiques des capteurs :**

- **L'étendue de la mesure** : C'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.
- **La stabilité** : c'est la plus petite variation d'une grandeur physique que peut détecter un capteur.
- **La rapidité** : c'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information prise en compte par la partie commande.
- **La précision** : c'est la capacité de respectabilité d'une information position, d'une vitesse,...

**II .4.3.Familles des capteurs :****II.4.3.1.Les capteurs TOR :**

Ce type de capteur permet de détecter un événement ou un objet lié au fonctionnement du système technique. Le signal électrique en sortie de ce capteur est de type logique (signal acceptant 2niveaux : niveau logique 0 (NL0) ou niveau logique 1 (NL1)).



**Figure II.6:** Capteur tout ou rien

**II.4.3.2.Capteurs analogique :**

Dans la pratique industrielle, on donne à ce type de matériel le nom de capteur, si on lui inclut une électrique de puissance dans son boîtier, on parle de capteurs

transmetteurs, l'information peut prendre toutes les valeurs possibles entre 2 certaines valeurs limites [13].



**Figure II.7:**Capteur analogique.

#### **II.4.3.3. Capteur numérique :**

Selon le cas portent les noms en pratique de codeurs ou de compteurs, l'information fournie par ce capteur permet à la partie commande d'en déduire un nombre binaire sur n bits, on parle alors d'un capteur numérique [13].



**Figure II.8:** Capteur numérique.

- **Choix d'un capteur :**

Les critères de choix dépendent de l'exploitation qui en est faites dans le système technique.

Le choix sera fait selon :

- ✓ Etendue de Mesure : qu'elle est la plage de mesures que l'on souhaite réaliser dans le système conçu ?

- ✓ Précision : de quelle précision a-t-on besoin dans le cadre du système technique ? Cette précision dépend de l'élaboration du cahier des charges.

**II.5. Mesure et détection de niveau :**

La conduite des opérations de transfert et de stockage de liquides dans des réservoirs requiert que puisse être apprécié l'état de leur remplissage, c'est-à-dire le niveau qu'atteint le produit présent. La saisie d'information relative au niveau peut être effectuée sous deux formes : la mesure continue et la détection de seuil.

En mesure continue, un capteur et son conditionneur délivrent un signal dont l'amplitude ou la fréquence traduisent la valeur du niveau de liquide dans le réservoir : à chaque instant, l'opérateur peut donc connaître le volume exact de produit présent ou le volume de stockage encore disponible.

En détection, le dispositif détecteur, constitué en fait d'un capteur, délivre une information binaire signalant qu'un niveau seuil déterminé est ou non atteint : la détection d'un niveau haut permet de stopper un remplissage, évitant le débordement ; la détection d'un niveau bas arrête l'extraction du produit, assurant une réserve minimale dans la cuve et évitant la marche à vide des pompes ; l'association de deux détecteurs de seuil, haut et bas respectivement, permet d'automatiser les opérations de remplissage et d'extraction des cuves. [14] [15]

Le choix d'une méthode appropriée amène à prendre en compte :

- Les propriétés physiques et chimiques du liquide ainsi que leurs variations éventuelles : conductivité, constante diélectrique, masse volumique, viscosité, formation de mousse, caractère corrosif ;
- Les conditions de stockage : température, pression, présence d'un agitateur ;
- Les facilités d'implantation. [15]

**II.5.1. Le capteur de niveau :**

Un capteur de niveau est conçu pour la détection de niveau de liquides de toutes natures, un capteur de niveau pour liquides est parfaitement fiable même en présence de dépôts, leur avantage est très vaste comme la protection contre la marche à vide des pompes.

Le principe de mesure est préconisé dans des conditions extrêmes où les autres principes de mesure ne conviennent pas ; il est équipé d'un détecteur synthétique souple pour la mesure continue de niveau ou la mesure d'interface sur des cuves rondes ou coniques. [16]

**II.5.2. Principe de fonctionnement :**

Son principe de fonctionnement est très simple, une source faiblement radioactive émet des radiations qui sont atténuées par la paroi de la cuve et par le produit, le détecteur, fixé à l'opposé de l'émetteur, réceptionne des radiations dont la puissance est proportionnelle au niveau. [16]

La figure suivante nous montre le capteur de niveau :



**Figure II.9:** Capteur de niveau.

**II-6.La régulation automatique****II .6.1.Définition de la régulation automatique**

La régulation automatique regroupe l'ensemble des moyens matériels et techniques mis en œuvre pour maintenir automatiquement (pas d'intervention Manuelle) une ou plusieurs grandeurs physiques (installation de production, robot, alimentation électronique stabilisée, ...etc.) 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.

Lorsque des perturbations ou un changement de consigne se produisent, la régulation automatique provoque une action correctrice sur une autre grandeur physique, parmi les grandeurs d'entrée du procédé (grandeur réglant), afin de ramener la grandeur réglée vers sa consigne initiale (cas de perturbations) ou vers sa nouvelle consigne (cas de changement de consigne c'est-à-dire changement de point de fonctionnement).

Les méthodes de l'automatique offrent donc la possibilité de modifier le comportement statique et dynamique d'une grandeur physique, afin qu'elle évolue conformément aux exigences de l'application [1][17].

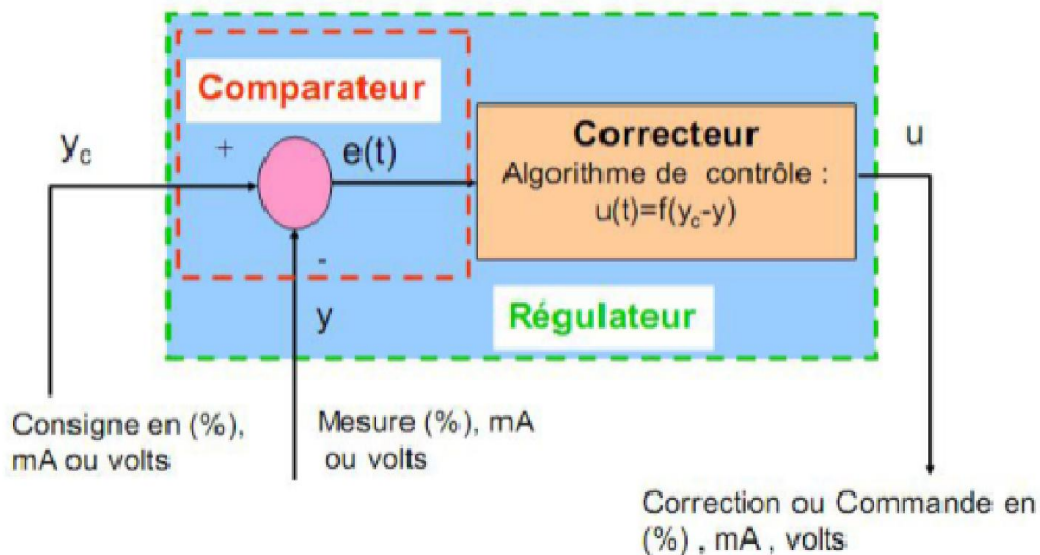


Figure II.10 : Présentation d'un schéma fonctionnel d'un régulateur

**II.6.2. Objectif de la régulation automatique**

L'objectif d'une régulation automatique d'un procédé est de le maintenir le plus près possible de son fonctionnement optimum, qui a été prédéfini ultérieurement par un cahier des charges (conditions ou performances imposées).

Aussi il faut impérativement prendre en compte les aspects de protection du personnel, des installations techniques et le respect de l'environnement. Le cahier des charges définit les critères qualitatifs à imposer qui sont traduits le plus souvent par des critères quantitatifs, comme par exemple, la stabilité, la précision et la rapidité [18].

**II 1.6.3. Notion de Boucle Ouverte/Fermée**

**II 1.6 .3.1. Système boucle ouverte**

Pour une variation manuelle d'amplitude finie de la commande u, on a une variation de la grandeur à maîtriser ou à réguler.

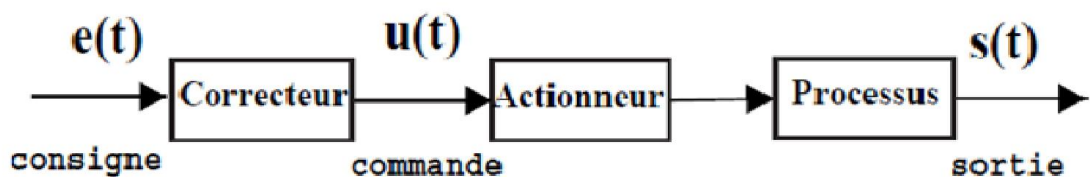


Figure II.11 : Système en BO

II.6 .3.2.Système boucle fermée :

La grandeur réglant exerce une influence sur la grandeur réglée, pour la maintenir dans des limites définies malgré les perturbations.

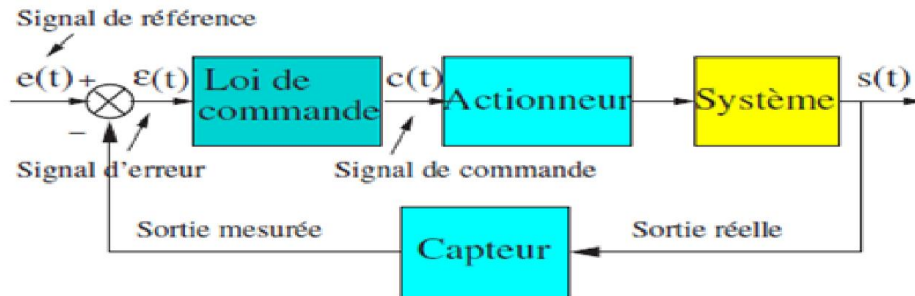


Figure II.12 : Système en BF

II .7.Définition de la régulation PID :

Le régulateur standard le plus utilisé dans l'industrie est le régulateur PID (proportionnel intégral dérivé), car il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse, ...) d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre.

Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et est relativement robuste par rapport aux variations des paramètres du procédé.

Si la dynamique dominante du système est supérieure à un deuxième ordre, ou si le système contient un retard important ou plusieurs modes oscillants, le régulateur PID n'est plus adéquat et un régulateur plus complexe (avec plus de paramètres) doit être utilisé, au dépend de la sensibilité aux variations des paramètres du procédé[1].

- Loi de commande du régulateur PID :

$$u(t) = k_p * (e(t) + \frac{1}{T_i} * \int_{-\infty}^t e(\tau) * d\tau + T_d * \frac{de}{dt}) \tag{II -1}$$

- Fonction de transfert du régulateur PID :

$$G_C(p) = \frac{U(P)}{E(P)} = K_p * \frac{1 + P * T_i + P * T_i * T_d}{P * T_i} \tag{II-2}$$

II .7.1.Notion de correcteur PID :

Sous forme d'un tableau récapitulatif, on résume les avantages et les limitations des actions de base des régulateurs PID :

Action	Points forts	Points faibles
<b>P</b>	Action instantanée	Ne permet pas d'annuler une erreur statique mais permet de la réduire
<b>I</b>	Annule l'erreur statique	Action lente Ralentit le système (effet déstabilisant)
<b>D</b>	Action très dynamique	Améliore la rapidité Apporte un effet stabilisant Sensibilité aux bruits Forte sollicitation de l'organe de commande

Table II .1 : Notion de correcteur PID.

**P = KR** :Est l'action proportionnelle, sur la plupart des régulateurs, on règle la Bande Proportionnelle au lieu de régler le gain du régulateur :

**I = 1/T i** :(min-1 en général) : est l'action intégrale

**D = Td** :(s en général) : est l'action dérivée

II .7.2.Structure des régulateurs électroniques :

Il existe trois types d'algorithme PID, le PID série, le PID parallèle et le PID mixte.

II .7.2.1.PID parallèle :

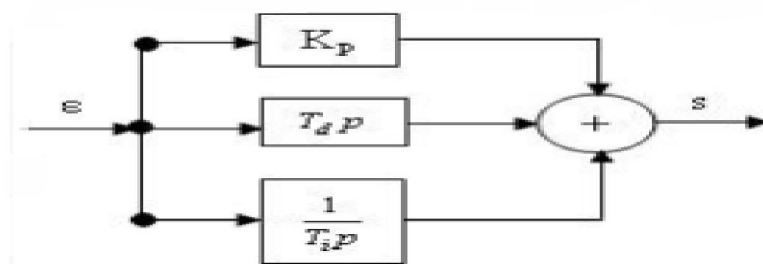


Figure II .13: Schéma d'un régulateur PID parallèle

Loi de commande s'écrit :

$$\frac{S(p)}{\varepsilon(p)} = K_p + T_d p + \frac{1}{T_i p} \tag{II -3}$$

## II .7.2.2.PID série :

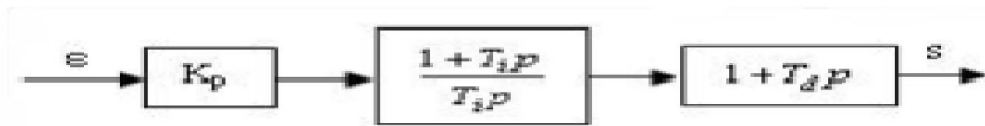


Figure II .14: schéma d'un régulateur PID série

Loi de commande s'écrit :

$$\frac{S(p)}{\varepsilon(p)} = K_p * \left(1 + \frac{1}{T_i p}\right) * (1 + T_d p) \quad (\text{II -4})$$

## II .7.2.3.PID mixte :

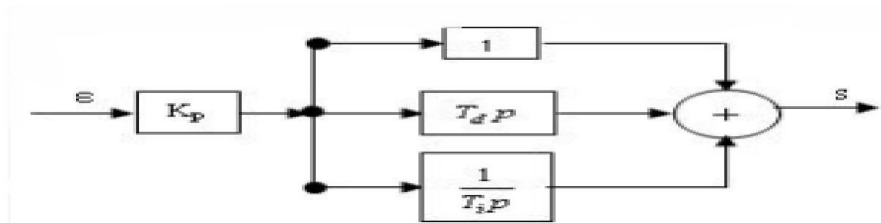


Figure II.15: schéma d'un régulateur PID mixte

## II.7.3.Utilisation :

Vous pouvez utiliser le régulateur comme régulateur PID de maintien mais régulations aussi comme régulateur en cascade, de mélange ou de rapport dans des à plusieurs boucles. Ses 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 et modulation de largeur d'impulsions pour régulations à deux ou trois échelons avec organes de réglage proportionnels[19].

## II .8.Conclusion :

Ce chapitre nous a permis d'avoir un aperçu sur les éléments essentiels constituant la remplisseuse, et de comprendre leurs fonctionnements et le rôle de chacun deux, et l'importance de ses éléments dans la chaine d'automatisation de notre système.

Nous avons présenté les différentes méthodes de la régulation, prouvent être résumées par l'utilisation des régulateurs PID, toute en essayant de répondre au cahier des charges élaboré.

# **Chapitre III :**

**Description Logiciel de programmation  
et simulation et supervision**

### **III.1. Introduction :**

Pour identifier la station de remplissage, vous constaterez que le programme implémentera des solutions d'automatisation avec un système intégré comprenant STEP7, SIMATIC WINCC et PLC SIM en concevant et en automatisant SIEMENS TIA PORTAL.

Dans ce chapitre, nous allons révéler à travers l'analyse fonctionnelle et la supervision des résultats de la programmation d'automatisation de notre système et de la simulation utilisée dans ce projet.

### **III.2. Logiciel de programmation « TIA Portal V13 » :**

#### **III.2.1 .Présentation du logiciel :**

La plate-forme TIA (Totally Integrated Automation) Portal est la dernière évolution des logiciels de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré, dans un seul logiciel cette plate-forme regroupe la programmation des différents dispositifs d'une installation. On peut donc programmer et configurer, en plus de l'automate, les dispositifs HMI les variateurs... etc. [20]

#### **III.2.2 .La conception d'un programme avec TIA PORTAL V13 :**

La stratégie à suivre pour faire la conception d'un programme en utilisant la plate-forme TIA PORTAL V13 est :

- La création d'un nouveau projet ;
- La configuration matérielle ;
- La compilation et le chargement de la configuration ;
- La création de la table des mnémoniques ;
- L'élaboration du programme ;
- La simulation avec le logiciel ;
- La visualisation d'état du programme (le test).

La conception d'une solution d'automatisation se fait par deux alternatives, soit on commence par la programmation ou par la configuration matérielle, dans notre cas on a commencé par la configuration [21].

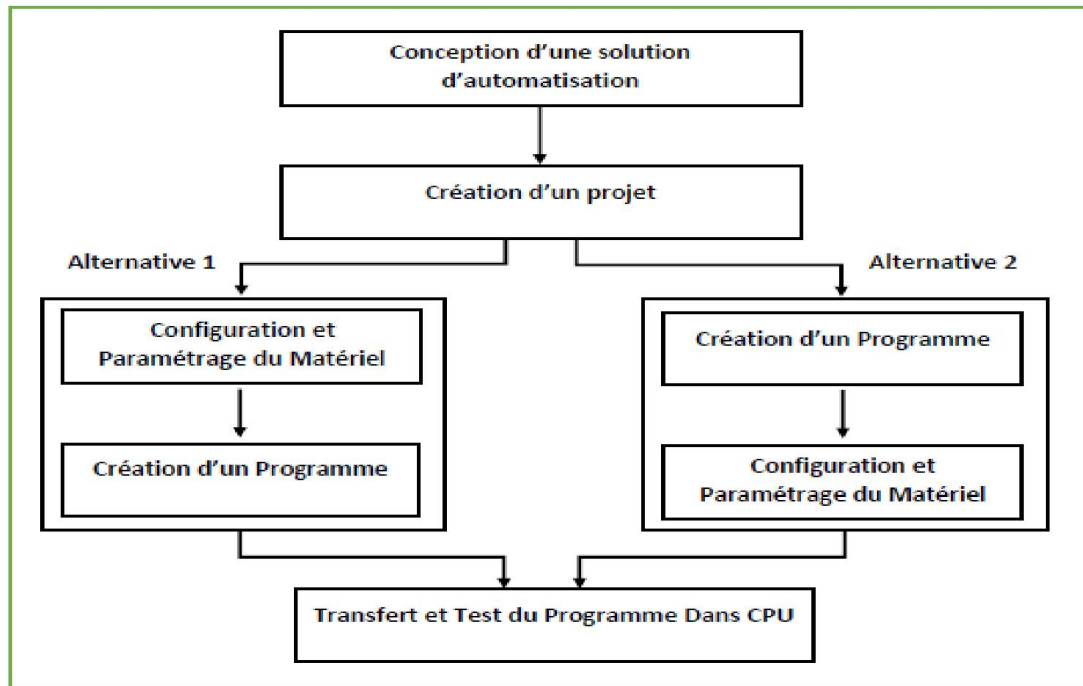


Figure III.1 : Organisation pour la création d'un projet sous TIA PORTAL [20].

### III.2.3. Vue du portail :

Chaque portail permet de traiter une catégorie de tâches (actions). La fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionnée [22].

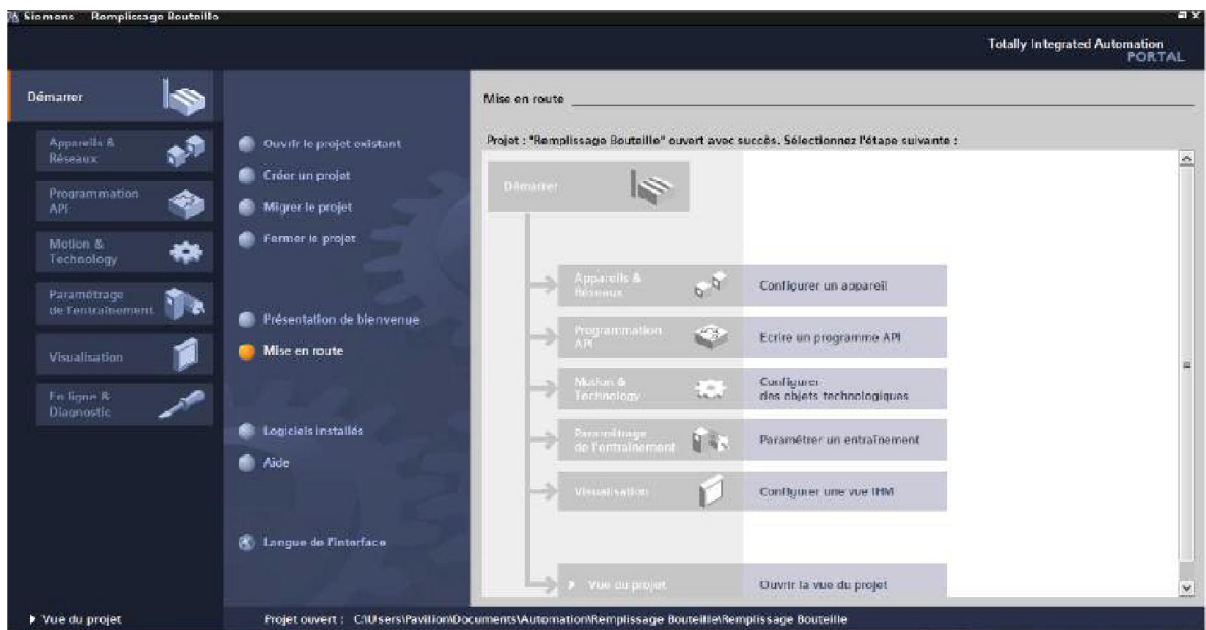


Figure III.2 : Vue détaillée du portail

### III.2.4. Vue du projet :

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée [22].

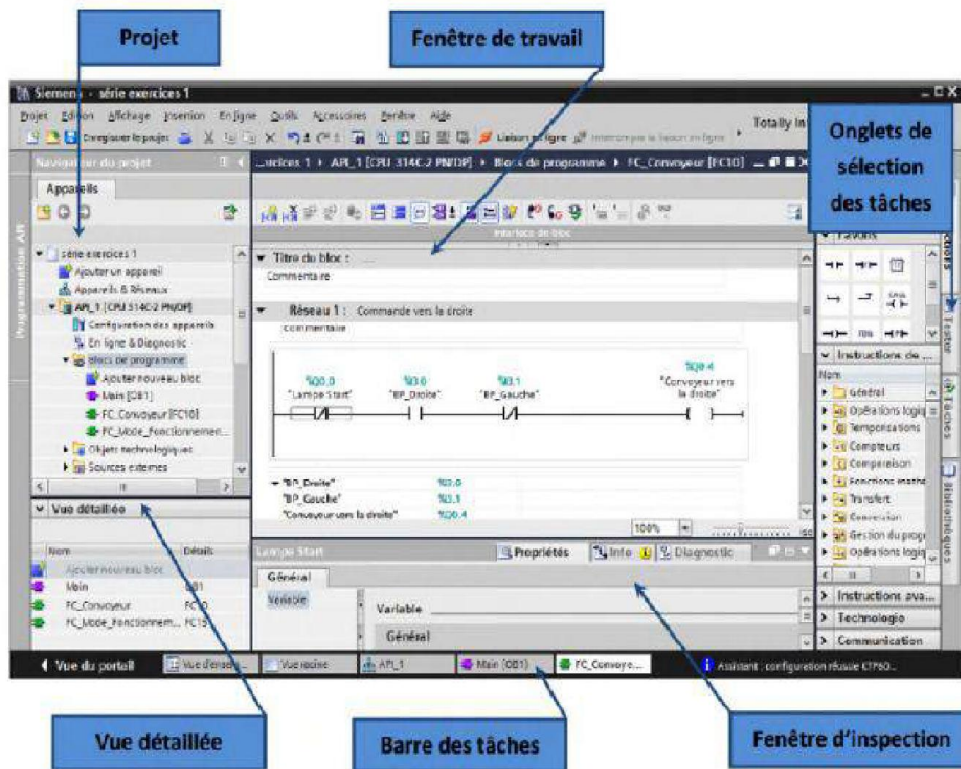
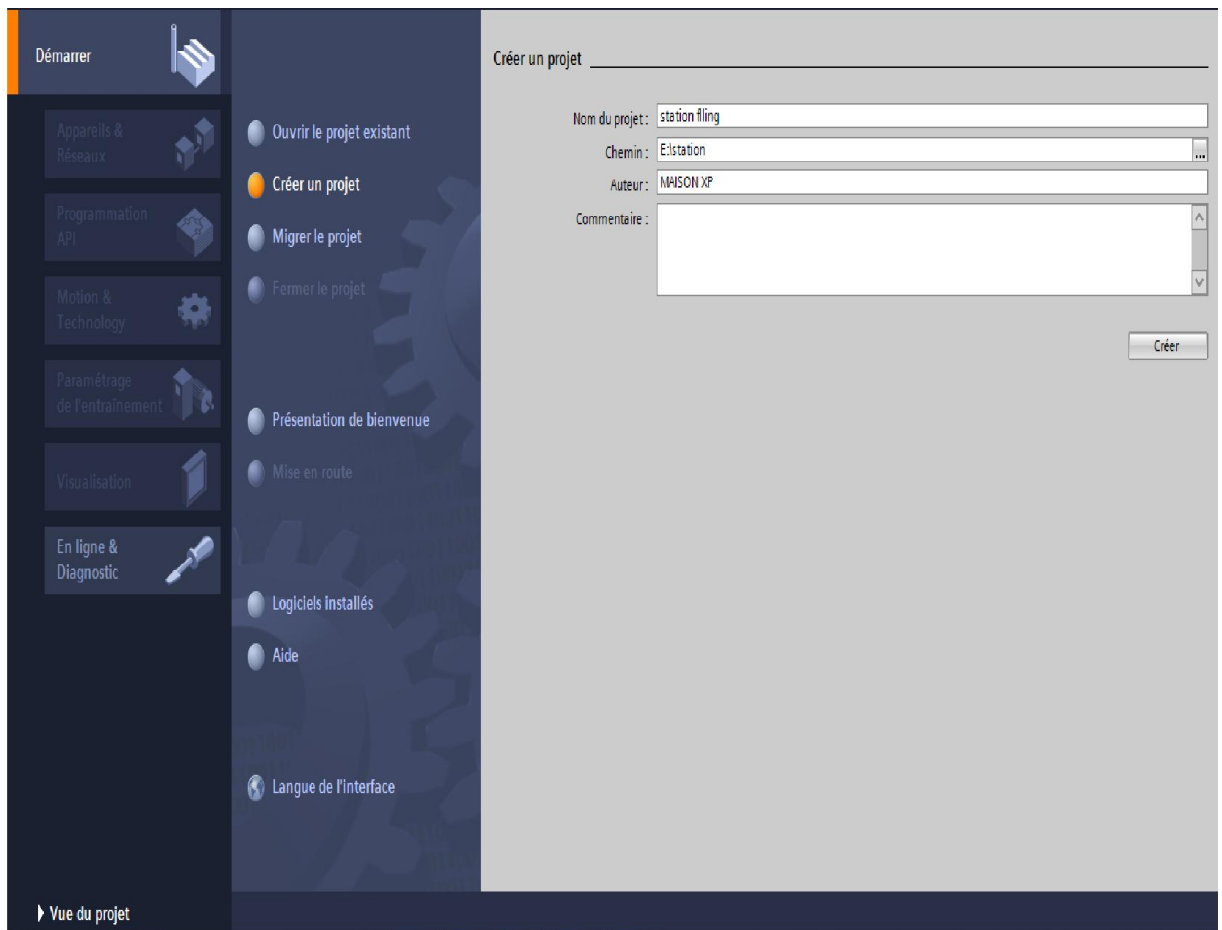


Figure III.3 : Vue détaillée du projet.

### III.2.5. Création du projet dans TIA PORTAL V13 :

Afin de créer un nouveau projet TIA PORTAL V13, nous utilisons « l'assistant de création de projet », en cliquant sur « créer un projet » ce qui nous permet de commencer la configuration, cette méthode nous permet de gérer notre projet aisément.

En sélectionnant l'icône « créer un projet », on affiche la fenêtre principale, on remplit le schéma vide de notre fenêtre et on appuie sur le bouton « créer », la figure suivante représente la mise en route de notre projet.



**Figure III.4 :** Mise en créer du projet.

On passe à la deuxième étape en cliquant sur le bouton « appareils et réseaux » ce qui nous permet de choisir les appareils qui constitueront notre système. Nous devons choisir un type d'automate PLC (programmable Logic Controller) et une interface homme machine IHM.

### **III.2.6. Configuration matérielle (Partie Hardware) :**

Notre partie dédiée à la configuration matérielle se divise en deux grandes parties PLC et IHM.

L'automate choisi est le S7-300 de SIEMENS et ses modules complémentaires, il est doté des éléments suivants :

- **Emplacement 1 :** PS 307 5A\_1 représente notre module d'alimentation.
- **Emplacement 2 :** CPU 315-2 PN/DP ; représente notre processeur.
- **Emplacement 4 :** DI8/DO8 x DC24V / 0,5A ; le module d'entrée/de sortie numérique

La figure ci-dessous est une représentation de notre automate S7-300.

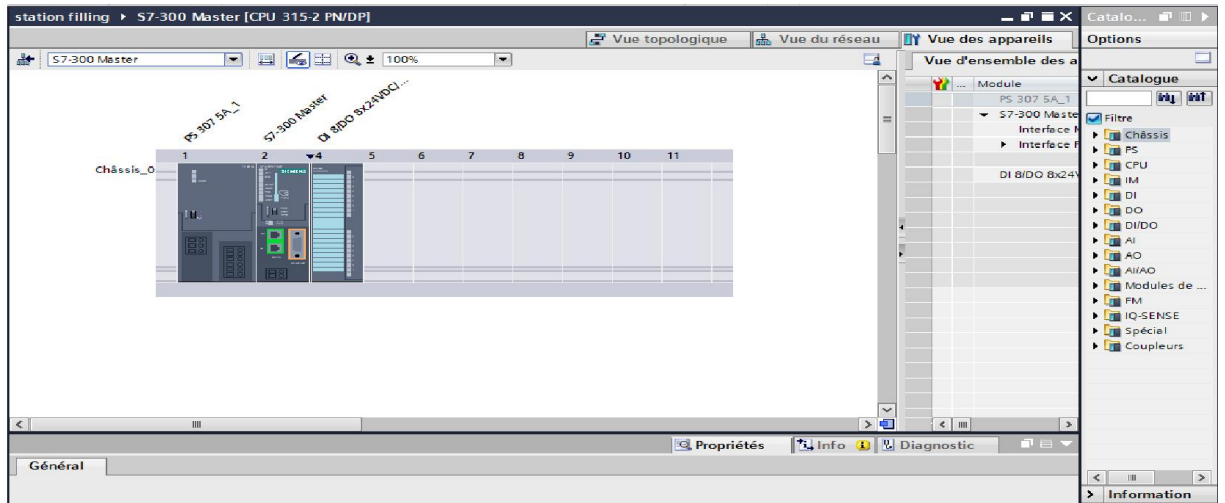


Figure III.5 : Représentation de notre automate S7-300.

### III.2.7. Adresse Ethernet de la CPU :

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet. Un double clic sur le connecteur Ethernet de la station fait apparaître la fenêtre d'inspection permettant de définir ses propriétés.

Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même réseau. On utilisera comme adresse pour l'automate 175.248.165.1 de l'automate.

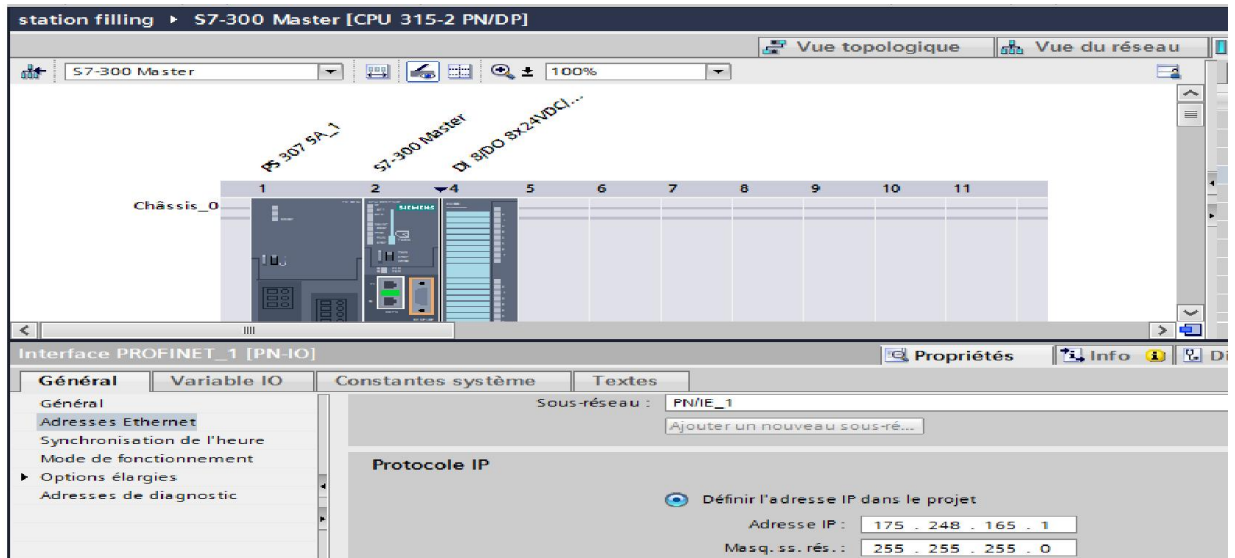


Figure III.6 : Adresse Ethernet de la CPU.

### III.2.8. Création du tableau des variables API :

Le tableau des variables API nous permet de définir la liste des variables qui seront utilisées lors de la programmation.

Il nous permet de désigner l'ensemble des entrées et sorties de notre système, les fins de course, les retours de marche et d'arrêt, les différents transmetteurs de température, de pression, et de niveau de remplissage, les défauts, les mémoires, l'ouverture et la fermeture des vannes....Etc., la figure ci-dessous représente notre tableau de variable API.

	Nom	Table des variables	Type de données	Adresse	Rémar...	Visibl...	Acces...	Commentaire
1	GRAPH_Group_Fault	Tags GRAPH Se...	Bool	%M10.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	GRAPH_Count_Bottle	Tags GRAPH Seque...	Int	%MW12		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Valve_water	Tags GRAPH Seque...	Bool	%Q1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Valve_AJC	Tags GRAPH Seque...	Bool	%Q1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Valve_OJC	Tags GRAPH Seque...	Bool	%Q1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	GRAPH_Mixer_ON	Tags GRAPH Seque...	Bool	%Q1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Conveyor_Direction	Tags Conveyor	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Conveyor_Start_Conveyor	Tags Conveyor	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Valve_Fill_Bottle	Tags Filling	Bool	%Q1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	GRAPH_Start_Labeling	Tags GRAPH Seque...	Bool	%Q2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	GRAPH_Filling_Complete	Tags GRAPH Seque...	Bool	%M20.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Start_GRAPH_Sequence	Tags GRAPH Seque...	Bool	%M100.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Conveyor_Drive_OK	Tags Conveyor	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Conveyor_Start_Forward	Tags Conveyor	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Conveyor_Start_Backward	Tags Conveyor	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Best_before_date	Tags Best before date	Int	%MW2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	<Ajouter>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure III.7 : Tableau de variable API.

### III.2.9.Types de variable utilisées en STEP7 :

Il existe plusieurs types de variables distinctes dans le step7, ces variables sont déclarées préalablement avant chaque début de programme. Le tableau suivant réunit ces différentes variables :

TYPE	TAILLE
Bool	1 bit
Real	32 bit
Int	16 bit
Byte	8 bit
Word	16 bit
DWord	32 bit
DInt	32 bit
Char	8 bit
Time	32 bit
S5Time	16 bit
Date	16 bit
Time of Day	32 bit

Tableau III.1 : Types de variables utilisées en step7 [23].

### III.2.10. Ecriture du programme :

On écrit le programme en utilisant les différents blocs :

- **Bloc d'organisation OB** : il commande le traitement du programme. Il est possible par l'intermédiaire des OB de réagir aux événements cycliques, temporisés ou déclenchés par alarme durant l'exécution du programme. Le programme d'OB sera un appel aux différentes fonctions (bloc Call).
- **Bloc fonctionnel FB** : C'est un bloc de code qui sauvegarde en permanence sa valeur dans un bloc de donnée d'instance qu'il soit possible d'y accéder même après le traitement du bloc.
- **FB1** : Contient les GRAFCET du procès existant (Avant de faire l'automatisation).
- **FB2** : Contient les GRAFCET après la migration.
- **Fonction FC** : les fonctions sont des blocs sans mémoires.
- **Bloc de données DB** : il sert à sauvegarder les données du programme. On ouvre « Blocs de programme » et on clique deux fois sur « Ajouter nouveau bloc », puis « Bloc fonctionnel », « Fonction » ou « Bloc de données » à savoir notre cahier de charge.

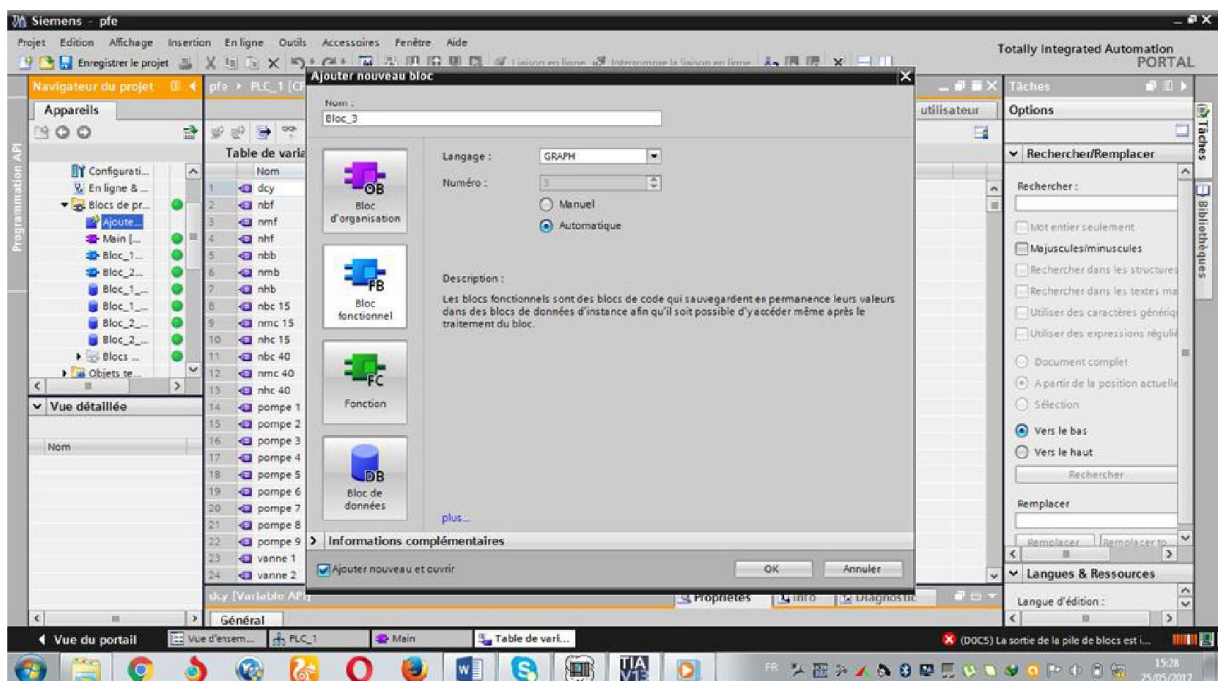


Figure III.8 : Création des blocs et des fonctions.

### III.3.Ajouter un GRAFCET :

Nous arrivons enfin à la programmation du Grafcet. Pour ajouter un Grafcet dans le projet, il faut ajouter un bloc fonctionnel au programme de l'automate en choisissant le langage GRAPH avant ajouter le bloc. Pour créer le Grafcet, il suffit de glisser les éléments désirés dans la fenêtre du Grafcet, des carrés apparaissant à l'endroit où le composant sélectionné pourra être déposé.

### III.4. Structure graphique du Grafcet :

Le Grafcet est un graphe cyclique composé alternativement de transition et d'étape, reliée entre elles par des liaisons orientées. Des actions peuvent être associées aux différentes étapes, la figure ci-dessous représente un Grafcet.

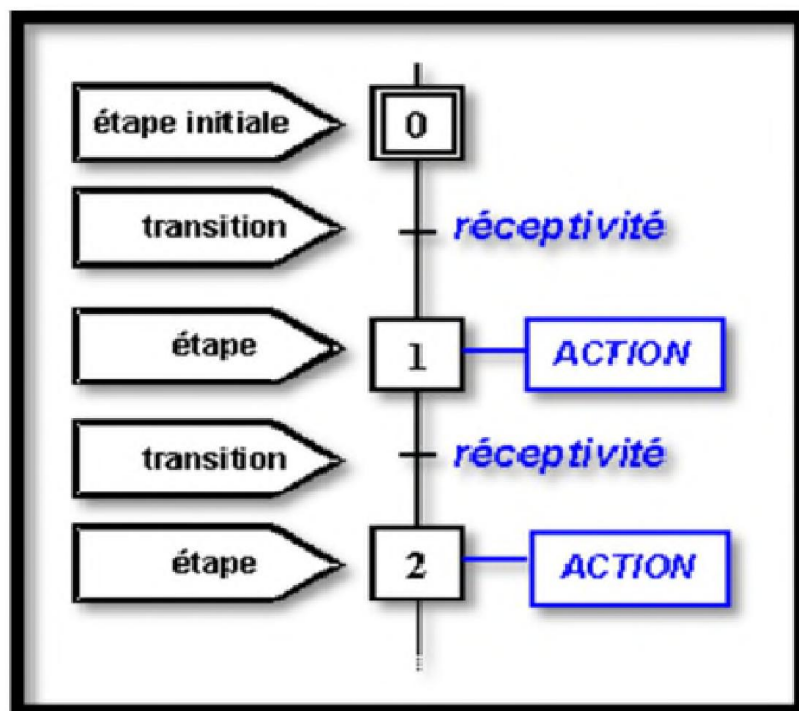


Figure III.9 : Représentation d'un Grafcet [24].

#### III.4.1.Etape :

Les tâches d'un graphe séquentiel sont divisées en différentes étapes. Dans les étapes, formulez des instructions exécutées par la CPU dans certaines conditions définies. Lors du déroulement du programme, chaque étape est traitée individuellement dans l'ordre [28].

Le graphique ci-dessous montre la représentation graphique d'une étape :



Figure III.10 : Etape [28].

### III.4.2. Transition :

Les transitions contiennent les conditions pour passer d'une étape à l'autre dans le graphe séquentiel. Une transition est valable lorsque toutes les conditions qu'elle contient sont remplies. Si les conditions d'une transition sont satisfaites, elle passe à l'étape suivante.

L'étape faisant partie de la transition ou les étapes correspondantes sont alors désactivées et l'étape suivante est activée [28].

Le graphique ci-dessous montre la représentation graphique d'une transition :



Figure III.11 : Transition [28].

### III.4.3. Branche OU :

Si plusieurs transitions suivent une même étape, il est possible d'y insérer une branche OU.

Une branche OU est constituée de plusieurs branches parallèles qui commencent chacune par une transition. Si plusieurs transitions sont satisfaites en même temps au début de branches différentes, c'est la transition qui se trouve le plus à gauche qui a la plus grande priorité [28].

Le graphique ci-dessous montre la représentation graphique d'une branche OU :

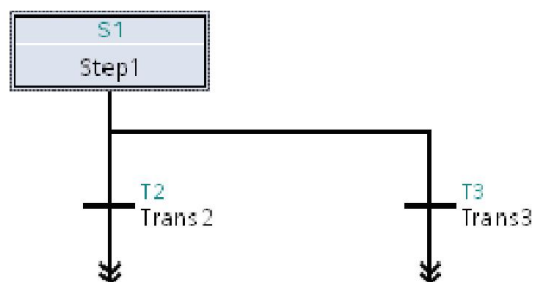


Figure III.12 : Branche OU [28] .

### III.4.4. Saut :

Un saut est la passerelle d'une transition vers une étape quelconque dans le graphe séquentiel et permet non seulement de ré exécuter mais aussi de traiter à nouveau des parties du graphe séquentiel du FB GRAPH. Le saut et la destination du saut sont toujours représentés sous forme de flèches [28].

Le graphique ci-dessous montre la représentation graphique d'un saut :

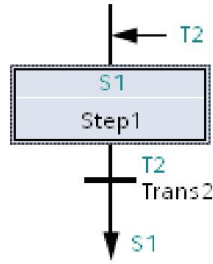


Figure III.13 : Saut [28].

### III.5. Simulation du programme sous step7 avec S7-PLCSIM :

#### III.5.1 Présentation de S7-PLCSIM :

S7-PLCSIM est une application qui permet d'exécuter et de tester le programme utilisateur élaboré dans un automate programmable (AP), la simulation s'exécute dans un ordinateur ou dans une console de programmation elle est complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7quelconque (CPU ou module de signaux).

S7-PLCSIM fournit une interface simple avec le programme utilisateur STEP 7 pour visualiser et forcer différents objets comme des entrées et des sorties. Tout en exécutant le programme dans l'AP de simulation, il procure également la possibilité de mettre en œuvre les diverses applications du logiciel STEP 7 comme, par exemple, la table des variables (VAT) afin d'y visualiser et d'y forcer des variables.

#### III.5.2 Caractéristiques de S7-PLCSIM :

Le S7-PLCSIM nous permet d'exécuter les tâches suivantes : [25]

L'activation ou la désactivation de la simulation dans le gestionnaire de projets SIMATIC ce fait en cliquant sur le bouton de simulation. Au démarrage de S7-PLCSIM, on peut ouvrir une simulation existante ou sélectionner un nœud dans un projet STEP 7 pour l'accès à la CPU. Lorsque S7-PLCSIM s'exécute, toute nouvelle liaison depuis STEP 7 est automatiquement dirigée vers la CPU de simulation.

S7-PLCSIM permet d'exécuter des programmes destinés aux CPU S7-300 ou S7-400, y compris la CPU 317-T sur l'AP de simulation.

La création des fenêtres dans lesquelles il y a la possibilité d'accéder aux zones de mémoire d'entrée et de sortie, aux accumulateurs ainsi qu'aux registres de la CPU de simulation. On peut également accéder à la mémoire par l'adressage symbolique.

Il fournit la possibilité d'exécuter des temporisations automatiquement ou les définir et les réinitialiser manuellement. La réinitialisation peut porter sur des temporisations individuelles ou sur toutes les temporisations à la fois.

- Il offre la possibilité de changer la position du commutateur de mode de fonctionnement de la CPU (RUN, RUN-P et STOP). De plus, la simulation de modules met à la disposition de l'utilisateur une fonction de pause qui permet d'interrompre momentanément la CPU, sans affecter l'état du Programme.
- Il permet aussi l'enregistrement d'une série d'évènements (manipulation des zones de mémoire d'entrée et de sortie, mémentos, temporisations et compteurs) et lire votre enregistrement afin d'automatiser les tests de programme.

### **III.6.SIMATIC Win CC Runtime Advanced (Portail TIA) :**

Solution HMI sur PC pour les systèmes mono-utilisateur directement sur la machine. La gamme de fonctions de WinCC Runtime Advanced inclut les composants IHM centralisés pour la visualisation et la génération de rapports, et peut être étendue aux besoins et aux coûts et Peut être intégré dans des solutions d'automatisation basées sur des réseaux TCP / IP Concepts étendus de service avec l'opération, le diagnostic et l'administration à distance sur l'intranet et l'Internet en combinaison avec la communication d'email [26].

#### **III. 6.1.Définition de l'interface homme-machine (IHM) :**

Le système d'interface homme-machine (IHM) constitue l'interface entre l'opérateur et le processus. Le déroulement du processus est commandé par la CPU. L'opérateur peut visualiser le processus ou intervenir dans le processus en cours par le biais d'un pupitre opérateur [27].

#### **III.6.2.Avantages :**

- ✓ Interface de configuration innovante basée sur les dernières technologies logicielles ;
- ✓ Concept de bibliothèque complet pour objets et façades définissables par l'utilisateur ;
- ✓ Des outils intelligents pour la configuration graphique et la gestion des données de masse.

### III.6.3. Configurer une vue IHM :

Sur la page d'accueil de TIA Portal, on choisit la visualisation en cliquant sur Configurer une vue IHM.

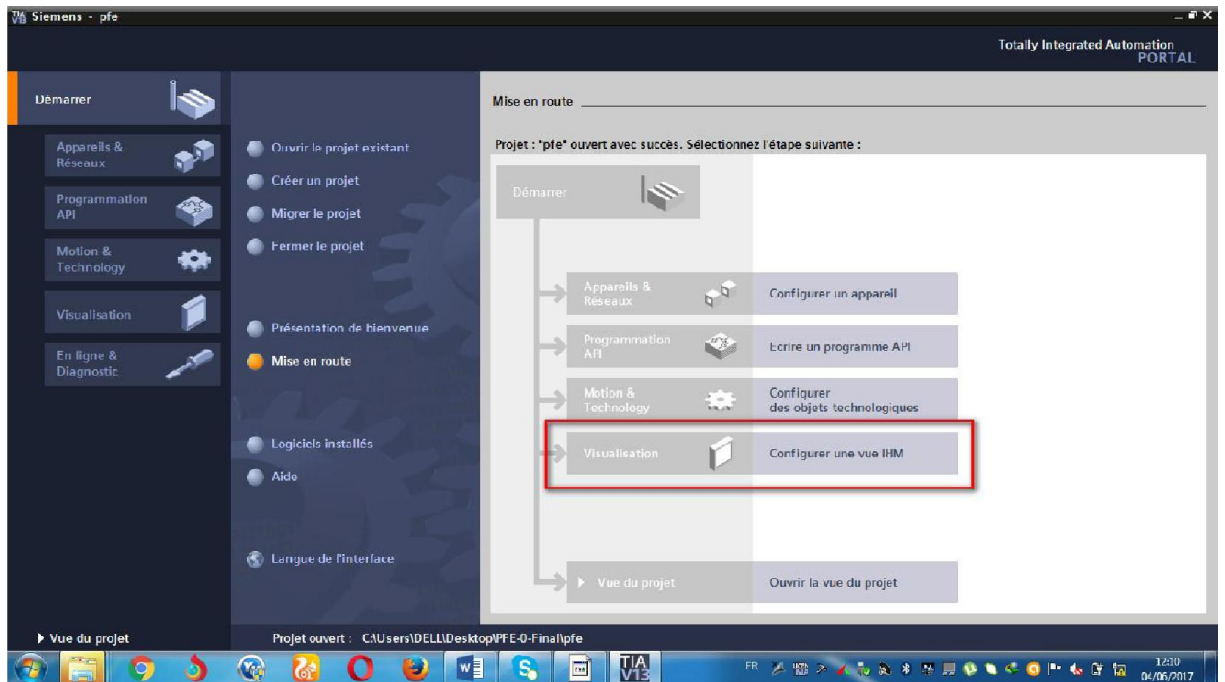


Figure III.14 : Configurer une vue IHM

On a choisi le pupitre le HMI Panel "TP900 Comfort".

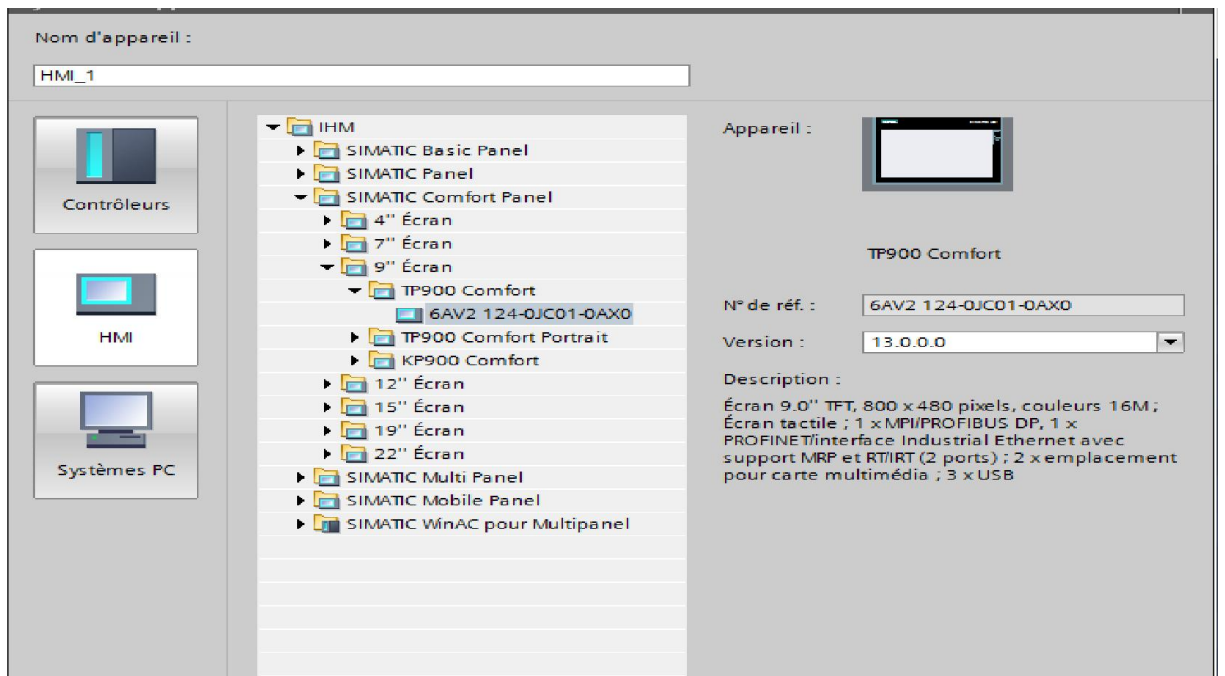


Figure III.15 : Choisir le pupitre

S notre projet on a configuré 3 vues :

1- 1ere vue : Recettes.

2- 2eme vue : Vue du diagnostic.

3- 3eme vue : Vue système.

La figure suivante représente la vue de la station de remplissage.

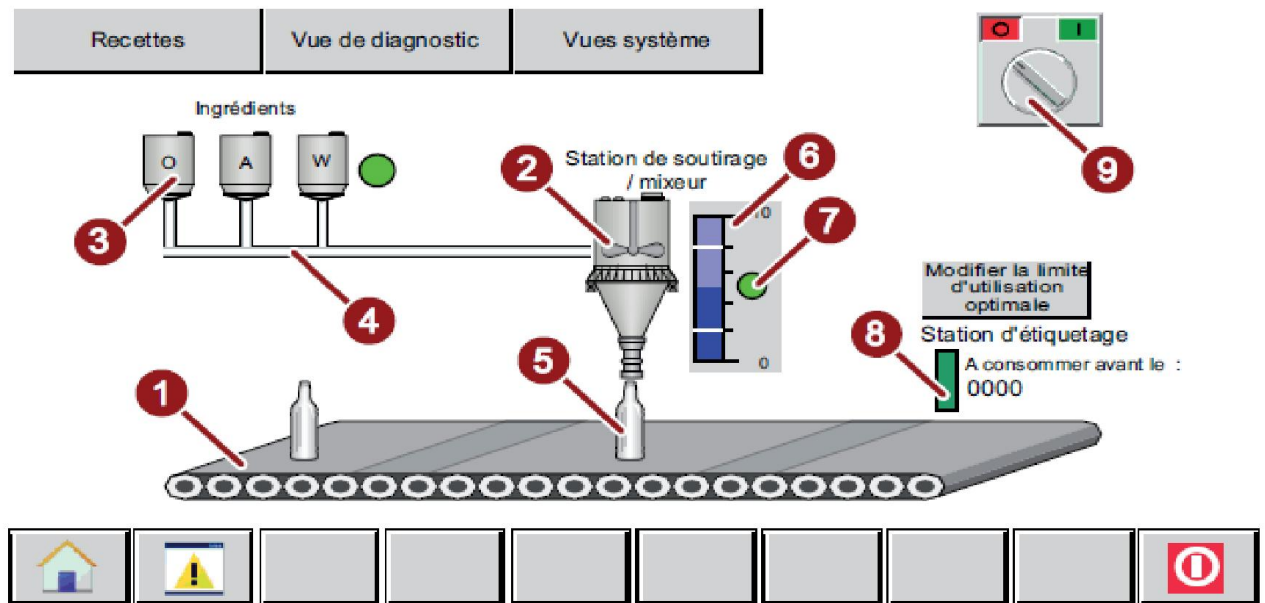


Figure III.16 : des éléments de la vue racine

### III. 6.4.Eléments de la vue racine :

1	Tapis roulant
2	Installation de remplissage avec mixeur
3	Cuves à boisson
4	Conduites
5	Bouteilles sur le tapis roulant
6	Affichage sous forme de diagramme à barres
7	Témoin de contrôle lumineux
8	Machine à étiqueter
9	Commutateur pour activer le graphe séquentiel

Tableau III.2 : Tableau d'éléments de la vue racine.

### III.7.Simulation de la station :

#### III.7.1. Les Blocs utilisateurs :

Nous devons construire nos blocs selon leur séquence selon notre système. Le format ci-dessous est la représentation de la fenêtre d'extension Nouveau bloc.

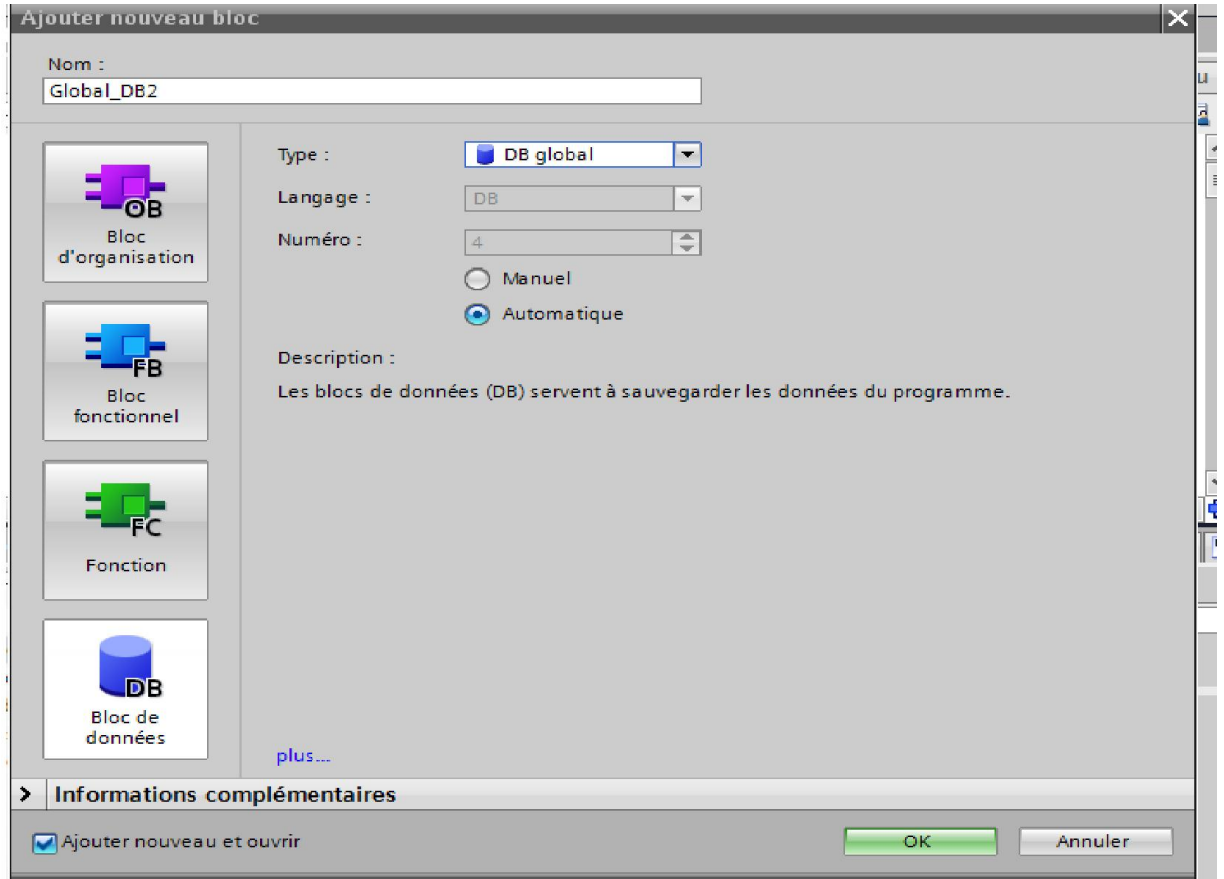


Figure III.17 : Fenêtre d'ajout de bloc.

##### III.7.1.1. Bloc d'organisation (OB) :

Contrôle le traitement du programme, les OB peuvent réagir aux événements périodiques ou à leur minutage ou être déclenchés par des alarmes pendant mise en œuvre du programme.

Les formes suivantes illustrent le bloc d'organisation (OB):

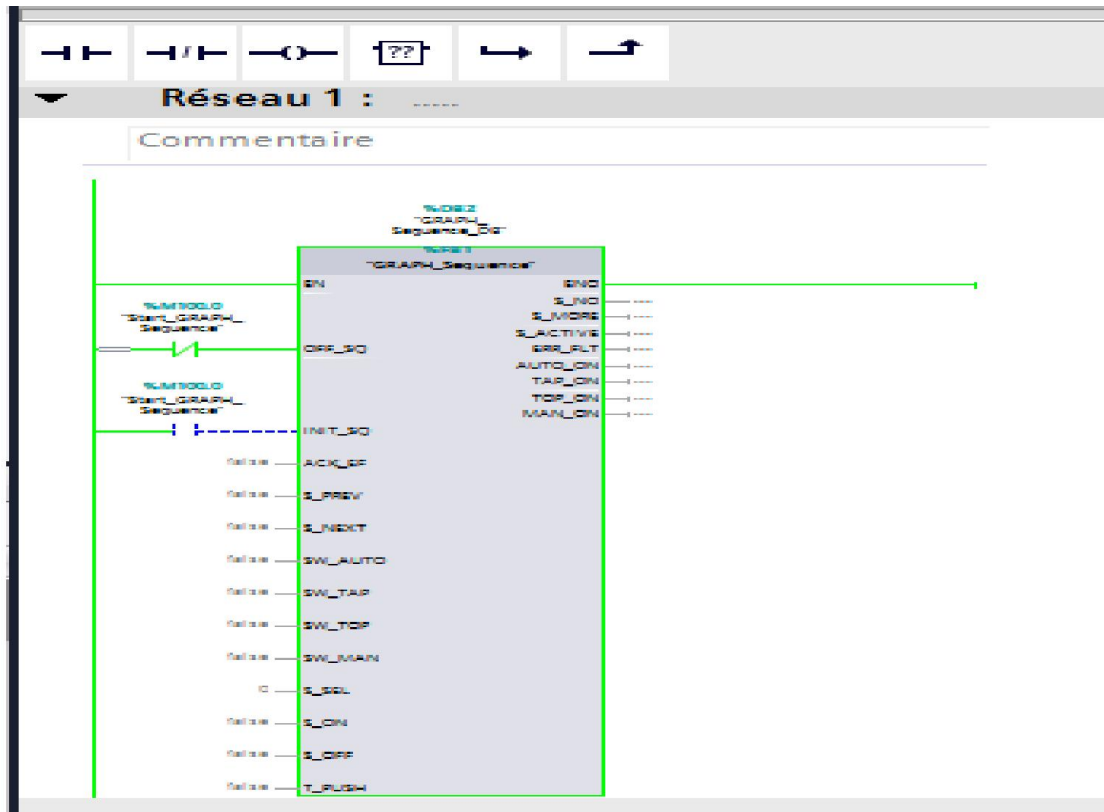


Figure III.18:Réseau 1dans l’OB 1.

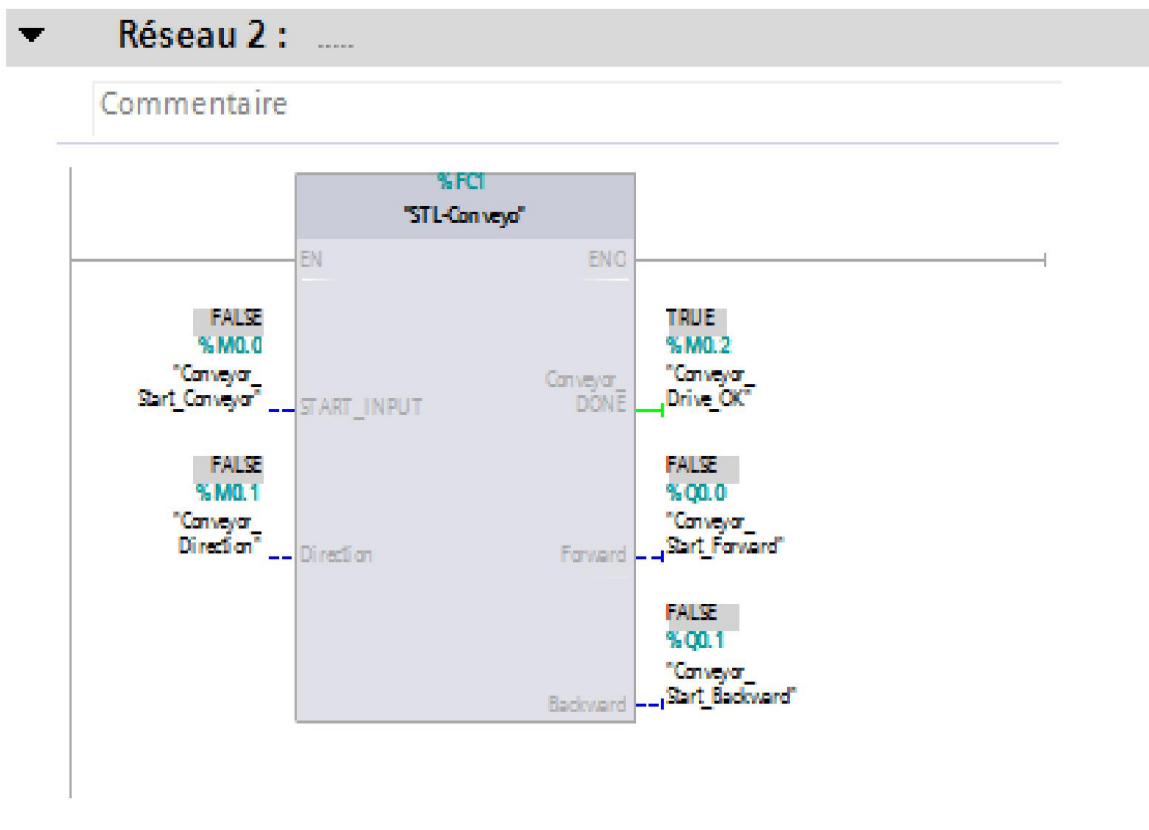


Figure III.19 : Réseau 2dans l’OB 1.

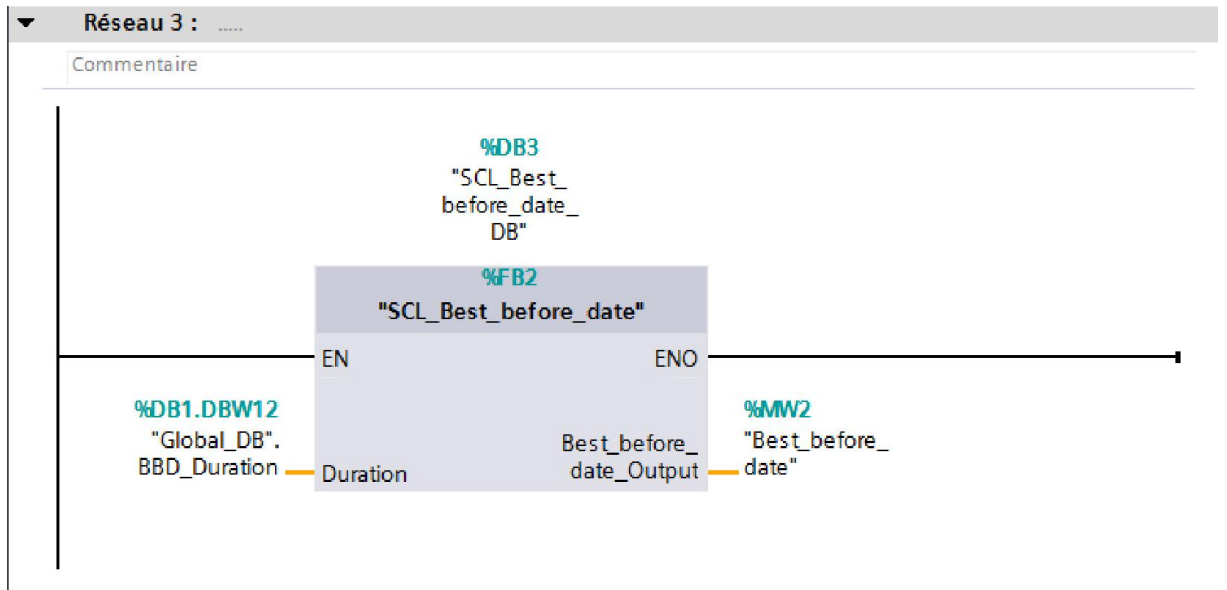


Figure III.20 : Réseau 3 dans l’OB 1.

**III.7.1.2. FB (Bloc de fonction) :**

Le FB est à disposition via un espace mémoire correspondant. Si un FB est appelé, il lui est attribué un bloc de données (DB). On peut accéder aux données de cette instance DB par des appels depuis le FB.

La figure suivante représente un Bloc de fonction [FB1]:

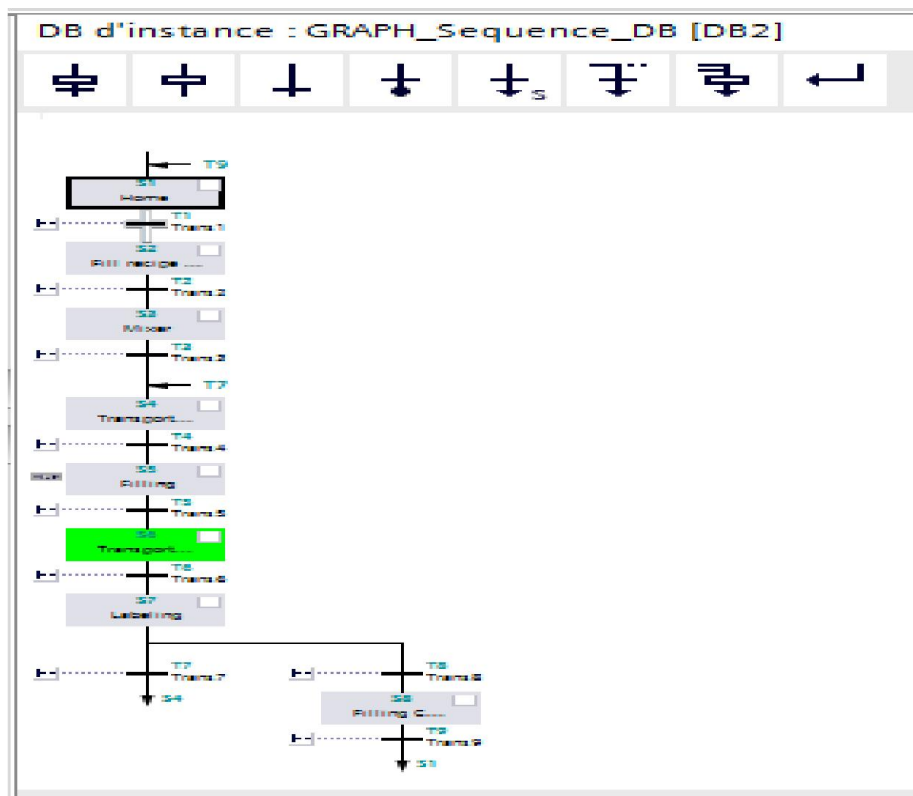


Figure III.21 : Bloc fonction [FB1]

### III.7.1.3. Blocs de données (DB) :

Le bloc de données DB contient les paramètres effectifs et les données statique de notre programme.

La figure suivante représente un bloc de données DB1 :

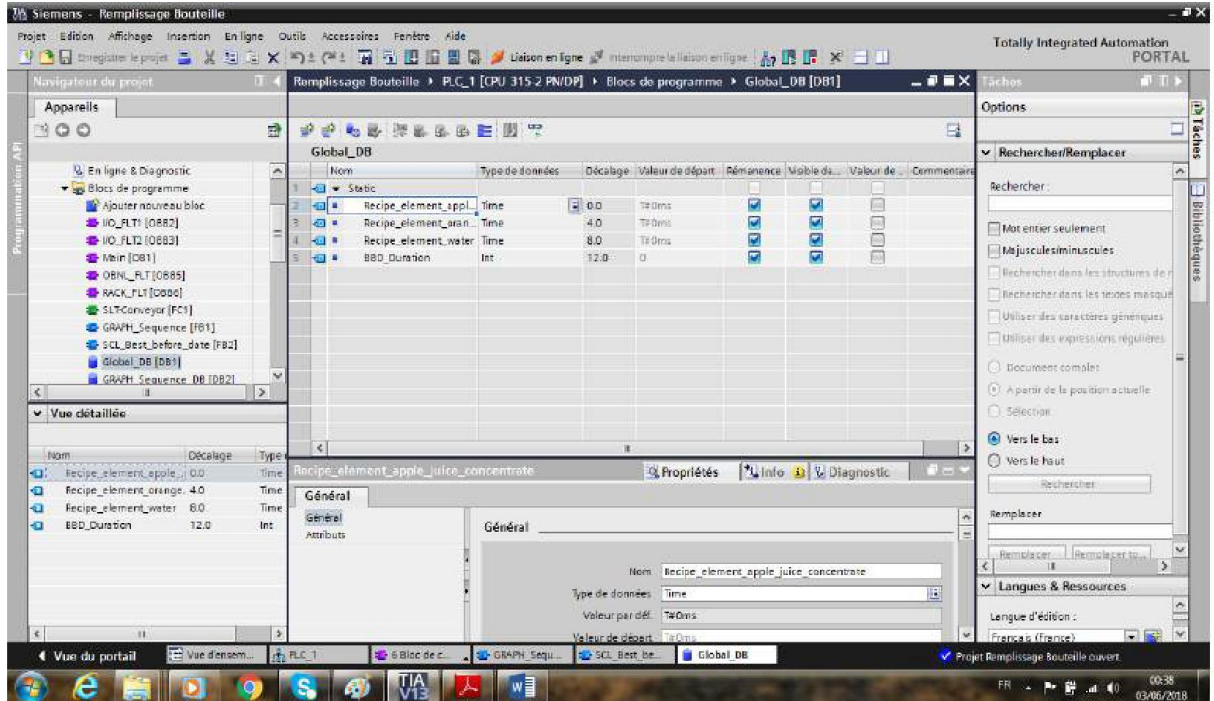


Figure III.22 : Bloc de donnée DB1

### III.8. Réalisation de la supervision :

La supervision consiste à suivre l'évolution et la commande du système automatisé grâce à un Ordinateur. Pour cela, un programme de supervision réalisé avec le logiciel Win CC est réalisé sur PC qui prend en charge la vue générale du procédé. Cette transparence s'obtient au moyen de l'interface Home Machine (IHM). Le contrôle proprement dit du processus est assuré par le système d'automatisation.

Une fois le pupitre mis sous réseau, il permet :

- De visualiser l'état du remplissage et régulation des bouteilles, ainsi que les différents capteurs : de température, pression, niveau...etc.
- D'afficher les alarmes ;
- D'agir sur le système.

#### III.8.1. Etablir une liaison directe :

La première chose à effectuer est de créer une liaison directe entre HMI et le S7-300, et ce dans le but que le HMI puisse lire les données se trouvant dans la mémoire de l'automate.

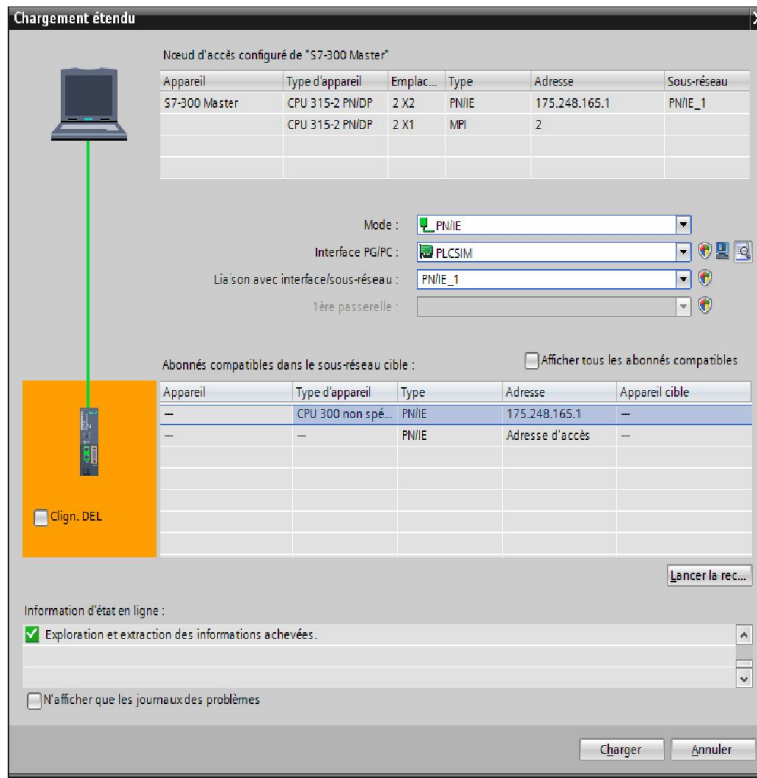


Figure III.23 : Liaison entre la PLC et IHM.

**III.8.2.Création de la table des variables :**

Maintenant que la liaison entre notre projet IHM et l’automate S7-300 est établie, il nous est possible d’accéder à toutes les zones mémoires de l’automate qui peuvent être des mémoires : entrée/sortie, memento, bloc de données.

La figure ci-dessous est une représentation de la table de variable IHM.

Table de variables standard						
	Nom	Type de données	Connexion	Nom API	Variable API	Adresse
	Best_before_date	Int	HMI_Liaison...	S7-300 Master	Best_before_date	%MW2
	Global_DB_BBD_Duration	Int	HMI_Liaison_1	S7-300 Master	Global_DB.BBD_Duration	%DB1.DBW12
	Global_DB_Recipe_element_ap...	Time	HMI_Liaison_1	S7-300 Master	Global_DB.Recipe_eleme...	%DB1.DBDO
	Global_DB_Recipe_element_or...	Time	HMI_Liaison_1	S7-300 Master	Global_DB.Recipe_eleme...	%DB1.DBDO4
	Global_DB_Recipe_element_wa...	Time	HMI_Liaison_1	S7-300 Master	Global_DB.Recipe_eleme...	%DB1.DBDO8
	GRAPH_Count_Bottle	Int	HMI_Liaison_1	S7-300 Master	GRAPH_Count_Bottle	%MW12
	GRAPH_Sequence_DB_Fill recip...	Bool	HMI_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.*Fil...	%DB2.DBX251.0
	GRAPH_Sequence_DB_Filling_X	Bool	HMI_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.Fill...	%DB2.DBX347.0
	GRAPH_Sequence_DB_Labeling...	Bool	HMI_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.La...	%DB2.DBX411.0
	GRAPH_Sequence_DB_Mixer_X	Bool	HMI_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.Mi...	%DB2.DBX283.0
	GRAPH_Sequence_DB_Transpo...	Bool	HMI_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.*Tr...	%DB2.DBX315.0
	GRAPH_Sequence_DB_Transpo...	Bool	HMI_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.*Tr...	%DB2.DBX379.0
	Numéro_vue_variable	UInt	<Variable intern...		<indéfini>	
	Position_Bottle	Int	<Variable intern...		<indéfini>	
	Position_Bottle(1)	Int	<Variable intern...		<indéfini>	
	Start_GRAPH_Sequence	Bool	HMI_Liaison_1	S7-300 Master	Start_GRAPH_Sequence	%M100.0
	<ajouter>					

Figure III.24 : Table de variable IHM.

### III.8.3.Création de vue :

L'interface TIA PORTAL V13 nous permet de créer des vues dans le but de contrôler et de commander notre installation. Lors de la création des vues, on dispose d'objets prédéfinis permettant d'afficher des procédures et de définir des valeurs de processus.

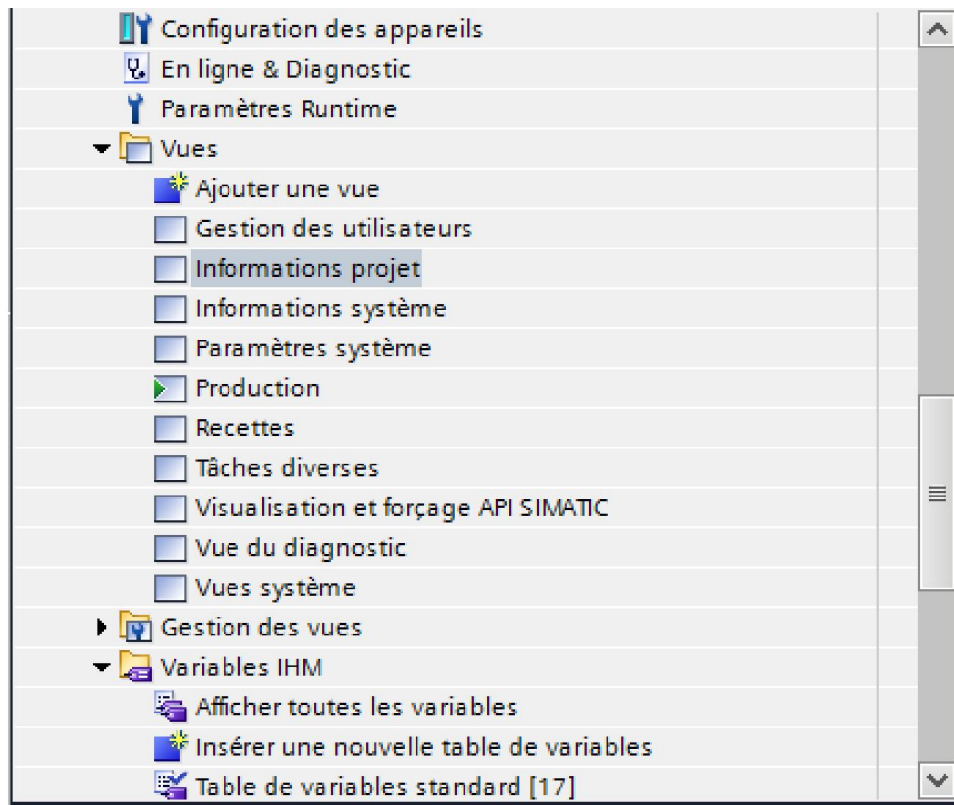


Figure III.25 : Création de vue.

### III.8.4.Vue générale :

La vue ci-dessous nous permet de :

- Visualiser l'ensemble des équipements de la station ;
- Visualiser la progression du processeur de mixage et remplissage les bouteilles en temps réel, ainsi qu'évaluez les différents capteurs configurés à l'aide du champ d'E / S.
- Gérer la mise en marche/arrêt de la station grâce aux boutons « marche machine » et « arrêt machine ».
- Visualiser l'état de chaque élément de la station.

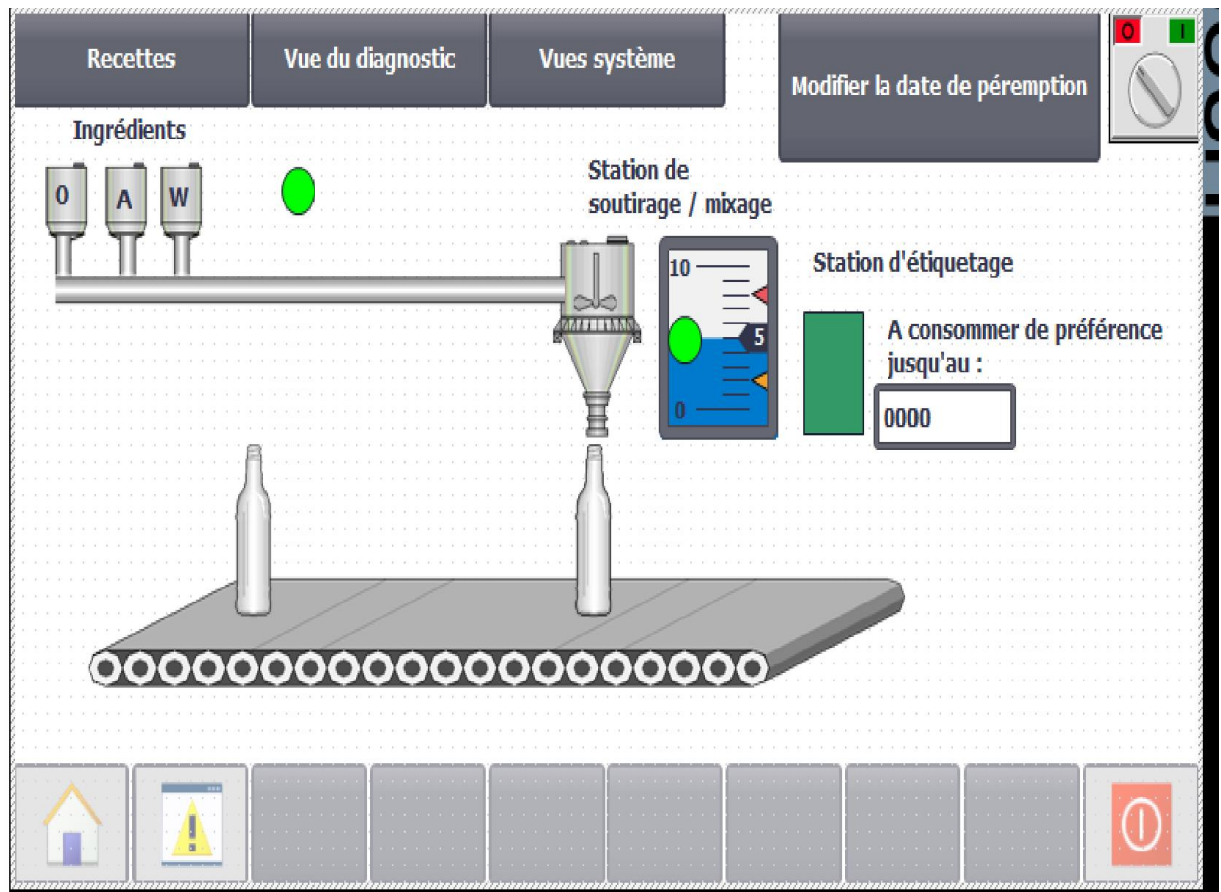


Figure III.26 : Vue générale.

### III.8.5.Compilation et Simulation :

Après avoir créé le projet et terminé la configuration, il est indispensable de vérifier la cohérence du projet, et de détecter les erreurs.

Pour simuler notre système en entier plusieurs étapes sont à effectuer, dans notre PLC on clique sur le bouton droit puis on choisit compiler.

On clique une fois sur « matériel et logiciel (uniquement modifications) », la figure ci-dessous représente l'étape de compilation PLC.

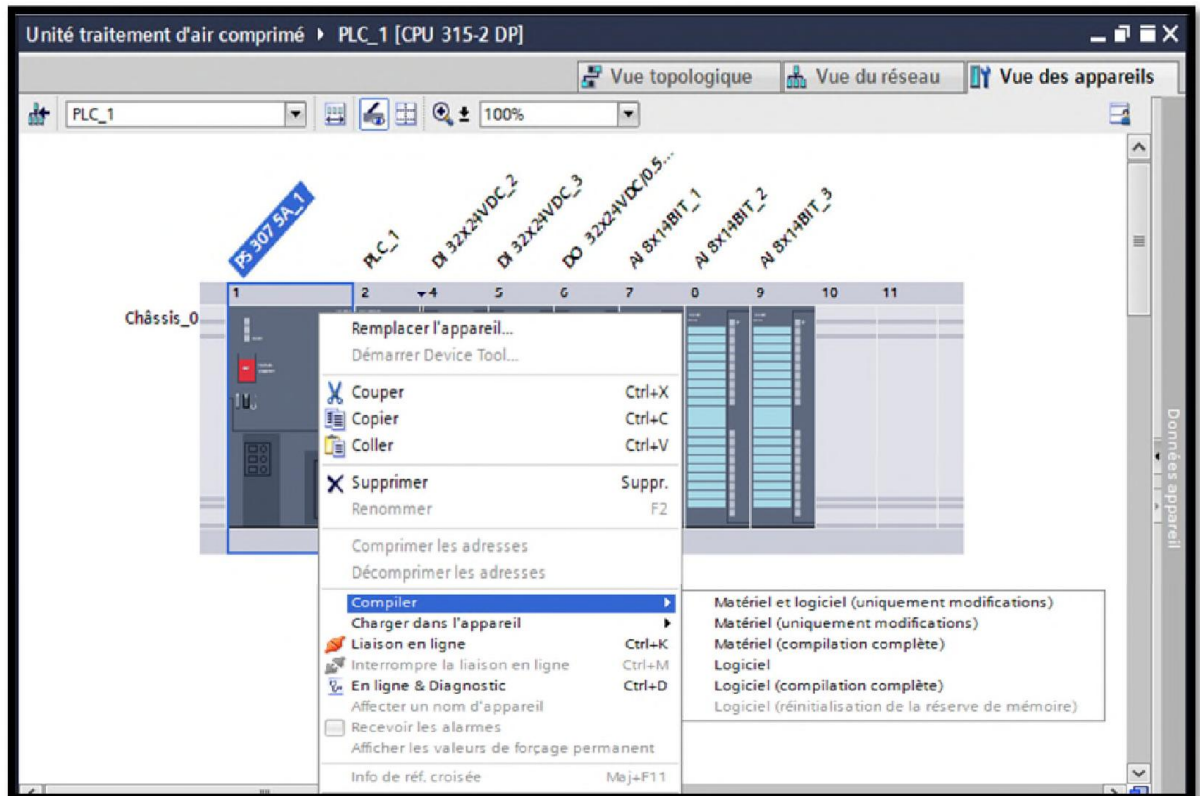


Figure III.27 : Etape de compilation PLC.

Après création du programme, on le charge dans l'automate, et comme il ne s'agit que d'une simulation (donc en l'absence de l'automate), on utilisera le logiciel S7-PLCSIM.

Le simulateur est lancé à partir de TIA PORTAL V13, son lancement nous permet de charger le programme.

Après le chargement des programmes dans le simulateur, on met la CPU en mode RUN-P (exécution du programme) voir la Figure III.23.

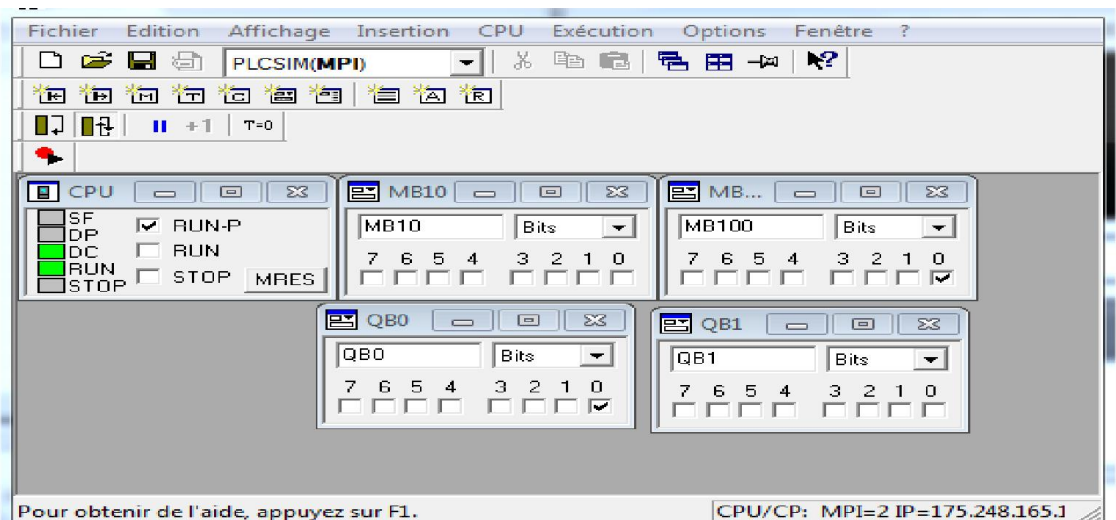


Figure III.28 : Simulateur S7-PLCSIM

### III.8.6.Simulation du programme sous WinCC :

La fenêtre principale représentant les différentes opérations à choisir.

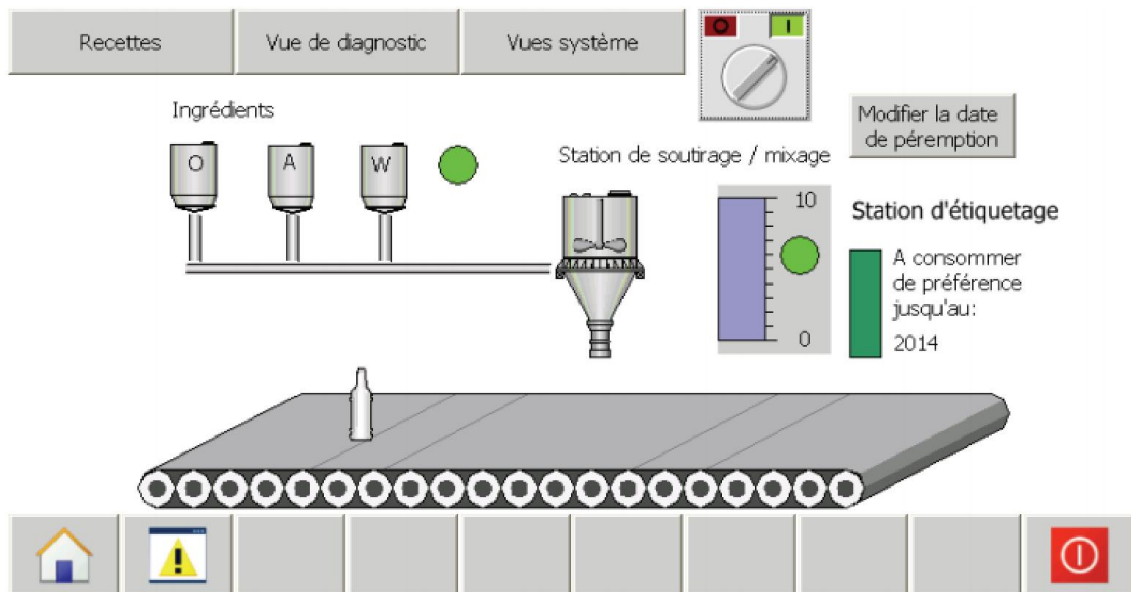


Figure III.29 :la vue production.

#### Vue du système :

En cliquant sur le bouton "vue de système", vous verrez la fenêtre ci-dessous, qui représente différentes propriétés du système.

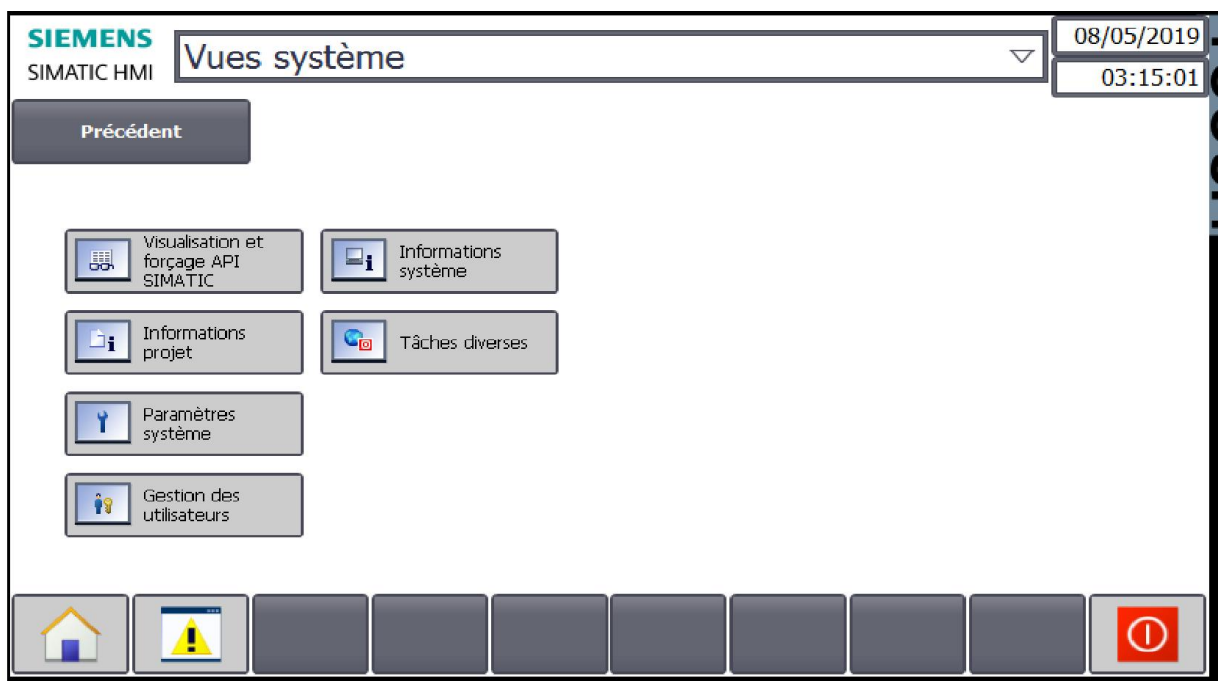


Figure III.30 : Vue du système

**III.9.Conclusion :**

Dans ce chapitre, nous avons fourni un aperçu de la procédure utilisée pour créer notre programme TIA PORTAL V13. Les blocs SIM STEP 7, SIMATIC WINCC et PLC SIM sont fournis et seront implémentés dans notre console S7-300.

La partie IHM nous permettra de contrôler notre unité et d'acquérir une bonne connaissance des performances de notre système et du langage dans lequel l'API est programmée pour s'y connecter.

# **Conclusion Générale**

# Conclusion générale

---

## Conclusion générale

Dans ce modeste travail, nous avons étudiés "L'Automatisation d'une station de remplissage", au sein de l'entreprise nous a permis d'identifier une approche de résolution des problèmes, en particulier dans un projet complexe tel que la création d'une unité industrielle, enrichissant ainsi notre connaissance de l'automatisation industrielle et tirant parti de l'expérience acquise dans ce domaine.

Le but de ce travail était d'installer le logiciel TIA Portal dans des automates programmables industriels afin de contrôler le niveau de mise en bouteille et la station dans son ensemble, ainsi que l'écran de contrôle sur l'ordinateur contrôlé par le S7-300, ce qui aiderait l'opérateur à surveiller les conditions de fonctionnement et à réagir aux défaillances. Effectuez des simulations pour voir la réponse du contrôle terminal et les alarmes potentielles avant de démarrer, en prenant en compte les défauts qui endommageront notre système. Le langage de programmation utilisé est Step7. Cela nous a permis d'exporter des entrées / sorties directement vers la table des symboles pour les utiliser comme variables externes dans le programme SIMATIC WinCC Runtime Advanced.

TIA PORTAL V13 est le dernier logiciel d'ingénierie développé par Siemens. L'IHM nous permettra de mieux contrôler le processus et de diagnostiquer rapidement tout effondrement possible.

Le parcours professionnel nous a permis de mieux comprendre la portée du projet et de nous familiariser avec les responsabilités des ingénieurs de terrain dans ce domaine.

Nous espérons que ce travail humble servira de point de départ à notre vie professionnelle et qu'il sera utile pour les promotions futures, si Dieu le veut.

# **Bibliographie**

## Bibliographie

---

### Bibliographie

- [1] SEKHSOKH. S, OUKILI. K « ÉTUDE D'UNE BOUCLE DE RÉGULATION DE NIVEAU : IMPLÉMENTATION DU RÉGULATEUR ET RÉGLAGE DU PROCÈDE » Projet de fin d'étude, 2010/2011.
- [2] ANDRE .S, Automates programmable, programmation, automatisation, et logique programmée, Edition L'ELAN, 1983.
- [3] “*Sommaire des cours*”. Siemens, centre de formation industrie, édition 2000.
- [4] BERGOUGNOUX.L., “*API automate programmable industriel*”. Poly Tech Marseille,  
Édition 2004-2005.
- [5] GRARE.P & KACEM.I., “*AUTOMATISME Ce qu'il faut s'avoir sur les automatismes*». Edition ellipses, 2008.
- [6] MICHEL. G, Les API, Architecture Et Application Des Automates Programmables. Industriels. Dunod, Paris,[1987].
- [7] Documentations techniques internes. “*Manuel d'instructions et d'entretien*”. CAA.
- [8] SIMATIC « Mise en route STEP7 » Édition 03 /2006.
- [9] «SIMATIC WinCC flexible», Brochure, Mars 2010.
- [10] GILLES.M, « MACHINES-REMP LISSEUSE-DOSEUSE-EAU-PLATE » ;
- [11] [www.ocme.com](http://www.ocme.com);
- [12] JEAN HENG, « *PRATIQUE DE LA MAINTENANCE PREVENTIVE* », Livre, Paris :  
L'usine nouvelle, 2011.
- [13] <http://www.techno-science.net>;
- [14]MICHEL.G« *INSTRUMENTATION INDUSTRIELLE, SPECIFICATION ET INSTALLATION DES CAPTEURS ET VANNES DE REGULATIONS* », Livre, Paris :  
Dunod, 2012.
- [15] GEORGES. A, «*LES CAPTEURS EN INSTRUMENTATION INDUSTRIELLE* », Livre, Paris : Dunod, 2010.
- [16] IKHLEF. M, LOUIBAY. Y « *AUTOMATISATION ET REGULATION D'UNE POMPE PAR UN DEBITMETRE* », Mémoire de fin d'études Master Automatique, Département Electronique, Université de Bejaia, 2016.

## Bibliographie

---

- [17] « COURS AUTOMATIQUE » École polytechnique fédérale de Lausanne, 2008/2009.
- [18] PATRICK PROUVOST « AUTOMATIQUE CONTROLE ET REGULATION COURS ET EXERCICES CORRIGES » DUNOD (2004).
- [19] ANDRE. S « AUTOMATES PROGRAMMABLE INDUSTRIELS NIVEAU 1 » Edition l'elan-liege 1991.
- [20] ALAIN GONZAGA « Les automates Programmables Industriels ». PDF
- [21] Aide Du Logiciel TIA PORTAL.
- [22] HATRAF LAHCENE MOHAMED AMINE.TOUZALA ABDERRAHMANE « automatisation et supervision d'une station de remplissage des bouteilles »
- [23] SIEMENS, « S7-1200\_System\_Manual », Numéro de référence du document : A5E02486682-AG 03/2014 ;
- [24] JARGOT. P, Langage De Programmation, Langages De Programmmations Pour API.
- [25] SIEMENS, « Simatic STEP7 V5.1 Getting started », édition 08/2000
- [26] [www.SIEMENS.com](http://www.SIEMENS.com)
- [27] [www.Wikipedia.fr](http://www.Wikipedia.fr)
- [28] STEP 7 Professional /WinCC Advanced V11 pour l'exemple de projet "Station de remplissage"

# **Annexe**

## ANNEXE

### A- Table Des mnémoniques :

Les fonctions dans le bloc de données.

The screenshot shows the 'Global\_DB' data block configuration in the software. The main table lists the following data points:

Nom	Type de données	Décalage	Valeur de départ	Rémanence	Visible da...	Valeur de ...	Commentaire
Static							
Recipe_element_appl...	Time	0.0	T#0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Recipe_element_oran...	Time	4.0	T#0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Recipe_element_water	Time	8.0	T#0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
BBD_Duration	Int	12.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

The left sidebar shows a project tree with 'Global\_DB [DB1]' selected. The bottom status bar shows 'Recipe\_element\_Apple\_juice\_concentrate'.

The screenshot shows the 'GRAPH\_Sequence\_DB' data block configuration. The main table lists the following data points:

Nom	Type de don...	Décalage	Valeur d...	Rémanence	Visible da...	Valeur de ...	Commentaire
T_PUSH	Bool	4.2	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Validation de commutation pour transition en...
Output							
S_NO	Int	6.0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Numéro d'étape
S_MORE	Bool	8.0	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Il existe d'autres étapes qui peuvent être affich...
S_ACTIVE	Bool	8.1	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>		L'étape affichée dans le paramètre S_NO est a...
ERR_FLT	Bool	8.2	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Erreur groupée Interlock or Supervision
AUTO_ON	Bool	8.3	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Mode automatique activé
TAP_ON	Bool	8.4	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Mode semi-automatique/Etape avec transition
TOP_ON	Bool	8.5	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Mode semi-automatique/Ignorer la transition...
MAN_ON	Bool	8.6	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Mode manuel activé
InOut							
Static							
Trans1	GraphTransit...	42.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Trans2	GraphTransit...	58.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Trans3	GraphTransit...	74.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Trans4	GraphTransit...	90.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Trans5	GraphTransit...	106.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Trans6	GraphTransit...	122.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Trans7	GraphTransit...	138.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Trans8	GraphTransit...	154.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Trans9	GraphTransit...	170.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Home	GraphStep	218.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Fill recipe ingredi...	GraphStep	250.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
OFF_SQ		0.0					
INT_SQ		0.1					
ACK_EF		0.2					
S_PREV		0.3					
S_NEXT		0.4					
SW_AUTO		0.5					
SW_TAP		0.6					
SW_TOP		0.7					
SW_MAN		1.0					

The left sidebar shows a project tree with 'GRAPH\_Sequence\_DB [DB2]' selected. The bottom status bar shows 'OFF\_SQ'.

# ANNEXE

Totally Integrated Automation PORTAL

station filling → S7-300 Master [CPU 315-2 PN/DP] → Blocs de programme → GRAPH\_Sequence\_DB [DB2]

Programmation API

Appareils

- W0\_FLT2 [OB83]
- Main [OB1]
- OBNL\_FLT [OB85]
- RACK\_FLT [OB86]
- STL-Conveyo [FC1]
- GRAPH\_Sequence [FB1]
- SCL\_Best\_before\_date [FB2]
- Global\_DB [DB1]
- GRAPH\_Sequence\_DB [DB2]
- SCL\_Best\_before\_date\_DB [D...]
- Blocs système
- Objets technologiques
- Sources externes

Vue détaillée

Nom	Décalage	Ty
OFF_SQ	0.0	Bc
INT_SQ	0.1	Bc
ACK_LEF	0.2	Bc
S_PREV	0.3	Bc
S_NEXT	0.4	Bc
SW_AUTO	0.5	Bc
SW_TAP	0.6	Bc
SW_TOP	0.7	Bc
SW_MAN	1.0	Bc

GRAPH\_Sequence\_DB

Nom	Type de don...	Décalage	Valeur d...	Rémanence	Visible da...	Valeur de...	Commentaire
24	InOut						
25	Static						
26	Trans 1	GraphTransit...	42.0				
27	Trans 2	GraphTransit...	58.0				
28	Trans 3	GraphTransit...	74.0				
29	Trans 4	GraphTransit...	90.0				
30	Trans 5	GraphTransit...	106.0				
31	Trans 6	GraphTransit...	122.0				
32	Trans 7	GraphTransit...	138.0				
33	Trans 8	GraphTransit...	154.0				
34	Trans 9	GraphTransit...	170.0				
35	Home	GraphStep	218.0				
36	Fill recipe ingredi...	GraphStep	250.0				
37	Mixer	GraphStep	282.0				
38	Transport Filling	GraphStep	314.0				
39	Filling	GraphStep	346.0				
40	Transport Labeling	GraphStep	378.0				
41	Labeling	GraphStep	410.0				
42	Filling Complete	GraphStep	442.0				
43	S_DISPLAY	Int	554.0	0			Affichage interne du paramètre de sortie S_NO
44	S_SEL_OLD	Int	556.0	0			Valeur précédente dans S_SEL
45	S_DISPIDX	Byte	558.0	16#0			Indice de l'étape affichée dans S_NO
46	T_DISPIDX	Byte	559.0	16#0			Indice de la transition affichée dans T_NO

OFF\_SQ

Propriétés Info Diagnostic

Général

Options

Rechercher/R...

Rechercher :

Mot entier seu

Majuscules/mi

Rechercher da

Rechercher da

Utiliser des ca

Utiliser des ex

Document co

A partir de la

Sélection

Vers le bas

Vers le haut

Recher

Remplacer

Remplacer

Langues & R...

Langue d'édition

Totally Integrated Automation PORTAL

station filling → S7-300 Master [CPU 315-2 PN/DP] → Blocs de programme → GRAPH\_Sequence\_DB [DB2]

Programmation API

Appareils

- W0\_FLT2 [OB83]
- Main [OB1]
- OBNL\_FLT [OB85]
- RACK\_FLT [OB86]
- STL-Conveyo [FC1]
- GRAPH\_Sequence [FB1]
- SCL\_Best\_before\_date [FB2]
- Global\_DB [DB1]
- GRAPH\_Sequence\_DB [DB2]
- SCL\_Best\_before\_date\_DB [D...]
- Blocs système
- Objets technologiques
- Sources externes

Vue détaillée

Nom	Décalage	Ty
OFF_SQ	0.0	Bc
INT_SQ	0.1	Bc
ACK_LEF	0.2	Bc
S_PREV	0.3	Bc
S_NEXT	0.4	Bc
SW_AUTO	0.5	Bc
SW_TAP	0.6	Bc
SW_TOP	0.7	Bc
SW_MAN	1.0	Bc

GRAPH\_Sequence\_DB

Nom	Type de don...	Décalage	Valeur d...	Rémanence	Visible da...	Valeur de...	Commentaire
29	Trans 4	GraphTransit...	90.0				
30	Trans 5	GraphTransit...	106.0				
31	Trans 6	GraphTransit...	122.0				
32	Trans 7	GraphTransit...	138.0				
33	Trans 8	GraphTransit...	154.0				
34	Trans 9	GraphTransit...	170.0				
35	Home	GraphStep	218.0				
36	Fill recipe ingredi...	GraphStep	250.0				
37	Mixer	GraphStep	282.0				
38	Transport Filling	GraphStep	314.0				
39	Filling	GraphStep	346.0				
40	Transport Labeling	GraphStep	378.0				
41	Labeling	GraphStep	410.0				
42	Filling Complete	GraphStep	442.0				
43	S_DISPLAY	Int	554.0	0			Affichage interne du paramètre de sortie S_NO
44	S_SEL_OLD	Int	556.0	0			Valeur précédente dans S_SEL
45	S_DISPIDX	Byte	558.0	16#0			Indice de l'étape affichée dans S_NO
46	T_DISPIDX	Byte	559.0	16#0			Indice de la transition affichée dans T_NO
47	MOP	Struct	564.0				Mode de fonctionnement
48	TICKS	Struct	568.0				Périodes d'horloge
49	SQ_FLAGS	Struct	580.0				Mémoires de graphe

OFF\_SQ

Propriétés Info Diagnostic

Options

Rechercher/R...

Rechercher :

Mot entier seu

Majuscules/mi

Rechercher da

Rechercher da

Utiliser des ca

Utiliser des ex

Document co

A partir de la

Sélection

Vers le bas

Vers le haut

Recher

Remplacer

Remplacer

Langues & R...

Langue d'édition

## ANNEXE

Variables spécifiques dans le modèle de projet dans les tables d'API de variables.

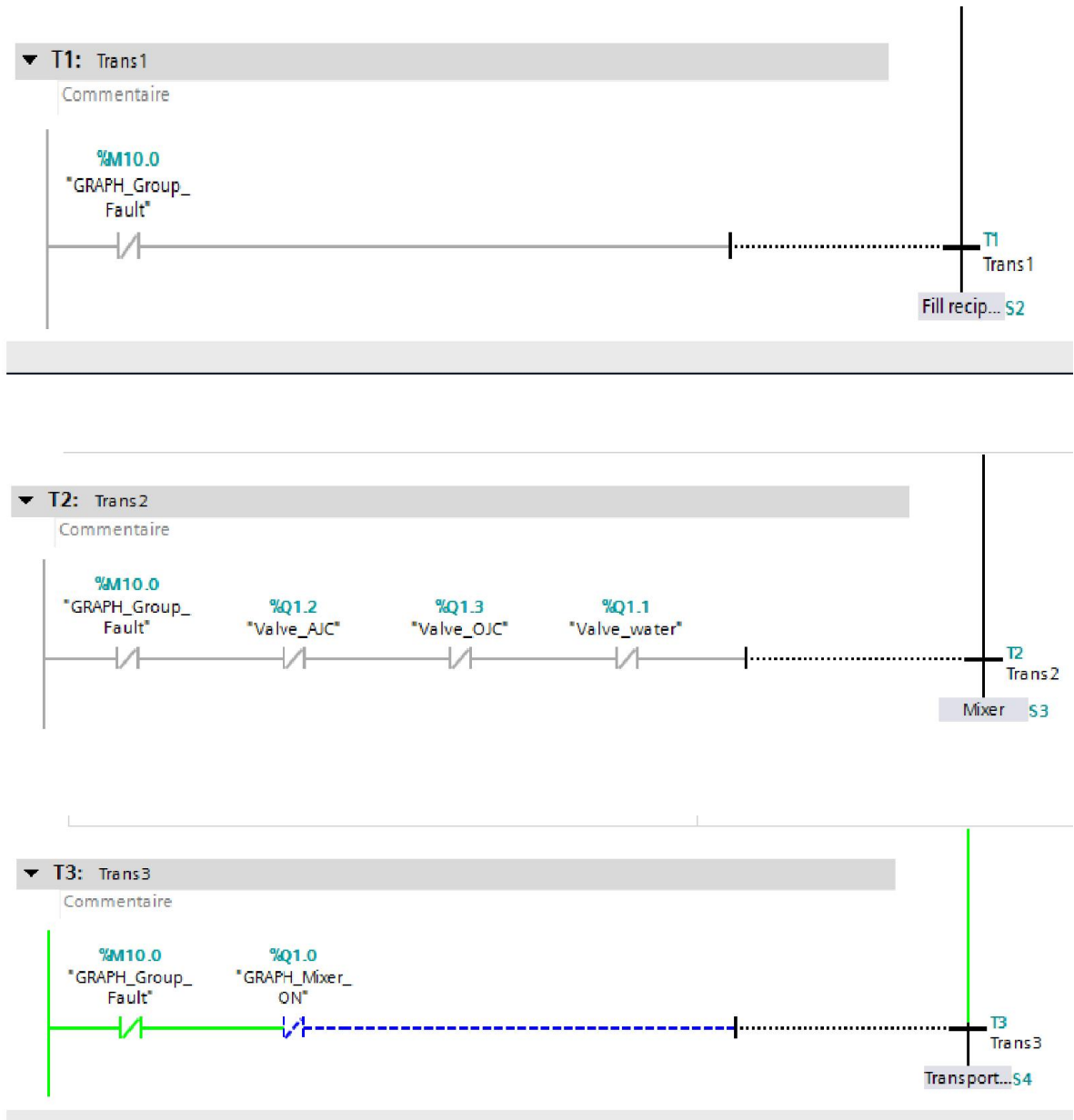
Variables API								
	Nom	Table des variables	Type de données	Adresse	Réma...	Visibl...	Acces...	Commentaire
1	GRAPH_Group_Fault	Tags GRAPH Se...	Bool	%M10.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	GRAPH_Count_Bottle	Tags GRAPH Seque...	Int	%MW12		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Valve_water	Tags GRAPH Seque...	Bool	%Q1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Valve_AJC	Tags GRAPH Seque...	Bool	%Q1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Valve_OJC	Tags GRAPH Seque...	Bool	%Q1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	GRAPH_Mixer_ON	Tags GRAPH Seque...	Bool	%Q1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Conveyor_Direction	Tags Conveyor	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Conveyor_Start_Conveyor	Tags Conveyor	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Valve_Fill_Bottle	Tags Filling	Bool	%Q1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	GRAPH_Start_Labeling	Tags GRAPH Seque...	Bool	%Q2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	GRAPH_Filling_Complete	Tags GRAPH Seque...	Bool	%M20.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Start_GRAPH_Sequence	Tags GRAPH Seque...	Bool	%M100.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Conveyor_Drive_OK	Tags Conveyor	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Conveyor_Start_Forward	Tags Conveyor	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Conveyor_Start_Backward	Tags Conveyor	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Best_before_date	Tags Best before date	Int	%MW2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	<ajouter>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Variables spécifiques dans le modèle de projet dans les tables HIM de variables.

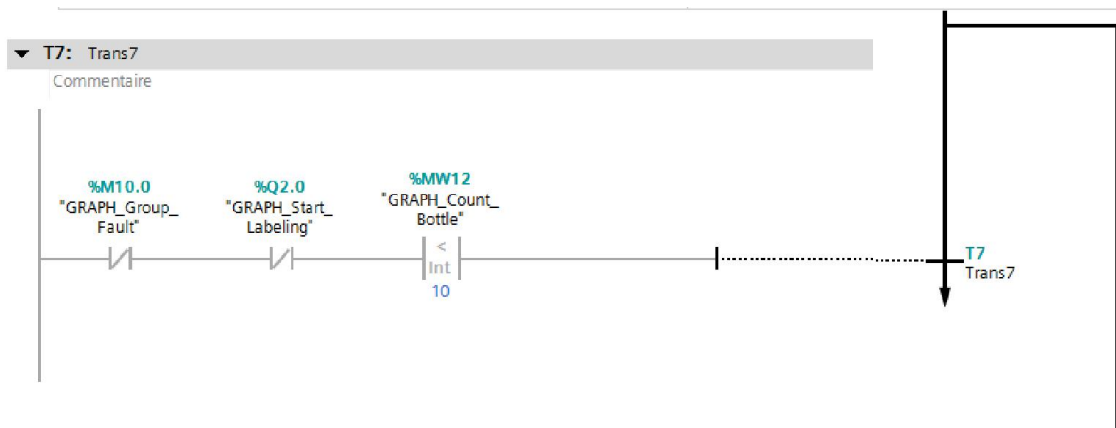
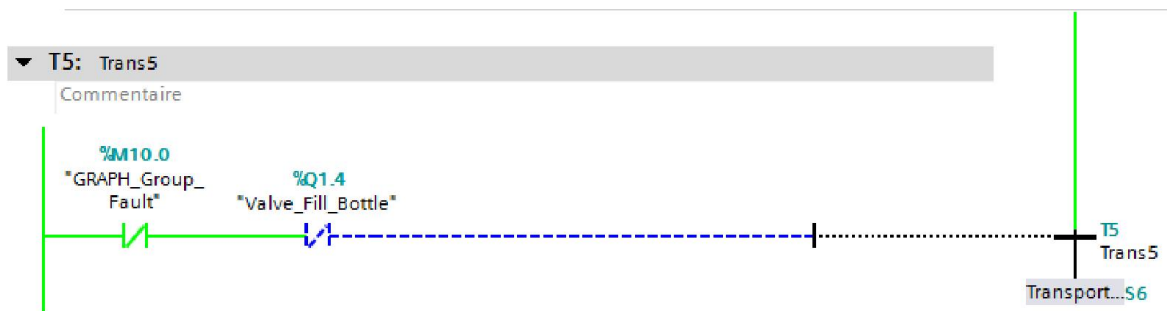
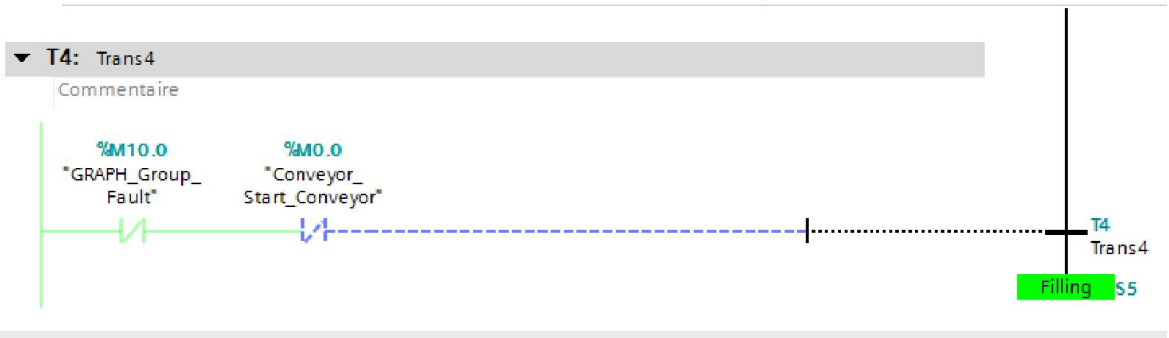
Table de variables standard						
	Nom	Type de données	Connexion	Nom API	Variable API	Adresse
	Best_before_date	Int	HML_Liaison_1	S7-300 Master	Best_before_date	%MW2
	Global_DB_BBD_Duration	Int	HML_Liaison_1	S7-300 Master	Global_DB.BBD_Duration	%DB1.DBW12
	Global_DB_Recipe_element_ap...	Time	HML_Liaison_1	S7-300 Master	Global_DB.Recipe_eleme...	%DB1.DBD0
	Global_DB_Recipe_element_or...	Time	HML_Liaison_1	S7-300 Master	Global_DB.Recipe_eleme...	%DB1.DBD4
	Global_DB_Recipe_element_wa...	Time	HML_Liaison_1	S7-300 Master	Global_DB.Recipe_eleme...	%DB1.DBD8
	GRAPH_Count_Bottle	Int	HML_Liaison_1	S7-300 Master	GRAPH_Count_Bottle	%MW12
	GRAPH_Sequence_DB_Fill recip...	Bool	HML_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.*Fil...	%DB2.DBX251.0
	GRAPH_Sequence_DB_Filling_X	Bool	HML_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.Fill...	%DB2.DBX347.0
	GRAPH_Sequence_DB_Labeling...	Bool	HML_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.La...	%DB2.DBX411.0
	GRAPH_Sequence_DB_Mixer_X	Bool	HML_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.Mi...	%DB2.DBX283.0
	GRAPH_Sequence_DB_Transpo...	Bool	HML_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.*Tr...	%DB2.DBX315.0
	GRAPH_Sequence_DB_Transpo...	Bool	HML_Liaison_1	S7-300 Master	GRAPH_Sequence_DB.*Tr...	%DB2.DBX379.0
	Numéro_vue_variable	UInt	<Variable intern...		<indéfini>	
	Position_Bottle	Int	<Variable intern...		<indéfini>	
	Position_Bottle(1)	Int	<Variable intern...		<indéfini>	
	Start_GRAPH_Sequence	Bool	HML_Liaison_1	S7-300 Master	Start_GRAPH_Sequence	%M100.0
	<ajouter>					

## B- Le programme :

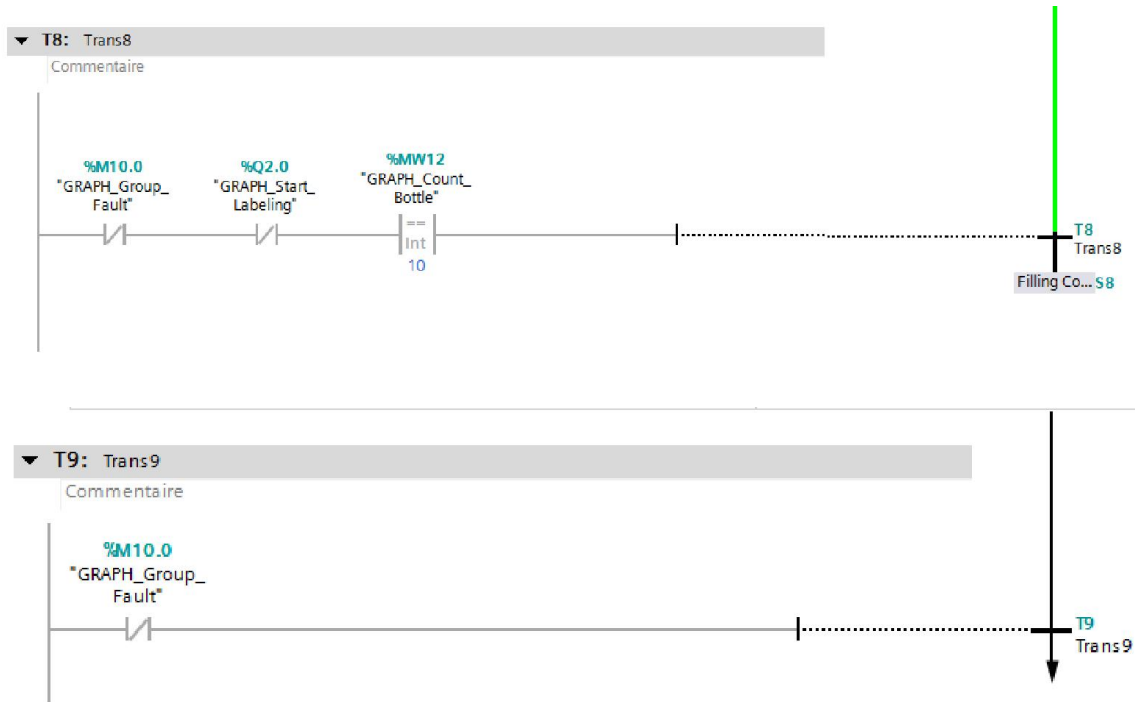
La transition de l'étape initiale vers l'une des étapes de station, respecte les conditions montrées dans la figure suivante :



# ANNEXE



# ANNEXE



Vous allez programmer la fonction LIST pour contrôler le tapis roulant. Pour cela, vous avez besoin de trois réseaux.

## ▼ Réseau 1 : .....

Commentaire

1	A	#Direction
2	A	#START_INPUT
3	S	#Forward
4	R	#Conveyor_DONE
5		
6		

## ▼ Réseau 2 : .....

Commentaire

1	AN	#Direction
2	A	#START_INPUT
3	S	#Backward
4	R	#Conveyor_DONE
5		

## ▼ Réseau 3 : .....

Commentaire

1	AN	#START_INPUT
2	R	#Forward
3	R	#Backward
4	S	#Conveyor_DONE
5		

## **Résumé :**

Le travail présenté dans ce mémoire, à automatiser toutes les opérations de remplissage, à réguler le niveau de remplissage de station et à contrôler les bouteilles. Ce dernier est une station de remplissage basée sur un API, en particulier le Siemens S7-300.

Les données de processus sont obtenues et traitées par le S7-300 intégré à step7. Nous créons également une interface utilisateur et un appareil avec WinCC Advanced Runtime pour TIA PORTAL V13 afin de faciliter la tâche de contrôle et de surveillance de la station par le travailleur.

## **Mots clés :**

Automatiser, station de remplissage, réguler le niveau de remplissage, API, Siemens, S7-300, step7, WinCC Advanced Runtime, TIA PORTAL V13.

## **Abstract:**

The work presented in this thesis, to automate all filling operations, to regulate the filling level of station and to control the bottles. The latter is a filling station based on an PLC, in particular the Siemens S7-300.

Process data is obtained and processed by the integrated S7-300 in step 7. We also create a user interface and a device with WinCC Advanced Runtime for TIA PORTAL V13 to facilitate the task of controlling and monitoring the station by the worker.

## **Key words :**

Automate, regulate the filling, API, Siemens, S7-300, step 7, WinCC Advanced Runtime, TIA PORTAL V13.

## **الخلاصة:**

العمل المقدم في هذه المذكرة، هو أتمتة جميع عمليات التعبئة، لتنظيم مستوى ملء المحطة والتحكم في الزجاجات. وهذا الأخير عبارة عن محطة التعبئة تستند إلى مبدأ التشغيل الخاص بـ API، وبشكل خاص برنامج سيمنس اس7-300. ويتم الحصول على البيانات العملية ومعالجتها من قبل اس7-300 متكاملة إلى الخطوة 7. كما قمنا أيضا بإنشاء واجهة إنسان ماكينة عن طريق برنامج وينسيس ل تيا بور تال نسخة 13 لتسهيل عملية الإشراف والتحكم في المحطة من طرف العامل.

## **الكلمات المفتاحية:**

تحكم المنطق الصناعي القابل للبرمجة. سيامانس. محطة التعبئة. API. اس7-300. الخطوة7. برنامج وينسيس. برنامج تيا بور تال نسخة13.