



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

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

معهد الصيانة و الأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Électromécanique
MÉMOIRE
Pour l'obtention du diplôme de Master

Filière : Électromécanique
Spécialité : Électromécanique industrielle

Thème :

Développement d'un Système de surveillance à distance pour les équipements d'instrumentation

Présenté et soutenu publiquement par :

CHOUIREB Abderrahmane
ZENTOU Nedjmeddine

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
BELHADFRI KHEIRA	MAA	IMSI-Univ Oran2	Président
HAIMOUR RACHIDA	MCB	IMSI-Univ Oran2	Encadreur
BENAMAR FATIMA	MAA	IMSI-Univ Oran2	Examineur

Année 2023/2024

Remerciements

Nous remercions tout d'abord Allah le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et le promoteur Madame HAIMOUR RACHIDA, on le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Notre remerciement s'adresse également à tous nos professeurs pour leurs générosités et la grande patience dont ils ont su faire preuve malgré leurs charges académiques et professionnelles.

Nous remercions également les membres de jury pour l'honneur qu'ils nous font de juger notre travail.

Nous remercions nos amis pour toute leurs sincère amitié le long des cinq dernières années d'étude.

Enfin, Nous remercions toute personne ayant contribué de près ou de loin à L'élaboration de ce travail.

Dédicaces

A nos chères mères :

Vous avez su porter pour nous les soins et consentir les efforts pour notre éducation. Aucune dédicace ne saurait exprimer tout le respect et l'amour que nous vous portons, vous nous avez toujours fait confiance. Veuillez trouver en ce travail la consolation et le témoin de la patience.

A nos chers pères :

Malgré les grandes responsabilités que vous assumez dans vos travaux ou autant que pères de familles, vous avez toujours été près de nous, pour nous écouter, nous soutenir, nous suivre et nous encourager. Puisse ce travail diminuer vos souffrances et vous porter bonheur.

A nos chers frères et sœur :

Nous vous réservons la plus grande partie de ce travail. Vous avez toujours été pour nous d'une aide précieuse. Nous vous remercions pour tous les bienfaits que chacun a pu faire pour nous.

A nos familles:

A tous nos amis et camarades de classe, puisse Dieu conserver notre amitié. A tous nos enseignants de l'Institut de Maintenance et Sécurité Industriel A tous ceux qui ont eu et qui ont confiance en nous.

Résumé :

Le mémoire traite de **la surveillance à distance des équipements d'instrumentation** dans les environnements industriels modernes. L'objectif du projet est de concevoir un système capable de surveiller en temps réel des paramètres tels que la température, la pression et les vibrations pour optimiser la maintenance et assurer la sécurité des opérateurs. Le projet utilise des automates programmables industriels (API) comme le **SIMATIC S7 de Siemens** et est programmé via le logiciel **TIA Portal**. Le système de supervision et de contrôle est simulé à l'aide du logiciel **Factory IO**, qui permet de modéliser des systèmes industriels. L'implémentation de ces outils vise à améliorer l'efficacité opérationnelle et à réduire les coûts liés aux défaillances des équipements.

Abstract :

The thesis focuses on **remote monitoring of instrumentation equipment** in modern industrial environments. The project's goal is to design a system capable of real-time monitoring of parameters such as temperature, pressure, and vibrations to optimize maintenance and ensure operator safety. The project uses **programmable logic controllers (PLCs)** like Siemens' **SIMATIC S7**, programmed via the **TIA Portal** software. The supervision and control system is simulated using the **Factory IO** software, which models industrial systems. These tools aim to improve operational efficiency and reduce costs associated with equipment failures.

المخلص :

يتناول هذا البحث المراقبة عن بُعد لمعدات القياس في البيئات الصناعية الحديثة. يهدف المشروع إلى تصميم نظام قادر على مراقبة المتغيرات في الوقت الفعلي مثل درجة الحرارة والضغط والاهتزازات من أجل تحسين الصيانة وضمان سلامة المشغلين. يعتمد المشروع على استخدام المتحكمات المنطقية القابلة للبرمجة (PLC) مثل **SIMATIC S7** من سيمنز، مع برمجته باستخدام برنامج **TIA Portal**. يتم محاكاة نظام الإشراف والتحكم باستخدام برنامج **Factory IO** الذي يسمح بنمذجة الأنظمة الصناعية. تهدف هذه الأدوات إلى تحسين الكفاءة التشغيلية وتقليل التكاليف المرتبطة بأعطال المعدات.

Sommaire :

Remerciements

Dédicaces

Résumé

Sommaire

Liste de Figures

Liste des abréviations

INTRODUCTION GENERALE :..... **12**

I. Chapitre I : La Surveillance à Distance et Les Équipements d’Instrumentation.	13
1. Introduction :.....	14
1.1. Présentation du projet :.....	14
1.2. Objectifs du projet :.....	14
1.3. Contexte et importance du système de surveillance à distance pour les équipements d’instrumentation :	15
2. Étude de l'existant (Conception du Système) :.....	15
2.1. Introduction :.....	15
2.2. Revue de la Littérature sur les Systèmes de Surveillance à distance : ...	15
2.3. Technologies Utilise dans les système de Surveillance à Distance :	15
2.4. Applications de la Surveillance à Distance :	17
2.5. Analyse des Besoins et des Exigences :	17
2.5.1. Besoins Fonctionnels :.....	17
2.5.2. Exigences Non Fonctionnelles :.....	17
2.6. Évaluation des Technologies Disponibles pour la Surveillance à Distance :.....	17
2.7. Analyse des Contraintes Techniques, Financières et Temporelles :.....	18
2.7.1. Contraintes Techniques :.....	18
2.7.2. Contraintes Financières :	18
2.7.3. Contraintes Temporelles :	18
2.8. Cas d'Utilisation du Système :	18
3. Les équipements d’instrumentations :	19

3.1.	Introduction :.....	19
3.2.	Types d'Équipements d'Instrumentation :.....	19
3.2.1.	Capteurs :.....	20
3.2.2.	Transducteurs :.....	24
3.2.2.1.	Types de Transducteurs Utilisés dans l'Industrie.....	24
3.2.3.	Dispositifs de Mesure :.....	25
3.2.4.	Systèmes de Contrôle :.....	26
3.3.	Principes de Fonctionnement :.....	26
3.3.1.	Détection et Mesure :.....	26
3.3.2.	Conversion et Transmission :.....	26
3.3.3.	Contrôle et Régulation :.....	26
3.4.	Applications des Équipements d'Instrumentation.....	27
3.5.	Importance des Équipements d'Instrumentation.....	27
4.	Conclusion :.....	28
II.	Chapitre II : Les Automates Programmable Industriel.....	29
1.	Introduction :.....	30
2.	L'Automatisme :.....	30
3.	Objectif de l'automatisation :.....	30
4.	Les systèmes automatisés :.....	31
4.1.	La partie opérative (PO) :.....	31
4.2.	La partie commande (PC) :.....	32
4.3.	La partie relation (PR) :.....	32
5.	Historique Des automates programmables :.....	32
6.	Définition de l'Automate programmable :.....	33
7.	Mise en œuvre d'un automate :.....	33
8.	Programmation des automates :.....	33
9.	Critères de choix d'un automate :.....	33
10.	Traitement du programme automate :.....	34
11.	Langage de programmation pour API :.....	35
11.1.	Le langage LD (Ladder Diagram) :.....	35
11.2.	Le langage IL (Instruction List) :.....	35
11.3.	Le langage FBD (Function Block Diagram) :.....	36

11.4.Le langage ST (Structured Text) :.....	36
11.5.Le langage SFC (Sequential Function Chart), ou GRAFCET :.....	36
12. Les avantages et les inconvénients de l’automatisation :.....	37
12.1. Les avantage :	37
12.2. Les inconvénients :.....	37
13. Domaines d'application des systèmes automatisés :	37
15. Structure interne des automates programmables :.....	38
15.1. Le processeur :.....	38
15.2. Les mémoires :.....	39
15.3. Les modules entrées/sorties :.....	39
15.4. L’alimentation :.....	40
15.5. Liaisons de communication :.....	40
III. Chapitre III : Logiciel TIA Portal	41
1. Introduction :.....	42
2. Vue du portail et vue de projet :.....	42
2.1. Vue du portail :.....	42
2.2. Vue du projet :.....	43
3. Création d’un projet :	44
3.1. identification du matériel :.....	45
3.2. Configuration et paramétrage du matériel :.....	46
3.3. Les variables API :	46
3.3.1. Adresses symbolique et absolue :.....	46
3.3.2. Adressage des E/S :.....	47
3.4. Table des variables API :	48
3.5. Tableau d’affectation :.....	49
3.6. Les références croisées :.....	50
4. Les blocs dans TIA Portal :.....	50
4.1. Les blocs d’organisation – OB :.....	50
4.2. Les fonctions – FC :	50
4.3. Les blocs fonctionnels – FB :.....	50
4.4. Les blocs de données – DB :.....	50
4.5. Création et programmation d’un bloc :.....	51

4.6.	Appel d'un bloc l'OB1 :	52
5.	Les langages de programmation dans le TIA Portal :	52
6.	Compilation et chargement :	52
7.	La simulation avec PLCSIM :	53
8.	Supervision du processus :	54
8.1.	Présentation du WinCC sous TIA Portal :	54
8.2.	Définition de l'interface homme-machine (HMI) :	54
8.2.1.	Représentation du processus :	55
8.2.2.	Commande du processus :	55
8.2.3.	Vue des alarmes :	55
8.2.4.	Archivage de valeurs processus et d'alarmes :	55
8.2.5.	Documentation de valeurs processus et d'alarmes :	55
8.2.6.	Gestion des paramètres de processus et de machine :	55
8.3.	Configuration du HMI dans le TIA portal :	55
8.4.	Éléments de l'interface utilisateur de WinCC :	57
8.5.	Définition des variables :	58
8.5.1.	Les variables externes :	58
8.6.	Les alarmes dans WinCC :	58
8.6.1.	Classes d'alarmes :	59
8.6.2.	Types des alarmes :	59
8.6.3.	Configuration d'alarme :	60
8.6.3.1.	Editeur "Alarmes TOR" :	60
8.6.3.2.	Éditeur "Alarmes Analogiques" :	60
8.6.4.	Etats des alarmes :	61
8.7.	Les recettes :	61
8.7.1.	Editeur de recette :	62
8.8.	Les archives :	63
8.8.1.	Archives d'alarmes	63
8.8.2.	Archives de variables	64
8.8.3.	Les méthodes d'archivage	64
8.9.	Le test de la visualisation du processus :	65
9.	Conclusion :	66

IV. Chapitre IV : La Simulation et La Supervision.	67
1. Introduction :	68
2. Présentation de Factory IO :	68
3. Étapes de Configuration et de Connexion :	68
4. Conclusion :	79
CONCLUSION GENERALE :	80
RÉFÉRENCE BIBLIOGRAPHIES :	81

LISTE DE FIGURES :

Figure I.1 : Schéma de l'Internet des Objets (IoT) dans l'industrie	16
Figure I.2 : Schéma de l'infrastructure Cloud Computing	16
Figure I.3 : Schéma de Surveillance dans L'industrie Pétrolière et Gazière	18
Figure I.4 : Système de commande et de protection d'une centrale hydroélectrique.	19
Figure I.5 : capteur de température Thermocouple	20
Figure I.6 : capteur de température Thermistance	20
Figure I.7 :Capteur de pression (Manomètre digital)	21
Figure I.8 : Capteur de pression piézorésistif	21
Figure I.9 : capteur de niveau capacitif.	22
Figure I.10 : capteur de niveau ultrasons.....	22
Figure I.11 : débitmètres à turbine.	23
Figure I.12 : débitmètre à effet vortex.....	23
Figure I.13 : Encodeur.	23
Figure I.14 : Potentiomètres.....	24
Figure I.15 : Transducteur de Pression	24
Figure I.16 : Transducteurs de Température	25
Figure I.17 : Transducteurs de Déplacement	25
Figure I.18 : Structure interne d'un automates programmables industriels (PLC)	27
Figure II.1 : Structure d'un système automatisé.	31
Figure II.2 : Fonctionnement cyclique d'un automate.	34
Figure II.3 : Structure interne d'un API.....	38
Figure III.1 : Vue de Portail.	43
Figure III.2 : Vue de Projet.	44
Figure III.3 : Vue création de projet.	45
Figure III.4 : Identification d'un module.	45
Figure III.5 : Onglet d'information.	46
Figure III.6 : Exemple d'adresse symbolique et physique.	47
Figure III.7 : Vue des appareils.	48
Figure III.8 : Table de Variable.	49
Figure III.9 : Table d'affectation.....	49
Figure III.10 : Vue des blocs disponible.	51
Figure III.11 : Programme d'un bloc.	52

Liste des Figures

Figure III.12 : Appel d'un bloc dans OB1.....	52
Figure III.13 : Les étapes de compilation.	53
Figure III.14 : Etat de la CPU.	53
Figure III.15 : Schema explicative.	54
Figure III.16 : Vue configuration HMI.....	56
Figure III.17 : Assistant HMI.....	56
Figure III.18 : Vue d'interface WinCC.	57
Figure III.19 : Vue des alarmes HMI.....	58
Figure III.20 : Vue classe d'alarme.....	59
Figure III.21 : Vue de configuration l'alarme analogique.....	61
Figure III.22 : Vue configuration de recette.....	63
Figure III.23 : Lancement du RUNTIME.....	66
Figure IV.1 : Logiciel Factory IO.....	68
Figure IV.2 : Ouverture de TIA Portal.....	69
Figure IV.3 : Ouvrir le projet.	69
Figure IV.4 : Vue de Projet.	70
Figure IV.5 : Commencer la simulation.....	70
Figure IV.6 : Charger le PLC.	71
Figure IV.7 : S7-PLCSIM.....	71
Figure IV.8 : Commencer la simulation de L'IHM.	72
Figure IV.9 : Vue de L'IHM sur RUNTIME.....	72
Figure IV.10 : Ouvrir Logiciel Factory IO.....	73
Figure IV.11 : Controle de Niveau.	73
Figure IV.12 : Selectionner "Drivers".....	74
Figure IV.13 : Choisir Siemens S7-PLCSIM.....	74
Figure IV.14 : Appuyer sur "Configuration".....	75
Figure IV.15 : La configuration des adresses I/O.....	75
Figure IV.16 : Schema des I/O et Variable de Simulation dans Factory IO.	76
Figure IV.17 : Connecter Factory IO avec PLCSIM.....	76
Figure IV.18 : Commencer la simulation de Factory IO.	77
Figure IV.19 : Ouvrir la vanne remplissage.....	77
Figure IV.20 : Ouvrir la vanne de décharge.....	78
Figure IV.21 : Vue d'Alarme.	78

LISTE DES ABRÉVIATIONS

LISTE DES ABRÉVIATIONS :

API : Automate Programmable Industriel.

PLC : Programmable Logic Controller (Automate Programmable).

DCS : Distributed Control System (Système de contrôle distribué).

TIA : Totally Integrated Automation (Automatisation totalement intégrée).

IHM/HMI : Interface Homme-Machine.

SCADA : Supervisory Control and Data Acquisition (Système de contrôle et d'acquisition de données).

CPU : Central Processing Unit (Unité centrale de traitement).

ROM : Read-Only Memory (Mémoire morte).

RAM : Random Access Memory (Mémoire vive).

EEPROM : Electrically Erasable Programmable Read-Only Memory.

IoT : Internet of Things (Internet des objets).

DB : Data Block (Bloc de données).

OB : Organisation Block (Bloc d'organisation).

FB : Function Block (Bloc fonctionnel).

FC : Function Call (Appel de fonction).

WSN : Wireless Sensor Network (Réseau de capteurs sans fil).

RT : Run Time (Temps d'exécution).

PLCsim : Simulation de PLC.

SCL : Structured Control Language (Langage de contrôle structuré).

WINCC : Windows Control Center (Logiciel de supervision industrielle).

MPI : Multi-Point Interface (Interface multi-point).

HMI RT : Human Machine Interface Runtime (Interface Homme-Machine en temps

INTRODUCTION GENERALE :

Dans un contexte industriel en constante évolution, les technologies de l'automatisation et de la surveillance à distance jouent un rôle clé dans l'optimisation des opérations. Les systèmes modernes, tels que les automates programmables et les solutions de simulation, permettent aux entreprises de mieux gérer leurs équipements d'instrumentation tout en améliorant la productivité et en réduisant les coûts de maintenance.

Ce mémoire se concentre sur l'utilisation de "TIA Portal" pour la programmation et la gestion des automates, et de "Factory IO" pour la simulation de systèmes automatisés. En combinant ces outils, il est possible de créer des environnements virtuels qui imitent des conditions industrielles réelles, facilitant ainsi la conception et l'optimisation des processus avant leur mise en œuvre. L'objectif de cette étude est de développer un système de surveillance à distance qui intègre la programmation avancée et la simulation en temps réel pour assurer une gestion efficace des équipements d'instrumentation.

Chapitre I : Généralité sur la surveillance à distance et les équipements d'instrumentation .

Chapitre II : Généralité sur les automates programmable industriel

Chapitre III : L'explication de Logiciel TIA Portal

Chapitre IV : L'explication de logiciel Factory IO et les étapes de simulation, la connexion de TIA Portal avec Factory IO .

**I. Chapitre I : La
surveillance à distance et
les équipements
d'instrumentation.**

1.Introduction :

Dans le paysage industriel contemporain où la technologie et l'innovation jouent un rôle de premier plan, le développement de systèmes de surveillance à distance des équipements d'instruments est impératif. Face aux enjeux croissants de maintenance et de suivi des équipements, la mise en œuvre de solutions innovantes et efficaces devient une priorité indéniable.

Thèse : Cette étude est basée sur la thèse : La mise en place d'un système de surveillance à distance des équipements d'instruments est d'une grande importance pour optimiser les opérations industrielles, réduire les coûts de maintenance et assurer la sécurité des opérateurs.

1.1. Présentation du projet :

Le projet vise à concevoir et développer un système innovant de surveillance à distance des équipements d'instrumentation qui permet une surveillance continue et en temps réel des paramètres clés de l'équipement, quelle que soit sa situation géographique. L'objectif principal est de fournir des solutions puissantes et efficaces pour améliorer les capacités de réponse aux incidents, réduire les coûts de maintenance et optimiser les performances opérationnelles.

Dans les environnements où les équipements d'instrumentation sont souvent dispersés sur de grands sites ou difficiles d'accès, de tels systèmes de surveillance à distance offrent une solution prometteuse pour surmonter les défis associés à la surveillance manuelle ou périodique actuelle.

1.2. Objectifs du projet :

- Concevoir un système de surveillance capable de collecter diverses données de paramètres telles que la température, la pression, les vibrations, etc. à partir de l'équipement d'instruments.
- Établir une infrastructure de communication fiable pour transmettre ces données au centre de contrôle à distance en temps réel.
- Développer une interface conviviale pour visualiser les données collectées et prendre des décisions éclairées concernant la maintenance et les opérations.
- Vérifier l'efficacité et la fiabilité du système grâce à des tests approfondis en laboratoire et sur le terrain.

- Discuter des implications éthiques et juridiques de la surveillance à distance des équipements industriels et proposer des mesures pour garantir la confidentialité et la sécurité des données.

1.3. Contexte et importance du système de surveillance à distance pour les équipements d'instrumentation :

Dans de nombreux secteurs industriels, la surveillance et la maintenance des équipements d'instrumentation sont essentielles pour assurer le bon fonctionnement des processus. Cependant, les méthodes traditionnelles de surveillance peuvent être limitées par leur capacité à fournir des données en temps réel et à permettre une intervention rapide en cas de problème. Pour répondre à ces défis, ce projet propose le développement d'un système de surveillance à distance innovant.

L'importance de la surveillance à distance des équipements d'instrumentation réside dans sa capacité à améliorer la productivité, la fiabilité et la sécurité des processus industriels. En permettant une surveillance continue et en temps réel des équipements, ce système offre la possibilité de détecter les anomalies et les défaillances potentielles avant qu'elles ne se transforment en problèmes majeurs. Cette approche proactive permet non seulement de réduire les temps d'arrêt non planifiés, mais aussi d'optimiser les performances opérationnelles et de garantir la conformité aux normes de sécurité et de qualité.

2. Étude de l'existant (Conception du Système) :

2.1. Introduction :

Les systèmes de surveillance à distance des équipements d'instrumentation sont devenus essentiels dans divers secteurs industriels pour améliorer la fiabilité, l'efficacité et la sécurité des opérations. Cette section offre une revue de la littérature sur les systèmes de surveillance existants, en mettant l'accent sur les technologies utilisées, les besoins et les exigences, ainsi que les contraintes techniques, financières et temporelles associées à leur mise en œuvre.

2.2. Revue de la Littérature sur les Systèmes de Surveillance à distance :

Les systèmes de surveillance à distance (Remote Monitoring Systems - RMS) permettent de collecter, analyser et visualiser des données provenant d'équipements d'instrumentation dispersés géographiquement. Ces systèmes utilisent généralement des technologies de communication sans fil et des plateformes basées sur le cloud pour offrir un accès en temps réel aux données critiques.

2.3. Technologies Utilisées dans le système de Surveillance à Distance :

Chapitre I

Les technologies couramment utilisées dans les systèmes de surveillance à distance incluent :

- **Internet des Objets (IoT) :** Permet la connexion de divers capteurs et dispositifs à un réseau centralisé pour la collecte de données.

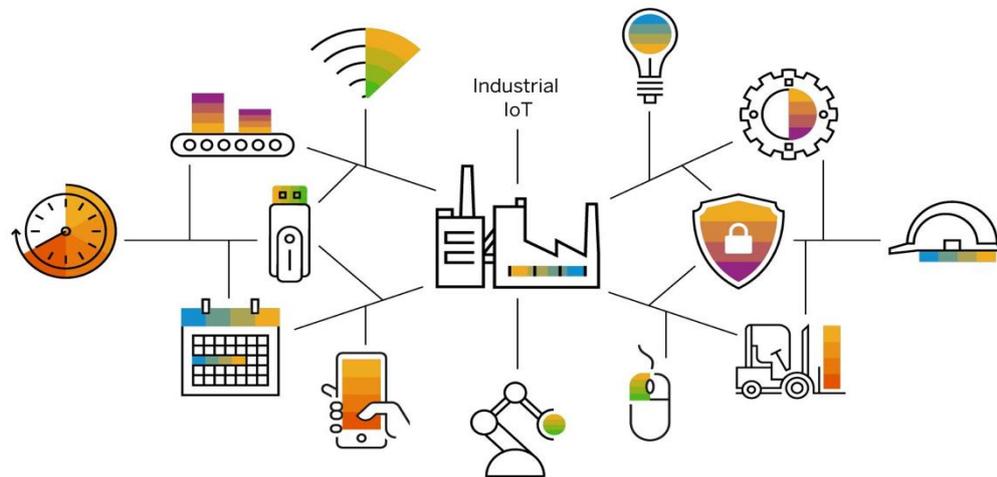


Figure I.1 : Schéma de l'Internet des Objets (IoT) dans l'industrie

- **Big Data et Analytique :** Utilisés pour analyser de grandes quantités de données en temps réel et fournir des informations exploitables.
- **Cloud Computing :** Fournit une infrastructure scalable et flexible pour le stockage et le traitement des données.



Figure I.2 : Schéma de l'infrastructure Cloud Computing

- **Intelligence Artificielle (IA) et Apprentissage Automatique** : Utilisés pour prédire les défaillances et optimiser les processus de maintenance. [1]

2.4. Applications de la Surveillance à Distance :

La surveillance à distance est appliquée dans divers secteurs industriels, y compris l'énergie, la fabrication, la santé et l'agriculture. Par exemple, dans l'industrie énergétique, les systèmes de surveillance à distance sont utilisés pour surveiller les performances des turbines, des panneaux solaires et des réseaux de distribution. Dans la fabrication, ils sont essentiels pour maintenir la qualité et la fiabilité des lignes de production. [2]

2.5. Analyse des Besoins et des Exigences :

2.5.1. Besoins Fonctionnels :

Les besoins fonctionnels d'un système de surveillance à distance incluent la capacité à :

- Collecter des données en temps réel à partir de divers capteurs (température, pression, vibrations, etc.).
- Transmettre ces données à un centre de contrôle à distance via une infrastructure de communication fiable.
- Analyser les données collectées pour détecter des anomalies et prévoir les défaillances potentielles.
- Générer des alertes et des rapports pour faciliter la prise de décision.

2.5.2. Exigences Non Fonctionnelles :

Les exigences non fonctionnelles comprennent :

- La fiabilité et la robustesse du système pour assurer une surveillance continue.
- La sécurité des données pour protéger les informations sensibles contre les cyberattaques.
- La scalabilité pour permettre l'ajout de nouveaux équipements et capteurs.
- La facilité d'utilisation pour assurer une adoption rapide par les opérateurs. [3]

2.6. Évaluation des Technologies Disponibles pour la Surveillance à Distance :

Diverses technologies sont disponibles pour la mise en œuvre de systèmes de surveillance à distance :

- Réseaux de Capteurs Sans Fil (WSN) : Utilisés pour collecter des données à partir de capteurs distribués.
- Protocoles de Communication (e.g., MQTT, CoAP) : Facilitent la transmission efficace des données entre les capteurs et le centre de contrôle.

Chapitre I

- Systèmes de Gestion de Données (DMS) : Stockent et organisent les données collectées pour une analyse facile. [4]

2.7. Analyse des Contraintes Techniques, Financières et Temporelles :

2.7.1. Contraintes Techniques :

Les contraintes techniques incluent la compatibilité des capteurs avec les équipements existants, l'interopérabilité des systèmes, et la robustesse des solutions de communication face aux interférences environnementales.

2.7.2. Contraintes Financières :

Les contraintes financières concernent le coût des capteurs, des infrastructures de communication, des plateformes de traitement des données, et des licences logicielles. L'analyse du retour sur investissement (ROI) est cruciale pour justifier les dépenses initiales.

2.7.3. Contraintes Temporelles :

Les contraintes temporelles incluent les délais de mise en œuvre, les phases de test et de validation, et les périodes d'arrêt nécessaires pour l'installation des systèmes. Une planification minutieuse est essentielle pour minimiser les interruptions de service.

2.8. Cas d'Utilisation du Système :

Les cas d'utilisation des systèmes de surveillance à distance incluent :

- Industrie Pétrolière et Gazière : Surveillance des pipelines et des plateformes de forage.

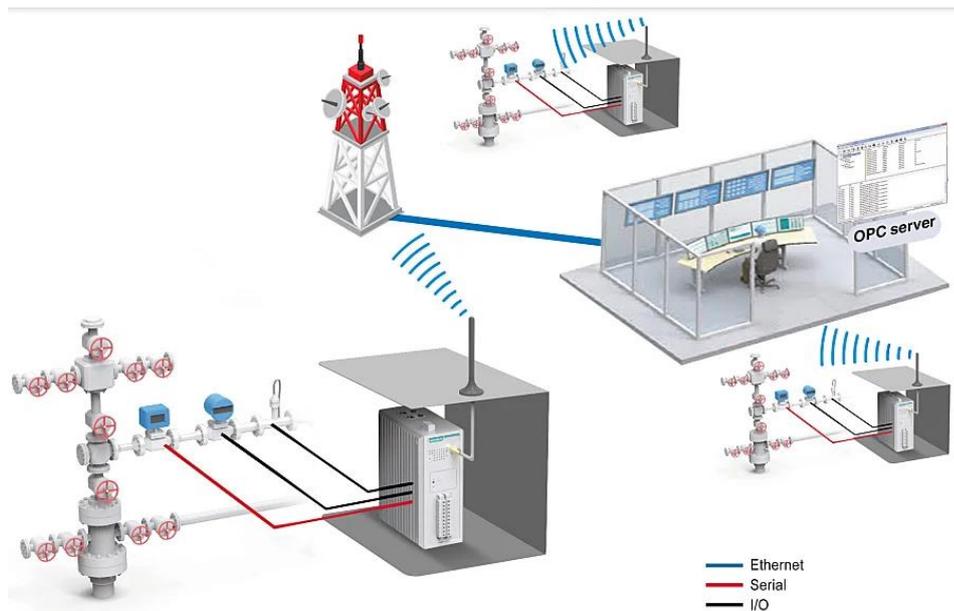


Figure I.3 : Schéma de Surveillance dans L'industrie Pétrolière et Gazière .

Chapitre I

- Production d'Énergie : Gestion des centrales électriques et surveillance des turbines.

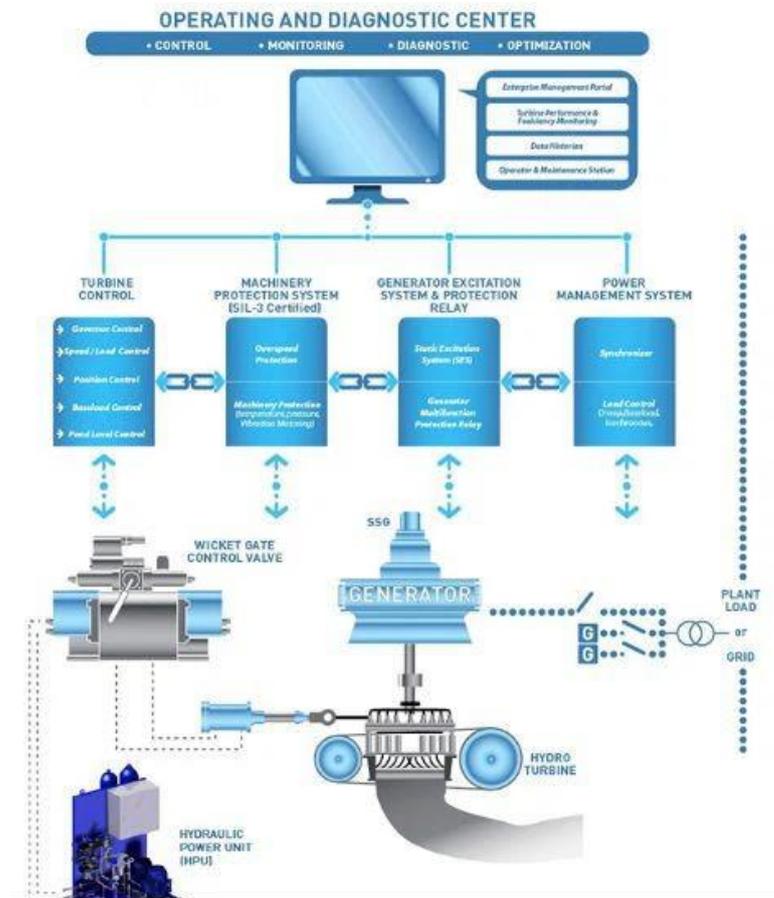


Figure I.4 : Système de commande et de protection d'une centrale hydroélectrique.

- Industrie Chimique : Contrôle des réactions chimiques et des équipements de production.
- Fabrication : Automatisation des lignes de production et contrôle de la qualité. [6]

3. Les équipements d'instrumentations :

3.1. Introduction :

Les équipements d'instrumentation sont des composants essentiels dans de nombreux secteurs industriels. Ils permettent de surveiller, de mesurer et de contrôler divers paramètres physiques et chimiques afin d'assurer le bon fonctionnement des processus industriels. Ce chapitre fournit une vue d'ensemble générale des équipements d'instrumentation, en mettant en lumière leurs types, leurs principes de fonctionnement, leurs applications et leur importance dans l'industrie moderne.

3.2. Types d'Équipements d'Instrumentation :

Chapitre I

Les équipements d'instrumentation se divisent en plusieurs catégories principales, chacune ayant des fonctions spécifiques et des applications variées :

3.2.1. Capteurs :

Les capteurs sont des dispositifs qui détectent des changements dans l'environnement physique et les convertissent en signaux électriques ou optiques. Les types de capteurs couramment utilisés incluent :

- **Capteurs de température** : Utilisés pour mesurer la température des solides, des liquides et des gaz.

Exemples : thermocouples, thermistances.



Figure I.5 : capteur de température Thermocouple



Figure I.6 : capteur de température Thermistance

- **Capteurs de pression** : Mesurent la pression des gaz ou des liquides.

Exemples : manomètres, capteurs piezorésistifs.



Figure I.7 :Capteur de pression (Manomètre digital)



Figure I.8 : Capteur de pression piézorésistif

- **Capteurs de niveau** : Détectent le niveau des liquides ou des solides dans les réservoirs.

Exemples : capteurs capacitifs, capteurs à ultrasons.



Figure I.9 : capteur de niveau capacitif.



Figure I.10 : capteur de niveau ultrasons

- **Capteurs de débit** : Mesurent le débit des liquides et des gaz.
Exemples : débitmètres à turbine, débitmètres à effet vortex.



Figure I.11 : débitmètres à turbine.



Figure I.12 : débitmètre à effet vortex.

- **Capteurs de position** : Déterminent la position ou le déplacement des objets.

Exemples : potentiomètres, encodeurs.[7]



Figure I.13 : Encodeur.



Figure I.14 : Potentiomètres

3.2.2. Transducteurs :

Les transducteurs convertissent une forme d'énergie en une autre. Dans le contexte de l'instrumentation, ils sont souvent utilisés pour transformer les signaux physiques détectés par les capteurs en signaux électriques. Par exemple, un microphone convertit les ondes sonores en signaux électriques. [8]

3.2.2.1. Types de Transducteurs Utilisés dans l'Industrie

- **Transducteurs de Pression :**

Ces dispositifs convertissent la pression physique en un signal électrique. Ils sont cruciaux dans les industries où le contrôle précis de la pression est nécessaire pour assurer la qualité et la sécurité des processus.



Figure I.15 : Transducteur de Pression

- **Transducteurs de Température :**

Parmi ces transducteurs, les plus communs sont les capteurs à résistance de platine (RTD) et les thermocouples. Les RTDs offrent une grande précision et sont utilisés dans les applications nécessitant une mesure de température précise et stable.



Figure I.16 : Transducteurs de Température

- **Transducteurs de Déplacement :**

Ils convertissent le mouvement ou le déplacement d'un objet en un signal électrique. Utilisés dans les applications de mesure de position et de vitesse, ces transducteurs sont essentiels pour la robotique et l'automatisation. [9]



Figure I.17 : Transducteurs de Déplacement

3.2.3. Dispositifs de Mesure :

Ces dispositifs sont utilisés pour quantifier les données collectées par les capteurs. Ils peuvent afficher, enregistrer et transmettre les mesures. Exemples :

- Multimètres : Mesurent la tension, le courant et la résistance électriques.
- Oscilloscopes : Visualisent les variations de signaux électriques dans le temps.

- Analyseurs de gaz : Mesurent la composition des gaz. [10]

3.2.4. Systèmes de Contrôle :

Les systèmes de contrôle régulent les processus industriels en utilisant les données collectées par les capteurs. Ils peuvent être manuels, automatiques ou semi-automatiques.

Exemples :

- **Contrôleurs logiques programmables (PLC) :** Utilisés pour automatiser les processus industriels.
- **Systèmes de contrôle distribués (DCS) :** Gèrent les opérations complexes dans les grandes installations industrielles. [11]

3.3. Principes de Fonctionnement :

3.3.1. Détection et Mesure :

Les capteurs et les transducteurs détectent des phénomènes physiques (comme la température, la pression, le niveau, etc.) et les transforment en signaux électriques proportionnels à la grandeur mesurée. Ces signaux sont ensuite interprétés par les dispositifs de mesure et les systèmes de contrôle. [12]

3.3.2. Conversion et Transmission :

Les transducteurs jouent un rôle clé dans la conversion des signaux physiques en signaux électriques, facilitant ainsi leur transmission et leur traitement. Les dispositifs de mesure amplifient, conditionnent et affichent ces signaux pour permettre une surveillance efficace. [13]

3.3.3. Contrôle et Régulation :

Les systèmes de contrôle utilisent les données des capteurs pour prendre des décisions et réguler les processus industriels. Les PLC et DCS sont programmés pour maintenir les paramètres du processus dans des plages prédéfinies, assurant ainsi un fonctionnement sûr et efficace. [14]

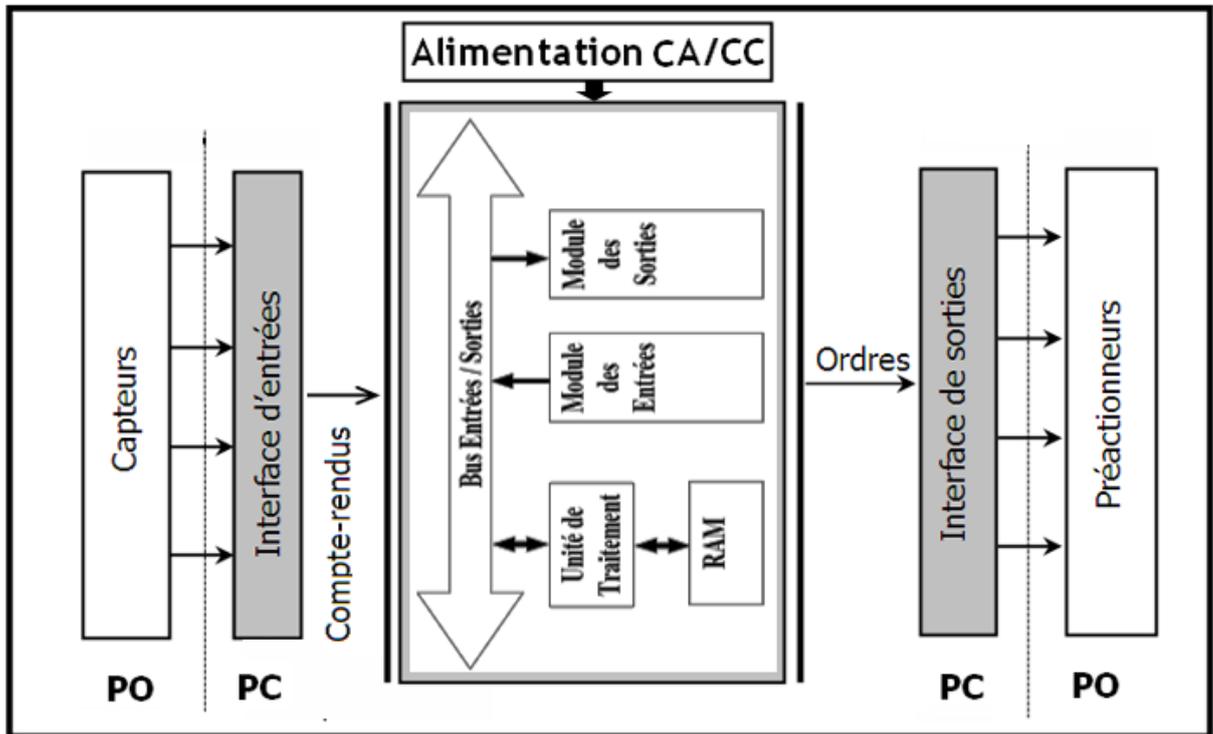


Figure I.18 : Structure interne d'un automates programmables industriels (PLC)

3.4. Applications des Équipements d'Instrumentation

Les équipements d'instrumentation trouvent des applications dans divers secteurs, notamment :

- **Industrie pétrolière et gazière** : Surveillance des pipelines, des réservoirs et des plateformes de forage.
- **Production d'énergie** : Gestion des centrales électriques, surveillance des turbines et des générateurs.
- **Industrie chimique** : Contrôle des réactions chimiques, surveillance des réacteurs et des colonnes de distillation.
- **Fabrication** : Automatisation des lignes de production, contrôle de la qualité des produits.
- **Services publics** : Gestion des systèmes de distribution d'eau et d'électricité. [15]

3.5. Importance des Équipements d'Instrumentation

Les équipements d'instrumentation sont cruciaux pour garantir la sécurité, l'efficacité et la fiabilité des processus industriels. Ils permettent :

- **La détection précoce des anomalies** : Réduisant ainsi les risques de pannes et d'accidents.

- **L'optimisation des opérations** : En améliorant l'efficacité énergétique et en réduisant les coûts de fonctionnement.
- **La maintenance préventive** : En facilitant la planification des interventions de maintenance avant que les pannes ne surviennent.
- **La conformité réglementaire** : En assurant que les processus respectent les normes de sécurité et environnementales.[16]

4.Conclusion :

Les équipements d'instrumentation, grâce à leur capacité à mesurer et à contrôler précisément divers paramètres, jouent un rôle clé dans le maintien de la qualité et de la fiabilité des processus. La surveillance à distance transforme radicalement les stratégies de maintenance, passant d'approches réactives à des méthodes proactives et prédictives. Ceci est crucial pour les industries où la continuité des opérations est essentielle et où les coûts associés aux temps d'arrêt sont considérables.

En conclusion, le système de surveillance à distance ne se contente pas de répondre aux exigences actuelles des industries modernes en matière de performance et de sécurité; il établit également un fondement solide pour l'avenir de la gestion industrielle, où la connectivité, l'intelligence et l'automatisation convergent pour créer des environnements de production plus résilients et efficaces.

II. Chapitre II : Les automates programmable industriel

1. Introduction :

Les automates programmables sont parus aux USA vers 1969 grâce à MODICON qui créa le premier automate programmable. Son succès donna naissance à une industrie mondiale qui s'est considérablement développée depuis.

De ce fait plusieurs constructeurs sont apparus sur le marché (SIEMENS, SCHNEIDER, TOSHIBA, ... etc.) Produisant différentes variétés d'automates qui utilisent des langages de programmation différents.

Aujourd'hui, l'API est le constituant le plus répondu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie, car il répond à tous les besoins d'adaptation et de flexibilité pour un grand nombre d'opérations. Cette émergence est due en grande partie, à la puissance de son environnement de développement et aux larges possibilités d'interconnexions.

L'automatisation de n'importe quel processus a pour but de rendre le système plus rapide, fiable et peut agir devant n'importe quel type de contraintes ou de problème aléatoire.

L'automatisation consiste à rendre automatique les opérations qui exigeaient auparavant l'intervention humaine.

Dans notre travail demandé nous avons opté pour la gamme SIMATIC S7 de la firme allemande SIEMENS, dont on trouve l'automate programmable SIMATIC S7-300, vu ses performances d'optimalités dans la résolution des problèmes et les diverses possibilités de communication ainsi celles de flexibilité qu'il offre; et on a utilisé pour la programmation le Totally Integrated Automation Portal (TIA Portal v15) de Siemens qui représente parmi la nouvelle génération de logiciels d'automatisation se satisfaisant d'un seul environnement, un seul logiciel pour toutes les tâches d'automatisation.

2. L'Automatisme :

L'automatisme est la discipline traitant d'une part la caractérisation des systèmes automatisés et d'autre part le choix de la conception et de la réalisation de la partie commande. Il s'agit donc d'étudier les systèmes :

Réalisant leurs fonctions en relative autonomie.

Assurant un contrôle des performances par la mise en place possible d'une chaîne de retour.

L'automatisation de la production consiste à transférer tout ou une partie des tâches de coordination, auparavant exécutés humains.

3. Objectif de l'automatisation :

Chapitre II

Hors les objectifs à caractères financiers on trouve :

- Éliminer les tâches répétitives ;
- Simplifier le travail de l'humain ;
- Augmenter la sécurité ;
- Accroître la productivité ;
- Économiser les matières premières et l'énergie ;
- S'adapter à des contextes particuliers ;
- Maintenir la qualité.

4. Les systèmes automatisés :

Un système automatisé est toujours composé d'une partie commande (PC), et d'une partie opérative (PO), pour faire fonctionner ce système, l'opérateur (personne qui va faire fonctionner le système) va donner des consignes à la partie (PC), celle-ci va traduire ces consignes en ordres qui vont être exécutés par la (PO).

Une fois les ordres accomplis, la PO va le signaler à la PC (compte -rendu) qui va à son tour le signaler à l'opérateur, ce dernier pourra donc dire que le travail a bien été réalisé.

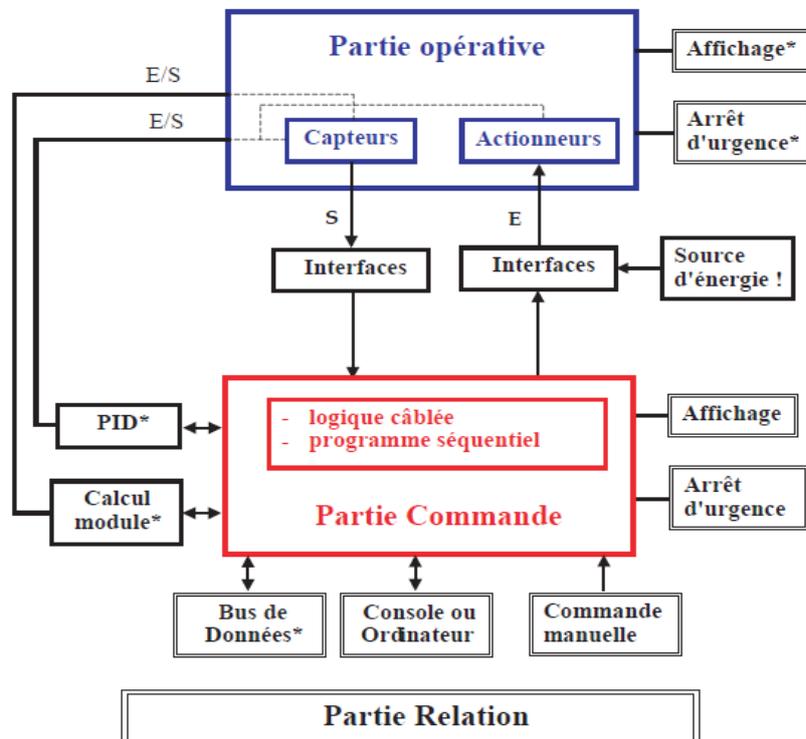


Figure II.1 : Structure d'un système automatisé.

4.1. La partie opérative (PO) :

Chapitre II

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

- Du programme qu'elle contient.
- Des informations reçues par les capteurs.
- Des consignes données par l'utilisateur ou par l'opérateur.

4.2. La partie commande (PC) :

Ce secteur de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la Partie Opérative, et les restitue vers cette même Partie en direction des pré-actionneurs et actionneurs. Elle comporte en générale un boîtier (appelé aussi bâti) contenant :

- Des actionneurs (transformant l'énergie reçu en énergie utile : moteur, vérin, lampe).
- Des capteurs (transforment la variation des grandeurs physiques liée au fonctionnement de l'automatisme en signaux électriques : capteur de position, de température, bouton poussoir).

4.3. La partie relation (PR) :

Sa complexité dépend de l'importance du système. Elle regroupe les différentes commandes nécessaires au bon fonctionnement du procédé, c'est à dire marche/arrêt, arrêt d'urgence, marche automatique, etc. L'outil de description s'appelle le Guide d'Études des Modes de Marches et d'Arrêts (GEMMA). Les outils graphiques, que sont le GRAFCET et le GEMMA, sont utilisés par les automaticiens et les techniciens de maintenance.

5. Historique Des automates programmables :

Les Automates Programmables Industriels (API) sont apparus aux Etats-Unis vers la fin des années soixante, à la demande de l'industrie automobile américaine (General Motors) qui réclamait plus d'adaptabilité de leurs systèmes de commande.

Les ingénieurs américains ont résolu le problème en créant un nouveau type de produit nommée automates programmables. Ils n'étaient rentables que pour des installations d'une certaine complexité, mais la situation a très vite changé, ce qui a rendu les systèmes câblés obsolètes. De nombreux modèles d'automates sont aujourd'hui disponibles ; depuis les nano automates bien adaptés aux machines et aux installations simples avec un petit

nombre d'entrées/sorties, jusqu'aux automates multifonctions capables de gérer plusieurs milliers d'entrées/sorties et destinés au pilotage de processus complexes.

6. Définition de l'Automate programmable :

De nos jours, les constructions de commande et les ingénieurs automaticiens n'ignorent plus rien des automates programmables ou (Programmable Logic Controller PLC). Pour de nombreux problèmes de commande, il convient donc de déterminer le mode de commande le mieux approprié et à cet égard, le choix se porte de plus en plus sur l'automate programmable.

Il s'agit d'ailleurs non seulement d'une question de prix, mais bien d'avantage de gain de temps, de souplesse accrue dans la manipulation, de haute fiabilité, de localisation et d'élimination rapide des erreurs simultanément. Le produit final c'est-à-dire la machine ou l'installation équipée d'un tel automate atteint un niveau technique plus élevé.

7. Mise en œuvre d'un automate :

La mise en œuvre de tout système automatisé implique une série de tâches qui constitue :

- Autant d'étapes successives naturellement indépendantes.
- L'étude préalable.
- L'étude proprement dite et préparation.
- Fabrication et essais.
- Mise en route et exploitation.

8. Programmation des automates :

Pour programmer l'automate, l'automaticien peut utiliser :

- Une console de programmation ayant pour avantage portabilité.
- Un pc avec lequel la programmation est plus conviviale, communiquant avec l'automate par le biais d'une liaison série rs232 ou d'un réseau de terrain « FIELDBUS ».

9. Critères de choix d'un automate :

Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe et les contacts commerciaux et expériences vécues sont déjà un point de départ.

Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir se retourner en cas de perte de vitesse de l'une d'entre elles.

Le personnel de maintenance doit toutefois être formé sur ces matériels et une trop grande diversité des matériels peut avoir de graves répercussions.

Chapitre II

Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions. La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables.

Il faut ensuite quantifier les besoins :

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

10. Traitement du programme automate :

Tous les automates fonctionnent selon le même mode opératoire :

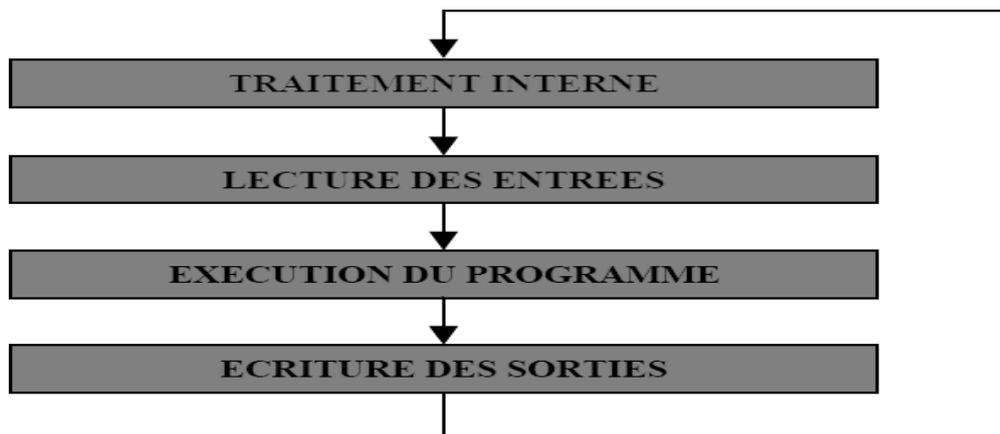


Figure II.2 : Fonctionnement cyclique d'un automate.

- **Traitement interne** : L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).
- **Lecture des entrées** : L'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.

- **Exécution du programme** : L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.
- **Écriture des sorties** : L'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

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

11. Langage de programmation pour API :

Les programmes utilisés avec les API peuvent être écrits dans différents formats.

Pour que les ingénieurs ayant peu de connaissances en programmation puissent élaborer des programmes pour les API, le langage à contacts a été conçu. La plupart des fabricants d'automates ont adopté cette méthode d'écriture des programmes.

Toutefois, puisque chacun a eu tendance à développer ses propres versions, une norme internationale a été établie pour le langage à contacts et, par voie de conséquence, pour toutes les méthodes de programmation employées avec les API. La norme IEC 1131-3 définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces cinq langages sont :

11.1. Le langage LD (Ladder Diagram) :

C'est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation.

11.2. Le langage IL (Instruction List) :

Un langage textuel de bas niveau. Il est particulièrement adapté aux applications de petite taille. Les instructions opèrent toujours sur un résultat courant (ou registre IL). L'opérateur indique le type d'opération à effectuer entre le résultat courant et l'opérande. Le résultat de l'opération est stocké à son tour dans le résultat courant.

Un programme IL est une liste d'instructions. Chaque instruction doit commencer par une nouvelle ligne, et doit contenir un opérateur, complété éventuellement par des modificateurs et, si c'est nécessaire pour l'opération, un ou plusieurs opérandes, séparés par des virgules (','). Une étiquette suivie de deux points (':') peut précéder l'instruction. Si un commentaire est attaché à l'instruction, il doit être le dernier élément de la ligne.

Chapitre II

Des lignes vides peuvent être insérées entre des instructions. Un commentaire peut être posé sur une ligne sans instruction . [17]

11.3. Le langage FBD (Function Block Diagram) :

C'est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standards, de fonctions ou de blocs fonctionnels.

Les principales fonctions sont :

- L'énoncé RETURN (peut apparaître comme une sortie du diagramme, si liaison connectée prend l'état booléen TRUE, la fin du diagramme n'est pas interprétée.
- Les étiquettes et les sauts conditionnels sont utilisés pour contrôler l'exécution du diagramme. Aucune connexion ne peut être réalisée à droite d'un symbole d'étiquette ou de saut.

Saut à une étiquette (le nom de l'étiquette est « LAB »).

11.4. Le langage ST (Structured Text) :

Un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques. C'est le langage par défaut pour la programmation des actions dans les étapes et des conditions associées aux transitions du langage SFC.

Un programme ST est une suite d'énoncés. Chaque énoncé est terminé par un point-virgule (« ; »). Les noms utilisés dans le code source (identificateurs de variables, constantes, mots clés du langage...) sont délimités par des séparateurs passifs ou des séparateurs actifs, qui ont un rôle d'opérateur. Des commentaires peuvent être librement insérés dans la programmation.

11.5. Le langage SFC (Sequential Function Chart), ou GRAFCET :

C'est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD.

Les principales règles graphiques sont :

- un programme SFC doit contenir au moins une étape initiale.
- une étape ne peut pas être suivie d'une autre étape.
- une transition ne peut pas être suivie d'une autre transition.

L'automate utilisé dans notre projet appartient à la gamme SIMATIC S7 de SIEMENS ;

Chapitre II

Le S7-300 « CPU 315-2 PN/DP », qui est un automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI), profibus et industriel Ethernet.

12. Les avantages et les inconvénients de l'automatisation :

12.1. Les avantages :

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

12.2. Les inconvénients :

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

13. Domaines d'application des systèmes automatisés :

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

- Automobile
- Aviation
- Industrie
- Médical
- Transport

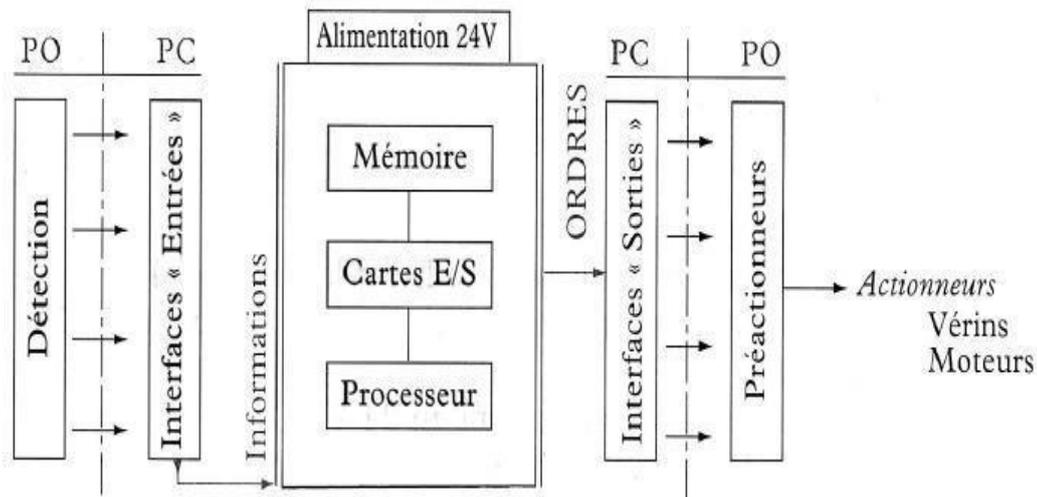
14. Caractéristiques d'un automatisme :

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

Un automatisme bien conçu :

- Simplifie considérablement le travail de l'homme qui, libéré vis-à-vis de la machine, peut se consacrer à des activités plus noble .
- Réduit les tâches complexes, pénibles ou indésirables en les faisant exécuter par la machine .
- Facilite les changements de fabrication en permettant de passer d'une quantité ou d'un type de production à un autre .

- Améliore la qualité des produits en asservissant la machine à des critères de fabrication et à des tolérances qui seront respectées dans le temps .
- Accroît la production ainsi que la productivité .
- Permet de réaliser des économies de matière et d'énergie .



- Augmente la sécurité du personnel .
- Contrôle et protège les installations et les machines.

15. Structure interne des automates programmables :

La structure matérielle interne d'un API obéit au schéma ci-dessous

Détaillons successivement chacun des composants qui apparaissent sur ce schéma.

15.1. Le processeur :

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

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

Figure II.3 : Structure interne d'un API.

vérifications de données.

Chapitre II

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

Les principaux registres existants dans un processeur sont :

- **L'accumulateur :**

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

- **Le registre d'instruction :**

Il reçoit l'instruction à exécuter et décode le code opération. Cette instruction est désignée par le pointeur.

- **Le registre d'adresse :**

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

- **Le registre d'état :**

C'est un ensemble de positions binaires décrivant, à chaque instant, la situation dans laquelle se trouve précisément la machine.

- **Les piles :**

Une organisation spéciale de registres constitue une pile, ces mémoires sont utilisées pour contenir le résultat de chaque instruction après son exécution. Ce résultat sera utilisé ensuite par d'autres instructions, et cela pour faire place à la nouvelle information dans l'accumulateur. [18]

15.2. Les mémoires :

Un système de processeur est accompagné par un ou plusieurs types de mémoires. Elles permettent :

- De stocker le système d'exploitation dans des ROM ou PROM,
- Le programme dans des EEPROM,
- Les données système lors du fonctionnement dans des RAM. Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type PCMCIA.

15.3. Les modules entrées/sorties :

Ils assurent le rôle d'interface entre la CPU et le processus, en récupérant les informations sur l'état de ce dernier et en coordonnant les actions.

Chapitre II

Plusieurs types de modules sont disponibles sur le marché selon l'utilisation souhaitée :

- **Modules TOR (Tout Ou Rien):** l'information traitée ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type d'information délivrée par une cellule photoélectrique, un bouton poussoir ...etc.
- **Modules analogiques :** l'information traitée est continue et prend une valeur qui évolue dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (débitmètre, capteur de niveau, thermomètre...etc.).
- **Modules spécialisés :** l'information traitée est contenue dans des mots codes sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

15.4. L'alimentation :

Elle a pour rôle de fournir les tensions continues nécessaires aux composants avec de bonnes performances, notamment face aux microcoupures du réseau électrique qui constitue la source d'énergie principale. La tension d'alimentation peut être de 5V, 12V ou 24V.

D'autres alimentations peuvent être nécessaires pour les châssis d'extension et pour les modules entrées/sorties. Un onduleur est nécessaire pour éviter les risques de coupures non tolérées.

15.5. Liaisons de communication :

Elles Permettent la communication de l'ensemble des blocs de l'automate et des éventuelles extensions. Les liaisons s'effectuent :

- Avec l'extérieur par des borniers sur lesquels arrivent des câbles transportant les signaux électriques.
- Avec l'intérieur par des bus reliant divers éléments, afin d'échanger des données, des états et des adresses.

16. Conclusion :

Ce chapitre met en avant l'importance des Automates Programmables Industriels (API) dans l'automatisation des processus industriels. Il retrace leur évolution historique, décrit leur fonctionnement à travers les systèmes automatisés, et explore les différents types d'automates ainsi que les langages de programmation utilisés (LD, FBD, ST). Les blocs de programme (OB, FC, FB, DB) sont également abordés pour expliquer la structuration des programmes. En somme, ce chapitre souligne que les API améliorent la productivité, la flexibilité et la sécurité, malgré des coûts initiaux élevés et des besoins de maintenance.

III. Chapitre III : Logiciel TIA Portal

1. Introduction :

Le TIA portal (Totally Integrated Automation portal), en français Portail d'automatisation totalement intégré, est une plateforme logicielle globale, comme son nom l'indique elle intègre Tous les progiciels requis, (step 7, winCC, startdrive) de la configuration matérielle à la visualisation du processus en passant par la programmation, dans un cadre complet d'ingénierie. Le TIA Portal permet pour la première fois de disposer, au sein d'un cadre, d'un environnement de travail commun pour une ingénierie transparente avec différents systèmes SIMATIC. Il est destiné à la programmation des automates s7 1200, s7 1500 et les CPU récentes de s7 300 et s7 400.

Avec TIA Professional V13, les fonctions suivantes peuvent être utilisées pour automatiser une installation :

- Configuration et paramétrage du matériel.
- Programmation et paramétrage de la communication.
- Test, mise en service et dépannage avec les fonctions d'exploitation et de diagnostic.
- Documentation.
- Génération d'écrans de visualisation pour les Basic Panels SIMATIC avec WinCC Basic intégré. Il est également possible de générer des écrans de visualisation pour les PC et autres Panels à l'aide d'autres progiciels WinCC.

2. Vue du portail et vue de projet :

Lorsqu'on lance TIA Portal, l'environnement de travail se décompose en deux types de vue :

- La vue du portail : elle est axée sur les tâches à exécuter et sa prise en main est très rapide.
- La vue du projet : elle comporte une arborescence avec les différents éléments du projet. Les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue.

2.1. Vue du portail :

La vue du portail offre un aperçu de toutes les étapes de configuration du projet et un accès orienté tâche de votre tâche d'automatisation.

Les différents portails ("Démarrage", "Appareils et réseaux", "Programmation API", "Visualisation", " En ligne et diagnostic", etc.) montrent de manière claire et ordonnée

Chapitre III

l'ensemble des étapes de travail nécessaires à l'exécution d'une tâche d'automatisation. La figure suivante montre la structure de la vue de portail :

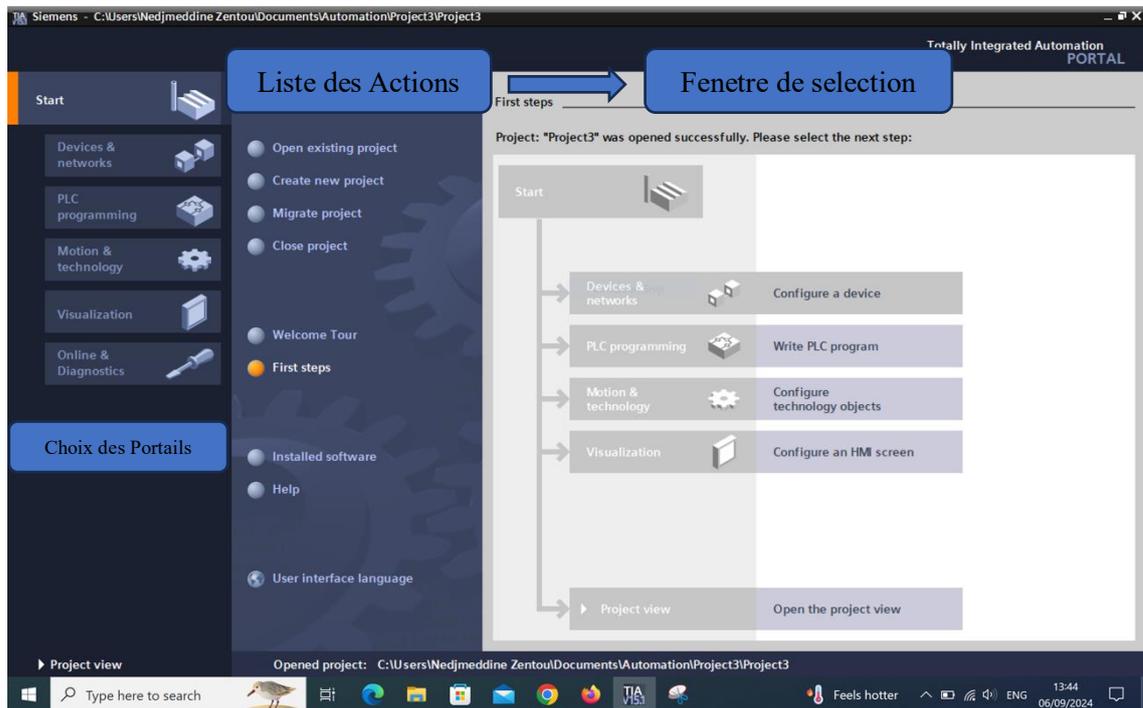


Figure III.1 : Vue de Portail.

2.2. Vue du projet :

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

- **La fenêtre de travail :** permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des HMI,...
- **La fenêtre d'inspection :** permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme,...).
- **Les onglets de sélection de tâches :** ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle, bibliothèques des composants, bloc de programme, instructions de programmation).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas. Il est également possible de redimensionner, réorganiser, désancrer les différentes fenêtres.

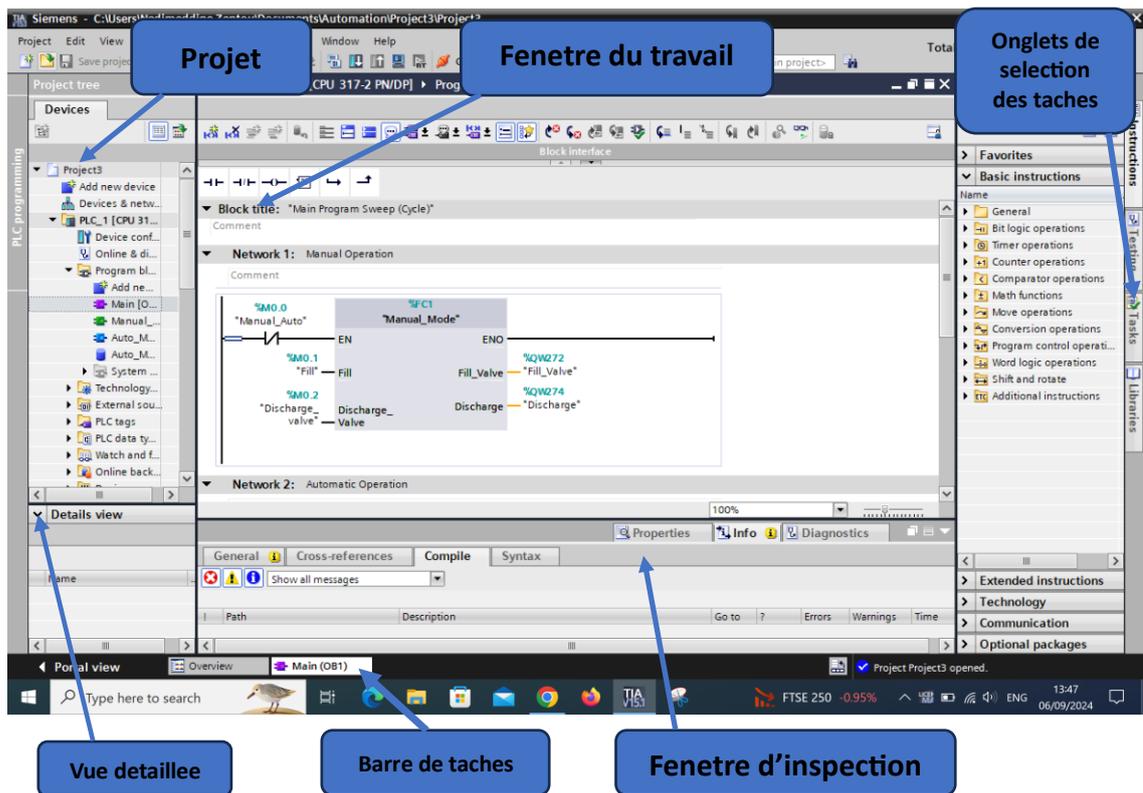


Figure III.2 : Vue de Projet.

3. Création d'un projet :

La création d'un projet commence toujours par la configuration du matériel. Cette dernière revient à lister tous les modules présents dans le projet. Par exemple l'alimentation, la CPU, les entrées-sorties, les modules de communications, etc... tous ces éléments se trouvent dans la bibliothèque du projet.

Pour créer un projet dans la vue du portail, il faut sélectionner l'action « Créer un projet ». On peut donner un nom au projet, choisir un chemin où il sera enregistré, indiquer un commentaire ou encore définir l'auteur du projet. Une fois que ces informations sont entrées, il suffit de cliquer sur le bouton « créer ».

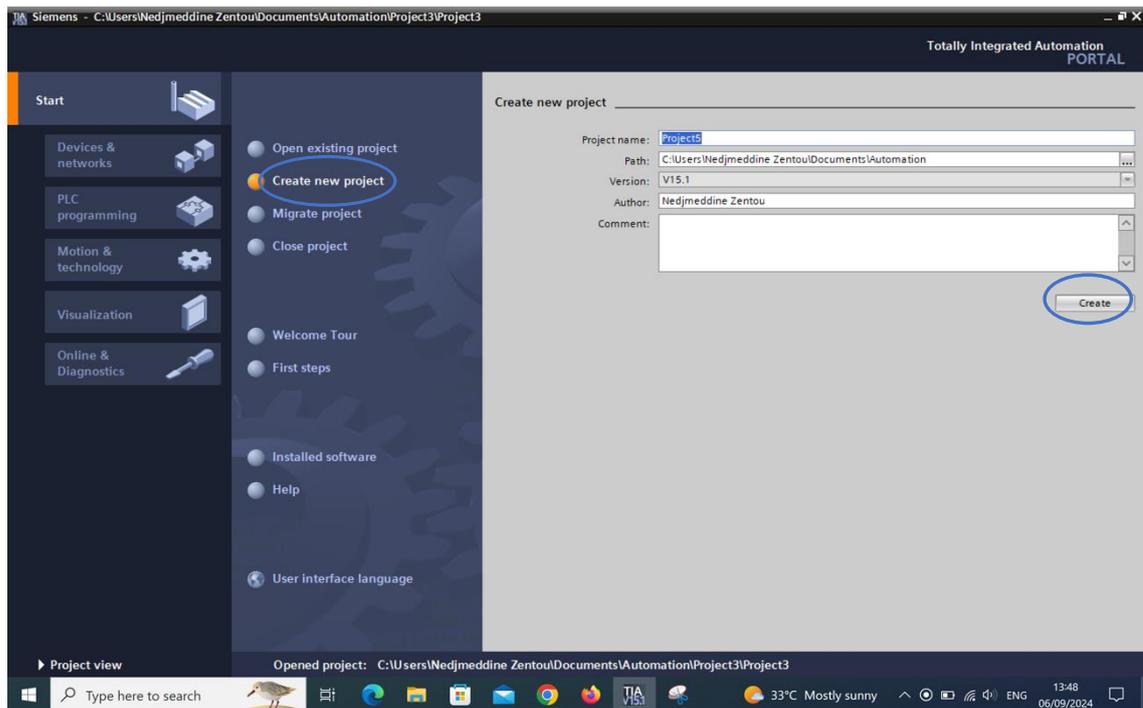


Figure III.3 : Vue création de projet.

3.1. identification du matériel :

Le moyen le plus sûr d'identifier le matériel est d'utiliser le nom et le numéro de référence parfois un numéro de firmware peut-être présent. L'identification de chaque module se fait toujours à l'aide des critères cités comme le montre les exemples suivants :

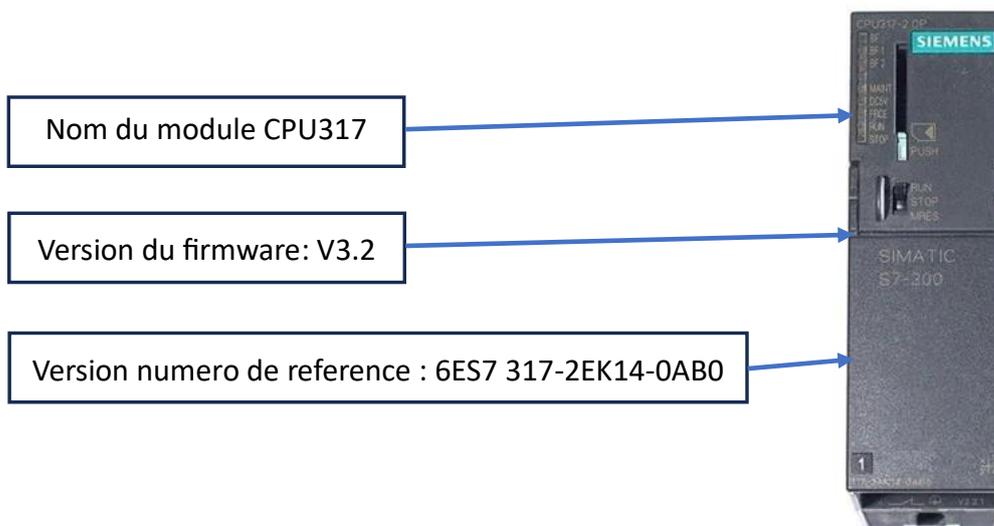


Figure III.4 : Identification d'un module.

3.2. Configuration et paramétrage du matériel :

Les modules complémentaires de l'API (alimentation, E/S TOR ou analogiques, module de communication AS-i,...) peuvent être ajoutés en utilisant le catalogue. Si l'on veut ajouter un écran ou un autre API, il faut repasser par la commande « **ajouter un appareil** » dans le navigateur du projet.

Une fois le projet crée, on peut configurer la station de travail. La première étape consiste à définir le matériel existant. Pour cela, on peut passer par la **vue du projet** et cliquer sur « **ajouter un appareil** » dans le navigateur du projet.

Dans cette partie on sélectionne « contrôleurs ». Lorsque l'élément à insérer est sélectionné dans le projet, une description est proposée dans l'onglet information.

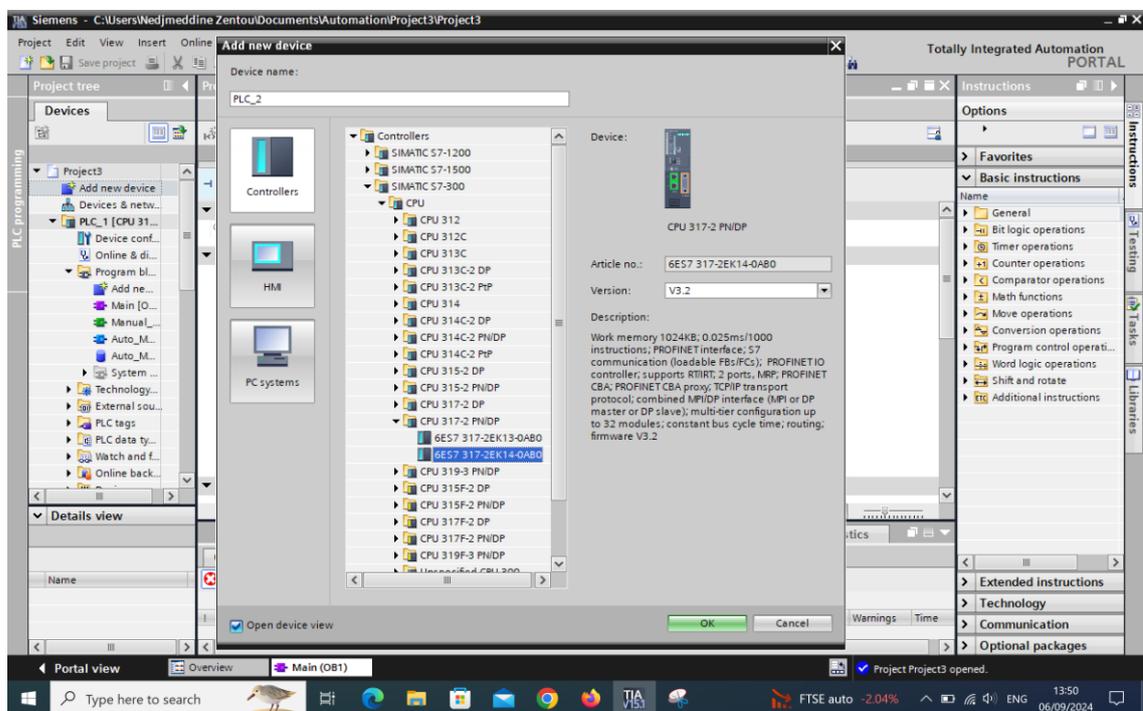


Figure III.5 : Onglet d'information.

3.3. Les variables API :

3.3.1. Adresses symbolique et absolue :

Dans TIA Portal, toutes les variables globales (entrées, sorties, mémentos,..) possèdent une adresse symbolique et une adresse absolue.

- L'adresse absolue représente l'identificateur de l'opérande (I, Q, M) et son adresse et numéro de bit et elle revient à utiliser l'adresse physique de l'automate.

Chapitre III

- L'adresse symbolique correspond au nom que l'utilisateur a donné à la variable (ex : Bouton Marche).

On utilise généralement l'adressage symbolique pour donner de la clarté au programme. Cela revient à substituer chaque adresse physique par une adresse symbolique qui représente au mieux la fonction. Le lien entre les adresses symbolique et absolue se fait dans la table des variables API. Lors de la programmation, on peut choisir d'afficher les adresses absolues, symboliques ou encore les deux simultanément.

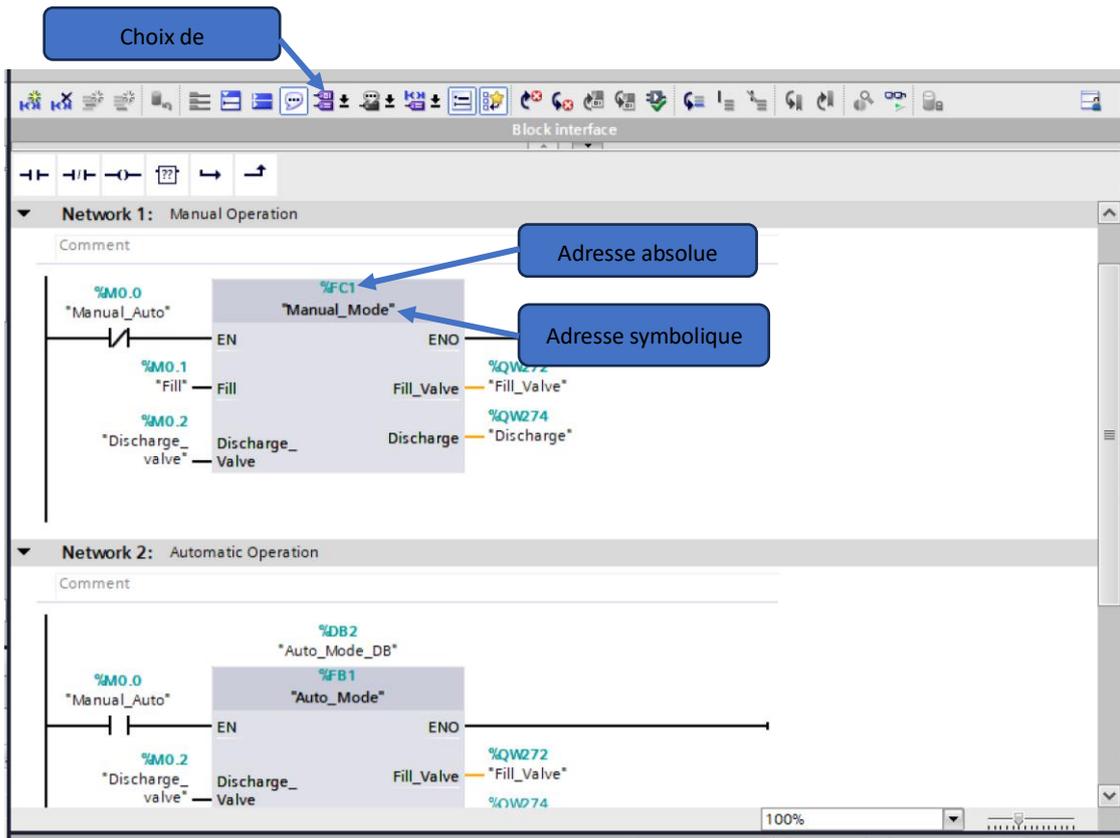


Figure III.6 : Exemple d'adresse symbolique et physique.

3.3.2. Adressage des E/S :

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matériel, on sélectionne « **appareil et réseau** » dans le navigateur du projet.

Dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet « **Vue des appareils** » et de sélectionner l'appareil voulu. On sélectionne la CPU puis à l'aide des deux petites flèches (voir figure), on fait apparaître l'onglet « **Vue d'ensemble des appareils** ». Les adresses des entrées et sorties apparaissent. On peut les modifier en entrant une nouvelle valeur dans la case correspondante.

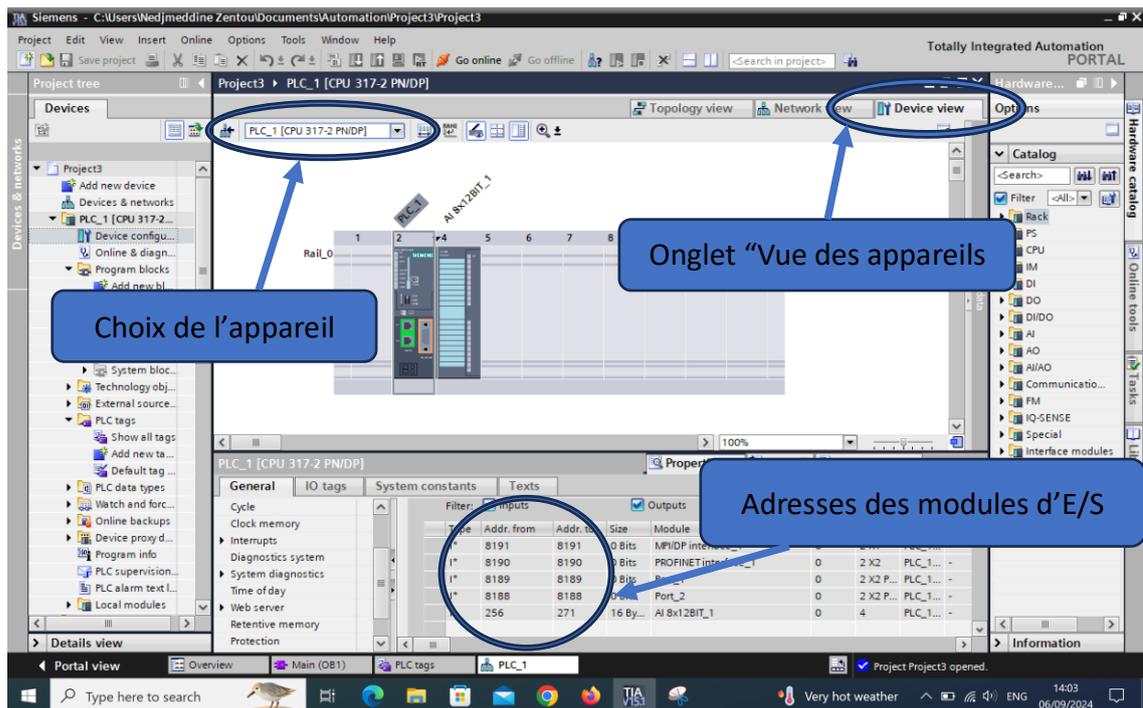


Figure III.7 : Vue des appareils.

3.4. Table des variables API :

Pour TIA Portal, la table de variables est l'outil qui permet de créer un adressage symbolique. C'est dans cette table qu'on va pouvoir déclarer toutes les variables et constantes utilisées dans le programme. Lorsqu'on définit une variable API, il faut définir :

- Un nom : c'est l'adressage symbolique de la variable.
- Le type de donnée : BOOL, INT, REAL... etc.
- L'adresse absolue : par exemple %Q1.5.

On peut également insérer un commentaire qui nous renseigne, sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable.

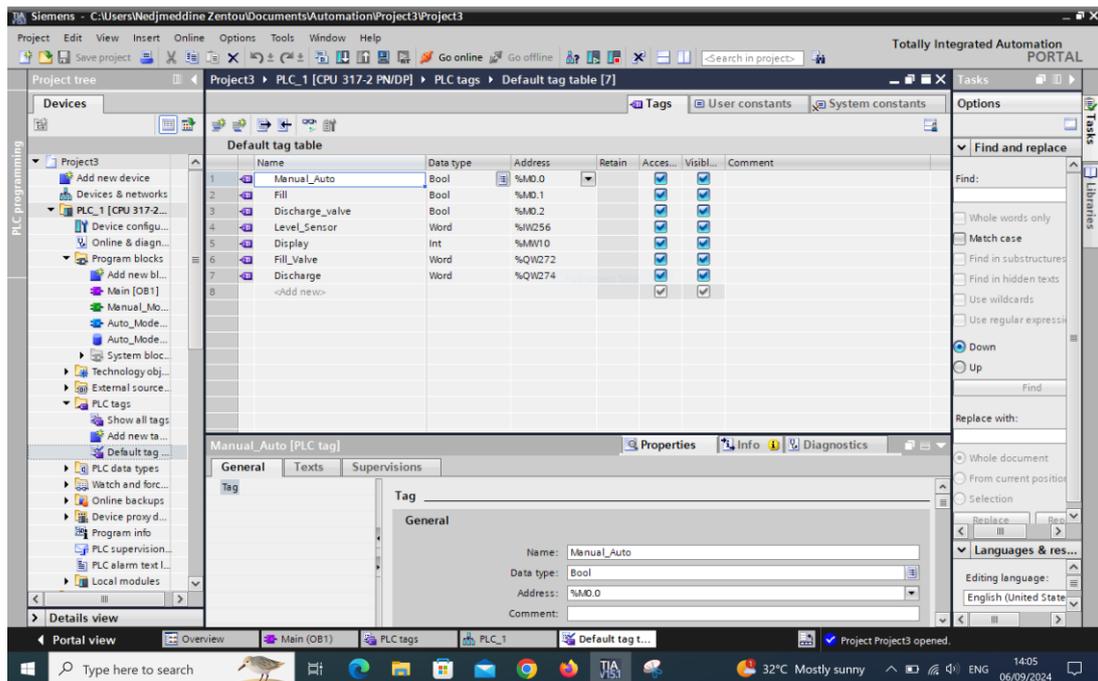


Figure III.8 : Table de Variable.

3.5. Tableau d'affectation :

Le tableau d'affectation est très utile pour connaître les entrées, sorties et mémoires déjà utilisées. Pour y accéder on sélectionne d'abord « Sauvegarde en ligne », puis « Informations sur le programme » ensuite l'onglet « Tableau d'affectation ».

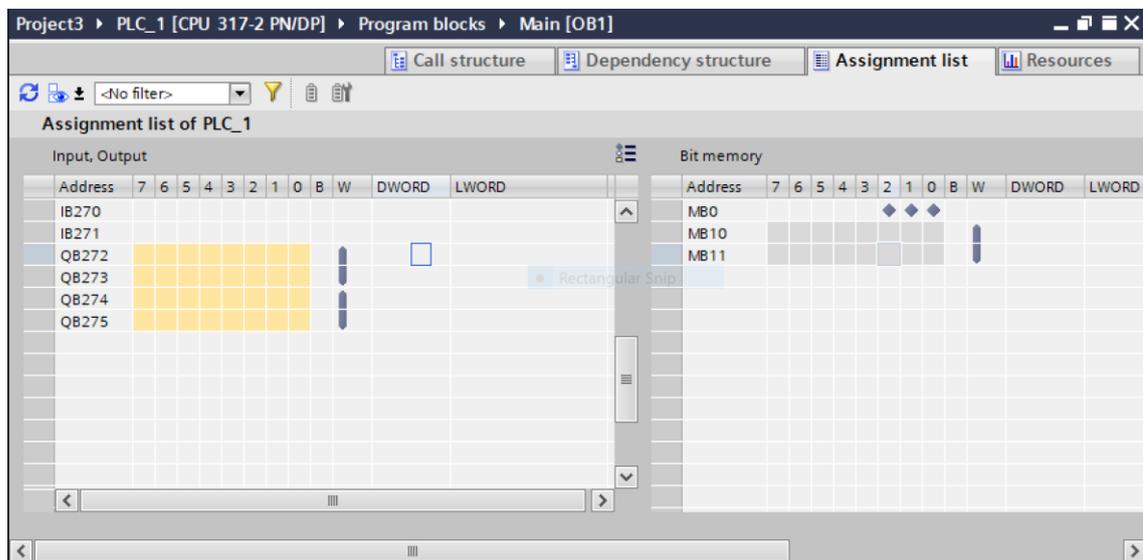


Figure III.9 : Table d'affectation.

Chapitre III

Dans ce tableau d'affectation on voit que les mémoires M0.0- M0.1- M0.2 sont utilisés. Ainsi que le Word MB10 ,et L'entrée IW256.

3.6. Les références croisées :

Les références croisées sont des liens qui permettent d'afficher les différents contacts et blocs de fonction lié à une entrée ou à une sortie. Ces liens permettent de retrouver facilement sans avoir à fouiller tout le programme un élément pour faire une modification ou un contrôle lors d'un dépannage par exemple.

4. Les blocs dans TIA Portal :

L'automate met à disposition différents types de blocs qui contiennent le programme et les données correspondantes. Selon les exigences et la complexité du processus, il est possible de structurer le programme en différents blocs : OB, FB et FC [19].

4.1. Les blocs d'organisation – OB :

Les OB sont appelés par le système d'exploitation en liaison avec les événements suivants:

- Exécution cyclique du programme et traitement des erreurs.
- Exécution du programme déclenchée par des alarmes (processus, diagnostic...).

Pour que le traitement du programme le projet doit posséder un OB cyclique par exemple OB1

4.2. Les fonctions – FC :

Ce sont des blocs de code sans mémoire, les données des variables temporaires sont perdues après l'exécution de la fonction, si on veut mémoriser ces données, il faut utiliser des opérandes globaux, Elles sont utilisées pour la programmation de fonction utilisées plusieurs fois.

4.3. Les blocs fonctionnels – FB :

Ce 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 de blocs.

4.4. Les blocs de données – DB :

Les blocs de donnée (DB) sont des zones données du programme qui contiennent des données utilisateur. On distingue deux types de bloc de donnée :

DB globale : Un DB global peut être créé indépendamment des autres blocs. Tous les blocs (FB, FC et OB) peuvent lire ou écrire les données contenues dans un bloc de données global.

Chapitre III

DB d'instance : Le DB d'instance est associé à un certain bloc fonctionnel et contient les données locales de ce FB associé. Il ne peut être créé que si le FB associé est déjà présent dans le programme.

4.5. Création et programmation d'un bloc :

Quatre genres de blocs peuvent être insérés : OB- FB- FC- DB. A part lors d'une programmation en GRAFCET ou d'une programmation d'un niveau avancé tous les blocs insérés seront des FC.

Sélectionner Bloc de programme double clic sur ajouter nouveau bloc. On sélectionne par exemple FC, on choisit le langage CONT (ou SCL) et on clique sur OK :

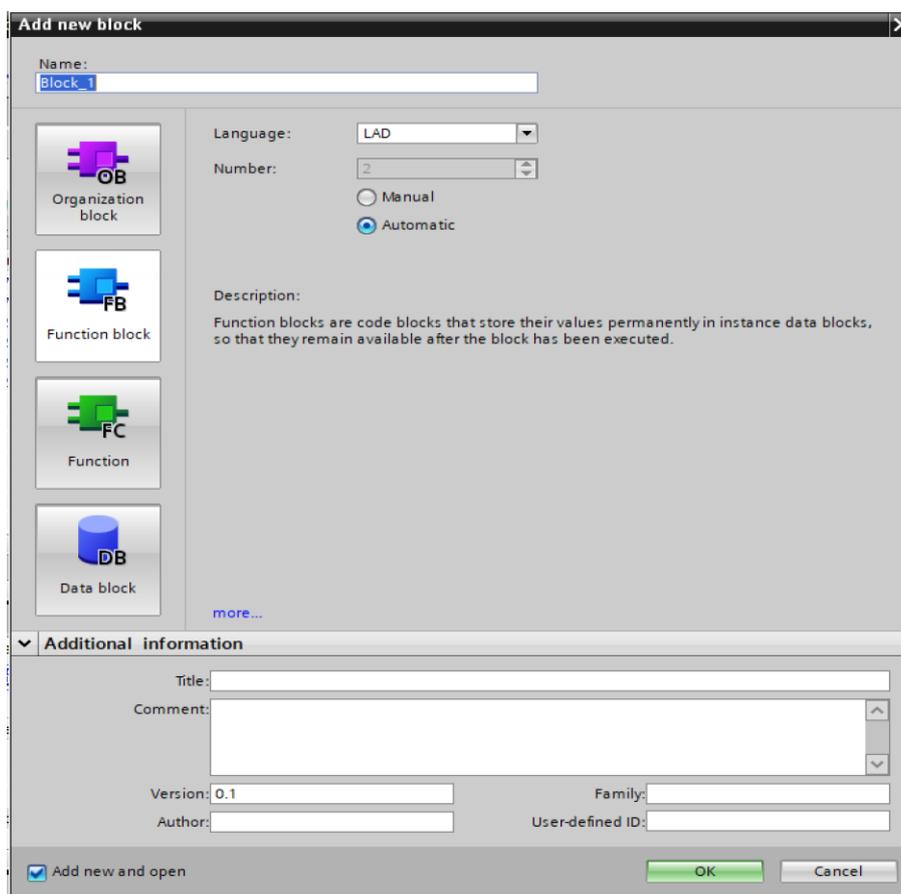


Figure III.10 : Vue des blocs disponible.

La fenêtre du Bloc_1 est ouverte et prête pour la programmation, pour tester le fonctionnement, saisissons : Entrée : %I0.0, Sortie : %Q0.0.

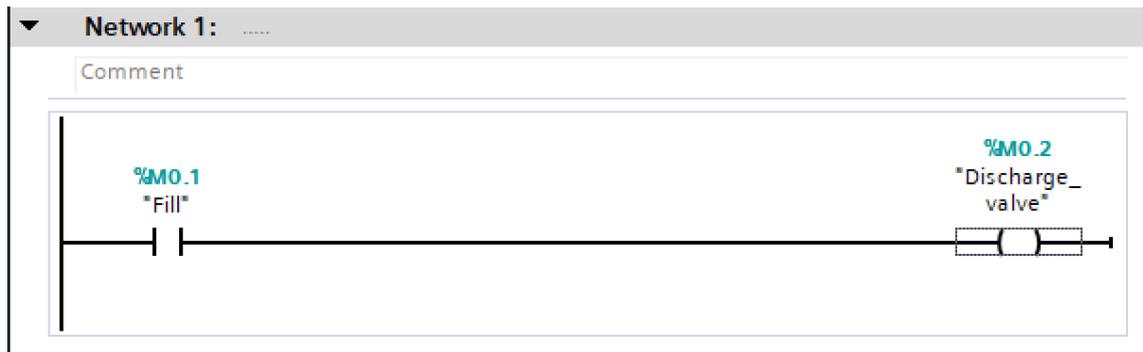


Figure III.11 : Programme d'un bloc.

4.6. Appel d'un bloc l'OB1 :

Les blocs de type FC et FB seront appelés dans l'OB1. Après l'ouverture de ce dernier on sélectionne le « Bloc_1 », on le glisse et le déposer sur le réseau 1 de OB1. On procède de la même manière avec les blocs de fonction FB.



Figure III.12 : Appel d'un bloc dans OB1.

5. Les langages de programmation dans le TIA Portal :

Le SIMATIC STEP 7 V15 propose des éditeurs de programmation performants pour la programmation des automates SIMATIC S7 :

- Le langage structuré (SCL), le schéma à contacts (CONT) et le logigramme (LOG) sont disponibles pour tous les automates.
- La liste d'instructions (LIST) et la programmation en graphes séquentiels (GRAPH, SFC) existent en outre pour les gammes d'automates S7-1500, S7-300, S7-400 et WinCC [19].

6. Compilation et chargement :

Chapitre III

La compilation est nécessaire au premier chargement et lors d'une modification de la configuration du matériel par exemple l'ajout ou suppression d'un module ou la

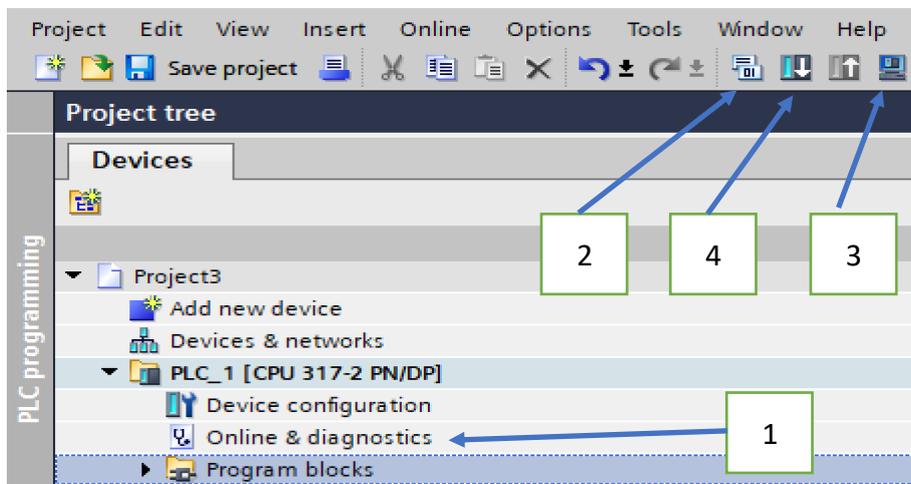


Figure III.13 : Les étapes de compilation.

modification des paramètres d'un module (mémento de cadence, rémanence, temps de cycle, adresse de byte, etc...).

1. Sélectionner le dossier.
2. PLC_1.Compile le projet.
3. Démarrer la simulation.
4. Charger dans l'automate.

7. La simulation avec PLCSIM :

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés dans le programme par exemple : activer ou de désactiver des entrées.

En outre, S7-PLCSIM offre la possibilité de changer l'état de fonctionnement de la CPU (STOP, RUN et RUNP) comme pour une CPU réelle. De plus, il dispose d'une fonction de pause qui permet d'interrompre momentanément la CPU, sans affecter l'état du programme.

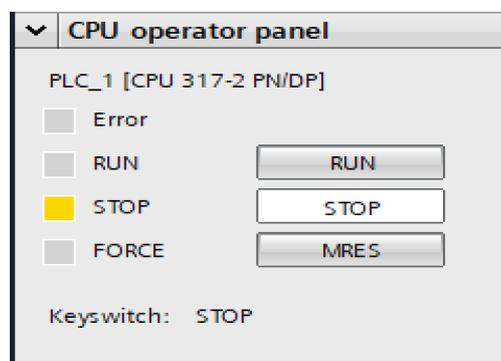


Figure III.14 : Etat de la CPU.

Chapitre III

On peut créer des « fenêtres » dans lesquelles on a la possibilité d'accéder aux zones mémoire d'entrée et de sortie, aux accumulateurs ainsi qu'aux registres de la CPU de simulation. On peut également accéder à la mémoire par adressage symbolique.

8. Supervision du processus :

Lors de la conception d'un système automatisé, on a souvent besoin de concevoir une partie contrôle commande ou supervision, permettant de surveiller et de contrôler le système. Dans la nouvelle plateforme TIA le logiciel de supervision et de génération d'écran de visualisation Win CC est intégré ce qui facilite la tâche de conception des vues.

8.1. Présentation du WinCC sous TIA Portal :

Le SIMATIC WinCC dans le TIA Portal est le logiciel pour la programmation et la configuration de solutions de commande, de visualisation et d'entraînement des applications HMI allant de solutions de commande simples avec des Basic Panels aux applications SCADA pour systèmes multipostes basés sur PC. SIMATIC WinCC se décline en plusieurs versions échelonnées en prix et performances. Ces versions s'appuient l'une sur l'autre et sont adaptées de façon optimale aux différentes classes de pupitres opérateurs. Le package de niveau supérieur englobe toujours les capacités de configuration du package de niveau inférieur [20].

8.2. Définition de l'interface homme-machine (HMI) :

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

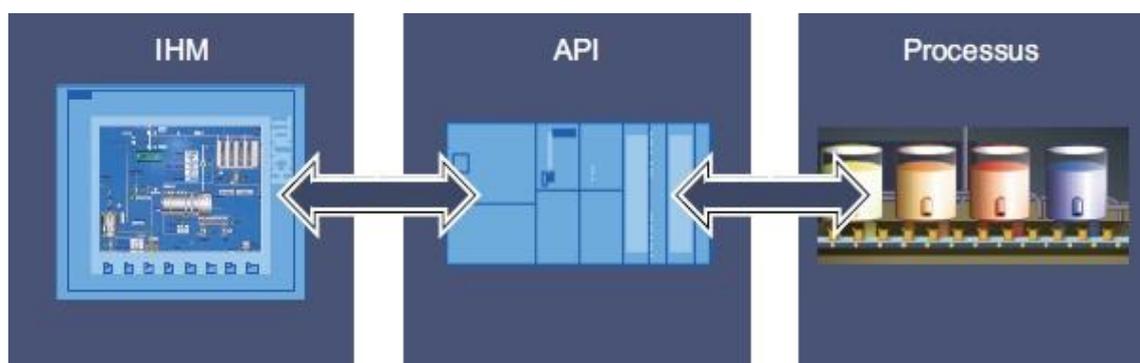


Figure III.15 : Schéma explicative.

Les possibilités suivantes vous sont en autres offertes pour le contrôle-commande des machines et installations :

8.2.1. Représentation du processus :

Le processus est représenté sur le pupitre opérateur. Lorsqu'un état du processus évolue.

8.2.2. Commande du processus :

L'opérateur peut commander le processus via l'interface utilisateur graphique.

8.2.3. Vue des alarmes :

Lorsque surviennent des états critiques dans le processus, une alarme est immédiatement Déclenchée.

8.2.4. Archivage de valeurs processus et d'alarmes :

Les alarmes et valeurs processus peuvent être archivées par le système IHM. On peut ainsi documenter la marche du processus et accéder ultérieurement aux données de la production écoulée.

8.2.5. Documentation de valeurs processus et d'alarmes :

Les alarmes et valeurs processus peuvent être éditées par le système IHM sous forme de journal.

8.2.6. Gestion des paramètres de processus et de machine :

Les paramètres du processus et des machines peuvent être enregistrés au sein du système IHM dans des recettes [20].

8.3. Configuration du HMI dans le TIA portal :

Comme précédemment une fois le projet crée, on peut configurer la station de travail. La première étape consiste à définir le matériel existant. Pour cela, on peut passer par la **vue du projet** et cliquer sur « **ajouter un appareil** » dans le navigateur du projet. Et on sélectionne HMI. On bien directement à partir de **la vue portail** en cliquant sur « Configurer un appareil », puis sur « **Ajouter un appareil** » et on sélectionne HMI. On peut nommer l'HMI dans le champ « **nom d'appareil** ».

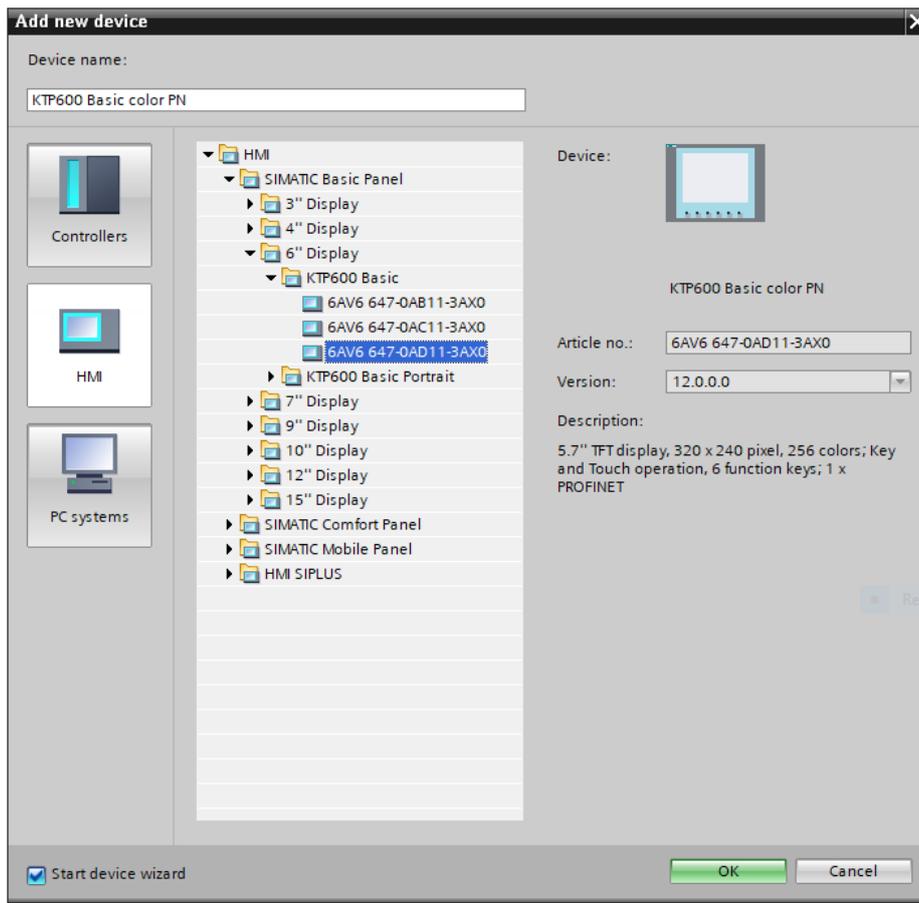


Figure III.16 : Vue configuration HMI.

On choisit par exemple le KTP600 BASIC une description des caractéristique de matériel s'affiche, on clique sur « ajouter » pour l'ajout de l'appareil. Après le choix de ce dernier une fenêtre d'assistance s'affiche qui contient les paramètres généraux de la configuration de l'HMI choisi par exemple la liaison avec l'API comme la figure suivante le montre :

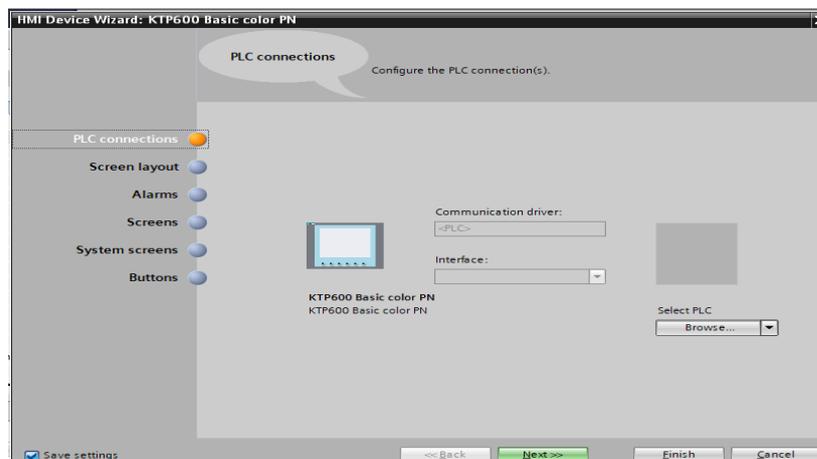


Figure III.17 : Assistant HMI.

8.4. Eléments de l'interface utilisateur de WinCC :

L'environnement de travail de WinCC se compose de plusieurs éléments. Certains de ces éléments sont liés à des éditeurs particuliers et uniquement visibles lorsque cet éditeur est activé.

- **La fenêtre de travail** : permet de visualiser l'HMI sélectionnée dans le projet.
- **Les onglets de sélection de tâches** : ont un contenu qui varie en fonction de l'objet sélectionné (accessoire, animation, apparence, instruction du script, tâches bibliothèques).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas.

- **La fenêtre des propriétés** : Elle permet de modifier les propriétés d'un objet sélectionné dans la zone de travail, le contenu de cette fenêtre dépend de l'objet sélectionné.
- **Fenêtre de projet** : La fenêtre du projet est le poste central de traitement. Tous les éléments et tous les éditeurs disponibles d'un projet sont affichés sous forme d'arborescence et peuvent être ouverts à partir de cette fenêtre. A chaque éditeur correspond une icône qui permet d'identifier les objets qui lui sont associés. Seuls les éléments pris en charge par le pupitre opérateur sélectionné apparaissent dans cette fenêtre. On peut aussi accéder aux paramètres du pupitre, à la localisation et à la gestion de versions.

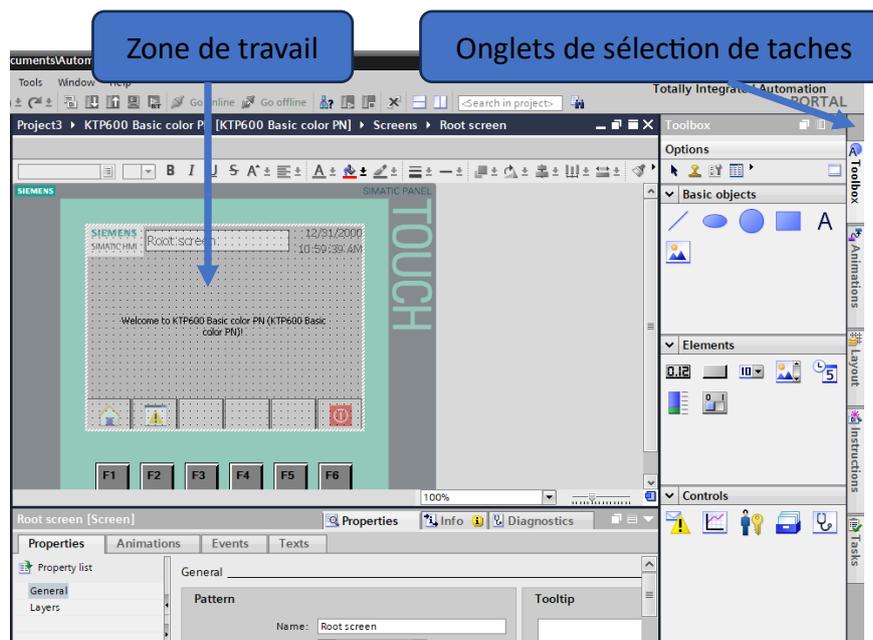


Figure III.18 : Vue d'interface WinCC.

8.5. Définition des variables :

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

8.5.1. Les variables externes :

Permettent d'échanger des données entre les composants d'un processus automatisé, et un pupitre operateur. Une variable externe est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture à cette cellule mémoire est possible aussi bien à partir du pupitre operateur que de l'automate.

8.5.2. Les variables internes :

Ne possèdent aucun lien avec l'automate. Elles sont enregistrées dans la mémoire du pupitre. Le pupitre operateur en question peut donc accéder en lecture et en écriture aux variables internes. Elles sont créées par exemple pour exécuter des calculs locaux à WinCC. La déclaration des variables s'effectue dans l'éditeur "Variables" .A leur déclaration, une configuration de base leur est attribuée. L'éditeur "Variables" nous permet d'adapter cette configuration aux besoins de notre projet, l'éditeur s'ouvre systématiquement par ajout de variable.

8.6. Les alarmes dans WinCC :

Les alarmes montrent les événements où les anomalies qui peuvent survenir dans le processus, elles peuvent servir par exemple à diagnostiquer les erreurs.

ID	Name	Alarm text	Alarm class	Trigger tag	Limit
1	Alarme Analogique	niveau Haut Atteint	Alarme_re...	Display	90
2	Analog_alarm_1	niveau Bas Atteint	Alarme_reser...	Display	10

Figure III.19 : Vue des alarmes HMI.

Chapitre III

8.6.1. Classes d'alarmes :

_ "Erreurs" pour les alarmes TOR et analogiques signalant des états critiques ou dangereux du fonctionnement et du processus. Les alarmes de cette classe doivent toujours être acquittées.

_ "Avertissements" pour les alarmes TOR et analogiques signalant des états normaux du fonctionnement et du processus ainsi que des déroulements du processus. Les alarmes de cette classe n'ont pas besoin d'être acquittées.

_ "Système" La classe d'alarmes "System" englobe les alarmes qui indiquent les états du pupitre opérateur et des automates. Les messages de la classe d'alarmes "System" appartiennent aux messages système.

_ " Diagnostic" La classe d'alarmes "Diagnostic Events" englobe les alarmes qui indiquent les états et les événements des automates SIMATIC S7. L'utilisateur n'acquiesce pas les alarmes appartenant à cette classe [21].

Nom d'affichage	Nom	Modèle d'acquiescement	Archive	Adresse e-mail	Couleur...	Couleur...
!	Errors	Alarme à acquiescement simple	<aucune archiv...	
\$	Warnings	Alarme sans acquiescement	<aucune archiv...		255...	255...
S7	System	Alarme sans acquiescement	<aucune archiv...		255...	255...
A	Diagnosis events	Alarme sans acquiescement	<aucune archiv...		255...	255...
NA	Acknowledgement	Alarme à acquiescement simple	<aucune archiv...		255...	255...
	No Acknowledgement	Alarme sans acquiescement	<aucune archiv...		255...	255...

Figure III.20 : Vue classe d'alarme.

8.6.2. Types des alarmes :

WinCC prend en charge les types d'alarme suivants :

- Les alarmes définies par le système servent à surveiller le pupitre opérateur ou l'automate.
- Alarmes définies par l'utilisateur qui servent à surveiller l'installation.

Alarmes analogiques : Les alarmes analogiques indiquent les dépassements de valeurs limites dans le process en cours.

Alarmes de bit : Les alarmes de bit indiquent un état dans le process en cour.

Chapitre III

Alarmes utilisateur : Les alarmes utilisateur indiquent des actions opérateur dans Runtime. Au besoin, on configure des alarmes utilisateur également dans des scripts dans Runtime pour des applications définies par l'utilisateur.

Alarmes de l'API : Une alarme de l'API définie par l'utilisateur est programmée dans STEP 7 par le concepteur de l'automate. Les valeurs d'état de l'automate, tels que horodatage et valeurs de process, y sont intégrées. Quand des alarmes de l'API sont configurées dans STEP 7, on les reprend dans le fonctionnement intégré de WinCC dès qu'une connexion à l'automate est établie [21].

8.6.3. Configuration d'alarme :

8.6.3.1. Editeur "Alarmes TOR" :

D'abord on crée une variable interne dans l'éditeur "variable" de type "Int", celui-ci un mot dont lequel, on peut sauvegarder chaque bit des alarmes TOR.

Par exemple un mot (MW 0) peut contenir 16 alarmes de bit de M 0.0 à M0 .7, M 1.0 à M 1.7 .

De même on crée un autre mot qui va servir à acquitter les alarmes si leur type est à acquitter Pour chaque alarme on doit définir :

Numéro (ID): numéro unique servant à l'identification de l'alarme (valeurs admissibles : de 1 à 32767).

Un texte d'alarme : le texte qui va s'afficher au Runtime.

Classe d'alarme : on choisit pour chaque alarme une classe d'alarme parmi celles définies ci-dessus (erreur, avertissement, acquittée, non acquittée).

Variable de déclenchement : c'est la variable de type INT créée auparavant.

Adresse de déclenchement : on choisit un bit du mot créé.

Variable d'acquiescement : la variable créée pour l'acquiescement.

Adresse d'acquiescement : on choisit un bit du mot d'acquiescement.

8.6.3.2. Éditeur "Alarmes Analogiques" :

Cet éditeur permet de créer des alarmes analogiques et d'en définir les propriétés.

Numéro (ID): numéro unique servant à l'identification de l'alarme (valeurs admissibles : de 1 à 32767).

Texte d'alarme: Le texte du message est affiché au Runtime.

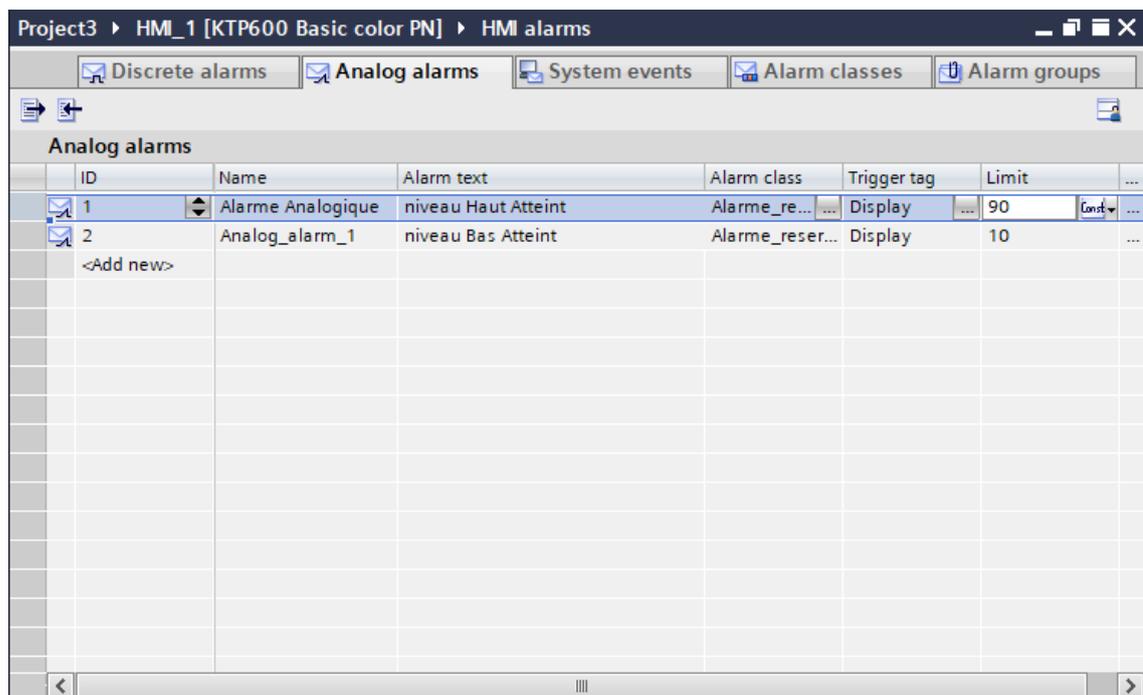
Classe d'alarme: Nom de la classe d'alarmes à laquelle appartient l'alarme (p. exemple erreur, avertissement).

Variable de surveillance : Cette variable fait l'objet d'une surveillance de limites.

Chapitre III

La valeur limite : c'est le seuil de déclenchement de l'alarme, elle peut être indiquée directement sous forme de constante ou indirectement sous forme de variable.

variable de Déclenchement: c'est la méthode de surveillance des valeurs limites. On peut choisir "Si front montant": l'alarme sera déclenchée en cas de dépassement de la valeur limite vers le haut, Ou "Si front descendant" l'alarme sera déclenchée en cas de dépassement de la valeur limite vers le bas.



The screenshot shows a software window titled "Project3 > HMI_1 [KTP600 Basic color PN] > HMI alarms". It contains several tabs: "Discrete alarms", "Analog alarms", "System events", "Alarm classes", and "Alarm groups". The "Analog alarms" tab is active, displaying a table with the following data:

ID	Name	Alarm text	Alarm class	Trigger tag	Limit	
1	Alarme Analogique	niveau Haut Atteint	Alarme_re...	Display	90	Const
2	Analog_alarm_1	niveau Bas Atteint	Alarme_reser...	Display	10	
<Add new>						

Figure III.21 : Vue de configuration l'alarme analogique.

8.6.4. Etats des alarmes :

Chaque alarme se trouve dans un état. Les états des alarmes résultent des événements suivants :

- **Apparaissant** : La condition de déclenchement d'une alarme est remplie. L'alarme s'affiche, par ex. "Pression cuve trop élevée".
- **Disparaissant** : La condition de déclenchement d'une alarme n'est plus remplie. L'alarme ne s'affiche plus, car la cuve a été purgée.
- **Acquitter** : L'utilisateur acquitte l'alarme.

8.7. Les recettes :

Les recettes regroupent des données correspondantes, comme le paramétrage d'une machine ou des données de production. Par exemples :

Chapitre III

_ Des paramétrages de machine qui sont nécessaires pour adapter la production à une autre variante de produit.

_ Des composants qui donnent des produits finaux suivant les proportions de Leur combinaison [21].

8.7.1. Editeur de recette :

On peut créer une recette à partir de la fenêtre "Projet " en cliquant sur " Recettes ". Les différents éléments de la recette sont définis par les paramètres :

Nom de la recette : Identifie la recette de manière univoque dans le pupitre opérateur.

Nom d'affichage : S'affiche dans Runtime, par exemple dans la vue de recette. On peut configurer le nom d'affichage dans plusieurs langues. On peut attribuer des noms ou désignations explicites que l'opérateur peut associer directement à une recette.

Numéro de recette : Identifie la recette de manière univoque dans le pupitre opérateur.

Version : Informations sur la recette. Par défaut, ce sont la date et l'heure de la dernière modification apportée à la recette qui s'affichent.

Chemin : Définit le lieu de stockage pour les recettes. Les recettes sont stockées sous forme de fichier.

Type de taille [non modifiable] : Par défaut, les enregistrements de recette sont limités à un nombre prédéfini.

Nombre d'enregistrements [non modifiable] : Nombre maximal d'enregistrements d'une recette dans Runtime. Ce nombre est limité par la mémoire de recettes du pupitre opérateur.

Type de communication [non modifiable] : Les enregistrements de recette sont écrits directement dans les adresses des variables de recette ou y sont directement lus.

Info-bulle : Info-bulle sur la recette, affichée pour l'opérateur dans Runtime [21].

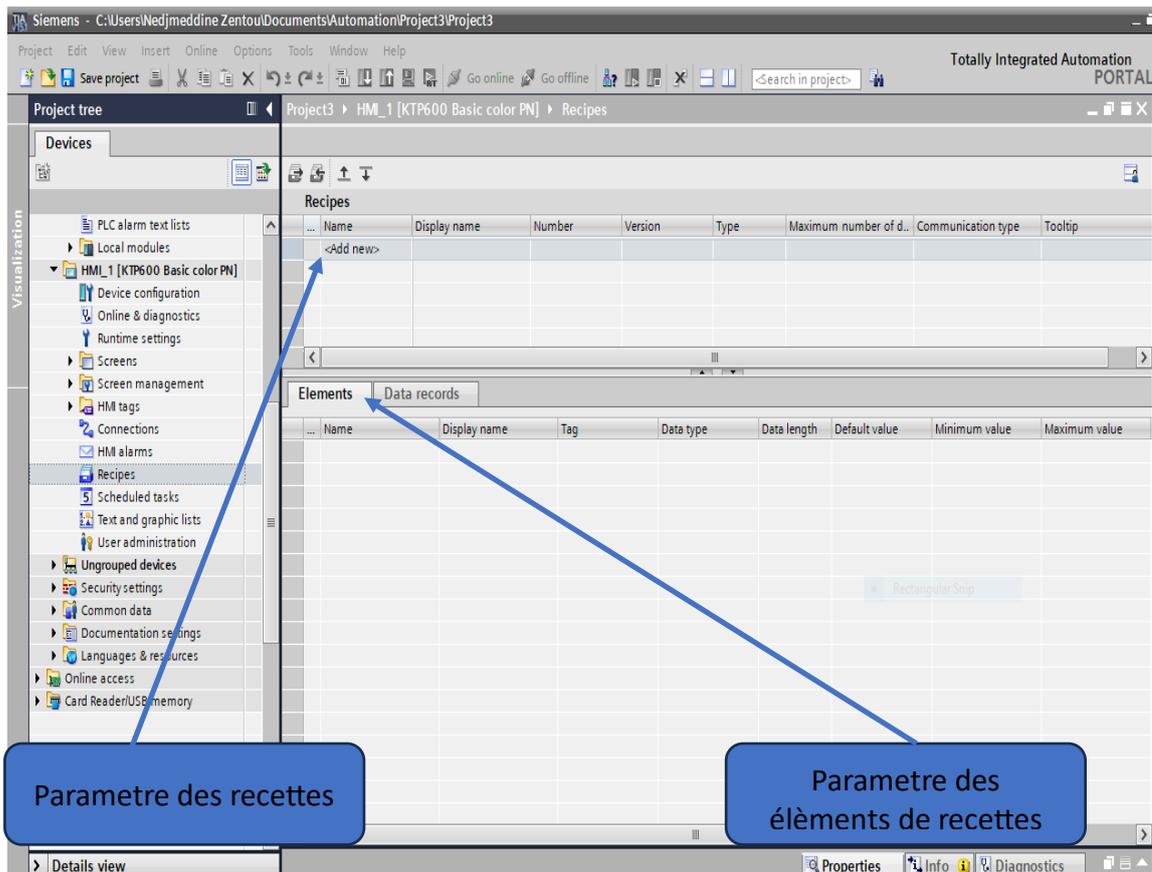


Figure III.22 : Vue configuration de recette.

8.8. Les archives :

Le WinCC propose les types d'archives suivants pour archiver des données de process pour HMI Runtime [21]:

8.8.1. Archives d'alarmes

Dans les archives d'alarmes, toutes les données concernant une alarme qui surviennent dans le process visualisé, y compris les données de configuration, sont enregistrées. A partir des archives, on peut lire par exemple toutes les propriétés d'une alarme, comme la classe d'alarme, l'horodatage et les textes de l'alarme. Une modification ultérieure des données de configuration d'une alarme a pour conséquence qu'un nouveau segment d'archive avec les nouvelles données de configuration est créé. Cette fonction empêche la modification d'avoir une influence sur des alarmes archivées déjà avant la modification. Pour archiver une alarme, réalisez les étapes de configuration suivantes :

- **Définir les "Paramètres d'archives"** : détermination des paramètres pour l'archivage d'alarmes dans les paramètres d'archives.
- **Activation de l'archivage d'alarmes** : décision d'archiver les alarmes de certaines classes d'alarmes.

- **Affectation de l'alarme à la classe d'alarmes archivable.**

8.8.2. Archives de variables

Une archive de variables sert à archiver les données de process d'une installation industrielle. Le système d'archives se charge en Runtime de l'archivage des variables. Le système d'archives traite les valeurs du process sauvegardées dans la base de données Runtime et les inscrit dans la base de données d'archives.

Les divers paramètres déterminent si et quand les valeurs du process doivent être acquises et archivées. La méthode d'archivage utilisée détermine ceux des paramètres suivants à configurer

- **Cycle d'acquisition** : Détermine quand la valeur d'une variable du process sera lue dans l'automate programmable. On configure un cycle d'acquisition par exemple pour l'archivage cyclique des variables.
- **Cycle d'archivage** : Détermine quand la valeur du process sera enregistrée dans la base de données d'archives une fois traitée. On configure un cycle d'archivage par exemple pour l'archivage cyclique des variables.
- **Variable marche-arrêt** : Lance d'archivage des variables quand la variable binaire affectée prend la valeur "1". L'archivage s'arrête dès que la condition de démarrage n'est plus remplie. On configure la variable marche-arrêt pour l'archivage cyclique sélectif des variables.
- **Archivage sur demande** : Les valeurs de process sont archivées à la demande. La demande est pilotée par une variable binaire. Pour la configuration, utiliser le mode de déclenchement "Sur demande".
- **Archivage de variable sur modification** : Les valeurs de processus sont uniquement archivées en cas de modification. Pour la configuration, utiliser le mode de déclenchement "Sur modification".

8.8.3. Les méthodes d'archivage

Les méthodes d'archivage suivantes sont disponibles :

- **Archive cyclique** : Quand une archive cyclique est entièrement remplie, les entrées les plus anciennes sont écrasées.
- **Archive cyclique segmentée** : Dans une archive cyclique segmentée, plusieurs segments d'archive de même taille sont créés et remplis l'un après l'autre. Quand

tous les segments d'archive sont entièrement remplis, le segment d'archive le plus ancien est écrasé.

- **Archive avec alarme système en fonction du niveau de remplissage** : Une alarme système se déclenche quand un niveau de remplissage défini est atteint.
- **Archive avec déclenchement d'un événement en fonction du niveau de remplissage** : Quand l'archive est entièrement remplie, l'événement "Débordement" est déclenché. Avec cet événement "Débordement", une fonction système est déclenchée.

8.9. Le test de la visualisation du processus :

Runtime est un logiciel utilisé pour la visualisation du processus des projets créés avec le logiciel WinCC dans le TIA Portal.

La simulation de l'HMI permet de tester le bon fonctionnement de la visualisation avant de démarrer le processus.

Avant tout la configuration matérielle et les blocs de programme de la CPU doivent être chargés dans PLCSIM.

On clique sur le HMI RT dans la navigation du projet et on démarre la simulation du Runtime dans le menu contextuel.

Les éléments configurés du HMI RT sont automatiquement compilés avant le démarrage du Runtime. L'état de compilation s'affiche dans la fenêtre d'inspection de l'onglet "Info".

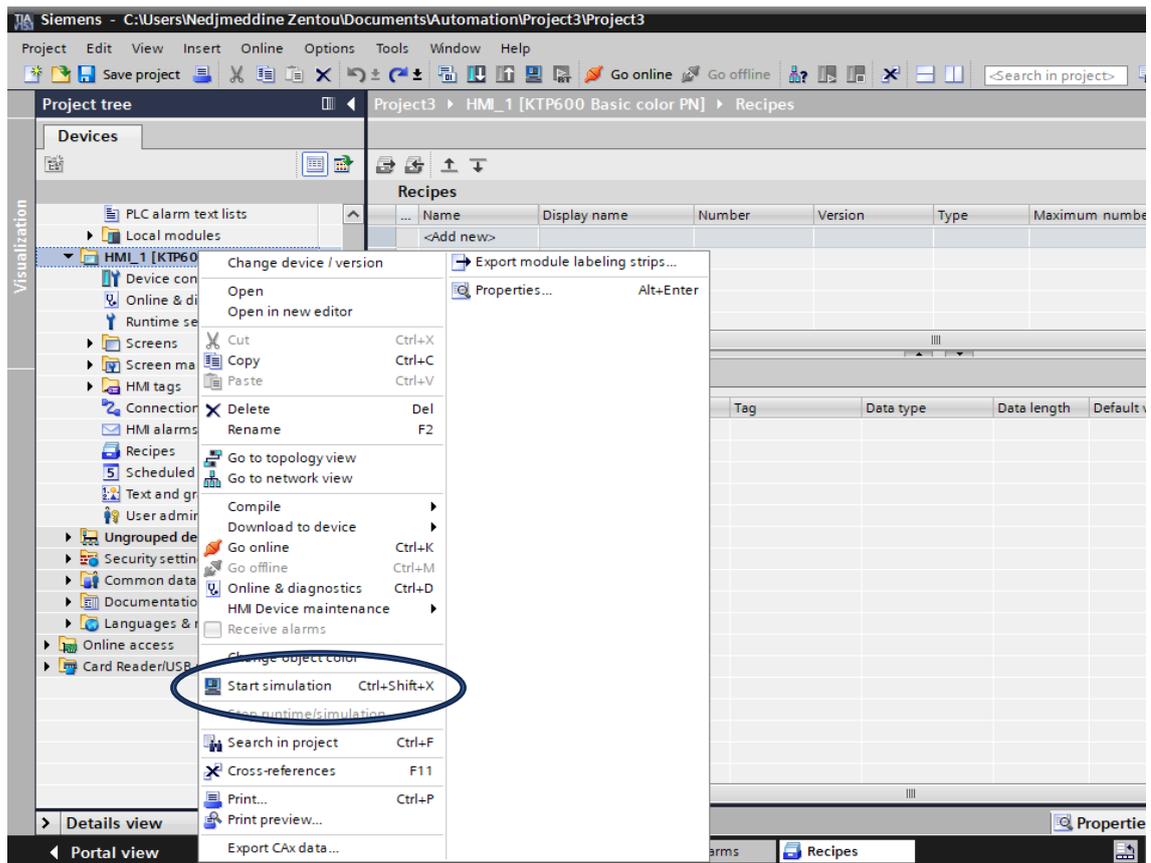


Figure III.23 : Lancement du RUNTIME.

9. Conclusion :

Ce chapitre regroupe les fonctionnalités essentielles du logiciel de programmation de SIEMENS le TIA Portal, utilisé pour la configuration matériel, la programmation en différents langages, la simulation des programmes ainsi que la création et paramétrage des interfaces HMI pour la visualisation des process.

IV. Chapitre IV : La Simulation et La Supervision.

1. Introduction :

Dans ce chapitre, nous allons aborder la simulation d'un système de régulation de niveau de réservoir (Tank Level) en utilisant Factory IO pour la modélisation et TIA Portal pour la programmation et le contrôle via un automate virtuel. Le but est de simuler un processus industriel où un automate régule le niveau de liquide dans un réservoir en fonction des capteurs de niveau et des actionneurs tels que des vannes.

Nous décrirons les étapes pour configurer un environnement virtuel, établir une communication avec l'automate virtuel et simuler la régulation du niveau dans le réservoir.

2. Présentation de Factory IO :

Factory IO est un simulateur 3D permettant de modéliser des systèmes automatisés industriels. Pour le projet de niveau de réservoir, il permet de créer un environnement où un réservoir, des capteurs de niveau, et des pompes ou vannes peuvent être ajoutés et configurés.



Figure IV.1 : Logiciel Factory IO.

3. Étapes de Configuration et de Connexion :

- Nous commençons d'abord par ouvrir logiciel TIA Portal V15.1 :

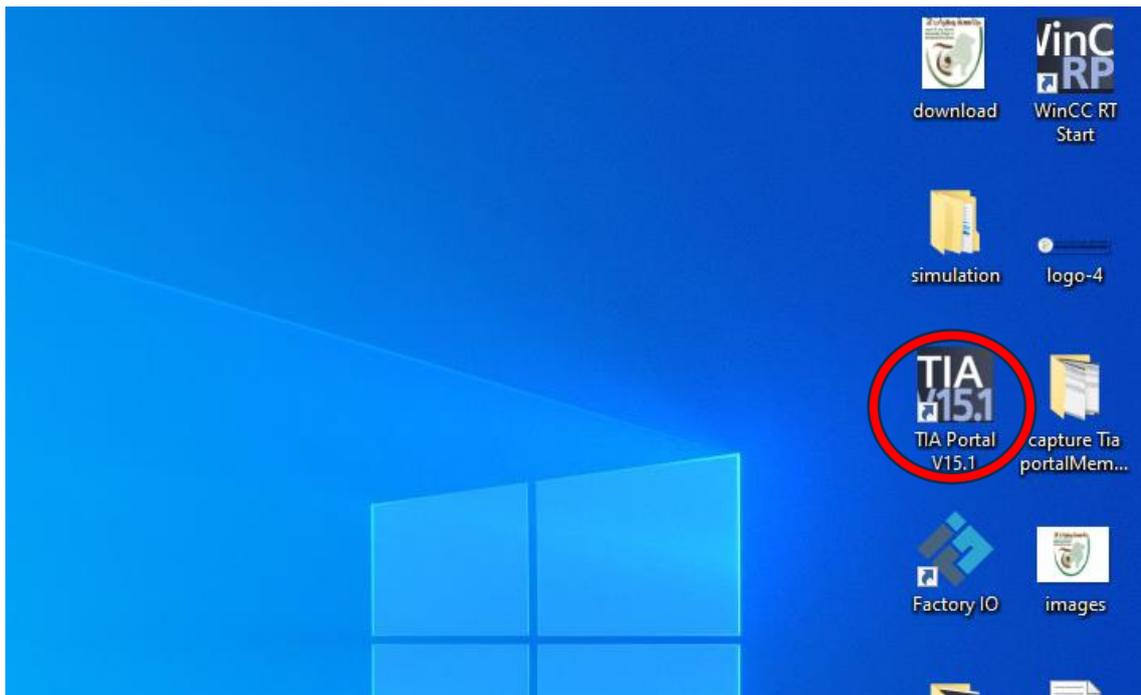


Figure IV.2 : Ouverture de TIA Portal.

- Apres, « Open Project » :

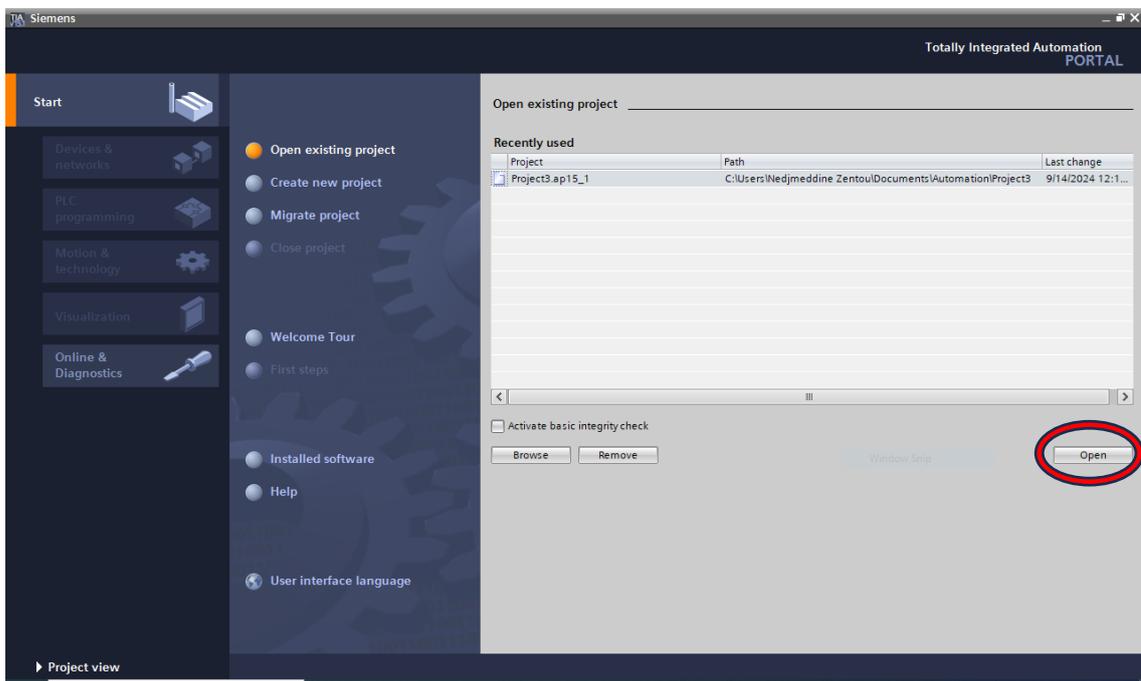


Figure IV.3 : Ouvrir le projet.

- Taper sur “Project view” :

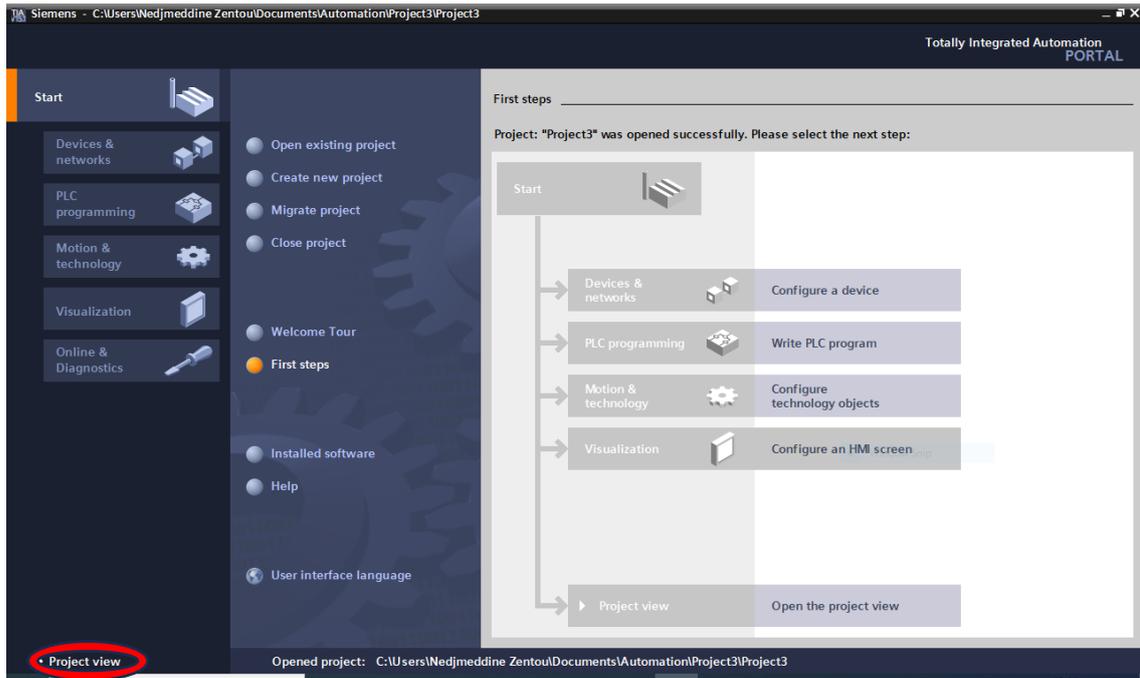


Figure IV.4 : Vue de Projet.

- Commencer la simulation :

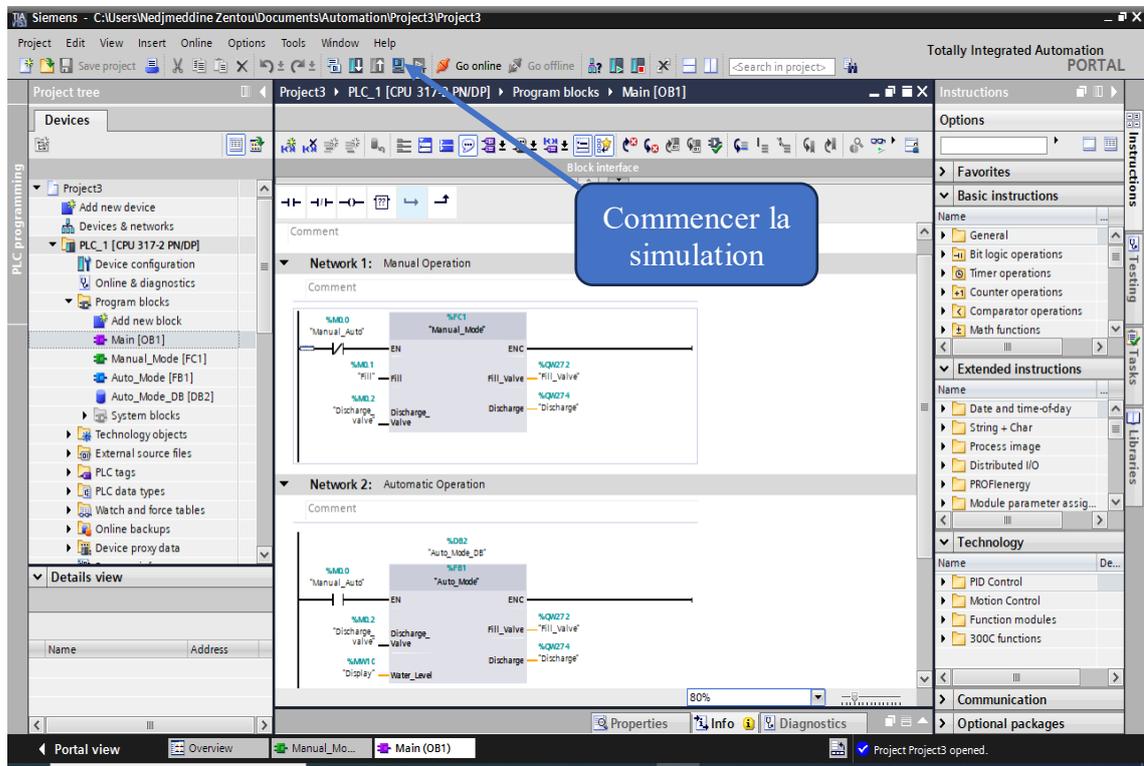


Figure IV.5 : Commencer la simulation.

- Charger Le PLC :

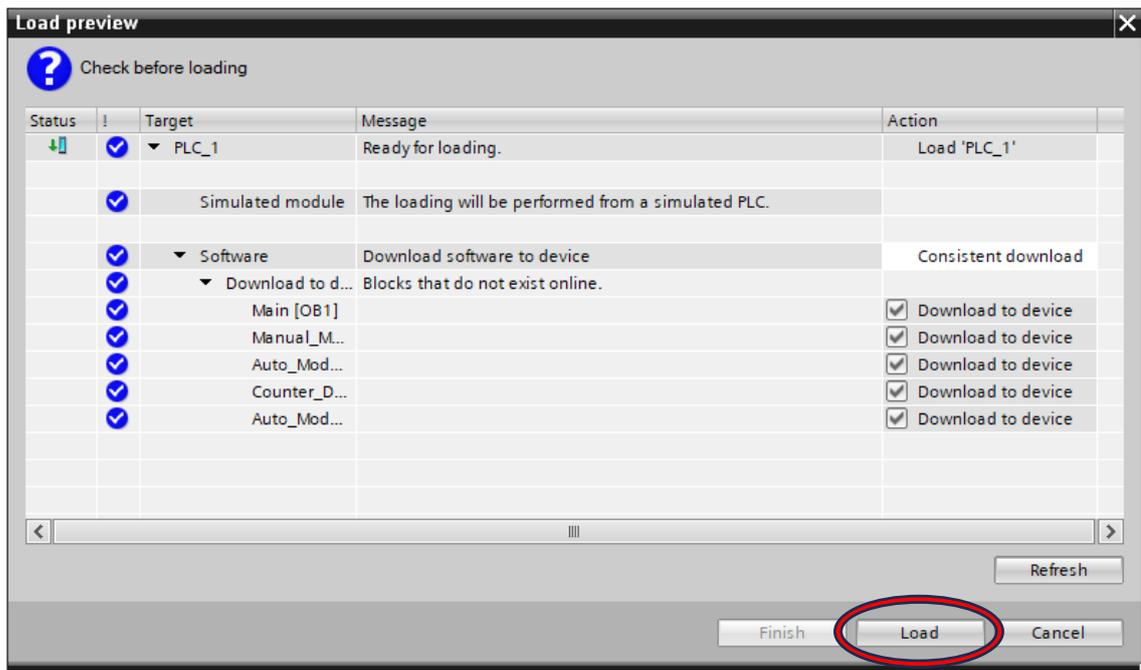


Figure IV.6 : Charger le PLC.

- Logiciel S7-PLCSIM s'ouvre automatiquement :

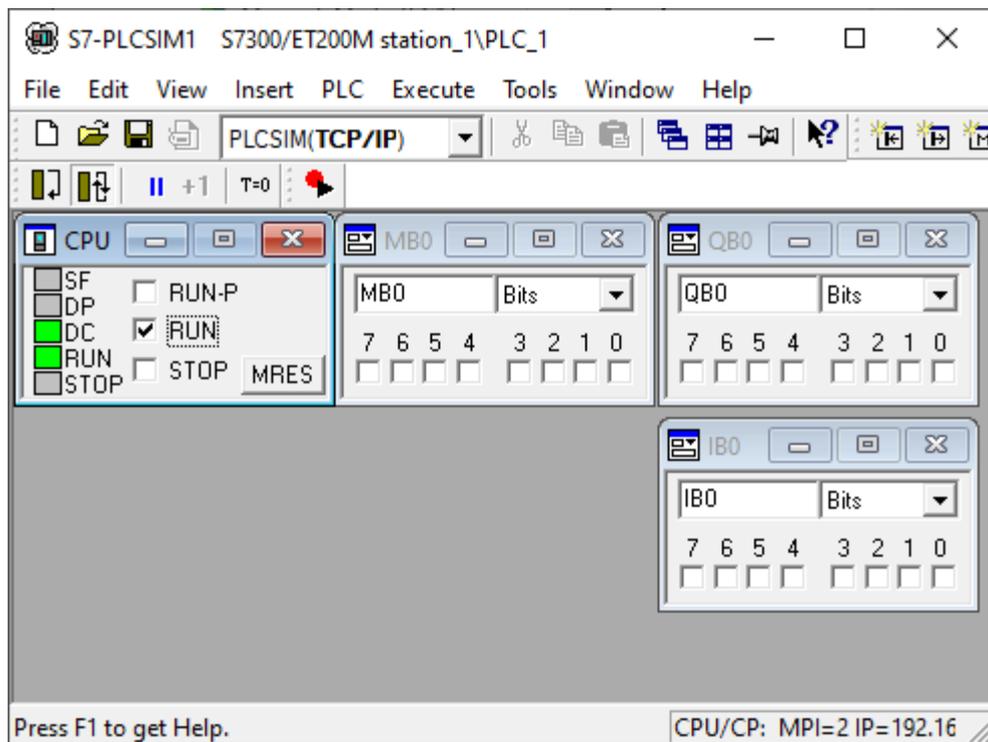


Figure IV.7 : S7-PLCSIM.

Chapitre IV

- Commencer la simulation de IHM : sélectionner L'IHM après tapez sur « Start Simulation ».

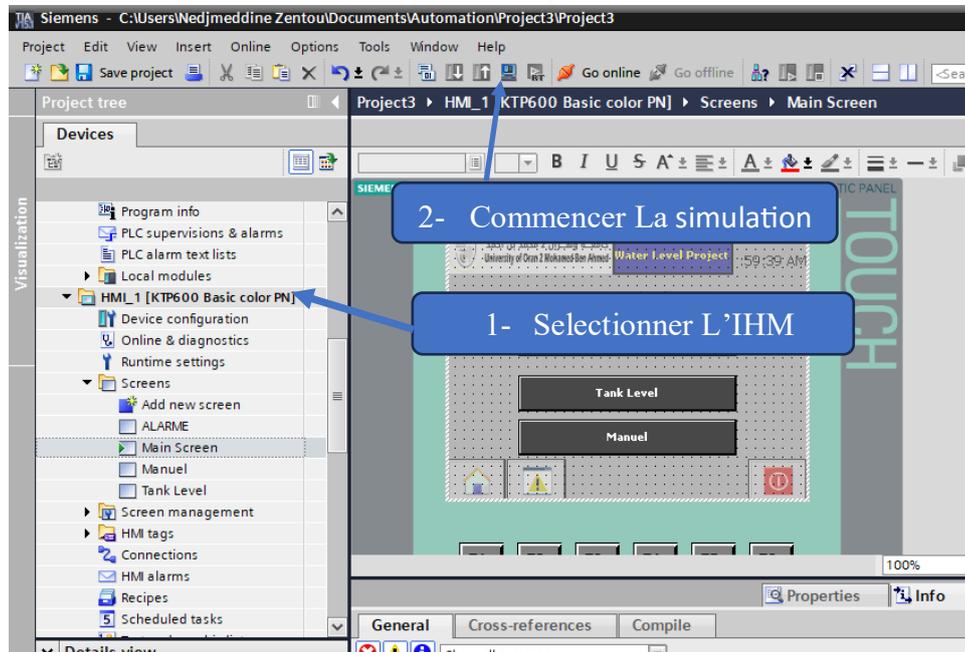


Figure IV.8 : Commencer la simulation de L'IHM.

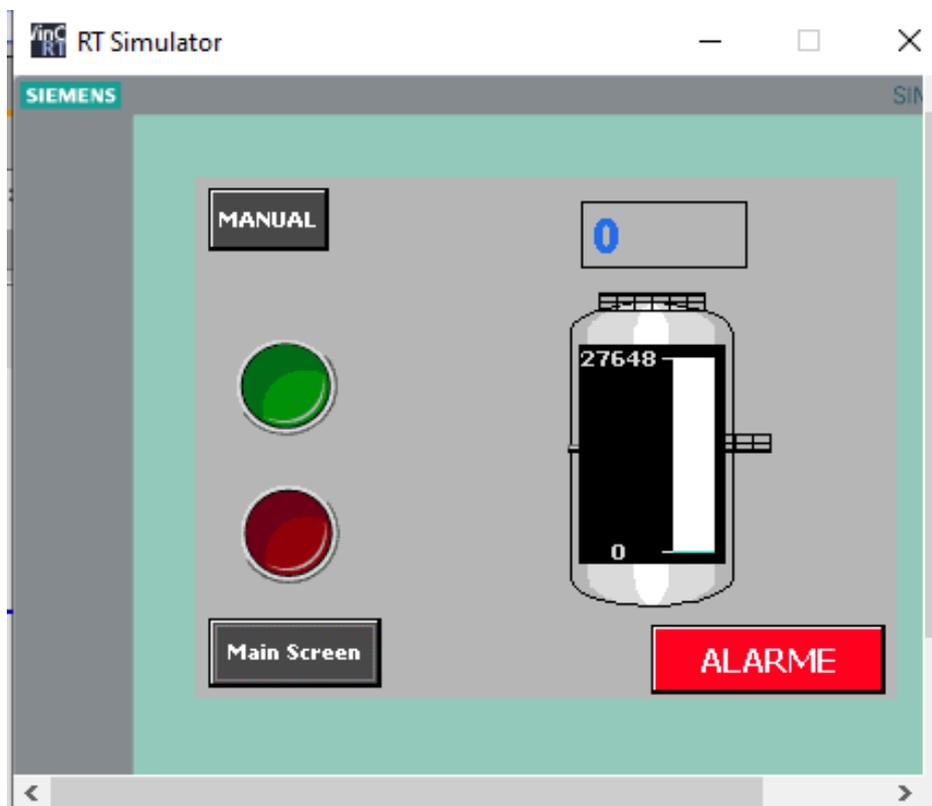


Figure IV.9 : Vue de L'IHM sur RUNTIME.

Chapitre IV

- Après avoir termine avec logiciel TIA Portal, nous ouvrons logiciel Factory IO :

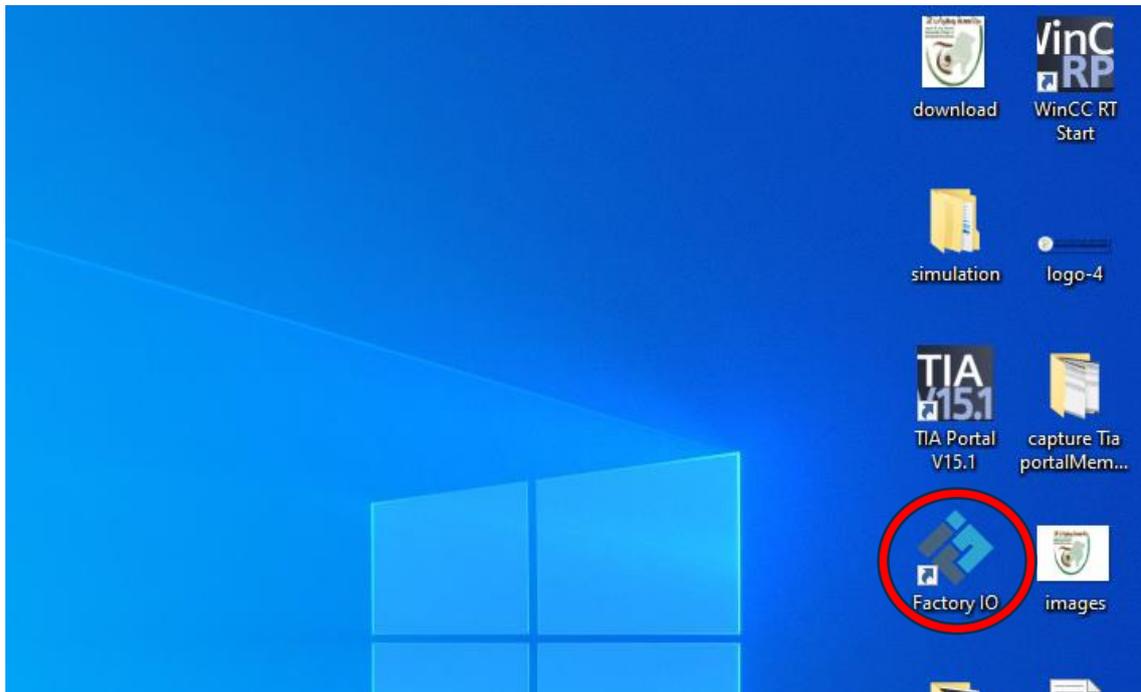


Figure IV.10 : Ouvrir Logiciel Factory IO.

- On ajoute une nouvelle Scène pour notre projet « Level Control » :

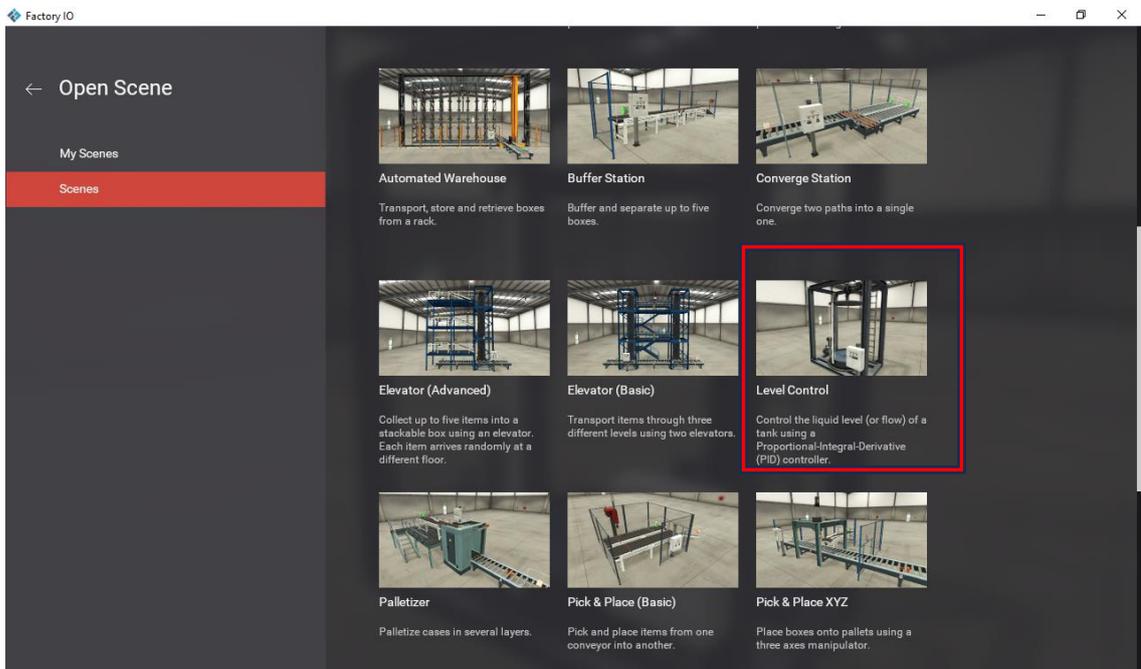


Figure IV.11 : Controle de Niveau.

Chapitre IV

- On appuie sur « Drivers » pour configurer les adresses de I/O :

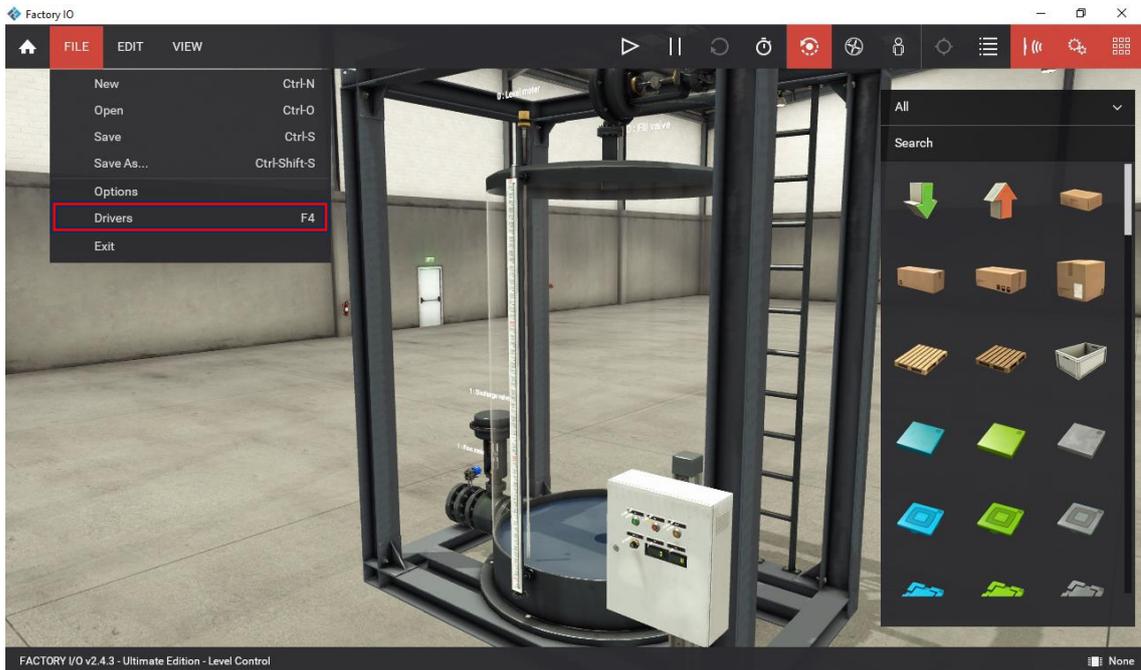


Figure IV.12 : Selectionner "Drivers".

- On choisit Siemens S7-PLCSIM :

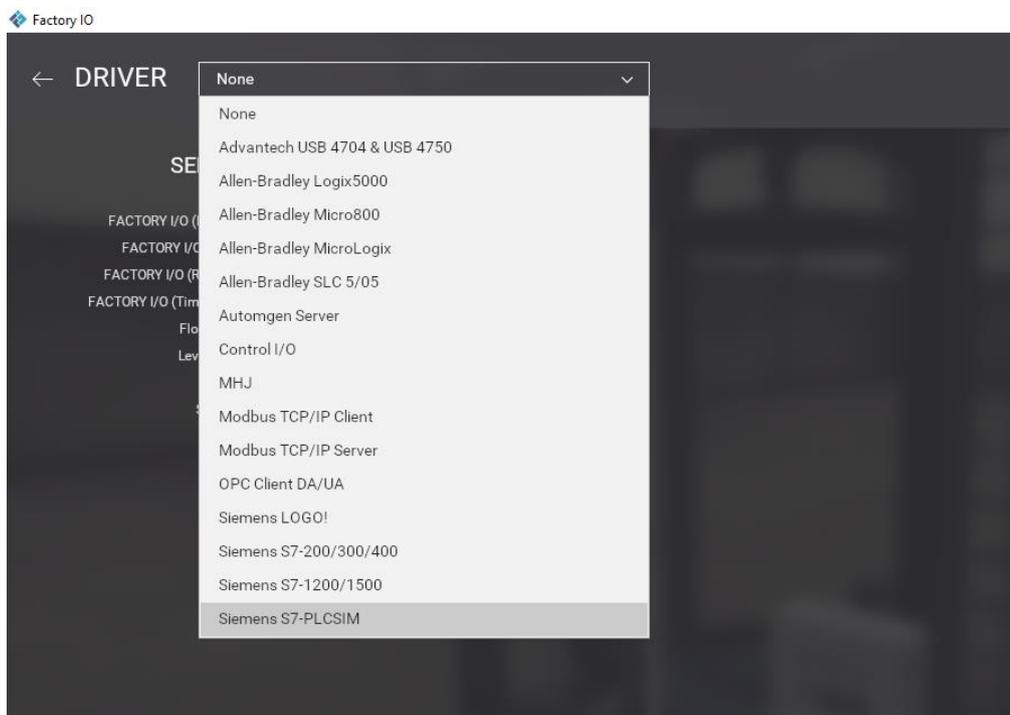


Figure IV.13 : Choisir Siemens S7-PLCSIM.

Chapitre IV

- On appuie sur Configuration pour entrer les I/O.

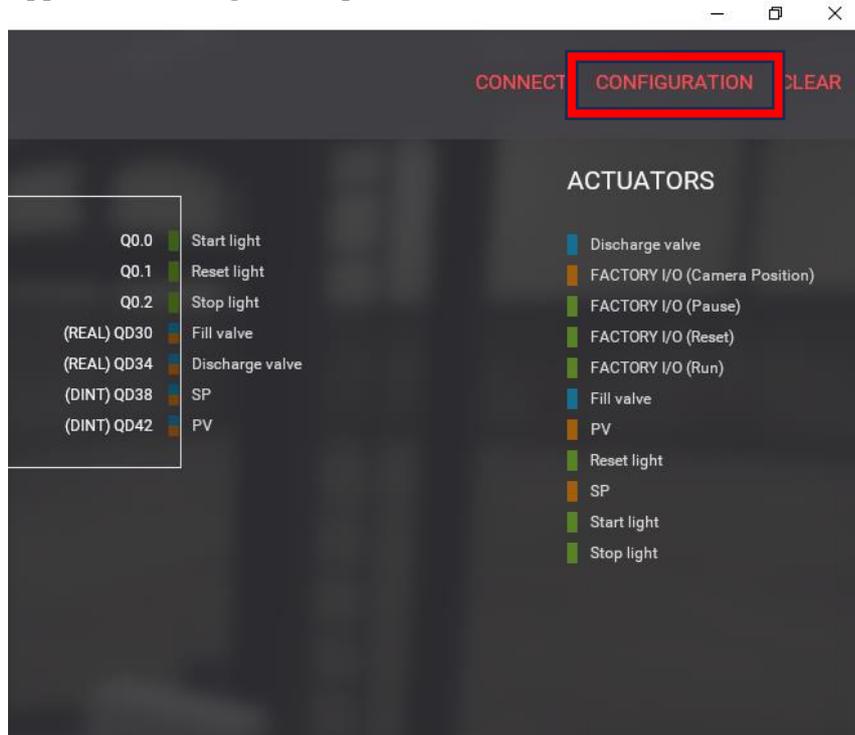


Figure IV.14 : Appuyer sur "Configuration".

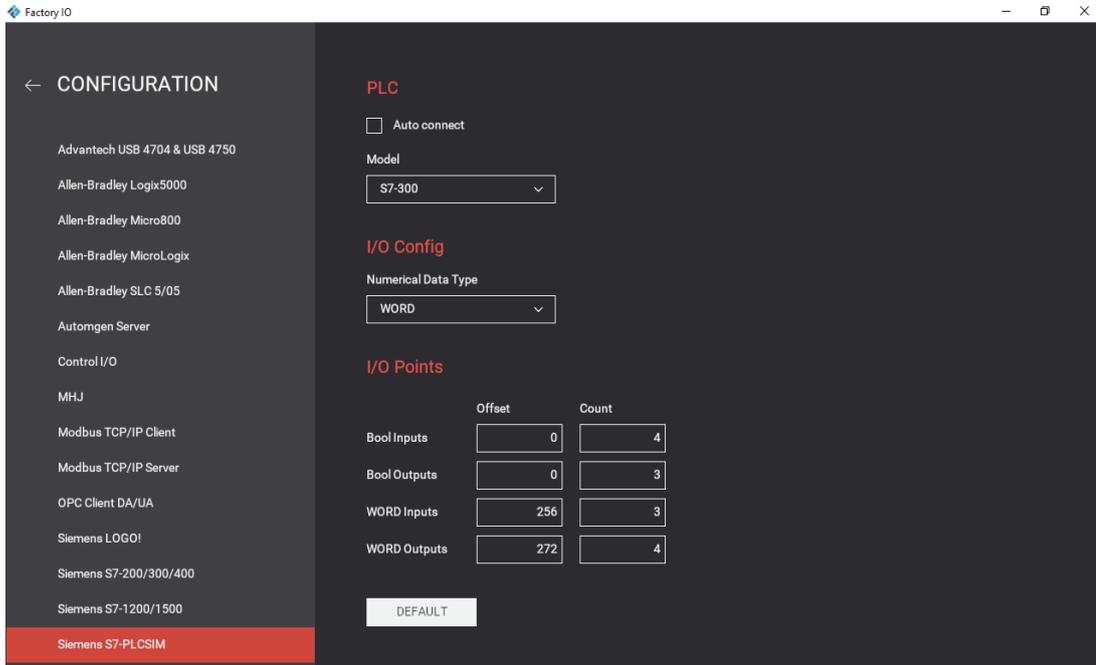


Figure IV.15 : La configuration des adresses I/O.



Figure IV.16 : Schema des I/O et Variable de Simulation dans Factory IO.

- Appuie Sur « Connect » pour connecter Factory IO avec L'Automate programmable.

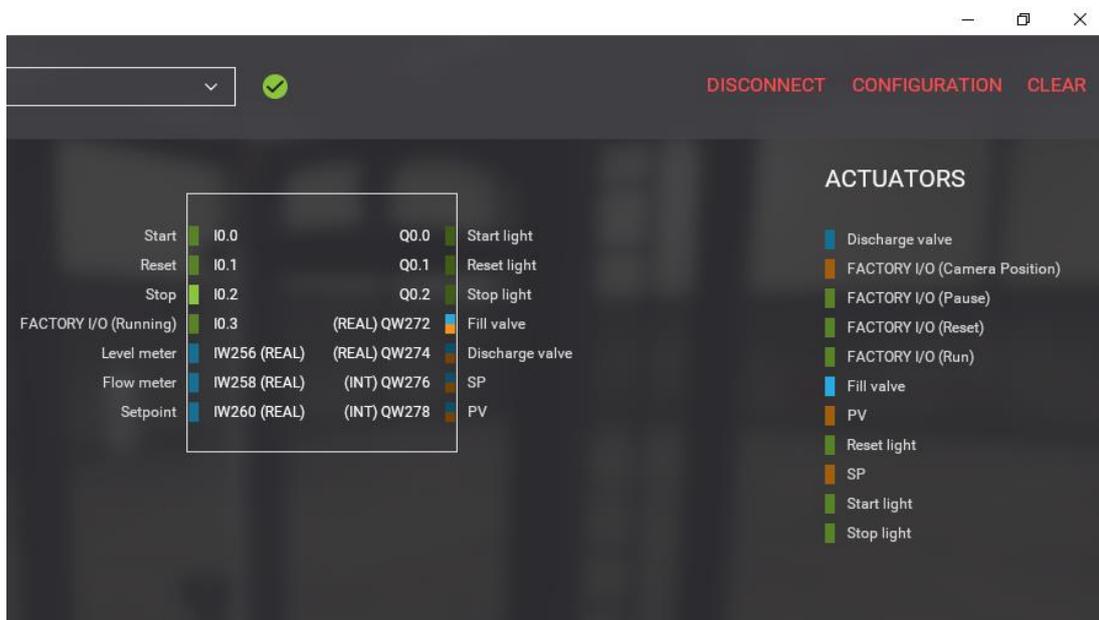


Figure IV.17 : Connecter Factory IO avec PLCSIM.

Chapitre IV

- En fin, on commence la simulation de Factory IO :

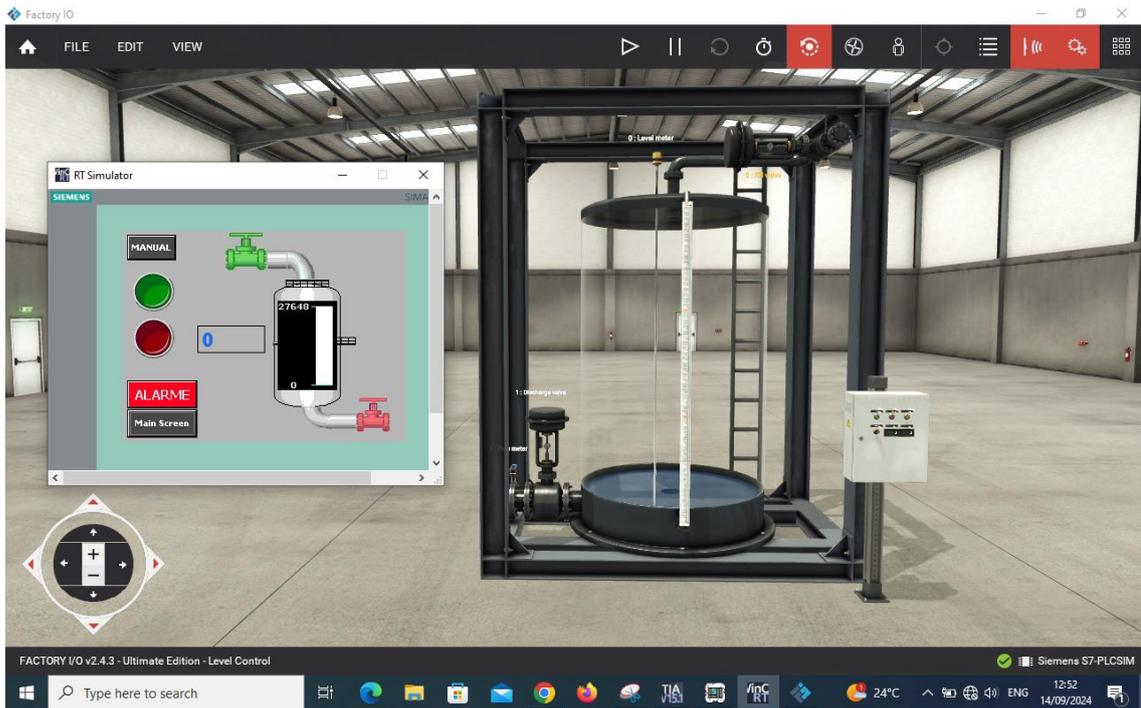


Figure IV.18 : Commencer la simulation de Factory IO.

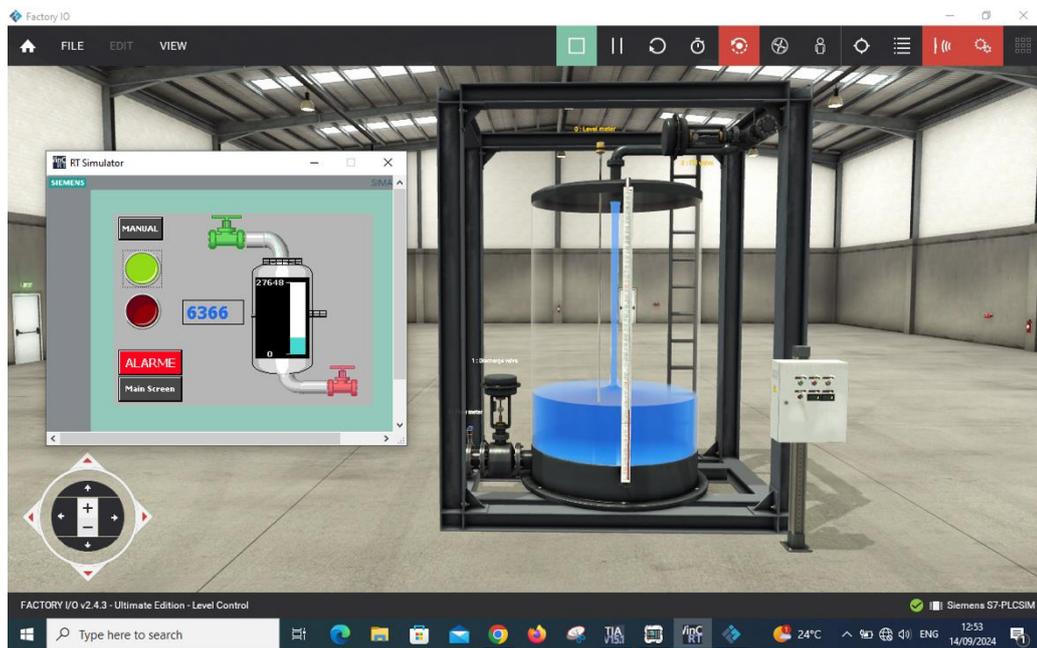


Figure IV.19 : Ouvrir la vanne remplissage.



Figure IV.20 : Ouvrir la vanne de décharge.

- Pour surveiller le niveau d'eau on utilise la fenêtre « ALARME ».
- Si le niveau d'eau tombe en dessous d'un seuil prédéfini (Niveau Minimum), une alarme de niveau bas est déclenchée.
- Si le niveau d'eau dépasse un seuil prédéfini (Niveau Maximum), une alarme de niveau haut est activée.

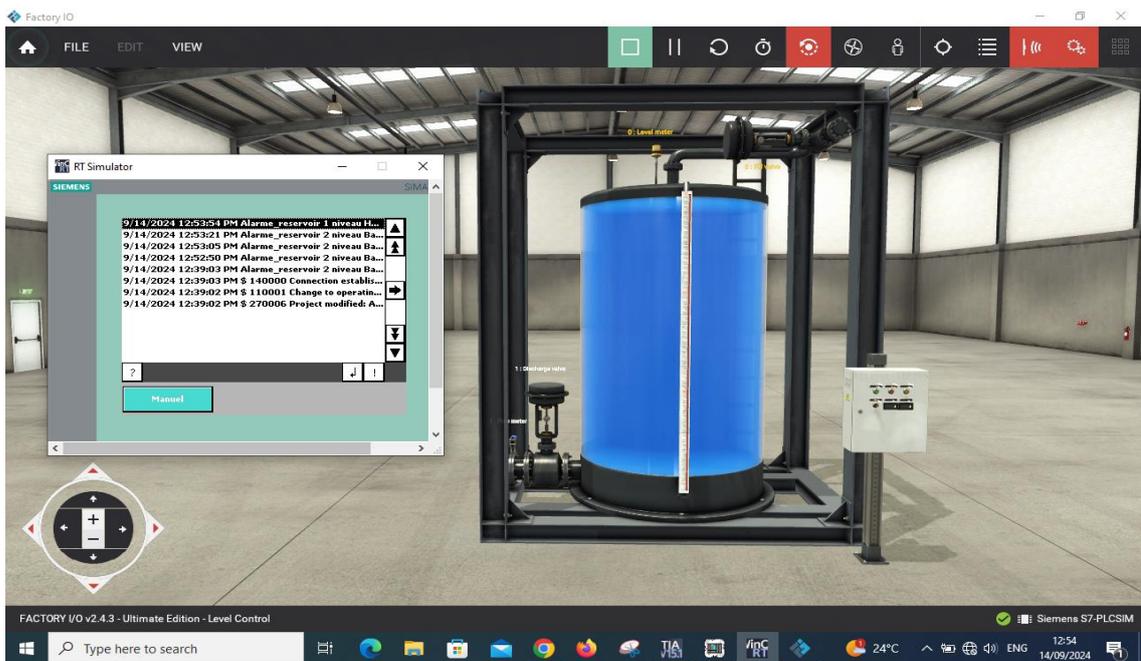


Figure IV.21 : Vue d'Alarme.

4. Conclusion :

La simulation du système de niveau de réservoir à l'aide de Factory I/O et d'un automate virtuel dans TIA Portal permet de tester et valider le comportement de régulation du niveau d'eau. Ce projet illustre l'importance de la simulation avant la mise en œuvre d'un système automatisé réel, minimisant les erreurs potentielles et optimisant la performance.

CONCLUSION GÉNÉRALE

CONCLUSION GÉNÉRALE :

Ce mémoire a démontré l'importance de l'intégration de la programmation et de la simulation dans la gestion des systèmes automatisés. Grâce à l'utilisation de "TIA Portal" pour la programmation des automates et de "Factory IO" pour la simulation des processus industriels, il a été possible de concevoir un système complet de surveillance à distance des équipements d'instrumentation. La simulation a joué un rôle crucial en permettant de tester et de valider les configurations avant leur implémentation réelle, réduisant ainsi les risques et les coûts liés aux erreurs de conception. L'approche intégrée proposée offre des perspectives prometteuses pour l'avenir des industries, où la virtualisation des processus permettra une automatisation plus intelligente, plus sécurisée et plus efficace.

Ces versions incluent les logiciels utilisés tout en soulignant leur importance dans votre étude.

Dans des travaux futurs on pourra envisager :

- La réalisation de ce projet sur le terrain, ainsi que son développement et son intégration avec l'intelligence artificielle.
- Améliorer la supervision par l'ajout de fonction de notification par envoi de sms et mail.

RÉFÉRENCE BIBLIOGRAPHIES :

- [1] J. Lee, H. A. Kao, et S. Yang, "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3-8, 2014.
- [2] Y. Li, L. Gao, et X. Zhao, "A real-time condition monitoring system for machine tools based on cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2391-2401, 2016.
- [3] J. Lee, H. A. Kao, et S. Yang, "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3-8, 2014.
- [4] K. G. Borlea et C. Anghel, "Technologies and applications for remote monitoring and control in the Industry 4.0 era," in *Proceedings of the 2019 IEEE International Conference on Automation, Quality, and Testing, Robotics (AQTR)*, pp. 1-6, 2019.
- [5] R. K. Natarajan et al., "Proactive Maintenance and Reliability: Role of Instrumentation," *Maintenance and Reliability Journal*, vol. 8, pp. 43-51, 2018.
- [6] L. H. Vargas et al., "Regulatory Compliance through Advanced Instrumentation," *Journal of Industrial Compliance*, vol. 5, no. 4, pp. 29-38, 2017.
- [7] J. Lopez, R. Rios, et A. Gomez, "A Review of Wireless Sensor Networks for Real-Time Condition Monitoring," *Sensors*, vol. 12, pp. 10929-10945, 2012.
- [8] K. G. Borlea et C. Anghel, "Technologies and applications for remote monitoring and control in the Industry 4.0 era," in *Proceedings of the 2019 IEEE International Conference on Automation, Quality, and Testing, Robotics (AQTR)*, pp. 1-6, 2019.
- [9] Blanchard, A. (2024). *Advanced Materials in Transducer Technology*. *Journal of Modern Engineering*.
- [10] S. M. Stewart et al., "Instrumentation in the Oil and Gas Industry: A Comprehensive Review," *Journal of Petroleum Science and Engineering*, vol. 176, pp. 122-135, 2019.
- [11] Y. Li, L. Gao, et X. Zhao, "A real-time condition monitoring system for machine tools based on cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2391-2401, 2016.
- [12] J. Lee, H. A. Kao, et S. Yang, "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3-8, 2014.
- [13] P. S. Raju et K. P. Madhusudhan, "Smart Instrumentation and Control Systems: Applications and Future Trends," *Journal of Control and Automation Engineering*, vol. 2, no. 3, pp. 152-160, 2015.

RÉFÉRENCE BIBLIOGRAPHIQUES

- [14] R. K. Natarajan et al., "Proactive Maintenance and Reliability: Role of Instrumentation," Maintenance and Reliability Journal, vol. 8, pp. 43-51, 2018
- [15] L. H. Vargas et al., "Regulatory Compliance through Advanced Instrumentation," Journal of Industrial Compliance, vol. 5, no. 4, pp. 29-38, 2017.
- [16] J. Lopez, R. Rios, et A. Gomez, "A Review of Wireless Sensor Networks for Real-Time Condition Monitoring," Sensors, vol. 12, pp. 10929-10945, 2012.
- [17] Manuel système d'automatisation S7-300 caractéristique de modules 2013
- [18] Automate programmable Siemens- Logiciel Siemens [en ligne], <http://www.lcautomatisme.fr/15.html>
- [19] SIEMENS, « SIMATIC Step7 dans le TIA Portal », SIMATIC 2013.
- [20] SIEMENS, « SIMATIC Wincc dans le TIA Portal », SIMATIC 2013.
- [21] Help TIA Portal.