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 أ حمد

Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة والأمن الصناعي

Institut de Maintenance et de Sécurité Industrielle

Département Maintenance en instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie industrielle

Spécialité : Maintenance en automatisme et instrumentation

Thème

Aide aux diagnostics dans un contexte industrie 4.0

Présenté et soutenu publiquement par :

Nom : BELGACEM **Prénom** : OMAR

Nom : AISSANI **Prénom** : ABD EL MONSIF

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
	MCA		**Président**
Pr. **Aissani Nassima**	MCA		**Encadrante**
	MCA		**Examinateur**

Année universitaire 2025/2026

Remerciement

Nous tenons à remercier toutes les personnes qui nous ont aidés à réaliser ce travail.

Nous remercions vivement ***Madame Aissani Nassima*** *pour son encadrement, ses conseils et le temps qu'elle nous a consacré. Son aide a été très importante pour nous.*

Nous remercions également tous les ***enseignants de l'université*** *pour la formation qu'ils nous ont offerte et pour leur soutien.*

Nous adressons nos sincères remerciements aux ***membres du jury*** *qui ont accepté de lire et d'évaluer ce travail. Nous leur sommes reconnaissants pour leur temps et leur attention.*

Enfin, nous remercions nos familles et nos proches pour leur soutien.



Dédicaces

Je dédie ce travail à toute ma famille ***BELGACEM****. À mes parents, pour leur amour et leur soutien. À mes frères et sœurs, pour leurs encouragements.*

Je dédie ce travail à tous ceux qui m'ont aidé tout au long de ce parcours. À mes amis et à mes proches, pour leur présence et leur aide.

Enfin, je dédie ce travail à toutes les personnes qui ont cru en moi.

Merci à tous pour votre soutien.



Dédicaces

*À ma famille **AISSANI**. À vous qui m'avez toujours soutenu et encouragé.*

Ce travail est le fruit de votre amour et de vos sacrifices.

À tous ceux qui m'ont tendu la main pendant ces années d'études.

À mes amis, à mes proches, à tous ceux qui ont cru en moi.

Un merci spécial à toutes les personnes qui ont contribué, de près ou de loin, à la réussite de ce parcours.

Résumé :

Ce travail porte sur la mise en place d'une architecture Internet Industriel des Objets (IIoT) appliquée à un système de régulation de niveau liquide. Une plateforme de simulation complète a été développée, s'appuyant sur un automate programmable industriel (API) virtuel SIMATIC S7-1500 via TIA Portal et S7-PLCSIM Advanced. La logique de commande, incluant la gestion marche/arrêt, le remplissage, la vidange et la mise à l'échelle des mesures, a été programmée en langage Ladder. Cette infrastructure virtuelle a été interconnectée à Node-RED, utilisé comme passerelle IoT, grâce au protocole S7. Les données temps réel (niveau d'eau, état des pompes, alarmes) sont publiées sur un broker MQTT (EMQX) et visualisées sur un tableau de bord dynamique (Dashboard Node-RED). Le système simule le comportement d'un réservoir PCT50, intégrant des seuils de sécurité (haut/bas) et des stratégies de régulation. Les résultats démontrent la faisabilité d'un jumeau numérique complet, permettant la supervision et le contrôle à distance sans matériel physique, tout en respectant les contraintes industrielles de latence et de fiabilité. Cette approche offre une solution flexible et scalable pour la formation, le prototypage rapide et le déploiement d'applications IIoT

Mots-clés : IIoT, Industrie 4.0, Automate virtuel (PLCSIM Advanced), Node-RED, MQTT, Régulation de niveau, TIA Portal, Jumeau numérique.

Abstract:

This work focuses on the implementation of an Industrial Internet of Things (IIoT) architecture applied to a liquid level control system. A complete simulation platform was developed using a virtual SIMATIC S7-1500 Programmable Logic Controller (PLC) via TIA Portal and S7-PLCSIM Advanced. The control logic, including start/stop management, filling, draining, and measurement scaling, was programmed in Ladder language. This virtual infrastructure was interconnected to Node-RED, used as an IoT gateway, through the S7 protocol. Real-time data (water level, pump status, alarms) are published to an MQTT broker (EMQX) and visualized on a dynamic dashboard (Node-RED Dashboard). The system simulates the behavior of a PCT50 tank, integrating safety thresholds (high/low) and control strategies. The results demonstrate the feasibility of a complete digital twin, enabling remote monitoring and control without physical hardware, while meeting industrial constraints of latency and reliability. This approach offers a flexible and scalable solution for training, rapid prototyping, and deployment of IIoT applications.

Keywords: IIoT, Industry 4.0, Virtual PLC (PLCSIM Advanced), Node-RED, MQTT, Level Control, TIA Portal, Digital Twin.

الملخص

يركز هذا العمل على تنفيذ بنية إنترنت الأشياء الصناعي (IIoT) المطبقة على نظام للتحكم في مستوى السائل. تم تطوير منصة محاكاة كاملة باستخدام متحكم منطقي قابل للبرمجة (PLC) افتراضي من نوع SIMATIC S7-1500عبر TIA Portal و .S7-PLCSIM Advancedتمت برمجة منطق التحكم، بما في ذلك إدارة التشغيل/الإيقاف، والتعبئة، والتفريغ، وتوسيع نطاق القياسات، بلغة السلم .(Ladder) تم ربط هذه البنية التحتية الافتراضية بـNode-RED ، المستخدم كبوابة إنترنت الأشياء، عبر بروتوكول S7.يتم نشر البيانات في الوقت الفعلي (مستوى الماء، حالة المضخات، الإنذارات) إلى وسيط MQTT (EMQX)وعرضها على لوحة تحكم ديناميكية .(Node-RED Dashboard) يحاكي النظام سلوك خزانPCT50 ، ويتضمن عتبات أمان (مرتفع/منخفض) واستراتيجيات تحكم. تُظهر النتائج جدوى توأم رقمي كامل، مما يتيح المراقبة والتحكم عن بعد دون أجهزة مادية، مع تلبية قيود زمن الوصول والموثوقية الصناعية. يقدم هذا النهج حلاً مرناً وقابلاً للتطوير للتدريب، والنمذجة الأولية السريعة، ونشر تطبيقات إنترنت الأشياء الصناعي.

الكلمات المفتاحية :إنترنت الأشياء الصناعي، الصناعة 4.0، متحكم منطقي افتراضي(PLCSIM Advanced) ، Node-RED، MQTT، التحكم في المستوى، TIA Portal، التوأم الرقمي.

Sommaire

1.1 Introduction 1

1.2 Industrie 4.0 1

1.3 Systèmes CyberPhysiques (CPS) 2

1.4 Systèmes d'automatisation et de contrôle industriels (IACS) 3

1.5 Systèmes de supervision et d'acquisition de données (SCADA) 3

1.5.1 Le procédé physique: L'installation industrielle à superviser (usine de production, réseau de distribution d'eau, feux de signalisation, etc.). 4

1.5.2 L'infrastructure intelligente: 4

1.6 Definition de l'internet des objets Industriel (IIoT) 5

1.7 Équipements intelligents (Smart Devices) 7

1.8 Passerelles IoT (IoT Gateways) 7

1.9 Plateformes Cloud 8

1.9.1 Siemens MindSphere 8

1.9.2 IBM Private Cloud for Manufacturing 8

1.9.3 Amazon Web Services (AWS) 9

1.9.4 Protocoles de données 9

1.10 Le protocole MQTT 10

1.10.1 Contexte historique 10

1.10.2 Caractéristiques du protocole MQTT 10

1.10.3 Architecture MQTT 11

1.10.4 Les Topics MQTT 11

1.11 Le protocole HTTP 12

1.11.1 Contexte historique 12

1.11.2 L'émergence du World Wide Web (WWW) 12

1.11.3 Définition du protocole HTTP 13

1.11.4 Architecture du protocole HTTP 13

1.11.5 Composants d'une architecture HTTP 14

1.11.6 Localisateur uniforme de ressource (URL) 15

1.11.7 Le protocole HTTP dans le contexte de l'IoT 17

1.11.8 Le protocole OPC UA (OPC Unified Architecture) 17

1.12 Protocoles réseau (Couche Transport et Accès) 18

1.12.1 Le WiFi, ... 18

1.12.2 La 4G 19
1.12.3 L'Ethernet 19
2. Sélection de matériel et de logiciel 21
2.1 Architecture fonctionnelle du système 22
2.1.1 La périphérie 22
2.1.2 Le réseau de transport et d'interconnexion 22
2.1.3 L'infrastructure Cloud 22
2.2 Spécifications matérielles 23
2.2.1 Automate Programmable Industriel SIMATIC S71200 23
2.2.2 Module d'entrée analogique SM 1231 25
2.2.3 Module de sortie analogique SM 1232 27
2.2.4 PCT50 : 28
2.2.5 FAN ELECTRIC : 29
2.2.6 Fonctions des commutateurs X200 29
2.2.7 Alimentation SITOP PSU100S 30
2.2.8 Relais 24V 220V 31
2.2.9 Définition d'un IHM industriel 32
2.2.10 Disjoncteur bipolaire 32
2.3 Exigences logicielles 34
2.3.1 TIA Portal 34
2.3.2 Ladder Logic 35
2.3.3 Microsoft Visual Studio 36
2.3.4 HTML5 37
2.3.5 CSS3 38
2.3.6 JavaScript 39
2.3.7 NodeRED 39
2.3.8 JSON 40
2.3.9 Service Cloud (Ubidots) 41
3. CHAPITRE 3 :Réservoirs, régulation de niveau et instrumentation associée 44
3.1 Typologie des réservoirs industriels 44
3.1.1 Classification technique : caractéristiques par géométrie et par pression 44
3.1.2 Caractéristiques de pression : Réservoirs ouverts et sous pression 45
3.1.3 Fonctions et domaines d'application : Synthèse des cas d'usage industriels 46

3.2 Stratégies de Régulation de Niveau 48
3.2.1 Régulation Tout-ou-Rien (On/Off) : fonctionnement par seuils de niveau 48
3.2.2 Régulation linéaire (P, PI, PID) : principes mathématiques et correction dynamique 49
3.2.3 Analyse des actions et correction dynamique du niveau 50
3.3 Instrumentation de Terrain pour la Mesure et l'Action 52
3.3.1 Technologies de capteurs de niveau : hydrostatique, ultrason, radar, capacitif 52
3.3.2 Organes de commande d'un fluide : pompes à vitesse variable, vannes motorisées et électrovannes 53
3.4 Spécification du Modèle d'Étude pour la Simulation 54
3.4.1 Description fonctionnelle du procédé à simuler : choix des limites du système de réservoir 55
3.4.2 Identification des variables : PV, CV et perturbations 56
4. Implémentation, Simulation et Validation 59
4.1 TIA Portal (Totally Integrated Automation) 59
4.1.1 Configuration Matérielle : 59
4.1.2 Configuration Réseau IP : 60
4.1.3 Liaisons de Communication : 60
4.1.4 Table des Variables : 61
4.2 Siemens S7-PLCSIM Advanced V4.0 : 62
4.2.1 Réseau 1 (Logique de commande) : 62
4.2.2 Réseau 2 : activation de la pompe de remplissage 63
4.2.3 Réseau 3 : activation de la pompe de vidange 64
4.2.4 Réseau 4 : calcul dynamique du volume d'eau 64
4.2.5 Réseau 5 : mise à l'échelle de la mesure (SCALE) 65
4.2.6 Réseau 6 : supervision des seuils de sécurité (comparateurs) 66
4.2.7 Établissement de la connexion matérielle virtuelle 67
4.3 Environnement de Simulation Virtuelle 67
4.3.1 Basculement vers la simulation virtuelle (S7-PLCSIM Advanced) 67
4.4 Panneau de contrôle S7-PLCSIM Advanced avec l'instance automate active 68
4.4.1 Intégration de l'environnement IoT (Node-RED) 68
4.4.2 Configuration du broker MQTT (EMQX) 69
4.4.3 Interconnexion avec l'automate virtuel (nœud S7) 70
4.4.4 Configuration de la table de correspondance (Mapping) sous Node-RED 71

4.4.5 Mapping (seconde partie) : variables numériques et analogiques 72
4.4.6 Configuration du nœud d'écriture S7 (S7 Out Node) 73
4.4.7 Architecture globale des flux Node-RED.....74
4.5 Node-RED et Node-RED Dashboard :.....75
4.5.1 Supervision locale : HMI sous TIA Portal.....75
4.5.2 Interface opérateur initiale (Dashboard Node-RED).....76
4.5.3 Phase transitoire de remplissage (Dashboard Node-RED).....77
4.5.4 Phase de vidange (Dashboard Node-RED).....78

LISTE DES FIGURE

Figure 2-1: Pyramide de l'automatisation 3
Figure 2-2 : SCADA SYTEME 4
Figure 22-3 : architecture des systèmes IoT 6
Figure 2-4: Capteur de carburant intelligent 7
Figure 2-5 : SIMATIC S7-1500 7
Figure 2-6 : SIMATIC IOT2040 8
Figure 2-7 : AiR PACE d'EtherWAN 8
Figure 2-8 : IoT Protocols 10
Figure 2-9 : MQTT Architecture 11
Figure 2-10: HTTP Architecture 14
Figure 22-11 : HTTP based system 15
Figure 22-12 : Localisateur Uniforme de Ressource (URL) 16
Figure 22-13 : Communication OPC UA 18
Figure 22-1:Architecture système 21
Figure 2-2:SIMATIC S7-1200 23
Figure 22-3:analogique SM 1231 26
Figure 22-4: fiche technique SM 1231 26
Figure 22-5:analogique SM 1232 27
Figure 22-6: fiche technique SM 1232 27
Figure 22-7: PCT50 29
Figure 2-8: FAN ELECTRIC 29
Figure 22-9:commutateurs X200 30
Figure 22-10:Alimentation SITOP PSU100S 31
Figure 2-11: Relais 24V 220V 31
Figure 22-12: IHM industriel 32
Figure 22-13Disjoncteur Bipolaire 33
Figure 22-14: MODEM IDOOM 4G ALGERIE 33
Figure 22-15: CABLE RJ45 34
Figure 22-16:Vue Projet de TIA Portal 35
Figure 22-17:Laddar logic 36
Figure 22-18: MICROSOFT VISUAL STUDIO 37
Figure 22-19: HTML CODE 38
Figure 22-20: CSS CODE 39
Figure 22-21: NOTERED 40
Figure 22-22: MESSAGE DE JSON 41
Figure 22-23: Page d'accueil de la plateforme Ubidots 42
Figure 4-1: Configuration matérielle de l'automate CPU 1515-2 PN sous TIA Portal 59
Figure 4-2:Vue réseau de l'interconnexion entre l'automate et l'IHM sous TIA Portal 60
Figure 4-3: Vue de réseau et liaison de communication PN/IE_1 sous TIA Portal 61
Figure 4-4:Table des variables globales de l'automate sous TIA Portal 62

Figure 4-5:Réseau 1 — Logique de commande Marche/Arrêt général en mode connecté (Ligne) 63
Figure 4-6:Réseau 2 — Logique d'activation de la pompe de remplissage (Pompe 1) en ligne 63
Figure 4-7:Réseau 3 — Logique d'activation de la pompe de vidange (Pompe 2) en ligne.....64
Figure 4-8:Réseau 4 — Blocs d'incrémentation (ADD) et de décrémentation (SUB) de simulation en ligne 65
Figure 4-9:Réseau 5 — Traitement de mise à l'échelle via le bloc SCALE en ligne.....66
Figure 4-10:Réseau 6 — Configuration des comparateurs de seuils de niveau en ligne 66
Figure 4-11:Panneau de contrôle S7-PLCSIM Advanced avec l'instance automate active.....68
Figure 4-12:Initialisation des services Node-RED et du module Dashboard dans le terminal système 69
Figure 4-13:Fenêtre de configuration du nœud serveur MQTT (Broker EMQX) sous Node-RED 70
Figure 4-14:Fenêtre de paramétrage de la liaison S7 et définition du point d'accès (Endpoint) sous Node-RED.....71
Figure 4-15:Table de mapping S7 (Partie 1) — Importation des variables de commande initiales.....72
Figure 4-16:Table de mapping S7 (Partie 2) — Importation du reste des variables numériques et de données.....73
Figure 4-17:Fenêtre de configuration du nœud d'écriture S7 pour l'envoi des commandes.....74
Figure 4-18:Architecture logique globale du Flux 3 interconnecté sous Node-RED 75
Figure 4-19:Vue de conduite et écran de supervision natif sur le pupitre HMI TP700 Comfort 76
Figure 4-20:Interface opérateur initiale — Dashboard Node-RED au repos 77
Figure 4-21:Courbe dynamique du niveau d'eau atteignant le seuil haut lors de la phase de remplissage 78
Figure 4-22:Courbe dynamique du niveau d'eau et inversion de cycle à l'atteinte du seuil bas (Phase de vidange).....79

LISTE DES TABLEAU

Table 21:Fiche technique SIMATIC S7120 25

Abréviations :

IIoT – Internet Industriel des Objets

IoT – Internet des Objets

IoS – Internet des Services

CPS – Systèmes Cyber-Physiques

SCADA – Système de Supervision et d'Acquisition de Données

IACS – Systèmes d'Automatisation et de Contrôle Industriels

ICS – Systèmes de Contrôle Industriels

IT – Technologie de l'Information

OT – Technologie Opérationnelle

MQTT – Message Queuing Telemetry Transport

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

OPC UA – OPC Unified Architecture

QoS – Qualité de Service

TCP/IP – Transmission Control Protocol/Internet Protocol

UDP – User Datagram Protocol

TLS – Transport Layer Security

PROFINET – Process Field Network

PROFIBUS – Process Field Bus

GPRS – General Packet Radio Service

RS485 – Recommended Standard 485

RS232 – Recommended Standard 232

API – Automate Programmable Industriel

PLC – Programmable Logic Controller

VFD – Variateur de Fréquence (Variable Frequency Drive)

PID – Proportionnel-Intégral-Dérivé

PV – Variable de Procédé (Process Variable)

CV – Variable de Commande (Control Variable)

MV – Variable Manipulée (Manipulated Variable)

SP – Consigne (Set Point)

DV – Variable de Perturbation (Disturbance Variable)

IaaS – Infrastructure as a Service

PaaS – Platform as a Service

SaaS – Software as a Service

URL – Localisateur Uniforme de Ressource

HTML – HyperText Markup Language

HTML5 – HyperText Markup Language version 5

CSS – Cascading Style Sheets

CSS3 – Cascading Style Sheets version 3

JSON – JavaScript Object Notation

XML – Extensible Markup Language

API – Interface de Programmation Applicative (Application Programming Interface)

REST – Representational State Transfer

IDE – Environnement de Développement Intégré (Integrated Development Environment)

HMI – Interface Homme-Machine (Human-Machine Interface)

IHM – Interface Homme-Machine

CPU – Unité Centrale de Traitement (Central Processing Unit)

V – Volt

DC – Courant Continu (Direct Current)

AC – Courant Alternatif (Alternating Current)

LED – Diode Électroluminescente (Light Emitting Diode)

PCT – Pressure Control and Level Teaching System (PCT50)

NF – Normalement Fermé (Normally Closed)

NO – Normalement Ouvert (Normally Open)

M2M – Machine to Machine

WWW – World Wide Web

W3C – World Wide Web Consortium

WHATWG – Web Hypertext Application Technology Working Group

OASIS – Organization for the Advancement of Structured Information Standards

ISO – Organisation Internationale de Normalisation (International Organization for Standardization)

IEC – Commission Électrotechnique Internationale (International Electrotechnical Commission)

IEEE – Institute of Electrical and Electronics Engineers

CERN – Conseil Européen pour la Recherche Nucléaire

IBM – International Business Machines

AWS – Amazon Web Services

SDN – Software Defined Networking

ISV – Éditeur de Logiciel Indépendant (Independent Software Vendor)

VFD – Variateur de Fréquence

VDE – Association pour les Technologies Électriques, Électroniques et de l'Information

EN – Norme Européenne (Europäische Norm)

NF – Norme Française

LED – Diode Électroluminescente

PCB – Carte à Circuit Imprimé (Printed Circuit Board)

E/S – Entrées/Sorties

IO – Input/Output

SM – Signal Module (Module de signaux)

CM – Communication Module (Module de communication)

SB – Signal Board (Carte de signaux)

CB – Communication Board (Carte de communication)

FB – Bloc Fonctionnel (Function Block)

FC – Fonction (Function)

HSC – Compteur Haute Vitesse (High-Speed Counter)

PWM – Modulation de Largeur d'Impulsion (Pulse Width Modulation)

PID – Proportionnel-Intégral-Dérivé

VFD – Variateur de Fréquence

CIP – Nettoyage En Place (Clean In Place)

NEP – Norme d'Hygiène

VHDL – Very High Speed Integrated Circuit Hardware Description Language

JTAG – Joint Test Action Group

PCB – Printed Circuit Board

RTOS – Système d'Exploitation Temps Réel (Real-Time Operating System)

INTRODUCTION GENERALE

Introduction :

Avec la révolution numérique portée par l'Industrie 4.0, les installations industrielles deviennent de plus en plus connectées et génèrent une quantité croissante de données exploitables pour anticiper les pannes plutôt que de les subir après coup. Or, les méthodes classiques de détection des défauts, qui interviennent généralement une fois le problème déjà survenu, engendrent des coûts de maintenance importants et des arrêts de production non négligeables. C'est pour répondre à cette problématique que ce travail propose la mise en place d'un système de diagnostic intelligent et accessible à distance, testé sur le cas concret d'une station de pompage d'eau automatisée. La solution développée associe la logique de commande programmée sous TIA Portal à une supervision assurée par Node-RED, dans le but de renforcer la fiabilité et la continuité de fonctionnement de l'installation.

CHAPITRE 1 :

Fondements théoriques

1.1 Introduction

Les technologies d'automatisation et de contrôle industriels, aussi connues sous le nom de technologies opérationnelles, constituent la colonne vertébrale de l'industrie moderne. La théorie du concept appelé Internet des Objets, qui est apparue en 1999, s'est progressivement étendue de l'espace résidentiel au domaine industriel [1] sous des appellations comme les systèmes cyberphysiques, les réseaux SCADA, Internet industriel et autres. Cette partie du texte est destinée à définir l'Industrie 4.0 et l'Internet industriel des objets.

1.2 Industrie 4.0



Le monde de l'électronique, quant à lui, connaît l'ère de l'Industrie 4.0, qui est un développement lancé par l'Allemagne en 2011 [2]. Il concerne les systèmes cyberphysiques, qui communiquent de manière indépendante pour une décision de type décentralisé et une efficacité optimale dans la productivité et la sécurité. Cela revient à intégrer le concept de Internet Industriel américain [3], qui révolutionne totalement les façons de produire et d'agir sur l'économie.

L'Internet Industriel peut se définir par la synergie entre l'objet, la machine, l'informatique et l'humain [3]. L'interconnectivité est alors rendue possible afin de gagner en efficacité dans les opérations industrielles, grâce au

De plus, l'idée d' « Industrie 4.0 » se trouve être synonyme direct avec l'IIoT [2]. On peut donner l'exemple de son utilisation dans l'industrie de ce que constitue l'IoT comme une collection de technologies et des concepts organisationnels pour la chaîne de valeurs. Dans les « Smart Factories » (usines intelligentes) sont utilisés les systèmes cybérphysiques (CPS), qui surveillent les phénomènes physiques et font une copie virtuelle du monde physique [4]. Ces systèmes commencent à se communiquer immédiatement avec le moyen de l'IoT et c'est une bonne plateforme pour la prise de décision décentralisée et la collaboration efficace entre les machines et les travailleurs. Enfin, par l'IoS, on peut utiliser les services internes et interorganisationnels qui sont disponibles pour tous les participants à la chaîne de valeurs. [5]

1.3 Systèmes CyberPhysiques (CPS)

Les systèmes cyberphysiques sont définis par la présence de composants physiques et logiciels, toujours connectés les uns aux autres [6]. Ces systèmes, qu'ils soient centralisés ou distribués, ont des capacités de détection, de régulation, de calculs et de communication [6]. L'enjeu de ces systèmes est de pouvoir agir dans le monde réel grâce à des phénomènes physiques programmés grâce à un programme numérique.

L'Internet industriel est défini par une synergie entre les objets, les machines, les systèmes informatiques et les humains. C'est grâce à cette synergie que les performances industriels peuvent être optimisées par l'analyse de leurs données.

En outre, le terme d'« Industrie 4.0 » est parfois utilisé pour faire référence à l'IIoT. En effet, on pourrait le décrire comme une combinaison de technologies et de principes organisationnels liés à la chaîne de valeur. À l'intérieur des « Smart Factories » (usines intelligentes), les cyberphysique systems (CPS) servent à surveiller les processus physiques en modélisant le monde réel par un clone virtuel. En utilisant Internet des Objets (IoT), ils peuvent s'entretenir de façon simultanée, ce qui facilite le processus de prise de décisions décentralisées et de

collaboration efficace entre les machines et l'homme. Par ailleurs, grâce au concept d'Internet des Services (IoS), il devient possible de fournir des services interne et interorganisationnels à tous les partenaires de la chaîne de valeur.

1.4 Systèmes d'automatisation et de contrôle industriels (IACS)

L'expression « IACS ou ICS » se rapporte à l'ensemble des dispositifs de contrôle ainsi que de l'instrumentation qui y est liée. Ces termes font allusion aux éléments matériels, aux programmes informatiques, aux systèmes de réseau ainsi qu'aux boucles de régulation nécessaires au fonctionnement de processus industriels.

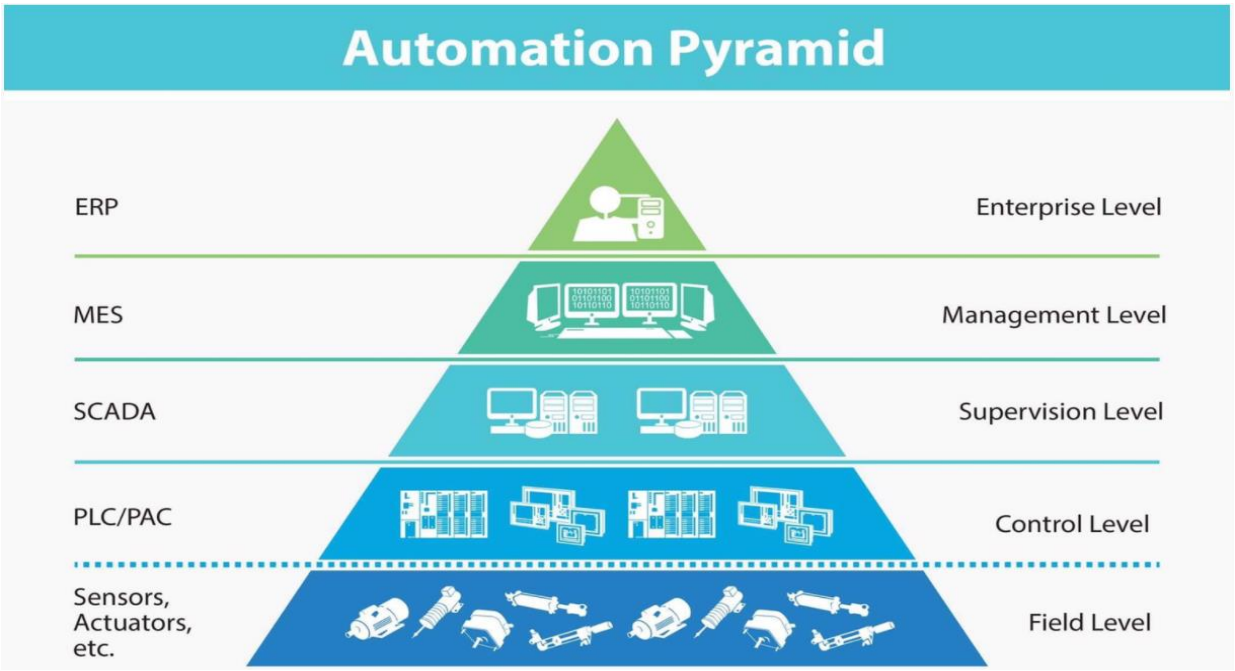


Figure 2-1: Pyramide de l'automatisation

1.5 Systèmes de supervision et d'acquisition de données (SCADA)

La mise en place d'un système SCADA permet donc à un agent de piloter de manière à distance et depuis un poste de commande centralisé des installations qui sont situées audelà de l'endroit où se trouve le système (par exemple, un réseau de conduites, un gisement de gaz ou une centrale hydroélectrique). On peut également utiliser cet outil pour régler les paramètres des contrôleurs, faire fonctionner des vannes et disjoncteurs, traiter les signalements et centraliser les données de terrain.

Structurellement, le concept SCADA comprend deux grands modules :

1.5.1 **Le procédé physique:** L'installation industrielle à superviser (usine de production, réseau de distribution d'eau, feux de signalisation, etc.).

1.5.2 **L'infrastructure intelligente:**

Une combinaison de matériels qui s'interconnectent avec le processus au moyen de senseurs et d'actionneurs pour recueillir des informations et effectuer des tâches commandées. Les améliorations technologiques dans les systèmes SCADA posent plusieurs questions concernant la relation qu'ils entretiennent avec l'IoT industriel. Certains experts incluent des enquêtes criminologiques sur les architectures SCADA dans la perspective des écosystèmes IIoT, tandis que d'autres les considèrent simplement comme des technologies antérieures. Cela est dû au fait que même si les derniers systèmes SCADA sont équipés de connectivité Internet, ils ne peuvent pas en général atteindre le degré de complexité que représente l'IIoT.

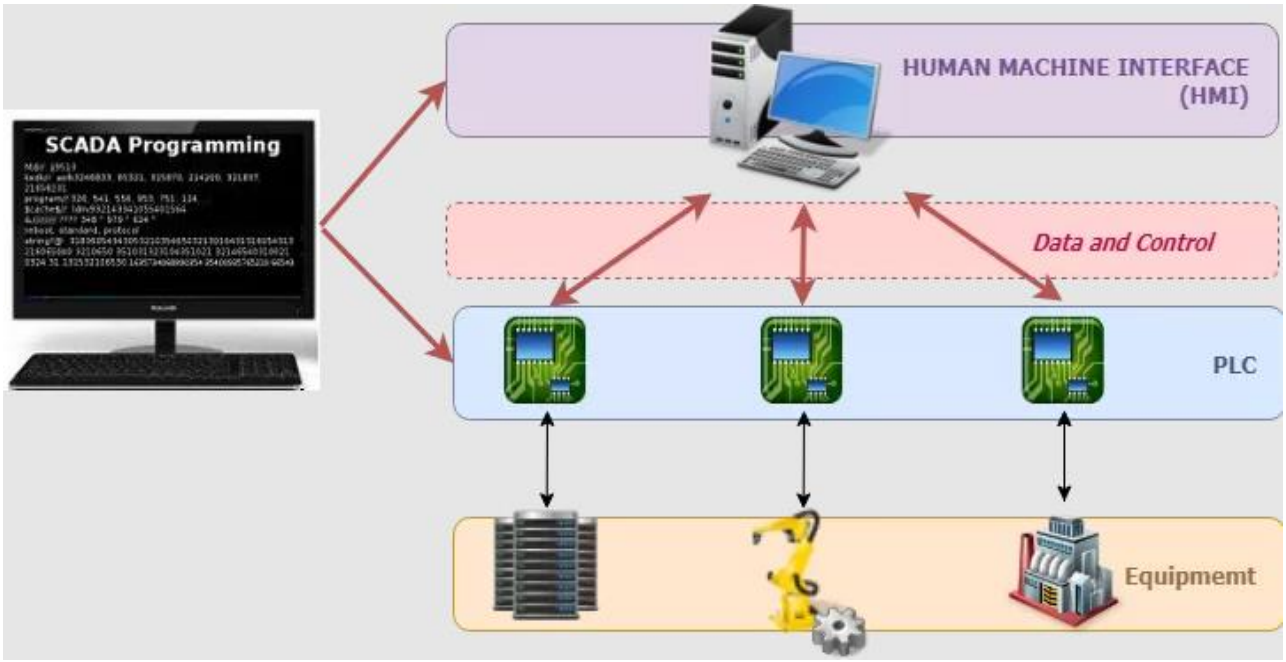


Figure 2-2 : SCADA SYTEME

Le concept de l'Internet Industriel est défini par deux concepts fondamentaux qui le soustendent. Premièrement, l'intégration des capteurs et des actionneurs des

machines au système informatique local ainsi qu'à l'Internet. Deuxièmement, l'établissement d'une connexion constante aux autres réseaux industriels, en mesure de créer de la valeur indépendamment. L'Internet Grand Public et le concept de l'Internet Industriel diffèrent d'abord en ce qui concerne la nature et le montant de création de la valeur par ce dernier, car Internet Social génère principalement ses recettes grâce à la publicité.

Cet approche offre l'opportunité d'efficacité et d'innovation au moyen de la convergence de plusieurs technologies. La synergie créée permet l'interaction directe des domaines physique et numérique. Par conséquent, les entreprises s'assurent du recueil des données par des capteurs, par des solutions informatiques et M2M pour ensuite analyser les flux de données produits par les dispositifs. Le résultat ainsi obtenu est mis en œuvre pour piloter les processus de production, mais aussi pour créer de nouveaux produits aux avantages considérables.

Bien que l'Internet classique et son équivalent industriel jouent tous deux un rôle similaire – offrir un grand réseau, l'Internet industriel se caractérise néanmoins par un but particulier. Selon les derniers écrits techniques, la capacité d'analyse de ce dernier devient de plus en plus évidente, prouvant ainsi que Big Data Analytics est la base de l'Internet industriel. Cela souligne l'une des principales similitudes entre l'Internet industriel et Industrie 4.0 dans leur intérêt commun pour les données.

1.6 Definition de l'internet des objets Industriel (IIoT)

L'IIoT est une approche d'écosystème qui implique des dispositifs intelligents de connexion, de cyberactifs physiques, d'informations générales et, si nécessaire, de Cloud Computing ou d'Edge Computing. L'architecture permet ainsi de disposer d'un accès, d'une collecte et d'une analyse de données intelligentes et autonomes, traitées en temps réel pour les procédés, produits ou services industriels.

Basée essentiellement sur la mise à profit de Big Data, l'approche IIoT consiste à valoriser la totalité de la chaîne de la production. Les bénéfices directs sont la performance dans la commercialisation des produits, l'efficacité de la production, l'économie du coût humain, la diminution des coûts énergétiques et des délais de production surmesure (buildtoorder).

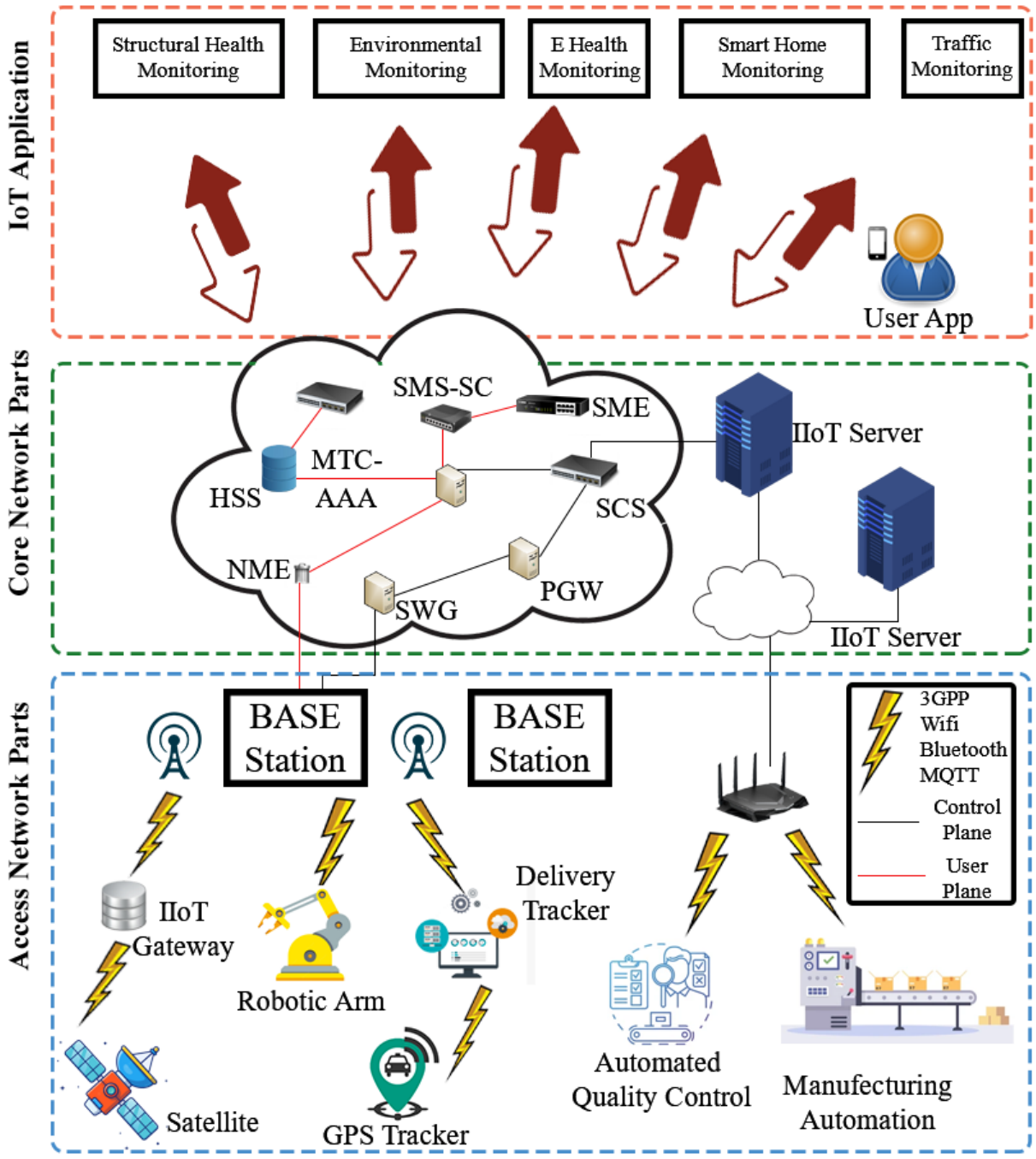


Figure 22-3 : architecture des systèmes IoT

1.7 Équipements intelligents (Smart Devices)

Au niveau opérationnel initial se trouvent le contrôleur intelligent, les capteurs et les actionneurs. Ce sont des éléments de liaison qui prennent en charge la supervision ou la régulation directe des processus physiques. Plus exactement, les capteurs mesurent et font connaître la modification des paramètres extérieurs ou industriels. Parmi ces paramètres figurent plusieurs phénomènes physiques, notamment la température, l'humidité, la composition chimique, le niveau des liquides dans les citernes, les débits au sein des tuyauteries, entre autres l'allure de l'assemblage.



Figure 2-5 : SIMATIC S7-1500



Figure 2-4: Capteur de carburant intelligent

1.8 Passerelles IoT (IoT Gateways)

IoT Industrial Gateways constituent le pont qui assure la liaison radio entre les systèmes installés et les infrastructures intelligentes modernes. Adaptées aux conditions de fonctionnement rigoureuses d'un environnement industriel et dotées de résistance mécanique, ces passerelles sont d'une grande utilité dans l'installation de nodes sensors et de modules E/S utilisés pour la réception des données nécessaires. La flexibilité du logiciel de ces appareils permet la mise en œuvre de divers protocoles de communication, assurant leur interopérabilité totale avec les systèmes de terrain.

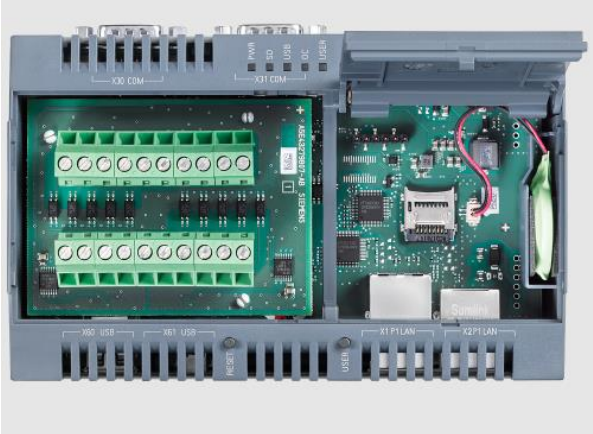


Figure 2-6 : SIMATIC IOT2040



Figure 2-7 : AiR PACE d'EtherWAN

1.9 Plateformes Cloud

1.9.1Siemens MindSphere

MindSphere, which is a product of Siemens – a major German corporation, refers to a cloud computing platform of the Platform as a Service (PaaS) class and designed particularly for the industrial sector. The system allows the consolidation and immediate processing of data streams provided by field sensors.

The PaaS paradigm implies the delegation of hardware and software resources to a third party through the Web. In regard to developers, this technique frees one from restrictions related to server installation and administration issues. Therefore, efforts can be devoted to application development alone.

1.9.2IBM Private Cloud for Manufacturing

La solution private cloud fournie par IBM, optimisée pour le monde de la production industrielle, est conçue pour répondre aux besoins essentiels de la chaîne de montage. Cette dernière offre des avantages comme la convergence IT/OT, la sécurité accrue, les analyses aux abords des appareils (edge analytics), la gestion d'appareils connectés, la programmation du routage (Software Defined Networking SDN), et l'optimisation des applications. Le cloud private de la part

d'IBM s'appuie essentiellement sur la plateforme de conteneurs Red Hat OpenShift et la gamme d'outils IBM Cloud Pak.

Par opposition aux infrastructures de type public, un cloud privé ou d'entreprise est un environnement de calcul informatique où les ressources matérielles et logicielles ne sont utilisées que pour un client unique. La combinaison de caractéristiques telles qu'élasticité, flexibilité et vitesse de déploiement du cloud computing avec un accès et une sécurité accrues et une personnalisation de l'infrastructure physique sur place sont spécifiques aux clouds privés.

1.9.3Amazon Web Services (AWS)

La technologie AWS se caractérise comme un cloud global évolué fourni par Amazon [7]. Il possède un écosystème intégré qui comprend des services IaaS, PaaS et SaaS.

1.9.3.1 *L'IaaS (Infrastructure as a Service) :*

Ce modèle fournit à la demande des ressources fondamentales de calcul, de stockage et de mise en réseau, généralement facturées à l'usage (payasyougo).

1.9.3.2 *Le SaaS (Software as a Service) :*

Il s'agit d'un mode de distribution où les applications sont hébergées dans les serveurs de cloud computing du fournisseur et sont directement disponibles pour les utilisateurs en passant par Internet. Dans ce cas, l'éditeur de logiciel indépendant (ISV) délègue son application au niveau de l'hébergement chez un autre fournisseur cloud computing.

1.9.4Protocoles de données

Les protocoles de communication IoT assurent l'échange de données entre les différents terminaux connectés. Parmi les standards existants, ce projet exploite principalement les protocoles **MQTT** et **HTTP**.

1.10Le protocole MQTT

1.10.1Contexte historique

Le protocole MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie inventé en 1999 par les ingénieurs Andy StanfordClark et Arlen Nipper. Les exigences principales à son époque étaient l'interaction des infrastructures des oléoducs et gazoducs avec des architectures SCADA. En effet, les infrastructures de cette époque subissaient les inconvénients de coexister avec des protocoles propriétaires et hétérogènes; ils avaient donc des problèmes d'interopérabilité. Avec l'introduction des fonctionnalités MQTT, il a été possible de résoudre ces problèmes d'intercommunication. En sus de l'interopérabilité, le protocole a été conçu afin de répondre aux contraintes suivantes: léger, optimal en bande passante, data agnostic, simple à mettre en œuvre et gestion des QoS.

1.10.2Caractéristiques du protocole MQTT

L'architecture du protocole MQTT repose sur un schéma de messagerie « Publish/Subscribe » qui permet une communication efficace entre différents dispositifs hétérogènes. Cependant, utilisant principalement le mode de communication TCP/IP, c'est un protocole qui convient parfaitement dans des connexions lointaines et soumises à des limitations matérielles notables. En outre, le MQTT étant un protocole ouvert réglementé par l'organisation OASIS, c'est un protocole recommandé au niveau ISO (ISO/IEC 20922). [8]

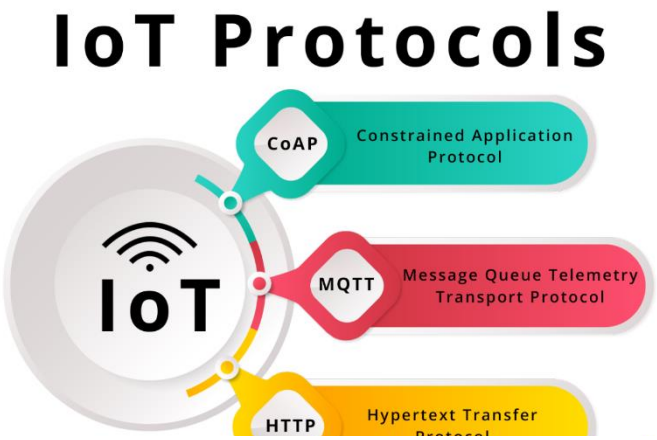


Figure 2-8 : IoT Protocols

1.10.3 Architecture MQTT

Dans un réseau architecture MQTT, les connecteurs sont identifiés sous le terme «clients» et échangent les informations au travers d'un serveur appelé broker. Ce broker gère et coordonne toutes les communications de données entre tous les clients. Lorsqu'un client, opérant au niveau de diffuseur ou publisher, envoie une information, celleci est envoyée au niveau d'une adresse logique appelée «topic» (sujet). Les topics constituent ainsi le mécanisme de l'adressage MQTT, permettant de hiérarchiser les données de façon plus organisée et optimale.

Le point fort principal de ce modèle, également appelé «Pub/Sub», consiste dans la totalité de la dissociation entre les entités. Ainsi, le diffuseur ne sait pas où sont les abonnés et combien d'entre eux sont là. De même, chaque client peut jouer les rôles de diffuseur et/ou d'abonné selon son choix. Il faut noter qu'aucun terminal ne communique directement avec un autre terminal; tout se fait au niveau du broker.

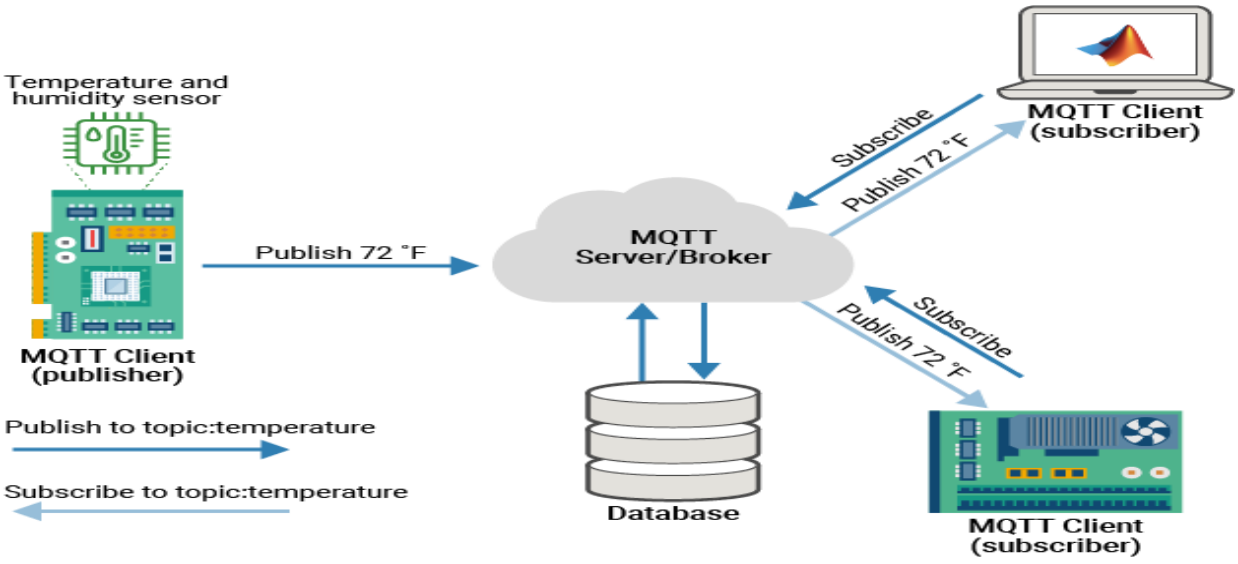


Figure 2-9 : MQTT Architecture

1.10.4 Les Topics MQTT

La distribution des messages envoyés par le diffuseur jusqu'aux utilisateurs s'appuie sur la structure arborescente des topics. Les topics sont structurés en chaîne de caractères en UTF8, dont les différents niveaux sont séparés par une

barre oblique (/). Tous les messages envoyés doivent nécessairement être classés dans un topic précis. Pour cette raison, tous les messages ne sont distribués que si les appareils intéressés ont souscrit au topic concerné.

1.11 Le protocole HTTP

1.11.1 Contexte historique

La technologie HTTP (HyperText Transfer Protocol) constitue le cadre de base du World Wide Web. Ce protocole a été créé originellement par Tim BernersLee et son groupe de travail entre 1989 et 1991. Au fil des années, il s'est avéré nécessaire de procéder à de nombreux développements importants au niveau de cette technologie, permettant ainsi de lui apporter une forte polyvalence et d'améliorer considérablement ses performances.

1.11.2 L'émergence du World Wide Web (WWW)

L'invention du World Wide Web au CERN par Sir Tim BernersLee a été un moment très important dans l'histoire de l'échange d'informations à travers le monde. À l'origine, le World Wide Web a été créé pour aider les scientifiques et les universités du monde entier à partager des données de manière automatisée. Le World Wide Web a donc joué un rôle clé dans la création des architectures réseaux modernes que nous utilisons aujourd'hui.

Cette innovation reposait sur la combinaison de trois technologies fondamentales encore utilisées aujourd'hui :

Le système d'adressage unique URL, qui sert à identifier les ressources.

Le langage de description de documents HTML, qui est utilisé pour créer des documents.

Le protocole de transfert HTTP, qui permet de relier ces documents hypertextes entre eux, et qui a ouvert la voie à l'Internet grand public, puis, par la suite, aux protocoles de communication de l'Internet des Objets.

1.11.3 Définition du protocole HTTP

Le protocole HTTP, ou Hypertext Transfer Protocol, est très important pour le World Wide Web. Il permet de charger des pages web et des ressources en utilisant des liens hypertextes. Le protocole HTTP fonctionne à un niveau spécifique dans le modèle OSI, appelé la couche application [9]. Il a été créé pour gérer le transfert de données entre des équipements qui sont connectés les uns aux autres. Pour cela, il utilise les couches inférieures de la pile réseau, comme la couche transport TCP.

Le fonctionnement de base du protocole HTTP est basé sur un modèle très simple : un client envoie une requête à un serveur, et le serveur répond avec les données demandées ou avec un message qui indique le résultat de l'opération. C'est un peu comme une conversation entre deux machines. La machine cliente envoie une demande au serveur cible, et le serveur traite la demande avant de renvoyer une réponse au client. C'est ainsi que le protocole HTTP permet de transférer des données entre des équipements différents sur le réseau.

1.11.4 Architecture du protocole HTTP

Le protocole HTTP repose sur une architecture de type client et serveur. Cela signifie que le protocole HTTP fonctionne par cycles de requêtes et de réponses.

Dans ce système, chaque entité a un rôle précis. Les clients HTTP sont ceux qui commencent la conversation. Ils peuvent être des navigateurs web, des scripts automatisés ou des agents d'indexation des moteurs de recherche

Le serveur HTTP, quant à lui, est le logiciel qui écoute sur un port spécifique, gère les requêtes et renvoie les ressources demandées.

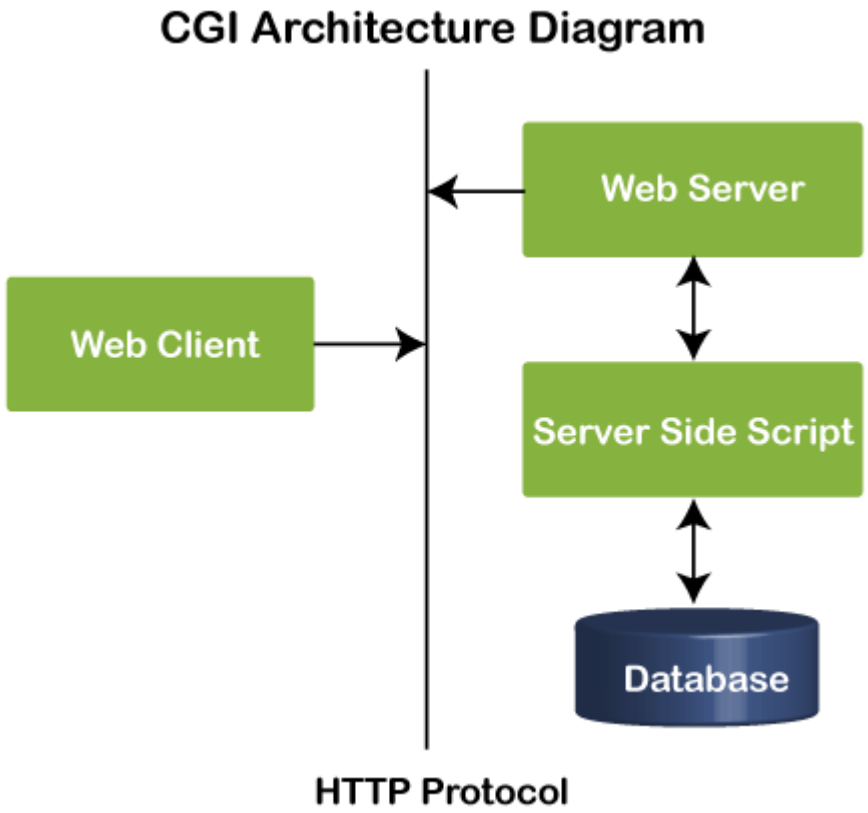


Figure 2-10: HTTP Architecture

1.11.5 Composants d'une architecture HTTP

L'écosystème basé sur le protocole HTTP repose sur une relation clientserveur classique. Dans cette architecture, les informations circulent entre les clients et les serveurs via des entités bien définies :

Les clients : ce sont les applications ou les navigateurs web qui envoient des requêtes au serveur pour accéder à des ressources.

1.11.5.1 Les serveurs : ils reçoivent les requêtes des clients, traitent les demandes et retournent les ressources demandées.

1.11.5.2 Les proxys : ils agissent comme intermédiaires entre les clients et les serveurs. Ils peuvent effectuer des tâches telles que la mise en cache, la filtration des requêtes ou la traduction des protocoles.

1.11.5.3 Les passerelles : ce sont des serveurs spécialisés qui relient différents réseaux ou protocoles. Ils permettent de communiquer entre les clients et les serveurs qui utilisent des protocoles différents.

Ces entités travaillent ensemble pour fournir une communication efficace et sécurisée sur Internet.

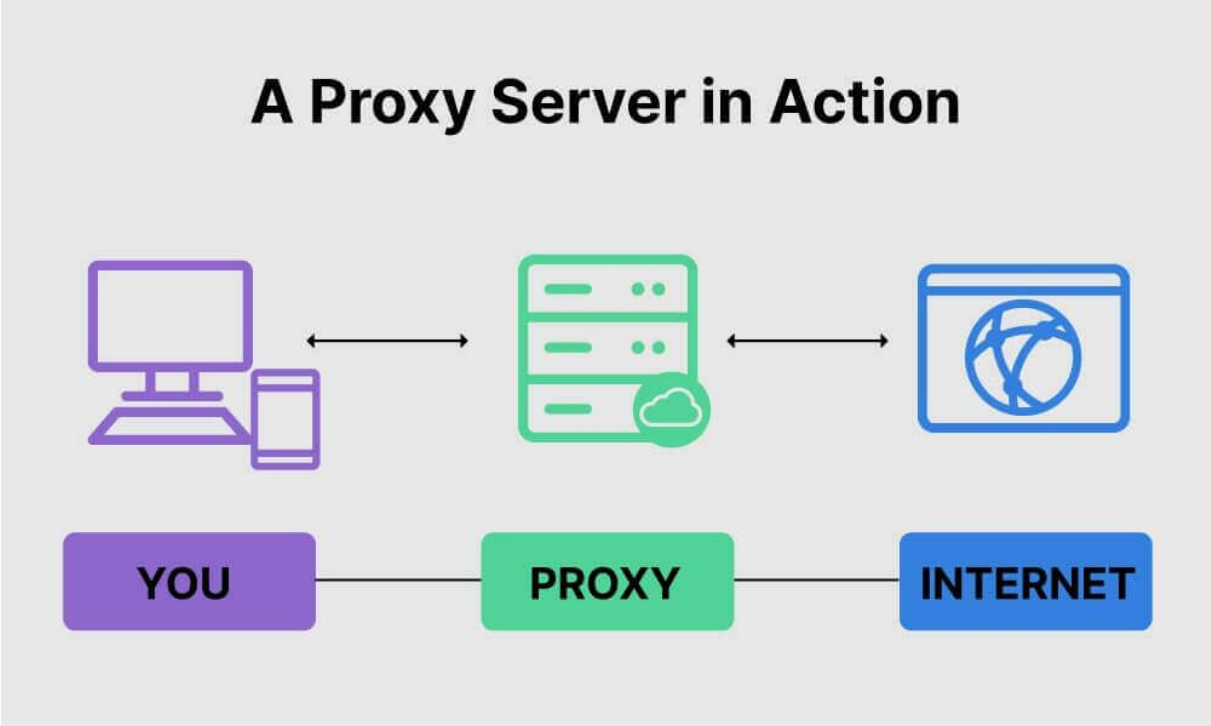


Figure 22-11 : HTTP based system

1.11.6 Localisateur uniforme de ressource (URL)

Pour accéder à une ressource spécifique à travers le réseau Internet, un client doit cibler son adresse exacte. Le protocole HTTP s'appuie sur le concept de Localisateur uniforme de ressource, un standard d'adressage universel qui

permet de localiser et de spécifier n'importe quel type d'information ou de document en ligne.

La syntaxe d'un Localisateur uniforme de ressource s'articule autour de quatre segments fondamentaux :

1.11.6.1 ***Le protocole:***

il indique le mécanisme de transfert utilisé pour l'échange, comme par exemple http ou https.

1.11.6.2 ***L'hôte:***

il identifie la machine ou le serveur cible qui héberge la ressource, via un nom de domaine ou une adresse IP.

1.11.6.3 ***Le port:***

il spécifie le canal de communication logique ouvert sur le serveur pour ce service, par défaut le port 80 pour HTTP.

1.11.6.4 ***Le chemin:***

il détermine l'emplacement exact et le nom du fichier ou de la ressource au sein de l'arborescence du serveur.



Figure 22-12 : Localisateur Uniforme de Ressource (URL)

1.11.7 Le protocole HTTP dans le contexte de l'IoT

Le protocole HTTP est la base de la communication de données sur Internet. Il est encore très utilisé aujourd'hui pour les appareils qui doivent transmettre de grandes quantités de données ou interagir avec des sites web standard. Cependant, utiliser HTTP peut être limité dans certains cas. Il consomme beaucoup d'énergie en raison de sa complexité, ce qui le rend moins adapté pour les appareils fonctionnant avec des piles ou qui nécessitent une gestion efficace des ressources réseau.

Malgré ces limites, HTTP est très utile pour certaines applications spécifiques, comme l'impression 3D. Dans ce cas, il permet de relier les ordinateurs et les imprimantes 3D sur un réseau local ou sur Internet. Cela facilite l'échange de fichiers de conception importants et la commande à distance des équipements pour fabriquer des objets et des prototypes en trois dimensions.

1.11.8 Le protocole OPC UA (OPC Unified Architecture)

Le protocole OPC UA, ou OPC Unified Architecture, est très important dans l'industrie. Il se démarque des anciens protocoles de communication parce qu'il permet aux équipements d'envoyer directement leurs données aux plateformes Cloud, sans avoir besoin de passerelles ou de serveurs intermédiaires.

Le protocole OPC UA est conçu pour fonctionner avec n'importe quel matériel ou logiciel, peu importe le fabricant. Cela signifie que les différents équipements peuvent communiquer entre eux de manière native, ce qui facilite leur intégration dans un même réseau.

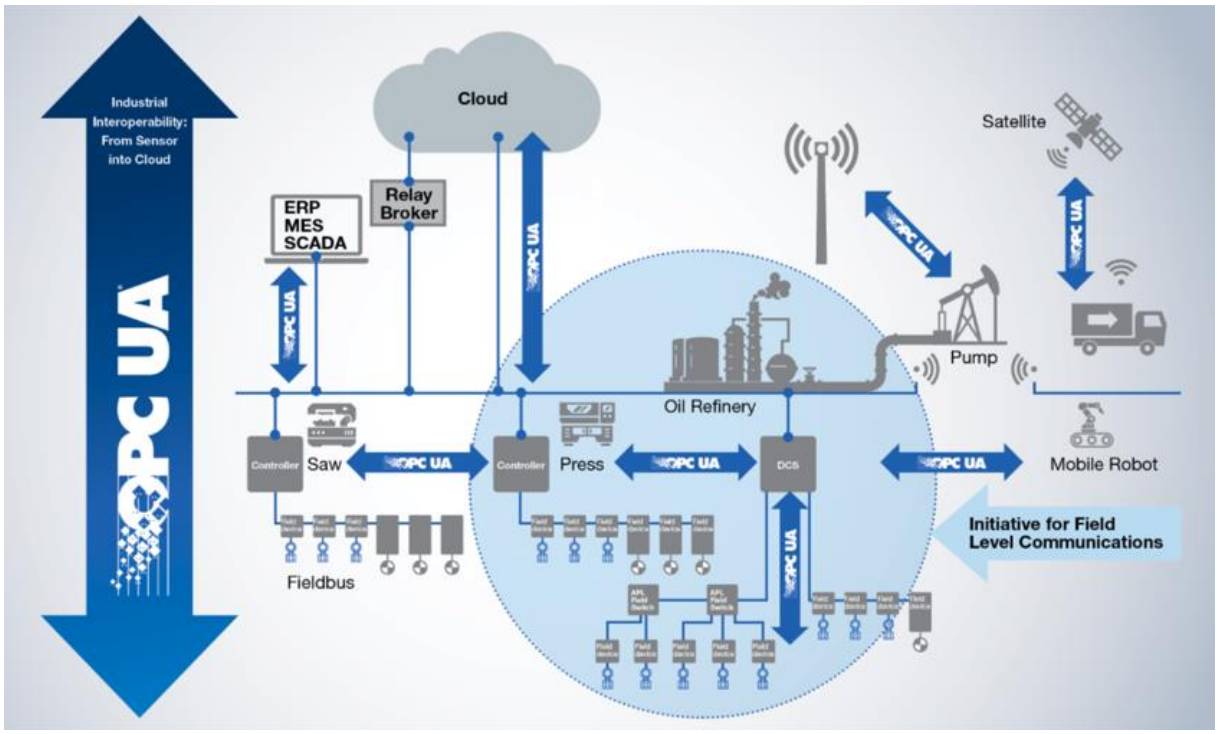


Figure 22-13 : Communication OPC UA

Comme le montre la figure 10, ce protocole prend en charge les connexions directes entre les appareils Edge et le Cloud, ainsi que les communications horizontales entre les machines au niveau de l'atelier. En analysant la pyramide d'automatisation, on voit qu'OPC UA peut gérer à la fois le contrôle décentralisé sur le site de production et la connexion verticale avec les infrastructures Cloud.

1.12Protocoles réseau (Couche Transport et Accès)

Les protocoles réseau de l'IoT nous permettent de relier les différents équipements entre eux et de faire transiter les données à travers un réseau de communication. Pour que tous les appareils du projet restent connectés, nous utilisons trois technologies qui se complètent bien :

1.12.1Le WiFi,

selon la norme IEEE 802.11, est utilisé pour les connexions sans fil sur de courtes et moyennes distances [10]. Cela nous permet d'avoir des débits de données élevés pour les capteurs fixes ou qui se déplacent un peu dans l'atelier.

1.12.2 La 4G

ou réseau cellulaire, est utilisée pour avoir une connexion fiable sur de longues distances. Cela nous permet de transmettre des données vers les serveurs Cloud, même si le réseau local n'est pas disponible.

1.12.3 L'Ethernet

selon la norme IEEE 802.3, est notre choix pour les connexions par câble directes [11]. Cela est nécessaire pour les équipements qui exigent une protection maximale contre les interférences électromagnétiques, une faible latence et une grande fiabilité, comme les contrôleurs principaux.

CHAPITRE 2 :

Sélection de matériel et de logiciel

2. Sélection de matériel et de logiciel

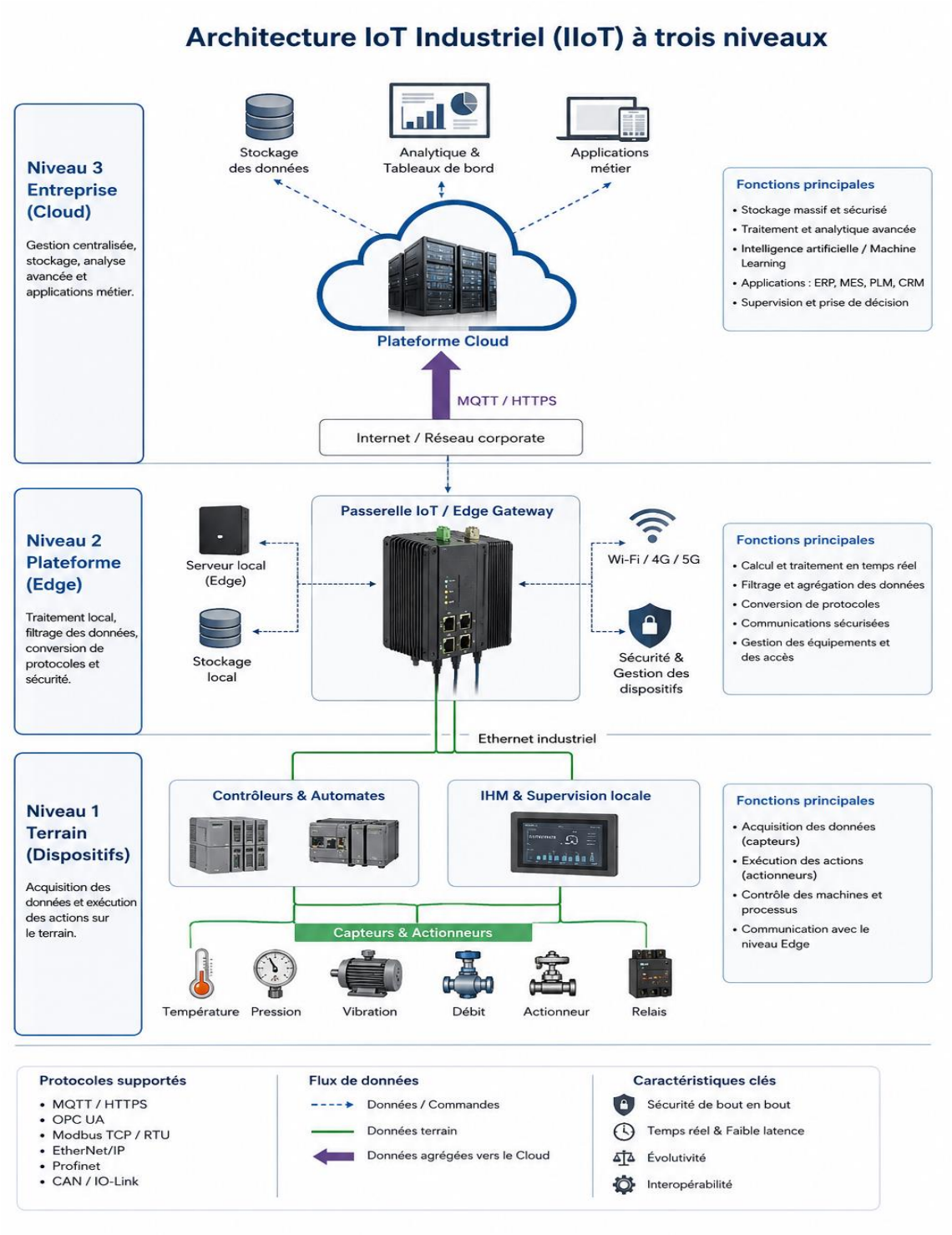


Figure 22-1:Architecture système

2.1 Architecture fonctionnelle du système

2.1.1La périphérie

La périphérie constitue le point de contact direct avec le procédé physique. C'est à ce niveau que le capteur de niveau et les actionneurs génèrent et reçoivent des flux de données en temps réel. L'intégration de l'infrastructure de calcul en périphérie permet de décentraliser le traitement de l'information au plus près de sa source d'origine. Cette approche résout les problématiques de latence critiques intrinsèques aux architectures Cloud exclusives, garantissant une réactivité optimale du système de contrôlecommande.

2.1.2Le réseau de transport et d'interconnexion

La connectivité et le routage des données entre les différentes strates reposent sur trois piliers technologiques :

- **Le réseau cellulaire :** Cette technologie occupe une place centrale pour assurer une connectivité étendue et résiliente, permettant de lier des systèmes distants ou mobiles aux infrastructures centrales.
- **La passerelle :** Cet équipement physique assure l'interface d'interconnexion essentielle. Elle centralise les flux issus des capteurs et des modules de terrain afin de les transmettre de manière sécurisée vers le Cloud.
- **L'environnement :** Spécifiquement conçu pour le déploiement d'applications, cet outil orchestre l'acquisition de données, le traitement local en périphérie et l'intégration Cloud via une approche de programmation visuelle. Cet outil simplifie l'interopérabilité et l'interconnexion entre les équipements traditionnels et les nouvelles technologies connectées.

2.1.3L'infrastructure Cloud

Le Cloud est un ensemble de serveurs distants, de logiciels et de bases de données centralisés dans des centres de données à travers le monde. Ces ressources sont accessibles via Internet. Grâce au Cloud Computing, l'infrastructure est libérée des contraintes de gestion locale. Les utilisateurs et les entreprises peuvent ainsi déployer des applications d'analyse à grande échelle. Ils n'ont pas besoin de

maintenir des serveurs physiques ou des environnements d'exécution sur leurs propres machines.

2.2 Spécifications matérielles

2.2.1Automate Programmable Industriel SIMATIC S71200

La gamme d'automates programmables industriels SIMATIC S71200 de Siemens a été choisie. Elle assure la logique de contrôle commande du procédé automatisé. [12]

Le S71200 est une solution robuste. Il offre une architecture compacte et un coût optimisé. Son jeu d'instructions est performant. Il est idéal pour la régulation d'applications industrielles variées.

Le S71200 est associé à son environnement de développement et de programmation sous Windows, appelé TIA Portal. Cela offre la flexibilité et la précision nécessaires. Il permet de résoudre des problématiques d'automatisation complexes. Il permet aussi de centraliser les variables de terrain avant leur intégration IoT.



Figure 2-2:SIMATIC S7-1200

Le contrôleur S71200 offre à la fois performance et souplesse pour piloter différents types d'équipements dans les systèmes d'automatisation. Grâce à sa structure compacte, à sa configuration adaptable et à ses nombreuses fonctionnalités intégrées, il constitue une solution efficace pour une grande variété d'applications industrielles.

L'unité centrale (CPU) regroupe dans un seul boîtier un microprocesseur, une alimentation intégrée, des circuits d'entrées/sorties, une interface PROFINET intégrée, des entrées/sorties dédiées au contrôle rapide des mouvements ainsi que des entrées analogiques embarquées. Cette intégration permet d'obtenir un automate à la fois puissant et peu encombrant.

Une fois le programme chargé, la CPU assure la surveillance et la commande des équipements du système automatisé. Elle analyse les signaux d'entrée puis adapte les sorties selon la logique définie dans le programme utilisateur. Celuici peut inclure des opérations logiques booléennes, des fonctions de comptage et de temporisation, des calculs mathématiques avancés ainsi que des échanges de données avec d'autres équipements intelligents.

La communication réseau est assurée par un port PROFINET intégré, permettant l'échange d'informations au sein d'un réseau industriel PROFINET [13]. Il est également possible d'ajouter des modules complémentaires pour assurer la communication via d'autres protocoles et réseaux tels que PROFIBUS, GPRS, RS485 ou RS232 [14]

Table 1 General

Technical data	CPU 1215C AC/DC/Relay	CPU 1215C DC/DC/Relay	CPU 1215C DC/DC/DC
Order number	6ES7 215-1BG31-0XB0	6ES7 215-1HG31-0XB0	6ES7 215-1AG31-0XB0
Dimensions W x H x D (mm)	130 x 100 x 75	130 x 100 x 75	130 x 100 x 75
Shipping weight	550 grams	585 grams	520 grams
Power dissipation	14 W	12 W	12 W
Current available (SM and CM bus)	1600 mA max. (5 VDC)	1600 mA max. (5 VDC)	1600 mA max. (5 VDC)
Current available (24 VDC)	400 mA max. (sensor power)	400 mA max. (sensor power)	400 mA max. (sensor power)
Digital input current consumption (24VDC)	4 mA/input used	4 mA/input used	4 mA/input used

Table 2 CPU features

Technical data		Description
User memory[1]	Work	100 Kbytes
	Load	4 Mbytes, internal, expandable up to SD card size
	Retentive	10 Kbytes
On-board digital I/O		14 inputs/10 outputs
On-board analog I/O		2 inputs/2 outputs
Process image size		1024 bytes of inputs (I)/1024 bytes of outputs (Q)
Bit memory (M)		8192 bytes
Temporary (local) memory		• 16 Kbytes for startup and program cycle (including associated FBs and FCs) • 4 Kbytes for standard interrupt events including FBs and FCs • 4 Kbytes for error interrupt events including FBs and FCs
Signal modules expansion		8 SMs max.
SB, CB, BB expansion		1 max.
Communication module expansion		3 CMs max.
High-speed counters		6 total, see table HSC input assignments for CPU 1215C • Single phase: 3 at 100 kHz and 3 at 30 kHz clock rate • Quadrature phase: 3 at 80 kHz and 3 at 20 kHz clock rate
Pulse outputs [2]		4
Pulse catch inputs		14

Table 21:Fiche technique SIMATIC S7120

2.2.2Module d'entrée analogique SM 1231

La CPU ne traite que des informations numériques. Par conséquent, toute grandeur analogique doit être transformée en données numériques avant d'être utilisée par l'automate. Cette transformation est effectuée par un convertisseur analogiquenumérique, également appelé module d'entrée analogique.

Le module d'entrée analogique fournit généralement un mot de 16 bits à la CPU pour les équipements SIMATIC. Le signal analogique est numérisé et sa valeur est représentée par des niveaux discrets sous forme de courbe en escalier.

Les principales caractéristiques du module d'entrée analogique sont la résolution et la vitesse de conversion. Ces caractéristiques ont une influence directe sur la précision et la rapidité du traitement des signaux analogiques.



Figure 22-3:analogique SM 1231

General information	
Product type designation	SM 1231, AI 8x16 bit RTD
Supply voltage	
Rated value (DC)	24 V
Input current	
Current consumption, typ.	40 mA
from backplane bus 5 V DC, typ.	80 mA
Power loss	
Power loss, typ.	1.5 W
Analog inputs	
Number of analog inputs	8; Resistance thermometer
permissible input voltage for voltage input (destruction limit), max.	±35 V
Technical unit for temperature measurement adjustable	Degrees Celsius/degrees Fahrenheit
Input ranges	
• Voltage	No
• Current	No
• Thermocouple	No
• Resistance thermometer	Yes; Resistance-type transmitter: Pt10, Pt50, Pt100, Pt200, Pt500, Pt1000, Ni100, Ni120, Ni200, Ni500, Ni1000, Cu10, Cu50, Cu100, LG-Ni1000
• Resistance	Yes; 150 Ω, 300 Ω, 600 Ω
Input ranges (rated values), resistance thermometer	
• Cu 10	Yes
— Input resistance (Cu 10)	10 Ω
• Ni 100	Yes
— Input resistance (Ni 100)	100 Ω
• Ni 1000	Yes
— Input resistance (Ni 1000)	1 000 Ω
• LG-Ni 1000	Yes
— Input resistance (LG-Ni 1000)	1 000 Ω
• Ni 120	Yes
— Input resistance (Ni 120)	120 Ω
• Ni 200	Yes
— Input resistance (Ni 200)	200 Ω
• Ni 500	Yes
— Input resistance (Ni 500)	500 Ω
• Pt 100	Yes
— Input resistance (Pt 100)	100 Ω

Figure 22-4: fiche technique SM 1231

2.2.3Module de sortie analogique SM 1232

La CPU fonctionne avec des données numériques. Le module de sortie analogique convertit ensuite ces données numériques en signaux analogiques, qu'il s'agisse de courant ou de tension, à l'aide d'un convertisseur numériqueanalogique.

Ces signaux analogiques permettent de commander des actionneurs analogiques dans le système. Ils sont utilisés pour contrôler divers équipements industriels tels que des variateurs de vitesse, des vannes proportionnelles ou des régulateurs.



Figure 22-5:analogique SM 1232

Données générales	
Modèle	Module de sortie analogique
Version	G1
Sorties analogiques	4

Données électriques	
Tension d'alimentation	24 V DC
Plage admissible	20,4...28,8 V DC
Résolution	0-10 V 14 bits, 0-20 mA 12 bits
Protection courts-circuits	non

Données mécaniques	
Dimensions (l x H x P)	45 x 100 x 75 mm
Assemblage	Rail DIN 35
Raccordement	Bornes à vis

Conditions ambiantes	
Température ambiante min.	-20 °C
Température ambiante max.	60 °C
Indice de protection	IP20

Plus d'informations	
Approbations	cULus
	UL
	CE
	UKCA
Les entrées d'alarme	oui

Figure 22-6: fiche technique SM 1232

2.2.4PCT50 :

Le PCT50, c'est un système de régulation du niveau d'eau plutôt simple à prendre en main et très lisible visuellement. Il se compose de deux cuves en acrylique transparent : une cuve de procédé montée audessus d'une cuve de stockage (ou « bâche »). L'eau est pompée vers la cuve de procédé, puis redescend vers la bâche par deux vannes : une à réglage manuel, l'autre commandée par logiciel (tout ou rien). On peut aussi équiper chacune des vannes d'orifices externes de diamètres différents, histoire de modifier leur comportement.

Cette flexibilité permet de monter tout un éventail de scénarios de régulation, y compris deux types de processus fondamentalement distincts :

- **Régulation de niveau par variation proportionnelle de la vitesse de la pompe** (contrôle par débit entrant, ou *inflow control*) ;
- **Régulation de niveau par ouverture à durée proportionnelle d'une vanne électromagnétique** (contrôle par débit sortant, ou *outflow control*).

Quand on utilise le contrôle par débit entrant, on peut générer des perturbations reproductibles au moyen de la vanne électromagnétique et on en fait varier l'amplitude en changeant la taille des orifices.
Avec le contrôle par débit sortant, on peut faire varier la vitesse de pompe par paliers, ce qui donne accès à toute une gamme de perturbations reproductibles.

Ces dispositifs permettent une comparaison directe entre différents réglages de correcteur.



Figure 22-7: PCT50

2.2.5FAN ELECTRIC :



Figure 2-8: FAN ELECTRIC

2.2.6Fonctions des commutateurs X200

Les commutateurs SCALANCE X200 (gérés, pour Ethernet industriel) permettent de monter des topologies en ligne, en étoile ou en anneau sans exploser

le budget. Le fait qu'on puisse créer des anneaux redondants les rend particulièrement adaptés aux réseaux où on exige une haute disponibilité.

Avec leur indice de protection IP30, leur boîtier métallique coulé costaud, et leurs collerettes filetées sur les ports RJ45, ils montrent clairement qu'ils sont faits pour tenir dans des environnements industriels.

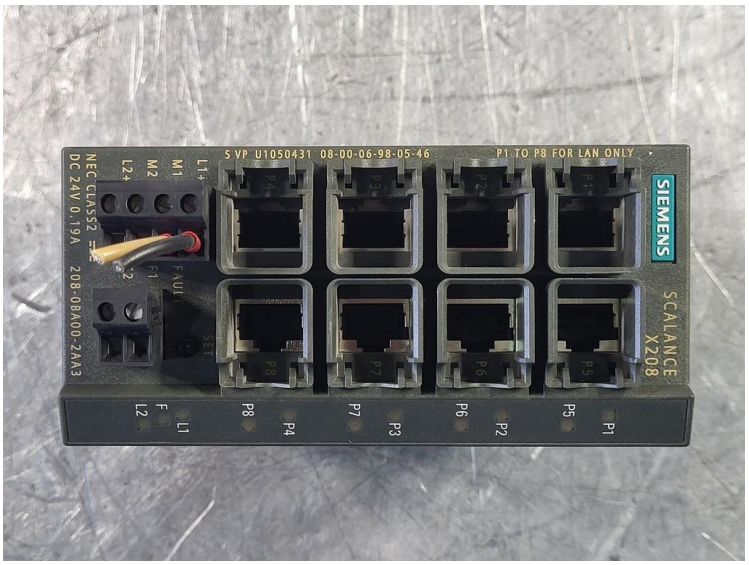


Figure 22-9:commutateurs X200

2.2.7Alimentation SITOP PSU100S

Le SITOP PSU100S, c'est une alimentation à découpage primaire, conçue pour être raccordée sur un réseau monophasé. En sortie, on obtient une tension continue régulée électroniquement, qu'on peut ajuster à l'aide d'un potentiomètre. La sortie est isolée électriquement, et elle supporte aussi bien le fonctionnement à vide que les courtscircuits donc pas de mauvaise surprise. Un voyant LED indique l'état de fonctionnement de l'alimentation. Enfin, on peut exploiter cet état via un contact de signalisation.



Figure 22-10:Alimentation SITOP PSU100S

2.2.8Relais 24V 220V

Un relais électrique est essentiellement composé d'un électroaimant qui contrôle des contacts inverseurs montés sur ressort. Lorsque le courant continu atteint l'électroaimant, celuici attire le mécanisme à ressort, puis le relâche en raison de son influence. Cela provoque un changement dans les bornes de sortie des contacts. Par conséquent, une charge électrique extérieure connectée à ces contacts s'allume ou s'éteint en fonction de la commande du relais.

Le relais 24V 220V sert à convertir un petit signal électrique d'entrée en une puissance de sortie beaucoup plus élevée. Cela est possible grâce à l'électromagnétisme, qui joue un rôle essentiel dans le fonctionnement du relais.

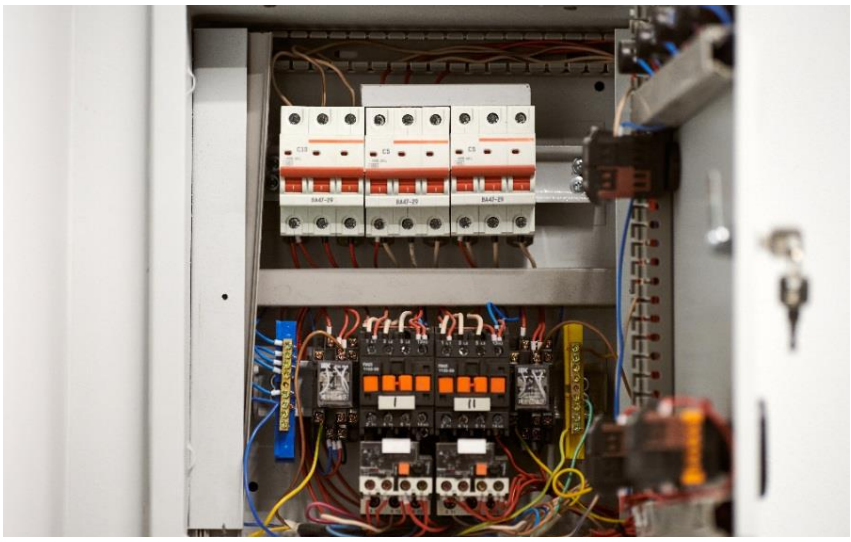


Figure 2-11: Relais 24V 220V

2.2.9Définition d'un IHM industriel

Un IHM industriel est une interface spécialement conçue pour les usines. Elle permet à un opérateur humain d'échanger des informations avec une machine ou un système industriel. Cette interface comprend à la fois des composants matériels et logiciels. Les opérateurs et les chefs de ligne utilisent cet outil pour coordonner les processus de fabrication dans l'usine.

Le panneau Siemens SIMATIC TP900 Comfort est un exemple d'IHM. C'est un écran tactile de 9 pouces qui utilise un processeur X86. Il est différent des anciens modèles SIMATIC car il peut éteindre son affichage de manière centrale pendant les pauses, grâce à PROFIenergy [15]

. Cela aide à réduire la consommation d'énergie. Le panneau Siemens SIMATIC TP900 Comfort est donc un choix intéressant pour les usines qui veulent diminuer leur consommation d'énergie.

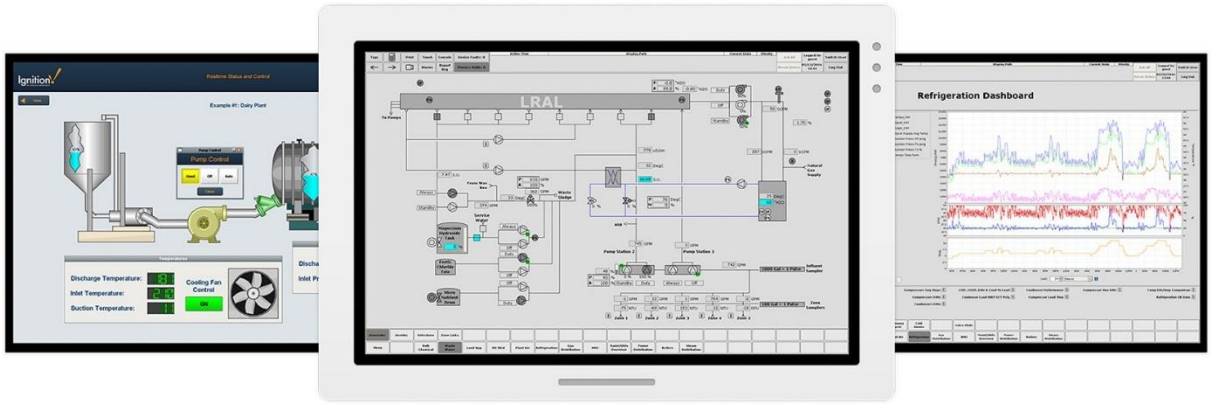


Figure 22-12: IHM industriel

2.2.10Disjoncteur bipolaire

Le disjoncteur bipolaire est un appareil qui respecte les normes en vigueur. Il est notamment conforme à la norme EN 60934 [16]. De plus, il répond à la norme NF F 62001, mais uniquement pour les courbes AN et AS, qui sont spécifiques au matériel roulant ferroviaire [17]. Le disjoncteur bipolaire est principalement utilisé pour protéger les circuits à courant alternatif monophasés. Ces circuits sont généralement de type phase plus neutre. Le disjoncteur bipolaire fonctionne à une

tension de 230 volts et à une fréquence de 50 ou 60 hertz, ce qui correspond aux normes standard pour les disjoncteurs bipolaires. Le disjoncteur bipolaire est donc un élément essentiel pour assurer la sécurité des circuits électriques.



Figure 22-13Disjoncteur Bipolaire

MODEM IDOOM 4G



Figure 22-14: MODEM IDOOM 4G ALGERIE

CABLE RJ45 DINTERNET



Figure 22-15: CABLE RJ45

2.3 Exigences logicielles

2.3.1TIA Portal

Le TIA Portal est une plateforme d'ingénierie qui permet de faire tout ce dont vous avez besoin pour travailler avec les produits Siemens. Vous pouvez utiliser le TIA Portal pour mettre en service, programmer et diagnostiquer les équipements et logiciels Siemens. Tout cela se fait à partir d'une seule interface logicielle. Le TIA Portal comprend des outils de diagnostic système, de communication et bien d'autres fonctionnalités utiles.

Siemens améliore constamment le TIA Portal en ajoutant de nouveaux produits et fonctions de programmation. Cela signifie que le TIA Portal offre une gamme de plus en plus large d'options pour les utilisateurs. Le TIA Portal est conçu pour être facile à utiliser et permet de gérer tous les aspects de l'automatisation à partir d'un seul endroit.

Le TIA Portal est un outil très puissant qui aide les utilisateurs à travailler de manière efficace avec les produits Siemens. Il est très utile pour les personnes qui ont besoin de mettre en service, programmer et diagnostiquer des équipements et

logiciels Siemens. Le TIA Portal est une plateforme d'ingénierie intégrée qui simplifie le travail avec les produits Siemens.

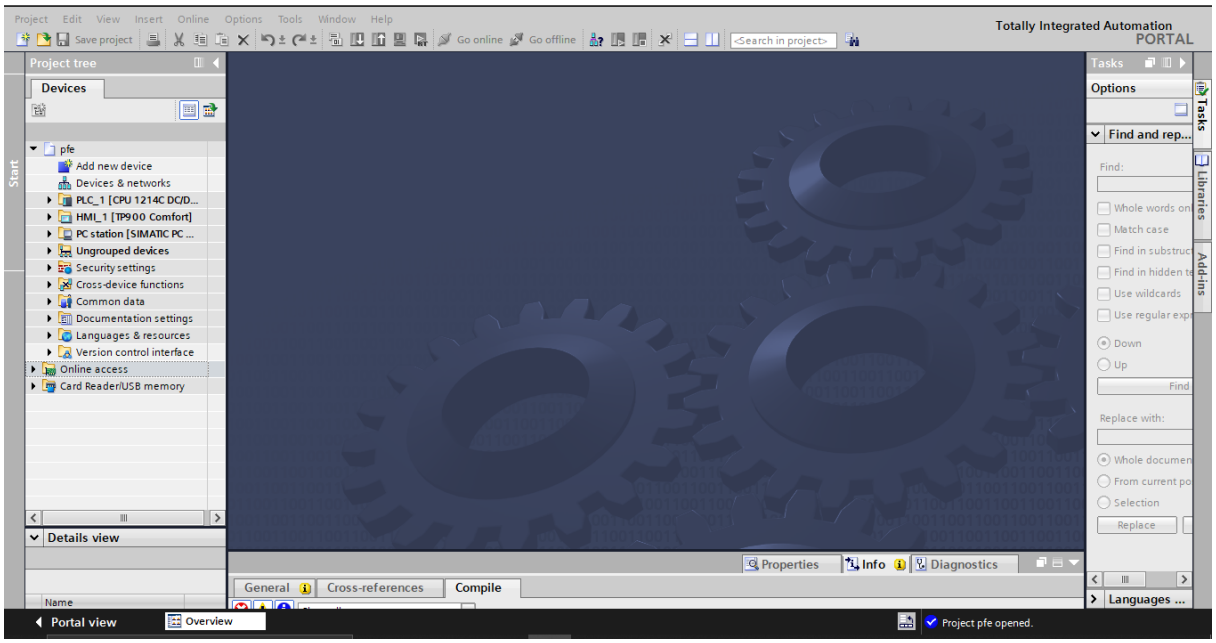


Figure 22-16:Vue Projet de TIA Portal

2.3.2Ladder Logic

également appelée logique à contacts, est une méthode qui a été développée à l'origine pour documenter la conception et le câblage des racks de relais utilisés dans l'industrie, comme dans les secteurs de la fabrication et de la régulation de procédés.

Les schémas à contacts montrent les différents appareils du rack de relais avec leurs symboles et leurs connexions. On y trouve aussi des équipements externes, tels que des pompes et des résistances chauffantes.

Ce type de schéma permet de bien comprendre et d'entretenir plus facilement les systèmes industriels.

Les racks de relais sont des ensembles de relais électriques qui servent à commander et à contrôler les processus industriels.

Actuellement, la logique à contacts est encore utilisée pour programmer les automates programmables industriels.

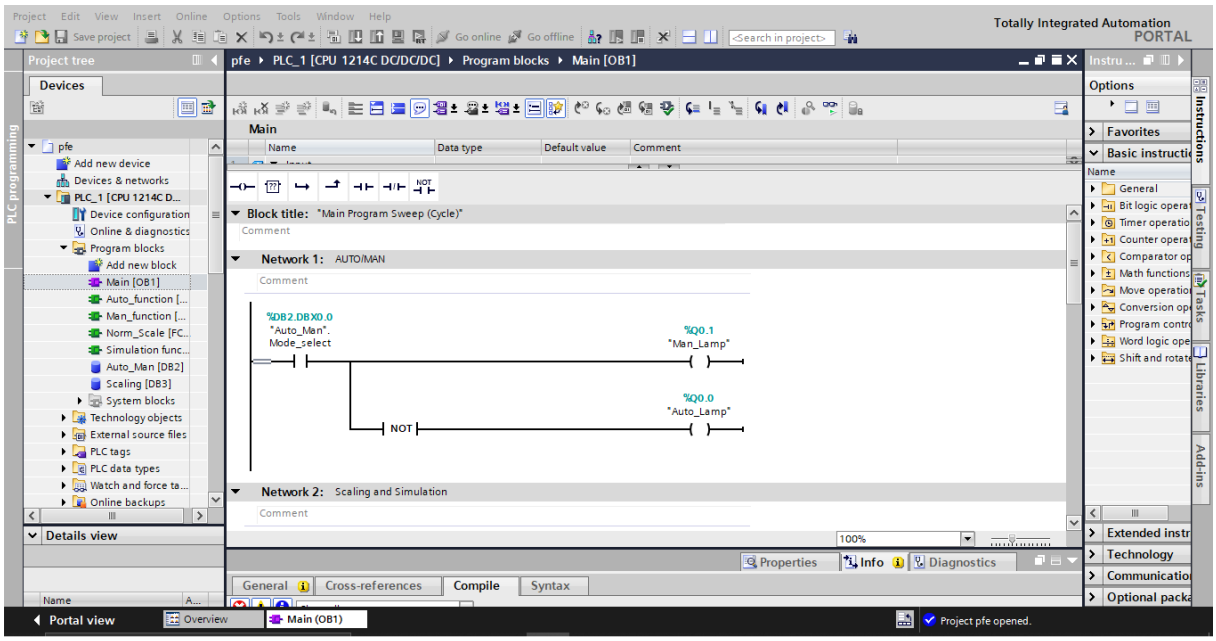


Figure 22-17:Laddar logic

Avec le temps, le Ladder a évolué pour devenir un véritable langage de programmation. Le principe : un programme est représenté par un schéma graphique qui s'inspire directement des schémas électriques des automates câblés d'époque. On utilise ce langage pour développer des logiciels pour API (automates programmables industriels), ceux qu'on trouve dans les applications de contrôle en industrie. Le nom vient simplement du fait que les programmes écrits dans ce langage ressemblent à des échelles avec deux rails verticaux, et une série d'échelons horizontaux entre eux.

À une époque, ces schémas à contacts étaient la seule manière de noter les programmes pour automate. Aujourd'hui, d'autres formats sont normalisés dans la norme IEC 611313. Par exemple, en alternative au Ladder graphique, il existe un langage plus proche du C, appelé Structured Text (texte structuré), qui fait également partie de cette même norme. [18]

2.3.3Microsoft Visual Studio

Visual Studio, c'est l'environnement de développement intégré (IDE) de Microsoft. Il permet d'écrire, déboguer et compiler du code pour à peu près tout ce qui tourne sous Windows et même audelà. On peut y faire du C#, du C++, du

Python, du web, des applications mobiles, etc. Ce qui le caractérise, c'est son côté "lourd" mais complet : débogueur puissant, conception d'interfaces graphiques en glisserdéposer, intégration avec les services Azure, et une flopée de plugins. Les versions Community (gratuite) et Professional/Enterprise (payantes) cohabitent selon les besoins.

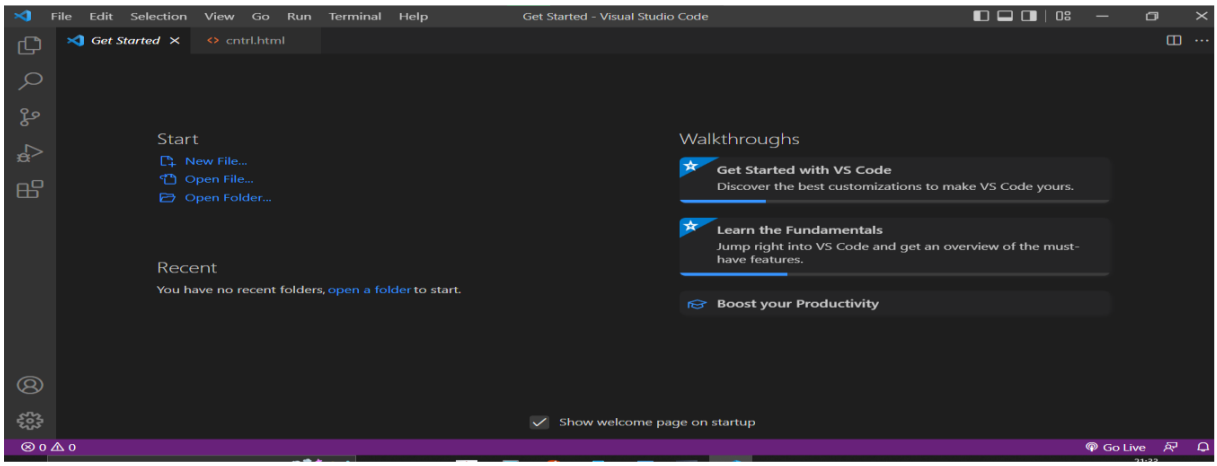


Figure 22-18: MICROSOFT VISUAL STUDIO

2.3.4HTML5

est un langage de balisage utilisé pour structurer et afficher du contenu sur le web. Il s'agit de la cinquième et dernière version majeure du HTML, approuvée par le W3C. Actuellement, la spécification est appelée le HTML Living Standard, gérée par le WHATWG.

Ce groupe est constitué des principaux éditeurs de navigateurs : Apple, Google, Mozilla et Microsoft.

HTML5 se distingue par des modèles de traitement détaillés qui améliorent la compatibilité entre les différentes implémentations.

Il améliore et clarifie le balisage pour les documents, introduit de nouvelles balises et des API pour les applications web complexes.

HTML5 est aussi idéal pour développer des applications mobiles multiplateformes.

Il intègre des fonctionnalités conçues pour les appareils avec peu de ressources.

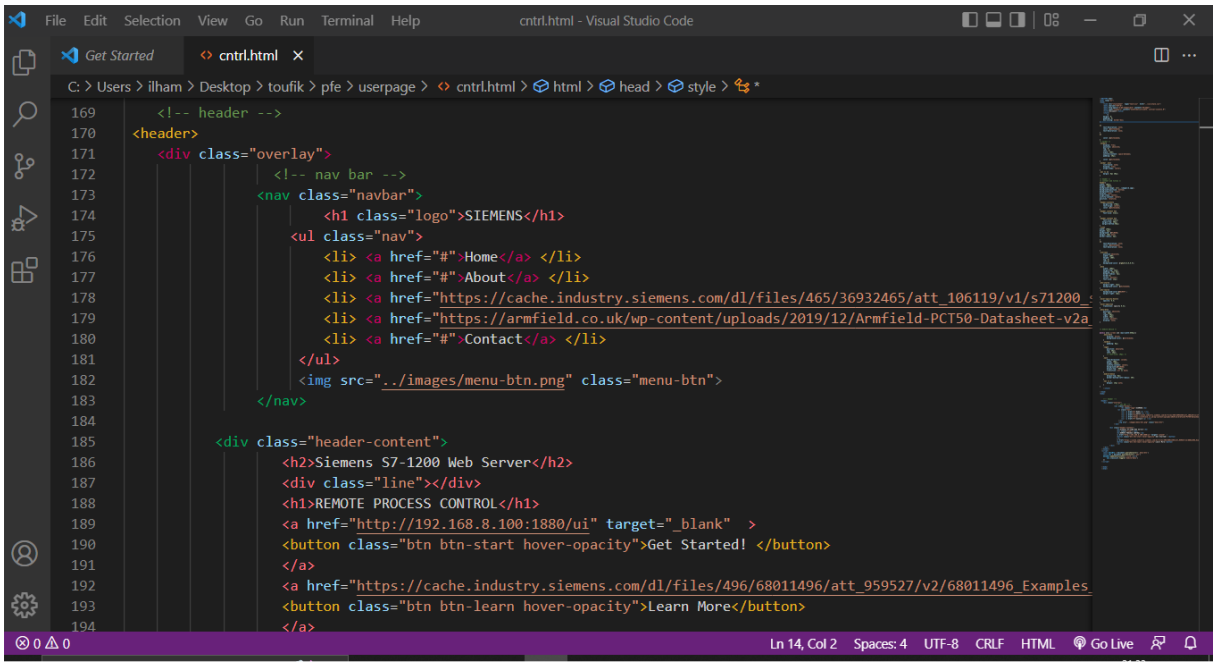


Figure 22-19: HTML CODE

2.3.5CSS3

Le langage CSS, ou feuilles de style en cascade, est utilisé pour décrire la présentation d'un document écrit en langage de balisage. CSS est utilisé avec HTML et JavaScript, qui sont des technologies de base du World Wide Web.

L'idée principale de CSS est de séparer la présentation et le contenu. Cela signifie que la mise en page, les couleurs et les polices de caractères sont gérées de manière indépendante. Cette approche présente plusieurs avantages. Elle améliore l'accessibilité du contenu, offre plus de flexibilité et de contrôle sur le rendu visuel. De plus, plusieurs pages web peuvent partager la même mise en forme en utilisant un fichier CSS séparé. Cela réduit la complexité et les répétitions dans le code structurel. Enfin, le fichier CSS peut être mis en cache, ce qui accélère le chargement des pages qui partagent le même fichier et la même mise en forme.

CSS3 est la dernière version de la spécification CSS. Elle apporte de nouvelles fonctionnalités stylistiques et des améliorations qui enrichissent les possibilités de présentation sur le web.

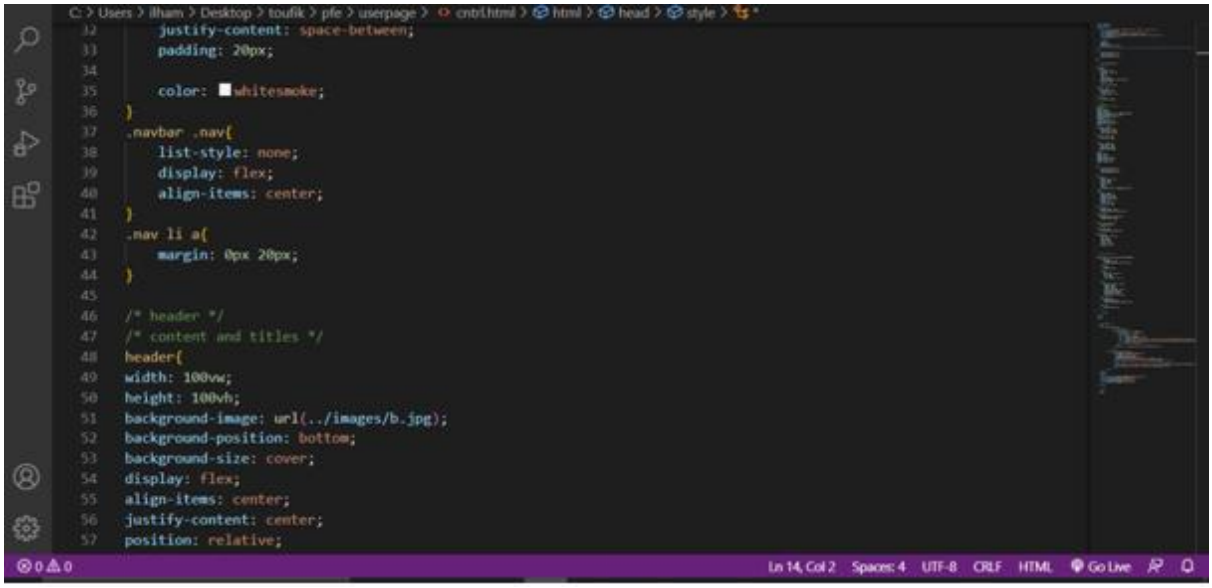


Figure 22-20: CSS CODE

2.3.6JavaScript

JavaScript, c'est un langage de programmation qui tourne principalement côté navigateur. Il sert à rendre les pages web interactives : gérer des clics, modifier le contenu à la volée, envoyer ou recevoir des données sans recharger la page (c'est là qu'on pense à AJAX, fetch, ce genre de choses). À l'origine, il était cantonné au frontend, mais avec Node.js, il a envahi le backend aussi. Aujourd'hui, difficile de faire du web moderne sans lui, même si son écosystème change tellement vite qu'on a parfois du mal à suivre.

2.3.7NodeRED

NodeRED est un outil de développement qui repose sur le principe des flux. Il a été créé à l'origine par IBM pour permettre à différents composants matériels, des API et des services en ligne de communiquer entre eux, notamment dans le contexte de l'Internet des Objets.

L'outil propose un éditeur de flux qui fonctionne dans un navigateur web. Avec cet éditeur, il est possible de créer des fonctions en utilisant le langage de programmation JavaScript. Les différents éléments que l'on assemble peuvent être sauvegardés ou partagés pour être réutilisés dans d'autres projets. Le moteur d'exécution de NodeRED est basé sur Node.js. Les flux créés avec NodeRED sont enregistrés au format JSON. Depuis la version 0.14, les nœuds MQTT peuvent établir des connexions sécurisées TLS correctement configurées.

En 2016, IBM a décidé de donner NodeRED au projet open source de la JS Foundation.

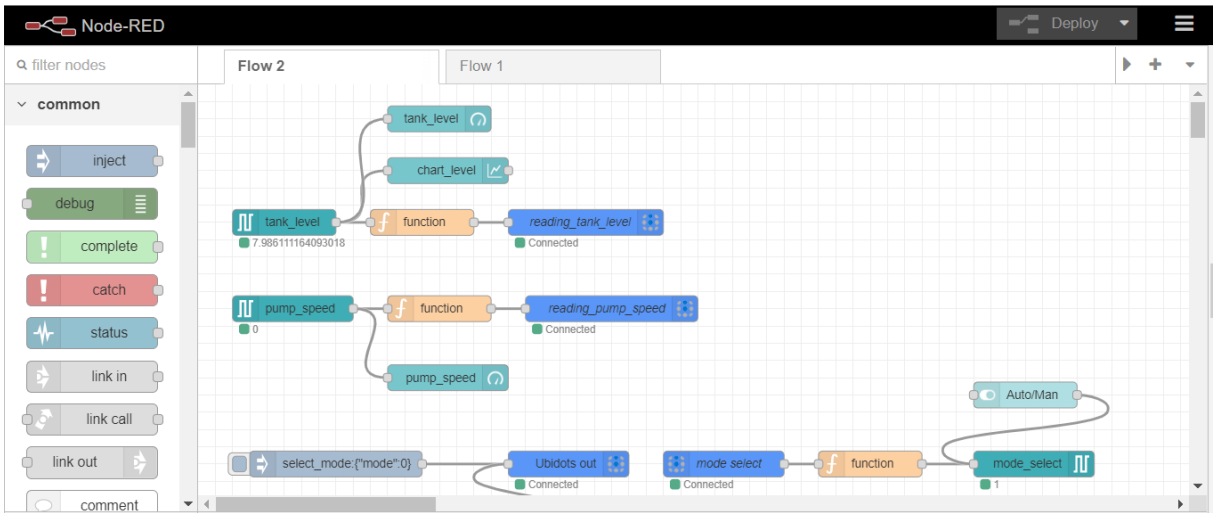


Figure 22-21: NOTERED

2.3.8JSON

JSON (JavaScript Object Notation, qu'on prononce souvent "djéson") est un format de fichier et d'échange de données, ouvert et standardisé. Il utilise du texte lisible par un humain pour stocker et transmettre des objets de données : essentiellement des couples clé–valeur (attribut–valeur) et des tableaux (arrays). C'est un format qu'on retrouve partout dans les échanges de données électroniques par exemple entre une application web et son serveur.



Figure 22-22: MESSAGE DE JSON

2.3.9Service Cloud (Ubidots)

Ubidots est une plateforme IoT polyvalente qui convient aussi bien aux prototypes, projets éducatifs ou de recherche qu'aux projets IoT industriels prêts à être déployés. Elle propose différentes formules, y compris une version gratuite pour les projets non commerciaux, qui inclut la plupart des fonctionnalités de la version payante.

Avec Ubidots, vous pouvez envoyer et recevoir des données vers le cloud depuis des appareils connectés à Internet. Plusieurs protocoles sont pris en charge, tels que MQTT, HTTP via leur API REST, ou TCP/UDP en fonction des besoins. L'un des avantages d'Ubidots est sa capacité à jouer le rôle de courtier, c'estàdire à servir de middleware pour les données.

La plateforme permet également de visualiser et de modifier les données transmises à l'aide de widgets glisserdéposer, qui peuvent être intégrés dans des tableaux de bord. Cela vous permet de créer une application IoT conviviale pour votre équipe ou vos clients en peu de temps.



Figure 22-23: Page d'accueil de la plateforme Ubidots

CHAPITRE 3 :

Réservoirs, régulation de niveau et instrumentation associée

3. CHAPITRE 3 :Réservoirs, régulation de niveau et instrumentation associée

La régulation de niveau est l'une des fonctions les plus courantes dans l'industrie. Qu'il s'agisse de maintenir un liquide à une hauteur donnée dans une cuve, d'éviter un débordement ou d'assurer un débit constant en sortie, on retrouve ce besoin dans des domaines très variés : traitement des eaux, chimie, agroalimentaire, énergie, etc. Avant de s'intéresser à un système particulier comme le PCT50, il est utile de faire un tour d'horizon des différents types de réservoirs que l'on rencontre en pratique, des stratégies de régulation qu'on peut leur appliquer, et des instruments qui permettent de mesurer et d'agir. Ce chapitre présente donc ces éléments de façon générale, afin de mieux situer le cas simulé qui sera détaillé plus loin.

3.1 Typologie des réservoirs industriels

3.1.1 Classification technique : caractéristiques par géométrie et par pression

Dans le contrôle des procédés et l'automatisation industrielle, l'analyse d'un système de régulation de niveau nécessite la prise en compte des caractéristiques géométriques et physiques du réservoir. Cellesci influencent directement la modélisation du système ainsi que le choix des dispositifs de mesure et de contrôle.

3.1.1.1 Caractéristiques géométriques : réservoirs horizontaux et verticaux

La géométrie d'un réservoir qu'elle soit en deux ou trois dimensions n'est jamais anodine. Elle détermine en effet comment le volume de liquide (V) se comporte par rapport à la hauteur mesurée (h). Et cette relation hauteurvolume, justement, est fondamentale quand on veut configurer correctement les algorithmes de commande dans un automate. Si on la méconnaît ou qu'on l'approxime mal, la

régulation risque de devenir instable ou inefficace, même avec un bon régulateur en théorie.

3.1.1.2 Réservoirs cylindriques verticaux

C'est la configuration qu'on rencontre le plus souvent dans la régulation industrielle. Du point de vue de l'automatisme, ces réservoirs ont un avantage majeur : leur section transversale (A) reste constante quelle que soit la hauteur de liquide. Du coup, l'équation de continuité se simplifie : on obtient une relation linéaire où le débit volumétrique (entrant ou sortant) entraîne une variation de hauteur uniforme, soit $\frac{dh}{dt} = \frac{Q_{entrant} - Q_{sortant}}{A}$

Cette linéarité change la donne quand on paramètre un régulateur PID. Parce que la dynamique du système ne varie pas avec le niveau de remplissage. En clair, si tu règles correctement tes gains pour un point de fonctionnement, ils restent valables pour toute la plage de niveau. C'est un gros avantage en pratique, car ça évite d'avoir à recalibrer sans cesse.

3.1.1.3 Réservoirs cylindriques horizontaux

Les réservoirs cylindriques horizontaux sont principalement utilisés pour le stockage de fluides sous pression ou lorsque la hauteur disponible est limitée. Contrairement aux cuves verticales, leur section liquide varie de manière non linéaire avec le niveau h, ce qui complique la régulation. Pour garantir une mesure précise du volume, un bloc de linéarisation est généralement intégré à l'automate afin de convertir la hauteur mesurée en volume réel. Sans cette correction, le comportement dynamique du système varie selon le niveau de remplissage, ce qui dégrade les performances de la régulation.

3.1.2Caractéristiques de pression : Réservoirs ouverts et sous pression

L'état de l'atmosphère interne du réservoir dicte le choix technologique des capteurs de mesure de niveau et influence les perturbations dynamiques du système.

3.1.2.1 Réservoirs ouverts (à pression atmosphérique)

Ces réservoirs qu'on appelle souvent bâches ou cuves ouvertes sont en contact direct avec l'air ambiant. Du coup, la pression à la surface du liquide reste constante et égale à la pression atmosphérique (Patm). C'est une configuration plutôt simple pour la mesure de niveau par pression hydrostatique : il suffit d'installer un capteur de pression relative au point bas de la cuve, et on déduit la hauteur via la loi de Pascal ($P = \rho \cdot g \cdot h$). Pas de perturbations liées à des variations de pression externe à gérer, ce qui facilite pas mal les choses.

3.1.2.2 Réservoirs fermés et sous pression

Les réservoirs fermés sont utilisés pour le stockage de fluides volatils, toxiques ou nécessitant une pression de service contrôlée. Dans ce cas, la pression interne s'ajoute à la pression hydrostatique du fluide, ce qui peut fausser la mesure du niveau avec un simple capteur de pression. Pour obtenir une mesure fiable, on utilise généralement des capteurs de pression différentielle ou des technologies moins sensibles à la pression gazeuse, telles que les capteurs radar ou à ultrasons.

3.1.3 Fonctions et domaines d'application : Synthèse des cas d'usage industriels

La maîtrise du niveau et du stockage des fluides est une problématique transversale en ingénierie des procédés. Selon le secteur industriel, les réservoirs n'assurent pas uniquement un rôle de confinement passif, mais participent activement à la chaîne de production à travers plusieurs fonctions clés :

3.1.3.1 Fonctions opératoires des réservoirs

Stockage et tampon (buffer) : Ici, le réservoir sert à absorber les variations de débit entre deux étapes de production. L'idée, c'est de garantir une alimentation continue en aval, même si en amont le débit n'est pas parfaitement régulier. C'est une fonction qu'on trouve très souvent dans les chaînes de fabrication où plusieurs unités fonctionnent à des cadences différentes.

Mélange et homogénéisation : Ce sont généralement des cuves équipées d'agitateurs. Le but est d'obtenir une solution homogène, que ce soit en termes de concentration ou de température. On veut éviter les zones où les propriétés diffèrent trop, sinon la suite du procédé peut être perturbée.

Décantation et séparation : On exploite ici tout simplement la gravité. Le mélange hétérogène (par exemple solide/liquide, ou bien deux liquides de densités différentes) est laissé au repos, et les phases se séparent naturellement. C'est une méthode courante dans le traitement des eaux ou dans certaines industries chimiques.

3.1.3.2 Domaines d'application et spécificités de contrôle

Traitement des eaux et stations de pompage : Dans les châteaux d'eau et les stations d'épuration, la gestion des réservoirs répond à des objectifs de distribution et de protection environnementale. La régulation de niveau y est essentielle, notamment pour optimiser les cycles de démarrage des pompes de forte puissance on cherche à limiter leur usure mécanique et leur surconsommation électrique. Les automatismes doivent aussi gérer des alarmes strictes : un "très haut niveau" pour éviter les débordements, et un "très bas niveau" pour protéger les pompes contre la marche à sec ou la cavitation.

Industries de procédé chimique et pharmaceutique : Ici, les réservoirs sont souvent des réacteurs. Maintenir un niveau rigoureux conditionne directement le temps de séjour des réactifs, et donc la cinétique chimique. Les enjeux de sécurité sont maximaux : le contrôlecommande doit intégrer des boucles d'urgence très fiables. Une simple défaillance de niveau peut déséquilibrer la température du procédé et déclencher des accidents industriels graves surpression, ou réactions exothermiques incontrôlées.

Industrie agroalimentaire : On trouve ces réservoirs dans les laiteries, les brasseries ou les cuves de fermentation. Ils imposent des contraintes sévères de neutralité et d'hygiène (normes NEP ou CIP, pour Nettoyage En Place). Côté

instrumentation, les capteurs ne doivent en aucun cas altérer le produit on privilégie donc les technologies sans contact comme les ultrasons ou le radar. La régulation doit être extrêmement précise pour garantir la reproductibilité des recettes et une gestion rigoureuse des matières premières.

3.2 Stratégies de Régulation de Niveau

Les stratégies de régulation de niveau permettent de maintenir le niveau d'un liquide à une valeur souhaitée en ajustant automatiquement le débit d'entrée ou de sortie du réservoir.

3.2.1 Régulation Tout-ou-Rien (On/Off) : fonctionnement par seuils de niveau

La régulation Tout-ou-Rien ou On/Off control est l'approche la plus basique et pourtant l'une des plus utilisées quand la précision dynamique n'est pas une priorité [19]. Le principe est simple : on commande l'actionneur de façon binaire (0 ou 1, marche ou arrêt, ouvert ou fermé) en fonction du franchissement de seuils de niveau qu'on a fixés à l'avance. Dès que le niveau atteint la consigne haute, on coupe ; dès qu'il descend jusqu'au seuil bas, on rallume. C'est rustique, mais ça suffit largement pour des procédés où les variations sont lentes et où on accepte une petite oscillation autour du point de consigne.

3.2.1.1 Principe de fonctionnement et bande morte (hystérésis)

Dans une boucle Tout-ou-Rien idéale, l'actionneur bascule dès que le niveau (y(t)) atteint la consigne (w). Mais en réalité, sans marge de tolérance, la moindre fluctuation du liquide provoquerait des commutations incessantes. Ce phénomène de chattering détruirait rapidement les contacteurs et les pompes. Pour éviter ça, on ajoute une bande morte (hystérésis) autour de la consigne [20]. Le système fonctionne alors avec deux seuils : un seuil bas (S_bas) qui

déclenche le remplissage, et un seuil haut (S_haut) qui l'arrête. En mode Inflow, la commande u(t) obéit à la logique suivante :

$$u(t) = \begin{cases} 1(\text{ON}) & \text{si} y(t) \leq S_{bas} \\ 0(\text{OFF}) & \text{si} y(t) \geq S_{haut} \\ u(t-\Delta t) & \text{si} S_{bas} < y(t) < S_{haut} \end{cases}$$

Cette équation traduit une chose simple : tant que le niveau reste à l'intérieur de la bande morte$[S_{bas}, S_{haut}]$, l'actionneur ne bouge pas il garde son état précédent $u(t-\Delta t)$. C'est comme ça qu'on assure la stabilité du système logique.

3.2.1.2 Analyse dynamique et limites industrielles

Dynamiquement parlant, la régulation Tout-ou-Rien n'atteint jamais un équilibre stable. Le niveau oscille en permanence entre le seuil bas et le seuil haut, dessinant une courbe en dents de scie. L'amplitude des oscillations dépend directement de la largeur de la bande morte, et leur fréquence du débit de remplissage et de vidange du réservoir. Une bande morte large réduit l'usure des actionneurs, mais au prix d'une moins bonne précision du niveau.

Côté programmation, c'est simple à faire dans un automate (quelques blocs de comparaison logique) et peu coûteux en instrumentation (des capteurs à flotteur suffisent). Mais les limites sont claires : ce type de régulation convient surtout au stockage brut châteaux d'eau, cuves de récupération. Dès qu'on a besoin d'un débit stable ou d'une consigne de niveau rigoureuse, on oublie le Tout-ou-Rien et on passe sur une régulation continue de type PID.

3.2.2Régulation linéaire (P, PI, PID) : principes mathématiques et correction dynamique

À l'inverse du Tout-ou-Rien qui fonctionne par à-coups, la régulation linéaire continue cherche à maintenir le niveau du réservoir (y(t)) rigoureusement égal à la consigne (w(t)). L'algorithme calcule en permanence l'écart, qu'on appelle signal d'erreur, défini par :

$$e(t) = w(t) - y(t)$$

on a l'erreur notée classiquement e(t)*e*(*t*) qui représente l'écart entre le niveau qu'on veut et celui qu'on a réellement. À partir de là, le régulateur PID calcule une commande u(t)*u*(*t*) qui va agir sur l'actionneur (par exemple, ouvrir plus ou moins une vanne, ou faire varier la vitesse d'une pompe) [21]. Cette commande n'est pas binaire (marche/arrêt), mais continue, et elle combine trois actions : proportionnelle, intégrale et dérivée. Chacune a son rôle.

3.2.2.1 Formulation mathématique globale

Dans sa version idéale, l'équation temporelle d'un PID s'écrit comme ceci :

$$u(t) = K_p \cdot e(t) + \frac{K_p}{T_i}\int_0^t e\,(\tau)d\tau + K_p \cdot T_d \cdot \frac{de(t)}{dt}$$

Où :

- K_p représente le gain proportionnel.
- T_i désigne la constante de temps intégrale.
- T_d correspond à la constante de temps dérivée.

Dans le domaine de Laplace, essentiel pour l'analyse fréquentielle et la modélisation de la fonction de transfert du système, cette relation devient :

$$C(s) = \frac{U(s)}{E(s)} = K_p(1 + \frac{1}{T_i s} + T_d s)$$

3.2.3Analyse des actions et correction dynamique du niveau

Chacune des trois actions du PID a un rôle bien précis quand on parle de réguler un réservoir.

3.2.3.1 L'action proportionnelle (Kp) :

elle réagit tout de suite, et son intensité dépend directement de l'erreur qu'on mesure. Plus on augmente Kp, plus le système répond vite et moins il met de temps à monter vers la consigne. Mais attention, si on utilise uniquement du

proportionnel, on ne pourra jamais annuler complètement l'erreur en régime permanent il restera toujours un petit écart, ce qu'on appelle une erreur statique. Et si on monte trop le gain, ça oscille et ça peut même partir en instabilité.

3.2.3.2 L'action intégrale (Ti) :

c'est elle qui vient corriger ce que le proportionnel laisse traîner. En accumulant les erreurs passées, elle finit par pousser la commande jusqu'à ce que l'écart soit nul (e(t)=0). Dans une cuve, ça permet d'absorber des perturbations constantes comme une petite fuite ou un débit de sortie légèrement mal connu.
L'inconvénient ? Ça ralentit le système, ça retarde la réponse, et ça peut provoquer un dépassement au départ. Autre point important : dans l'automate, il faut penser à configurer l'anti-windup, sinon l'intégrale peut saturer si la vanne est bloquée [22].

3.2.3.3 L'action dérivée (Td) :

elle ne regarde pas l'erreur elle-même, mais sa vitesse de variation. Autrement dit, elle anticipe. Si le niveau se met à monter ou descendre brusquement, la dérivée agit tout de suite pour freiner le mouvement avant que ça devienne problématique. C'est un effet d'amortissement qui réduit les dépassements et stabilise le système. Mais en pratique, sur un réservoir d'eau, on s'en sert rarement sauf si le procédé a des temps de retard importants. Pourquoi ? Parce que la dérivée amplifie les bruits de mesure (turbulences, vagues à la surface), et il faut alors ajouter un filtre passe-bas dans l'automate pour éviter que ça devienne contre-productif.

3.2.3.4 En pratique pour un réservoir standard :

on utilise souvent un régulateur PI, pas un PID complet. L'inertie du système est modérée, l'action dérivée n'apporte pas grand-chose et peut même être gênante. On la réserve pour des procédés plus complexes où les temps de retard sont significatifs.

3.3 Instrumentation de Terrain pour la Mesure et l'Action

Ensemble des capteurs et actionneurs utilisés pour mesurer les variables du procédé et agir sur celui-ci en temps réel.

3.3.1 Technologies de capteurs de niveau : hydrostatique, ultrason, radar, capacitif

Avant de pouvoir réguler, il faut mesurer. Et le choix du capteur dépend de ce qu'on mesure (fluide corrosif ? visqueux ?), de la forme du réservoir, et des conditions ambiantes (pression, température, turbulences). On distingue deux grandes familles : les capteurs à contact et ceux sans contact.

3.3.1.1 Mesure hydrostatique (contact)

C'est la loi de Pascal : $P_{hydro} = \rho \cdot g \cdot h$ [23]On place un capteur de pression au point bas de la cuve. Pour un réservoir ouvert, un capteur de pression relative suffit. Pour un réservoir sous pression, il faut un capteur de pression différentielle (ΔP) pour soustraire la pression du ciel gazeux. C'est robuste, économique, mais ça dépend de la densité du fluide (ρ). Si la température varie ou si le fluide change, il faut compenser dans l'automate [24].

3.3.1.2 Mesure ultrasonique (sans contact)

Le capteur, placé au-dessus du liquide, émet une onde acoustique et mesure le temps qu'elle met pour revenir (time-of-flight) [25]. Pas de contact, donc idéal pour les fluides corrosifs ou l'agroalimentaire. Par contre, la vitesse du son dépend de la température et de la nature du gaz heureusement, la plupart des capteurs intègrent une compensation thermique. Attention aussi : ça ne fonctionne pas sous vide, ni en forte pression, ni si la surface du liquide est couverte de mousse (l'onde se perd).

3.3.1.3 Mesure radar (sans contact ou guidée)

Même principe que l'ultrason, mais avec des ondes électromagnétiques qui voyagent à la vitesse de la lumière [26]. C'est le haut du panier en fiabilité : les micro-ondes ne sont pas sensibles à la température, à la pression, aux vapeurs ou

à la poussière. Précision millimétrique. Deux versions existent : à émission libre (sans contact) ou à ondes guidées (une sonde métallique plonge dans le liquide). Seule limite : si le fluide a une constante diélectrique très faible (certains hydrocarbures purs), le signal réfléchi est faible on passe alors en version guidée.

3.3.1.4 Mesure capacitive (contact)

Ici, on transforme le réservoir en condensateur. Une sonde isolée descend dans la cuve, la paroi métallique fait office d'autre armature [27]. Au fur et à mesure que le liquide monte, il remplace l'air entre les armatures, et la capacité varie linéairement avec la hauteur. Très efficace pour les liquides homogènes, les poudres, et ça tient des températures ou pressions extrêmes (chaudières industrielles). L'inconvénient : l'encrassement. Si un dépôt collant reste sur la sonde après vidange, la mesure devient fausse (le capteur croit qu'il reste du liquide).

3.3.2 Organes de commande d'un fluide : pompes à vitesse variable, vannes motorisées et électrovannes

Si les capteurs permettent de "voir" ce qui se passe dans le réservoir, les actionneurs font le travail inverse : ils reçoivent les ordres de l'automate (PID ou Tout-ou-Rien) et agissent physiquement sur le débit pour stabiliser le niveau. Trois grandes technologies dominent dans les installations hydrauliques.

3.3.2.1 Pompes à vitesse variable (commande continue)

C'est la solution privilégiée pour le contrôle par débit entrant (inflow). La pompe n'est pas branchée directement sur le secteur, mais pilotée par un variateur électronique (VFD). L'automate envoie un signal analogique (4-20 mA ou 0-10 V) ou numérique (PROFINET) au variateur, qui ajuste la fréquence et la tension du moteur. Le débit d'une pompe centrifuge étant proportionnel à sa vitesse de rotation, le PID peut régler l'apport d'eau très finement. Avantages : bonne efficacité énergétique (la puissance varie avec le cube de la vitesse), et possibilité

de programmer des rampes d'accélération douces fini les coups de bélier et l'usure prématurée.

3.3.2.2 Vannes motorisées (commande continue)

Quand l'écoulement se fait par gravité ou depuis un réseau sous pression constante, on utilise plutôt une vanne de régulation (à siège, à bille ou à papillon) équipée d'un servomoteur. Un positionneur intégré reçoit la consigne de l'automate (ex : 50 % d'ouverture) et déplace la tige ou l'axe pour modifier la section de passage. La relation entre la course de la vanne et le débit définit sa caractéristique (linéaire, ouverture rapide, ou égal pourcentage). Indispensable pour le contrôle du débit sortant (outflow), mais attention à l'autorité de la vanne si elle ne représente qu'une faible part des pertes de charge du réseau, la boucle PID risque d'être instable.

3.3.2.3 Électrovannes (commande Tout-ou-Rien)

L'électrovanne, c'est du binaire pur : ouverte ou fermée, pas de position intermédiaire [28]. Principe simple : une bobine magnétique (solénoïde) reçoit du courant (souvent 24 V CC ou 230 V CA), génère un champ qui attire un noyau mobile et vient vaincre un ressort. Deux types existent : normalement fermée (NF) elle s'ouvre seulement sous tension et normalement ouverte (NO) elle se ferme sous tension, souvent pour des raisons de sécurité. L'électrovanne est parfaite pour la régulation Tout-ou-Rien, les purges rapides d'urgence, ou le basculement entre deux cuves. C'est économique et très rapide (quelques millisecondes). Mais à éviter pour du PID, puisqu'on ne peut pas moduler le débit en continu.

3.4 Spécification du Modèle d'Étude pour la Simulation

Cette section précise les paramètres physiques et opératoires du système simulé : géométrie du réservoir, fluide, actionneurs, capteurs, et perturbations. Le but est d'obtenir un modèle de référence reproductible pour évaluer et comparer les stratégies de régulation.

3.4.1 Description fonctionnelle du procédé à simuler : choix des limites du système de réservoir

Comme on ne dispose pas d'un banc d'essai réel, on doit construire un jumeau numérique — une simulation qui reste crédible d'un point de vue industriel. Pour ça, on s'inspire d'un système bien connu dans l'enseignement : le module PCT50 d'Armfield (ou un équivalent). C'est une maquette qui sert couramment à tester des boucles de régulation de niveau.

3.4.1.1 Architecture globale du système

Le système simulé est un circuit hydraulique fermé avec deux éléments :

- **Le réservoir de processus** (cuve supérieure) : cylindrique vertical, ouvert à l'atmosphère. C'est là qu'on veut stabiliser le niveau d'eau.
- **Le réservoir d'alimentation** (bâche inférieure) : il stocke l'eau et récupère ce qui vient du réservoir supérieur par gravité.

La régulation se fait en mode **débit entrant (inflow control)**. Une pompe centrifuge à vitesse variable (commandée par l'automate) remonte l'eau de la bâche vers la cuve supérieure [29]. En bas de cette cuve, une vanne de vidange (réglable manuellement ou par programmation) évacue l'eau vers la bâche — elle joue le rôle d'une perturbation, comme une consommation variable en aval.

3.4.1.2 Limites du système pour la simulation

Pour que les calculs restent simples sans perdre en réalisme, on pose quelques hypothèses :

- **Fluide** : eau pure, température constante (20°C). Masse volumique (ρ = 1000 kg/m³) et viscosité fixes.
- **Géométrie** : section du réservoir constante (A). On ignore les effets de courbure du fond et les turbulences de surface. Le niveau h(t) varie entre 0% (cuve vide) et 100% (hauteur max avant débordement).

- **Actionneurs** : la pompe reçoit une consigne entre 0 V (arrêt, débit nul) et 10 V (débit max Qmax). Le temps de réponse du variateur et l'inertie du moteur sont simplifiés en un retard du premier ordre.
- **Perturbations** : la vanne de vidange est la perturbation principale. Dans la simulation, son ouverture sera modélisée soit comme une fuite constante, soit comme un échelon de débit programmable. Ça permet de tester si le régulateur PID arrive à compenser.

Avec ces limites bien posées, le réservoir se comporte comme un système dynamique à paramètres localisés. Il ne reste plus qu'à le traduire en équations différentielles, puis en blocs fonctionnels dans l'automate.

3.4.2Identification des variables : PV, CV et perturbations

Avant de programmer l'automate, il faut classer proprement les grandeurs physiques qui interviennent dans notre système. Dans notre simulation de réservoir (inspirée du PCT50), on distingue trois types de variables.

3.4.2.1 La variable de procédé (PV - Process Variable)

C'est ce qu'on veut contrôler et maintenir stable. Ici, il s'agit du niveau d'eau h dans le réservoir supérieur, exprimé en cm ou en m, mais aussi souvent normalisé entre 0 et 100% dans l'IHM. En conditions réelles, un capteur (hydrostatique, ultrason, etc.) convertit cette hauteur en signal 4-20 mA ou 0-10 V. Dans la simulation, ce signal servira de retour (feedback) pour le PID.

3.4.2.2 La variable de commande (CV ou MV - Control/Manipulated Variable)

C'est le signal que l'automate envoie pour agir sur le procédé et corriger l'erreur e = SP - PV. Dans notre boucle en mode inflow, la variable de commande est le débit entrant Qentrant. Concrètement, l'automate module la vitesse de la pompe via un variateur (VFD). Attention, cette commande u(t) est physiquement saturée : elle ne peut sortir de l'intervalle [0% (pompe à l'arrêt) ; 100% (vitesse max)].

C. Les variables de perturbation (DV - Disturbance Variables)

Les perturbations, ce sont toutes les grandeurs externes qui viennent perturber le niveau sans qu'on les commande directement.

- **Perturbation principale** : le débit sortant Qsortant, à la base du réservoir. En usine, il varie selon la demande des unités aval. Dans la simulation, on fera varier l'ouverture de la vanne de vidange par échelons pour tester la robustesse du régulateur.
- **Perturbations secondaires** : on peut aussi ajouter une petite fuite non contrôlée, ou un bruit haute fréquence sur la mesure pour simuler les turbulences de surface.

D. Synthèse pour l'automate

Ces variables seront mappées dans la mémoire de l'automate (entrées/sorties, blocs de données) avant d'être éventuellement remontées vers le cloud.

Type de Variable	Dénomination Technique	Symbole	Rôle dans le Bloc PID de l'API	Destination Cloud (Ubidots)
Consigne	*Set Point* (SP)	$w(t)$	Valeur cible fixée par l'opérateur	Visualisation et modification
Procédé	*Process Variable* (PV)	$h(t)$	Entrée de mesure (Feedback)	Tendance temporelle (Graphique)
Commande	*Control Variable* (CV)	$u(t)$	Sortie analogique vers l'actionneur	Monitorage de la performance
Perturbation	*Disturbance Variable* (DV)	$Q_{sortant}$	Action externe (Fuite/Soutirage)	Indicateur d'état du réseau

CHAPITRE 4 :

Implémentation, Simulation et Validation

4. Implémentation, Simulation et Validation

4.1 TIA Portal (Totally Integrated Automation)

C'est l'environnement d'ingénierie centralisé qu'on a utilisé pour toute la partie automatisme du projet. Concrètement, avec TIA Portal, on a configuré le matériel (un automate CPU 1515-2 PN), on a créé la table des variables globales, et on a programmé la logique de commande en langage Ladder. Cette logique gère les cycles marche/arrêt, l'activation des pompes, le traitement de la mesure via le bloc SCALE, et la gestion des seuils d'alarme [30].

4.1.1Configuration Matérielle :

C'est l'une des premières choses qu'on fait après avoir créé un projet sous TIA Portal. En gros, on construit virtuellement le rack, on y place les modules et les appareils de la périphérie centralisée. Dans notre cas, on a structuré graphiquement l'architecture de l'automate — un CPU 1515-2 PN de la gamme S7-1500 — dans le châssis principal (Châssis_0). Cette étape peut sembler anodine, mais elle est indispensable : elle définit les capacités de calcul pour notre logique de régulation et active les interfaces réseau natives dont on aura besoin pour la suite.

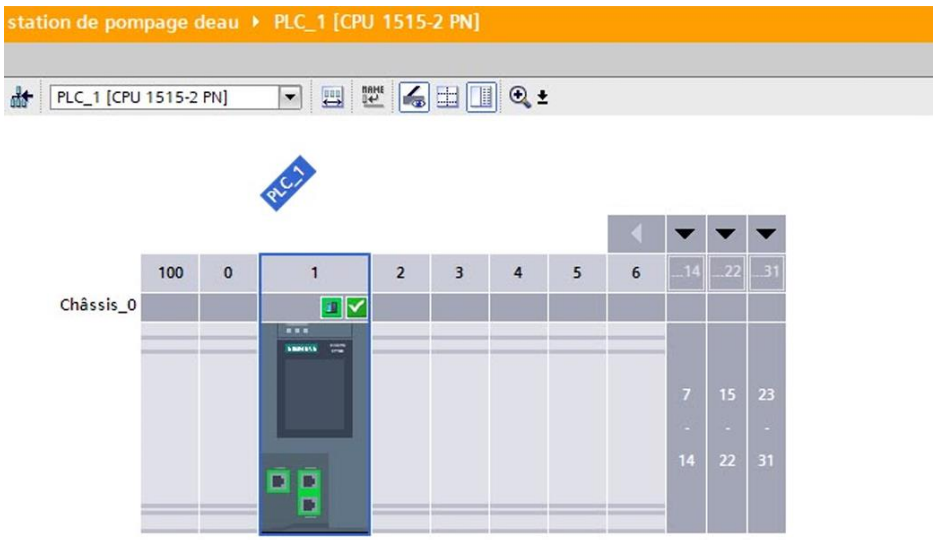


Figure 4-1: Configuration matérielle de l'automate CPU 1515-2 PN sous TIA Portal

4.1.2Configuration Réseau IP :

Une fois que chaque équipement a sa propre adresse réseau, il faut les relier entre eux pour que la communication fonctionne. Dans l'onglet « Appareils et réseaux » de TIA Portal, on interconnecte graphiquement l'automate (PLC_1) et l'IHM (HMI_1) sur le même réseau logique, qu'on appelle PN/IE_1. Cette vue d'ensemble permet de vérifier visuellement la topologie industrielle de l'installation — autrement dit, de confirmer que l'automate virtuel et sa supervision partagent bien le même canal de communication avant de passer à la programmation.

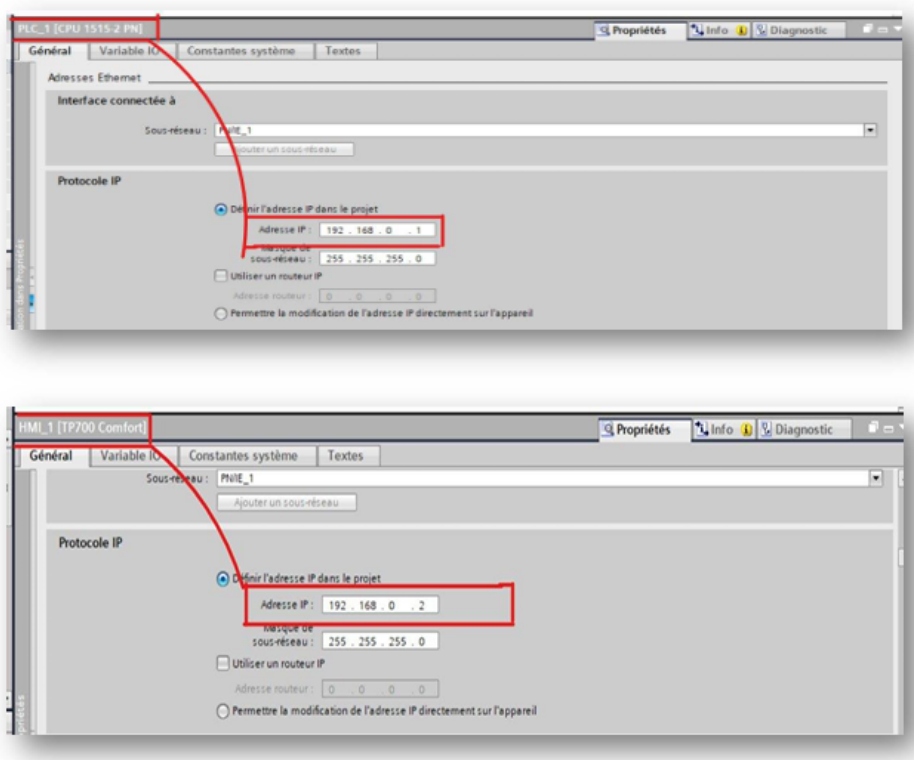


Figure 4-2:Vue réseau de l'interconnexion entre l'automate et l'IHM sous TIA Portal

4.1.3Liaisons de Communication :

Une fois les adresses réseau configurées, il faut relier les équipements pour formaliser l'architecture de communication. Dans l'onglet « Appareils et réseaux » de TIA Portal, on trace une liaison logique (PN/IE_1) entre l'interface PROFINET de l'automate (PLC_1) et celle de l'IHM tactile (HMI_1). Cela valide visuellement la topologie industrielle et confirme que l'automate et sa supervision

partagent bien le même canal de communication pour échanger leurs données en temps réel.

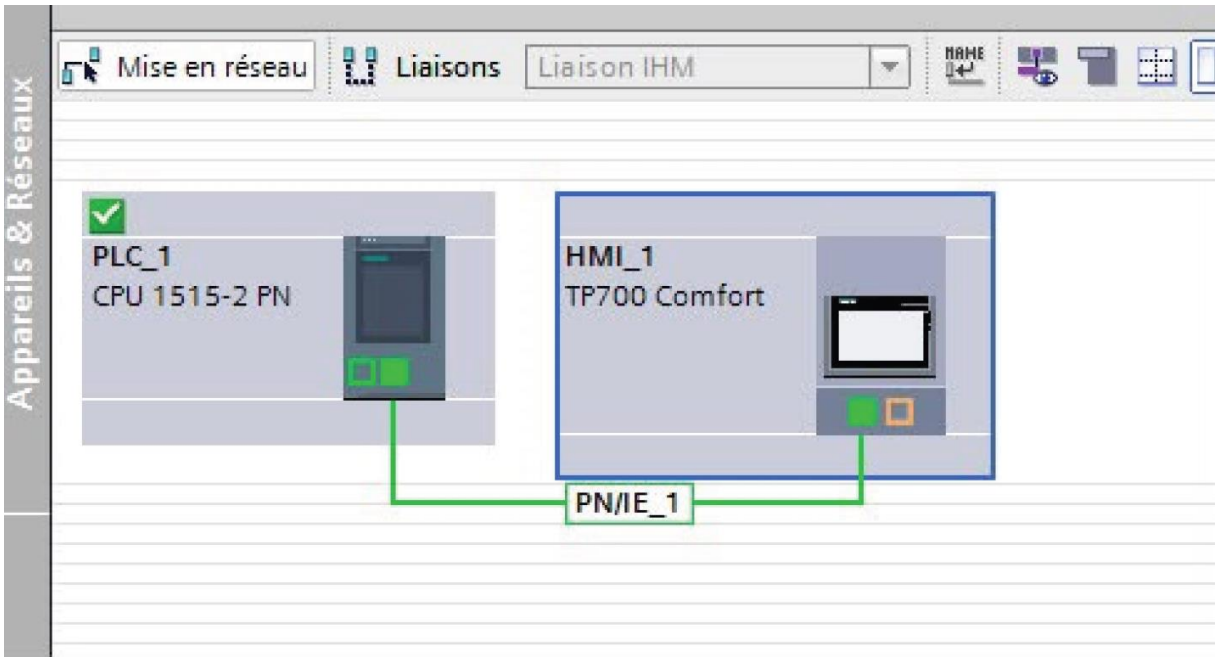
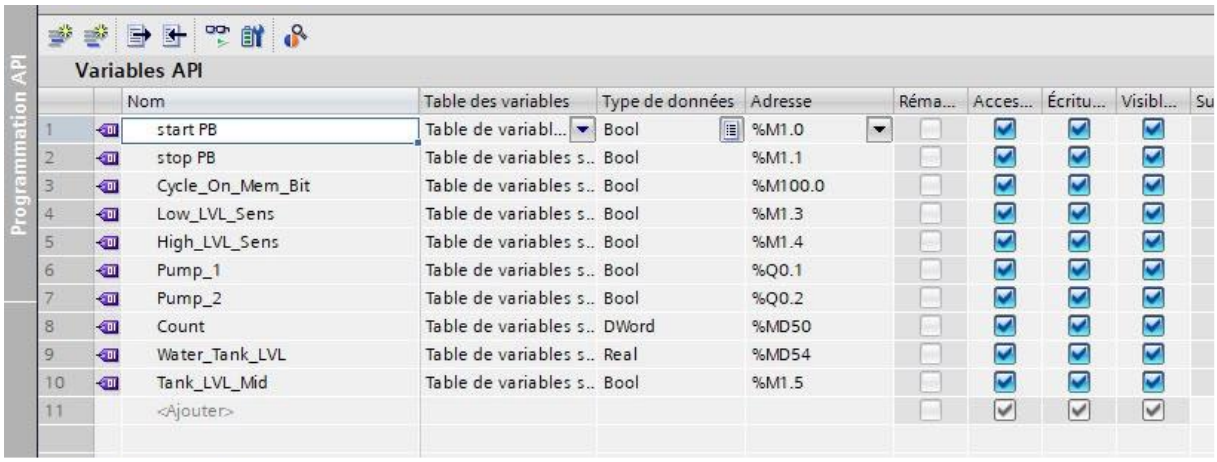


Figure 4-3: Vue de réseau et liaison de communication PN/IE_1 sous TIA Portal

4.1.4Table des Variables :

Avant d'écrire le moindre bloc de programmation, on crée la table des variables de l'automate. Elle liste tous les signaux d'entrée, de sortie et les bits internes nécessaires au fonctionnement : boutons poussoirs start PB et stop PB, commandes des pompes, registres numériques comme Count ou Water_Tank_LVL. Pour chaque nom symbolique, on associe un type de donnée (booléen, entier, réel) et une adresse mémoire absolue (%M, %Q, %MD). Cette table sert de dictionnaire de référence pour le programme Ladder et sécurise les échanges avec la supervision Node-RED.



	Nom	Table des variables	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Su
1	start PB	Table de variabl...	Bool	%M1.0		☑	☑	☑	
2	stop PB	Table de variables s..	Bool	%M1.1		☑	☑	☑	
3	Cycle_On_Mem_Bit	Table de variables s..	Bool	%M100.0		☑	☑	☑	
4	Low_LVL_Sens	Table de variables s..	Bool	%M1.3		☑	☑	☑	
5	High_LVL_Sens	Table de variables s..	Bool	%M1.4		☑	☑	☑	
6	Pump_1	Table de variables s..	Bool	%Q0.1		☑	☑	☑	
7	Pump_2	Table de variables s..	Bool	%Q0.2		☑	☑	☑	
8	Count	Table de variables s..	DWord	%MD50		☑	☑	☑	
9	Water_Tank_LVL	Table de variables s..	Real	%MD54		☑	☑	☑	
10	Tank_LVL_Mid	Table de variables s..	Bool	%M1.5		☑	☑	☑	
11	<Ajouter>					☑	☑	☑	

Figure 4-4:Table des variables globales de l'automate sous TIA Portal

4.2 Siemens S7-PLCSIM Advanced V4.0 :

Ce logiciel a été déployé pour simuler virtuellement le comportement de notre automate de la gamme S7-1500 directement sur PC, sans matériel physique. Contrairement au simulateur standard, la version *Advanced* permet de créer une instance de processeur virtuel dotée d'une adresse IP propre (192.168.0.1) et d'activer les communications réseau. Cette fonctionnalité est indispensable pour ouvrir les canaux d'échange de données nécessaires avec l'environnement extérieur [31].

4.2.1Réseau 1 (Logique de commande) :

Une fois l'architecture matérielle et les variables définies, on attaque la programmation. Le premier réseau (Réseau 1) est dédié au circuit de commande principal. On y implémente une logique d'auto-maintien en langage Ladder pour gérer le démarrage et l'arrêt global du système. Le principe est simple : l'appui sur le bouton poussoir start PB (contact normalement ouvert) active un bit interne de marche générale qui reste verrouillé en mémoire, même après relâchement du bouton. Pour arrêter, l'opérateur actionne le contact normalement fermé du bouton stop PB, ce qui coupe immédiatement le système. La capture d'écran montre ce réseau en mode connecté (visualisation en ligne) : les lignes vertes continues indiquent que le courant virtuel passe et que la commande de marche est bien active.

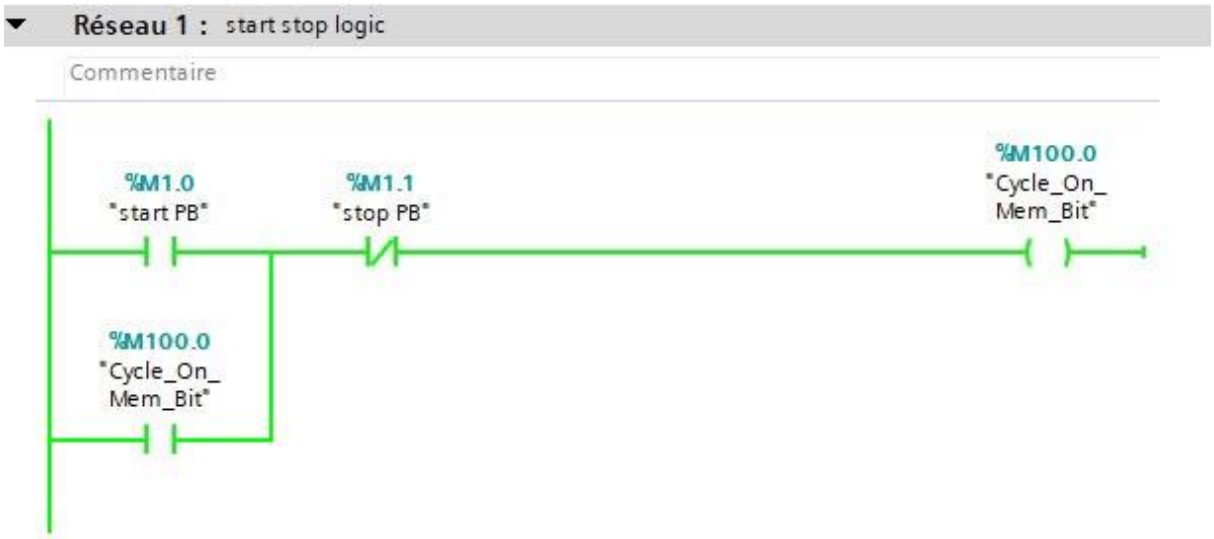


Figure 4-5:Réseau 1 — Logique de commande Marche/Arrêt général en mode connecté (Ligne)

4.2.2Réseau 2 : activation de la pompe de remplissage

Le deuxième réseau gère l'automatisation de la phase de remplissage du réservoir en pilotant la première pompe (Pompe 1). Son démarrage est conditionné à deux critères essentiels : le circuit de commande général doit être activé (bit de marche issu du Réseau 1), et il ne doit pas y avoir d'alerte de niveau haut provenant des capteurs. Une fois ces conditions de sécurité validées, la bobine de sortie de la Pompe 1 s'excite et l'admission d'eau commence. En mode connecté (visualisation en ligne), le surbrimage vert confirme que les conditions d'interverrouillage sont remplies et que la pompe de remplissage est bien en marche.

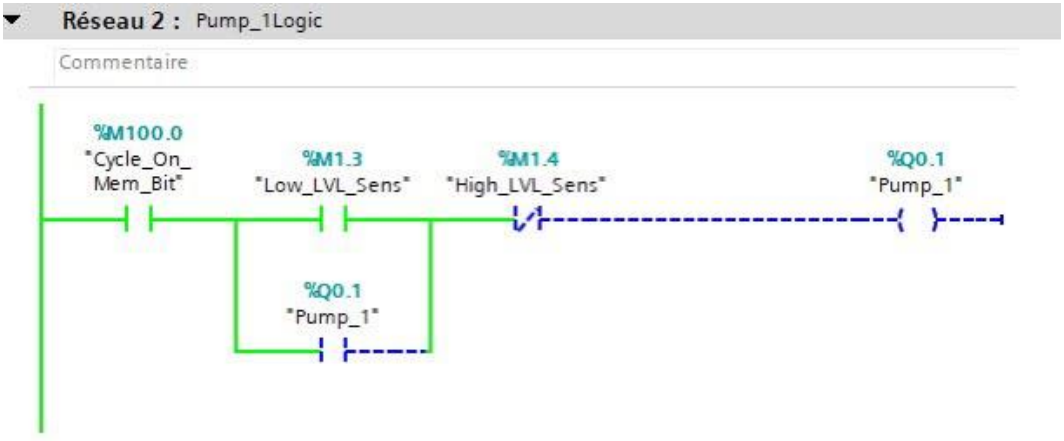


Figure 4-6:Réseau 2 — Logique d'activation de la pompe de remplissage (Pompe 1) en ligne

4.2.3Réseau 3 : activation de la pompe de vidange

Le troisième réseau pilote la phase de vidange du réservoir en activant la seconde pompe (Pompe 2). En langage Ladder, la commande de vidange est soumise à des conditions de sécurité strictes : la marche générale du système doit être validée, et il ne doit pas y avoir d'alerte de niveau bas — histoire d'éviter une marche à sec de l'équipement. Quand les conditions sont réunies, la bobine de sortie de la Pompe 2 s'excite et l'évacuation du fluide commence. En mode connecté, la capture d'écran montre le flux logique (lignes vertes), confirmant que la pompe de vidange est active et que les sécurités sont bien verrouillées pendant la simulation.

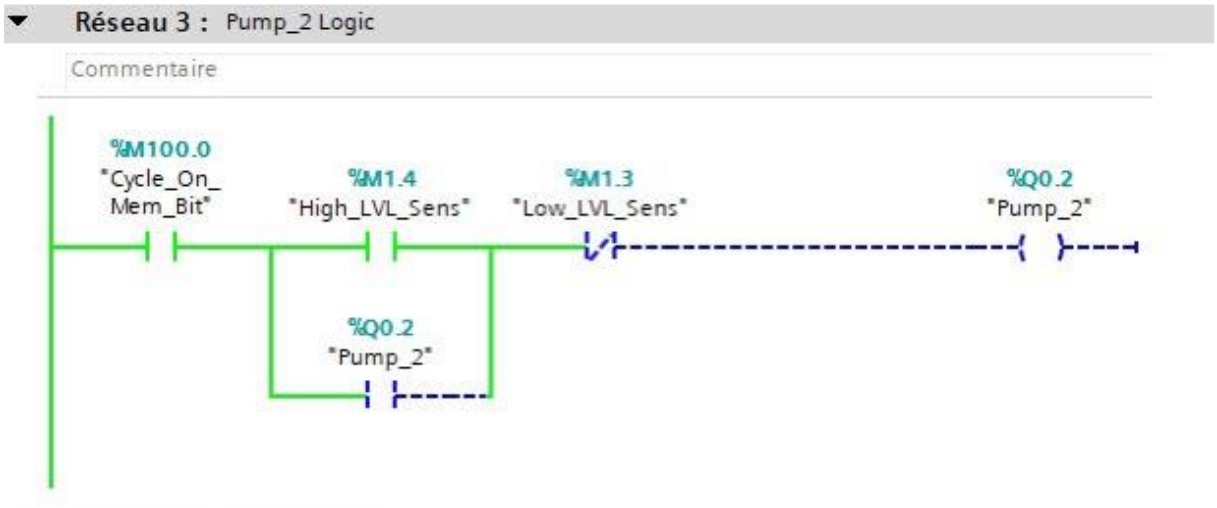


Figure 4-7:Réseau 3 — Logique d'activation de la pompe de vidange (Pompe 2) en ligne

4.2.4Réseau 4 : calcul dynamique du volume d'eau

Le quatrième réseau simule l'évolution du volume d'eau dans le réservoir, puisqu'on n'a pas de capteurs physiques réels. Ça repose sur des blocs mathématiques standard conditionnés par l'état des pompes. Quand la pompe de remplissage (Pompe 1) est active, un bloc d'addition (ADD) incrémente régulièrement un compteur (Count). À l'inverse, quand la pompe de vidange (Pompe 2) tourne, un bloc de soustraction (SUB) le décrémente. En mode

connecté, on voit les calculs en temps réel — la simulation logicielle imite donc la montée ou la baisse du fluide selon les commandes envoyées.

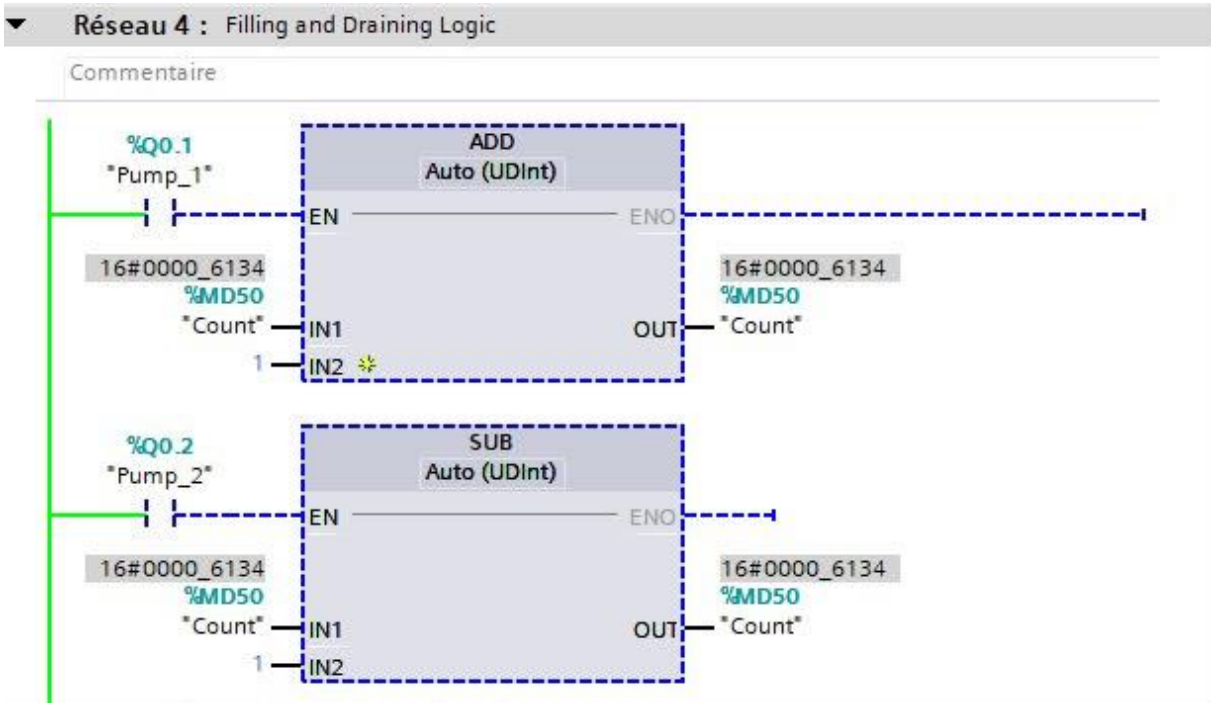


Figure 4-8:Réseau 4 — Blocs d'incrémentation (ADD) et de décrémentation (SUB) de simulation en ligne

4.2.5Réseau 5 : mise à l'échelle de la mesure (SCALE)

Ce réseau est incontournable pour le traitement analogique. La valeur brute (entière) issue du compteur de simulation doit être convertie en une grandeur physique exploitable — ici, le niveau d'eau en pourcentage. Pour ça, on utilise le bloc fonctionnel standard SCALE. Il prend la variable brute en entrée, applique les limites physiques du réservoir (par exemple 0.0 pour vide, 100.0 pour plein), et fait une interpolation linéaire. En mode connecté, la capture d'écran montre le fonctionnement dynamique du bloc : on valide ainsi la conversion instantanée d'une valeur numérique brute en une valeur réelle, prête à être affichée sur les interfaces de supervision.

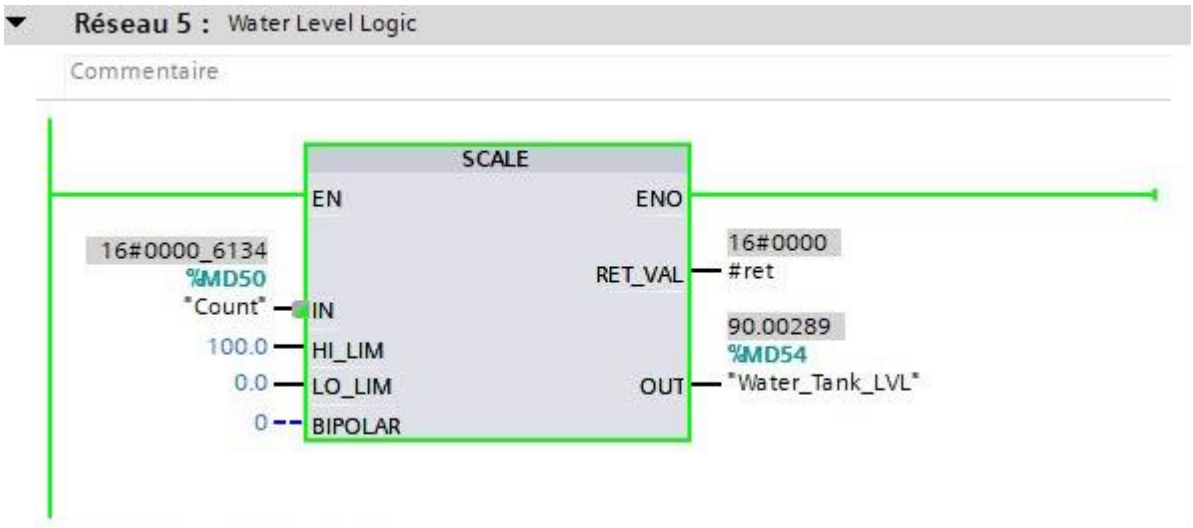


Figure 4-9:Réseau 5 — Traitement de mise à l'échelle via le bloc SCALE en ligne

4.2.6Réseau 6 : supervision des seuils de sécurité (comparateurs)

Ce réseau assure la sécurité de l'installation en surveillant le niveau du réservoir avec des blocs de comparaison. Le principe est simple : on confronte en continu la valeur réelle du niveau (convertie par le bloc SCALE) à des consignes fixes qui définissent les seuils critiques. On utilise des comparateurs logiques « Supérieur ou Égal » (>=) et « Inférieur ou Égal » (<=) pour détecter un niveau trop haut ou trop bas. En mode connecté, on voit en temps réel les bits d'alarme s'activer dès que la mesure franchit ces limites — ce qui déclenche l'arrêt automatique des pompes concernées pour protéger le réservoir.

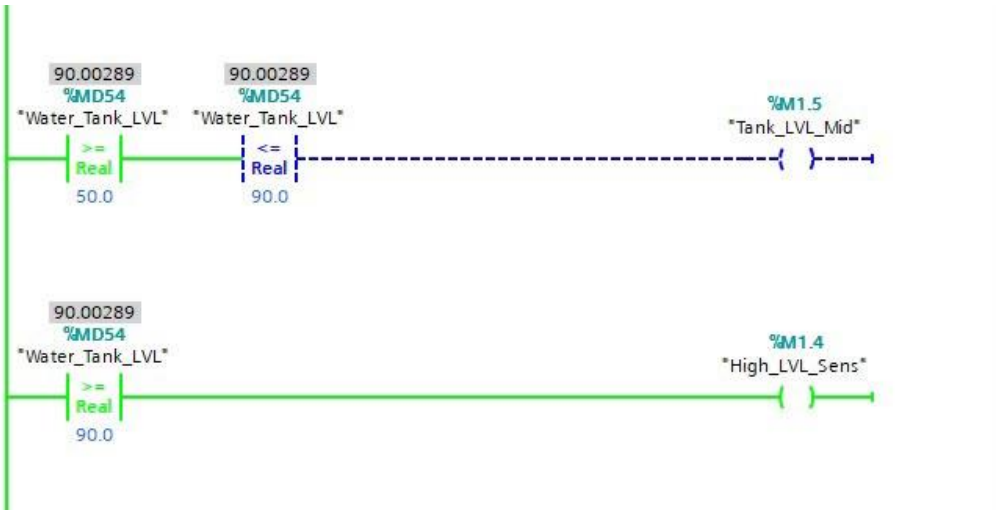


Figure 4-10:Réseau 6 — Configuration des comparateurs de seuils de niveau en ligne

4.2.7Établissement de la connexion matérielle virtuelle

Cette étape marque la transition entre le développement de la logique métier et le déploiement de la simulation. L'écran de confirmation du statut d'accès sous TIA Portal valide que la liaison entre la station de configuration et l'automate virtuel est opérationnelle. Les indicateurs graphiques et les barres de diagnostic passent au vert, confirmant qu'il n'y a pas de divergence entre le programme local et le code chargé dans la CPU cible. Ce statut d'accès sécurisé atteste que le canal est prêt pour la supervision en temps réel et les échanges de données avec l'environnement IoT.

4.3 Environnement de Simulation Virtuelle

4.3.1Basculement vers la simulation virtuelle (S7-PLCSIM Advanced)

Pour se passer d'un automate physique, on utilise le panneau de contrôle S7-PLCSIM Advanced. Il crée une instance logicielle qui émule fidèlement le comportement d'une CPU S7-1500 réelle. Sur l'interface du simulateur, on configure le point d'accès virtuel (PLCSIM Virtual Ethernet Adapter) et on démarre une instance nommée qui passe en statut « Run » (voyant vert). Cette activation est indispensable : elle permet de charger le programme issu de TIA Portal et d'ouvrir le port de communication TCP/IP standard (192.168.0.1). L'automate virtuel devient alors visible et accessible pour des applications tierces comme Node-RED

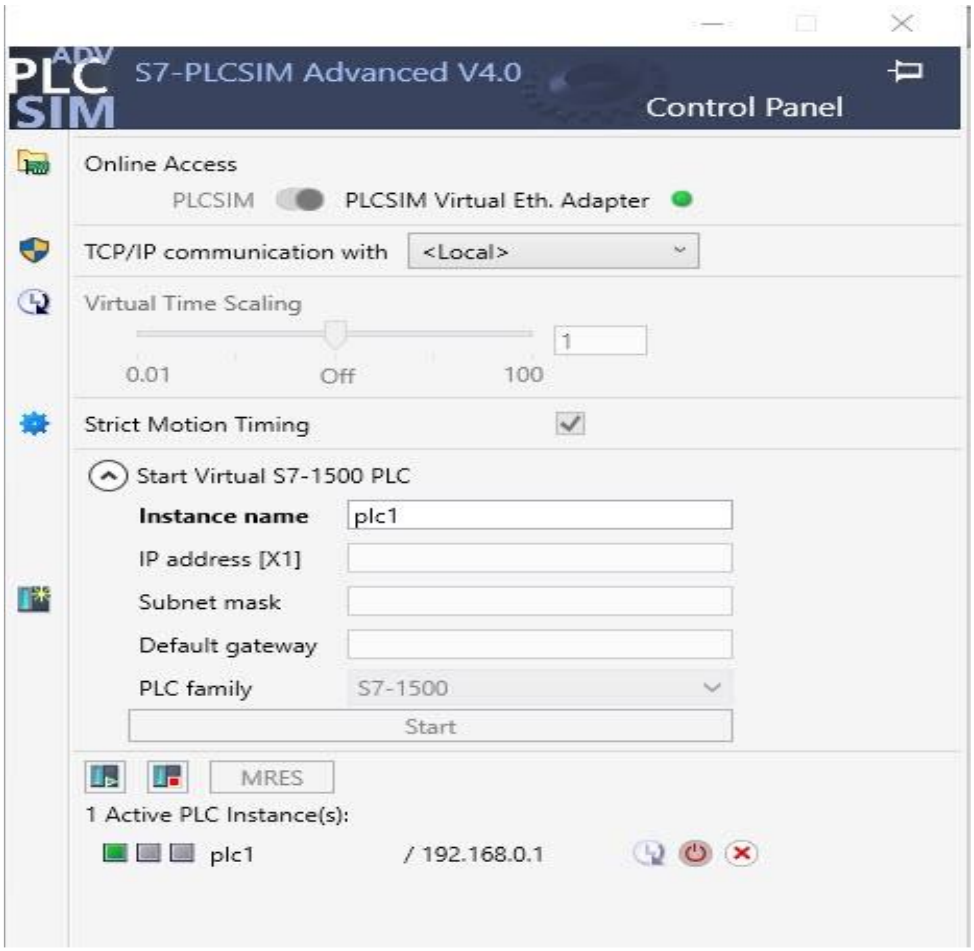


Figure 4-11:Panneau de contrôle S7-PLCSIM Advanced avec l'instance automate active

4.4 Panneau de contrôle S7-PLCSIM Advanced avec l'instance automate active

4.4.1Intégration de l'environnement IoT (Node-RED)

L'intégration de l'Internet des Objets commence par l'ouverture du terminal système pour lancer le serveur Node-RED. On exécute l'environnement Node.js via l'invite de commande, ce qui charge l'ensemble des dépendances du projet — y compris le module d'interface graphique Dashboard. Les logs qui défilent à l'écran confirment que le serveur a bien démarré, qu'il n'y a pas d'erreur de syntaxe, et que le port réseau local par défaut (1880) est ouvert. Ce texte de validation indique que le moteur de programmation par flux est opérationnel et prêt à accueillir la logique d'interconnexion.

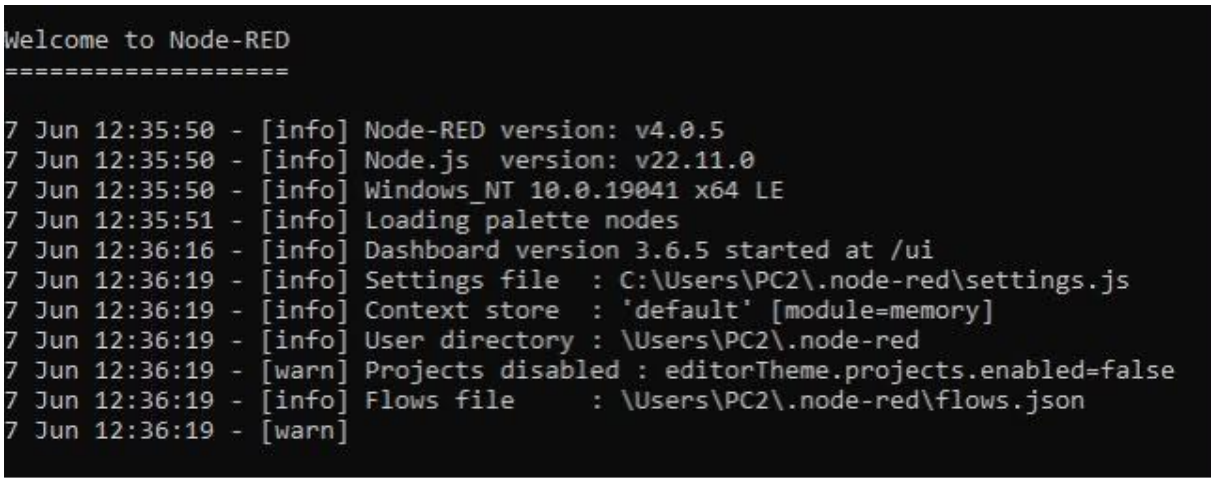


Figure 4-12:Initialisation des services Node-RED et du module Dashboard dans le terminal système

4.4.2Configuration du broker MQTT (EMQX)

L'étape suivante consiste à paramétrer la couche de transport pour la supervision distante. On utilise EMQX, un broker performant adapté aux architectures IoT industrielles. Dans Node-RED, on intègre un nœud de configuration MQTT (MQTT Broker Node) où l'on définit les paramètres d'accès : l'adresse de l'hôte (localhost ou l'IP du serveur), le port standard (1883), et éventuellement les identifiants de sécurité. Ce paramétrage établit un canal de communication de type publication/abonnement (Publish/Subscribe), indispensable pour acheminer les variables du réservoir vers des clients externes ou des tableaux de bord distants.

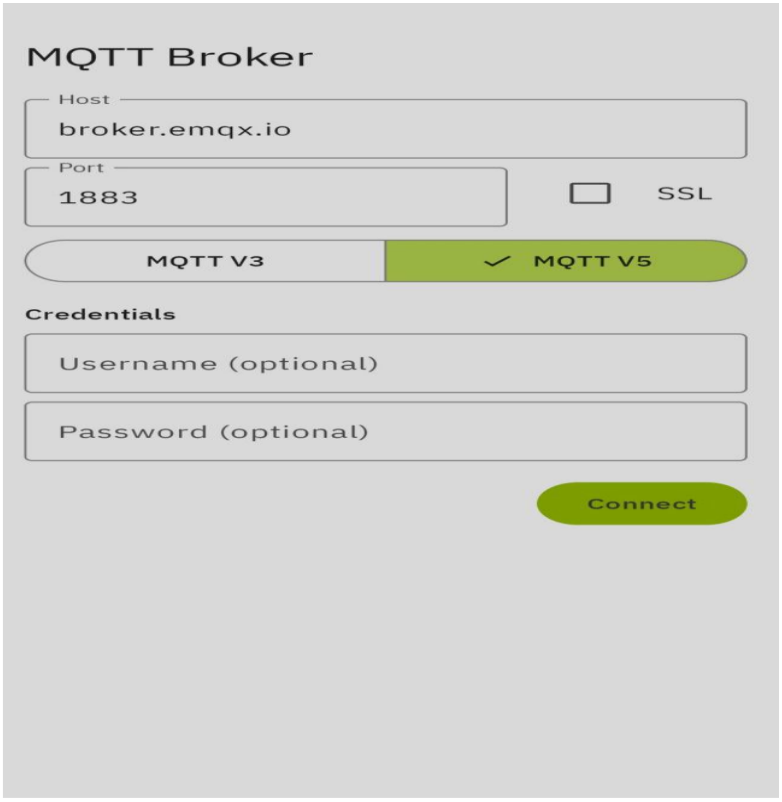


Figure 4-13:Fenêtre de configuration du nœud serveur MQTT (Broker EMQX) sous Node-RED

4.4.3Interconnexion avec l'automate virtuel (nœud S7)

Une fois MQTT configuré, il faut interconnecter Node-RED avec l'automate virtuel via le protocole propriétaire de Siemens. Pour ça, on paramètre le nœud de configuration S7 (S7 Endpoint) dans Node-RED. On y renseigne l'adresse IP fixe configurée sur PLCSIM Advanced (192.168.0.1), le numéro de rack (0) et l'emplacement de la CPU (Slot 1). Ce paramétrage permet à la passerelle IoT d'ouvrir un canal de communication direct et stable pour interroger et modifier la mémoire de l'automate en temps réel.

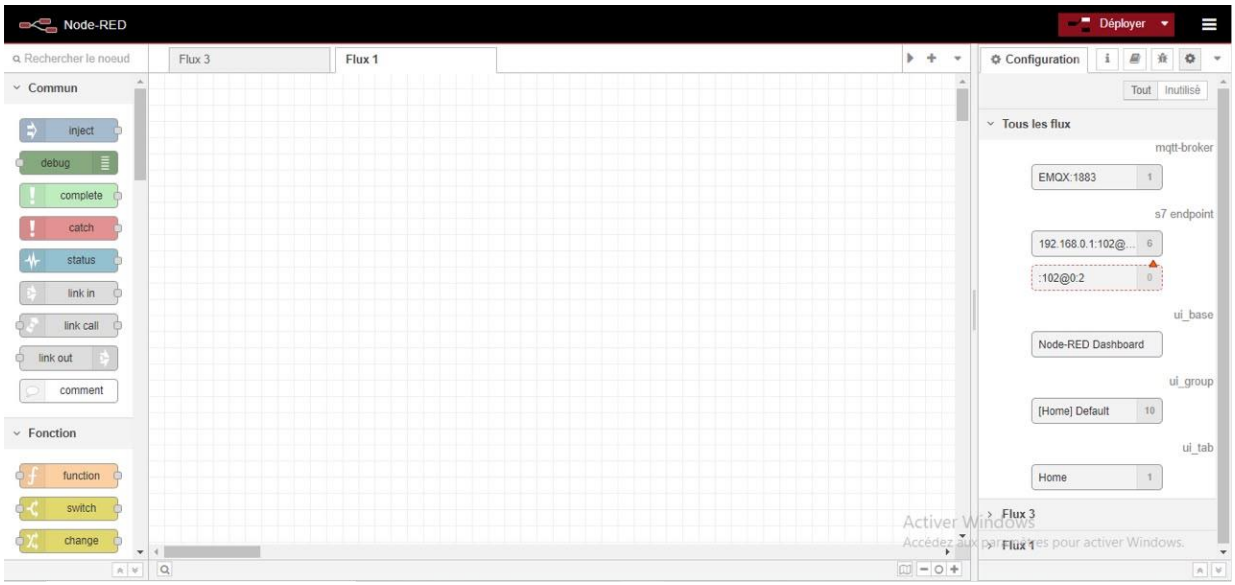


Figure 4-14:Fenêtre de paramétrage de la liaison S7 et définition du point d'accès (Endpoint) sous Node-RED

4.4.4Configuration de la table de correspondance (Mapping) sous Node-RED

Après avoir établi le point d'accès réseau vers l'automate, on configure une table de correspondance (mapping) dans Node-RED. Celle-ci relie les adresses mémoire de la CPU à des variables exploitables par les flux. Dans la première partie, on importe et définit les variables logiques et de commande : chaque ligne associe un nom de variable (boutons poussoirs, retours d'état des pompes) à son adresse absolue dans l'automate (par exemple %M0.0 ou %M0.1) et à son type (booléen). Node-RED sait ainsi exactement où lire et écrire les informations binaires nécessaires au pilotage du cycle de régulation.

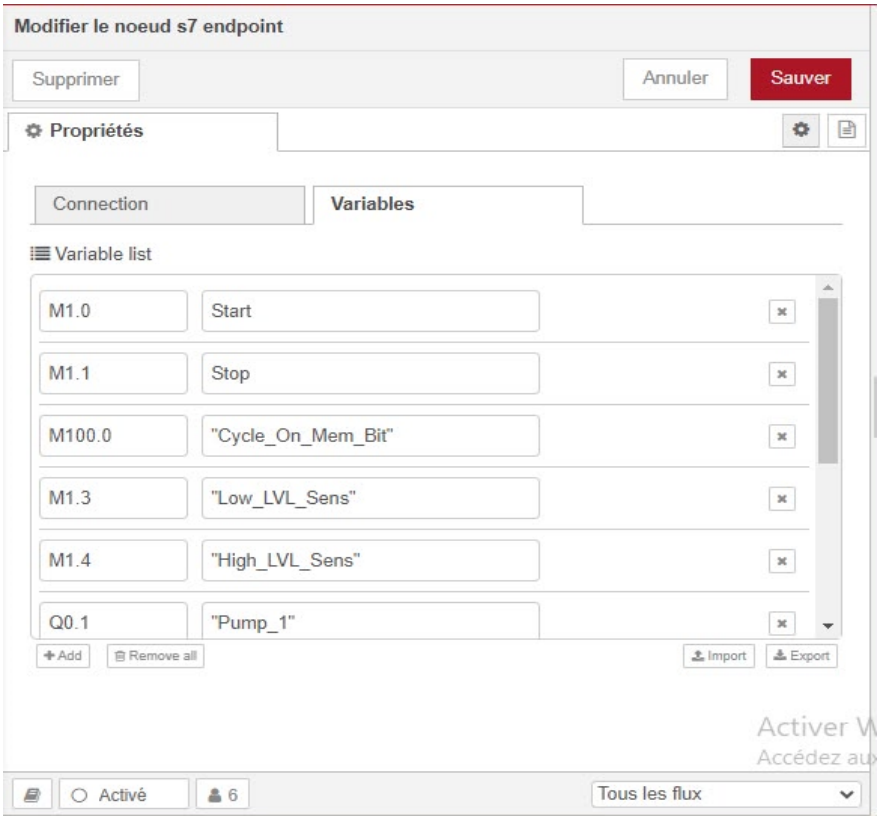


Figure 4-15:Table de mapping S7 (Partie 1) — Importation des variables de commande initiales

4.4.5Mapping (seconde partie) : variables numériques et analogiques

Dans la seconde partie de la table de correspondance, on importe le reste des variables de données : les temporisations, les compteurs de volume, et surtout le niveau d'eau réel du réservoir issu du bloc SCALE (Water_Tank_LVL). Contrairement aux variables binaires, ces grandeurs sont associées à des adresses mémoire de type mots ou doubles mots (comme %MD) et à des types adaptés (Entier ou Réel). Ce mapping exhaustif achève la configuration du dictionnaire de communication, permettant un échange bidirectionnel complet entre l'automate virtuel et les flux IoT.

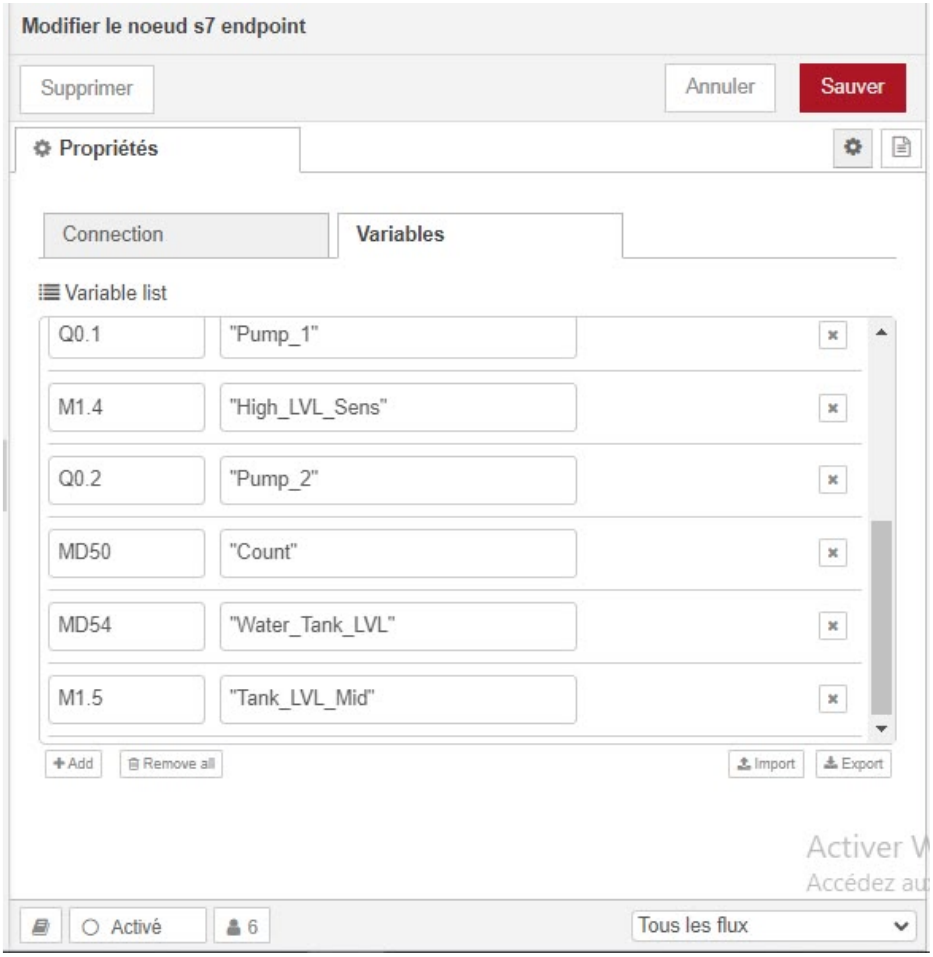


Figure 4-16:Table de mapping S7 (Partie 2) — Importation du reste des variables numériques et de données

4.4.6Configuration du nœud d'écriture S7 (S7 Out Node)

Une fois le mapping finalisé, on paramètre le comportement des nœuds d'échange selon le sens des données. Ici, on se concentre sur le nœud d'écriture S7 (S7 Out Node), qui transmet les ordres de l'opérateur vers la mémoire de l'automate. Dans sa fenêtre de configuration, on sélectionne le mode d'accès approprié et on cible la variable de commande souhaitée (par exemple le pilotage des pompes). Ce réglage est indispensable pour que chaque action sur l'interface graphique (clic sur un bouton) soit immédiatement traduite en une écriture réseau, modifiant instantanément l'état logique des variables dans l'automate.

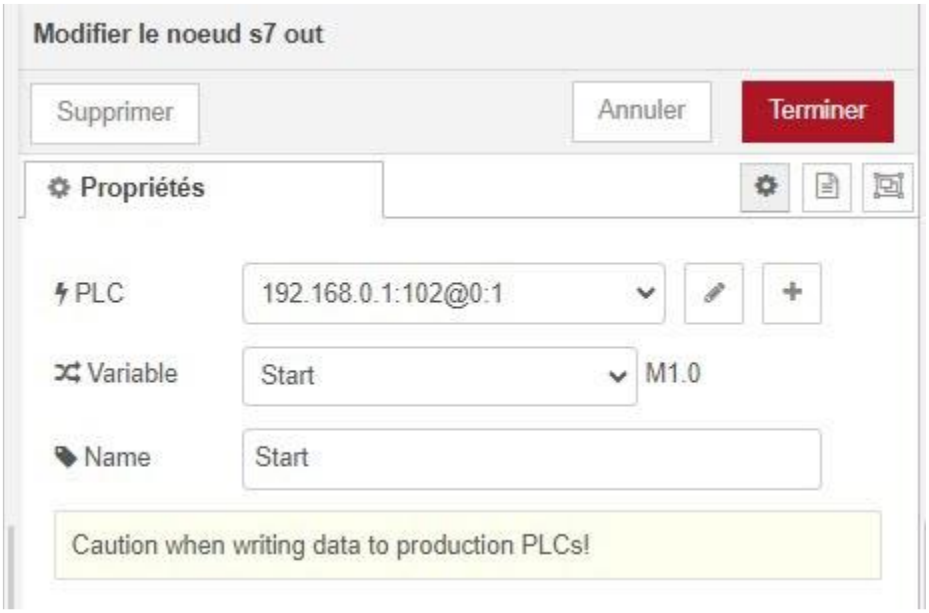


Figure 4-17:Fenêtre de configuration du nœud d'écriture S7 pour l'envoi des commandes

4.4.7Architecture globale des flux Node-RED

Après avoir configuré les liaisons, le mapping et les serveurs, cette étape présente l'aboutissement de la programmation sous Node-RED. Le diagramme logique du « Flux 3 » montre l'interconnexion bidirectionnelle complète de l'application. À gauche, les nœuds d'entrée S7 collectent en temps réel les variables de l'automate virtuel (niveau du réservoir, état des pompes). Au centre, des nœuds de fonction traitent et structurent ces messages. À droite, les flux se séparent pour alimenter à la fois les composants graphiques du Dashboard local et les nœuds de publication MQTT vers le broker EMQX. Cette vue d'ensemble valide la cohérence de notre passerelle IoT et sa capacité à orchestrer les échanges entre l'automate et les interfaces de supervision.

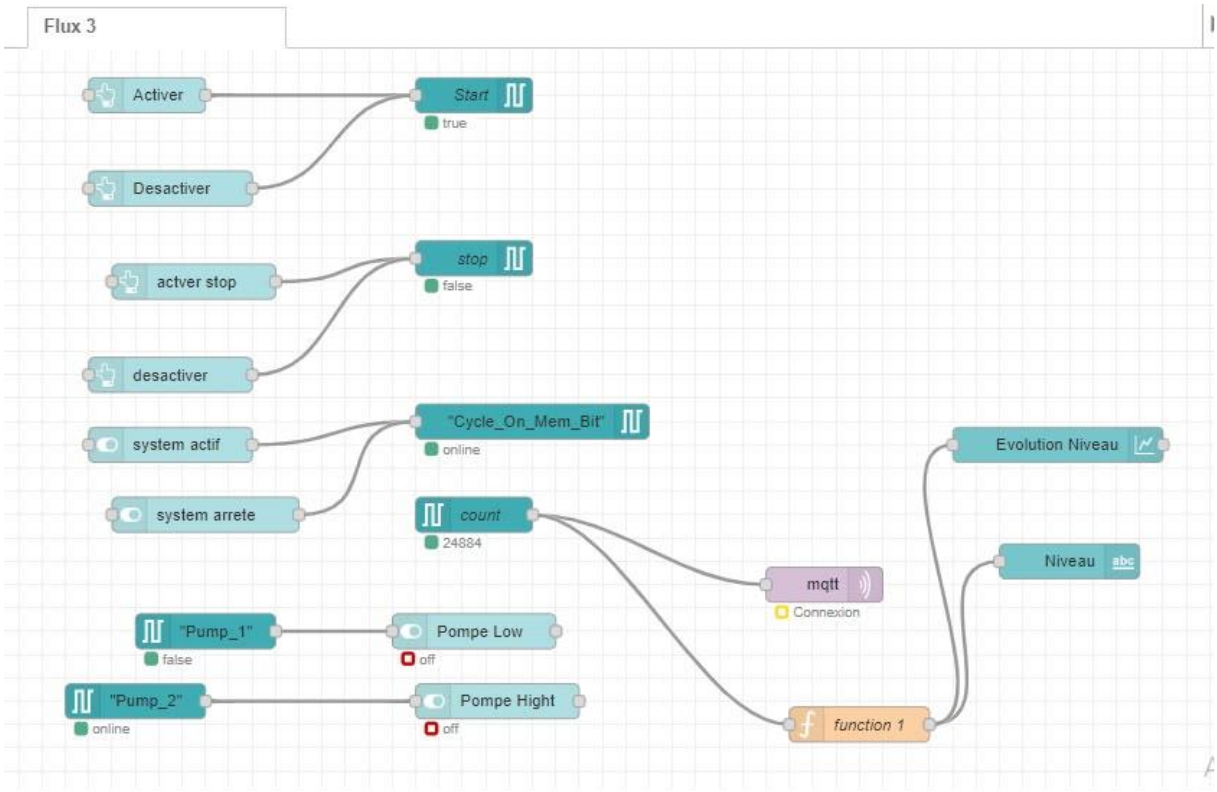


Figure 4-18:Architecture logique globale du Flux 3 interconnecté sous Node-RED

4.5 Node-RED et Node-RED Dashboard :

Node-RED est un outil de programmation événementielle basé sur des flux, exécuté sur Node.js, utilisé ici comme passerelle de communication (middleware). Il assure la liaison en lisant et écrivant en temps réel les variables de l'automate virtuel via le protocole S7. L'extension *Dashboard* associée a permis de concevoir l'interface graphique finale (IHM locale de supervision), offrant à l'opérateur une visualisation dynamique de l'évolution du niveau du réservoir et le contrôle des commandes.

4.5.1Supervision locale : HMI sous TIA Portal

Cette étape présente la vue de conduite HMI développée sous TIA Portal. L'écran du pupitre tactile SIMATIC TP700 Comfort offre une interface ergonomique, claire et industrielle pour piloter l'installation en temps réel. On y trouve une animation dynamique du réservoir qui se remplit ou se vide selon les calculs de simulation, des boutons virtuels (Marche/Arrêt général), des voyants d'état des pompes (Pompe 1 et Pompe 2), et des indicateurs d'alarme pour les seuils haut et bas. Cet écran constitue l'outil principal d'exploitation sur site, garantissant un contrôle direct et sécurisé du procédé de régulation.

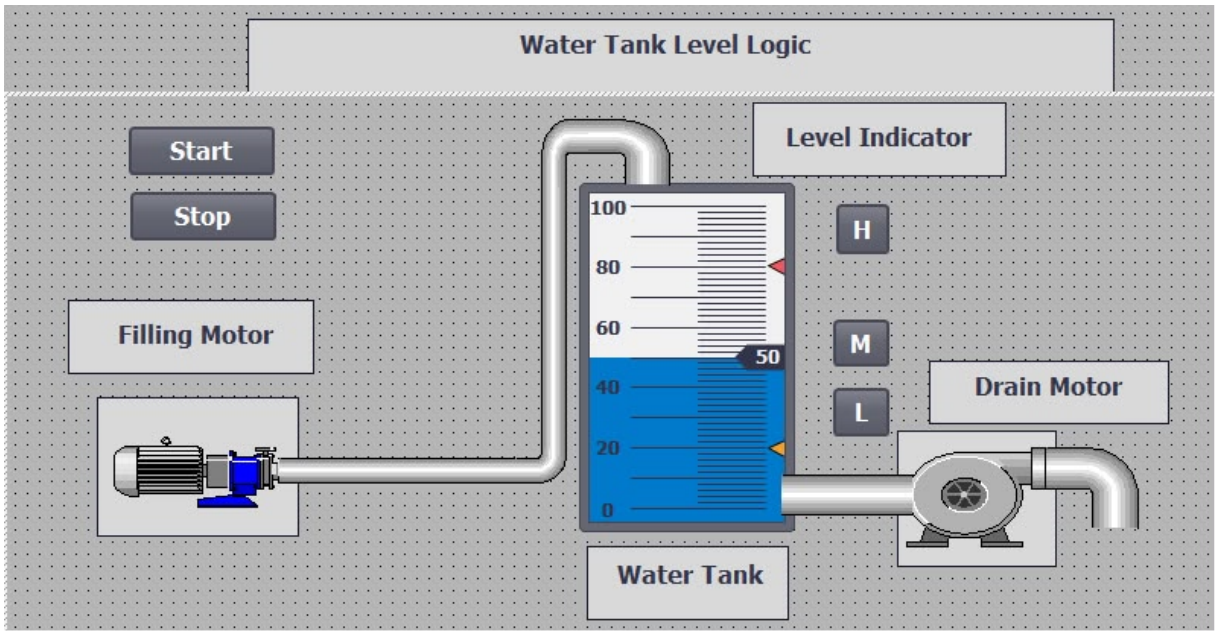


Figure 4-19:Vue de conduite et écran de supervision natif sur le pupitre HMI TP700 Comfort

4.5.2Interface opérateur initiale (Dashboard Node-RED)

Cette étape matérialise l'aboutissement de la chaîne IoT à travers l'interface web globale. L'écran initial du tableau de bord (Dashboard Node-RED) se présente à l'arrêt, avant le lancement du cycle de régulation. Les indicateurs visuels (jauges, graphiques de tendance) sont à zéro ou figés sur leurs valeurs par défaut. Le réservoir virtuel est vide (niveau 0 %), et les voyants d'état signalent que les pompes de remplissage et de vidange sont à l'arrêt. Cette vue initiale confirme que les styles graphiques sont chargés et que le tableau de bord est prêt à recevoir les premières données dynamiques dès l'activation du système.

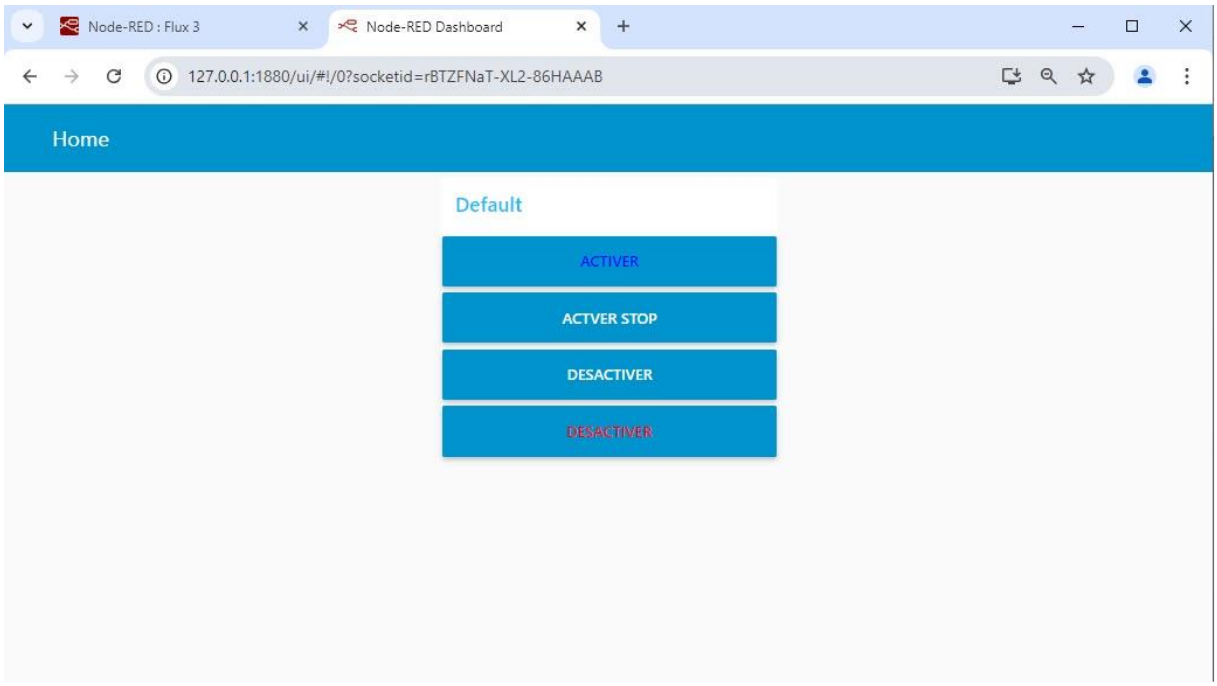


Figure 4-20:Interface opérateur initiale — Dashboard Node-RED au repos

4.5.3Phase transitoire de remplissage (Dashboard Node-RED)

Cette étape montre les résultats dynamiques du système lors de la phase de remplissage. Sur la capture d'écran du Dashboard, la courbe d'évolution (Trend) affiche une progression linéaire et croissante du niveau d'eau. La jauge graphique suit fidèlement cette montée jusqu'au seuil critique haut préconfiguré. À cet instant, le système réagit immédiatement : le voyant d'alarme de niveau haut s'allume, et la pompe de remplissage (Pompe 1) s'arrête automatiquement pour éviter tout débordement.

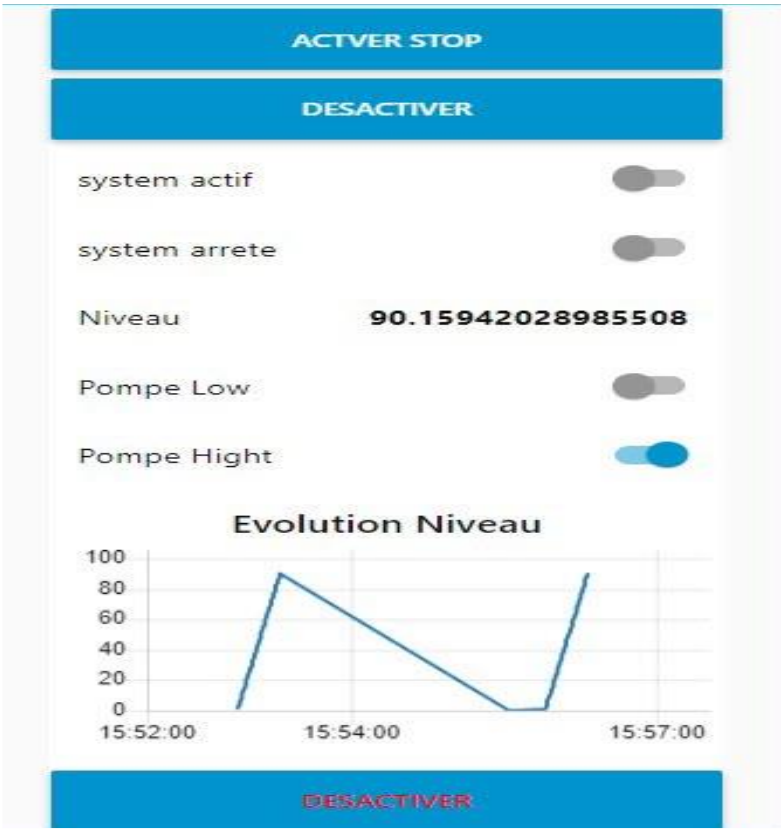


Figure 4-21:Courbe dynamique du niveau d'eau atteignant le seuil haut lors de la phase de remplissage

4.5.4Phase de vidange (Dashboard Node-RED)

Cette étape illustre le comportement du système lors de la vidange. Sur la capture d'écran du Dashboard, la courbe de tendance (Trend) affiche une décroissance linéaire et régulière du volume d'eau, traduisant l'action de la pompe d'évacuation (Pompe 2). Dès que le niveau atteint le seuil bas préconfiguré, la logique de comparaison dans l'automate coupe instantanément la pompe de vidange pour éviter une marche à sec. Simultanément, le voyant d'alarme de niveau bas s'active sur l'interface, validant ainsi la robustesse du contrôle-commande et la parfaite synchronisation des échanges de données en temps réel.

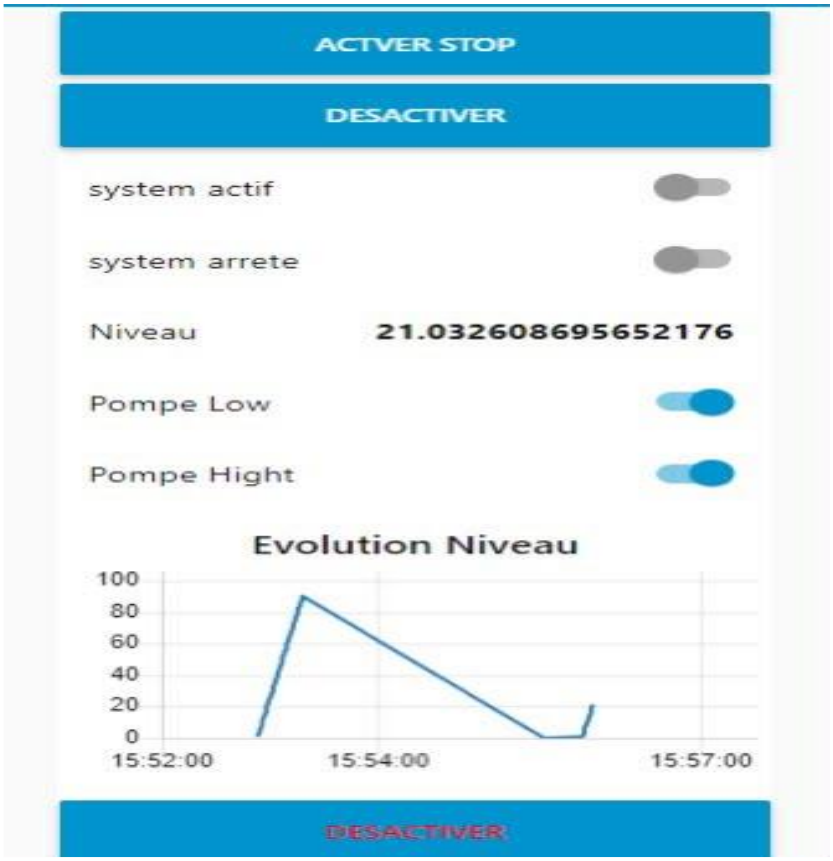


Figure 4-22:Courbe dynamique du niveau d'eau et inversion de cycle à l'atteinte du seuil bas (Phase de vidange)

LES RÉFÉRENCES

[1] "1999, Internet des Objets (IoT) – Théorie de".

[2] "allemande., Allemagne (2011). Industrie 4.0. Initiative gouvernementale".

[3] "Industriel, Industrial Internet Consortium (IIC) – Internet".

[4] "intelligentes), Smart Factories (Usines".

[5] "(IoS), Internet des Services".

[6] "(CPS), Systèmes Cyber-Physiques".

[7] "(AWS), Amazon Web Services".

[8] "MQTT, ISO/IEC 20922 – Norme".

[9] "OSI, Modèle".

[10] "WiFi, IEEE 802.11 – Norme".

[11] "Ethernet, IEEE 802.3 – Norme".

[12] "1215C), Siemens AG – SIMATIC S7-1200 (CPU".

[13] "PROFINET, Siemens AG –".

[14] "PROFIBUS, Siemens AG –".

[15] "PROFIenergy, Siemens AG –".

[16] "disjoncteurs, EN 60934 – Norme".

[17] "ferroviaire, NF F 62-001 – Norme".

[18] "API, IEC 61131-3 – Norme des langages de programmation".

[19] "Control), Régulation Tout-ou-Rien (On/Off".

[20] morte), Hystérésis (Bande.

[21] "(Proportionnel-Intégral-Dérivé), Régulateur PID".

[22] "Anti-windup".

[23] "hydrostatique, Loi de Pascal – Capteur de pression".

[24] "(ΔP), Capteur de pression différentielle".

[25] "ultrasons, Capteur à".

[26] radar, Capteur.

[27] "capacitif, Capteur".

[28] "(Solénoïde), Électrovanne".

[29] variable, Pompe centrifuge à vitesse.

[30] "Automation), Siemens AG – TIA Portal (Totally Integrated".

[31] "générale), Siemens AG – S7-PLCSIM Advanced V4.0 (Définition".