الجمهـــوريـــــة الجزائـــريـــــة الديمقـــراطيـــة الشـعبيـــة
République Algérienne Démocratique et Populaire
وزارة التـــعلـــيـــــم العـــــالـــــــي والبـــحـــــــث العـلـمـــــي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أ حمد
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة والأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industrielle
Spécialité : Maintenance de l'Automatismes et de l'Instrumentation

Thème

Automatisation et Supervision d'un Système de Contrôle

Présenté et soutenu publiquement par :

MEDJEBER Abderezak **HADRI Abderrahim**

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
ADJLOUA Abdelaziz	MCB	IMSI-Univ.D'Oran2	**President**
BELHADRI Kheira	MCB	IMSI-Univ.D'Oran2	**Encadreur**
KACIMI Abderrahmane	MCB	IMSI-Univ.D'Oran2	**Co-Encadreur**
BENARBIA Taha	MCB	IMSI-Univ.D'Oran2	**Examinateur**

Année 2024/2025

Remerciements

Tout d’abord, nous remercions Dieu tout puissant de nous avoir donné le courage et la patience durant toutes ces années d’études, et que grâce à lui, ce travail a pu être réalisé.

Comme nous tenons à exprimer notre profonde gratitude à notre encadreur madame **BELHADRI Kheira** qui nous a soutenus et orienter tout au long de notre travail.

Nous remercions aussi notre Co-Encadreur **Mr. KACIMI Abderrahmane** pour l’aide et pour les conseils précieux pendant la réalisation de ce travail.

Nos remerciements s’adressent également aux membres de jury qui nous font l’honneur de juger notre travail.

Enfin, nos remerciements vont à toutes les personnes qui nous ont aidés et soutenue de près ou de loin dans ce travail.

Dédicaces

Je dédie ce modeste travail à Commençant par ma Chère Mère et mon Chère Père qui m'ont soutenu et aidé tout le long de mon parcours

A mes Frères et sœurs A tous les membres de la famille chacun en son nom.

A tous mes amis et les gens qui m'aiment. A mon binôme ABDEREZAK MEDJEBER.

HADRI Abderrahim

Dédicace

C'est avec un grand plaisir que je dédie ce modeste travail :

A mes chers parents," ma mère " et" mon père " qui m'ont soutenu avec leurs douae et aider tout au long de mon parcourt.

A mes chers frères et sœurs.

A tous les membres de ma famille.

A mon binôme ABDERRAHIM HADRI.

A tous mes amis.

A tous mes collègues de la promotion Master 2 MAI.

ABDEREZAK MEDJEBER

Résumé

Ce travail a pour objectif la conception et la mise en œuvre d'un système automatisé destiné à contrôler le niveau de liquide dans un réservoir. Le projet s'appuie sur des logiciels tels que TIA Portal et Factory I/O. Après l'étude du système et des équipements utilisés, nous avons développé une solution basée sur un API de type S7-1200.Par ailleurs, une interface homme-machine (HMI) a été développée à l'aide du logiciel WinCC, afin de faciliter les opérations de supervision et de visualisation des données en temps réel.

Mots clés: TIA Portal, Factory I/O, API type S7-1200, WinCC, HMI.

Abstract

This work aims to design and implement an automated system intended to control the liquid level in a tank. The project relies on software tools such as TIA Portal and Factory I/O. After studying the system and the equipment used, we developed a solution based on a programmable logic controller (PLC) of the S7-1200 type. Additionally, a human-machine interface (HMI) was developed using WinCC software, to facilitate the supervision, and real-time data visualization operations.

Key words: TIA Portal, Factory I/O, PLC S7-1200 type, HMI, WinCC.

الملخص

يهدف هذا العمل إلى تصميم وتنفيذ نظام آلي للتحكم في مستوى السائل داخل خزان. ويعتمد المشروع على برامج مثل TIA Portal وFactory I/O.وبعد دراسة النظام والمعدات المستخدمة، قمنا بتطوير حل يعتمد على وحدة تحكم قابلة للبرمجة من نوع S7-1200 .بالإضافة إلى ذلك، تم تطوير واجهة رجل-آلة (HMI) باستخدام برنامج WinCC وذلك لتسهيل عمليات الإشراف وعرض البيانات في الزمن الحقيقي.

الكلمات المفتاحية: وحدة تحكم المنطقي، TIA Portal،Factory I/O ، واجهة رجل-آلة HMI

Table des matières

Introduction générale 1

CHAPITRE I : Système Automatisé et Automate Programmable Industriel

I.1 Introduction 3
I.2 Système automatisé 3
I.2.1 Définition 3
I.2.3 Structure d'un système automatisé 4
I.2.4 Partie opérative 4
I.2.5 Partie commande 6
I.2.6 Partie pupitre 7
I.3 Automates programmables industriels 7
I.3.1 Définition et historique des API 7
I.3.1.1 Définition 7
I.3.1.2 Historique 7
I.3.2 Domaines d'emploi des automates 7
I.3.3 Notions de choix d'un automate programmable (API) 7
I.3.4 Architecture des(API) 8
I.3.4.1 Aspect extérieur 8
I.3.4.2 Structure interne d'un automate 9
I.3.5 Nature des données manipulées par API 10
I.3.6 Programmation d'un API 10
I.3.7 Principe de fonctionnement d'un API 12
I.4 Présentation de quelques gammes SIMATIC S7 12
I.4.1 SIMATIC S7-200 12
I.4.2 SIMATIC S7 -300 13
I.4.3 SIMATIC S7-400 13
I.4.4 SIMATIC S7-1200 14
I.4.5 SIMATIC S7-1500 14
I.5 Présentation de l'automate S7-1200 15
I.5.1 Choix de la CPU S7-1200 16
I.5.2 Sélection des modules d'entrées/sorties 16
I.5.3 Options pour l'extension de la CPU S7-1200 16
I.5.4 Communication et les réseaux industriels 17
I.5.5 Différence entre PROFIBUS et PROFINET 17
I.6 Logiciel TIA portal 18

I.6.1 TIA Portal V16 18
I.6.2 Vue de Portail 18
I.6.3 Vue de projet 19
I.6.4 Blocs 19
I.6.4.1 Blocs d'organisation 20
I.6.4.2 Blocs fonctionnels (FB) 20
I.6.4.3 Bloc fonction (FC) 21
I.6.4.4 Blocs de données (DB) 21
I.7 S7-PLCSIM V16 21
I.8 Supervision des systèmes automatisés 22
I.8.1 Définition de la supervision 22
I.8.2 Importance de la Supervision 22
I.8.3 Techniques de supervision 22
I.8.4 Supervision dans un environnement SCADA 24
I.8.4.1 Définition du SCADA 24
I.8.4.2 Eléments du système SCADA 24
I.9 Logiciel de supervision WINCC 25
I.10 Conclusion 26

CHAPITRE II : Généralité sur la Régulation

II.1 Introduction 28
II.2 Définition 28
II.3 Objectif de la régulation 28
II.4 Asservissement et régulation 28
II.4.1 Asservissement 29
II.4.2 Régulation 29
II.5 Concept de boucle ouverte/fermée 29
II.5.1 Boucle fermée 29
II.5.2 Boucle ouverte 29
II.6 Constituants d'une chaine de régulation 30
II.7 performances d'un système 31
II.7.1 Stabilité 32
II.7.2 Rapidité 32
II.7.3 Précision 33
II.7.4 Dépassement 33
II.8 Types des régulateurs industriels 34

II.8.1 Régulateur Tout ou Rien (TOR) 34
II.8.2 Régulateurs standards P, PI, PD, PID 34
II.8.2.1 Actions de base 34
II.8.2.2 Régulateur PI 35
II.8.2.3 Régulateur PD 35
II.8.2.4 Régulateur PID 36
II.9 Effet des actions du Régulateur PID 38
II.10 Sens d'action d'un régulateur 39
II.11 Choix et dimensionnement de régulateur PID 39
II.11.1 Méthode de Ziegler-Nichols 39
II.11.1.1 Approche de Ziegler-Nichols pour les systèmes en boucle ouverte 40
II.11.1.2 Approche de Ziegler-Nichols en boucle fermée 41
II.12 Conclusion 43

CHAPITRE III : Programmation, Simulation et Supervision

III.1Introduction 45
III.2 Cahier des charges 45
III.3 Logiciel Factory I/O 46
III.3.1 Interface du logiciel 46
III.3.2 Choix du système simulé 47
III.3.3 Conception de système 48
III.3.4 Description du matériel de simulation 49
III.3.4.1 Réservoir d'eau 49
III.3.4. 3 Capteur de niveau capacitif 50
III.3.4.4 Débitmètre 50
III.3.5 Description des entries/sorties de système 51
III.4 Création d'un projet et configuration du matériel 51
III.4.1 Création d'un projet 51
III.4.2 Configuration du matériel 52
III.5 Programmation 53
III.5.1 Tableau des variables 53
III.5.2 Blocs de programmation utilisée 54
III.5.3 Programmes 54
III.5.3.1 Bloc OB1 54
III.5.3.2 Bloc FB 55
III.5.3.3 Bloc OB30 61

III.5.3.4 Bloc BD 62
III.6 Supervision 62
III.6.1 Création et configuration de L'HMI 62
III.6.2 Etablissement de la connexion entre la CPU et L'HMI 63
III.6.3 Variables HMI 63
III.6.4 Création des vues HMI 64
III.6.5 Présentation des différant vues du projet 64
III.6.5.1 Vue global 64
III.6.5.2 Vue mode PID 65
III.6.5.3 Vue mode ON/OFF 65
III.6.5.4 Vue alarmes 66
III.6.6 Configuration des alarmes au niveau de l'HMI 66
III.6.7 Gestion des utilisateurs 66
III.7 Simulation de projet 67
III.7.1 Chargement de programme 67
III.7.2 Démarrage de HMI 69
III.7.3 Simulation de système 70
III.7.3.1 Automatique PID 70
III.7.3.2 Automatique ON/OFF 71
III.7.3.3 Mode manuel 71
III.7.3.4 Création d'une alarme 72
III.8 Conclusion 72
Conclusion général 74
Annexes 81

Liste des figures

CHAPITRE I : Système Automatisé et Automate Programmable Industriel

Figure I. 1: Les objectifs de l'automatisation 3
Figure I. 2 : Structure d'un système automatisé 4
Figure I. 3 : Capteur de niveau liquide 5
Figure I. 4 : Capteur de température (RTD) 5
Figure I. 5 : Distributeur pneumatique 5
Figure I. 6 : Contacteur 5
Figure I. 7 : Moteur électrique 6
Figure I. 8 : Vérins pneumatiques 6
Figure I. 9 : Convoyeur 6
Figure I. 10: Automate compact LOGO 8
Figure I. 11: Automate modulaire Siemens 8
Figure I. 12: Structure interne d'un automate 9
Figure I. 13: SIMATIC S7-200 13
Figure I. 14: SIMATIC S7-300 13
Figure I. 15: SIMATIC S7-400 14
Figure I. 16: Automate programmable S7-1200 14
Figure I. 17: S7-1500 15
Figure I. 18: Le CPU S7 1200 17
Figure I. 19: L'interface de lancement de TIA Portal V16 18
Figure I. 20: Vue de portail de TIA Portal 19
Figure I. 21: Vue de projet de tai portal 19
Figure I. 22: Organisation des blocs OB dans un automate siemens 20
Figure I. 23: Les blocs de données 21
Figure I. 24: SIMATIC RTU3030C 24
Figure I. 25: Présentation de l'enveniment HMI avec TIA portal 25

CHAPITRE II : Généralité sur la Régulation

Figure II. 1: Système en boucle fermée (BF) 29
Figure II. 2: Système en boucle ouverte (BO) 29
Figure II. 3: Représentation de schéma fonctionnelle d'une boucle de régulation 30
Figure II. 4: Performances d'un système de commande 31
Figure II. 5: Stabilité du système 32

Figure II. 6: Rapidité du système 32
Figure II. 7: Précision du système 33
Figure II. 8: Dépassement du système 33
Figure II. 9: Régulateur proportionnelle-intégral 35
Figure II. 10: Régulateur proportionnelle-dérivée PD 36
Figure II. 11: Structure mixte 37
Figure II. 12: Structure parallèle 37
Figure II. 13: Structure série 38
Figure II. 14: Sens d'action 39
Figure II. 15: Régulateur sens inverse 39
Figure II. 16: Régulateur sens direct 39
Figure II. 17: Réponse indicielle en boucle ouverte 40
Figure II. 18: Essai d'instabilité en boucle fermée 42

CHAPITRE III : Programmation, Simulation et Supervision

Figure III. 1: La barre des tâches 46
Figure III. 2: Scènes disponibles en Factory I/O 47
Figure III. 3: Matériel utilisé en Factory I/O 48
Figure III. 4: Réservoir 49
Figure III. 5 :Vanne de contrôle 49
Figure III. 6: Capteur de niveau capacitif 50
Figure III. 7: Débitmètre 50
Figure III. 8: Entries/sorties de système 51
Figure III. 9: Création d'un nouveau projet 51
Figure III. 10: Choix de l'API et CPU 52
Figure III. 11: Module d'extension AQ 2x14BIT 53
Figure III. 12: Tableau des variables 53
Figure III. 13: Communication avec Factory I/O 54
Figure III. 14: Contrôle de niveau 55
Figure III. 15: Norme et scale 56
Figure III. 16: Programme marche, arrêter 57
Figure III. 17: Remplissage ON/OFF 57
Figure III. 18 : Mode manuel 58
Figure III. 19: Ouverture de vanne d'évacuation 58

Figure III. 20: Fermeture de vanne d'évacuation 59
Figure III. 21: Alarmes mode PID 59
Figure III. 22: Alarmes niveau haut et bas 60
Figure III. 23: Alarmes sur Factory I/O 60
Figure III. 24: Bloc PID compacte 61
Figure III. 25: Tableau de variables BD 62
Figure III. 26: Insertion d'une station HMI dans TIA Portal 62
Figure III. 27: Liaison entre api et HMI 63
Figure III. 28: Variables HMI 63
Figure III. 29: Création d'une vue 64
Figure III. 30: Vue global 64
Figure III. 31: Vue mode PID 65
Figure III. 32: Vue mode ON/OFF 65
Figure III. 33: Vue alarmes 66
Figure III. 34: Configuration d'alarmes 66
Figure III. 35: Sécurité d'HMI 67
Figure III. 36: Création des autorisations 67
Figure III. 37: Chargement de programme 68
Figure III. 38: PLC SIM 68
Figure III. 39: Menu drivers 69
Figure III. 40: Connexion avec PLC SIM 69
Figure III. 41 : Variation de PV et SP en mode PID 70
Figure III. 42: Réservoir rempli 70
Figure III. 43: Variation de mesure en mode ON/OFF 71
Figure III. 44: Alarme niveau haut 72

Liste des tableaux

CHAPITRE I : Système Automatisé et Automate Programmable Industriel

Tableau I. 1: Les éléments clés (contacts et bobines) d'un réseau LD 11

CHAPITRE II : Généralité sur la Régulation

Tableau II. 1: Méthode de Ziegler-Nichols en boucle ouverte..... 41

Tableau II. 2: Méthode de Ziegler-Nichols en boucle fermée..... 42

Liste des abréviations

PO : Partie Opérative

PC : Partie Commande

API : Automate Programmable Industriel

RTD : Résistance Température Detector

HMI : Human Machine Interface

TOR : Tout Ou Rien

E/S : Entrée/Sortie

SFC : Séquentiel Fonction Chart

GRAFCRT : Graphe Fonctionnel de Commande des Etapes et Transitions

FBD : Fonction Bloc Diagramme

ST : Structure de texte

IL: Instruction List

LD: Ladder Diagramme

RAM: Randam Access Memory

CPU: Central Processing Unite

SP: Signal Power supplay

FM: Function Module

SM: Signal Module

ROFIBUS: Procès Field Bus

PROFINET: Process Field Network

IP: Internet Protocol

TCP: Transmission Control Protocol

GPRS: General Packet Radio Service

SIMATIC: Siemens Automatic

TIA PORTAL: Totally Integrated Automation Portal

OB : Bloc d'organisation

FB : Bloc de fonction

FC: Fonction

DB: Data Bloc

SCADA: Supervisory Control And Data Acquisition

RTU : Remote Terminal Unite

MTU : Master Terminal Unit

WINCC : Windows Contrôle Center

BF : Boucle Fermée

BO : Boucle Ouverte

Kp : Coefficient de Gain Proportionnel

P : Proportionnel

I : Intégral

D : Dérivé

PID : Proportionnel Intégral Dérivé

Ti : Temps d'intégration

Td : Temps dérivation
SP : Set point
PV : Process variable

Introduction Générale

Introduction générale

Dans l'industrie l'automatisation joue un rôle crucial dans le processus d'optimisation des processus de production, de réduction des coûts et d'amélioration de la qualité des produits fabriqués. Par conséquent, la combinaison de technologies avancées telles que les automates programmables industriels, les interfaces homme-machine, les systèmes de contrôle et d'acquisition des données a rendu possible la conception du système intelligent, flexible, et hautes performances. Ce travail vise précisément cette initiative et propose une étude approfondie de la conception, de programmation, de simulation et de supervision d'une unité de contrôle automatisée. En effet, nous avons abordé les concepts théoriques de l'automatisation et de la régulation en intégrant des régulateurs PID les plus utilisés en industriel.

Ce projet met en œuvre une solution d'automatisation pour le contrôle du niveau d'eau dans un réservoir, basée sur une modélisation réaliste, une programmation fiable et une interface intuitive. Il s'appuie sur des outils industriels tels que TIA Portal, Factory I/O et WinCC pour la conception, la simulation et la supervision du système. L'approche adoptée répond aux standards industriels en matière de contrôle automatisé et de supervision temps réel.

Ce mémoire est divisé en trois chapitres principaux.

- Le premier chapitre se consacre à l'étude des systèmes automatisés et des automates programmables industriels. En premier temps, nous définissons les principaux composants d'un système automatisé, le fonctionnement des API, ainsi que les caractéristiques de la gamme Siemens S7. En particulier, nous considérons le S7-1200 installé dans notre situation.
- Le deuxième chapitre est centré sur la régulation. Nous définissons ici les trois types de régulateurs régulièrement utilisés dans l'industrie : P, PI et PID. En outre, nous nous concentrerons sur les performances du système et la méthode du réglage établie par Ziegler-Nichols. Notre objectif est garantir un fonctionnement optimal de la régulation du niveau d'eau dans le réservoir.
- Le troisième chapitre est consacré à la programmation, la simulation et la supervision. On y trouvera en détail la simulation du système à l'aide de Factory I/O, la configuration matérielle et logicielle sous TIA Portal, l'implémentation de trois différents modes de fonctionnement soit manuel, automatique ON/OFF, et PID et l'interface HMI développé à l'aide de WinCC pour une exécution intuitive et facile.

CHAPITRE I : Système Automatisé et Automate Programmable Industriel

I.1 Introduction

L'automatisation industrielle est très importante pour l'optimisation des processus de production. Ce chapitre examine les principes de base des systèmes automatisés et des automates programmables industriels, qui assurent efficacité, exactitude et sécurité au sein d'un contexte industriel contemporain. Il expose les mécanismes de ces technologies et les instruments nécessaires pour leur application, garantissant ainsi une gestion astucieuse et efficace des activités industrielles.

I.2 Système automatisé

I.2.1 Définition

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

I.2.2 Objectif d'un système automatisé

L'automatisation permet d'accroître la valeur ajoutée en visant plusieurs objectifs comme illustré sur la figure I.1 [2] :

- ✓ Réduire les coûts de production.
- ✓ Améliorer la qualité grâce à des contrôles automatisés.
- ✓ Rendre la production plus flexible.
- ✓ Eliminer les tâches pénibles ou dangereuses.
- ✓ réaliser des opérations complexes ou impossibles à effectuer manuellement.

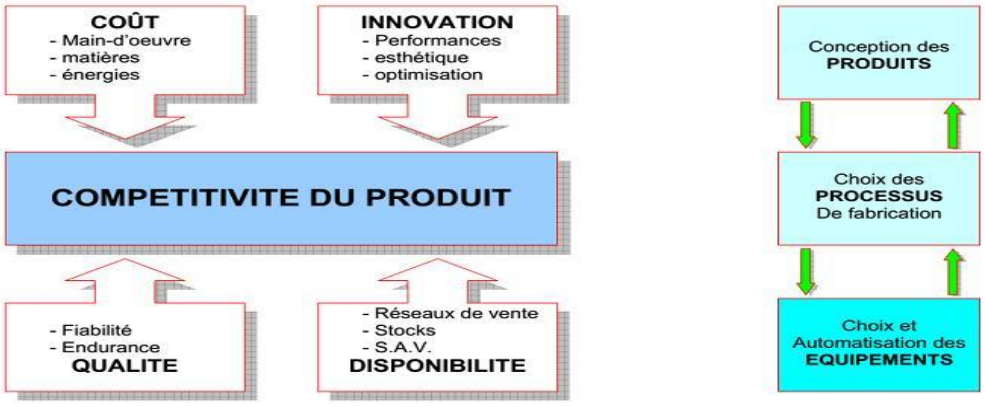


Figure I.1: Les objectifs de l'automatisation [6].

I.2.3 Structure d'un système automatisé

Tous les systèmes automatisés possèdent une structure générale composée de 3 parties fondamentales d'un partie commande, d'une partie opérative et une partie pupitre (supervision) [3].La figure I.2 montre les structure d'un système automatisé.

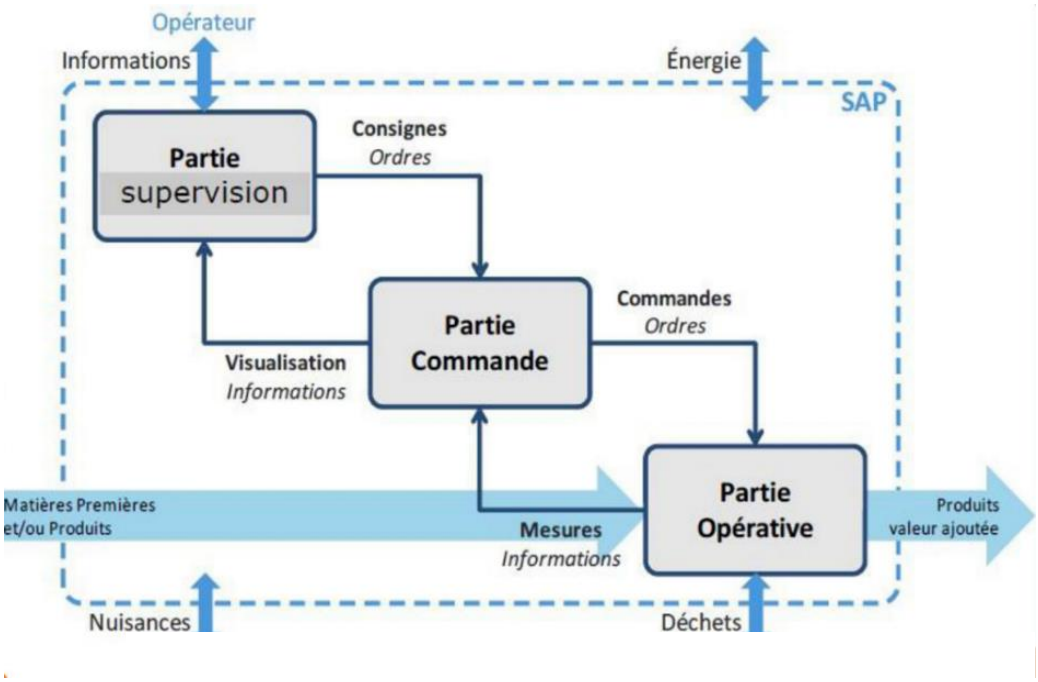


Figure I. 2 : Structure d'un système automatisé [3].

I.2.4 Partie opérative

Sa fonction principale est d'apporter de la valeur ajoutée sur la matière première. Est une représentation physique d'un processus industriel. Elle obéit aux instructions envoyées de la commande par les opérateurs humains de supervision afin d'agir sur les actionneurs et pré-actionneurs du procédé.

En retour, la PO renvoie les informations du procédé aux opérateurs humains de supervision via des capteurs) [3].

- **Capteurs**

Est élément qui capte les états (position, vitesse, température, force, pression…) de la PO ou de l'environnement externe, puis les transforme en signaux exploitables (généralement électrique : tension ou intensité) sous forme :

– Logique (0 ou 1).

– Analogique.

– Numérique.

Enfin transmet ces signaux à la partie commande [2].

Les figures (I.3, I.4) montrent des exemples sur les captures.



Figure I. 3 : Capteur de niveau liquide



Figure I. 4 : Capteur de température (RTD)

- **Pré-actionneur**

Dispositif qui permet le transfert de l'énergie vers l'actionneur sur ordre de la partie commande. (Le pré-actionneur réalise l'interface de dialogue (PC/PO) [2].

Exemples sur les pré-actionneurs (voir les figures I.6, I.5) :

- Des contacteurs destinés aux moteurs électriques.

- Distributeurs pour vérins à air ou hydrauliques.



Figure I. 6 : Contacteur



Figure I. 5 : Distributeur pneumatique

- **Actionneurs**

Actionneur est l'organe de la partie opérative qui, recevant un ordre de la partie commande via un éventuel pré-actionneur. Les actionneurs (voir les figuresI.7, I.8) convertissent l'énergie d'entrée disponible sous une certaine forme (électrique, pneumatique, hydraulique) en une énergie utilisable sous une autre forme, par exemple :

- Energie mécanique destinée à provoquer une translation de chariot (l'actionneur pouvant être un vérin hydraulique ou pneumatique).

- Energie mécanique destinée à provoquer une rotation de broche (l'actionneur pouvant être alors un moteur électrique [4].



Figure I. 7 : Moteur électrique



Figure I. 8 : Vérins pneumatiques

- **Effecteurs**

Ce sont les éléments terminaux (voir la figure I.9). Ils agissent directement sur la matière d'œuvre en vue de lui apporter une valeur ajoutée. Ils transforment l'énergie reçue de l'adaptateur en une action ou un effet appliqué à la matière d'œuvre [2].



Figure I. 9 : Convoyeur

I.2.5 Partie commande

Elle est en généralement constituée d'un automate qui contient un programme qui gère le fonctionnement du Système. Elle détermine les ordres nécessaires au fonctionnement de la partie opérative en fonction des consignes transmises via l'interface homme-machine (par l'opérateur) ou par acquisition des données (les informations reçues par des capteurs) [2].

Les fonctions essentielles exécutées par la section de commande sont [2] :

- Communiquer des informations avec l'opérateur.
- Contrôler la puissance.
- Obtenir les informations.
- Partager des données avec d'autres systèmes**.**

I.2.6 Partie pupitre

Elle est constituée de boutons de contrôle et d'indication, permettant à l'opérateur de piloter le système (démarrer, arrêter, lancer le cycle), elle permet également la possibilité d'observer les divers états du système grâce à des indicateurs une interface utilisateur (HMI) [4].

I.3 Automates programmables industriels

I.3.1 Définition et historique des API

I.3.1.1 Définition

Un automate programmable industriel (API), également connu sous le nom de programmable logique Controller (PLC) en anglais, est un dispositif électronique destiné pour piloter des procédés automatiques en temps réel dans des environnements industriels. L'API peut être programmée par un automaticien non informaticien [5].

I.3.1.2 Historique

Les API sont apparues aux Etats-Unis vers 1969 en réponse à la demande de l'industrie automobile pour développer des chaînes de fabrication automatisées capables de s'adapter aux évolutions technologiques et aux modèles de production. L'API a ainsi remplacé les armoires à relais en raison de sa flexibilité, ainsi que des coûts élevés de câblage et de mise au point associés aux systèmes de commande complexes. Les premiers fabricants américains d'api étaient les entreprises Modicon et Allen Bradley. Au fil du temps, les API ont bénéficié de mises à jour tous les4 à 7 ans pour tirer parti des avancées des composants électroniques tels que les microprocesseurs et les microcontrôleurs [5].

I.3.2 Domaines d'emploi des automates

Les automates programmables sont désormais utilisés dans un large éventail d'industries pour le contrôle des machines et des lignes de production et même l'automatisation des bâtiments : comme l'éclairage, le chauffage [3].

I.3.3 Notions de choix d'un automate programmable (API) [3]

- Le nombre d'entrées/sorties.
- La nature des entrées/sorties (TOR, numérique, analogique).
- Temps de traitement.
- Capacité de la mémoire.
- Nombre de compteurs.
- Nombre de temporisateurs.

- Les moyens de dialogue et langage de programmation.

I.3.4 Architecture des(API)

I.3.4.1 Aspect extérieur

Les API peuvent être de type compact ou modulaire

- **Automate compact** : tous les composants fondamentales (processeur, alimentation, entrées/sorties) est rassemblé dans un seul boîtier. Ce type regroupe les micro-automates comme le LOGO de Siemens ou le ZELIO de Schneider (voir la figure I.10). Il peut exécuter certaines fonctions supplémentaires limitées (comptage rapide, E/S analogiques) et accepte peu d'extensions. Il est destiné aux automatismes simples et de petite taille [6].



Figure I. 10: Automate compact LOGO [6].

- **Automate modulaire** : les diverses fonctions (processeur, alimentation, E/S…) sont réparties sur plusieurs modules séparés, installés sur un ou plusieurs racks. Ce type est utilisé dans des automatismes complexes où une grande puissance de traitement (voir la figure I.11), une grande capacité d'extension et une flexibilité sont nécessaires [6].



Figure I. 11: Automate modulaire Siemens [6].

I.3.4.2 Structure interne d'un automate

La structure interne d'un API peut s'être représentée comme suit (voir la figure I.12) [7] :

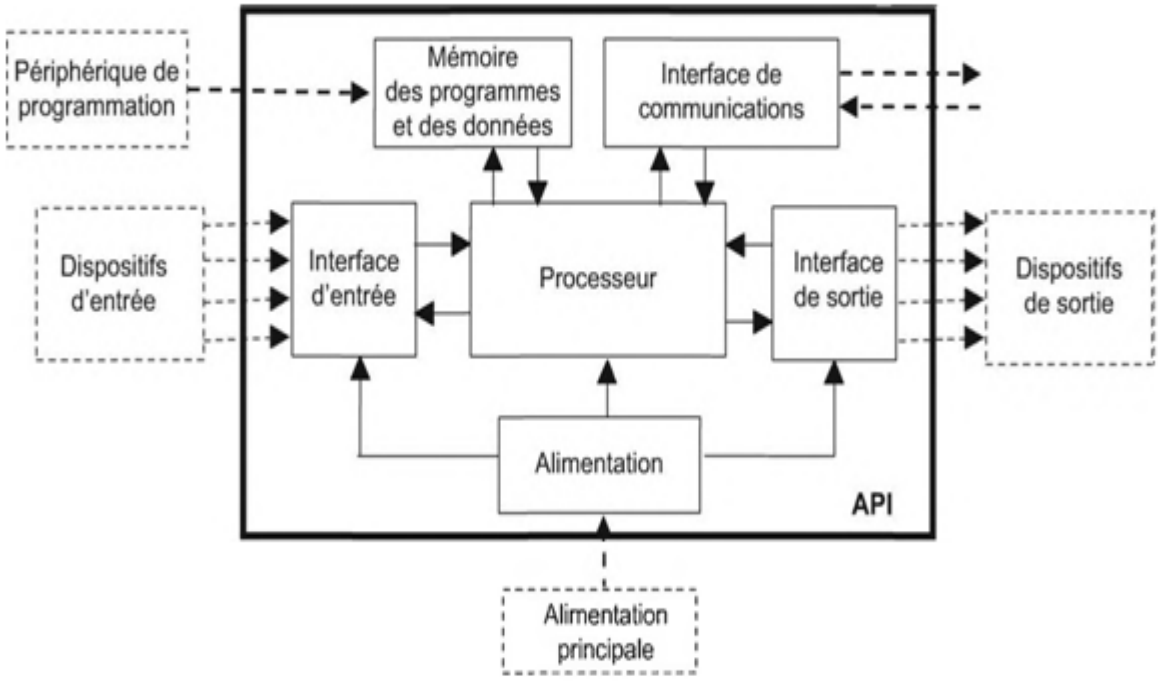


Figure I. 12: Structure interne d'un automate [7].

- **Processeur ou unité de traitement :** Intègre le microprocesseur, le CPU interprète les signaux d'entrée et exécute les actions de commande conformément au programme enregistré en mémoire, en communiquant aux sorties les décisions sous forme de signaux d'action.
- **Périphérique de programmation :** Est utilisé pour entrer le programme dans la mémoire du processeur. Ce programme est développe sur le périphérique, puis transféré dans la mémoire de API**.**
- **Mémoire :** Contient le programme qui détermine les actions de commande exécutées par le microprocesseur. Elle stocke également les données qui proviennent des entrées en vue de leur traitement, ainsi que celles des sorties.
- **Interfaces d'entrées/sorties :** Permettent au processeur de recevoir et d'envoyer des informations aux dispositifs extérieurs. Les entrées peuvent être des interrupteurs, comme dans le cas d'une perceuse automatisée, ou de capteurs tels que des cellules photoélectriques pour le comptage, des sondes de température, des débitmètres. Les sorties peuvent être des bobines de moteur, des électrovannes. Les dispositifs d'entrées-sorties se classent en trois catégories, selon qu'ils produisent des signaux TOR,

numériques ou analogiques. Les dispositifs qui génèrent des signaux TOR ou numériques sont ceux dont les sorties sont de type tout ou rien.

- **Module d'alimentation :** Unité d'alimentation est essentielle, puisqu'elle transforme une tension alternative en une basse tension continue (5 V) nécessaire au processeur et aux modules d'entrées-sorties.
- **Interface de communication :** Est utilisée pour recevoir et transmettre des données sur des réseaux de communication qui relient AP1 a d'autres API distants. Elle intervient dans des opérations telles que la vérification des périphérique, l'acquisition de données, la synchronisation entres des applications et la gestion de la connexion.

I.3.5 Nature des données manipulées par API

Les informations peuvent être des types suivants [8] :

- **Analogique :** L'information est de nature continue et peut prendre une valeur compris dans un intervalle précisément définie. Il s'agit du genre de renseignement fourni par un capteur (pression, température, variateur de vitesse, etc.).
- **Numérique :** L'information est contenue dans des mots codés sous forme binaire ou en hexadécimale. Ce type de donnée est généralement délivré par un ordinateur ou un module intelligent.
- **Tout ou rien (TOR) :** L'information ne peut prendre que sous deux formes (vrai/faux, ou 1/0...). C'est le genre d'information fournie par un capteur ou un interrupteur.

I.3.6 Programmation d'un API

Chaque automate est configuré à l'aide d'une console de programmation propriétaire ou d'un ordinateur équipé d'un logiciel spécialisé. La norme IEC61131-3 définit cinq langages de programmation utilisables pour le développement d'applications d'automatisation [7].

- **SFC « séquentiel fonction char » ou Grafcet** : Ce langage facilite la programmation de l'ensemble des opérations séquentielles [7].
- **FBD « fonction block diagramme » ou schéma par blocs :** Ce langage offre une programmation graphique à l'aide de blocs, qui représentent des variables, des opérations ou des fonctions. Il gère toutes sortes de variables [7].
- **ST « structure texte » ou texte structuré** : C'est un langage script, permet de concevoir tout type d'algorithmes plus ou moins complexes.
- **IL « instruction List** » : C'est un langage script, à une instruction par ligne. Il peut être comparé au langage assembleur [7].

- **LD « Ladder diagramme » ou schéma à relais** : le langage à contacts LD (Diagramme en échelle) est constitué de réseaux que l'automate lit successivement. Ces réseaux comprennent divers symboles qui illustrent les entrées et sorties de l'automate, les procédés séquentiels (des temporisations, compteurs, etc.), les opérations ainsi que les bits internes au système de l'automate (ces derniers servent à activer ou désactiver certaines fonctionnalités de l'automate, comme l'initialisation des grafcets), on distingue trois catégories d'éléments de langages [9] :

• Les entrées (aussi appelées contact), qui permettent de lire une variable booléenne,

• Les sorties (également connues sous le nom de bobines), qui autorisent l'écriture d'une variable booléenne,

• Les blocs fonctionnels, qui facilitent l'exécution de fonctions sophistiquées.

Le tableau présente les éléments clés (contacts et bobines) d'un réseau LD (voir le tableau I.1).

Object graphique	Nom
-| |-	Contact ouvert
-|/|-	Contact inverseur
-| P |-	Contact avant montant
-|N|-	Contact fermé au front descendant
-( )-	Bobine normalement ouverte
-(/)-	Bobine normalement fermée
-(S)-	Bobine Latch (maintenu à 1 une fois actionné)
-(R)-	Bobine Reset (remise à 0 de la bobine latch)
-(P)-	Bobine active au front montant de son entrée
-(N)-	Bobine active au front descendant de son entrée

Tableau I. 1: Les éléments clés (contacts et bobines) d'un réseau LD [9].

I.3.7 Principe de fonctionnement d'un API

Le traitement a lieu en trois phases :

- **Acquisition des entrées** : Prise en compte des informations du module d'entrées et écriture de leur valeur dans RAM (zone donnée).
- **Traitement des données :** le processeur exécute les instructions de la mémoire programme en fonction des informations ´ de la mémoire des données. Cette exécution se traduit par la modification de certaines variables et leur mise à jour dans la zone correspondante.
- **Emissions des ordres :** les images des sorties dans la mémoire des données sont transférées dans le module de sortie pour être convertis en signaux électriques pour la commande des pré-actionneurs et des dispositifs de visualisation. Ces valeurs sont verrouillées jusqu'au cycle prochain [10].

I.4 Présentation de quelques gammes SIMATIC S7

Dans la gamme S7, on identifie cinq principales catégories d'automates programmables industriels qui sont détaillées dans les paragraphes suivants.

I.4.1 SIMATIC S7-200

La gamme S7-200 comprend des automates programmables de petite taille, adaptés à diverses applications d'automatisation. La figure I.13 illustre un automate S7-200 de petite taille.

Grâce à son format réduit, ses options d'extension, son coût modique et sa large gamme d'opérations, cet outil se présente comme la solution parfaite pour la gestion de petites applications.

De plus, l'éventail étendu de tailles et de tensions de CPU fournit la flexibilité requise pour apporter une solution à un problème d'automatisation. Un contrôleur S7-200 programmable se compose d'un CPU S7-200, qui peut être enrichi de divers modules d'extension optionnels raccordés à l'aide d'un connecteur de bus fourni avec chaque module additionnel [11].



Figure I. 13: SIMATIC S7-200 [11].

I.4.2 SIMATIC S7 -300

La gamme S7-300 (voir la figure I.14) se compose d'automates programmables modulaires destinés à des applications de niveau, et ils ont la capacité d'être interconnectés via un câble-bus PROFIBUS. Un automate S7-300 se compose d'un CPU, d'un module d'alimentation SP, d'un module de signaux FM, d'un module de signaux SM et d'un processeur de communication [11].



Figure I. 14: SIMATIC S7-300 [11].

I.4.3 SIMATIC S7-400

Le S7-400 est un automate programmable industriel conçu à la fois pour l'industrie manufacturière et le génie des procédés (voir la figure I.15). Il se distingue grâce à ses hautes performances, sa puissance de communication et ses grandes capacités de mémoire. Cet

automate programmable industriel peut s'adapter à toutes les applications spécifiques telles que la disponibilité élevée et la sécurité [12].



Figure I. 15: SIMATIC S7-400 [12].

I.4.4 SIMATIC S7-1200

L'automate SIMATIC S7-1200 (voir la figure I.16) de siemens est un dispositif modulaire et compact, adaptable, conçu pour des tâches d'automatisation simples mais d'une précision remarquable.

Il constitue ainsi une solution fiable et une réponse idéale à une vaste gamme d'applications. Cet automate, grâce à son architecture modulaire et adaptable, son interface de communication respectant les normes les plus strictes de l'industrie et sa vaste gamme de fonctionnalités technologiques avancées et intégrées, constitue un élément essentiel d'une solution d'automatisation globale [13].



Figure I. 16: Automate programmable S7-1200 [13].

I.4.5 SIMATIC S7-1500

Le processeur central (CPU) représente le cœur du contrôleur SIMATIC S7-1500. Il exécute le programme utilisateur et connectent le contrôleur à d'autres éléments d'automatisation.

Les CPU du SIMATIC S7-1500 (voir la figure I.17), grâce à leurs nombreuses innovations, procurent un avantage décisif en termes de productivité et d'efficacité.
L'équipement est très compact et conforme aux normes (IP20 ou IP65/67) [14].



Figure I. 17: S7-1500 [17].

I.5 Présentation de l'automate S7-1200

Le S7-1200 est un contrôleur qui fournit la flexibilité et l'efficacité requises pour gérer une variété étendue d'équipements, afin de satisfaire vos exigences en termes d'automatisation. Sa structure réduite, sa flexibilité et son large éventail d'instructions en font un choix parfait pour le contrôle d'une multitude d'applications.

Le CPU réunit un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des (E/S) à haute vitesse pour la commande du mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de concevoir un contrôleur performant. Après le chargement de votre programme, la CPU intègre la logique requise pour gérer et contrôler les dispositifs dans votre application. Le processeur supervise les entrées et ajuste-les sorties en accord avec la logique de votre programme utilisateur, qui peut inclure des commandes booléennes, des instructions de dénombrement, des ordres de temporisation, des directives mathématiques sophistiquées ainsi que des instructions pour interagir avec d'autres dispositifs intelligents.

La CPU dispose d'un port PROFINET qui permet de communiquer via un réseau PROFINET. Il existe des modules additionnels pour la communication via les réseaux PROFIBUS, GPRS, RS485 ou RS232 [15].

I.5.1 Choix de la CPU S7-1200

Le système SIMATIC S7-1200 propose plusieurs catégories de performances pour ses unités centrales par exemple : CPU 1211 C, CPU 1212 C, CPU 1214 C et CPU 1215 C.

Ces unités peuvent être adaptées et étendues selon les exigences spécifiques de la station. Il est envisageable d'installer une carte d'extension sur chaque processeur pour augmenter le nombre de sorties TOR ou analogiques sans changer la taille de l'automate. Il est possible d'ajouter des modules d'entrée/sortie supplémentaires sur le côté droit de l'unité centrale de traitement afin d'accroître la capacité d'e/s TOR ou analogiques [13].

Fonctions de communication : l'automate doit être capable d'interagir avec les autres systèmes de contrôle (API, supervision...) et proposer des options de communication conformes à des normes standardisées (PROFIBUS**)** [13].

I.5.2 Sélection des modules d'entrées/sorties

La sélection des modules d'entrée/sortie se fait en fonction des critères suivants [13] :

➢ La nature et la valeur de la tension entrante ou sortante.

➢ Le nombre des entries /sorties.

➢ Le type d'entrée ou de sortie (sonde, thermocouple, électrovanne...).

I.5.3 Options pour l'extension de la CPU S7-1200

La série S7-1200 propose une variété de modules et de cartes enfichables pour étendre les fonctionnalités du CPU grâce à des entrées/sorties additionnelles ou d'autres protocoles de communication [16].

Dotée d'une alimentation de 24V intégrée et d'entrées/sorties numériques intégrées, la CPU du S7-1200 est opérationnelle, sans nécessiter de composants additionnels (voir la figure I.18).

La CPU est dotée d'un port TCP/IP intégré pour établir une communication avec l'appareil de programmation. Grâce à un réseau ETHERNET, la CPU a la capacité d'interagir avec des dispositifs de commande HMI ainsi qu'avec d'autres unités centrales de traitement [17].

1. Alimentation à 24V.
2. Logement pour carte mémoire sous le volet supérieur.
3. Connecteurs amovibles pour le câblage utilisateur (derrière les volets).
4. Led d'état pour les E/S intégrées.

5. Connecteur PROFINET (sur la face inférieure de la CPU).

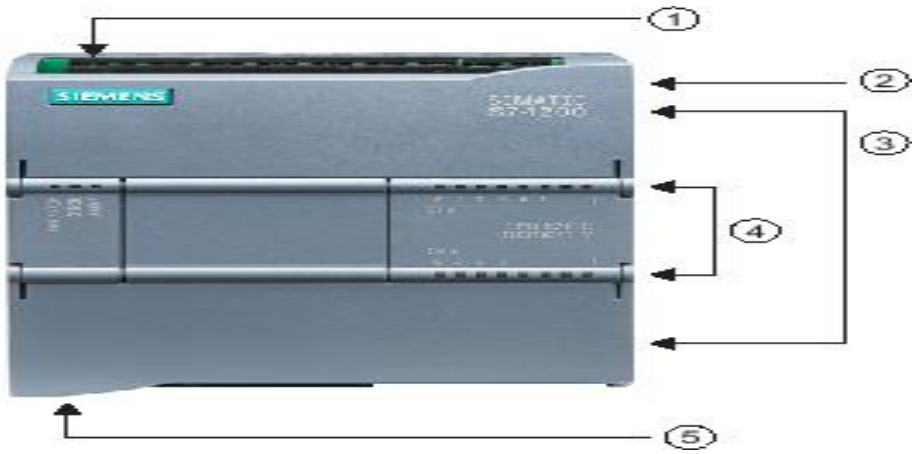


Figure I. 18: Le CPU S7 1200 [17].

I.5.4 Communication et les réseaux industriels

Pour interagir avec les agents d'exploitation, il est nécessaire d'examiner la connexion de l'automate programmable industriel avec divers matériels à processeur ainsi que les fonctions qu'il doit remplir dans son environnement. Il y a différents bus de terrain industriels qui peuvent être universels ou spécifiques à une entreprise (le bus est alors propre à cette dernière). On distingue certains protocoles de communication : Mode bus, BUSCAN, UNITELWAY, CANOPEN, PROFIBUS, PROFINET, MPI, etc….

- **PROFIBUS**

  PROFIBUS (Processus Field Bus) est le bus de terrain ouvert qui connaît un grand succès dans un large domaine d'application pour une production efficace et économique. PROFIBUS est un réseau en bus pour la communication de processus et de terrain dans les réseaux de cellules mettant en œuvre un petit nombre de stations et des appareils de terrain ainsi que pour la communication de données conforme à CEI 61158/EN 50 170 [18].

- **PROFINET**

  PROFINET est le standard Industriel Ethernet ouvert de PROFIBUS International pour l'automatisation industrielle. PROFINET permet de coupler des appareils du niveau terrain jusqu'au niveau conduite. Il garantit une communication fluide, autorise une ingénierie à l'échelle de l'installation et utilise les standards IT jusqu'au niveau terrain [19].

I.5.5 Différence entre PROFIBUS et PROFINET

PROFIBUS est un bus de terrain série classique, souvent caractérisé par des câbles RS-485 à paire unique de couleur pourpre. Il opère à une vitesse de 12 mégabits par seconde et supporte une communication de type maître/esclave. D'autre part, PROFINET est une norme d'ETHERNET industriel, typiquement caractérisée par des câbles ETHERNET industriels de

couleur verte, opérant à une vitesse de 100 mégabits par seconde et supportant une communication de type fournisseur/client [18].

Le choix entre PROFIBUS et PROFINET dépend des besoins spécifiques d'une installation. PROFINET, avec sa communication plus rapide et flexible, est généralement préféré pour les applications nécessitant une grande bande passante et une intégration facile avec les réseaux IP existants. D'autre part, PROFIBUS peut être plus approprié pour les installations existantes avec des besoins de communication série [18].

I.6 Logiciel TIA portal

I.6.1 TIA Portal V16

Le nouvel espace de travail de Siemens, « Totally Integrated Automation Portal », facilite la mise en œuvre des solutions d'automatisation grâce à un système d'ingénierie intégré qui inclut les logiciels SIMATIC STEP7 V16 et SIMATIC WINCC V16 [20].

La figure I.19 montre l'interface de lancement de TIA Portal V16.



Figure I. 19: L'interface de lancement de TIA Portal V16

I.6.2 Vue de Portail

Lors du démarrage de TIA Portal, l'environnement de travail se divise en deux perspectives principales : l'interface du portail : elle se concentre sur les missions à réaliser et son utilisation est rapide à assimiler (voir la figure I.20).

Perspective du projet : elle inclut une structure arborescente des divers composants du projet, ainsi que les éditeurs nécessaires.

Ces perspectives s'ouvrent en fonction des tâches à accomplir : Il est possible de visualiser les données, les paramètres et les éditeurs en une seule interface [20].

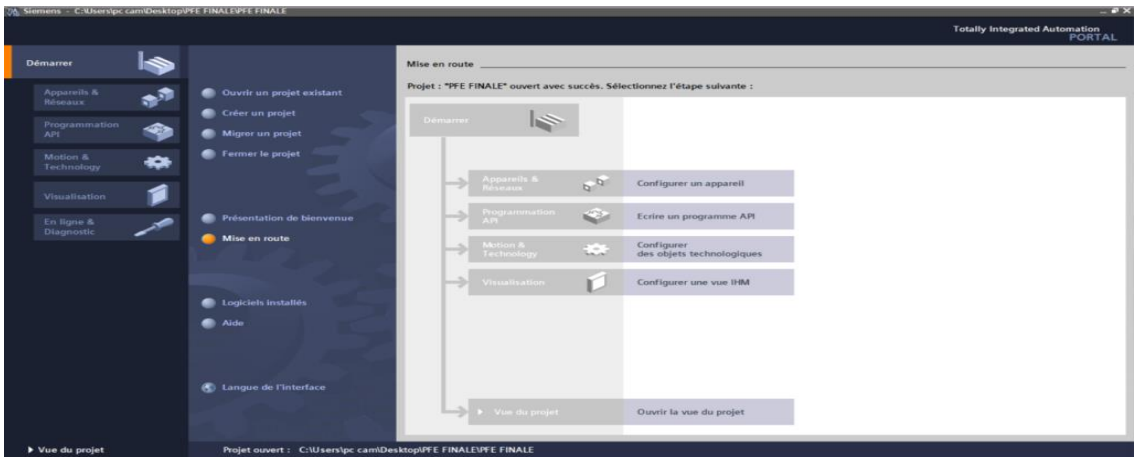


Figure I. 20: Vue de portail de TIA Portal

I.6.3 Vue de projet

La perspective du projet offre une représentation organisée de tous les composants du projet. Conformément à la norme, la barre de menu qui inclut les fonctions principales est positionnée en haut de la fenêtre. Le navigateur du projet ainsi que tous ses composants se trouvent sur la gauche. Tandis que les menus dédiés aux diverses opérations (incluant les manuels et les bibliothèques par exemple) sont disposés à droite (voir la figure I.21).

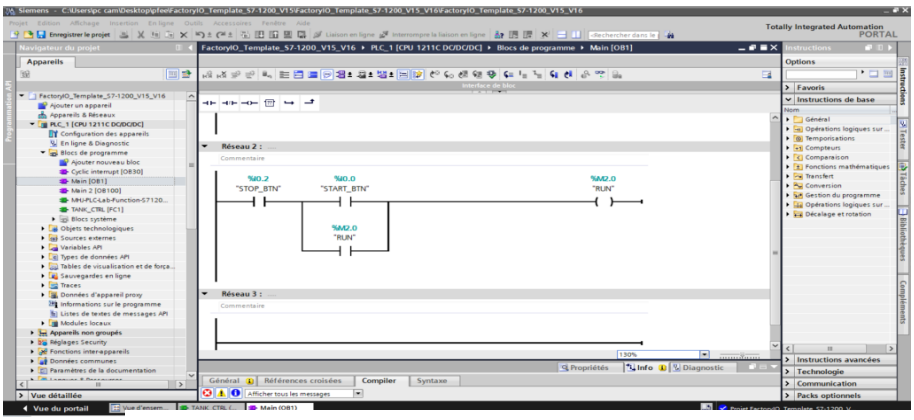


Figure I. 21: Vue de projet de tai portal

I.6.4 Blocs

Pour exécuter la tâche d'automatisation, il est nécessaire de charger dans l'automate les blocs qui renferment les divers programmes et les informations (voir la figureI.23).

Les types de blocs utilisés comprennent OB, FB, FC. Ils renferment les programmes ainsi que les blocs de données d'instance DB et globaux DB qui contenant les paramètres du

programme. Les blocs d'organisation cyclique utilisé est OB1 et OB100, ainsi que d'autres blocs DB et FC [20].

I.6.4.1 Blocs d'organisation

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation de l'automate (CPU) et le programme utilisateur (voir la figure I.22).

Ils sont appelés par le système d'exploitation et gèrent les opérations suivantes :

– Traitement cyclique du programme (par ex. OB1)

– Comportement au démarrage de l'automate.

– Traitement du programme déclenché par une alarme.

– Traitement des erreurs.

Un projet doit contenir au moins un bloc d'organisation pour le traitement cyclique du programme**.** Des priorités sont définies pour les différents OB afin qu'OB1 cyclique puisse par exemple être interrompu par un OB82 pour le traitement des erreurs [21].

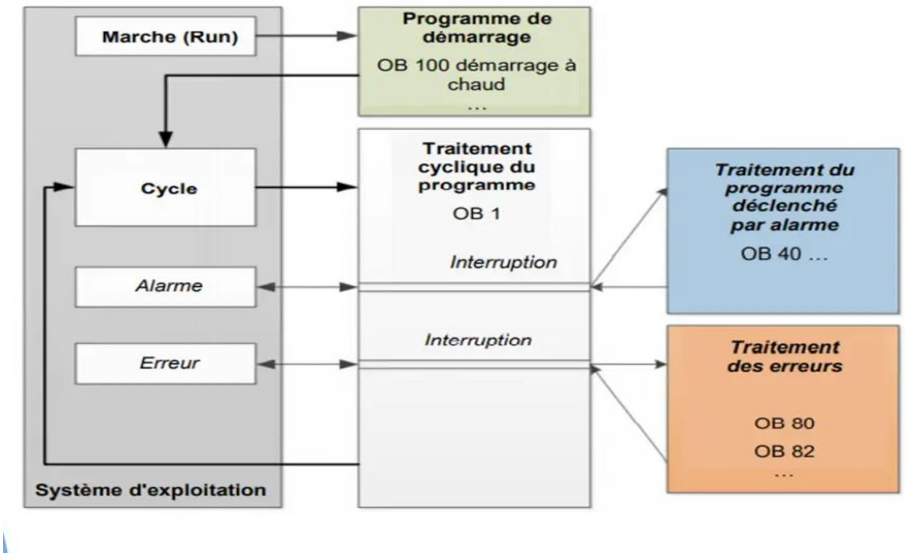


Figure I. 22: Organisation des blocs OB dans un automate siemens [21].

I.6.4.2 Blocs fonctionnels (FB)

Les blocs fonctionnels sont des blocs de code qui mémorisent durablement leurs paramètres d'entrée, de sortie et d'entrée/sortie dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement des blocs. C'est pourquoi ils sont également appelés blocs avec mémoire.

Les blocs fonctionnels peuvent aussi travailler avec des variables temporaires. Cependant, les variables temporaires ne sont pas enregistrées dans la DB d'instance elles sont disponibles uniquement pendant un cycle [20].

I.6.4.3 Bloc fonction (FC)

Le FC contient des séquences d'instructions fréquemment utilisées. Contrairement au FB, il ne possède pas de mémoire propre.

Ses variables temporaires sont enregistrées dans la pile de données locales. Il peut néanmoins accéder à des blocs de données globaux pour stocker des données nécessaires [20].

I.6.4.4 Blocs de données (DB)

Les blocs de données permettent de conserver les informations nécessaires au programme. On distingue deux types [20] :

- **Blocs de données globaux :** Ils regroupent des données accessibles depuis l'ensemble des blocs logiques (OB, FB, FC) du programme utilisateur. Ils ne sont pas liés à un bloc en particulier.
- **Blocs de données d'instance** : Ils sont systématiquement liés à un bloc fonctionnel (FB). Les informations contenues dans ce type de (DB) ne doivent être utilisées par le FB auquel il est rattaché.

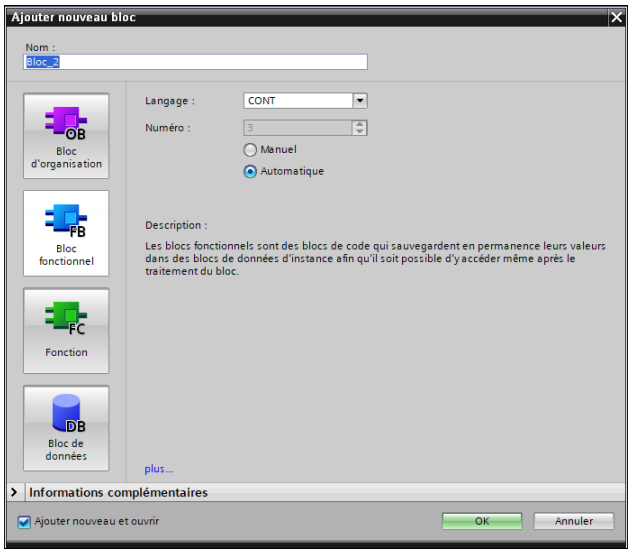


Figure I. 23: Les blocs

I.7 S7-PLCSIM V16

S7-PLCSIM permet le débogage et la validation d'un seul programme d'automate sans avoir recours à un équipement physique.

S7-PLCSIM vous offre la possibilité d'exploiter tous les outils de débogage de STEP 7, tels que la table de surveillance, le statut du programme, les fonctions en ligne et de diagnostic,

ainsi que divers d'autres outils. S7-PLCSIM propose aussi ses propres outils, notamment une table SIM, un éditeur de séquence, un éditeur d'événements et un système de balayage, S7-PLCSIM fonctionne avec STEP7 dans le portail TIA [20].

I.8 Supervision des systèmes automatisés

I.8.1 Définition de la supervision

La supervision industrielle est le processus par lequel un superviseur guide, contrôle et évalue le travail des employés dans un environnement de production. Le rôle principale de supervision est assurer la réalisation des objectifs de production tout en maintenant la qualité et la sécurité [22].

I.8.2 Importance de la Supervision [22]

- **Performance améliorée** : Une supervision efficace peut augmenter la productivité des employés.
- **Qualité du produit** : Garantit que les normes de qualité sont respectées.
- **Sécurité au travail** : Surveille les pratiques de sécurité pour prévenir les accidents.

I.8.3 Techniques de supervision

La supervision est d'un niveau supérieur et qui superpose à la boucle de commande, elle assure les conditions d'opérations pour lesquelles les algorithmes d'estimation et de commande ont été conçus. Parmi les taches principales de la supervision se trouve la acquisition de données la surveillance, l'aide à la décision, le diagnostic et la détection.

- **Acquisition de données** : est la première étape de la supervision, tel que, elle consiste à recueillir, à valider et à assurer l'acheminement des informations sur l'état du système jusqu'au poste de pilotage, cette tâche, est exécutée sans interruption et à chaque instant, ces opérations impliquent l'utilisation des capteurs permettant de mesurer les différentes variables du processus. Ces informations seront utilisées dans des relations de résidus pour accomplir l'étape de détection [23].
- **La surveillance** : La surveillance utilise les données provenant du système pour représenter l'état de fonctionnement puis en détecter les évolutions. La surveillance intervient en phase d'exploitation bien qu'elle soit prise en compte dès la phase de conception elle sert à filtrer les signaux et les événements issus du procédé et de la commande afin d'établir l'état du système. En fonctionnement normal, elle communique des comptes rendus filtrés à la commande. Elle permet également de détecter et diagnostiquer les fautes et les erreurs dans le système. En cas de dysfonctionnement, elle en informe le module de maintenance et le module de supervision.

IL existe deux types de surveillance : la surveillance du système opérant et la surveillance de la commande.

La surveillance du système opérant se décompose elle-même en deux types de surveillance : la surveillance curative et la surveillance prédictive [23].

- **Diagnostic :** Cette étape consiste à partir des défauts détectés, de localiser l'élément défaillant et d'identifier la cause qui a provoqué ce défaut. On appelle signature d'un défaut l'effet de celui-ci sur un ou plusieurs résidus. Si l'on dispose de la connaissance de la signature des défauts, il est possible, à partir de celle-ci, de remonter des effets aux causes, c'est-à-dire partir des résidus non nuls de remonter aux éléments défaillants. Cette étape nécessite donc un modèle de défauts du système permettant la résolution du problème inverse [23].
- **Aide à la décision :** Dans quelques situations même l'opérateur le plus expérimenté et le plus qualifié commet des erreurs en prenant des décisions inappropriées, cette situation est due à la fatigue de l'opérateur, au nombre impressionnant d'alarmes qui surviennent en même temps et que l'opérateur est incapable de gérer ou tout simplement par une mauvaise interprétation de sa part. L'aide à la décision consiste à aider l'opérateur à prendre la bonne décision devant toute situation, et cela en proposant une liste d'actions qui pourraient restaurer les grandeurs optimales du système.

  Dans un système d'aide à la décision, l'opérateur est toujours maître de la situation, car le système d'aide à la décision n'agit jamais. Il informe et conseille seulement, de ce fait le système ne calcule pas une valeur précise, mais propose plutôt quel moyen d'action doit être exécuté, par exemple, il pourrait conseiller de changer un point de référence d'une boucle de commande parce qu'un capteur a une dérive ou de changer la commande manuelle parce que le régulateur travaille hors de son domaine de stabilité.

  Mais il n'est pas censé trouver le point de réglage ou la valeur précise de l'action, ceci a besoin d'une étude plus soigneuse de validation, qui est toujours un problème de recherche [23].
- **Maintenance :** La maintenance est l'étape qui intervient généralement après l'étape de prise de décision elle consiste à maintenir ou à restaurer les performances des composants ou du système d'une façon globale. pour l'accomplissement de sa tache requise, ces activités sont une combinaison d'activités techniques, administratives et de gestion. Lorsqu'au cours d'une tâche préventive un composant interne du matériel est trouvé ou jugé défaillant, sa réparation ou son remplacement doit être considéré comme de la maintenance corrective. S'il est trouvé non défaillant mais dégradé, même au-delà

de la valeur de défaillance potentielle, sa réparation ou son remplacement est de nature préventive [23].

I.8.4 Supervision dans un environnement SCADA

I.8.4.1 Définition du SCADA

La Supervision, le Contrôle et l'Acquisition de Données (SCADA) est un système informatisé largement utilisé dans diverses industries pour surveiller et gérer des processus, des systèmes et des équipements complexes. Cette technologie joue un rôle essentiel dans la collecte de données en temps réel, l'automatisation des opérations, la prise de décision et la gestion de systèmes industriels. Au fil des décennies, les systèmes SCADA ont évolué pour devenir des outils clés dans la gestion de l'infrastructure critique, allant des centrales électriques aux usines chimiques en passant par les réseaux de distribution d'eau [24].

I.8.4.2 Eléments du système SCADA

Le système SCADA est principalement composé des éléments suivants [24] :

- **Site central :** Il s'agit de la station de contrôle qui gère l'ensemble du système SCADA. Il fournit généralement à l'utilisateur une interface pour afficher les informations et contrôler les sites distants.
- **RTU** (REMOTE TERMINAL UNITE) **:** Unités de terrain ces dispositifs sont installés sur le terrain pour collecter des données à partir de capteurs et d'instruments (voir la figure I.24). Ils transmettent ensuite ces données au système SCADA centrale [30].

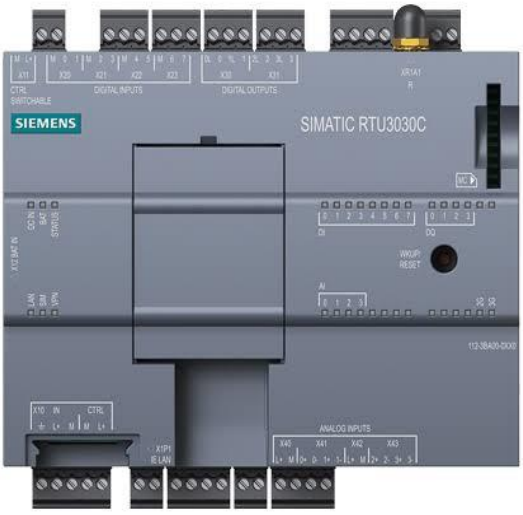


Figure I. 24: SIMATIC RTU3030C [39]

- **MTU (Master Terminal Unit) :** Cette unité est le cerveau du système SCADA .Elle reçoit, traite et stocke les données collectées des RTU, et permet aux opérateurs de visualiser les informations en temps réel [24]

- **Human Machine Interface (HMI)** :C'est un dispositif qui affiche les données de processus à surveiller par un opérateur humain. Il est utilisé en conjonction avec le logiciel et les bases de données du système SCADA pour fournir des informations de gestion, y compris les procédures de maintenance planifiée, des schémas détaillés, des informations logistiques et des données de diagnostic pour un capteur ou une machine spécifique. Les systèmes HMI permettent aux opérateurs de visualiser des informations sous forme graphique. HMI est une composante critique qui sert de point d'interaction entre les opérateurs humains et le processus industriel à surveiller et contrôler. L'HMI fournit un affichage clair et simple de diverses données du système (états d'entrée/ sortie, alarmes, etc.) en temps réel [24].
- **Système de communication :** Les données circulent entre les RTU, la MTU et HMI via un réseau de communication, généralement basé sur des protocoles spécifiques [30].

I.9 Logiciel de supervision WINCC

WINCC (Windows Control Center), est le logiciel qui permet de créer une Interface Homme Machine (HMI) graphique, qui assure la visualisation et le diagnostic du procédé. Il permet la saisie, l'affichage et l'archivage des données (voir la figure I.25), tout en facilitant les tâches de conduite et de surveillance aux exploitants. Il offre une bonne solution de supervision, car il met à la disposition de l'opérateur des fonctionnalités adaptées aux exigences d'une installation industrielle.

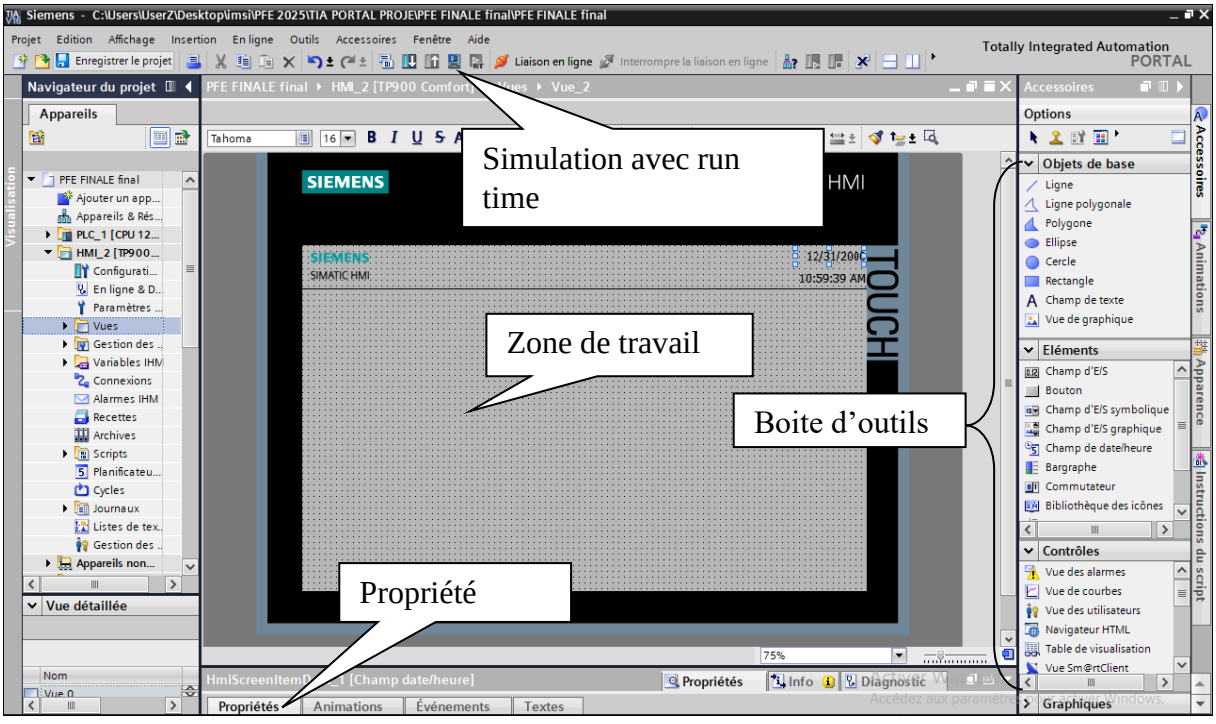


Figure I. 25: Présentation de l'enveniment HMI avec TIA portal

I.10 Conclusion

Dans ce chapitre a établi les bases nécessaires pour évaluer les systèmes automatisés et les automates programmables industriels comme éléments clés de l'automatisation moderne. L'analyse de l'API en général et de la gamme SIMATIC S7 de Siemens en particulier met en évidence l'adéquation de ces systèmes sur des échelles d'automatisation variées en raison de leurs architectures compactes ou modulaires. L'utilisation normalisée des langages de programmation et l'incorporation d'outils de supervision comme TIA Portal et WinCC rendent ces API d'une flexibilité inégalée, ils héritent d'une robustesse, d'une précision et d'une fiabilité considérables dans des environnements industriels névralgiques.

L'interaction évolutive avec des systèmes SCADA et des réseaux industriels comme le PROFIBUS et le PROFINET, un dialogue transparent est possible pour une interopérabilité avancée.

CHAPITRE II : Généralité sur la Régulation

II.1 Introduction

Les systèmes de contrôle et de régulation représentent les fondamentaux essentiels des processus et des équipements d'automatisation moderne et ce dans l'ensemble des secteurs industriels. Dans la plupart des cas, les systèmes de régulation occupent une place d'honneur quel que soit l'industrie : chimique, énergétique, agroalimentaire, en particulier. Il est indispensable pour assurer la stabilité du processus et l'efficacité de l'équipement.

Dans ce chapitre, nous aborderons les principes fondamentaux de la régulation, en essayant de comprendre son fonctionnement, ses différents types, ainsi que les principaux composants sur lesquels elle s'appuie afin d'assurer un contrôle efficace des différentes grandeurs physiques.

II.2 Définition

La régulation des procédés industriels désigne l'ensemble des dispositifs matériels et techniques utilisés pour maintenir une grandeur physique à régler, égale à une valeur désirée, appelé consigne. Lorsque des perturbations ou des changements de consigne se produisent, la régulation provoque une action correctrice sur une grandeur physique du procédé appelée grandeur réglant [25].

II.3 Objectif de la régulation

L'objectif d'un système régulation ou d'un asservissement est de garantir le bon fonctionnement d'un procédé en respectant les critères définis par un cahier des charges.

Le cahier des charges définit des critères qualitatifs à imposer qui sont traduits le plus souvent par des critères quantitatifs, comme par exemple, de stabilité, de précision, de rapidité ou de lois d'évolution. Voici quelques exemples d'objectifs qualitatifs [26] :

- Obtenir une combustion air-gaz correcte dans un brûleur.
- Maintenir une qualité constante d'un mélange de produits.
- Obtenir un débit de fluide constant dans une conduite en fonction des besoins.
- Faire évoluer une température d'un four selon un profil déterminé.

II.4 Asservissement et régulation

Lorsqu'il y a un retour d'information de la grandeur observée sur le régulateur, on parle d'un asservissement du système ou d'une régulation du système [26].

II.4.1 Asservissement

La consigne, traduisant l'objectif désiré du procédé, n'est pas constantes et les grandeurs perturbatrices n'existent pas ou sont très peu influentes sur la grandeur à maîtriser [26].

II.4.2 Régulation

La consigne, traduisant l'objectif désiré du procédé, est constante et les grandeurs perturbatrices influencent fortement la grandeur à maîtriser [26].

II.5 Concept de boucle ouverte/fermée

II.5.1 Boucle fermée

On parle de boucle fermée lorsque la régulation agit directement sur la valeur mesurée (voir la figure II.1). Pour cela, la valeur à régler doit être mesurée et renvoyée en retour à l'entrée du régulateur [27].

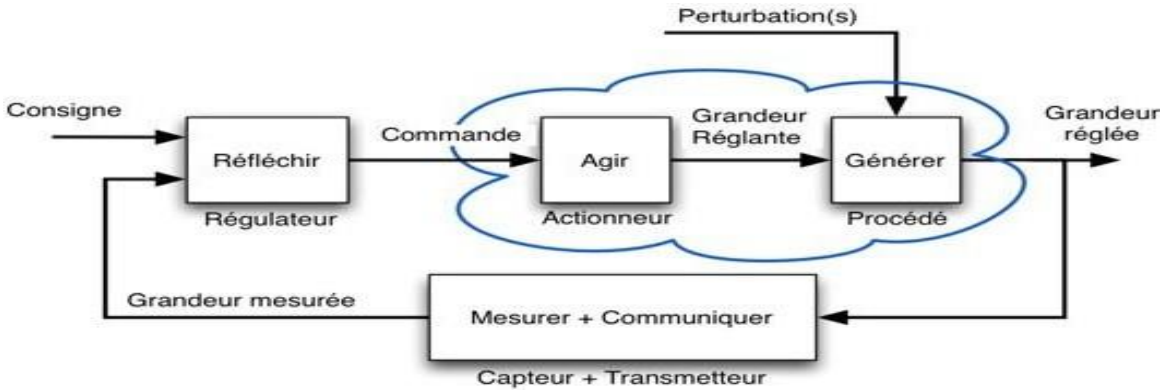


Figure II. 1: Système en boucle fermée (BF) [33].

II.5.2 Boucle ouverte

On parle de boucle ouverte lorsqu'on ne mesure pas la valeur sur laquelle on agit (voir la figure II.2). Le régulateur n'a donc aucune information de résultat sur son action et ne peut donc pas la corriger [27].

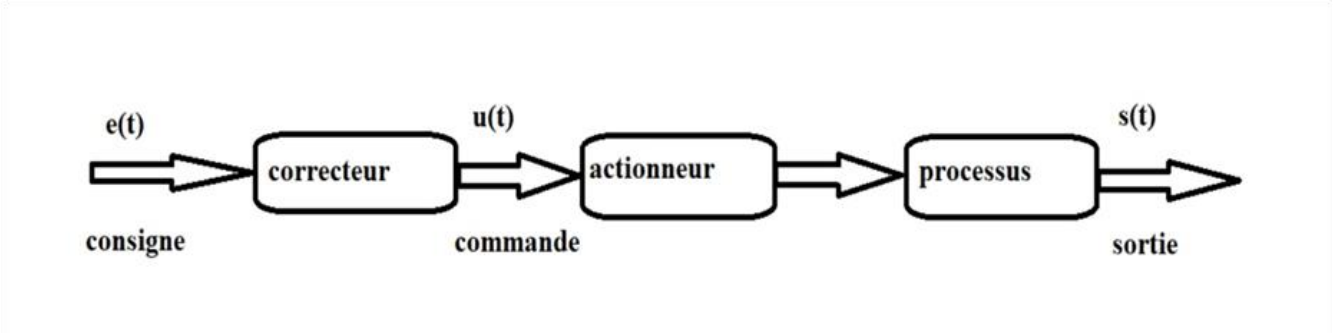


Figure II. 2: Système en boucle ouverte (BO) [33].

II.6 Constituants d'une chaine de régulation

En général, on peut représenter une boucle de régulation de la façon suivante (voire la figure II.3) :

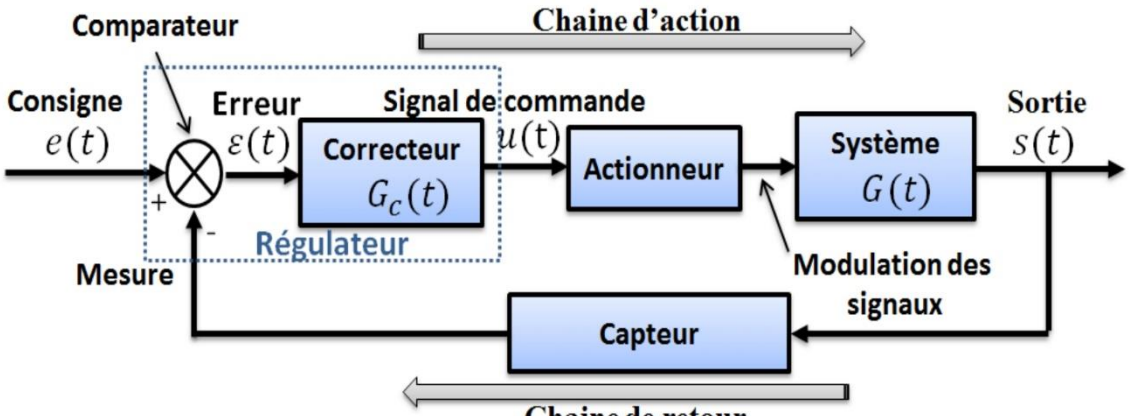


Figure II. 3: Représentation de schéma fonctionnelle d'une boucle de régulation [28].

Cette structure organisationnelle constitue le fondement que l'on retrouve dans tous les systèmes régulés ou contrôlés. Elle implique deux chaînes, une chaîne d'action et une chaîne de retour ou d'information.

La chaîne d'action comprend tous les éléments de puissance (requérant une source d'énergie externe) et qui effectue le travail.

La chaine de retour ou de mesure : Évalue et quantifie le travail accompli, puis communique au comparateur une valeur physique proportionnelle à ce travail. Elle intègre habituellement un capteur qui fournit une évaluation de la dimension, laquelle est par la suite amplifiée et convertie avant son utilisation [28].

- **Grandeur réglée :** La grandeur réglée est l'élément physique que l'on souhaite contrôler. Elle est à l'origine du nom donné à la régulation : régulation de température, niveau, pression, débit, vitesse [29].
- **Consigne (Set Point) :** La valeur que doit avoir la mesure ajustée. La directive peut consister en une transformation via un potentiomètre, des touches tactiles, etc. de l'intervention manuelle d'un opérateur humain [29].
- **Grandeur mesurée** : Grandeur mesurée donnée par le capteur-transmetteur et mise en comparaison avec la valeur de consigne [29].
- **Régulateur :** Le régulateur est l'élément central d'une régulation. Il évalue l'erreur calculée par le comparateur, c'est-à-dire l'écart entre la sortie et la consigne, et en

déduit à partir de celle–ci une valeur régulée «commande» ou valeur de correction à transmettre au procédé, afin de corriger la sortie [29].

- **Actionneur :** L'actionneur peut être considéré comme l'organe exécutif de la régulation. C'est dispositif mécanique, électrique, pneumatique ou hydraulique permettant d'agir sur une machine, un système pour modifier son fonctionnement ou son état [29].
- **Système :** Le système constitue cœur du système régulé, ou encore son élément initial. C'est aussi la partie qui agit directement la valeur de sortie [29].
- **Capteur** : Dispositif qui produit, à partir d'une grandeur physique, une autre grandeur généralement électrique, dépendant de la première et immédiatement exploitable pour la mesure ou le contrôle [29].
- **Perturbations :** Grandeurs physiques qui influencent la grandeur réglée. Elles ne sont pas toujours mesurables. Exemples : température extérieure, débit de soutirage, couple résistant [29].

II.7 performances d'un système

La figure II.4 illustre les performances des systèmes réglés.

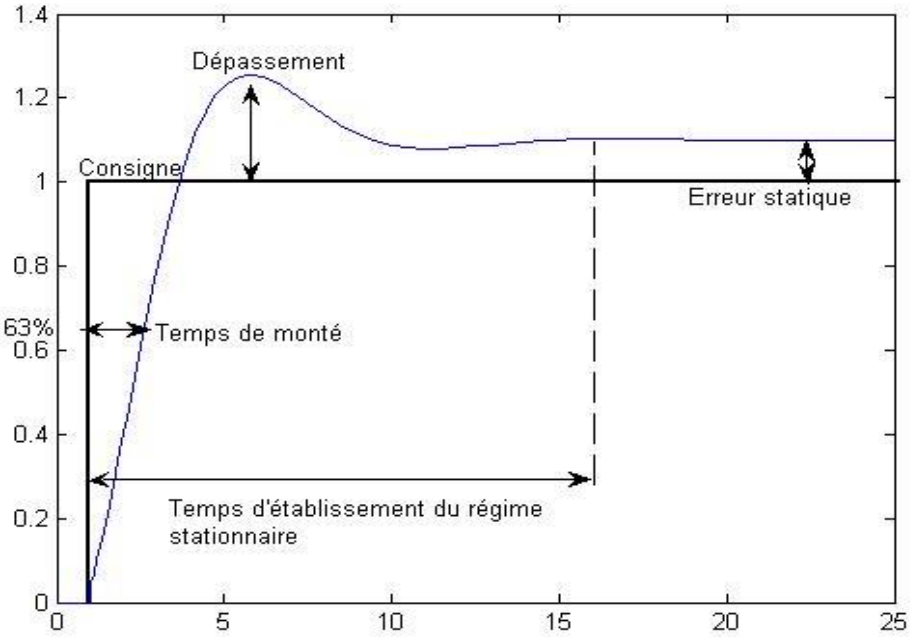


Figure II. 4: Performances d'un système de commande [33].

II.7.1 Stabilité

Pour une consigne ou référence l'entrée e(t) constante, la sortie s(t) du système doit tendre vers une valeur constante lorsque $t \longrightarrow +\infty$, apparition de n'importe quelle perturbation doit être rejetée au cours du temps (voir la figure II.5) [28].

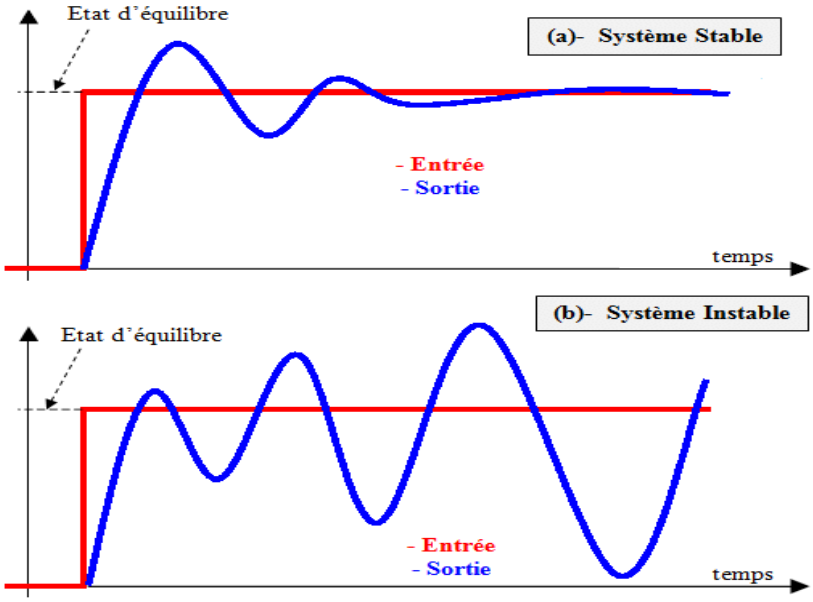


Figure II. 5: Stabilité du système [33].

II.7.2 Rapidité

Elle exprime pratiquement la durée transitoire (voir la figure II.6). Elle se manifeste précisément par le délai de réponse Te ou temps d'établissement, qui correspond au laps de temps requis par la mesure pour atteindre sa valeur finale à ± 5% de variation et maintenir cette fourchette des ± 5% [33]. Rapidité = temps de réponse Te

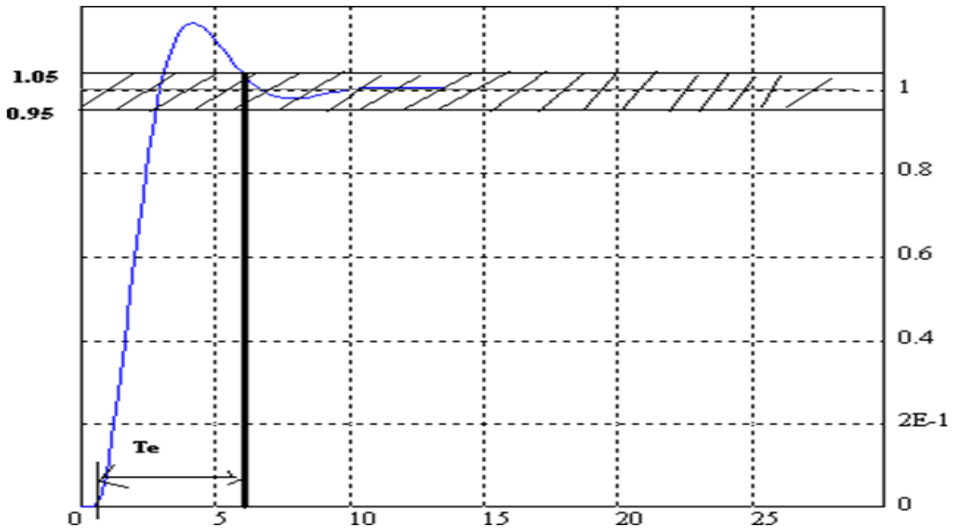


Figure II. 6: Rapidité du système [33].

II.7.3 Précision

Elle est déterminée à partir de l'erreur statique ε en régime stable, comme l'illustre (voir la figure II.7) [33].

$$\text{Erreur de précision (\%)} = (\varepsilon/C).100 \qquad \text{II.1}$$

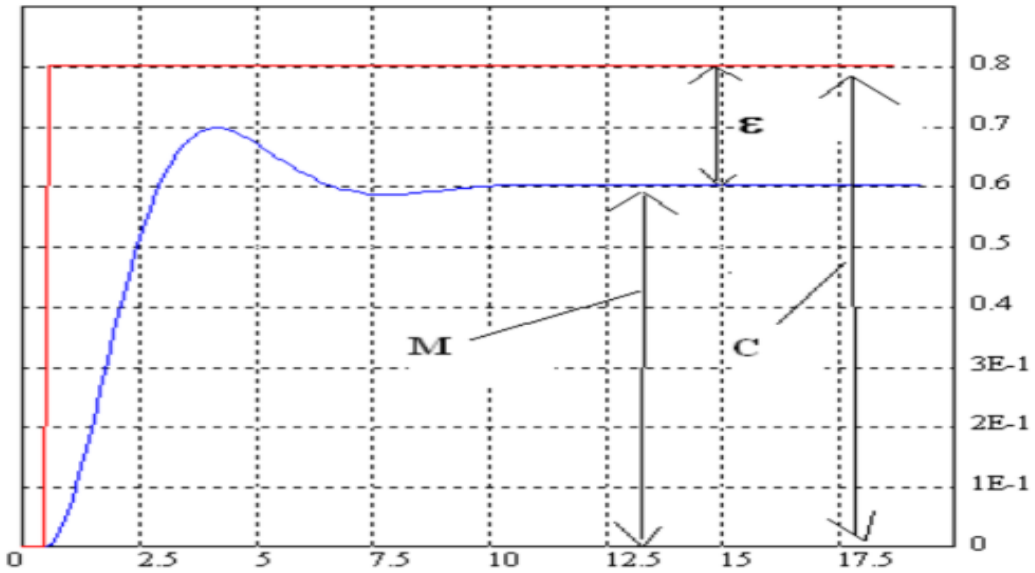


Figure II. 7: Précision du système [33].

II.7.4 Dépassement

En général, on recommande un système de régulation dont le régime transitoire soit bien amorti et dont le dépassement ne dépasse pas 5% à 10% de la valeur nominal (voir la figure II.8). Un système asservi idéal est stable, précis et rapide. Cependant, il faut faire un compromis entre ces critères pour un système en boucle fermée [33].

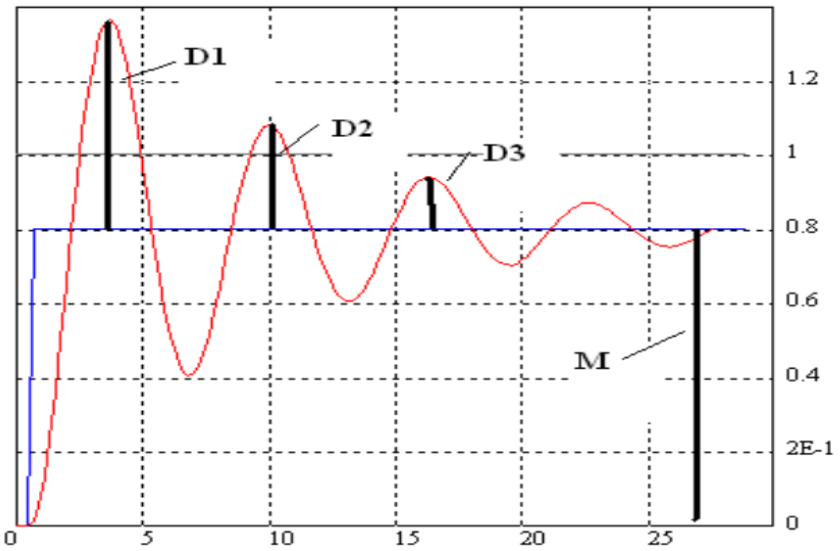


Figure II. 8: Dépassement du système [33].

D1, D2 sont exprimés par les mêmes unités (mm, %, unité physique).

II.8 Types des régulateurs industriels

II.8.1 Régulateur Tout ou Rien (TOR)

Un régulateur «tout ou rien» est un régulateur qui élabore une action de commande discontinue qui prend deux positions ou deux états 0 et 1 (ou 0 et 100%).

Les régulateurs tout ou rien sont utilisés pour la commande des systèmes ayant une grande inertie où la précision de régulation n'est pas importante [30].

La grandeur réglée oscille autour du point de fonctionnement. A chaque dépassement des seuils de commutation, la sortie du régulateur change d'état. Compte tenu de l'inertie du système, la valeur absolue de l'erreur peut dépasser le seuil.

II.8.2 Régulateurs standards P, PI, PD, PID

II.8.2.1 Actions de base

II.8.2.1.1 Action proportionnel

Le régulateur est un amplificateur de gain réglable défini par :

$$C(S) = k_P \qquad \text{II.2}$$

Ce régulateur a pour mission de modifier le gain statique initial du système. L'influence du gain statique sur les performances :

Si $k_p < 1$, autrement dit s'il s'agit d'un atténuateur, on améliore la stabilité du système et on diminue son dépassement en boucle fermée. En revanche, la rapidité et la précision sont dégradées.

Si $k_p > 1$, on améliore la rapidité et la précision du système en boucle fermée mais on diminue la stabilité (ce qui peut aller jusqu'à rendre le système instable) et on accroît son dépassement [31].

II.8.2.1.2 Action intégral

L'action intégrale agit proportionnellement à la surface de l'écart entre la consigne et la mesure, et elle poursuit son action tant que cet écart n'est pas nul. On dit que l'action intégrale donne la précision statique (elle annule l'erreur statique). L'action intégrale est conditionnée par le temps d'intégrale Ti [35].

La fonction de transfert de l'action intégrale prend la forme générale suivante :

$$C(s) = \frac{1}{T_i\ S} \qquad \text{II.3}$$

II.8.2.1.3 Action dérivée

C'est une action qui amplifie les variations brusques de la consigne. Elle a une action opposée à l'action intégrale. Cette fonction est remplie par l'opérateur mathématique : dériver par rapport au temps. Ainsi, dans un régulateur, on définit l'action dérivé à partir du temps dérivé Td [35].

$$C(s) = T_d\, S \qquad \text{II.4}$$

II.8.2.2 Régulateur PI

L'action proportionnelle est une fonction qui fait varier le signal de sortie du régulateur proportionnellement à l'écart entre la mesure et la consigne. Le correcteur intégral est fréquemment combiné au correcteur proportionnel pour améliorer le régime permanent et le régime transitoire. Cette combinaison est appelée correcteur proportionnel-intégral (voir la figure II.9).

Le coefficient de proportionnalité, appelé gain du régulateur, est obtenu par le réglage du paramètre du régulateur [30].

La fonction de transfert de Régulateur proportionnel-intégral est :

$$C(s) = \frac{U(s)}{E(s)} = K_P\left(1 + \frac{1}{T_i\, S}\right) \qquad \text{II.5}$$

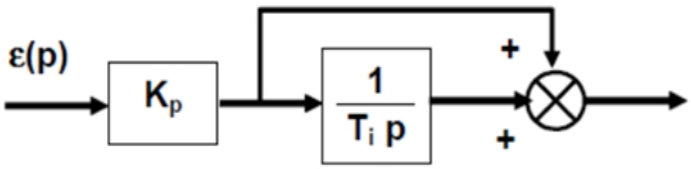


Figure II. 9: Régulateur proportionnelle-intégral [26].

II.8.2.3 Régulateur PD

En général, le régulateur ne fonctionne pas en action dérivée pure. L'inconvénient majeur du correcteur dérivé est son insensibilité aux variations lentes de l'erreur. Pour cette raison, ce type de correcteur n'est jamais utilisé seul. La solution permettant de minimiser cet inconvénient est de le combiner au correcteur proportionnel (voir la figure II.10), on donne sa fonction de transfert [30].

$$C(s) = K_P(1 + K_d\, \mathrm{S}) \qquad \text{II.6}$$

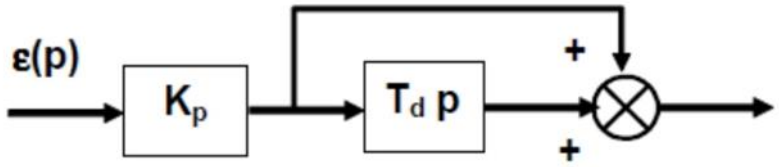


Figure II. 10: Régulateur proportionnelle-dérivée PD [26].

II.8.2.4 Régulateur PID

II.8.2.4.1 Définition

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. Ils ont l'avantage d'améliorer le comportement des systèmes, par ailleurs, ces correcteurs font évaluer l'erreur entre le signal de référence et le signal de sortie, celle-ci est corrigée par l'un des gains du correcteur PID.

Ce type de régulation permet donc d'agir simultanément sur la rapidité, la précision et la stabilité du système, ce qui en fait une solution robuste et efficace pour un large éventail d'applications industrielles [32].

II.8.2.4.2 Les différentes structures du PID

Différentes possibilités d'associations des modules P, I et D existent. Ces structures sont fonctionnellement équivalentes, et il est facile de convertir les coefficients utilisés dans l'une pour obtenir ceux d'une autre. Les trois configurations les plus utilisées sont :

- **Forme standard (forme mixte)**

Le correcteur PID standard (voir la figure II.11) génère un signal de commande conformément à l'équation :

$$U(t) = K_P\, e(t) + \frac{K_P}{Ti}\int_0^t e(\tau)d\tau + K_P\, T_d\, \frac{de(t)}{dt} \qquad \text{II.7}$$

La forme standard de la fonction de transfert d'un correcteur PID est :

$$C(s) = \frac{u(s)}{e(s)} = K_P\left(1 + \frac{1}{T_i\, S} + T_d s\right) \qquad \text{II.8}$$

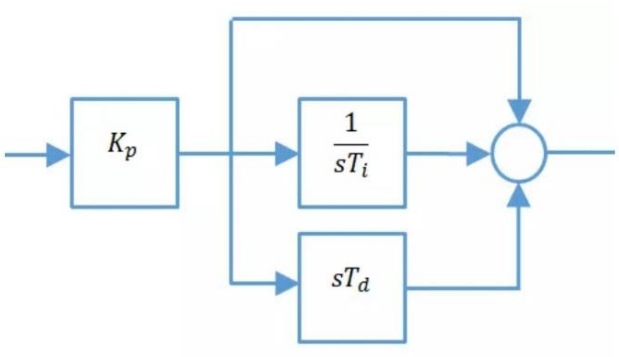


Figure II. 11: Structure mixte [26].

- **Forme parallèle**

Le correcteur PID parallèle (voir la figure II.12) génère un signal de commande conformément à l'équation :

$$U(t) = K_P\, e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \qquad \text{II.9}$$

La forme parallèle de la fonction de transfert d'un correcteur PID est :

$$C(s) = \frac{u(s)}{e(s)} = K_P + \frac{1}{T_i\, S} + T_d s \qquad \text{II.10}$$

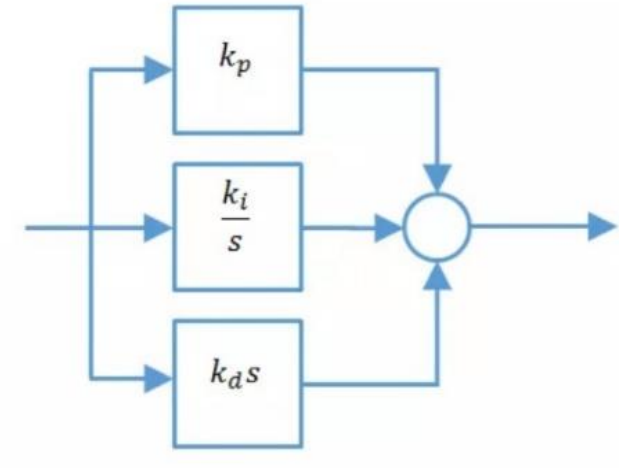


Figure II. 12: Structure parallèle [26].

- **Forme série**

Le correcteur PID série (voir la figure II.13) génère un signal de commande conformément à l'équation :

$$U(t) = K_P(1 + \frac{T_d}{T_i})\, e(t) + \frac{T_p}{T_i}) \int_0^t e(\tau)d\tau + K_p\, T_d\, \frac{de(t)}{dt} \qquad \text{II.11}$$

La forme série de la fonction de transfert d'un correcteur PID est :

$$C(s) = \frac{u(s)}{e(s)} = K_P\left(1 + \frac{1}{T_i s}\right)(1 + T_d s) \qquad \text{II.12}$$

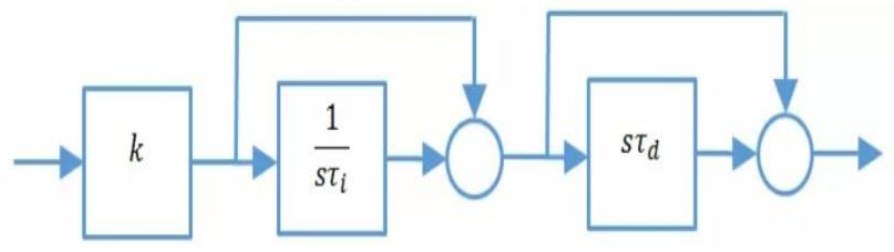


Figure II. 13: Structure série [26].

II.9 Effet des actions du régulateur PID

- **Effet du gain du régulateur K_P** : l'augmentation du gain fait diminuer l'erreur statique mais ne l'annule pas et rend le système mais on est limité par un gain critique où le système entre dans la zone d'instabilité [34].
- **Effet du temps d'intégration T_i**: la diminution de la constante de temps (augmenter l'action intégrale) annule complètement l'erreur statique et diminue aussi le temps de réponse mais augmente l'instabilité du système en boucle fermée [34].
- **Effet du temps de dérivation** : l'augmentation de la constante de dérivation peut améliorer la stabilité du système dans une certaine limite, car l'action dérivée ralentit la montée et réduit les dépassements. Elle agit sur la dérivée de l'erreur, ce qui en fait une action anticipatrice, réagissant rapidement aux variations. Cependant, un gain dérivé trop élevé rend le système instable, donc on ne peut pas l'augmenter indéfiniment. Elle est particulièrement utilisée dans les systèmes à forte inertie, notamment en régulation de température [34].

II.10 Sens d'action d'un régulateur

Un procédé est direct, quand sa sortie varie dans le même sens que son entrée (voir la figure II.14). Dans un régulateur, la mesure est considérée comme une entrée E. Dans le cas contraire, le procédé est dit inverse [35].



Figure II. 14: Sens d'action [35].

Si le procédé est DIRECT : il faut mettre le sens d'action du régulateur sur INVERSE (voire la figure II.16).

- Si le procédé est INVERSE : il faut mettre le sens d'action du régulateur sur DIRECT (voire la figureII.15).

- **Sens inverse :**



Figure II. 15: Régulateur sens inverse [35].

- **Sens direct**

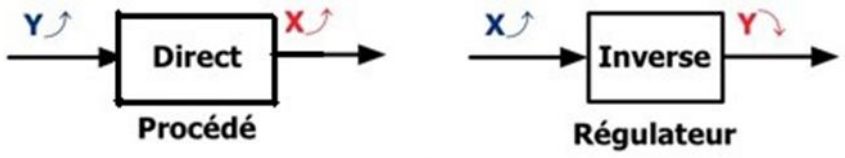


Figure II. 16: Régulateur sens direct [35].

II.11 Choix et dimensionnement de régulateur PID

II.11.1 Méthode de Ziegler-Nichols

Si un modèle mathématique du processus est disponible, des techniques de conception appropriées peuvent être appliquées pour déterminer les paramètres d'un PID, qui satisferait les spécifications du système ferme en termes de réponse transitoire et de régime permanent.

Si le processus est complexe et qu'un modèle mathématique précis n'est pas disponible, alors les méthodes analytiques ou computationnelles ne peuvent pas être appliquées, et dans ce cas

des méthodes expérimentales sont utilisées pour régler le régulateur. Le processus de sélection des paramètres PID pour obtenir la performance désirée est appelé réglage du contrôleur.

Les méthodes expérimentales les plus connues sont : les règles de Ziegler-Nichols, la première nécessite l'enregistrement de la réponse indicielle du système à régler, alors que la deuxième exige d'amener le système en boucle fermée à sa limite de stabilité. Ces méthodes ne requièrent pas de modèle explicite, tiré des lois physiques, du système à commander. Elles permettent rarement de satisfaire des spécifications sévères. En particulier, le dépassement de la valeur de consigne est souvent trop important. Il est important de souligner que ces méthodes ne s'appliquent en général qu'à des systèmes sans comportement oscillant .Ces systèmes possèdent souvent un retard pur et/ou plusieurs constantes de temps. On les rencontre surtout dans les processus physicochimiques tels que la régulation de température, de niveau, de pression [36].

II.11.1.1 Approche de Ziegler-Nichols pour les systèmes en boucle ouverte

Cette méthode est conforme aux systèmes dont la réponse indicielle en boucle ouverte peut être relevée et ne présente pas de dépassement (réponse apériodique) (voir la figure II.17).

Les correcteurs proposés ont été calculés pour convenir pour un système à commander de fonction de transfert $H(s) = \frac{ke^{-T_u S}}{1+T_a S}$ les paramètres k, T_u et T_a sont relevés sur la réponse du système réel à Commander à un échelon d'amplitude. Notons que d'autres méthodes d'identification peuvent être utilisées pour le calcul des valeurs des paramètres du modèle du 1er ordre [36].

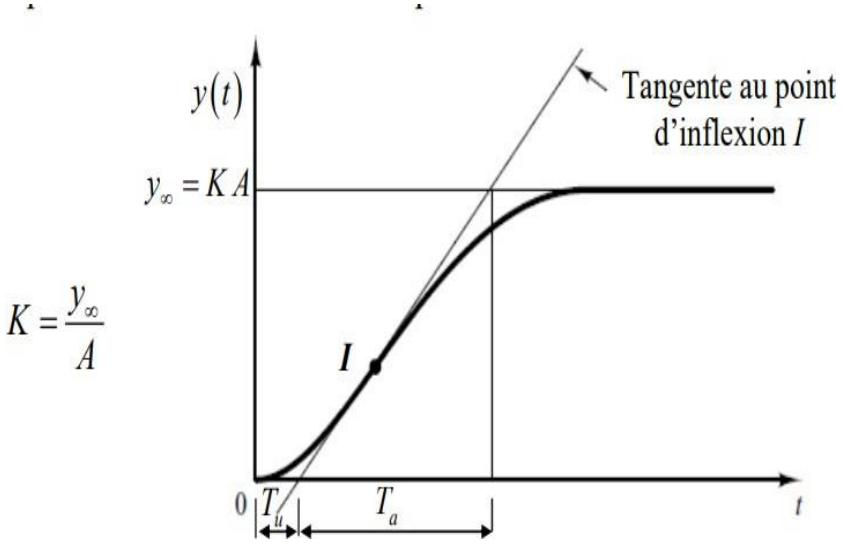


Figure II. 17: Réponse indicielle en boucle ouverte [33].

Ziegler et Nichols proposent de calculer les paramètres des correcteurs P, PI ou PID à l'aide des recommandations suivantes (voir le tableau II.1) [36] :

Paramètre du correcteur	k_p	T_i	T_d
P	$\frac{T_a}{T_u}$	∞	-
PI	$\frac{0.9T_a}{T_u}$	$3.3\ T_u$	-
PID	$\frac{1.2T_a}{T_u}$	$2T_u$	$0.5T_u$

Tableau II. 1: Méthode de Ziegler-Nichols en boucle ouverte [32].

- **Procédure** [30]
  - Obtenir la réponse indicielle en boucle ouverte du système a commandé.
  - Déterminer le temps T_u, T_a, K à partir de cette réponse indicielle.
  - Calculer les paramètres du correcteur choisi à partir du tableau.
  - Si nécessaire, ajuster les paramètres du correcteur.

II.11.1.2 Approche de Ziegler-Nichols en boucle fermée

L'essai en boucle ouverte est irréalisable pour certains systèmes pour deux raisons, parce qu'ils sont instables, ou parce que le gain en boucle ouverte est grand au point de provoquer des variations de la sortie dangereuses pour l'installation. On peut alors utiliser une méthode basée sur un essai en boucle fermée, dite méthode du pompage limite (oscillations entretenues) (voir la figure II.18).

La deuxième méthode de Ziegler-Nichols repose sur la connaissance des paramètres du point Critique K_{cr}, T_{cr}

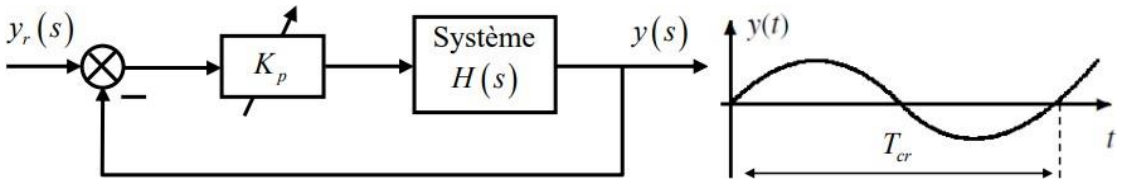


Figure II. 18: Essai d'instabilité en boucle fermée [30].

Ziegler et Nichols proposent de calculer les paramètres des correcteur P, PI ou PID à l'aide des recommandations suivantes (voir le tableau II.2) [36] :

Paramètre du correcteur	k_p	T_i	T_d
P	0.5 K_{cr}	∞	0
PI	0.45 K_{cr}	0.83 T_{cr}	0
PID	0.6 Kcr	0.5 Tcr	0.125 T_{cr}

Tableau II. 2: Méthode de Ziegler-Nichols en boucle fermée [36].

- **Procédure** [36]
  - Placer le système en boucle fermée avec un correcteur proportionnel de gainK_P.
  - Augmenter progressivement le gain K_P usqu'à obtenir une réponse périodique.
  - Relever la valeur **K_{cr}** du gain K_P ainsi atteinte, et la période Tcr des oscillations de la sortie.
  - Calculer les paramètres des correcteurs choisis à partir du tableau.
  - si nécessaire, ajuster les paramètres du correcteur.

II.12 Conclusion

Ce chapitre a porté sur les bases de la régulation automatique, l'un des piliers de l'automatisation industrielle. Les concepts directeurs comme l'asservissement et les boucles ouverte et fermée ont été définis en plus des composants d'un système de régulation. Il a aussi été question des performances visées stabilité, précision, rapidité, dépassement ainsi que des différents régulateurs employés par l'industrie, tels que les régulateurs P, PI, PD et PID. Ce dernier a été choisi en raison de sa prévalence en plus de sa grande flexibilité et performance.

Les méthodes de réglage, telles que celles de Ziegler-Nichols, sont des outils qui permettent d'optimiser le comportement du régulateur selon le processus. L'ensemble de ces notions constitue une base solide pour la compréhension et la mise en œuvre des systèmes de contrôle dans les environnements industriels complexes.

CHAPITRE III : Programmation, Simulation et Supervision

III.1Introduction

Dans ce chapitre, nous abordons à la mise en œuvre de notre système de contrôle de niveau à travers une simulation pratique. L'objectif est de concevoir une solution automatisée en utilisant les outils industriels. Pour cela, la programmation et la supervision seront réalisées à l'aide du logiciel TIA Portal, tandis que la modélisation 3D du processus a été effectuée avec le logiciel Factory I/O.

Nous traiterons les différentes étapes de conception, du cahier des charges à la simulation complète. En d'autres termes, nous effectuerons le travail de la configuration du matériel, la conception de HMI, et la programmation des divers modes de fonctionnement.

III.2 Cahier des charges

a. Contexte et objectif du projet

Dans le cadre d'un projet de simulation industrielle, ce cahier des charges définit les spécifications techniques pour la conception et la simulation d'un système automatisé de régulation du niveau d'eau dans un réservoir.

La solution sera développée à l'aide des outils suivants :

- **TIA Portal** pour la programmation de l'automate programmable (API/PLC) et la conception de l'interface de supervision HMI.
- **Factory I/O** pour la modélisation et la simulation 3D du processus.

b. Fonctions principales du système

Le système automatisé devra assurer les fonctionnalités suivantes :

- **Acquisition du niveau d'eau**

  Lecture en temps réel du niveau d'eau via un capteur analogique de niveau connecté à une entrée analogique de l'API.

- **Commande de la vanne de remplissage**

  Pilotage d'un actionneur (vanne) pour réguler le niveau d'eau en fonction de la valeur mesurée.

- **Supervision du processus**

  Affichage en temps réel sur une interface HMI des informations suivantes :

  - Niveau d'eau mesuré (affichage graphique et numérique).
  - État de la vanne (ouverte/fermée).
  - Alarmes en cas de dépassement de seuils (haut ou bas niveau).

Le système dispose de deux modes de fonctionnement, sélectionnables depuis l'interface HMI manuel et automatique.

Mode manuel : l'utilisateur contrôle directement l'ouverture de la vanne en définissant un pourcentage via l'HMI.

Mode automatique : intègre deux stratégies de contrôle : ON/OFF et PID.

Mode ON/OFF : la vanne s'ouvre lorsque le niveau descend en dessous d'un seuil bas, et se ferme à l'atteinte d'un seuil haut ; les deux seuils sont configurables depuis l'HMI.

Mode PID : un régulateur ajuste de manière continue l'ouverture de la vanne pour stabiliser le niveau d'eau autour d'une consigne (setpoint) défini par l'utilisateur.

III.3 Logiciel Factory I/O

Factory I/O est un outil de simulation d'usine en 3D pour l'apprentissage des technologies d'automatisation. Conçu pour être facile à utiliser et permet de construire rapidement des systèmes virtuels à l'aide d'une sélection de pièces industrielles courantes. Factory I/O offre également un niveau élevé de réalisme lors de la simulation. Le logiciel est compatible avec des automates physiques et virtuels, ce qui permet de réaliser le contrôle depuis l'extérieur.

III.3.1 Interface du logiciel

Le logiciel possède une interface relativement intuitive avec des menus et divers boutons pour la gestion des caméras, l'affichage de la palette, l'affichage des capteurs, l'affichage des actionneurs, la réalisation de simulations (voir la figure III.1).

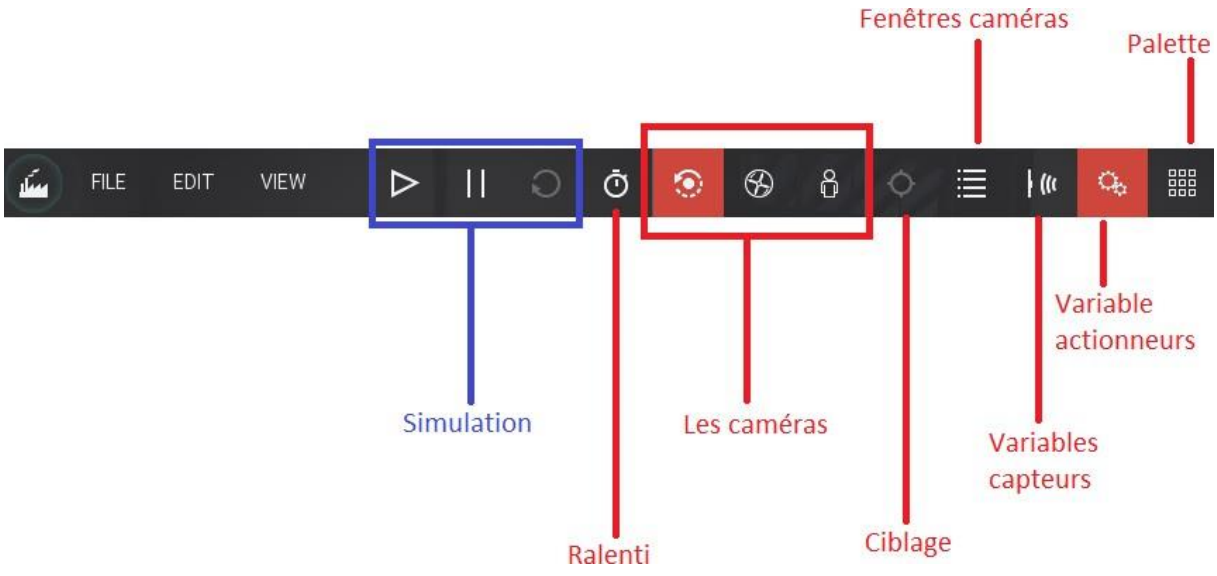


Figure III. 1: La barre des tâches

III.3.2 Choix du système simulé

Factory I/O offre une large gamme de scènes prêtes à l'emploi, inspiré d'applications réelles dans le domaine industriel (voir la figure III.2). Ces scènes comprennent divers domaines tels que l'assemblage, la classification et le contrôle de niveaux.

Le programme se caractérise par une grande flexibilité, car l'utilisateur peut :

- Choisir un scénario préexistant et le modifier selon les besoins.
- Concevoir un système personnalisé à partir de zéro en utilisant les composants disponibles dans la bibliothèque du programme.

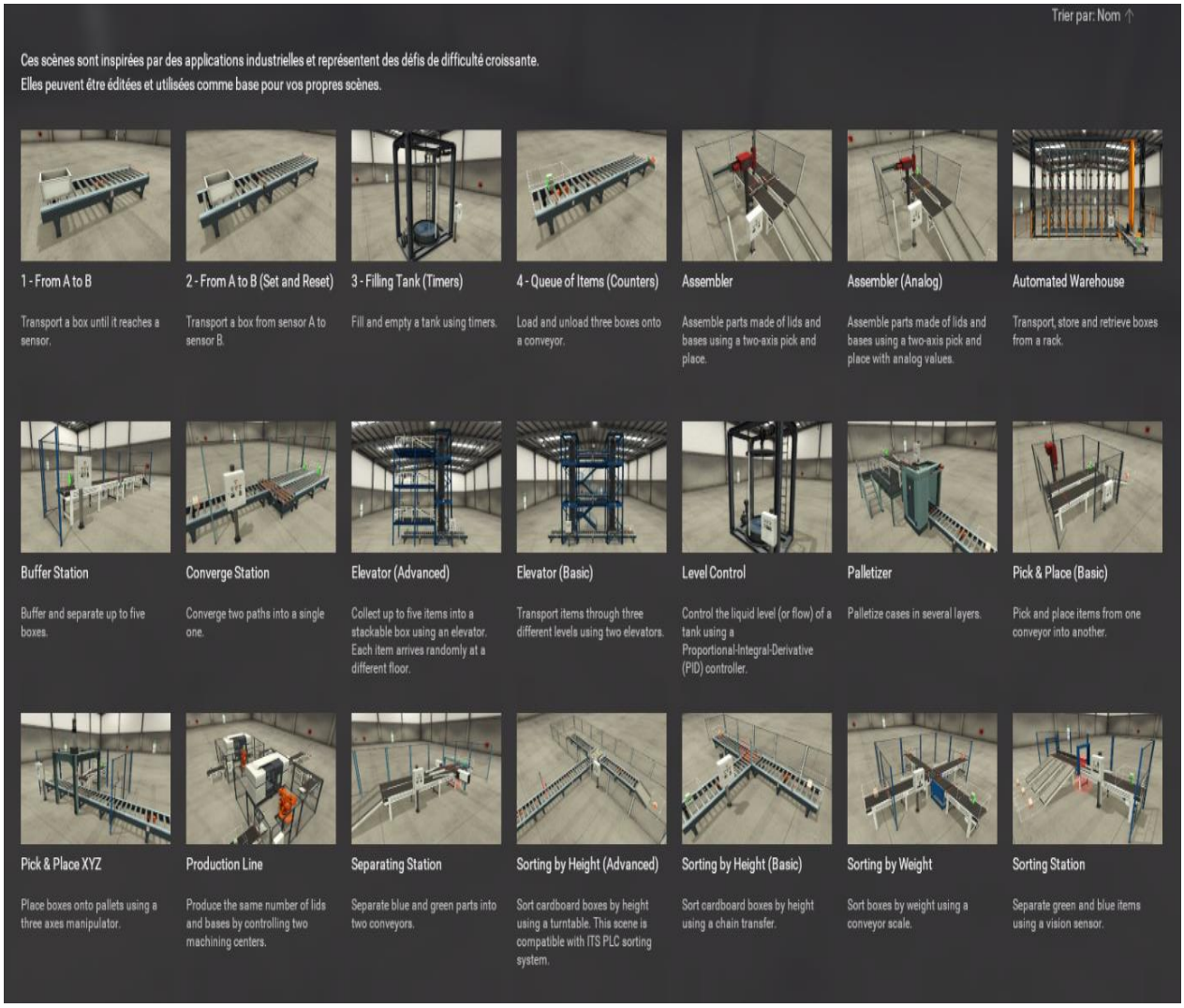


Figure III. 2: Scènes disponibles en Factory I/O

III.3.3 Conception de système

Dans La barre des tâches, nous sélectionnons la fenêtre « palette » nous choisissons ensuite (voir la figure III.3) :

- Réservoir
- Tableau électrique
- Bouton de démarrage
- Bouton d'arrêt
- Voyants vert et rouge
- Voyant d'avertissement
- Sonnette d'alarme

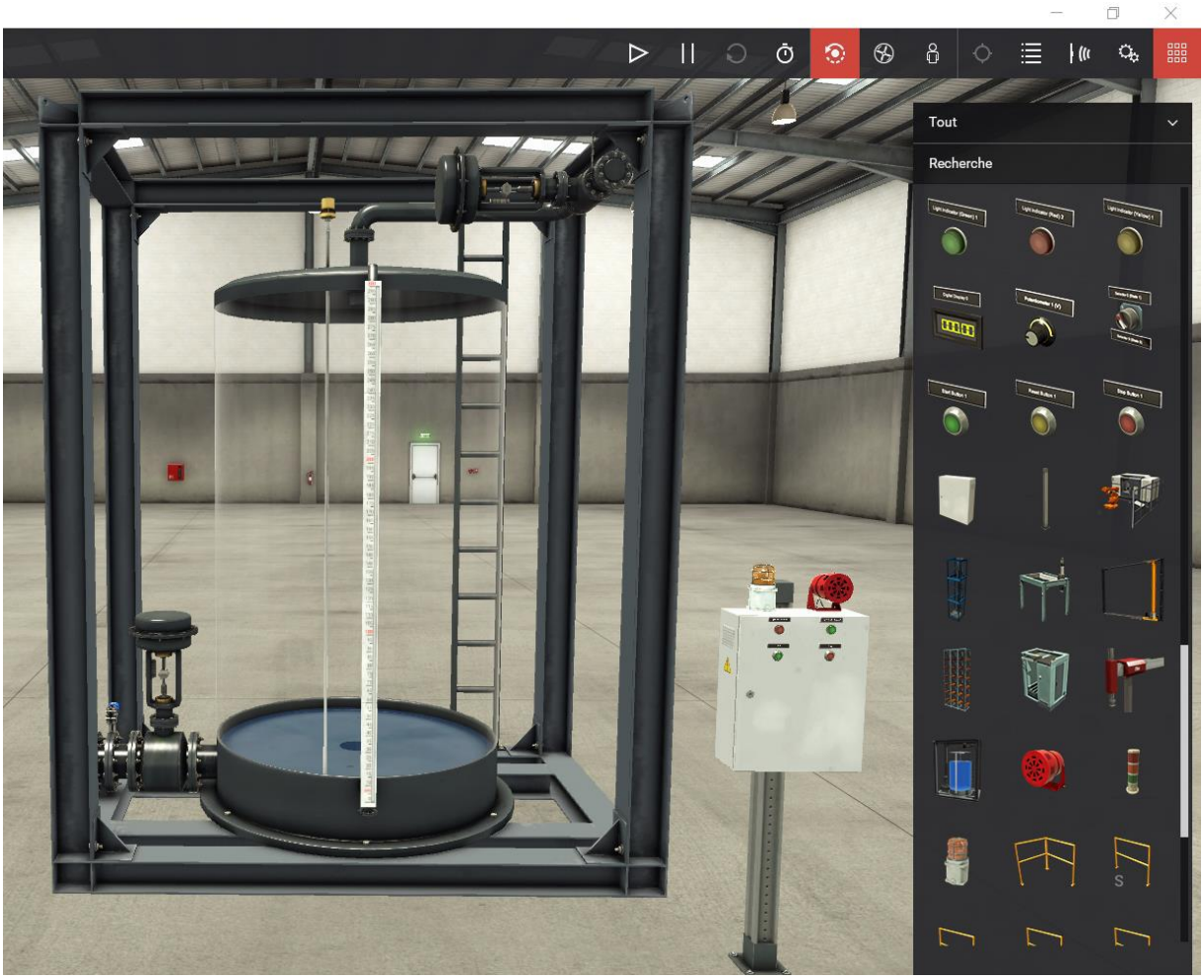


Figure III. 3: Matériel utilisé en Factory I/O

III.3.4 Description du matériel de simulation

III.3.4.1 Réservoir d'eau

Dans notre projet, nous avons utilisé un réservoir cylindrique d'une hauteur de 3 mètres et d'un rayon de 1 mètre, ce réservoir reçoit un débit d'entrée de 0,25 m^3/s lorsque la vanne de remplissage est ouverte (100%) (Voir la figure III.4). Lorsque le réservoir rempli complétement et la vanne d'évacuation ouverte à 100%, il fournit un débit de sortie de 0,3543 m^3/s [38].



Figure III. 4: Réservoir

III.3.4.2 Vanne de contrôle

Une vanne (voir la figure III.5) est un orifice variable utilisé pour réguler le débit de fluide dans un procès en fonction des conditions de service.



Figure III. 5 : Vanne de contrôle

III.3.4. 3 Capteur de niveau capacitif

C'est un instrument utilisé pour mesure le niveau d'un matériau dans un réservoir, à travers la variation de capacité électrique entre deux électrodes, lorsque le niveau du matériau augmente, ce qui permet de mesure le niveau (voir la figure III.6) [37].

Le rôle de ce capteur est convertir la grandeur physique du niveau d'eau dans le réservoir a un signal électrique exploitable (0-10V).

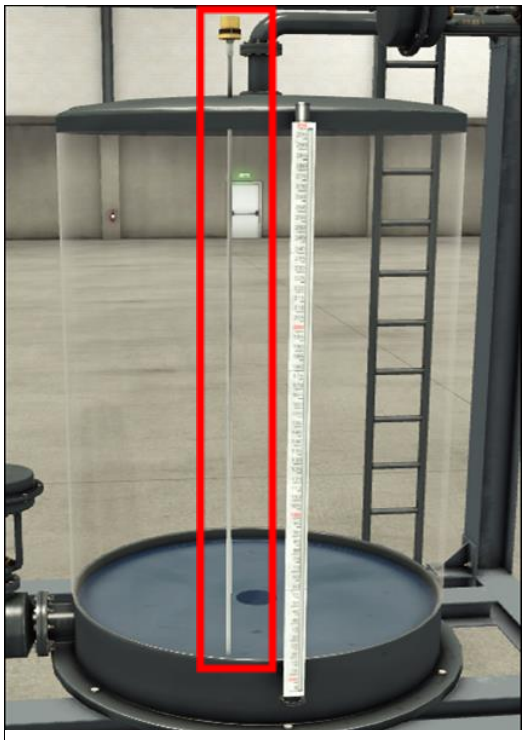


Figure III. 6: Capteur de niveau capacitif

III.3.4.4 Débitmètre

Un débitmètre est un instrument utilisé pour mesurer le débit d'un liquide dans un tuyau à un moment donné, c'est-à-dire la quantité d'eau qui traverse une section donnée sur une unité de temps (voir la figure III.7). C'est pratique surtout quand on veut surveiller ou contrôler un système ou le débit est important [37].

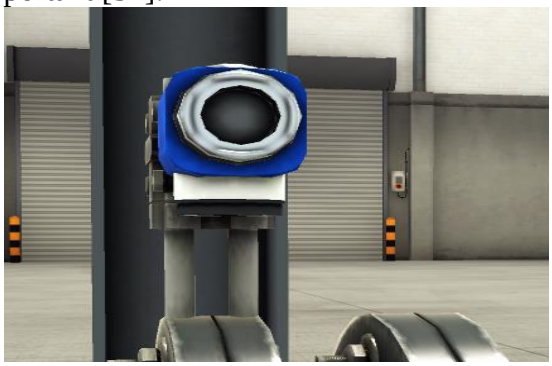


Figure III. 7: Débitmètre

III.3.5 Description des entries/sorties de système

Notre système contient les entries/sorties suivant (voir la figure III.8) :

- 3 entrées TOR
- 2 entrées analogiques
- 4 sorties TOR
- 2 sorties analogiques



Figure III. 8: Entries/sorties de système

III.4 Création d'un projet et configuration du matériel

III.4.1 Création d'un projet

Dans la vue du TIA Portal on doit cliquer sur l'action « Créer un projet », après on donne un nom au projet et on choisit l'emplacement ou on veut l'enregistrer, on peut aussi donner un nom d'auteur et ajouter un commentaire, une fois que les informations sont saisies on peut cliquer sur le bouton « Créer » et on aura la création de notre nouveau projet (voir la figure III.9).

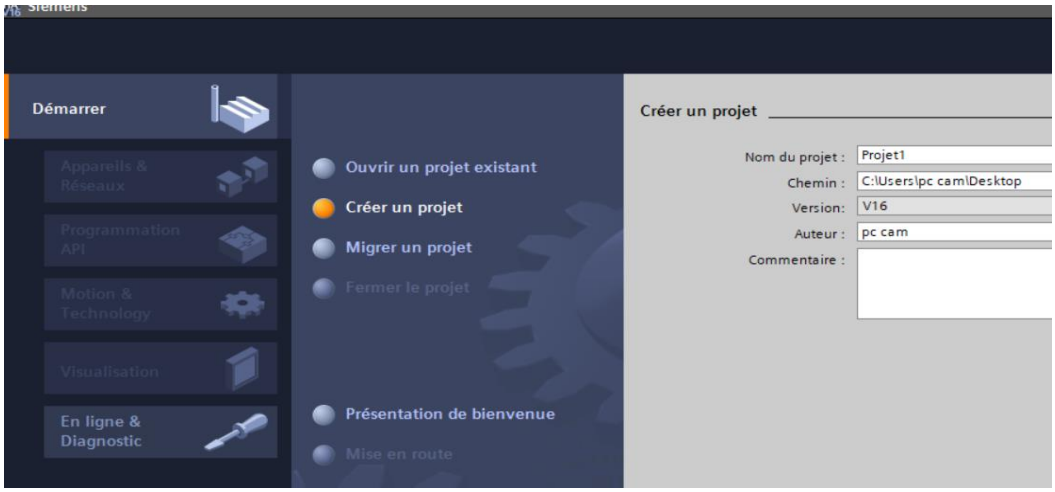


Figure III. 9: Création d'un nouveau projet

III.4.2 Configuration du matériel

Après avoir créé un nouveau projet, l'étape suivante consiste à la configuration du matériel, pour cela on doit aller vers la vue du projet, puis vers le navigateur du projet, là on clique sur « Ajouter un appareil », une fenêtre s'affiche et on aura la liste de tous les matériel tel que les différents automates ainsi que leurs CPU avec tous les caractéristiques et informations nécessaires, on aura aussi les HMI et systèmes PC disponibles. Pour notre projet nous opterons sur SIMATIC S7-1200 avec un CPU 1214C DC/DC/DC, numéro d'article 6ES7 214-1AG40-0XB0 version 4,4 comme le montre la figure III.10.

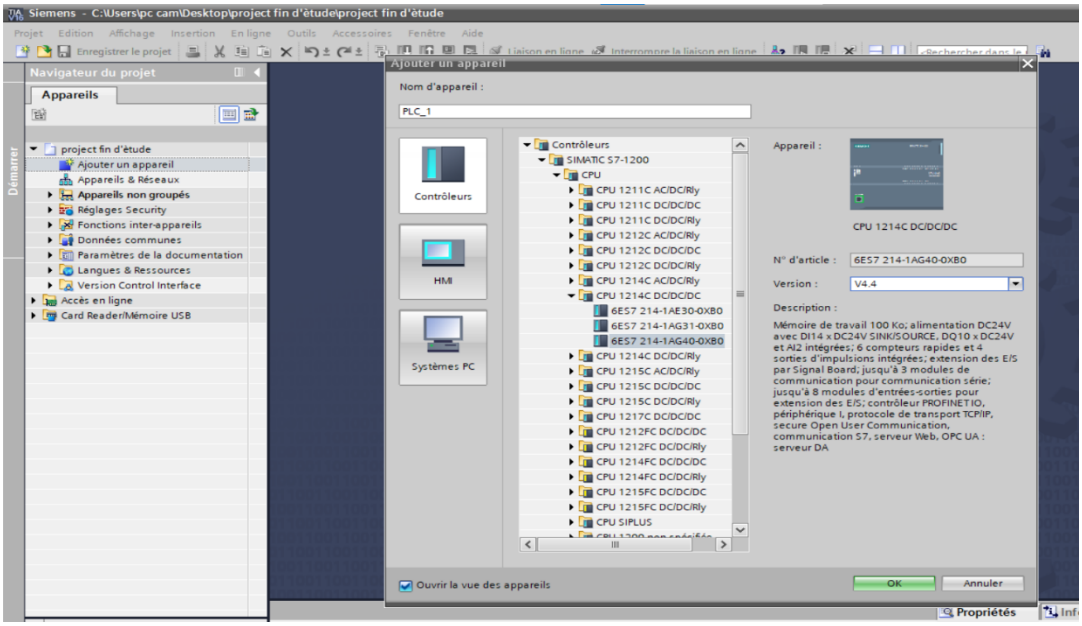


Figure III. 10: Choix de l'API et CPU

Puisque nous avons choisi le SIMATIC S7-1200 avec un CPU 1214C DC/DC/DC le nombre d'entrées et de sorties est :

- 14 entrées numériques : DI 14x24V DC
- 10 sorties numériques : DQ 10x 24V DC
- 2 entrées analogiques : AI 2 intégrées

En l'absence de sorties analogiques sur notre API nous avons ajouté une carte analogique AQ 2x14BIT_1 qui contient deux sorties analogique (voir la figure III.11).

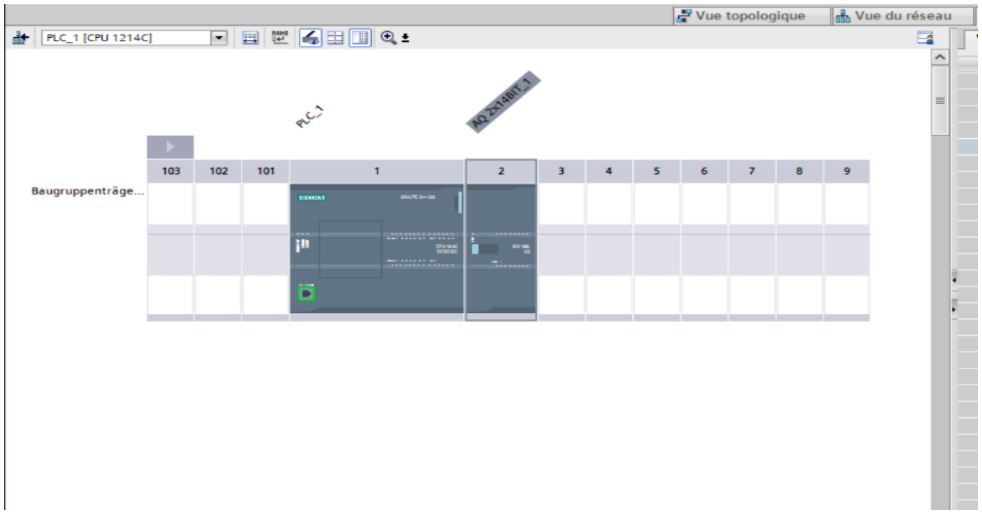
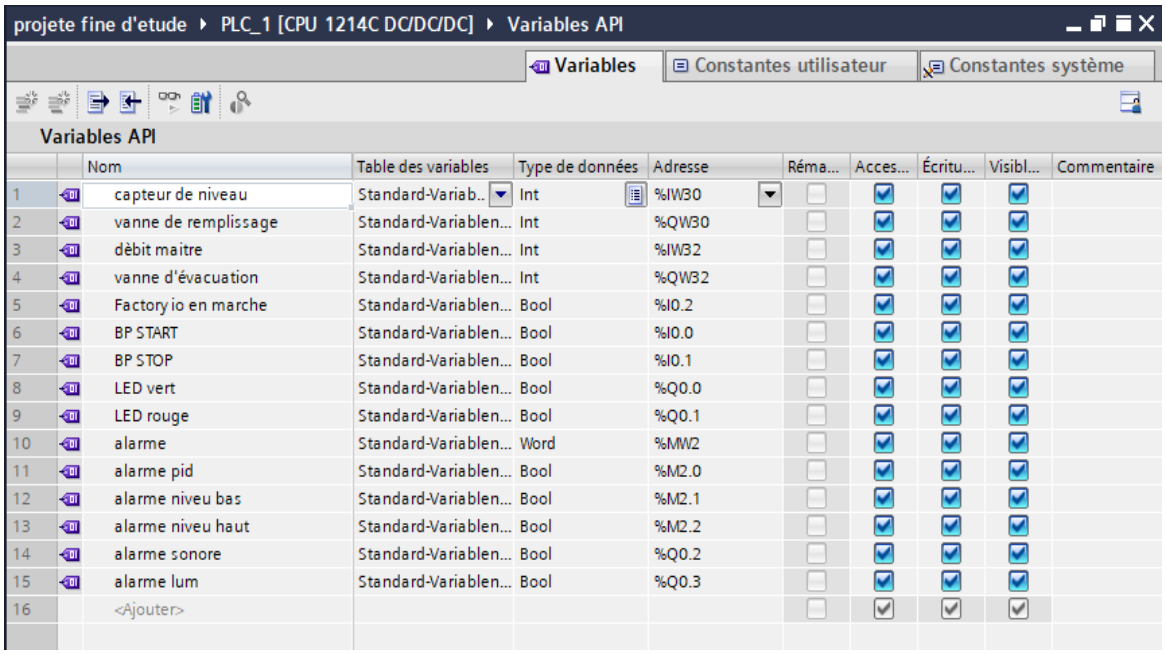


Figure III. 11: Module d'extension AQ 2x14BIT

III.5 Programmation

III.5.1 Tableau des variables

La première étape dans la phase de programmation consiste à créer le tableau des variables. Cette étape est fondamentale car elle permet de définir clairement tous les éléments que le programme devra utiliser, la figure III.12 montre le tableau des variables de notre projet.



	Nom	Table des variables	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Commentaire
1	capteur de niveau	Standard-Variab..	Int	%IW30	☐	☑	☑	☑	
2	vanne de remplissage	Standard-Variablen...	Int	%QW30	☐	☑	☑	☑	
3	dèbit maitre	Standard-Variablen...	Int	%IW32	☐	☑	☑	☑	
4	vanne d'évacuation	Standard-Variablen...	Int	%QW32	☐	☑	☑	☑	
5	Factory io en marche	Standard-Variablen...	Bool	%I0.2	☐	☑	☑	☑	
6	BP START	Standard-Variablen...	Bool	%I0.0	☐	☑	☑	☑	
7	BP STOP	Standard-Variablen...	Bool	%I0.1	☐	☑	☑	☑	
8	LED vert	Standard-Variablen...	Bool	%Q0.0	☐	☑	☑	☑	
9	LED rouge	Standard-Variablen...	Bool	%Q0.1	☐	☑	☑	☑	
10	alarme	Standard-Variablen...	Word	%MW2	☐	☑	☑	☑	
11	alarme pid	Standard-Variablen...	Bool	%M2.0	☐	☑	☑	☑	
12	alarme niveu bas	Standard-Variablen...	Bool	%M2.1	☐	☑	☑	☑	
13	alarme niveu haut	Standard-Variablen...	Bool	%M2.2	☐	☑	☑	☑	
14	alarme sonore	Standard-Variablen...	Bool	%Q0.2	☐	☑	☑	☑	
15	alarme lum	Standard-Variablen...	Bool	%Q0.3	☐	☑	☑	☑	
16	<Ajouter>				☐	☑	☑	☑	

Figure III. 12: Tableau des variables

III.5.2 Blocs de programmation utilisée

Dans notre système de contrôle, nous avons utilisé un ensemble de blocs afin de concevoir un système intégré à savoir les blocs OB1, OB30, FB, BD.

III.5.3 Programmes

Dans cette partie, nous avons présenté la programmation liée au processus de remplissage d'un réservoir, en utilisant principalement le langage **Ladder**.

III.5.3.1 Bloc OB1

- **Réseau 01**

Ce réseau établit la communication entre TIA Portal et Factory I/O à l'aide de la fonction FC9000, cette fonction permet l'échange de données entre l'automate virtuel et la scène simulée sur Factory I/O (voir la figure III.13).

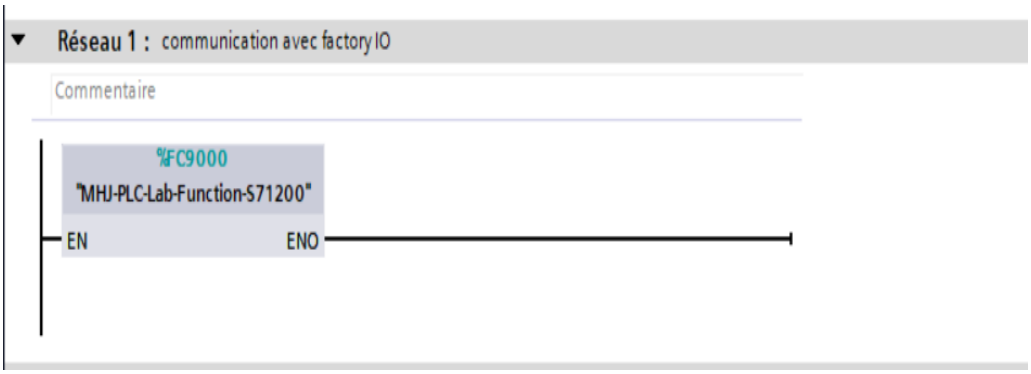


Figure III. 13: Communication avec Factory I/O

- **Réseau 02**

Dans ce réseau nous avons appelé le bloc fonctionnel FB1 : contrôle de niveau, qui contient toute la logique de contrôle de niveau du réservoir (voir la figure III.14), à l'exception du mode PID.

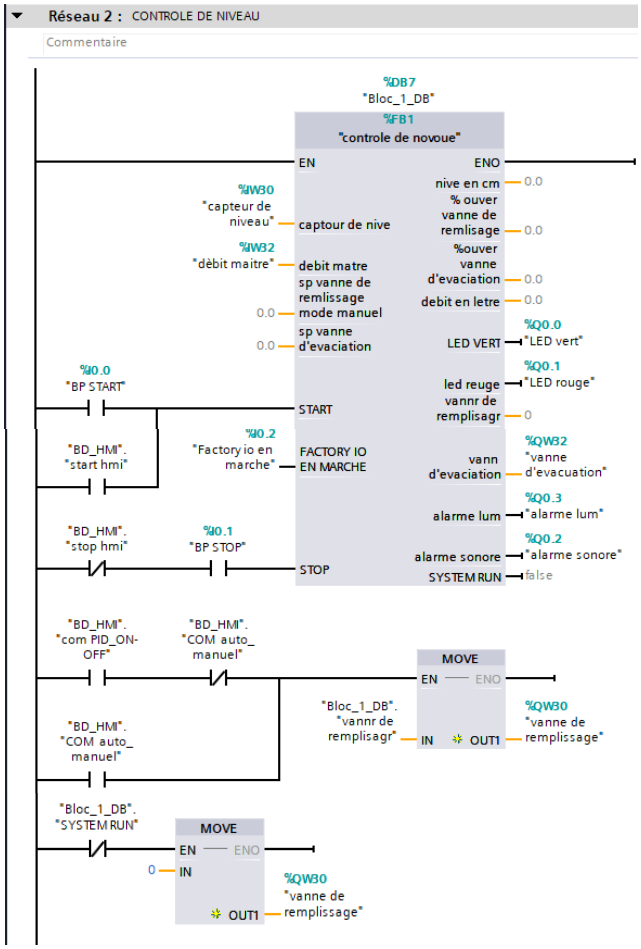


Figure III. 14: Contrôle de niveau

III.5.3.2 Bloc FB

- **Réseau 01**

Dans ce réseau, les blocs fonctionnels NORM_X et SCALE_X ont été utilisés afin de garantir la conversion d'un ensemble de valeurs provenant soit des signaux analogiques des capteurs, soit des commandes manuelles saisies via l'interface HMI (voir la figure III.15).

La fonction NORM_X commence par normaliser les valeurs d'entrée dans une plage relative entre 0 et 1, puis la fonction SCALE_X étend ces valeurs vers une plage réelle correspondant aux grandeurs physiques requises.

Ces blocs ont été utilisés pour atteindre les objectifs suivants :

- La conversion du niveau du réservoir et du débit de sortie en valeurs physiques réelles.
- La détermination du pourcentage réel d'ouverture des vannes de remplissage et d'évacuation.
- La conversion des commandes manuelles (provenant de l'interface HMI) au format INT pour les transmettre correctement aux actionneurs via des sorties analogiques.

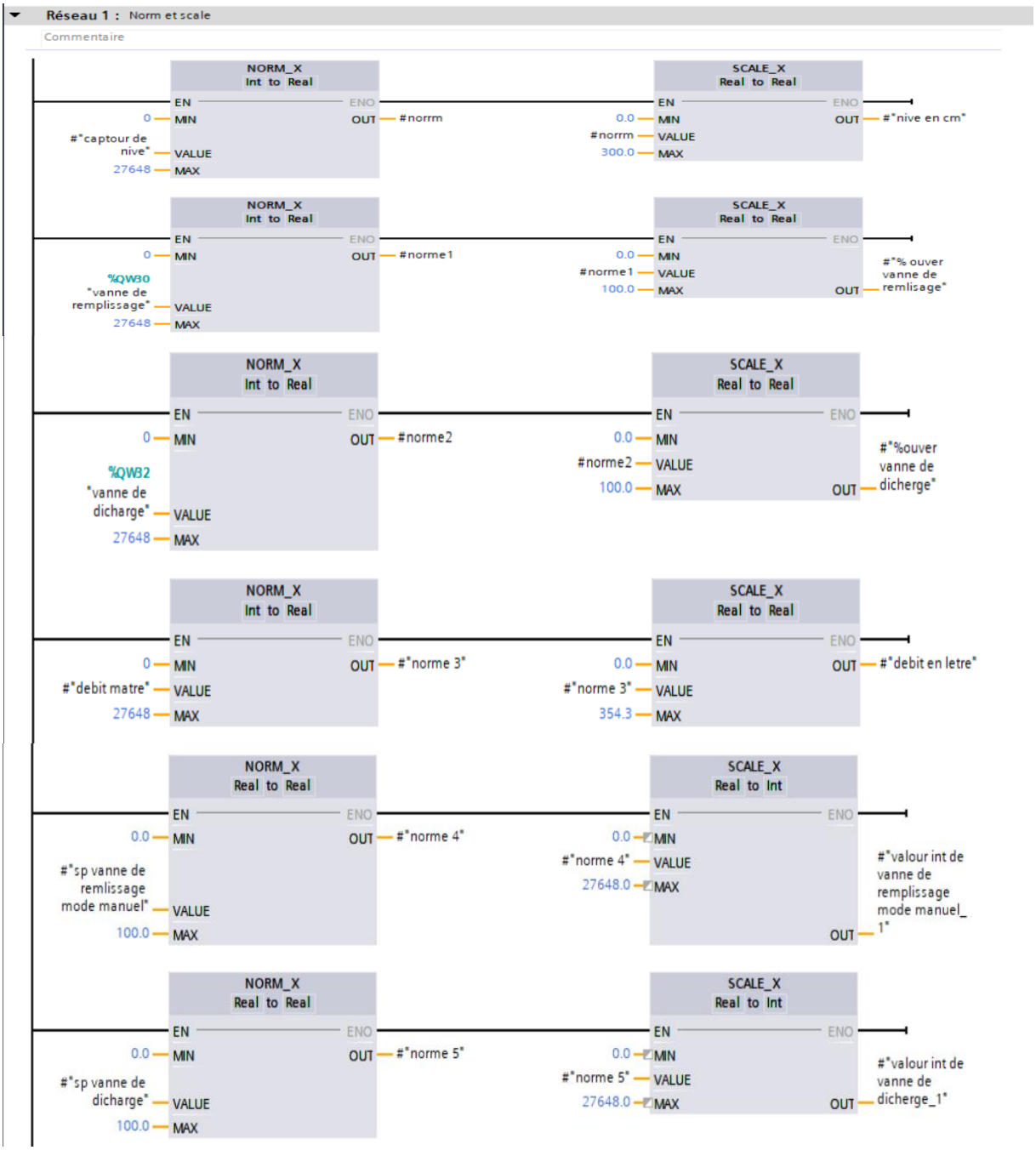


Figure III. 15: Norme et Scale

➢ **Réseau 02**

Ce réseau représente les conditions de démarrage du système et son état (voir la figure III.16).

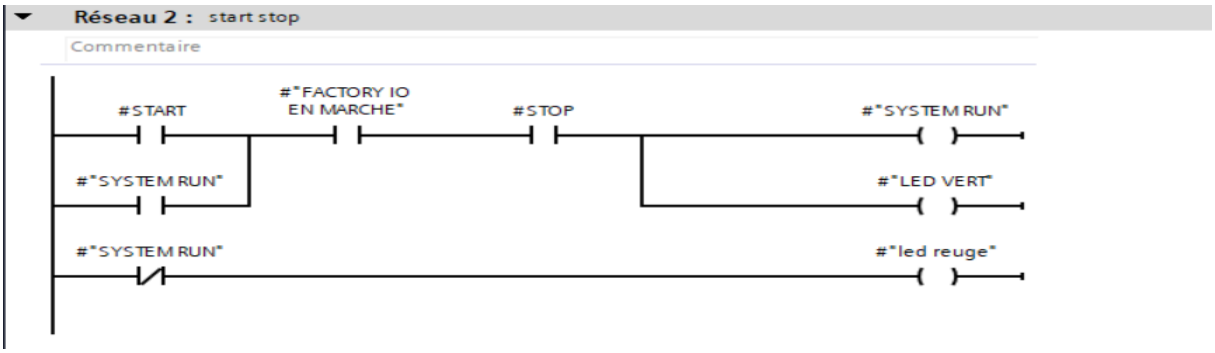


Figure III. 16: Programme marche, arrêter

➢ **Réseau 03**

Ce réseau gère automatiquement le niveau d'un réservoir (voir la figure III.17). Quand le liquide descend en dessous d'une certaine valeur, une vanne s'ouvre automatiquement pour le remplir, puis elle se referme une fois que le niveau voulu est atteint.

Les valeurs pour ouvrir et fermer la vanne sont définies depuis l'HMI. Pour éviter que le niveau dépasse la limite qu'on a fixée, on a ajouté une petite correction dans le système. Cela permet d'être plus précis et de ne pas avoir d'erreurs pendant le fonctionnement.

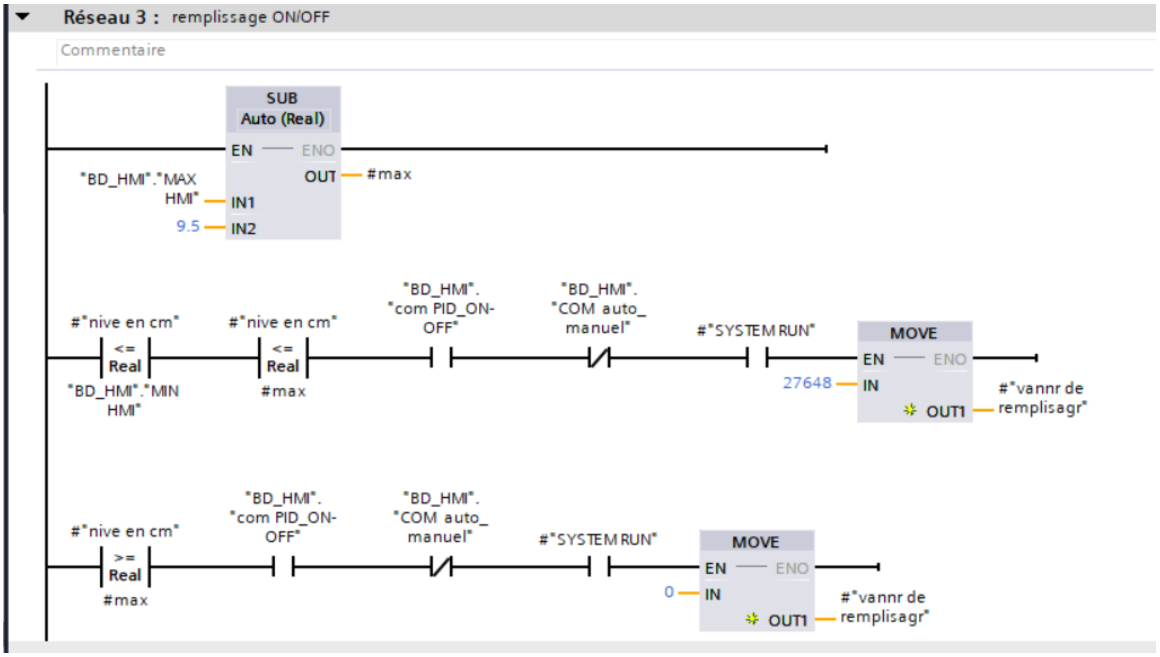


Figure III. 17: Remplissage ON/OFF

- **Réseau 04**

Ce réseau permet la commande manuelle d'une vanne de remplissage (voir la figure III.18). Son principe se base sur deux conditions essentielles :

- système doit être en marche
- mode de commande doit être manuel

Si ces conditions sont remplies, une instruction **move** transfère la consigne en mode manuel vers la sortie de la vanne de remplissage.

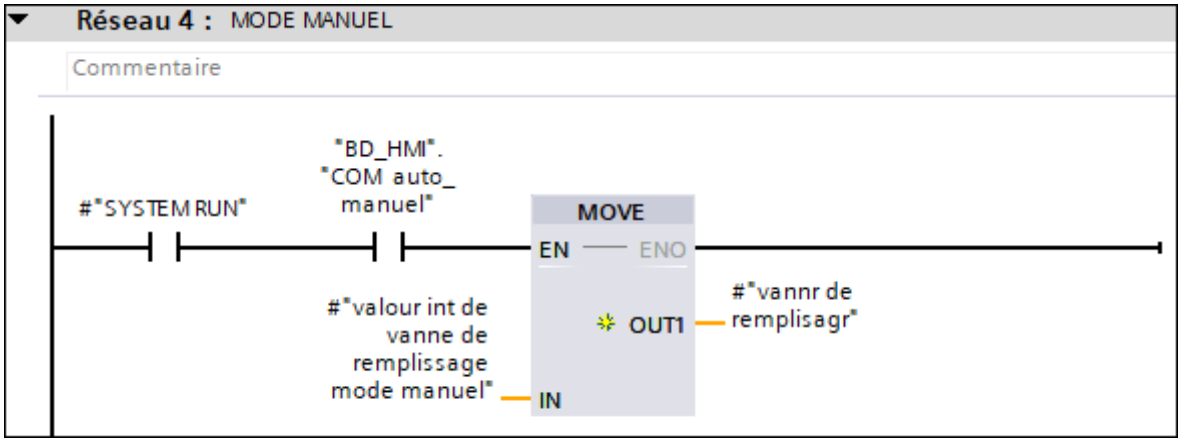


Figure III. 18 : Mode manuel

- **Réseau 05**

Ce réseau est responsable de l'ouverture de la vanne d'évacuation (voir la figure III.19). L'utilisateur saisit la valeur souhaitée dans l'interface correspondante sur HMI.

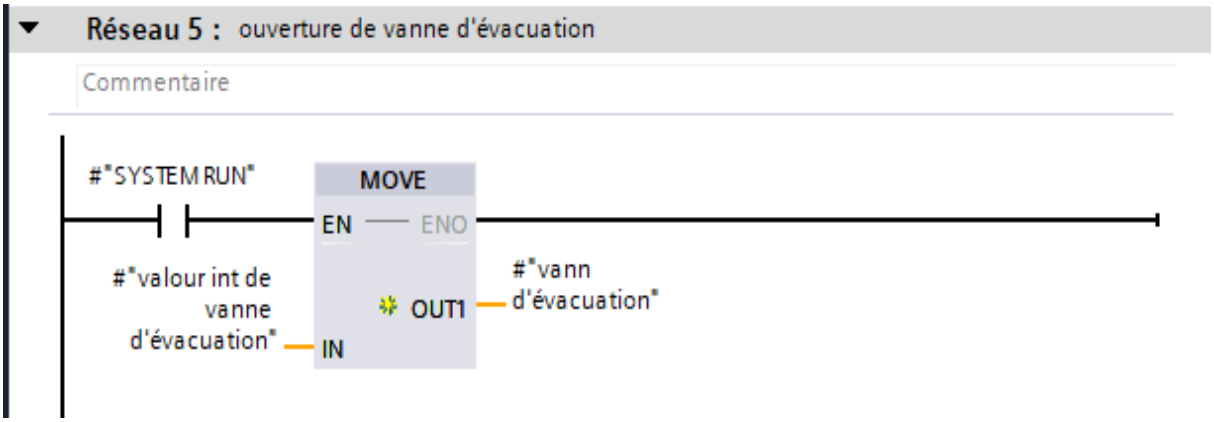


Figure III. 19: Ouverture de vanne d'évacuation

➢ **Réseau 06**

Ce réseau représente la fermeture des vannes, la vanne d'évacuation être fermé lorsque le système est en arrêt (voir la figure III.20). La vanne de remplissage se ferme nous changeons de mode, elle se ferme lors nous changeons entre le mode manuel et automatique, ainsi que lors nous changeons entre le mode PID et ON/OFF. Nous utilisons une instruction **move** pour attribuer les valeurs correctes de fermeture de vanne.

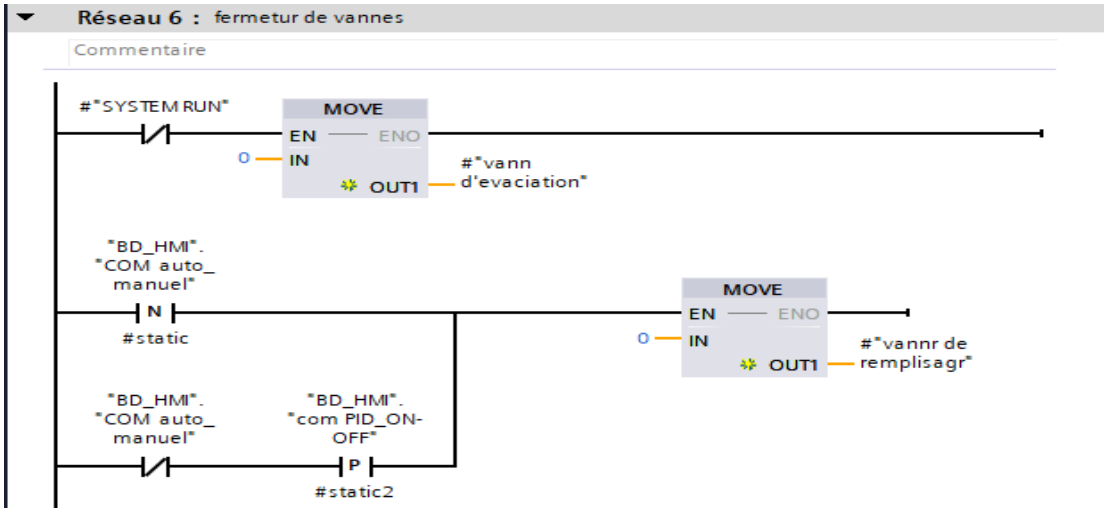


Figure III. 20: Fermeture des vannes

➢ **Réseau 07**

Ce réseau surveille les écarts entre la consigne et la valeur réelle dans un automate programmable (voir la figure III.21). Lorsque l'écart dépasse un seuil défini sur HMI, une temporisation est activée avant de déclencher une alarme pour signaler une dérive critique du processus PID. Cela garantit un contrôle précis et une meilleure stabilité du système.

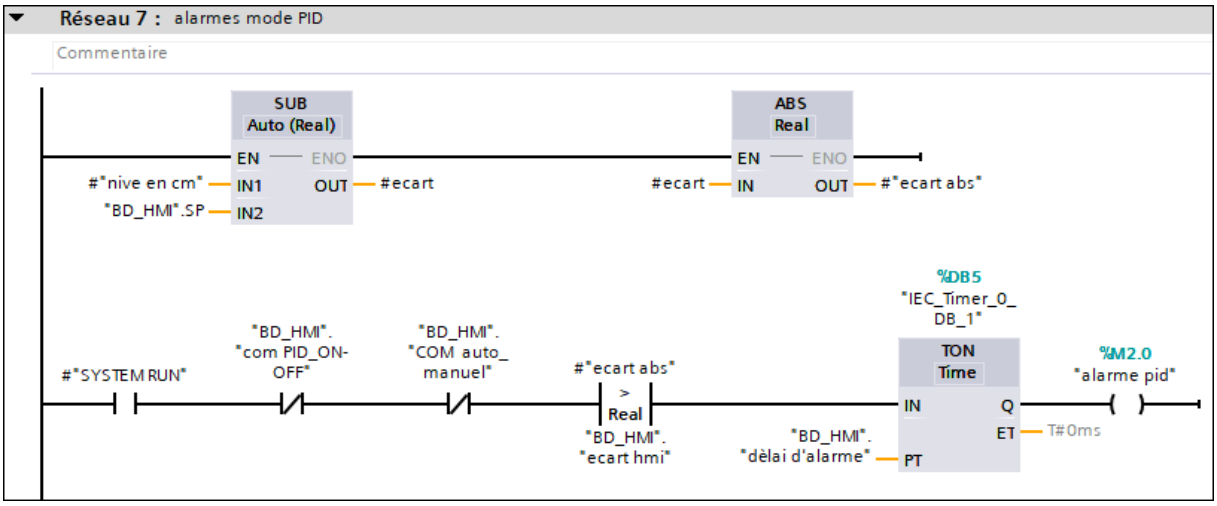


Figure III. 21: Alarmes mode PID

ABS :(Absolue) prend une valeur en entrée et renvoie sa valeur absolue en sortie. Peu importe si l'entrée est positive ou négative, le résultat sera toujours positif.

TON : lorsqu'une condition est vraie, il démarre un compte à rebours. Si cette condition reste active jusqu'à la fin du temps défini, une sortie s'active (une alarme), si la condition disparaît avant la fin du temps, le temporisateur se réinitialise.

SUB : effectue une soustraction entre deux valeurs. Il prend in1 (niveau en cm) et in2 (BD_HM.SP), puis calcule la différence (OUT = écart).

- **Réseau 08**

Ce réseau active une alarme si le niveau est trop bas ou trop haut, lorsque le système est en marche (voir la figure III.22).

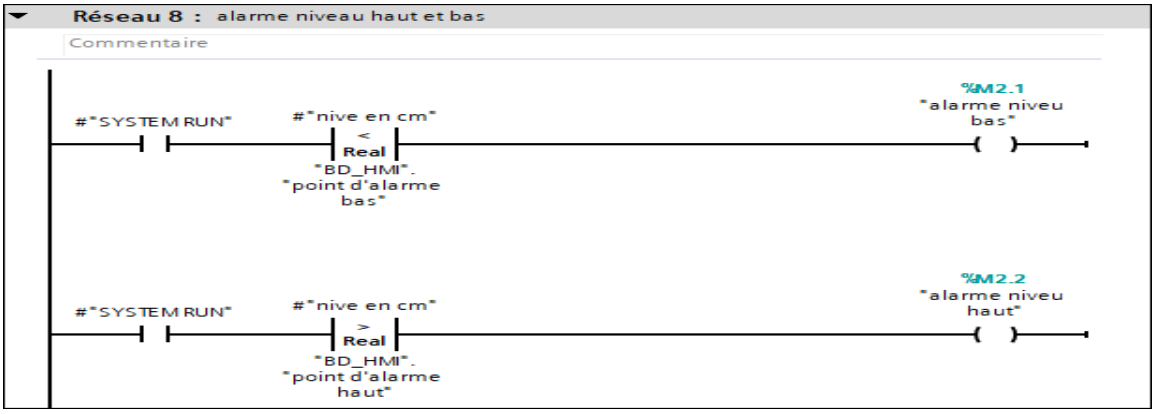


Figure III. 22: Alarmes niveau haut et bas

- **Réseau 09**

Ce réseau permet la gestion des alarmes dans Factroy I/O (voir la figure III.23). Il regroupe trois conditions pour déclencher l'alarme : un niveau trop bas, un niveau trop haut et un problème dans le PID.Si l'un de ces problèmes se produit, deux alarmes sont activées simultanément.

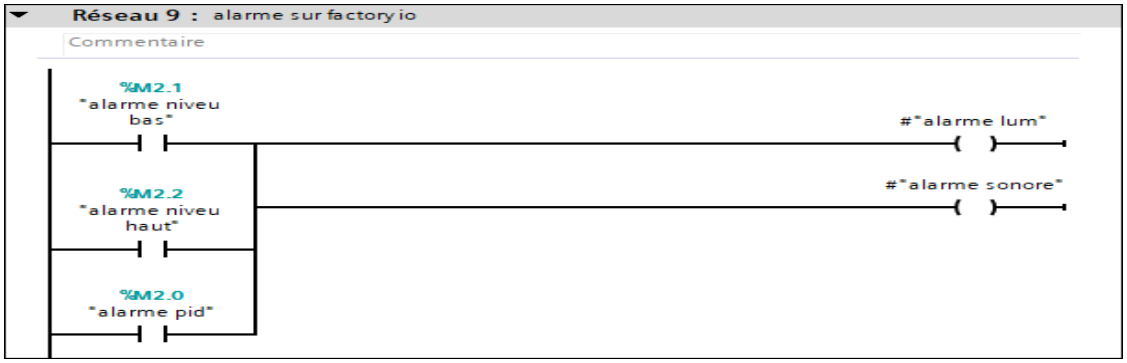


Figure III. 23: Alarmes sur Factory I/O

III.5.3.3 Bloc OB30

- **Réseau 01**

Ce réseau met en œuvre un régulateur PID Compact pour assurer le contrôle du niveau dans un réservoir (voir la figure III.24). L'activation du bloc est conditionnée par le mode automatique et l'état global de notre système. La sortie du régulateur est ensuite transmis à une vanne de remplissage, permettant d'ajuster le débit de manière continue et précise selon le besoin. Le PID Compact est un bloc fonctionnel intégré dans TIA Portal, conçu pour assurer une régulation continue et précise. Il compare la valeur mesurée à la consigne (Set point) et ajuste automatiquement la sortie pour minimiser l'écart entre les deux.

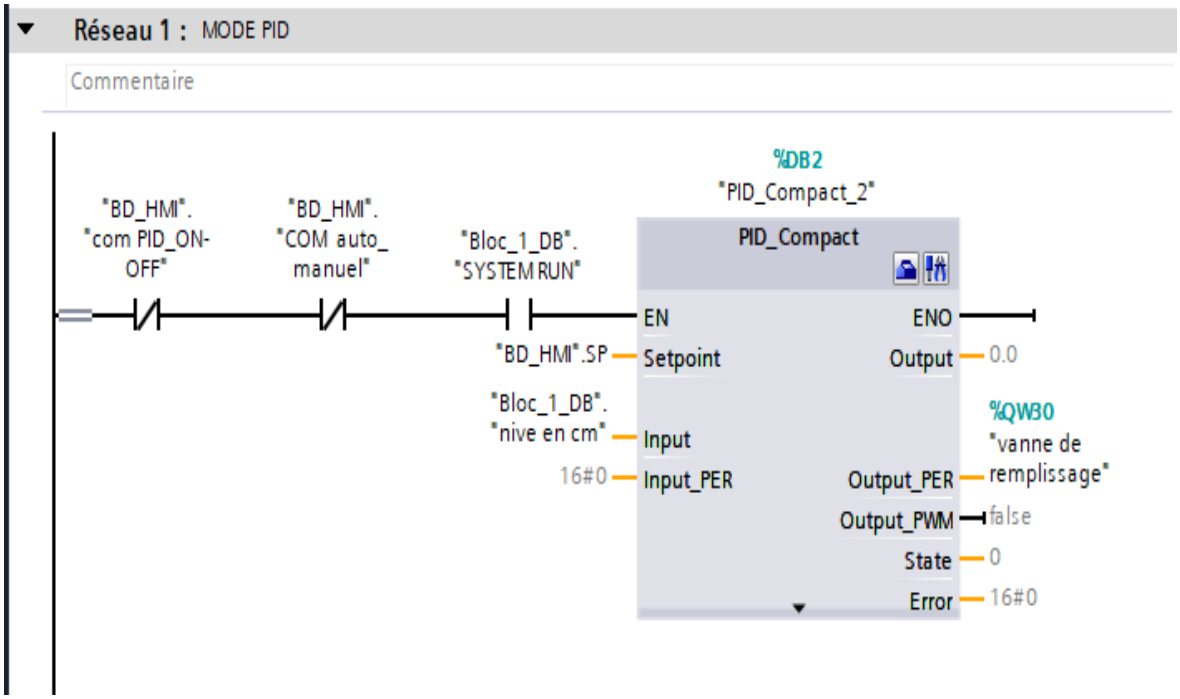
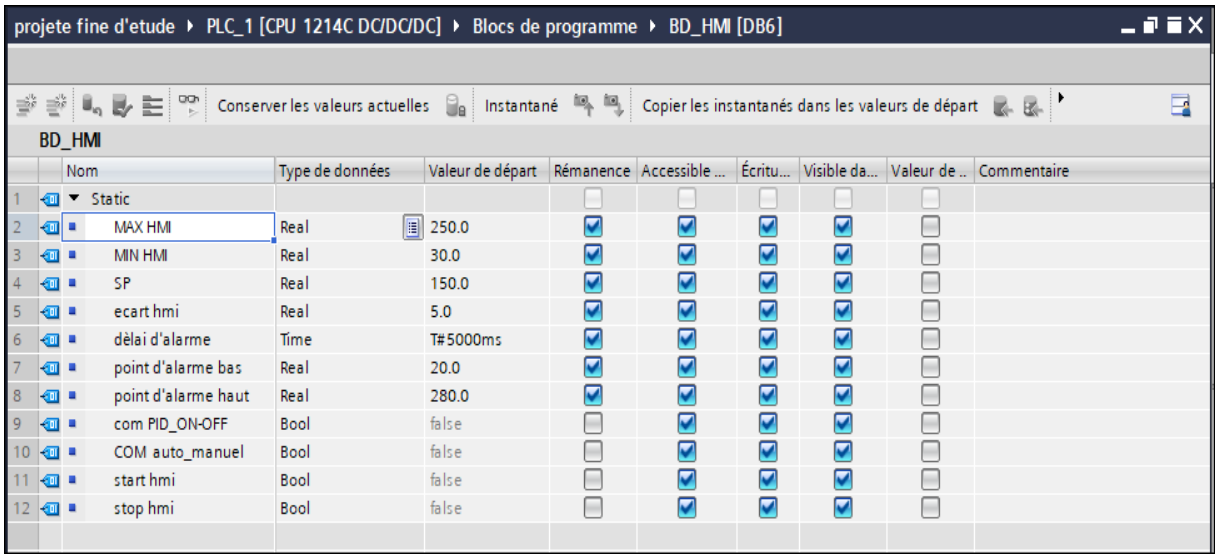


Figure III. 24: Bloc PID compacte

III.5.3.4 Bloc BD

Ce bloc contient les variables d'entrée du système de contrôle qui se trouvent sur l'interface HMI (voir la figure III.25). Nous avons utilisé ces variables dans notre programmation.



	Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Écritu...	Visible da...	Valeur de ..	Commentaire
1	Static								
2	MAX HMI	Real	250.0	☑	☑	☑	☑	☐	
3	MIN HMI	Real	30.0	☑	☑	☑	☑	☐	
4	SP	Real	150.0	☑	☑	☑	☑	☐	
5	ecart hmi	Real	5.0	☑	☑	☑	☑	☐	
6	dèlai d'alarme	Time	T#5000ms	☑	☑	☑	☑	☐	
7	point d'alarme bas	Real	20.0	☑	☑	☑	☑	☐	
8	point d'alarme haut	Real	280.0	☑	☑	☑	☑	☐	
9	com PID_ON-OFF	Bool	false	☐	☑	☑	☑	☐	
10	COM auto_manuel	Bool	false	☐	☑	☑	☑	☐	
11	start hmi	Bool	false	☐	☑	☑	☑	☐	
12	stop hmi	Bool	false	☐	☑	☑	☑	☐	

Figure III. 25: Tableau de variables BD

III.6 Supervision

III.6.1 Création et configuration de L'HMI

Pour ajouter une station HMI, il faut Suivez les étapes suivants (voir la figure III.26) :

- Cliquer sur « ajouter un appareil ».
- Cliquer sur « HMI ».
- Choisir l'HMI correspondant puis la version.

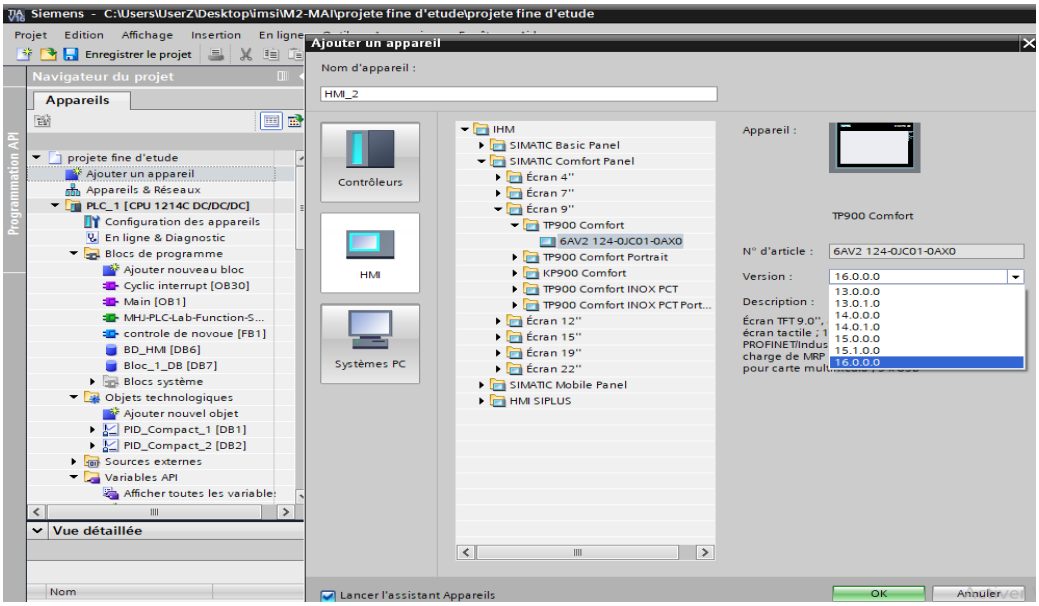


Figure III. 26: Insertion d'une station HMI dans TIA Portal

III.6.2 Etablissement de la connexion entre la CPU et L'HMI

Pour établir la liaison entre la CPU et l'interface homme machine TP 900 COMFORT on doit aller à appareils et réseaux développés, en lie le API avec l'HM comme illustre dans la figure III.27. La liaison entre le CPU 1214C et HMI TP 900 COMFORT est faites par PROFINET.

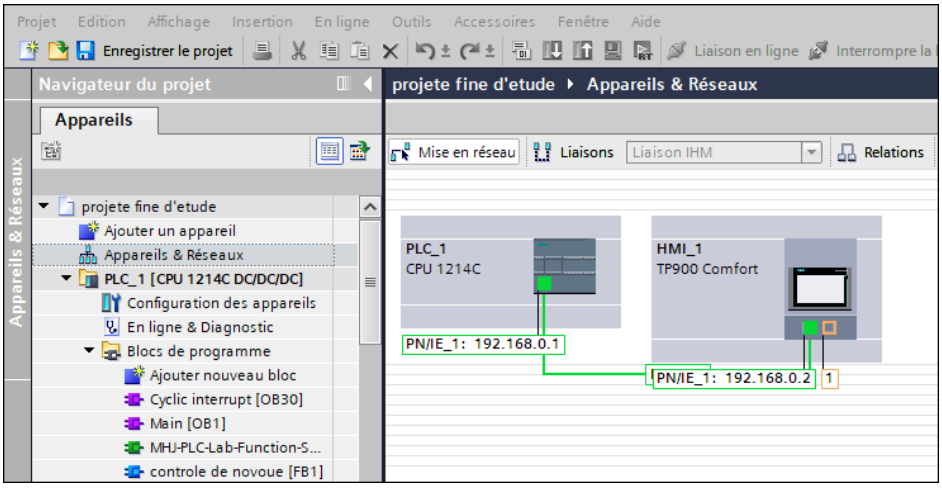
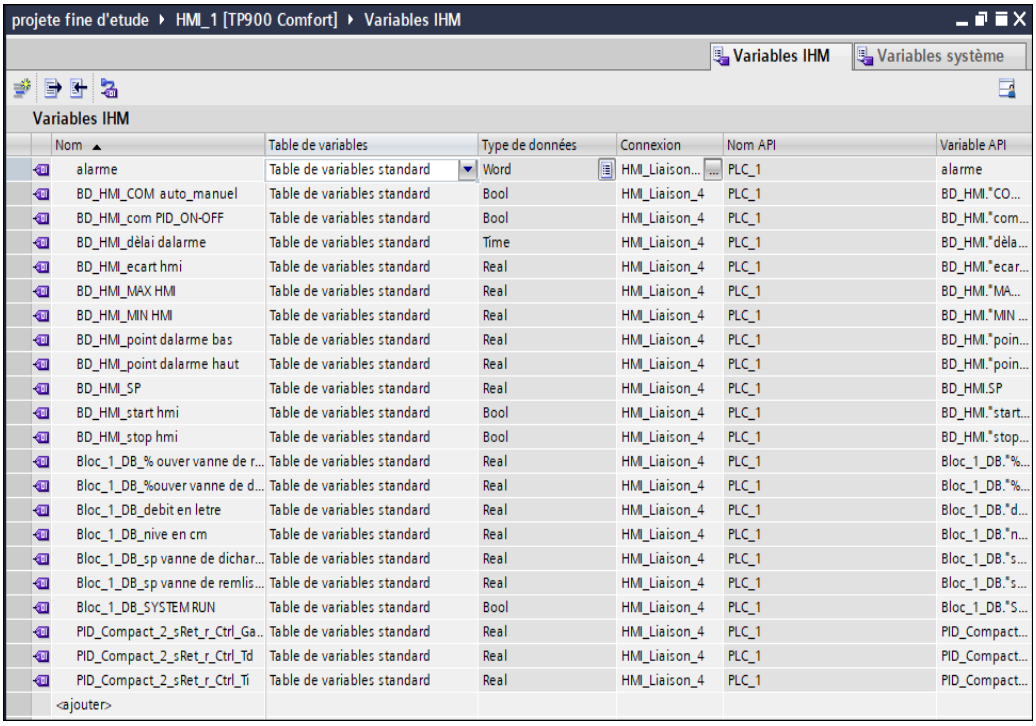


Figure III. 27: Liaison entre api et HMI

III.6.3 Variables HMI

Après avoir terminé la création de notre HMI et l'avoir connectée à API, l'étape suivante consiste à établir le tableau des variables HMI comme illustré dans la figure III.28.



Nom	Table de variables	Type de données	Connexion	Nom API	Variable API
alarme	Table de variables standard	Word	HMI_Liaison...	PLC_1	alarme
BD_HMI_COM auto_manuel	Table de variables standard	Bool	HMI_Liaison_4	PLC_1	BD_HMI."CO...
BD_HMI_com PID_ON-OFF	Table de variables standard	Bool	HMI_Liaison_4	PLC_1	BD_HMI."com...
BD_HMI_dèlai dalarme	Table de variables standard	Time	HMI_Liaison_4	PLC_1	BD_HMI."dèla...
BD_HMI_ecart hmi	Table de variables standard	Real	HMI_Liaison_4	PLC_1	BD_HMI."ecar...
BD_HMI_MAX HMI	Table de variables standard	Real	HMI_Liaison_4	PLC_1	BD_HMI."MA...
BD_HMI_MIN HMI	Table de variables standard	Real	HMI_Liaison_4	PLC_1	BD_HMI."MIN ...
BD_HMI_point dalarme bas	Table de variables standard	Real	HMI_Liaison_4	PLC_1	BD_HMI."poin...
BD_HMI_point dalarme haut	Table de variables standard	Real	HMI_Liaison_4	PLC_1	BD_HMI."poin...
BD_HMI_SP	Table de variables standard	Real	HMI_Liaison_4	PLC_1	BD_HMI.SP
BD_HMI_start hmi	Table de variables standard	Bool	HMI_Liaison_4	PLC_1	BD_HMI."start...
BD_HMI_stop hmi	Table de variables standard	Bool	HMI_Liaison_4	PLC_1	BD_HMI."stop...
Bloc_1_DB_% ouver vanne de r...	Table de variables standard	Real	HMI_Liaison_4	PLC_1	Bloc_1_DB."%...
Bloc_1_DB_%ouver vanne de d...	Table de variables standard	Real	HMI_Liaison_4	PLC_1	Bloc_1_DB."%...
Bloc_1_DB_debit en letre	Table de variables standard	Real	HMI_Liaison_4	PLC_1	Bloc_1_DB."d...
Bloc_1_DB_nive en cm	Table de variables standard	Real	HMI_Liaison_4	PLC_1	Bloc_1_DB."n...
Bloc_1_DB_sp vanne de dichar...	Table de variables standard	Real	HMI_Liaison_4	PLC_1	Bloc_1_DB."s...
Bloc_1_DB_sp vanne de remlis...	Table de variables standard	Real	HMI_Liaison_4	PLC_1	Bloc_1_DB."s...
Bloc_1_DB_SYSTEM RUN	Table de variables standard	Bool	HMI_Liaison_4	PLC_1	Bloc_1_DB."S...
PID_Compact_2_sRet_r_Ctrl_Ga...	Table de variables standard	Real	HMI_Liaison_4	PLC_1	PID_Compact...
PID_Compact_2_sRet_r_Ctrl_Td	Table de variables standard	Real	HMI_Liaison_4	PLC_1	PID_Compact...
PID_Compact_2_sRet_r_Ctrl_Ti	Table de variables standard	Real	HMI_Liaison_4	PLC_1	PID_Compact...
<ajouter>					

Figure III. 28: Variables HMI

III.6.4 Création des vues HMI

Pour créer une vue sur l'HMI, nous devants suivre les étapes suivante :

Double-cliquez sur l'option HMI_1 puis clique sur vues puis sur ajouter une vue.

Une fois cette étape terminée, notre interface sera prête et il sera possible d'y ajouter des éléments selon les besoins (voir la figure III.29), tels qu'un réservoir par exemple.

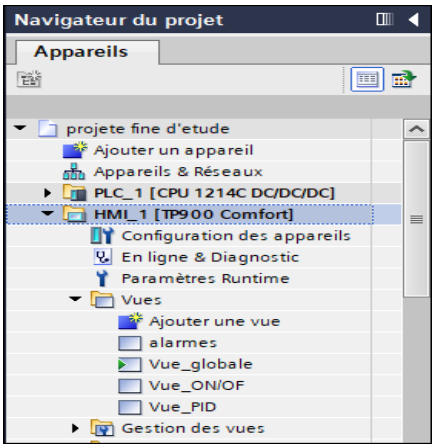


Figure III. 29: Création d'une vue

III.6.5 Présentation des différant vues du projet

III.6.5.1 Vue global

Cette interface nous permet de suivre le niveau du liquide dans le réservoir et nous indique le niveau réel de l'eau en cm, le débit de sortie du réservoir, ainsi que l'état des vannes de remplissage et de vidange (voir la figure III.30). L'utilisateur peut également choisir un mode de fonctionnement, et il est possible de connaitre l'état du système grâce aux voyants lumineux .L'interface permet aussi de navigateur vers d'autres interfaces de l'HMI.

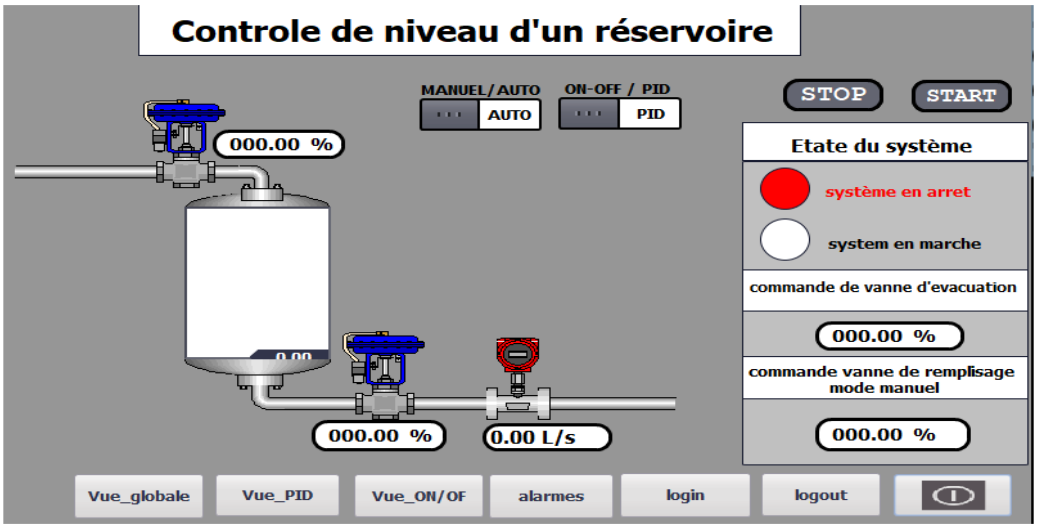


Figure III. 30: Vue global

III.6.5.2 Vue mode PID

Cette interface représente le mode PID du système où l'on peut suivre en temps réel la valeur mesurée PV, avec une courbe montrant son évolution comparée à la consigne SP (voir la figure III.31).

Elle permet également de modifier la valeur de SP, ainsi que de régler les paramètres du PID.On peut aussi définir une erreur autorisée entre SP et PV, ainsi que la durée pendant laquelle cet écart doit persister avant qu'une alarme ne soit déclenchée.

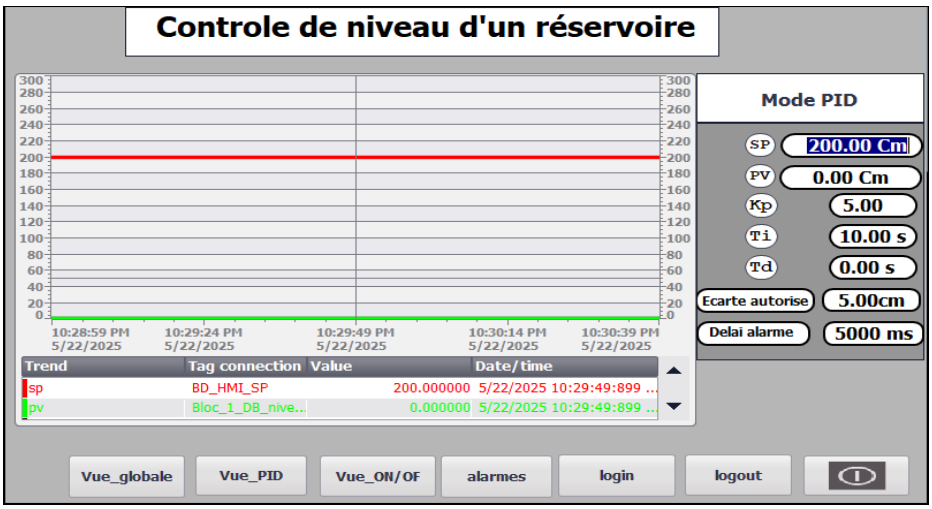


Figure III. 31: Vue mode PID

III.6.5.3 Vue mode ON/OFF

La figure III.32 montre une interface de supervision pour le contrôle de niveau d'un réservoir en mode ON/OFF. Le graphique montre le remplissage et l'évacuation du liquide au cours du temps. L'utilisateur peut modifier les valeurs d'ouverture et de fermeture de la vanne de remplissage et peut fixer des limites qui déclencheront les alarmes de niveaux haut et bas.

.

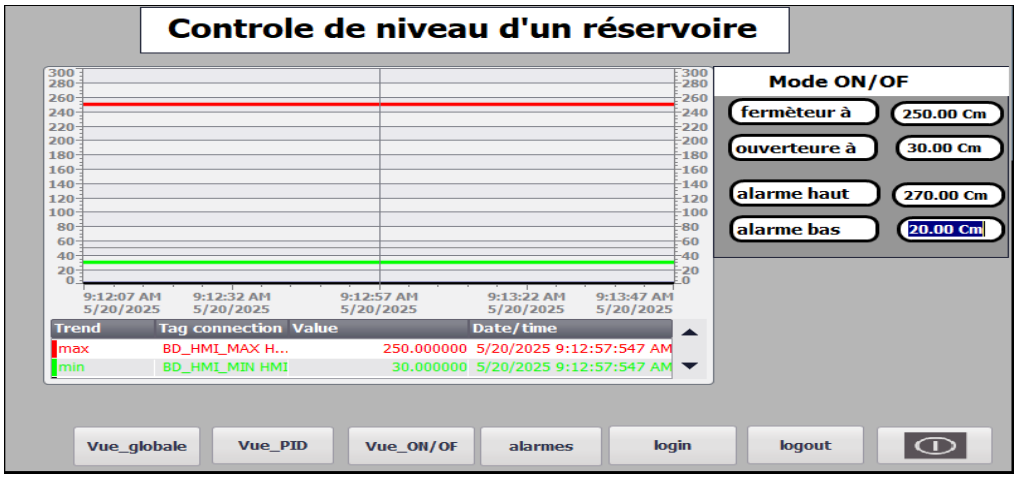


Figure III. 32: Vue mode ON/OFF

III.6.5.4 Vue alarmes

La figure III.33 représente l'interface de l'archive des alarmes du système, elle permet de visualiser les alertes signalées.

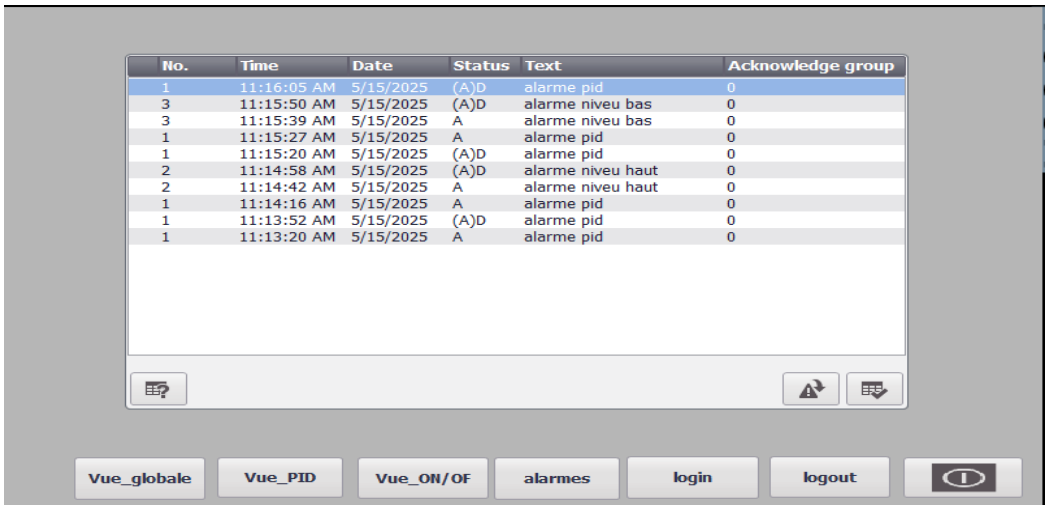


Figure III. 33: Vue alarmes

III.6.6 Configuration des alarmes au niveau de l'HMI

Pour créer des alarmes, il est nécessaire de se rendre dans la section "Alarme HMI" et de choisir le type d'alarme, que ce soit une alarme de bit ou une alarme analogique (voir la figure III.34).

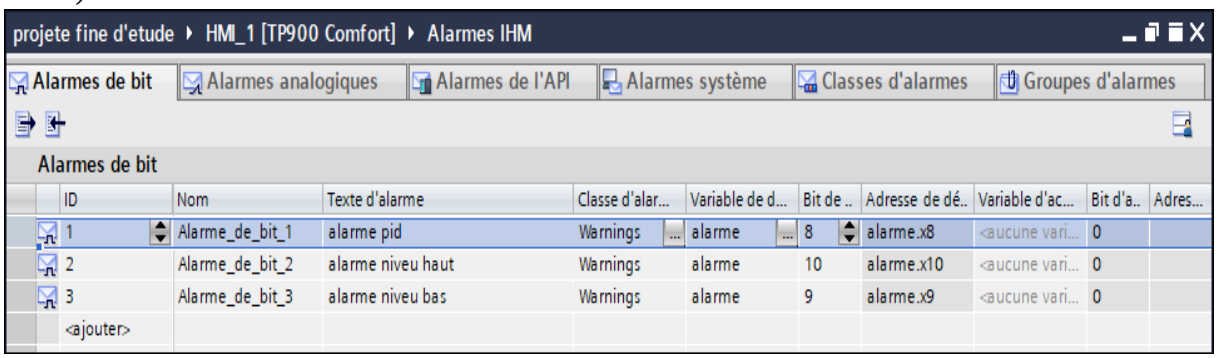


Figure III. 34: Configuration d'alarmes

III.6.7 Gestion des utilisateurs

La gestion des utilisateurs consiste en la création, la modification et l'assignation de rôles personnalisés à tous les opérateurs concernant les accès autorisés du système. En termes de sécurité, les mots de passe puissants empêchent toute forme d'intrusion. Dans ce contexte, pour sécuriser notre système un mot de passe a été défini pour éviter une éventuelle modification sans autorisation (voir la figure III.35). Elles comprennent les étapes suivantes :

- Cliquez sur le bouton gestion des utilisateurs.
- Définissez le mot de passe.

- Sélectionnez le nombre d'emplacements.

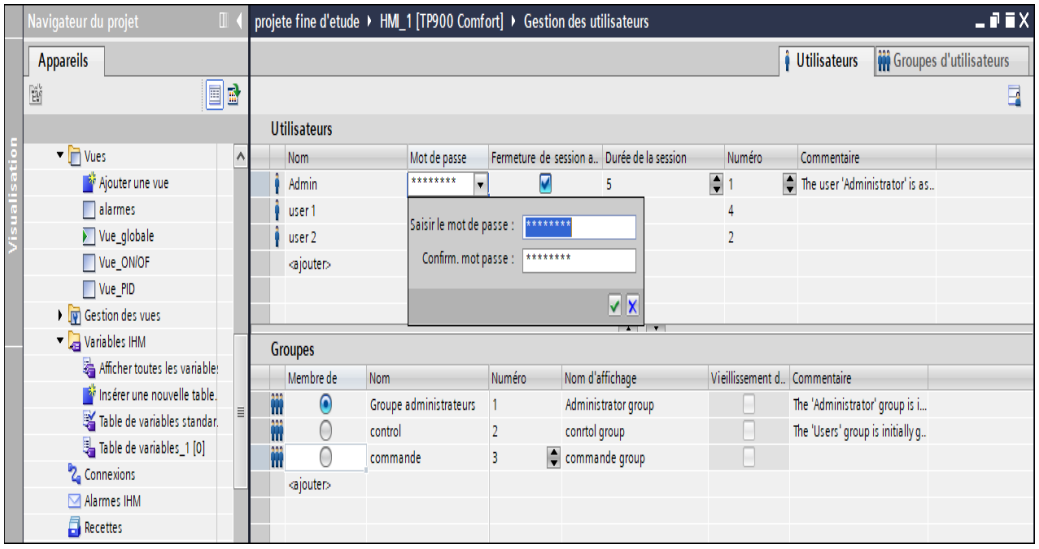


Figure III. 35: Sécurité d'HMI

Pour donner les autorisations aux utilisateurs, il faut accéder à la gestion des utilisateurs après clique sur groupes d'utilisateurs. Comme indiqué sur la figure III.36.

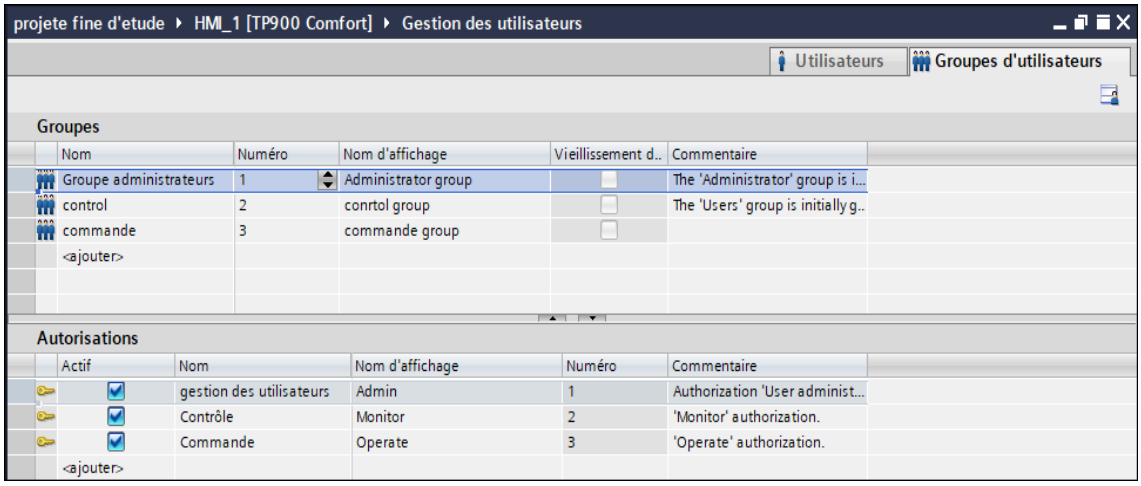


Figure III. 36: Création des autorisations

III.7 Simulation de projet

III.7.1 Chargement de programme

Pour charger le programme dans S7 PLC SIM V16, on commence avec la compilation du programme a l'aide du bouton disponible dans la barre des d'outils pour vérifier qu'il n'y a

aucune erreur dans le système ou le programme, après on clique sur charger qui se trouve aussi dans la barre d'outils (voir la figure III.37).

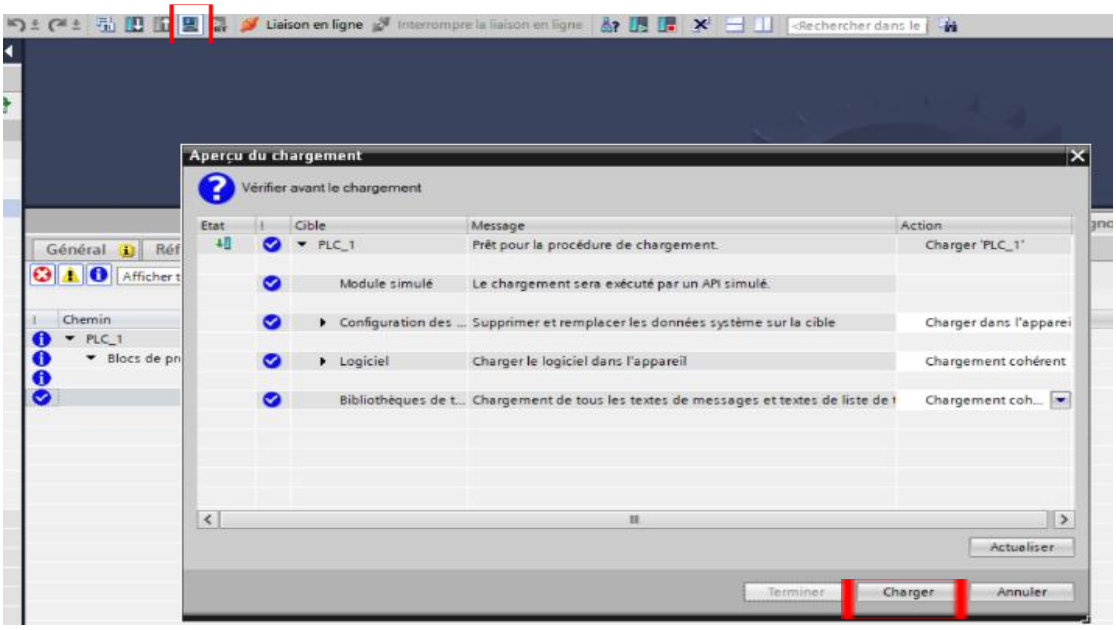


Figure III. 37: Chargement de programme

Après avoir terminé le chargement du programme dans PLC SIM son interface s'affiche comme illustré dans la figure III.38.

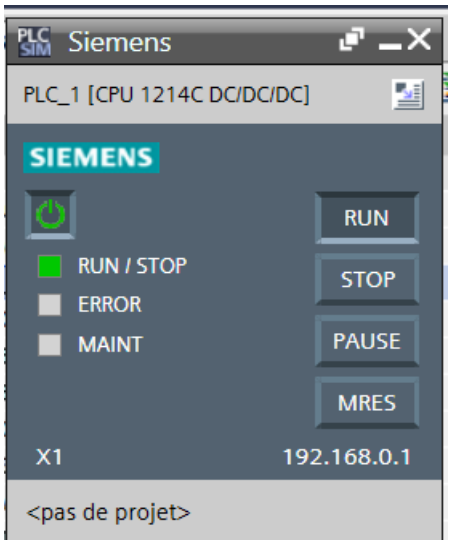


Figure III. 38: PLC SIM

Une fois le programme chargé dans PLC SIM, on le connecte à Factory I/O. Puis on suit les étapes suivantes (voir la figure III.39) :

- On lance Factory I/O.
- Depuis la barre des taches on choisit **fichier** et on sélectionne **drivers.**

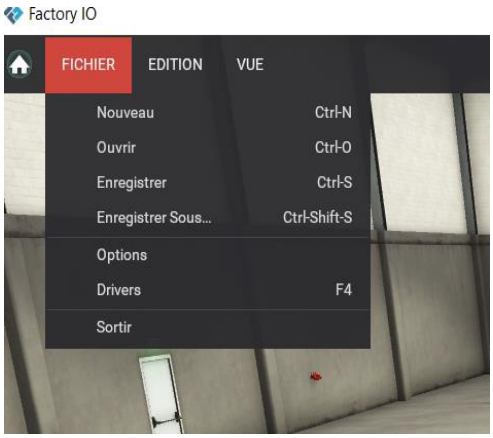


Figure III. 39: Menu drivers

Une interface s'affiche alors nous permet de lier les entrées et les sorties à celles définies dans TIA Portal, à condition qu'elles soient compatibles.

Ensuite en clique sur **connecter** (voir la figure III .40).

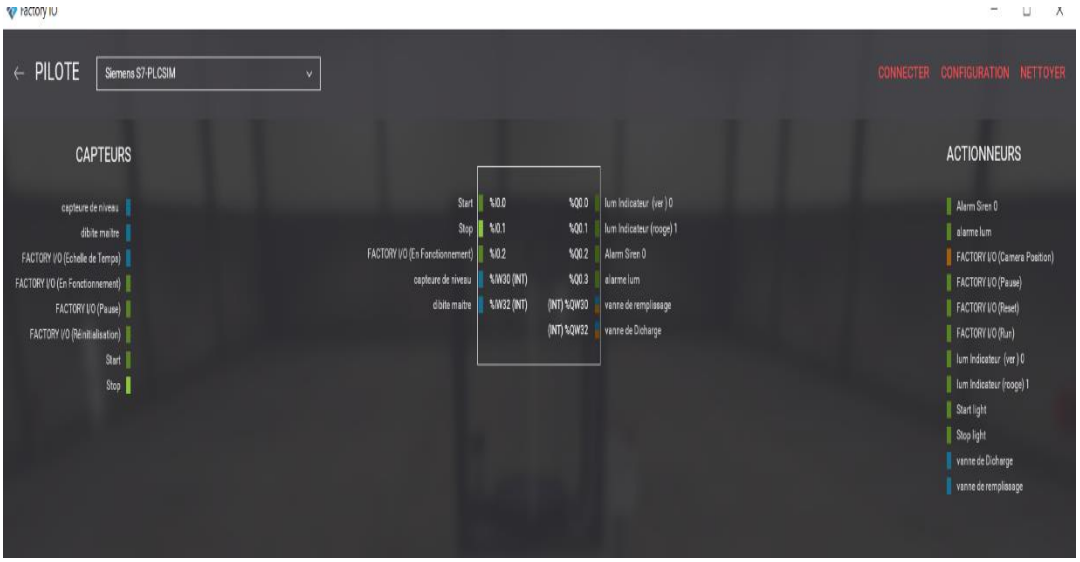


Figure III. 40: Connexion avec PLC SIM

III.7.2 Démarrage de HMI

Pour lancer le HMI on peut cliquer sur l'icône de l'HMI qui est située dans le navigateur de projet puis on clique sur chargé et le HMI se charge automatiquement.

III.7.3 Simulation de système

III.7.3.1 Automatique PID

Depuis l'HMI en sélectionne le mode automatique PID et on configure ensuite les paramètres du régulateur PID. On lance le processus en appuyant sur **start** dans l'objet technologique PID compacte dans TIA Portal. Dans la figure III.41, les trois courbes représentent la SP (courbe rouge) et la PV (courbe verte) et le pourcentage d'ouverture de la vanne de remplissage (courbe noire).

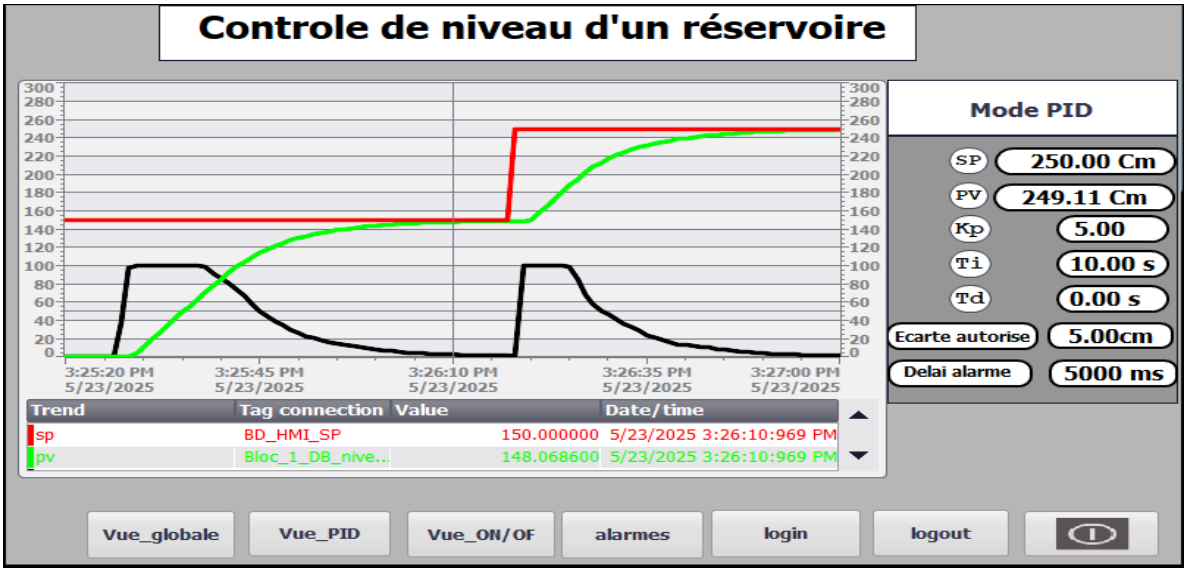


Figure III. 41 : Variation de PV et SP en mode PID



Figure III. 42: Réservoir rempli

III.7.3.2 Automatique ON/OFF

Depuis l'HMI, en sélectionne le mode automatique et le mode ON/OFF. Il est possible de modifier les valeurs d'ouverture et de fermeture de la vanne de remplissage selon le besoin. Dans la figure III.43, les trois courbes représentent le niveau max (courbe rouge) et le niveau min (courbe verte) et le pourcentage d'ouverture de la vanne de remplissage (courbe noire), la mesure (courbe bleue).

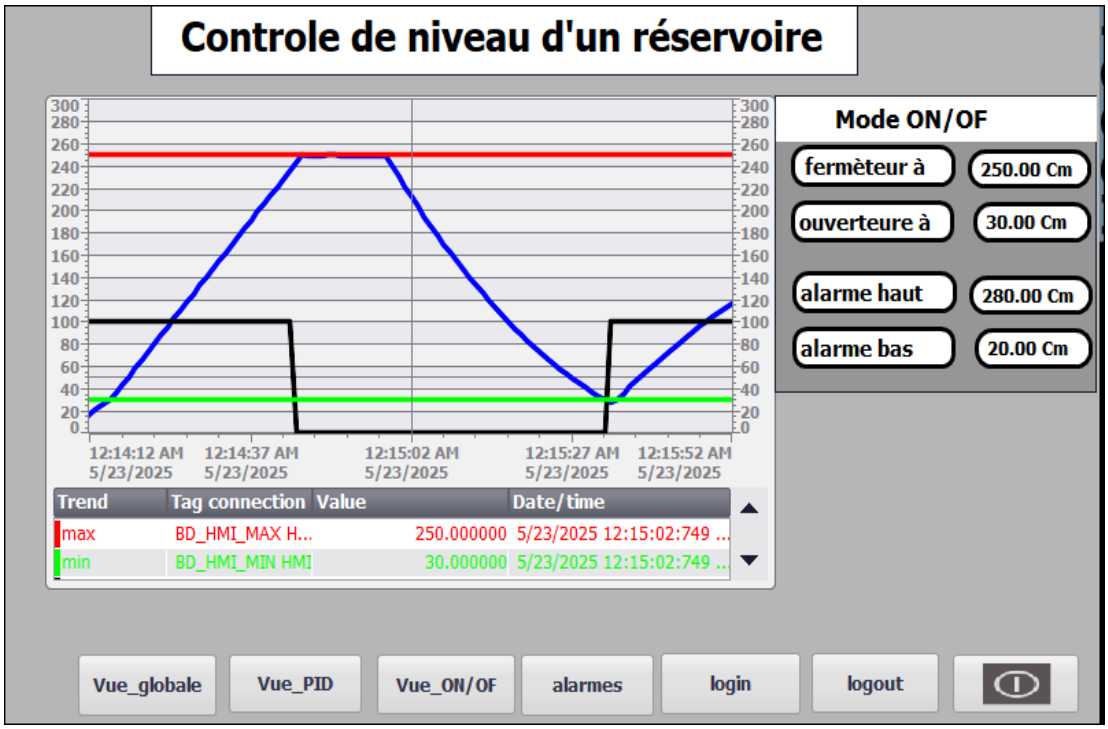


Figure III. 43: Variation de mesure en mode ON/OFF

III.7.3.3 Mode manuel

Depuis l'HMI en sélectionne le mode manuel et on peut contrôler la vanne de remplissage manuellement.

III.7.3.4 Création d'une alarme

La figure III.44 illustre une alarme de niveau haut, signalée par un voyant lumineux et une alarme sonore dans Factory I/O, ainsi que par une indication sur le tableau des alarmes de l'HMI.

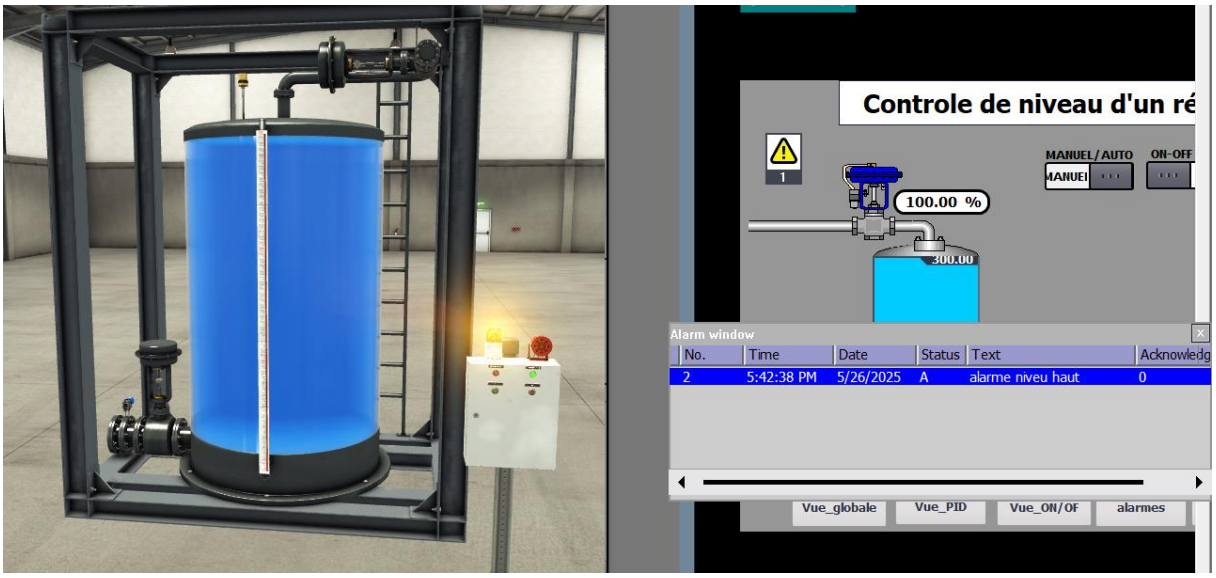


Figure III. 44: Alarme niveau haut

III.8 Conclusion

Ce chapitre a présenté de manière détaillée la conception, l'implémentation et la simulation d'un système de contrôle automatisé de niveau d'eau, basé sur l'intégration des outils logiciels comme TIA Portal pour la programmation et la supervision et Factory I/O pour la modélisation 3D. En effet, nous avons traité le choix des composants, la configuration matérielle de l'API et la programmation des différents modes de fonctionnement (manuel, automatique, ON/OFF, PID), ainsi que le développement de l'interface HMI pour la supervision du procédé.

L'ensemble des fonctionnalités a été validé par la simulation, confirmant la capacité du système à assurer un contrôle efficace du niveau d'eau dans un réservoir. Ainsi, tous les objectifs spécifiés dans le cahier des charges ont été atteints, démontrant que la solution développée est réaliste.

Conclusion Général

Conclusion général

Ce travail de fin d'études se positionne dans le domaine de l'automatisation industrielle, ce qui est un domaine essentiel pour les exigences de modernité des systèmes de production en matière de performance, de fiabilité et de flexibilité.

Dans le cadre de notre étude, nous avons obtenu une solution totale depuis la conception, la programmation, la simulation et la supervision, sur le contrôle de niveau d'eau dans un réservoir à l'aide des outils industriels suivants : TIA portal, Factory I/O et WinCC.

Pour la première partie, nous avons étudié les aspects théoriques relatifs aux fondements des systèmes automatisés, les automates programmables industriels avec une emphase sur la gamme SIMATIC S7-1200 en raison de la modularité et de la performance excellente dans les applications industrielles.

Dans la deuxième partie, nous avons passé en revue les principes de base de la régulation, P, PI et PID et les méthodes de réglage, à savoir la méthode de Ziegler-Nichols en raison de sa large adoption dans l'industrie.

L'approche pratique suivit une progression logique de la première à la dernière étape. Nous avons défini un cahier de charge, des modèles du système en utilisant Factory I/O, et la sélection des capteurs, actionneurs et instruments reliant chacun sinon chaque étape.

Nous avons entamé la mise en œuvre par la programmation d'un mode de remplissage ON/OFF simple. Ensuite, en vue d'améliorer ses performances, nous avons réalisé un régulateur PID plus performant de manière à améliorer la stabilité du système, à réduire l'erreur statique et à rendre plus dynamique le contrôle du niveau de l'eau. Nous avons élaboré en parallèle une interface HMI à travers WinCC, qui permet à l'utilisateur de visualiser les mesures et d'interagir avec le système en temps réel et de manière autonome dans chacun des modes de fonctionnement (manuel, automatique ON/OFF, PID).

Ce projet nous a permis de consolider nos fondements théoriques tout en acquérant des compétences techniques avancées dans le contexte de production et de système usuel. Et encore, ce projet laisse place à d'autres horizons comme l'utilisation de solutions d'intelligence artificielle pour un contrôle prédictif ou d'une connexion avec l'IoT pour la supervision à distance, eu égard aux préceptes de l'industrie 4.0.

Références Bibliographiques

[1] Habbab Mohamed. «Architecture des Systèmes Automatisés», support de cours, Université de Bechar, 2021/2022

[2] Dr. HERIZI Abdelghafour. «Système d'automatisés», support de cours, université de Messila, 2021.

[3] Dr.Taha Benarbia, «Automate programmable industrielle », support de cours, 2023/2024.

[4] Dr .Benfekire. «Interdiction aux Automates programmables », MAI, support de cours, 2022/2023.

[5] https://cabinetnpm.com/instrumentation-automates-programmables-industriels-api/, [Consulté le : 30-05-2025].

[6] Dr. Fares Nafa. «Automate programmable industriel», support de cours et exercices, 2019/2020.

[7] Wiluam Bolton. «Automates programmables industriels», 2ème Edition, 2015.

[8] Dr.Mohsen Ben Ammar. «Automates programmables industriels (Théorie et application) », Document PDF, 2013.

[9] Formation modulaire, «Automatisme, reproduction interdite association ouvrière des compagnons du devoir du tour de France», 2004.

[10] Dr. Okba Kraa. «Automatisme et Informatique Industrielle», support de cours, université de Biskra, 2018/2019.

[11] MICHEL. G. «Les API, Architecture et Application des automates programmables industriels», Dunod, Paris, 1987.

[12] https://www.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-400.html, [Consulté le : 30-05-2025].

[13] Siemens, « S7-1200_System_Manual », Numéro de référence du document : A5E02486682-AG 03/2014.

[14] https://www.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500/cpus.html, [Consulté le : 25-05-2025].

[15] https://media.automation24.com/manual/fr/61777246_s71200_System_Manual_fr-FR_fr-FR.pdf, [Consulté le : 25-05-2025].

[16] Perrin J., Binet F, Dumery J.J, Merlaud C, Trichard J.P. « Automatique et informatique industrielle : Bases théoriques, méthodologiques et technique», Nathan Technique, novembre 2004.

[17] https://sti-monge.fr/maintenancesystemes/wp-content/uploads/2013/02/Initiation-1-TIA-Portal-MS1.pdf, [Consulté le : 25-05-2025].

[18] https://en.niagara.tech/blog/comprendre-le-reseau-profibus, [Consulté le : 25-05-2025],

[19] https://support.industry.siemens.com/cs/attachments/19292127/profinet_system_description_fr-FR_fr-FR.pdf, [Consulté le : 25-05-2025].

[20] SIEMENS, « Manuel de programmation pour S7-1200/S7-1500», 2018–2019.

[21] Dr. AISSANI. «Informatique industrielle», support de cours, MAII, université Oran 2, 2022/2023.

[22] Dr.BENFEKIR. «Supervision industrielle», support de cours, Université Oran 2, 2024/2025.

[23] Ikhlef Boualem. «Contribution à l'Etude de la Supervision Industrielle Automatique dans un Environnement SCADA», Mémoire de Magister, université de boumrdasse, 2009.

[24] Dr. Bouchreb Faouzi. « Supervision industrielle », université Badji Mokhtar Annaba, 2023.

[25] https://cabinetnpm.com/instrumentation-industrielle-regulation-et-asservissement/, [Consulté le : 25-05-2025].

[26] Prouvost Patrick. « Automatique contrôle et régulation», Dunod, France Reims, 2éme édition, 2010.

[27] Dr. Boucerredj Leila. «Régulation Industrielle», support de cours pédagogique, Université 8 Mai 1945 – Guelma, 2021/2022.

[28] Dr.Mekki Ibrahim EL khalil. « Asservissement», support de cours ,2022/2023.

[29] https://mescoursautomatique.blogspot.com/2016/07/,[Consulté le : 30-05-2025].

[30] Toucherift Hocine, Taouint Kocéila. « Etude de la régulation P, PI et PID de débit sur le banc didactique de type 38-001», Mémoire de fin d'étude, 2017.

[31] Yves Granjon. «Automatique systèmes linéaires, non linéaires, à temps continu, à temps discret, représentation d'état», 2ème édition, 2010.

[32] Ghania Boukerche. « Etude et Synthèse d'un Contrôleur PI et Application », Mémoire de fin d'étude, Université Badji Mokhtar-Annaba, 2017.

[33] https://www.technologuepro.com , Régulation industrielle, Représentation Symbolique de la Régulation.pdf, [Consulté le : 30-05-2025].

[34] Dr. Hamida Abdellah. « Instrumentation industrielle», département maintenance en instrumentation», 2023/2024.

[35] Total Formation. «Maintenance instrumentation, le régulateur et ses fonctions», 2008.

[36] Katsuhiko Ogata. «Modern Control Engineering», 5éme edition, 2010.

[37] Tony R. Kuphaldt. «Lessons in Industrial Instrumentation», version 3.01, 2022.

[38] https://docs.factoryio.com/manual/parts/stations/#tank, [Consulté le : 30-05-2025].

[39] https://support.industry.siemens.com/cs/attachments/109765515/sitransl_rtu3030c_fi01_fr.pdf, [Consulté le : 01-06-2025].

Annexes

Annexes

Configuration PID compacte :

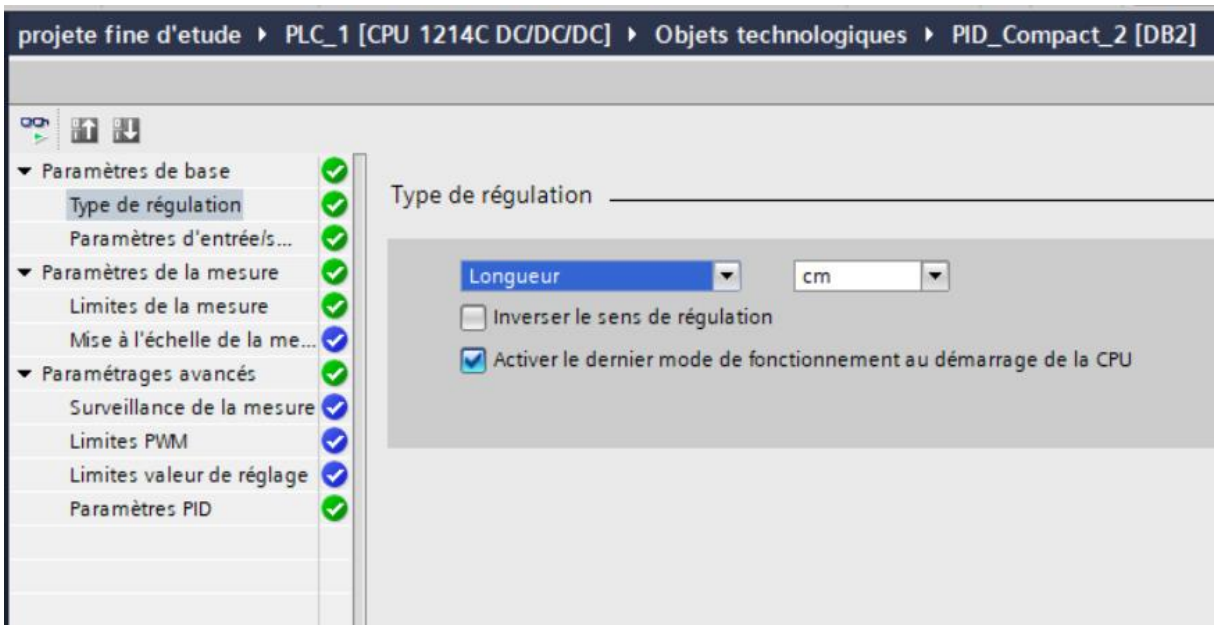


Figure 1 : Choix de type de régulation

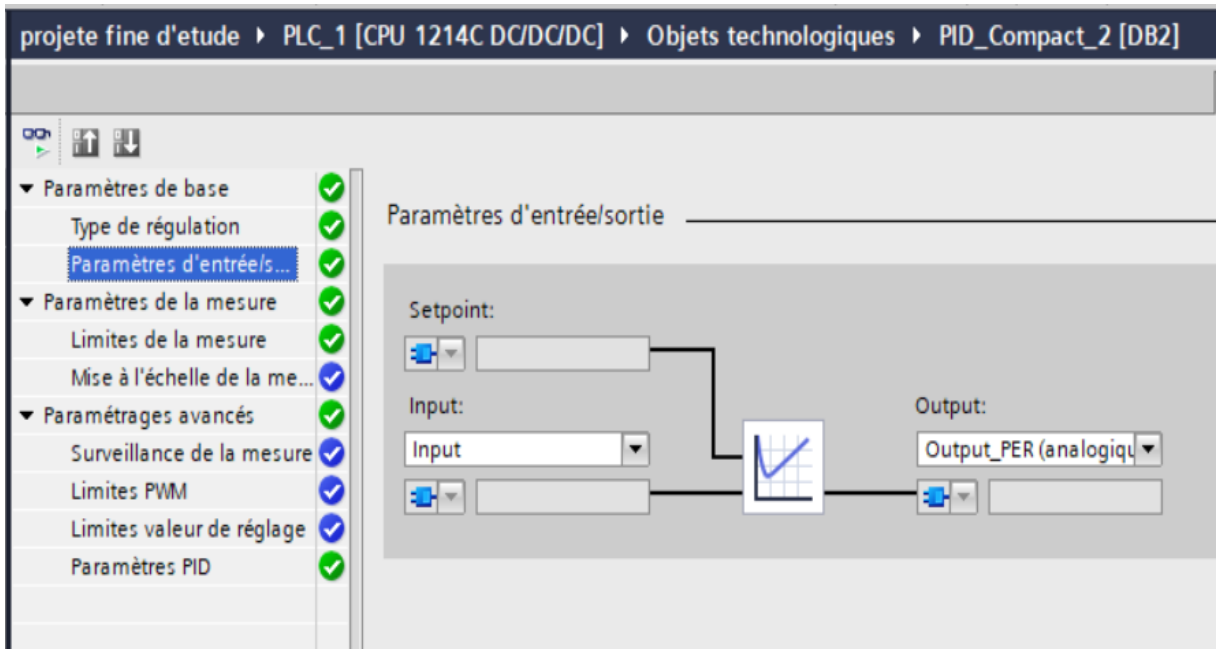


Figure 2 : Paramètres d'entrée/sorties

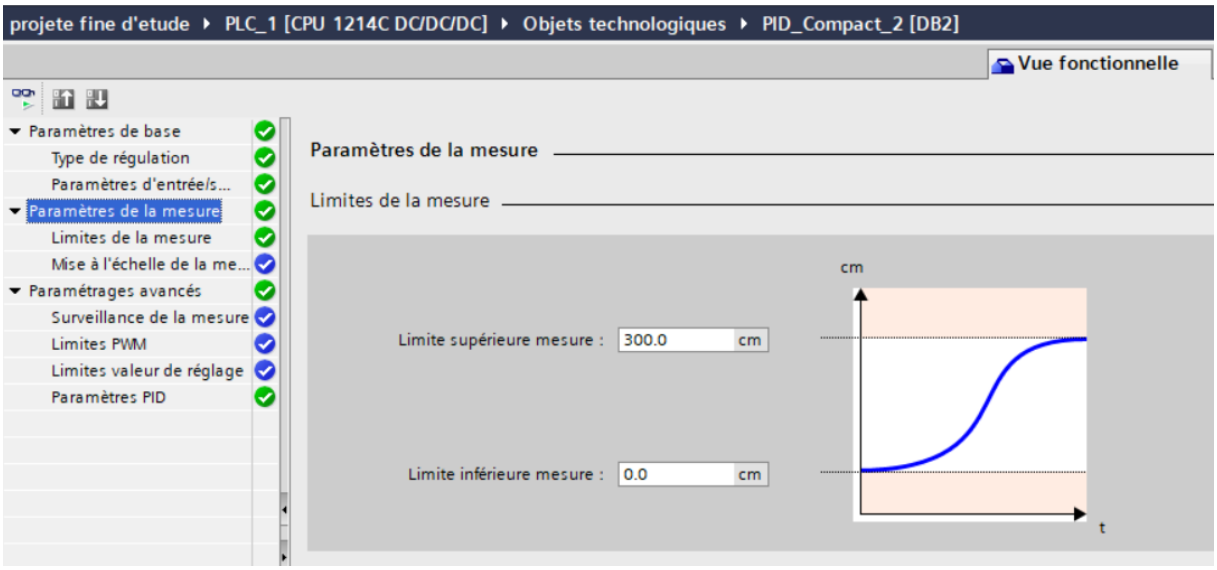


Figure 3 : Limites de la mesure

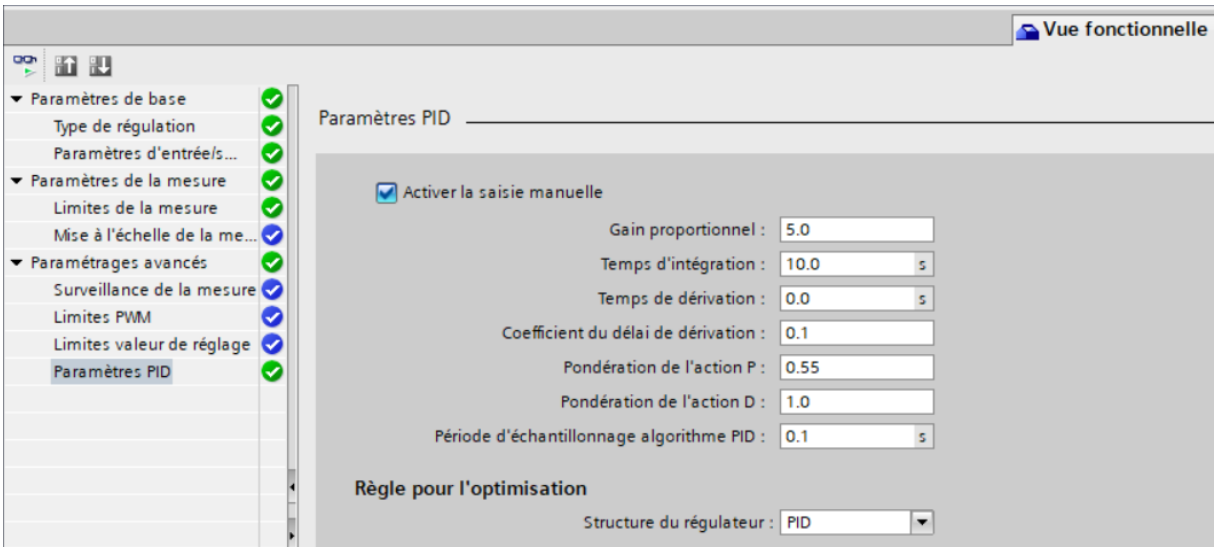


Figure 4 : Paramètres PID