الجمهـــوريــــة الجزائـــريــــة الديمقـــراطيــــة الشــعبيــــة
République Algérienne Démocratique et Populaire
وزارة التـــعـلــيـــــم العـــــــالــــــي والبـــحــــــث العــلــمـــي
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 و الأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de sécurité industrielle et environnement

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Maintenance des Automatismes et de l'instrumentation industriels

Thème

L'utilisation du Webserver des automates S7-1200 pour le control à distance (IoT)

Présenté et soutenu publiquement par :

Sam Bouafia Toufik

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
Mekki Ibrahim Elkhalil	MCA	IMSI-Univ.D'Oran2	**Président**
Aissani Nassima	MCA	IMSI-Univ.D'Oran2	**Encadreur**
Belkhoudja Leila	MCB	IMSI-Univ.D'Oran2	**Examinateur**
Gaham Mehdi	MR	Centre Algérien de Développement des Technologies Avancées	**Examinateur invité**

Année 2021/2022

Dedication

To my Family for the lifelong encouragement, support and love

I dedicate this work

Acknowledgment

I would like to thank Mrs Aissani Nassima for her trust and support during the project life cycle.

Abstract

Historically, Industrial Automation and Control Systems (IACS) were largely isolated from conventional digital networks such as enterprise ICT environments.

The adoption and deployment of ‘Internet of Things’ (IoT) technologies is leading to architectural changes to IACS, including greater connectivity to industrial systems. The aim of this project is to realize an IoT based SCADA system.

Table of Contents

Table of Abbreviations

API: Application Programming Interface

ADC: Analog to Degital Converter

AWS: Amazon web service

CSS : Cascading Style Sheets

CERN: the European Organization for Nuclear Research

CPS: Cyber Physical system

CPU: central processing unit

CPE: Customer premises equipment

DCS: Distributed Control System

DAC: Digital to Analog Converter

EN: European Standards

ERP: Enterprise resource planning

GSM : Global System for Mobile Communications

GPRS: General Packet Radio Service

HTML : HyperText Markup Language

HTTP: HyperText Transfer Protocol

HMI: Human Machine Interface

HSPA : High Speed Packet Access

IaaS: Infrastructure as a service

IACS: Industrial Automation and Control Systems

ICS: Industrial Control System

IIoT: Industrial Internet of Things

IoS: Internet of Service

ISO: International Organization for Standardization

ISV: Independent Software Vendor

IEC : International Electrotechnical Commission

IP : Internet Protocol

IEEE : Institute of Electrical and Electronics Engineers

JSON: JavaScript Object Notation, pronounced

LAN: Local Area Network

LTE: Long Term Evolution

LED: light Emitting Diode

M2M: machine to machine

MQTT: Message Queuing Telemetry Transport

MES: Manufacturing Execution system

MNO: Mobile Network Operators

OT: Operational Technology

OASIS: The Organization for the Advancement of Structured Information Standards

OS: Operating System

OT-IT: Operational Technology – Information Technology

OPC UA: The OPC Unified Architecture

PAN: Personal Area Network

PaaS : platform as a service

RFID : Radio frequenc y identification

SaaS: Software as a Service

TCP : Transmission control protocol

TLS: Transport Layer Security

URLLC: Ultra-Reliable Low Latency Communications

UMTS: Universal Mobile Telecommunications Service

UDP: User Datagram Protocol

WAN: Wide Area Network

WSN: Wireless Sensor Network

Table of Figures

List of Tables

Chapter 1: Introduction

1.1 Thesis introduction

Industry 4.0 is making a shift in the way companies manufacture, improve and distribute their products. Many organizations are integrating new technologies, including Internet of things and cloud computing and analytics and AI. Manufacturers are exploring new ways to leverage IoT. Its potential, advantages and with expectations to enhance the overall throughput of their existing process automation, by overcoming existing challenges and also integrating the entire value chain of the organization.

Democratization of data in a much federated enterprise is the new age trend of digitization. Currently in large manufacturing and process oriented enterprises like the Steel, Power, Refineries, the most often asked questions are [71]:

Will IoT replace Supervisory Control and Data Acquisition (SCADA) in a new era for manufacturing?

How can IoT help me in leveraging existing SCADA?

I already have process automation, have implemented latest technologies to enhance my existing SCADA, do I still need IoT?

1.1.1 Objective

This project aims to illustrate the concept of Industrial internet of things through a simple design and implementation of an IoT based SCADA system:

The requirements of an IoT based SCADA system:

1. Field instrumentation (sensor and actuators in my case PCT50)
2. Programmable logic controller (SIMATIC S7-1200)
3. Human Machine Interface
4. Network (Ethernet – Wi-Fi – 4G LTE)
5. Cloud Service (Ubidots)
6. Remote monitoring and control
7. Data analytics – reporting – predictive maintenance

The design should include all the necessary steps taken by an industrial automation and instrumentation engineer to implement the IoT based SCADA system, the steps are listed below:

1- Studying the process to be automated
2- Developing an algorithm for the system to operate with
3- Connecting and placing the selected items properly in the system to meet optimum system performance
4- Developing a program using TIA portal V16 based on the suggested algorithm
5- Developing a program code for the user defined page using HTML and CSS
6- Using Node Red to design a dashboard to control and monitor the system using PLC web server
7- Using Node Red to interact with Ubidots cloud Service (data publishing and Subscribing)
8- Remote monitoring and control of the system from any place in the world
9- Data analysis – reporting - storage

1.2 The Internet of Things

Internet of Things Almost every area, device, sensor, software, etc are connected to each other. The ability to access these devices through a Smartphone or through a computer is called IoT. These devices are accessed from a distance. For example, an Air Conditioner’s sensor can gather the data regarding the outside temperatures, and accordingly adjust its temperature to increase or decrease it with respect to the outside climate. Similarly, your refrigerators can also adjust their temperature accordingly. This is how devices can interact with a network.

Imagine a world where billions of objects can sense, communicate and share information, all interconnected over public or private Internet Protocol (IP) networks. These interconnected objects have data regularly collected, analyzed and used to initiate action, providing a wealth of intelligence for planning, management and decision making. This is the world of the Internet of Things (IOT) [1].

Just like Internet has changed the way we work & communicate with each other, by connecting us through the World Wide Web (internet), IoT also aims to take this connectivity to another level by connecting multiple devices at a time to the internet thereby facilitating man to machine and machine to machine interactions.

People who came up with this idea, have also realized that this IoT ecosystem is not limited to a particular field but has business applications in areas of home automation, vehicle automation, factory line automation, medical, retail, healthcare and more[2].

1.3 A brief History on the internet of things

The concept of connected devices itself dates back to 1832 when the first electromagnetic telegraph was designed. The telegraph enabled direct communication between two machines through the transfer of electrical signals. However, the true IoT history started with the invention of the internet a very essential component in the late 1960s, which then developed rapidly over the next decades [3].

The 1980s

This might be hard to believe, but the first connected device was a Coca-Cola vending machine situated at the Carnegie Melon University and operated by local programmers. They integrated micro-switches into the machine and used an early form of the internet to see if the cooling device was keeping the drinks cold enough and if there were available Coke cans. This invention fostered further studies in the field and the development of interconnected machines all over the world [3].

The 1990s

In 1990, John Romkey connected a toaster to the internet for the very first time with a TCP/IP protocol. One year later, University of Cambridge scientists came up with the idea to use the first web camera prototype to monitor the amount of coffee available in their local computer lab's coffee pot. They programmed the webcam to take pictures of the coffee pot three times per minute, then send the images to local computers, thus allowing everyone to see if there was coffee available.

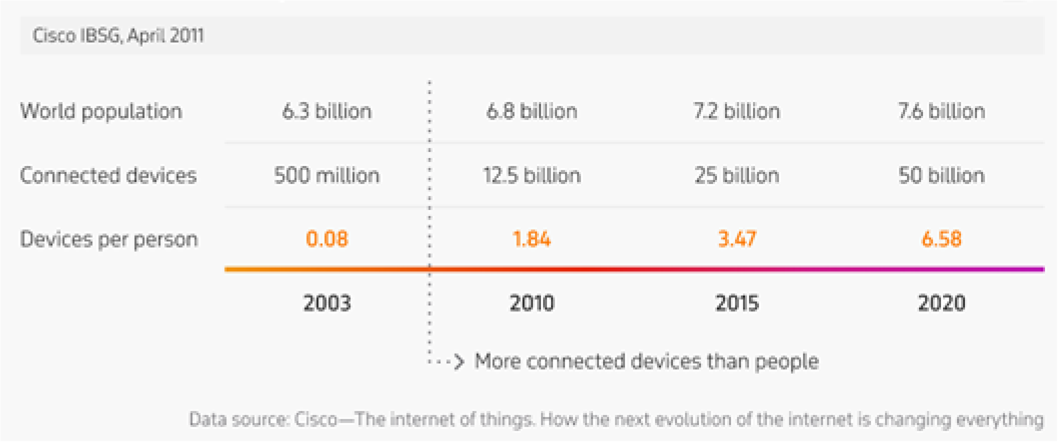
The year 1999 was easily one of the most significant for the IoT history, as Kevin Ashton coined the term "the internet of things." A visionary technologist, Ashton was giving a presentation for Procter & Gamble where he described IoT as a technology that connected several devices with the help of RFID tags for supply chain management. He specifically used the word "internet" in the title of his presentation in order to draw the audience's attention since the internet was just becoming a big deal that time. While his idea of RFID-based device connectivity differs from today's IP based IoT, Ashton's breakthrough played an essential role in the internet of things history and technological development overall [3].

The 2000s

At the beginning of the 21st century, the term "internet of things" came into widespread use by the media, with outlets like The Guardian, Forbes, and the Boston Globe making mention of it. Interest in the IoT technology was steadily increasing, which led to the 1st International Conference on the Internet of Things held in Switzerland in 2008, where participants from 23 countries discussed RFID, short-range wireless communications, and sensor networks.

Moreover, several major developments fostered the IoT evolution. One was a refrigerator connected to the internet that was introduced by LG Electronics in 2000, allowing its users to shop online and make video calls. Another essential development was a small rabbit-shaped robot named Nabaztag created in 2005 that was capable of telling the latest news, weather forecast, and stock market changes.

Even back then the number of interconnected devices surpassed that of people on Earth, according to Cisco [3].



Cisco IBSG, April 2011

World population	6.3 billion	6.8 billion	7.2 billion	7.6 billion
Connected devices	500 million	12.5 billion	25 billion	50 billion
Devices per person	0.08	1.84	3.47	6.58
	2003	2010	2015	2020

More connected devices than people

Data source: Cisco—The internet of things. How the next evolution of the internet is changing everything

Table1 : Connected Devices by 2020

The 2010s

The IoT boom was supported by its addition to the Gartner Hype Cycle for emerging technologies in 2011.In the same year, IPv6 a network layer protocol that is central to IoT was launched publicly.

Since then, interconnected devices have become widespread and commonplace in our everyday lives. Global tech giants like Apple, Samsung, Google, Cisco, and General Motors are focusing their efforts on the production of IoT sensors and devices from interconnected thermostats and smart glasses to self-driving cars. IoT has found its way into almost every industry: manufacturing, healthcare, transportation, oil & energy, agriculture, retail, and many more. This dramatic shift has us convinced that the IoT revolution is right here, right now.

As of today, IoT platforms maintain a strong hold on their position among the top trends in this year's Gartner Hype Cycle, along with virtual assistants, connected homes, and level 4 self-driving cars. The technology will reach its plateau of productivity in 5–10 years [3].

1.4 What exactly is the internet of things?

Despite the diversity of research on IoT, its definition remains somewhat fuzzy. There are different definitions that reflect different perspectives; the following are the most prominent among them:

“IoT is a group of infrastructures interconnecting sensor(s) and/or actuator(s) with (limited) computing capabilities, and allowing their management, access and transfer of the data they generate over the internet without requiring human to human or human to computer interaction “[4].

“IoT is a global network and service infrastructure of variable density and connectivity with self-configuring capabilities based on standard and interoperable protocols and format [which] consists of heterogeneous things that have identities physical and virtual attributes and are seamlessly and securely integrated into the internet”[5].

“IoT is a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual “things” have identities physical attributes and virtual personalities and use intelligent interfaces and are seamlessly integrated into the information network “[6].

“IoT is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment the IoT comprises an ecosystem that includes things communication, applications and data analysis “[7].

“IoT refers to a loosely coupled decentralized system of devices augmented with sensing processing and network capabilities” [8].

1.5 Enabling Technologies for IoT

Internet of things (IoT) is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies

With the Internet of Things the communication is extended via Internet to all the things that surround us. The Internet of Things is much more than machine to machine communication, wireless sensor networks, sensor networks , 2G/3G/4G,GSM,GPRS,RFID, WI-FI, GPS, microcontroller, microprocessor etc. These are considered as being the enabling technologies that make “Internet of Things” applications possible.

Enabling technologies for the Internet of Things are considered in [9] and can be grouped into three categories: (1) technologies that enable “things” to acquire contextual information, (2) technologies that enable “things” to process contextual information, and (3) technologies to improve security and privacy. The first two categories can be jointly understood as functional building blocks required building “intelligence” into “things”, which are indeed the features that differentiate the IoT from the usual Internet. The third category is not a functional but rather a de facto requirement, without which the penetration of the IoT would be severely reduced. [10]

The Internet of Things is not a single technology, but it is a mixture of different hardware & software technology. The Internet of Things provides solutions based on the integration of information technology, which refers to hardware and software used to store, retrieve, and process data and communications technology which includes electronic systems used for communication between individuals or groups.

There is a heterogeneous mix of communication technologies, which need to be adapted in order to address the needs of IoT applications such as energy efficiency, speed, security, and reliability. In this context, it is possible that the level of diversity will be scaled to a number a manageable connectivity technologies that address the needs of the IoT applications, are adopted by the market, they have already proved to be serviceable, supported by a strong technology alliance. Examples of standards in these categories include wired and wireless technologies like Ethernet, WI-FI, Bluetooth, ZigBee, GSM, and GPRS. [9, 10]

The key enabling technologies for the Internet of Things is presented in Figure 1

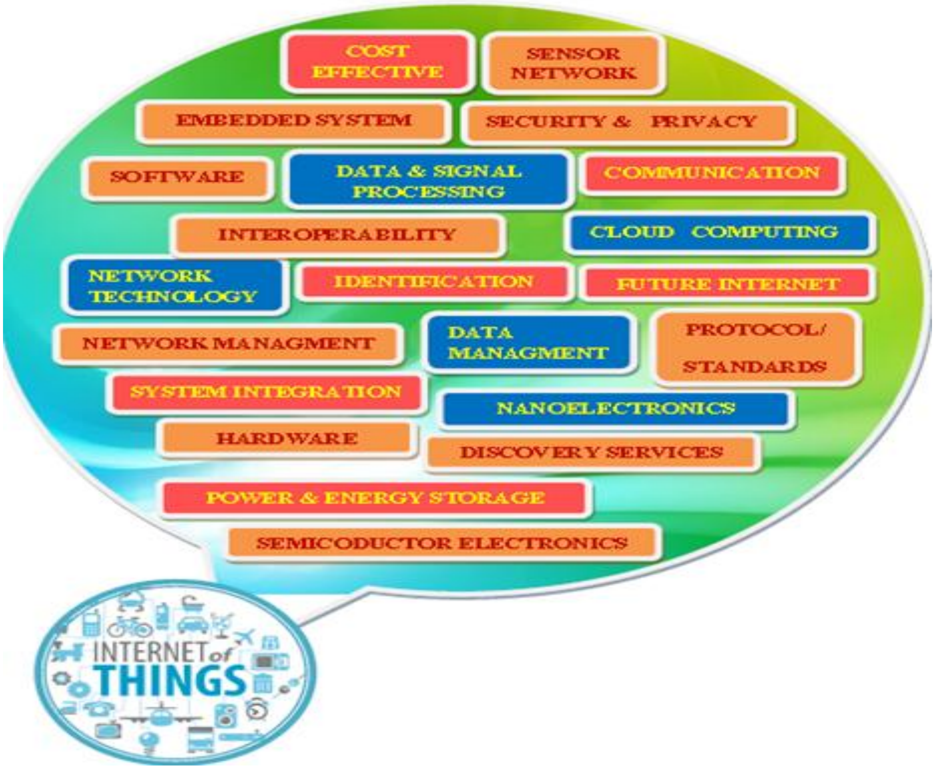


Figure 1: IoT enabling technologies

1.6 Characteristics

The fundamental characteristics of the IoT are as follows [10, 11]:

1.6.1 Interconnectivity: With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.

1.6.2 Things-related services: The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in physical world and information world will change.

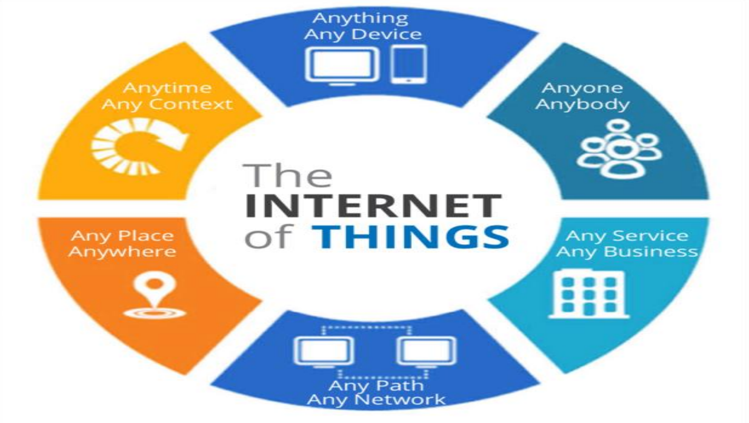


Figure 2: Internet of things

1.6.3 Heterogeneity: The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

1.6.4 Dynamic changes: The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.

1.6.5 Enormous scale: The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet. Even more critical will be the management of the data generated and their interpretation for application purposes. This relates to semantics of data, as well as efficient data handling.

1.6.6 Safety: As we gain benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes the safety of our personal data and the safety of our physical well-being. Securing the endpoints, the networks, and the data moving across all of it means creating a security paradigm that will scale.

1.6.7 Connectivity: Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data.

1.7 IoT Architecture

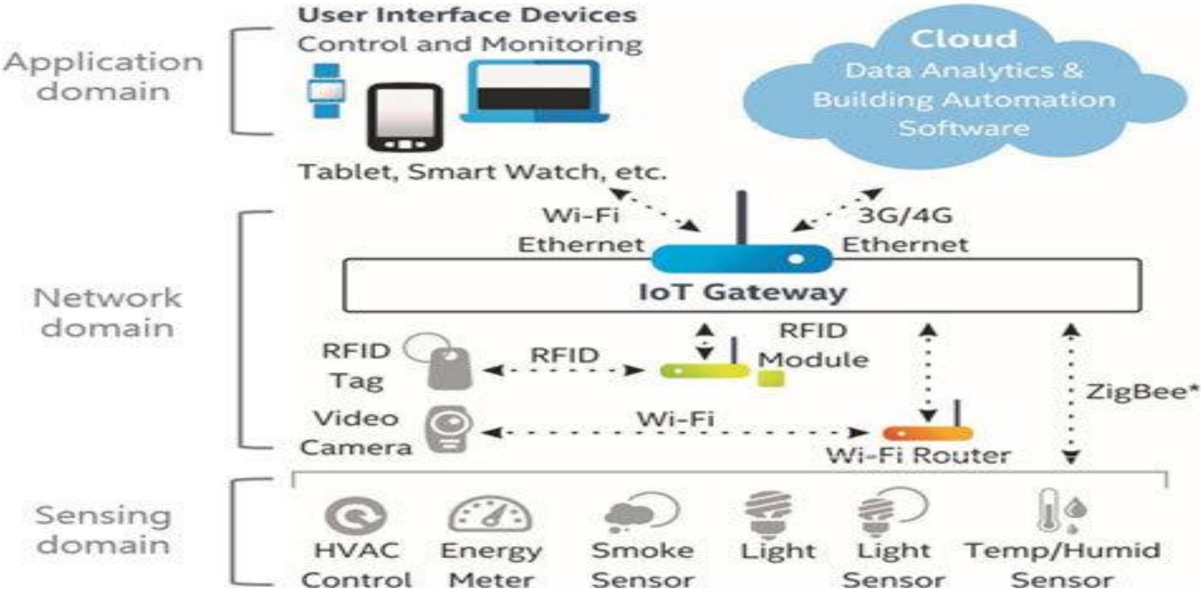


Figure 3: IoT architecture

The functionality of each layer is described below [10, 12]:

1.7.1 Smart device / sensor layer:

The lowest layer is made up of smart objects integrated with sensors. The sensors enable the interconnection of the physical and digital worlds allowing real-time information to be collected and processed. There are various types of sensors for different purposes. The sensors have the capacity to take measurements such as temperature, air quality, speed, humidity, pressure, flow, movement and electricity etc. In some cases, they may also have a degree of memory, enabling them to record a certain number of measurements.

A sensor can measure the physical property and convert it into signal that can be understood by an instrument. Sensors are grouped according to their unique purpose such as environmental sensors, body sensors, home appliance sensors and vehicle telematics sensors, etc.

Most sensors require connectivity to the sensor gateways. This can be in the form of a Local Area Network (LAN) such as Ethernet and Wi-Fi connections or Personal Area Network (PAN) such as ZigBee, Bluetooth and Ultra Wideband (UWB). For sensors that do not require connectivity to sensor aggregators, their connectivity to backend servers/applications can be provided using Wide Area Network (WAN) such as GSM, GPRS and LTE. Sensors that use low power and low data rate connectivity, they typically form networks commonly known as wireless sensor networks (WSNs). WSNs are gaining popularity as they can accommodate far more sensor nodes while retaining adequate battery life and covering large area

1.7.2 Gateways and Networks

Massive volume of data will be produced by these tiny sensors and this requires a robust and high performance wired or wireless network infrastructure as a transport medium. Current networks, often tied with very different protocols, have been used to support machine-to-machine (M2M) networks and their applications. With demand needed to serve a wider range of IOT services and applications such as high speed transactional services, context-aware applications, etc, multiple networks with various technologies and access protocols are needed to work with each other in a heterogeneous configuration. These networks can be in the form of a private, public or hybrid models and are built to support the communication requirements for latency, bandwidth or security. Various gateways (microcontroller, microprocessor...) & gateway networks (WI-FI, GSM, GPRS…)

An IoT gateway is an intelligent central hub for IoT devices. IoT gateways connect devices within the Internet of Things to one another and to the cloud, translating communication between the devices and filtering data into useful information.

A true IoT gateway contains communication technologies connecting end-devices (sensors, actuators or more complex devices) and backend platforms (data, device and subscriber management) to the gateway. It has a computing platform allowing pre-installed or user-defined applications to manage data (for routing and computing at edge), devices, security, communication and other aspects of the gateway.

1.7.3 Management Service Layer

The data needs to be stored for further in-depth analysis which is why data storage is such an important stage of an IoT architecture. It helps with follow-up revision for feedback as well. Cloud storage is the preferred storage method in IoT implementations.

That's also because more in-depth processing which doesn't require immediate feedback can be carried out in the cloud or at physical data centers. There, more capable IT systems can manage, analyze and more securely store the data. This is also where sensor data can be combined with other data sources for more detailed insights

1.7.4 Application Layer

The IoT application covers "smart" environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

1.8 Applications of IoT [13]

1.8.1 Smart Homes

One of the best and the most practical applications of IoT, smart homes really take both, convenience and home security, to the next level. Though there are different levels at which IoT is applied for smart homes, the best is the one that blends intelligent utility systems and entertainment together. For instance, your electricity meter with an IoT device giving you insights into your everyday water usage, your set-top box that allows you to record shows from remote, Automatic Illumination Systems, Advanced Locking Systems, Connected Surveillance Systems all fit into this concept of smart homes. As IoT evolves, we can be sure that most of the devices will become smarter, enabling enhanced home security

Figure 4: Smart homes

1.8.2 Smart City

Not just internet access to people in a city but to the devices in it as well – that's what smart cities are supposed to be made of. And we can proudly say that we're going towards realizing this dream. Efforts are being made to incorporate connected technology into infrastructural requirements and some vital concerns like Traffic Management, Waste Management, Water Distribution, Electricity Management, and more. All these work towards eliminating some day-to-day challenges faced by people and bring in added convenience.

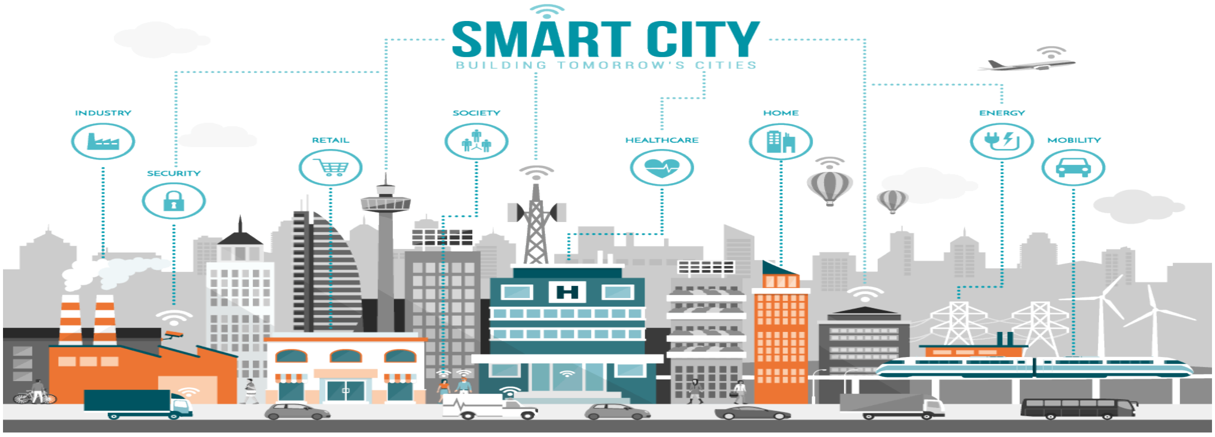


Figure 5: Smart cities

1.8.3 Industrial internet of things

Industrial Internet of Things (IIoT) is **a way to digital transformation in manufacturing**. Industrial IoT employs a network of sensors to collect critical production data and uses cloud software to turn this data into valuable insights about the efficiency of the manufacturing operations

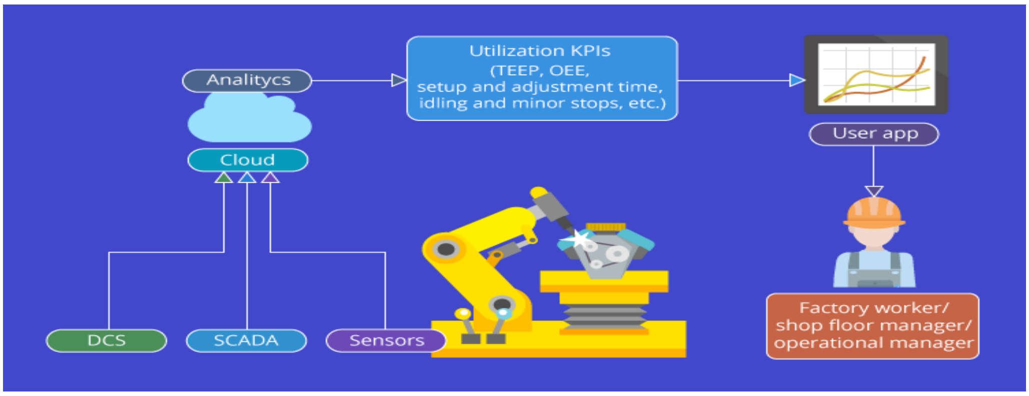


Figure 6: Smart Manufacturing

1.8.4 IoT Retail Shops

If you haven't already seen the video of Amazon Go – the concept store from the ecommerce giant, you should check it out right away. Perhaps this is the best use of the technology in bridging the gap between an online store and a retail store. The retail store allows you to go cashless by deducting money from your Amazon wallet. It also adds items to your cart in real-time when you pick products from the shelves

If you change your mind and pick up another article, the previous one gets deleted and replaces your cart with the new item. The best part of the concept store is that there is no cashier to bill your products. You don't have to stand in line but just step out after you pick up your products from shelves. If this technology is effective enough to fetch more patronage, this is sure to become a norm in the coming years.

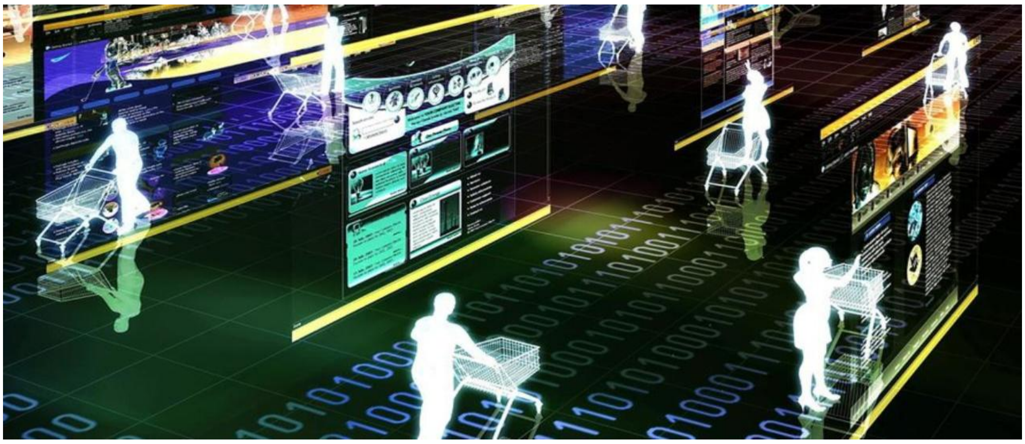


Figure 7: IoT Retail Shops

1.8.5 Farming

Farming is one sector that will benefit the most from the Internet of Things. With so many developments happening on tools farmers can use for agriculture, the future is sure promising. Tools are being developed for Drip Irrigation, understanding crop patterns, Water Distribution, drones for Farm Surveillance, and more. These will allow farmers to come up with a more productive yield and take care of the concerns better.

Figure 8: Farming

1.8.6 Wearable

Wearables remain a hot topic in the market, even today. These devices serve a wide range of purposes ranging from medical, wellness to fitness. Of all the IoT startups, Jawbone, a wearables maker, is second to none in terms of funding.

Figure 9: Wearables

1.8.7 Smart Grids

One of the many useful IoT examples, a smart grid, is a holistic solution that applies an extensive range of Information Technology resources that enable existing and new gridlines to reduce electricity waste and cost. A future smart grid improves the efficiency, reliability, and economics of electricity.

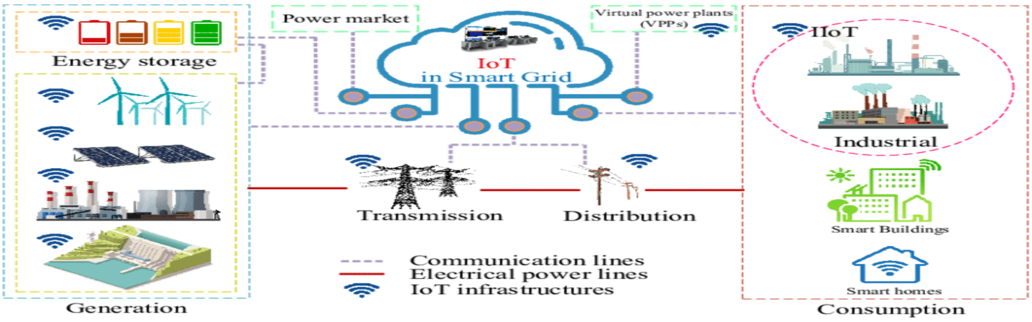


Figure 10: Smart grid

1.8.8 Telehealth

Telehealth, or Telemedicine, hasn’t completely flourished yet. Nonetheless, it has great future potential. IoT Examples of Telemedicine include the digital communication of Medical Imaging, Remote Medical Diagnosis & Evaluations, Video Consultations with Specialists, etc

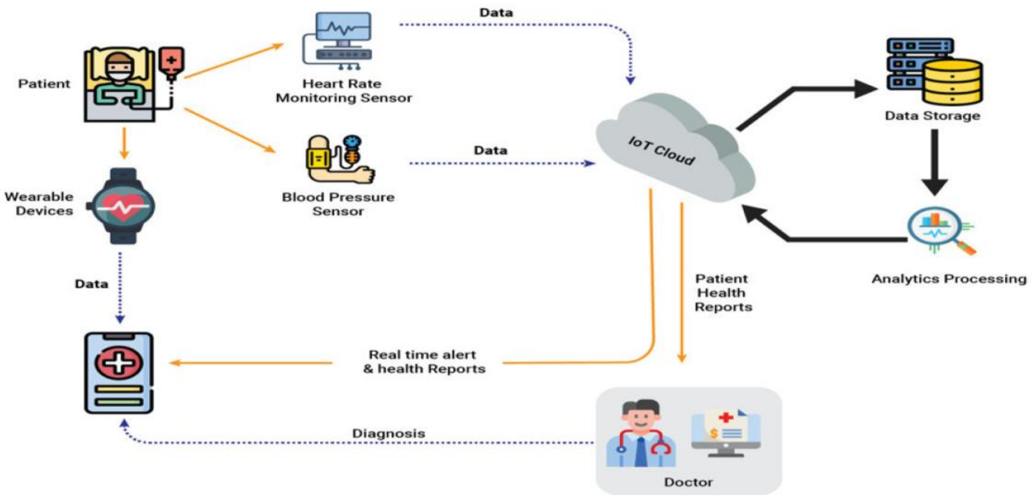


Figure 11: Telehealth

1.8.9 Smart Supply-chain Management

Supply-chains have stuck around in the market for a while now. A common example can be Solutions for tracking goods while they are on the road. Backed with IoT technology, they are sure to stay in the market for the long run

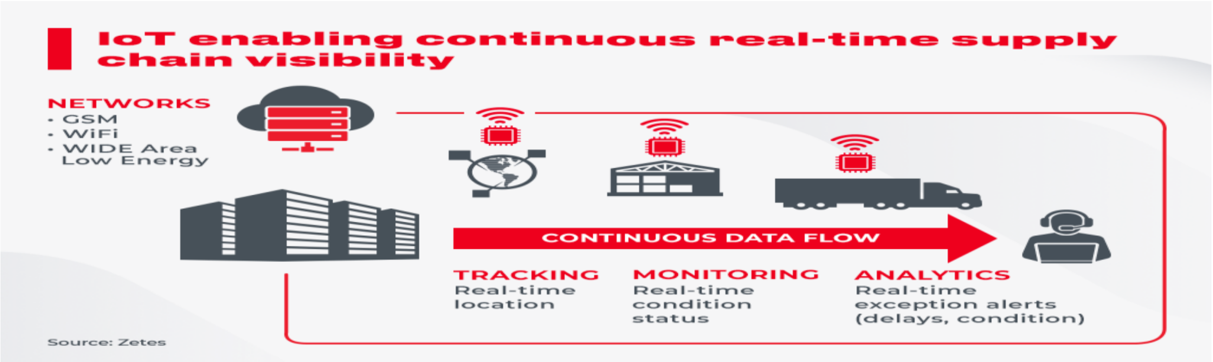

IoT enabling continuous real-time supply chain visibility
NETWORKS
· GSM
· WiFi
· WIDE Area Low Energy
CONTINUOUS DATA FLOW
TRACKING
Real-time location
MONITORING
Real-time condition status
ANALYTICS
Real-time exception alerts (delays, condition)
Source: Zetes

Figure 12: Smart Supply-chain Management

1.8.10 Traffic Management

Car traffic management in large cities can be greatly improved with the help of the Internet of Things (IoT). The Internet of Things helps us stay informed and improves traffic monitoring by allowing us to use our mobile phones as sensors to collect and share data from our vehicles through apps like Google Maps. This feeds and improves the data on the various routes to the same destination, distance, and estimated arrival time.

Analysis of traffic patterns over a long period is another IoT application. It provides an idea of what might happen during peak hours. Commuters will be better prepared to avoid traffic and delays by being made aware of possible alternatives

1.8.11 Water/ Waste Management

Many cities are adopting water recycling using water treatment units. Using an IoT application, you can see how much wastewater is being produced, how much is being consumed in a specific area, and how waste production is changing over time.

We can effectively deal with this problem using Internet of Things applications and smart sensor technology. With a smart waste management system, authorities will be able to predict how much waste will be generated in a specific location, how to properly process it, when to clear it, and how to analyze data for future planning, among other things.

An overview of waste generated in every neighborhood and how much waste is generated

Over time can be easily assessed using analytics solutions. This data will be used to plan the city’s expansion and upgrade projects. Smart analytics solutions can be used to manage waste collection and treatment fleets, as well as to predict future trends

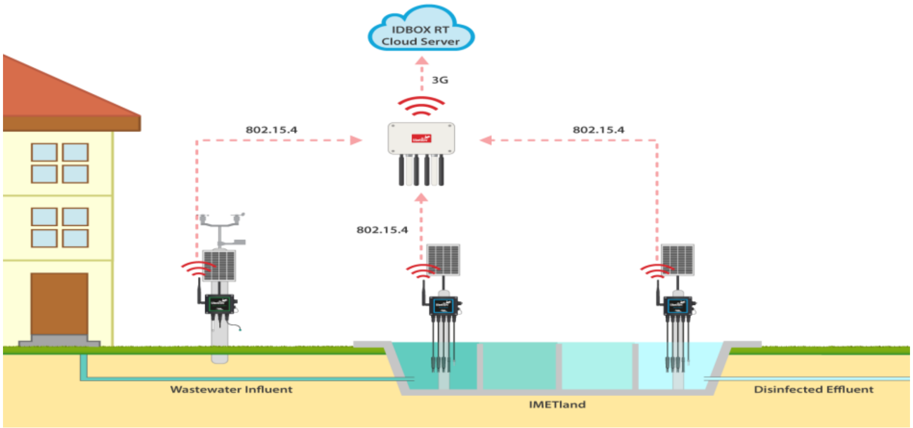


Figure 13: Water waste management

1.9 Conclusion

Internet of Things is the concept in which the virtual world of information technology connected to the real world of things and it is a key research topic for researcher in embedded, computer science & information technology area due to its very diverse area of application & heterogeneous mixture of various communications and embedded technology in its architecture.

Chapter 2: Theoretical background

2.1 Introduction

The concept of Industrial Automation and Control Systems (IACS) is well established. These systems, often referred to as Operational Technology (OT), are employed in diverse industries including manufacturing, transportation and utilities, and are sometimes referred to as cyber-physical systems (CPS). Since the term Internet of Things (IoT) [14] was first used in 1999, it has been applied to connected devices in consumer, domestic, business and industrial settings. Cyber Physical Systems (CPS), Industrial Control Systems (ICS), Supervisory Control and Data Acquisition (SCADA), and Industrial Internet. Although not an exhaustive list, it does represent the most commonly used terms in both academic and relevant non-academic literature, for white papers and corporate blogs. In the rest of this section we define Industry 4.0 and review the above terms before moving on to develop a definition of IIoT

2.2 Industry 4.0

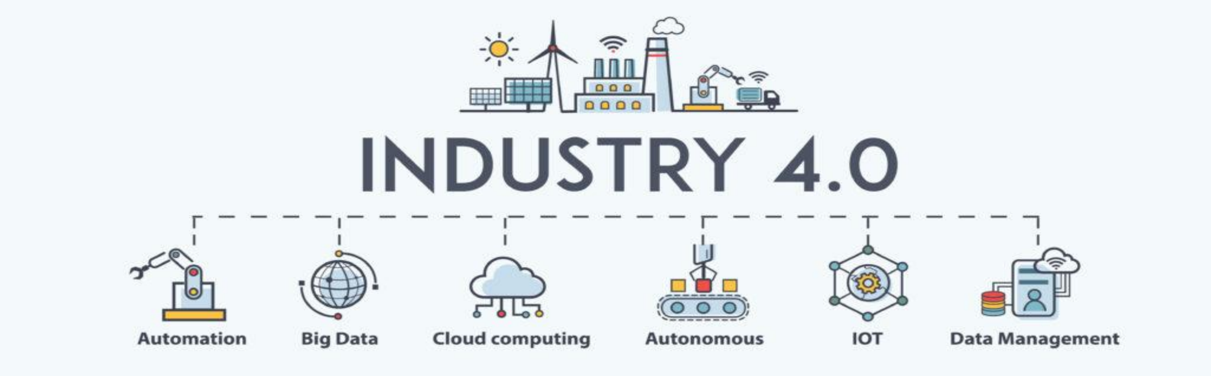


Figure 14: Industry 4.0 enabling technologies

The first three industrial revolutions are characterized as being driven by mechanical production relying on water and steam power, use of mass labor and electrical energy, and the use of electronic, automated production respectively [15].Whilst the supposed fourth industrial revolution (‘Industry 4.0’) was first proposed in 2011 in the context of the goal of developing the German economy [16]. This revolution is characterized by its reliance on the use of CPS capable of communication with one another and of making autonomous, decentralized decisions, with the aim of increasing industrial efficiency, productivity, safety, and transparency. There is a considerable overlap between the concept of Industry 4.0 developed in Germany and the Industrial Internet concept, which originated in the United States. The definition of the latter now encompasses change for both business and individuals:

“the industrial internet is an internet of things, machines, computers and people enabling intelligent industrial operations using advanced data analytics for transformational business outcomes, and it is redefining the landscape for business and individuals alike”[17].

A definition of ‘Industry 4.0′ a term which, in its English cognate, the authors treat as synonymous with IIoT, is:

“We define Industry 4.0 as follows: Industry 4.0 is a collective term for technologies and concepts of value chain organization. Within the modular structured Smart Factories of Industry 4.0, CPS monitor physical processes, creates a virtual copy of the physical world and makes decentralized decisions. Over the IoT, CPS communicate and cooperate with each other and humans in real time. Via the IoS [Internet of Services], both internal and cross organizational services are offered and utilized by participants of the value chain.” [18]

2.3 Cyber physical systems

“A system comprising a set of interacting physical and digital components, which may be centralized or distributed, that provides a combination of sensing, control, computation and networking functions, to influence outcomes in the real world through physical processes.”[19]

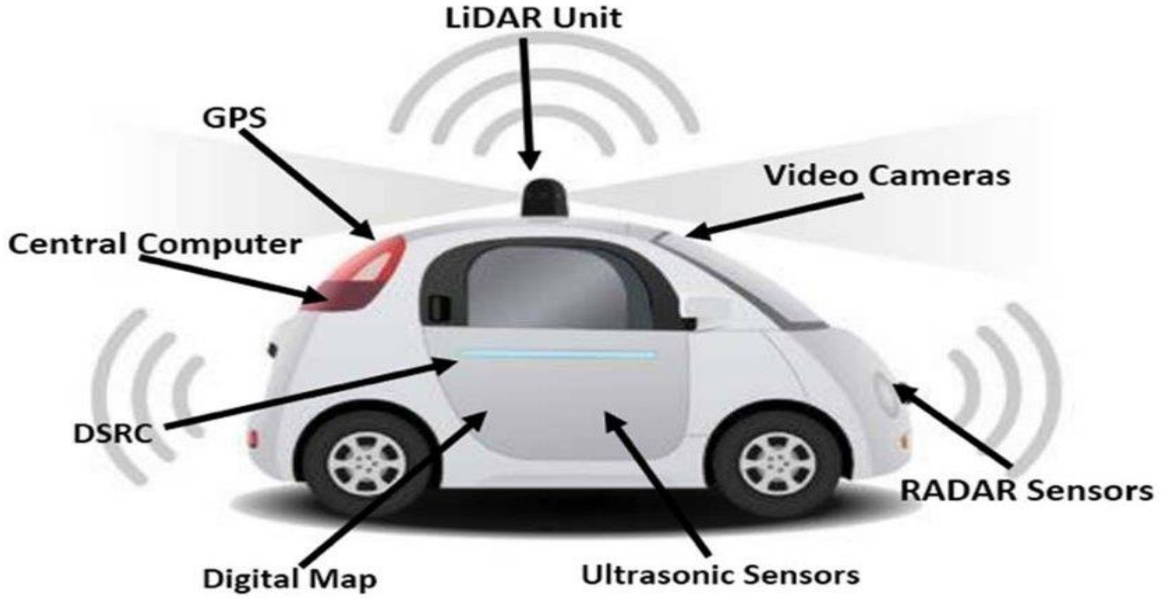


Figure 15: cyber physical system

What sets CPS apart from more conventional information and communications systems (IT or ICT) is the real-time character of their interactions with the physical world. Whilst both CPS and ICT systems process data and/or information, the focus of CPS is on the control of physical processes. CPS use sensors to receive information about, including measurements of, physical parameters, and actuators to engage in control over physical processes. CPS often involve a large degree of autonomy. For example, CPS often have the capacity to determine whether to change the state of an actuator or to draw a human operator's attention to some feature of the environment being sensed.

2.4 Industrial automation and control systems

IACS or ICS is a collective term typically used to describe different types of control systems and associated instrumentation, which include the devices, systems, networks, and controls used to operate and/or automate industrial processes.

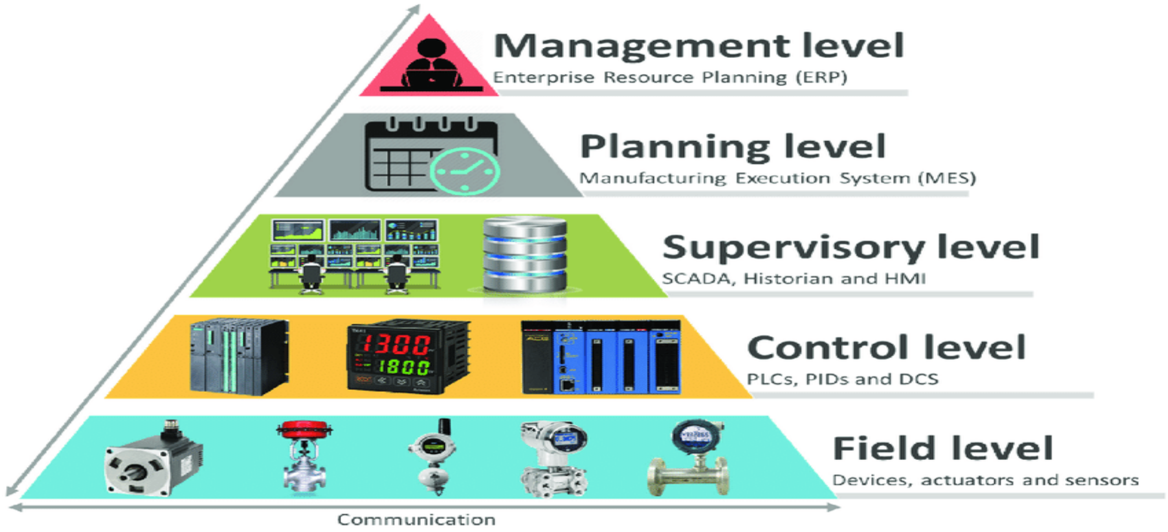


Figure 16: Automation pyramid

2.5 Supervisory control and data acquisition (SCADA)

A system that allows an operator, in a location central to a widely distributed process, such as an oil or gas field, pipeline system, or hydroelectric generating complex, to make set point changes on distant process controllers, to open or close valves or switches, to monitor alarms, and to gather measurement information [20]

SCADA applications are made up of two elements: the process/ system/machinery you want to monitor and control, which can take the form of a power plant, a water system, a network or a system of traffic lights; and a network of intelligent devices that interface with the first system through sensors and control outputs. This network, which is the platform system, provides the capability to measure and control specific elements of the first system [21].

The nature of SCADA has led to conflicting views as to whether it forms part of the IIoT ecosystem. For example, discussion of SCADA system forensic analysis within IIoT [19] contrasts with a view that SCADA is simply the predecessor to IIoT especially as SCADA systems have evolved to connect to the internet but do not have the analytics and level of connectivity that is found in IIoT [22]

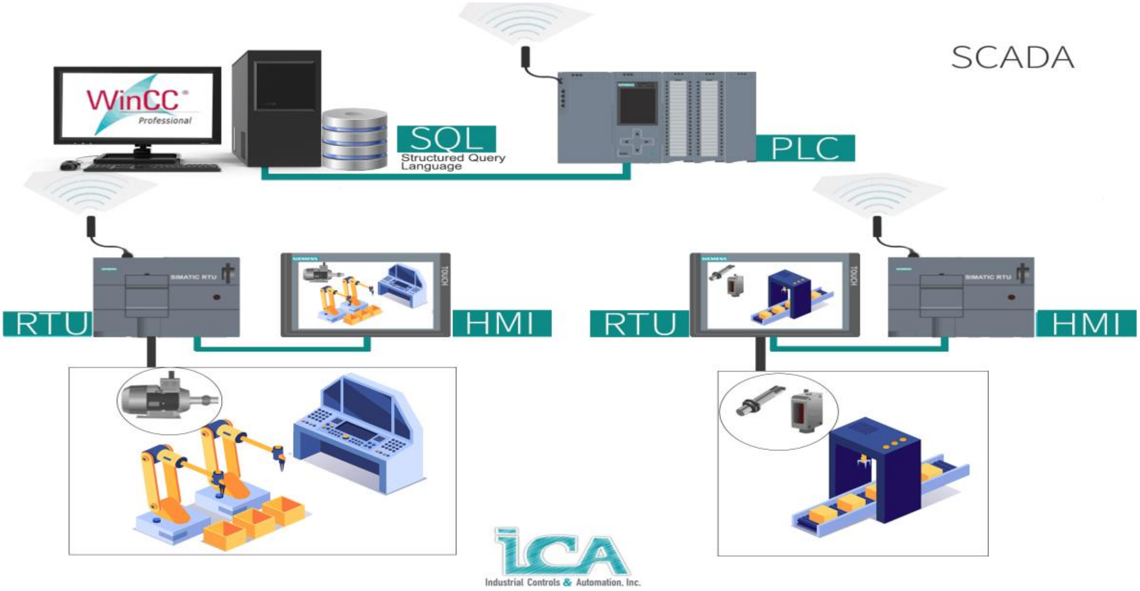

SCADA
WinCC Professional
SQL
Structured Query Language
PLC
RTU
HMI
RTU
HMI
ICA
Industrial Controls & Automation, Inc.

Figure 17: SCADA

2.6 Industrial Internet

The definition of the Industrial Internet includes two key components:

The connection of industrial machine sensors and actuators to local processing and to the Internet The onward connection to other important industrial networks that can independently generate value. The main difference between the consumer/social Internets and the Industrial Internet is in how and how much value is created. For consumer / social Internets, the majority of value is created from advertisements [23].

This description clearly separates the Internet and the Industrial Internet, although in both cases the function of the Internet is to provide the wide area networking. More recently the Industrial Internet has been defined as:

"… a source of both operational efficiency and innovation that is the outcome of a compelling recipe of technology developments. The resulting sum of those parts gives you the Industrial Internet—the tight integration of the physical and digital worlds. The Industrial Internet enables companies to use sensors, software, machine-to machine learning and other technologies to gather and analyze data from physical objects or other large data streams and then use those analyses to manage operations and in some cases to offer new, value-added services [24]."

From this definition, it is apparent that the authors consider a key component of the Industrial Internet to be the ability to analyse data, which is corroborated by a statement later in their report, in which it is stated that "···Big Data analytics is the foundation of the Industrial Internet…" This desire to collect and analyses data is a feature in common with Industry 4.0

2.7 Definition of Industrial internet of things

A system comprising networked smart objects, cyber-physical assets, associated generic information technologies and optional cloud or edge computing platforms, which enable real-time, intelligent, and autonomous access, collection ,analysis, communications, and exchange of process, product and/or service information, within the industrial environment with a strong focus on big data, so as to optimize overall production value. This value may include; improving product or service delivery, boosting productivity, reducing labor costs, reducing energy consumption, and reducing the build to-order cycle [17].

2.8 IIoT Architecture

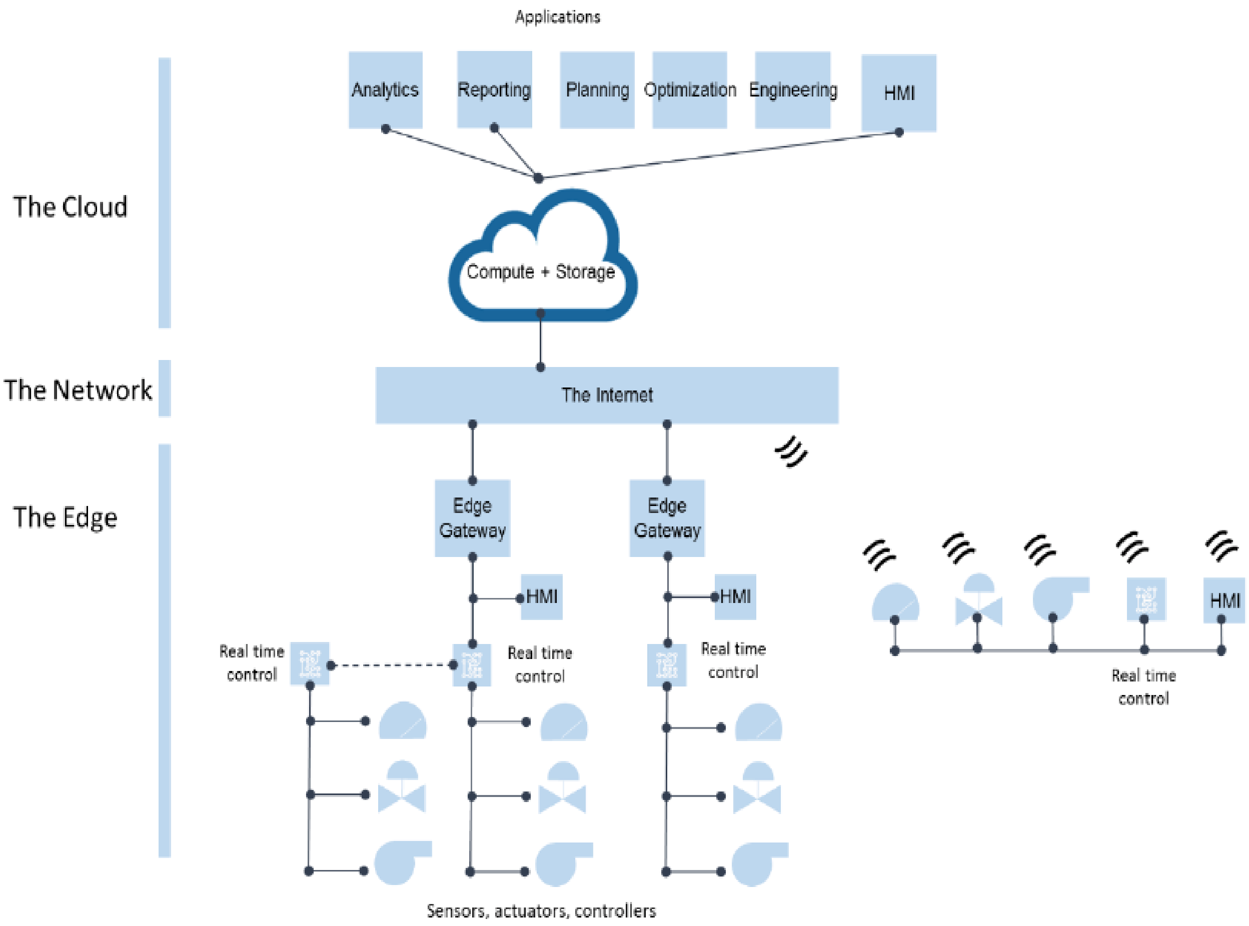


Figure 18: IIoT system architecture

2.8.1 Smart Devices

The process starts with Smart Controllers, sensors and actuators, the connected devices that monitor (in the case of sensors) or control (in the case of actuators) some “thing” or physical process. Sensors capture data regarding the status of a process or an environmental condition, such as temperature, humidity, chemical composition, fluid levels in a tank, fluid flow in a pipe, or the speed of an assembly line as well as much more.

Figure 19: SIMATIC S7-1500

Figure 20: Smart fuel sensor

2.8.2 IoT Gateway

Industrial IOT Gateways enable wireless connectivity of legacy and remote industrial devices with next-generation intelligent infrastructure

Industrial IOT Gateways are rugged by design and intended for critical systems and industrial environments where sensor nodes or I/O devices are utilized to obtain data. IoT Gateways can be configured with a range of different protocols which allows them to communicate with these end sensors nodes or I/O devices [25]

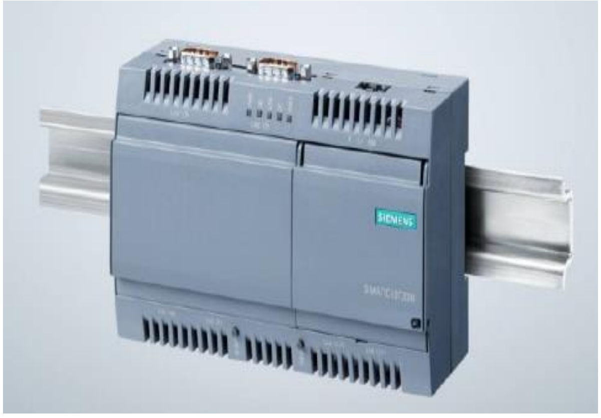


Figure 21: SIMATIC IOT2040

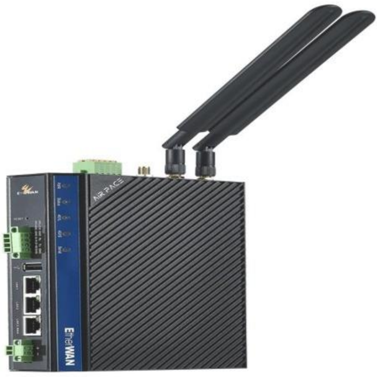


Figure 22: AiR PACE Industrial Smart IoT Edge Computing Gateway

2.8.3 Cloud platforms

2.8.3.1 Siemens MindSphere

MindSphere is a cloud-based platform as a service (PaaS) operating system for IOT devices in industrial and manufacturing sectors used to collect data from sensors in real time. The OS was created by Siemens, an industrial manufacturing company based in Germany.

Platform as a service (PaaS) is a cloud computing model where a third-party provider delivers hardware and software tools to users over the internet. Usually, these tools are needed for application development. A PaaS provider hosts the hardware and software on its own infrastructure. As a result, PaaS frees developers from having to install in-house hardware and software to develop or run a new application.

2.8.3.2 IBM Private cloud for manufacturing

IBM private cloud for manufacturing satisfies key needs within the manufacturing floor, such as OT-IT integration, security, edge analytics and device management, software-defined networking, and application modernization, based on Red Hat OpenShift Container Platform and IBM Cloud Pak software [28]

Private cloud (also known as an internal cloud or corporate cloud) is a cloud computing environment in which all hardware and software resources are dedicated exclusively to, and accessible only by, a single customer. Private cloud combines many of the benefits of cloud computing including elasticity, scalability, and ease of service delivery with the access control, security, and resource customization of on-premises infrastructure [29].

2.8.3.3 Amazon Web Services

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings [30].

Infrastructure as a service (IaaS) is a type of cloud computing service that offers essential compute, storage, and networking resources on demand, on a pay-as-you-go basis.

Software as a service (SaaS) is a software distribution model in which a cloud provider hosts applications and makes them available to end users over the internet. In this model, an independent software vendor (ISV) may contract a third-party cloud provider to host the application [31].

2.9 Data Protocols

IoT Data protocols are used to connect IoT devices. Among many others, we used MQTT and HTTP in this project

2.9.1 MQTT

2.9.1.1 History of MQTT

MQTT was invented in 1999 by engineers Andy Stanford-Clark and Arlen Nipper, as a method of allowing pipelines in the oil and gas industry to communicate with Supervisory Control and Data Acquisition (SCADA) systems. At the time, these systems used disparate, proprietary protocols, and, as such, were not able to communicate with each other. Adding MQTT capabilities helped overcome inter-communication problems. In addition to interoperability, the original goals for the protocol were that it should be lightweight, bandwidth efficient, data agnostic and simple to implement while offering quality-of-service data delivery [32].

2.9.1.2 MQTT (Message Queuing Telemetry Transport)

An MQTT (Message Queuing Telemetry Transport) is a lightweight IoT data protocol. It features a publisher-subscriber messaging model and allows for simple data flow between different devices. The protocol usually runs over TCP/IP and it is designed for connections with remote locations where resource constraints exist or the network bandwidth is limited. The protocol is an open OASIS standard and an ISO recommendation (ISO/IEC 20922) [33].

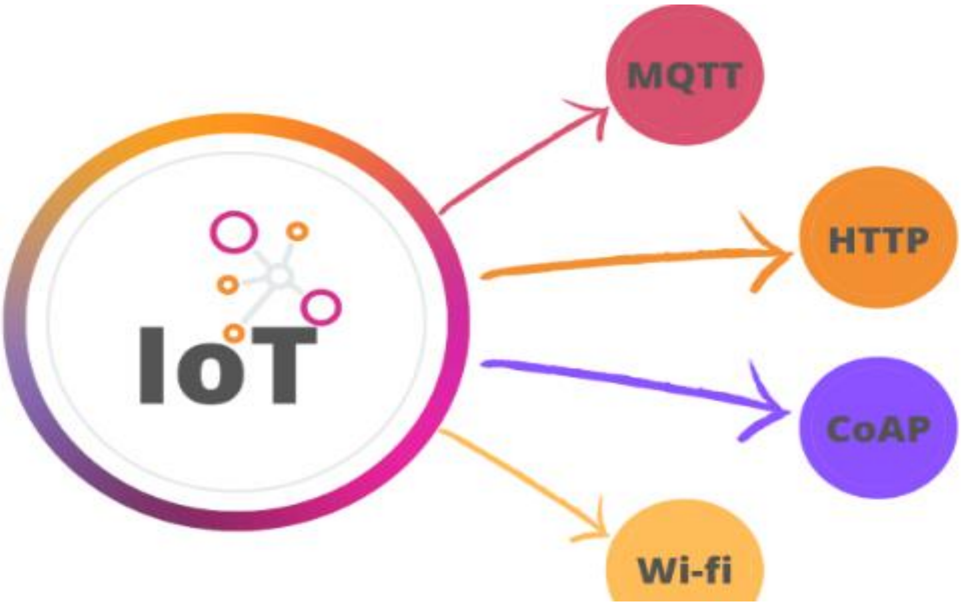


Figure 23: IoT Protocols

2.9.1.3 MQTT architecture

The connected devices in the MQTT protocol are known as “clients,” which communicate with a server referred to as the “broker.” The broker handles the task of data transmission between clients.

Whenever a client (known as the “publisher”) wants to distribute information, it will publish to a particular topic (MQTT topics are a form of addressing that allows MQTT clients to share information), the broker then sends this information to any clients that have subscribed to that topic (known as “subscribers”).

The publisher does not need any data on the number or the locations of subscribers. In turn, subscribers do not need any data about the publisher. Any client can be a publisher, subscriber, or both. The clients are typically not aware of each other, only of the broker that serves as the intermediary. This setup is popularly known as the “pub/sub model.”[32]

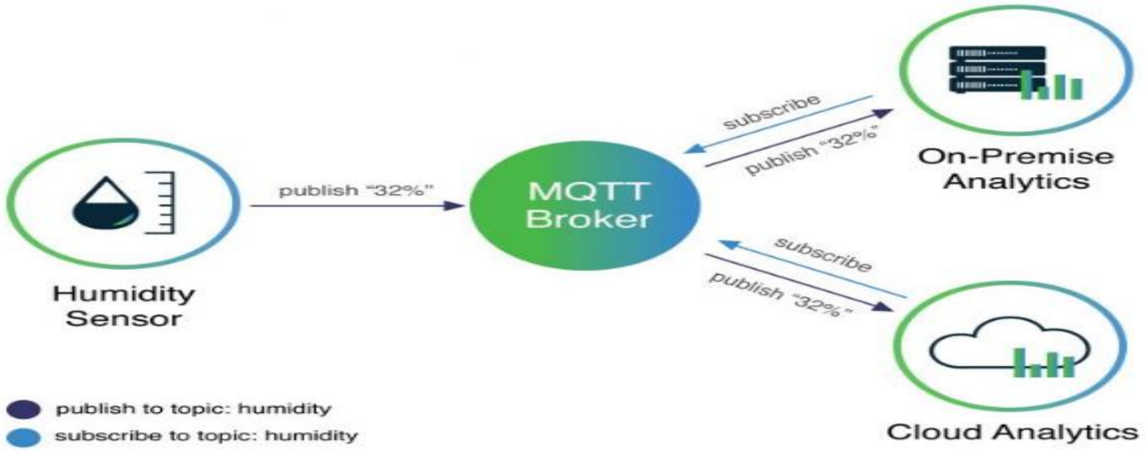


Figure 24 : MQTT Architecture

2.9.1.4 MQTT Topics

Messages make their way from a publisher, through a broker, to one or more subscribers using topics. Topics are hierarchical UTF-8 strings. Each level in a topic is delimited by a forward slash. Every message from a publisher must include a topic. To receive the published message, the entity that consumes the message must subscribe to the same topic. A broker sends the message it receives only to those clients that have subscribed to the same topic. [32]

2.9.2 HTTP

2.9.2.1 History

HTTP (HyperText Transfer Protocol) is the underlying protocol of the World Wide Web. Developed by Tim Berners-Lee and his team between 1989-1991, HTTP has gone through many changes that have helped maintain its simplicity while shaping its flexibility. Keep reading to learn how HTTP evolved from a protocol designed to exchange files in a semitrusted laboratory environment into a modern internet maze that carries images and videos in high resolution and 3D [34].

2.9.2.2 Invention of the World Wide Web

In 1989, while working at CERN, Tim Berners-Lee wrote a proposal to build a hypertext system over the internet. Initially called the Mesh, it was later renamed the World Wide Web during its implementation in *1990.* Built over the existing TCP and IP protocols, *it* consisted of 4 building blocks:

- A textual format to represent hypertext documents, the HyperText Markup Language (HTML).
- A simple protocol to exchange these documents, the HyperText Transfer Protocol (HTTP).
- A client to display (and edit) these documents, the first web browser called the **World Wide Web.**
- A server to give access to the document, an early version of httpd.(The Apache HTTP Server)

These four building blocks were completed by the end of 1990, and the first servers were running outside of CERN by early 1991. On August 6, 1991, Tim Berners-Lee posted on the public alt.hypertext newsgroup. This is now considered to be the official start of the World Wide Web as a public project [34].

2.9.2.3 Definition

The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web, and is used to load web pages using hypertext links. HTTP is an application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack. A typical flow over HTTP involves a client machine making a request to a server, which then sends a response message [35].

2.9.2.4 Architecture

The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server [36].

2.9.2.5 Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection [36].

2.9.2.6 Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content [36].

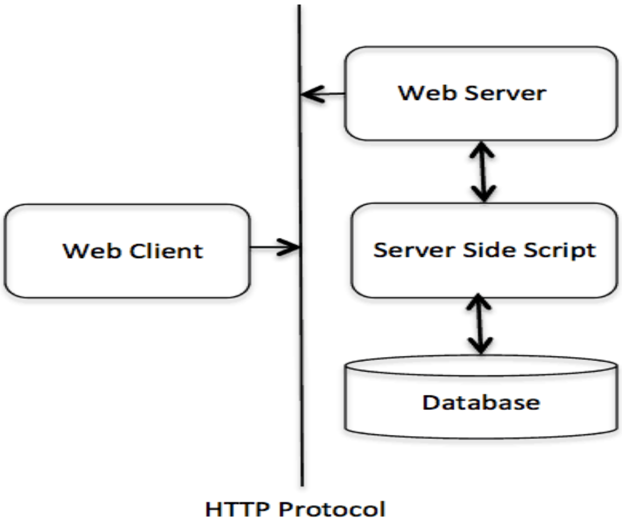


Figure 25: HTTP Architecture

2.9.2.7 Components of HTTP based system

HTTP is a client-server protocol: requests are sent by one entity, the user-agent (or a proxy on behalf of it). Most of the time the user-agent is a Web browser, but it can be anything, for example, a robot that crawls the Web to populate and maintain a search engine index.

Each individual request is sent to a server, which handles it and provides an answer called the response. Between the client and the server there are numerous entities, collectively called proxies, which perform different operations and act as gateways or caches, for example [36].

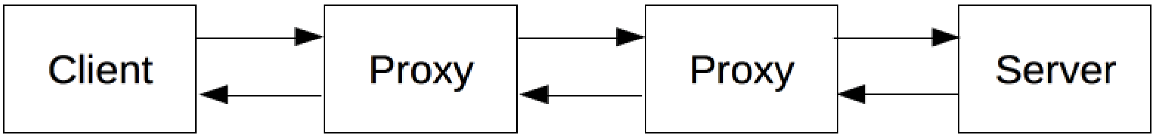


Figure 26: HTTP based system

2.9.2.7.1 Client: the user-agent

The user-agent is any tool that acts on behalf of the user. This role is primarily performed by the Web browser, but it may also be performed by programs used by engineers and Web developers to debug their applications.

The browser is **always** the entity initiating the request. It is never the server (though some mechanisms have been added over the years to simulate server-initiated messages).

To display a Web page, the browser sends an original request to fetch the HTML document that represents the page. It then parses this file, making additional requests corresponding to execution scripts, layout information (CSS) to display, and sub-resources contained within the page (usually images and videos). The Web browser then combines these resources to present the complete document, the Web page. Scripts executed by the browser can fetch more resources in later phases and the browser updates the Web page accordingly.

A Web page is a hypertext document. This means some parts of the displayed content are links, which can be activated (usually by a click of the mouse) to fetch a new Web page, allowing the user to direct their user-agent and navigate through the Web. The browser translates these directions into HTTP requests, and further interprets the HTTP responses to present the user with a clear response [36].

2.9.2.7.2 The Web server

On the opposite side of the communication channel is the server, which serves the document as requested by the client. A server appears as only a single machine virtually; but it may actually be a collection of servers sharing the load (load balancing), or a complex piece of software interrogating other computers (like cache, a DB server, or e-commerce servers), totally or partially generating the document on demand [36]

2.9.2.7.3 Proxies

Between the Web browser and the server, numerous computers and machines relay the HTTP messages. Due to the layered structure of the Web stack, most of these operate at the transport, network or physical levels, becoming transparent at the HTTP layer and potentially having a significant impact on performance.

Those operating at the application layers are generally called **proxies**. These can be transparent, forwarding on the requests they receive without altering them in any way, or non-transparent, in which case they will change the request in some way before passing it along to the server. Proxies may perform numerous functions [36]:

- caching (the cache can be public or private, like the browser cache)
- filtering (like an antivirus scan or parental controls)
- load balancing (to allow multiple servers to serve different requests)
- authentication (to control access to different resources)
- logging (allowing the storage of historical information)

2.9.2.8 Uniform Resource Locator (URL)

A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL). The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet. The URL defines four parts: method, host computer, port, and path [36].

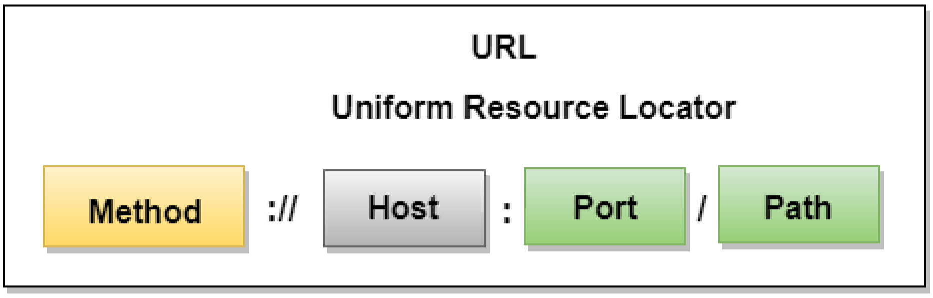


Figure 27: Uniform Resource Locator

- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.
- **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.
- **Path:** Path is the pathname of the file where the information is stored. The path itself contains slashes that separate the directories from the subdirectories and files.

2.9.2.9 HTTP and IoT

HyperText Transfer Protocol is the best example of IoT network protocol. This protocol has formed the foundation of data communication over the web. It is the most common protocol that is used for IoT devices when there is a lot of data to be published. However, the HTTP protocol is not preferred because of its cost, battery-life, energy saving, and more constraints.

Additive manufacturing/3D printing is one of the use cases of the HTTP protocol. It enables computers to connect 3D printers in the network and print three-dimensional objects and pre-determined process prototypes [38].

2.9.3 OPC UA

The latest of the communication protocols to be introduced in the industry is OPC UA (OPC Unified Architecture). The way that OPC UA differs from the other communication protocols Listed previously, is that it decentralizes the need for having a hierarchy with the system.
This means that OPC UA enabled devices can communicate directly to the cloud by themselves without the need to have intermediary devices in between to send the data up Stream. Since OPC UA is mentioned as an agnostic system, it allows for devices from different suppliers to be easily integrated into the communication network [46]. As shown in figure 10, OPC UA can be used as a direction communication between the Edge devices and the cloud, as well as being a communication protocol among the factory operations that are in

place. The automation pyramid best describes how OPC UA can be implemented both within a manufacturing system as well as being a part of the cloud communication [46]

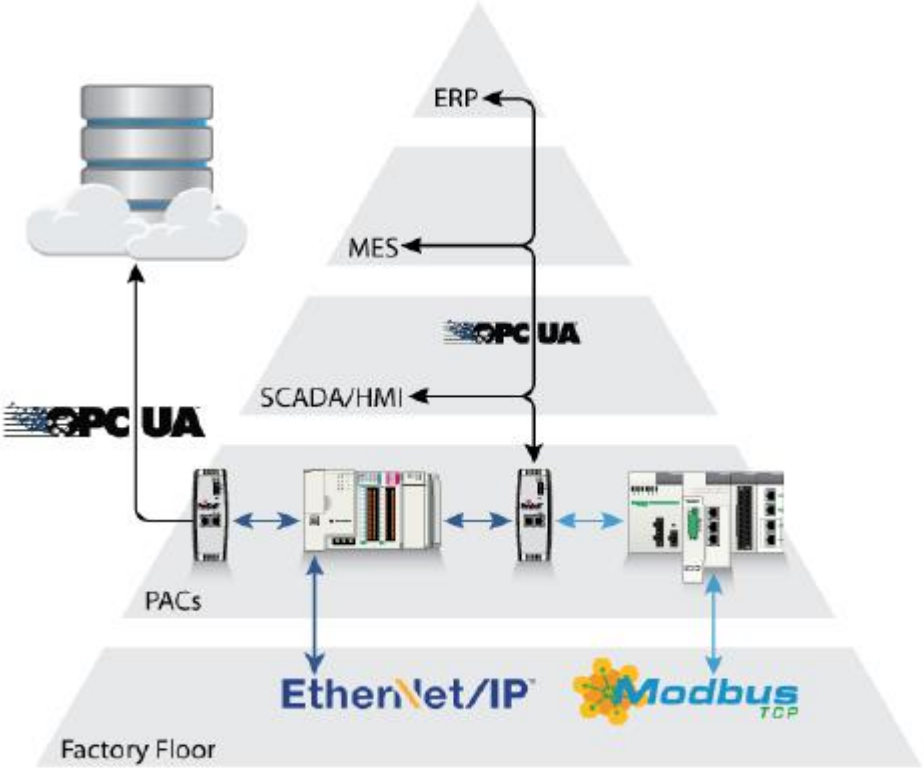


Figure 28: OPC UA communication protocol in automation pyramid

2.10 Network Protocols

IoT network protocols are used to connect devices over a network. We are using Wi-Fi, 4G and Ethernet in this project.

2.10.1 Wi-Fi

2.10.1.1 Wi-Fi definition

Wireless Fidelity (Wi-Fi) is a technology that utilizes Radio frequency for exchange of Information in Local Area Network (LAN) without cable and wiring. It was brought under The name WAVELAN in the 1980s and further developed into IEEE 802.11 standard in 1997 [39]. the standard supported only 2Mbps connections which were too slow until new versions Of standard such as 802.11a, 802.11b, 802.11g, 802.11n, and so on were developed [40].
Wi-Fi is an open standard technology, and its operation is in the unlicensed 2.4GHz radio band with 11mbps or 54mbps data rate and ten times faster than the regular dial-up connection [40]. Wi-Fi technology was widely accepted and integrated into digital products to

realize a wireless digital network. Victor Bahl first used it in 2000 for indoor positioning. Later on, in 2009, it was introduced in subway systems and used as an outdoor location technology by Nobuo Kawaguchi [41]. Wireless Fidelity has a broad availability of terminal devices in the cordless phone, camera, media players, personal data assistants, and laptops. Its connectivity to the internet can be from anywhere in the world, like a hotel room, conference room, coffee shop, pharmaceutical shops, etc. Networking of Wi-Fi is possible through three elements, which are Radio signals, Access point, and Network interface card

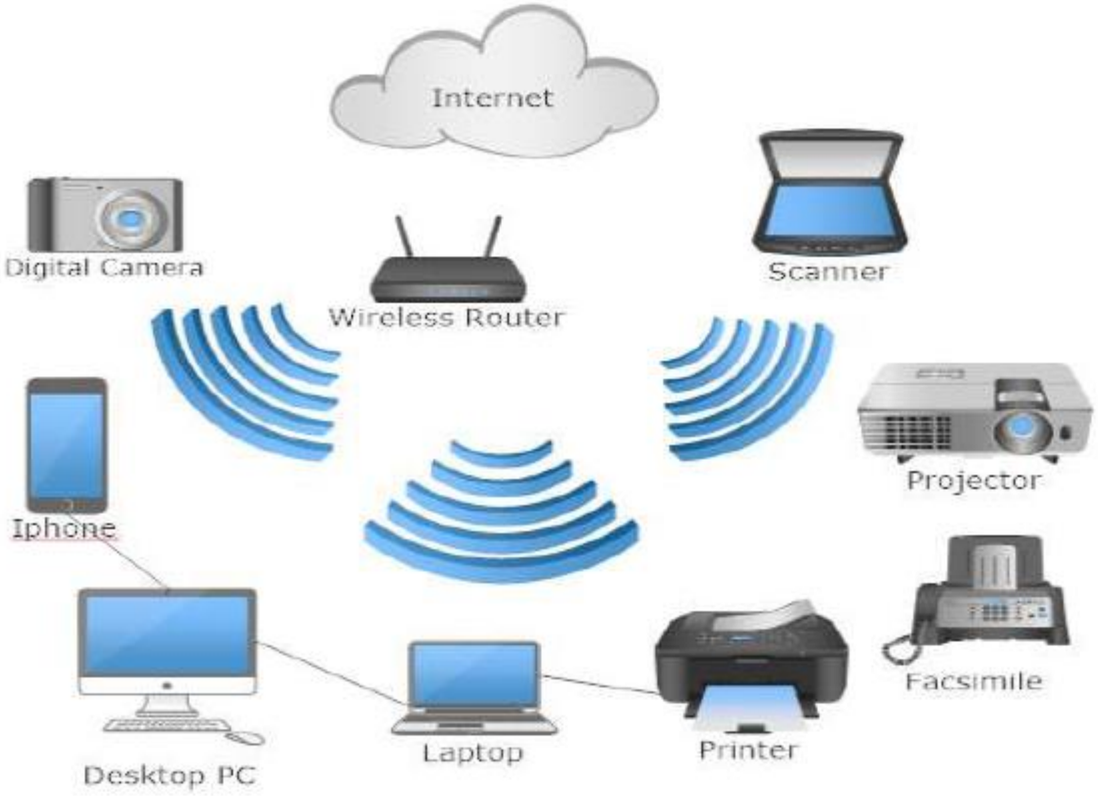


Figure 29: Wi-Fi connection

2.10.1.2 Wi-Fi and IoT

Wi-Fi has the advantage of addressing a very wide variety of profiles because of the proliferation of its family of standards. This means it will play a role in most IoT environments, alone or interworking with more specialized protocols, or with cellular. Some IoT applications, such as vehicular services, or video-based apps like connected security cameras, will need the bandwidth of the wireless broadband network, implemented to enable other requirements like low latency (In critical environments this may take place in a private network or slice).Wi-Fi is uniquely placed to support broadband and narrowband IoT applications from a common platform that can work at varying levels of power consumption and signal range. The next release of 5G standards, Release 16, will prioritize IoT-focused capabilities such as latency below four milliseconds and very high availability, to support emerging cases in the URLLC (ultra-reliable low latency communications) category [43].

2.10.2 4G

2.10.2.1 What is 4G?

4G is the short name for fourth-generation wireless, the stage of broadband mobile communications that supersedes 3G (third-generation wireless) and is the predecessor of 5G (fifth-generation wireless).

The 4G wireless cellular standard was defined by the International Telecommunication Union (ITU) and specifies the key characteristics of the standard, including transmission technology and data speeds.

Each generation of wireless cellular technology has introduced increased bandwidth speeds and network capacity. 4G users get speeds of up to 100 Mbps, while 3G only promised a peak speed of 14 Mbps.

With 4G download speeds, wireless users can stream high-definition video and audio. 4G also enables wireless broadband, which provides a way for users to get internet connectivity without the need for a fixed, wired connection from an internet service provider (ISP) [44].

2.10.2.2 4G LTE

In telecommunications, Long-Term Evolution is a standard for wireless broadband communication for mobile devices and data terminals, based on the GSM/EDGE and UMTS/HSPA standards.

2.10.2.3 Using 4g for IoT connectivity

Connectivity for IoT projects can be achieved in several ways – Ethernet, Wi-Fi and ZigBee are just some of them, but one of the most powerful and adaptable methods is to use existing cellular networks provided by Mobile Network Operators (MNOs). Using 4G for IoT is one of the leading options. Although 5G is grabbing all the headlines right now, its predecessor, 4G, is still a highly capable technology and in many ways is ideal for offering IoT connectivity for many types of use cases. In fact, data from ABI Research shows that 4G continues to dominate in the IoT connectivity space, with more than 60% of IoT modules using the technology [45].

2.10.3 Ethernet technology

The IOT devices designed were connecting with the internet network via Wi-Fi. Wi-Fi is a wireless standard. It is commonly used to provide data communication among mobile devices and the internet network. Another common standard used for networking devices and computers is Ethernet.

Ethernet is most commonly used Local Area Network (LAN) technology. It provides a wired communication to connect the devices to the Internet. It is a **physical and link layer protocol** in the TCP/IP stack which describes how the networked devices share their data to the computer or other network devices through the physical medium. It is based on IEEE 802.3 standard. Within an IOT system, Ethernet can be used to connect stationary or fixed IOT devices. For example, the Ethernet cables are used for connecting the computers with the routers to provide the Internet connectivity. The Ethernet cable serves as wired medium for connecting the computer and the router [37].

Figure 30: Ethernet cables

2.11 Thesis layout

Throughout the next chapters the details of the project are shown. The third chapter describes the components used in the project and how they were used.

The programming and implementation are detailed in the fourth chapter. Then the results are shown and discussed.

Chapter 3: Selection of hardware and software

3.1 My proposition for the system architecture

I propose this architecture to show that IoT starts where traditional SCADA ends. This architecture shows how Data generated from SCADA systems acts as one of the data sources for IoT. SCADA's focus is more on Remote monitoring and control. IoT's focus is on data analytics to improve organizations productivity

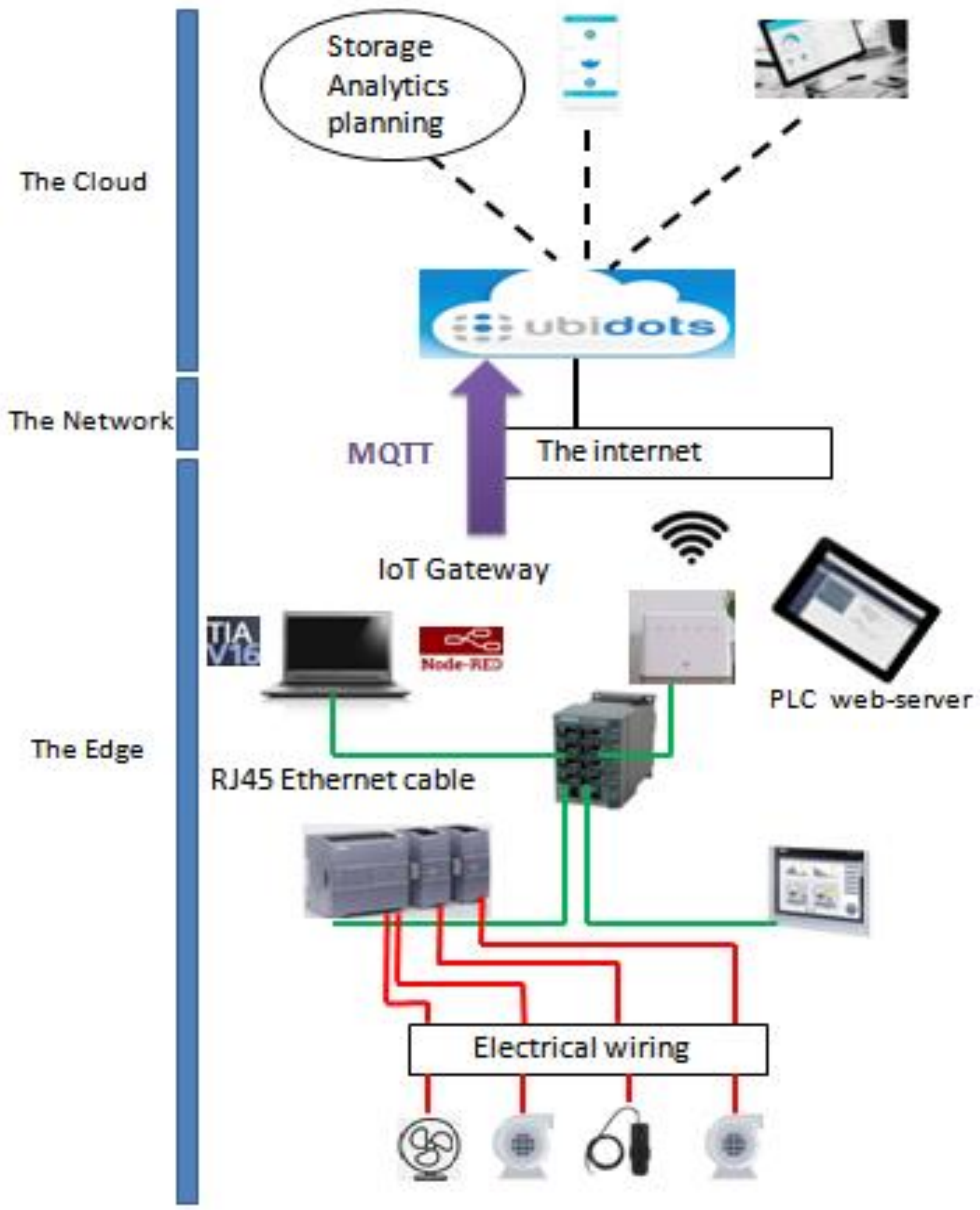


Figure 31: Proposed system architecture

3.1.1 The Edge

The Internet-of-Things (IoT) edge is where Level sensor and Actuators (pumps- and fan) communicate real-time data to a network. IoT edge computing solves latency issues associated with the cloud, as data is processed closer to its point of origin.

3.1.2 The Network

4G LTE currently holds the key position for international IoT connectivity, linking up automobiles, elevator and more

An IoT gateway is a physical device or virtual platform that connects sensors IoT modules and smart devices to the cloud

Node-RED was designed to realize IoT applications, data acquisition, local processing, and cloud integration without programming knowledge. With Node-RED, it is easy to create a connection between old and new industrial technologies in terms of IoT (Internet of Things)

3.1.3 The Cloud

The cloud refers to servers that are accessed over the Internet and the software and databases that run on those servers. Cloud servers are located in data centers all over the world. By using cloud computing, users and companies do not have to manage physical servers themselves or run software applications on their own machines

3.2 Hardware requirements:

3.2.1 SIMATIC S7-1200

The S7-1200 series is a line of programmable logic controllers (PLCs) that can control a variety of automation applications. Compact design, low cost, and a powerful instruction set make the S7-1200 a perfect solution for controlling a wide variety of applications. The S7-1200 models and the Windows-based programming tool give you the flexibility you need to solve your automation problems [52]

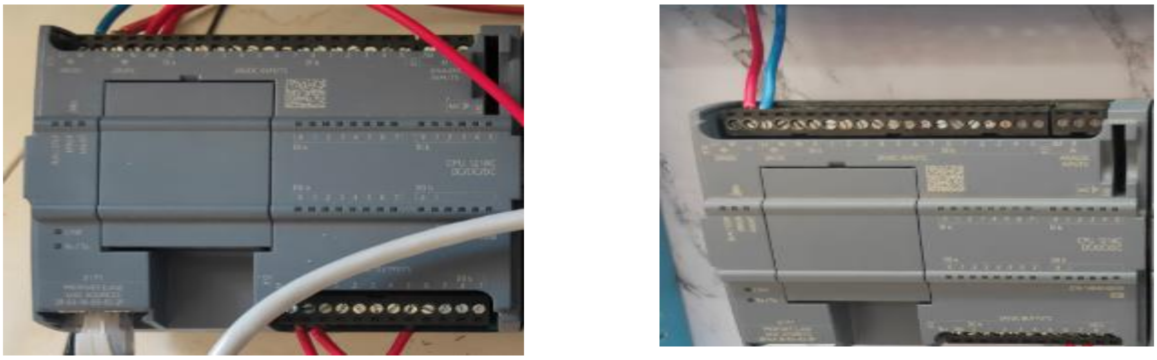
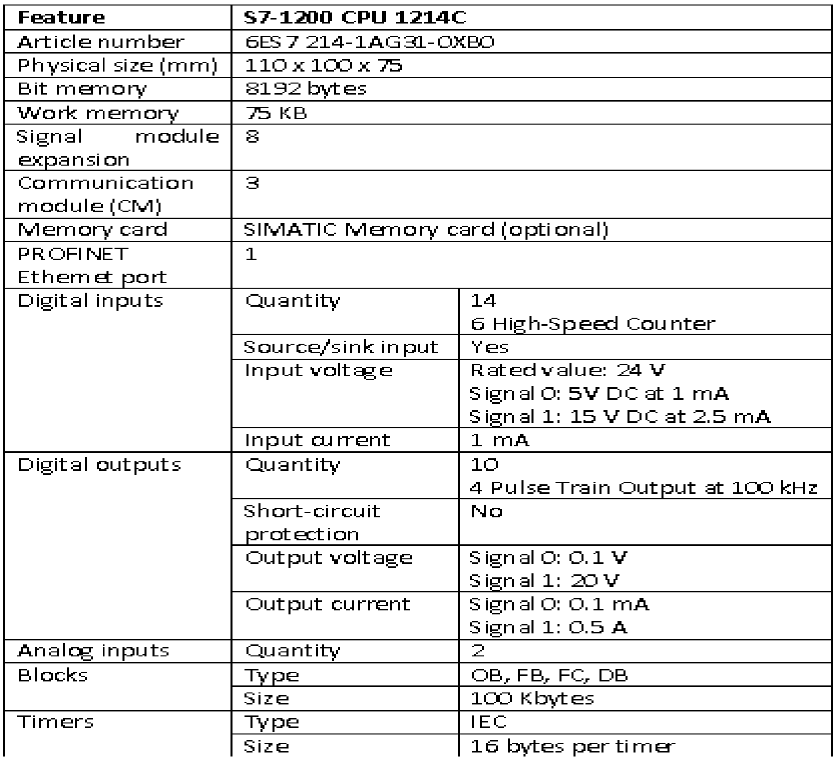


Figure 32: SIMATIC S7-1200

The S7-1200 controller provides the flexibility and power to control a wide variety of devices in support of your automation needs. The compact design, flexible configuration, and powerful instruction set combine to make the S7-1200 a perfect solution for controlling a wide variety of applications. The CPU combines a microprocessor, an integrated power supply, input and output circuits, built-in PROFINET, high-speed motion control I/O, and on-board analog inputs in a compact housing to create a powerful controller. After you download your program, the CPU contains the logic required to monitor and control the devices in your application. The CPU monitors the inputs and changes the outputs according to the logic of your user program, which can include Boolean logic, counting, timing, complex math operations, and communications with other intelligent devices. The CPU provides a PROFINET port for communication over a PROFINET network. Additional modules are available for communicating over PROFIBUS, GPRS, RS485 or RS232 networks [51].



Feature	**S7-1200 CPU 1214C**	
Article number	6ES 7 214-1AG31-OXBO	
Physical size (mm)	110 x 100 x 75	
Bit memory	8192 bytes	
Work memory	75 KB	
Signal module expansion	8	
Communication module (CM)	3	
Memory card	SIMATIC Memory card (optional)	
PROFINET Ethernet port	1	
Digital inputs	Quantity	14 6 High-Speed Counter
	Source/sink input	Yes
	Input voltage	Rated value: 24 V Signal O: 5V DC at 1 mA Signal 1: 15 V DC at 2.5 mA
	Input current	1 mA
Digital outputs	Quantity	10 4 Pulse Train Output at 100 kHz
	Short-circuit protection	No
	Output voltage	Signal O: 0.1 V Signal 1: 20 V
	Output current	Signal O: 0.1 mA Signal 1: 0.5 A
Analog inputs	Quantity	2
Blocks	Type	OB, FB, FC, DB
	Size	100 Kbytes
Timers	Type	IEC
	Size	16 bytes per timer

Table 2: SIMATIC S7-1200 Data Sheet

3.2.2 SM 1231 Analog input

A CPU processes information exclusively in digital format. The analog value is therefore converted into a bit pattern. Conversion is done using an integrated ADC (analog-to-digital converter) in the analog input module. For the CPU, this conversion always returns a 16-bit word for SIMATIC products. The ADC used digitalizes the analog signal and approximates its value with a stepped curve. Resolution and conversion speed are the most important ADC parameters [53].

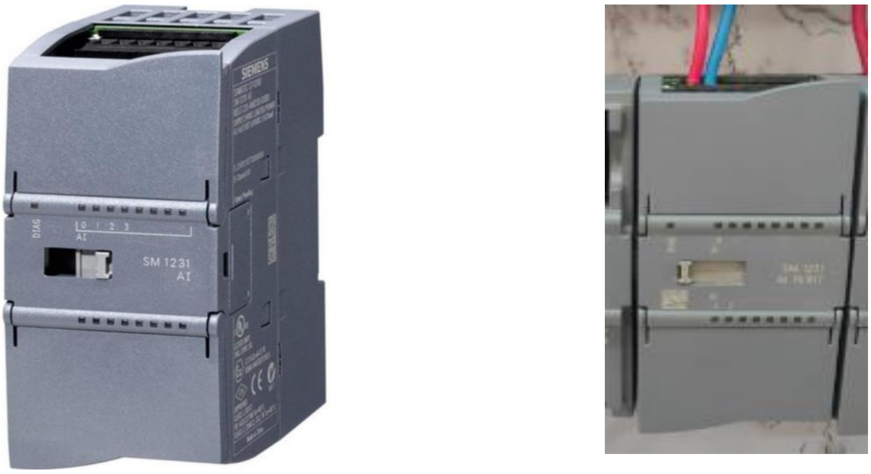
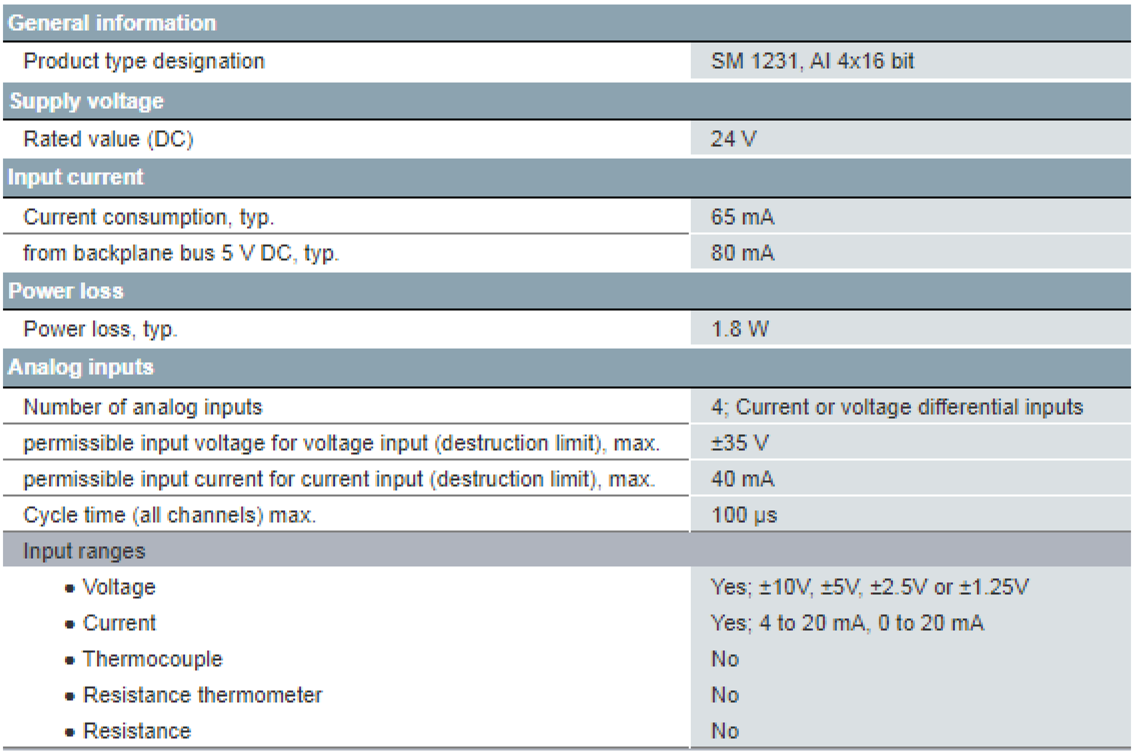


Figure 33: SM 1231



General information	
Product type designation	SM 1231, AI 4x16 bit
Supply voltage	
Rated value (DC)	24 V
Input current	
Current consumption, typ.	65 mA
from backplane bus 5 V DC, typ.	80 mA
Power loss	
Power loss, typ.	1.8 W
Analog inputs	
Number of analog inputs	4; Current or voltage differential inputs
permissible input voltage for voltage input (destruction limit), max.	±35 V
permissible input current for current input (destruction limit), max.	40 mA
Cycle time (all channels) max.	100 µs
Input ranges	
• Voltage	Yes; ±10V, ±5V, ±2.5V or ±1.25V
• Current	Yes; 4 to 20 mA, 0 to 20 mA
• Thermocouple	No
• Resistance thermometer	No
• Resistance	No

Table 3: SM 1231 data sheet

3.2.3 SM 1232 Analog output

Once the CPU has processed the digital signal, an integrated DAC (digital-to-analog converter) in the analog output module converts the output signal to an analog current or voltage value. The resulting value of the output signal corresponds to the output value used to control the analog actuators [53].

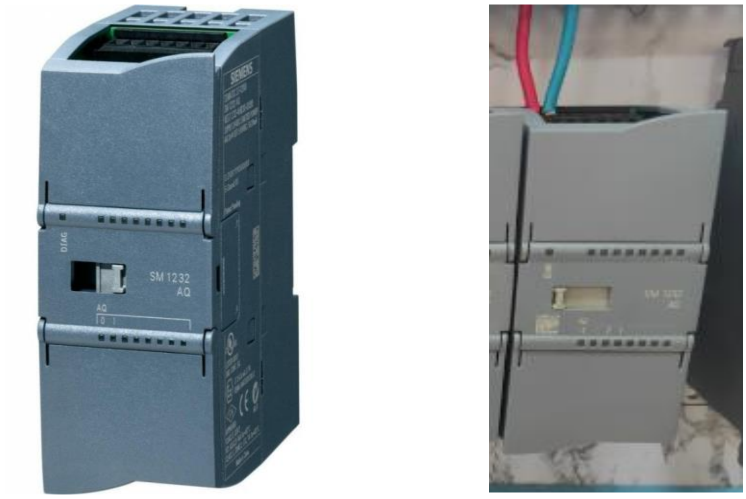
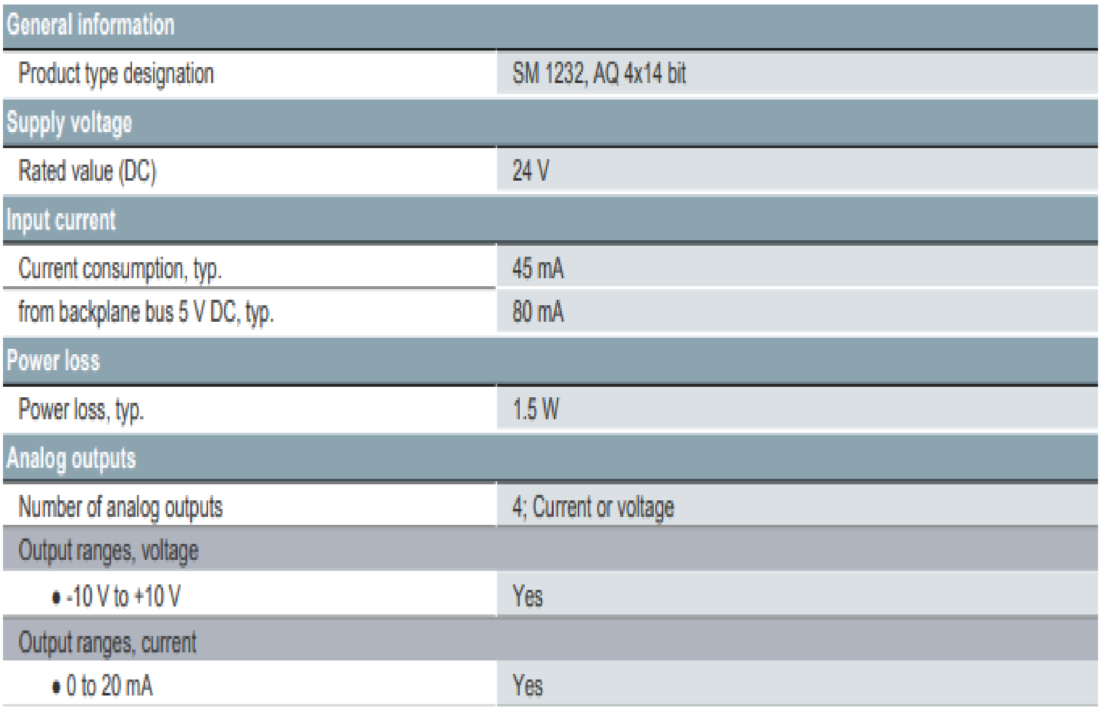


Figure 34: SM 1232



General information	
Product type designation	SM 1232, AQ 4x14 bit
Supply voltage	
Rated value (DC)	24 V
Input current	
Current consumption, typ.	45 mA
from backplane bus 5 V DC, typ.	80 mA
Power loss	
Power loss, typ.	1.5 W
Analog outputs	
Number of analog outputs	4; Current or voltage
Output ranges, voltage	
• -10 V to +10 V	Yes
Output ranges, current	
• 0 to 20 mA	Yes

Table 4: SM 1232 data sheet

3.2.4 PCT50:

PCT50 is a highly visible and easy to understand water level control process. It comprises two clear acrylic tanks; a process tank mounted above a sump tank. Water is pumped up to the process tank and drains back to the sump tank via two valves, one manually variable and the other switched by software. Each valve can also be fitted with different sized external orifices to change their characteristics. This flexible arrangement allows a wide range of control scenarios to be set up, including two fundamentally different processes Level control by proportionally varying pump speed (Inflow control) Level control by time-proportioned opening of a solenoid valve (Outflow control) When using inflow control, repeatable disturbances can be implemented using the solenoid valve, and varied by using different orifice sizes. When using outflow control the pump speed can be stepped to provide a full range of repeatable disturbances. These techniques allow direct comparison of different controller settings [54].

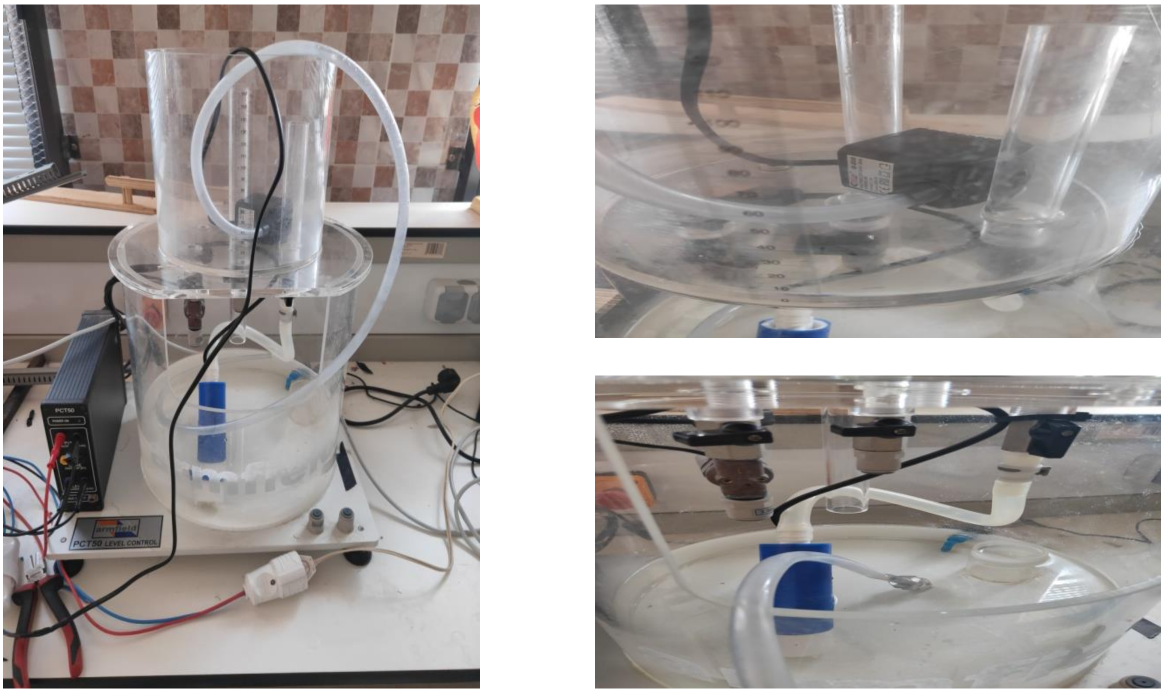


Figure 35: PCT50

Note:

The Electrovalve doesn’t work on my PCT50 this is why I used a pump to drain water back to the sump tank

3.2.5 An Electric fan

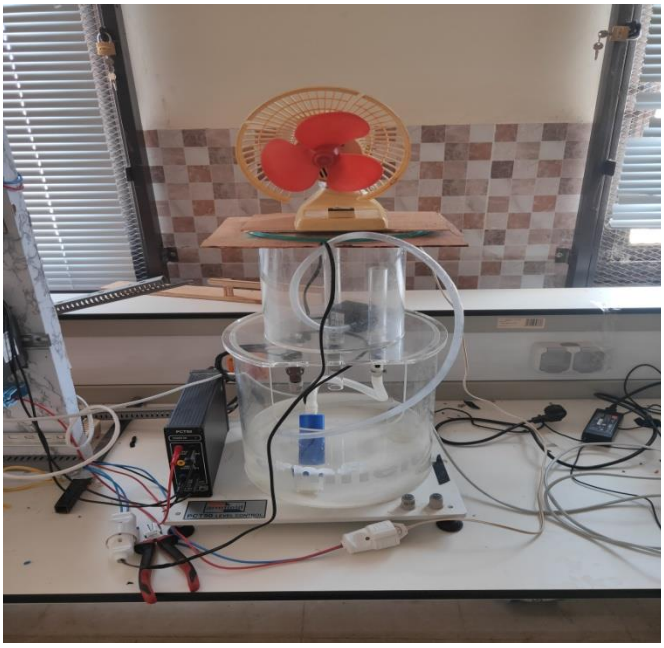


Figure 36: An Electric fan

3.2.6 Functions of the X-200 Switches

The SCALANCE X-200 Industrial Ethernet managed switches permit cost-effective configuration of Industrial Ethernet line, star or ring topologies. The possibility to create redundant rings makes it the perfect fit for networks in which high availability is required.. The IP30 protection, rugged cast metal housing, and RJ45 thread collars prove its capability to serve in industrial applications [55]

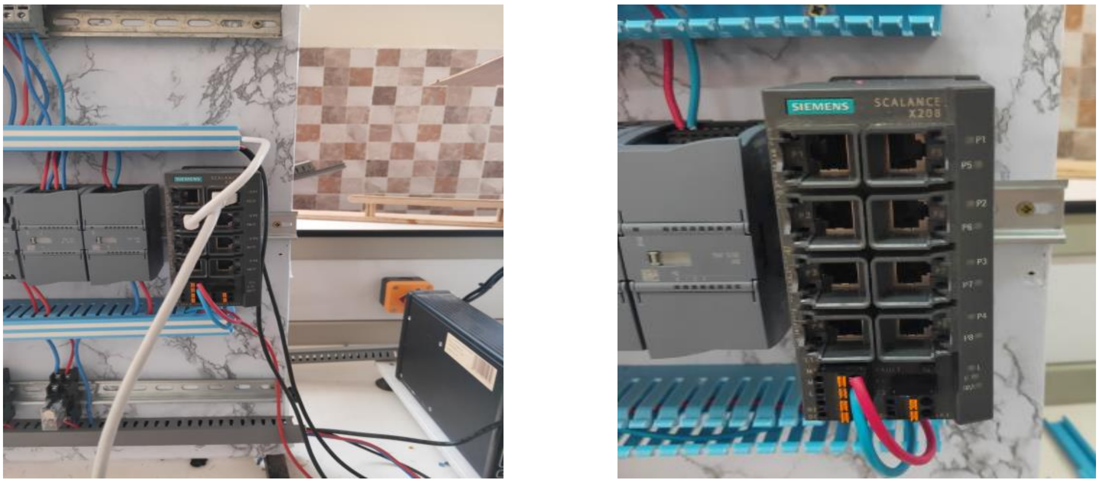


Figure 37: Switch Scalance X208

3.2.7 Siemens SITOP PSU100S 24v

SITOP PSU100S is a primary-clocked power supply for connection to a 1-phase AC line supply. An electronically regulated DC voltage that can be set via a potentiometer is available at the output of the device. The output of the device is isolated, no-load proof and short-circuits proof. The LED display indicates the operating status. The operating state of the device can be processed via the signaling contact. [56]

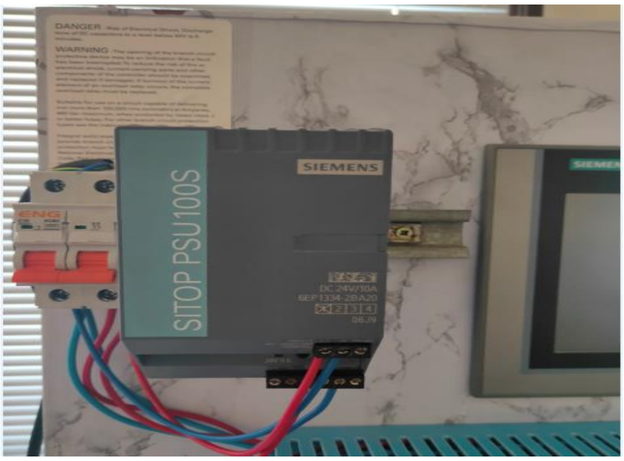


Figure 38: SITOP PSU100S

3.2.8 Relay 24v 220v

An electrical relay consists of an electromagnet and a spring loaded changeover contacts. When the electromagnet is switched ON/OFF with a DC supply, the spring loaded mechanism is corresponding pulled and released by this electromagnet, enabling a changeover across the end terminals of these contacts. An external electrical load connected across these contacts is subsequently switched ON/OFF in response to relay electromagnet switching.

The purpose of a relay 24v 220v is to convert a small electrical input into a high output. This conversion is carried out through electromagnetism

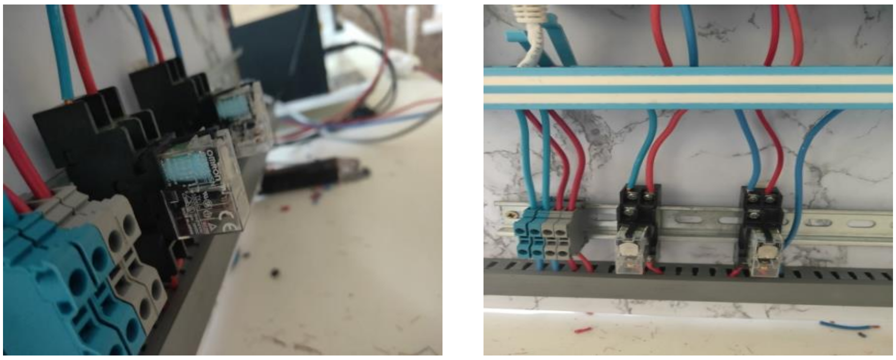


Figure 39: Relay 24/220V

3.2.9 Definition of Industrial HMI

Industrial HMI is a human-machine interface (HMI) designed for industrial applications. A human-machine interface encompasses both the hardware and software components necessary for the exchange of information between a human operator and a machine — in this case, an industrial system or industrial machinery or equipment. It’s the primary tool that industrial operators and line supervisors rely on for the coordination and control of manufacturing or industrial processes, typically in a plant environment [57].

The Siemens SIMATIC TP900 comfort panel has a 9inch widescreen TFT HMI LCD touch screen display. It uses X86 type processor. This innovative HMI is capable of coordinating and centrally shutting down its display via PROFI energy during break times, to reduce energy consumption, in comparison to previous SIMATIC panels [58].

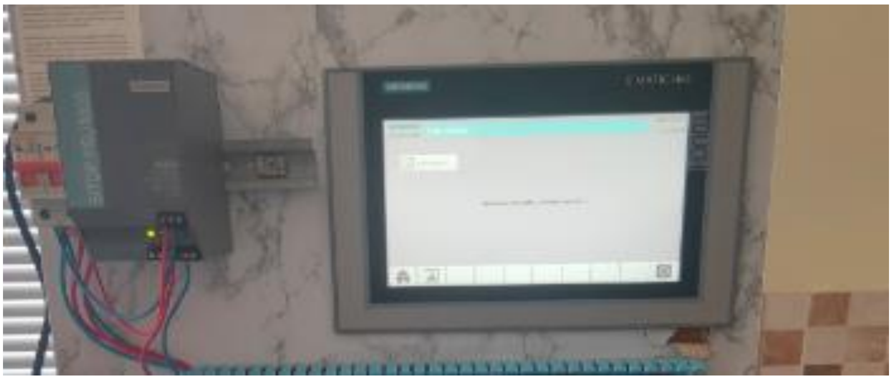


Figure 40: TP900 Comfort

3.2.10 Bipolar Circuit breaker

This bipolar circuit breaker is according to the EN 60934 standard and meets the NF F 62-001 standard (railway rolling stock).The most frequent uses for this circuit breaker are protection: Single phase alternating current circuits PHASE + NEUTRAL (230 Volts 50/60 Hz) [59].

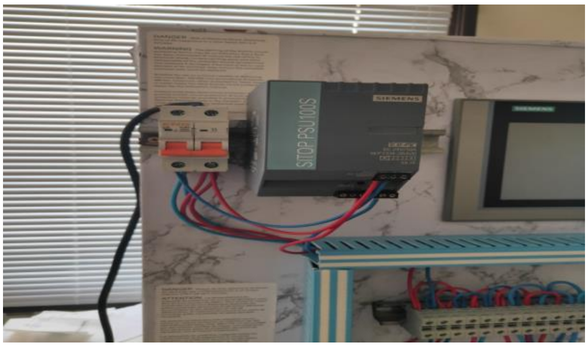


Figure 41: Bipolar circuit breaker

3.2.11 4G LTE CPE Wi-Fi Router

Huawei B310 is a new 4G LTE CPE with LTE Category 4 technology. Huawei B310 could achieve LTE connection of up to 150Mbps through its Gigabit Ethernet port. The B310 LTE CPE could cover smoother wireless network for 250 meters [60].

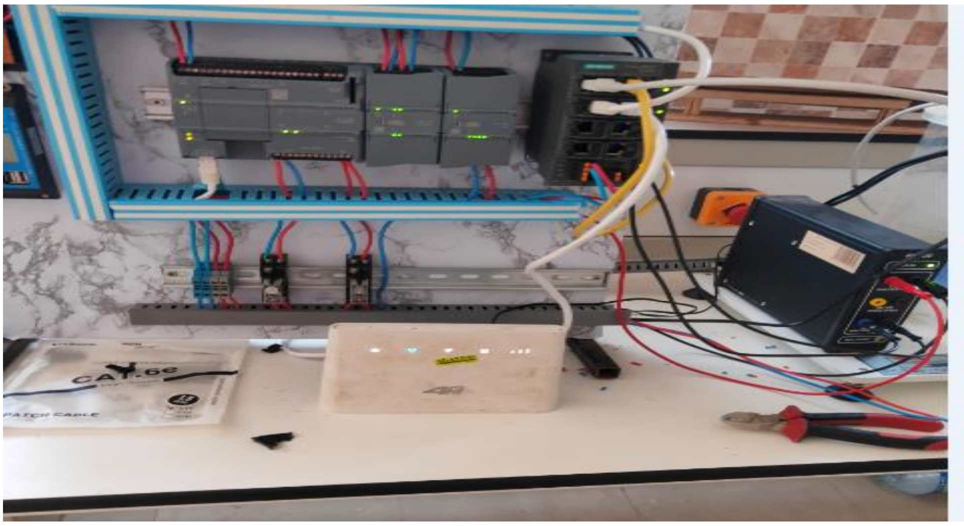


Figure 42: Huawei B310

3.2.12 Ethernet cable RJ45

An Ethernet cable is a common type of network cable used with wired networks. Ethernet cables connect devices such as PCs, routers, and switches within a local area network.

These physical cables are limited by length and durability. If a network cable is too long or of poor quality, it won't carry a good network signal. These limits are one reason there are different types of Ethernet cables that are optimized to perform certain tasks in specific situations [61].

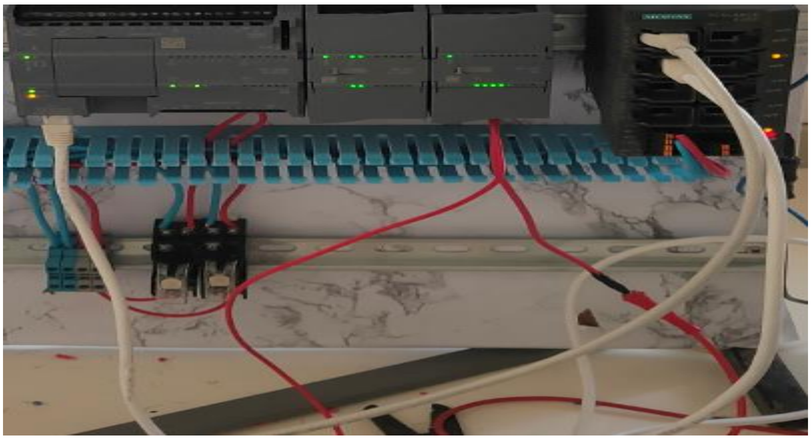
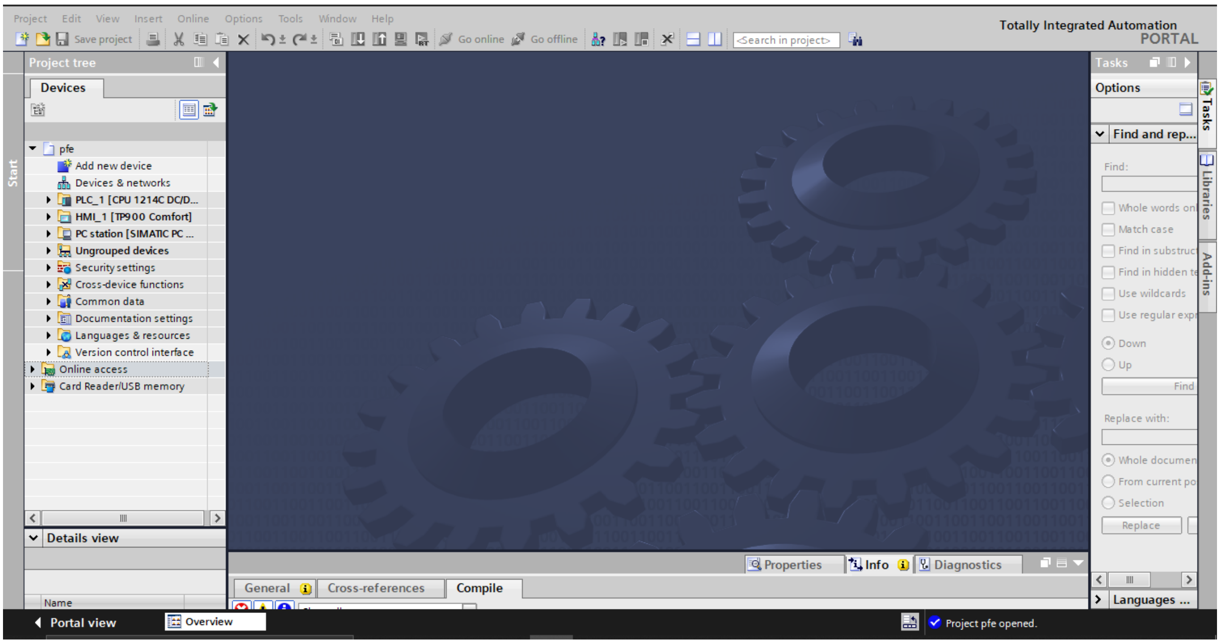


Figure 43: Ethernet cable connecting the plc to a switch

3.3 Software requirements

3.3.1 TIA Portal

Totally Integrated Automation (TIA) Portal is an all-in-one engineering platform that allows users to effectively commission, program, and diagnose Siemens automation hardware and software, all within a single software interface that allows for integrated system diagnostics, communication, and other features. As additional versions are released, newly released hardware and updated programming functions are added to the wide variety of products and options TIA Portal offers [62].



Project pfe opened.

Figure 44: TIA Portal Project view

3.3.1.1 Ladder Logic

Ladder logic was originally a written method to document the design and construction of relay racks as used in manufacturing and process control. Each device in the relay rack would be represented by a symbol on the ladder diagram with connections between those devices shown. In addition, other items external to the relay rack such as pumps, heaters, and so forth would also be shown on the ladder diagram **[63].**

Ladder logic has evolved into a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. Ladder logic is used to develop software for programmable logic controllers (PLCs) used in industrial control applications. The name is based on the observation that programs in this language resemble ladders, with two vertical rails and a series of horizontal rungs between them. While ladder diagrams were once the only available notation for recording programmable controller programs, today other forms are standardized in IEC 61131-3 (For example, as an alternative to the graphical ladder logic form, there is also a language more like C called Structured text within the IEC 61131-3 standard).

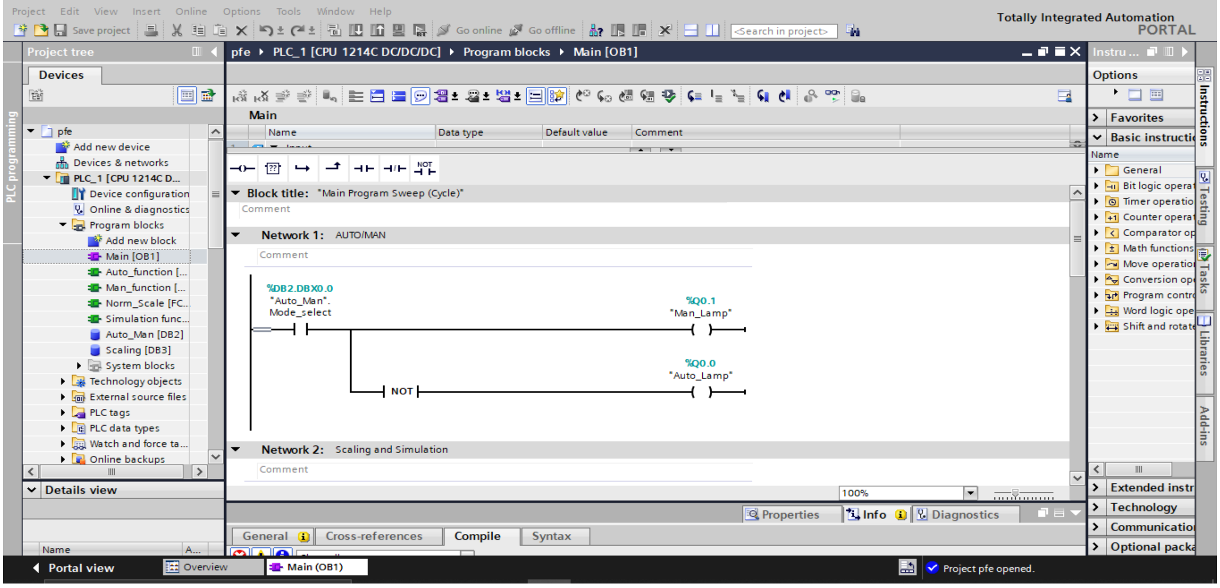
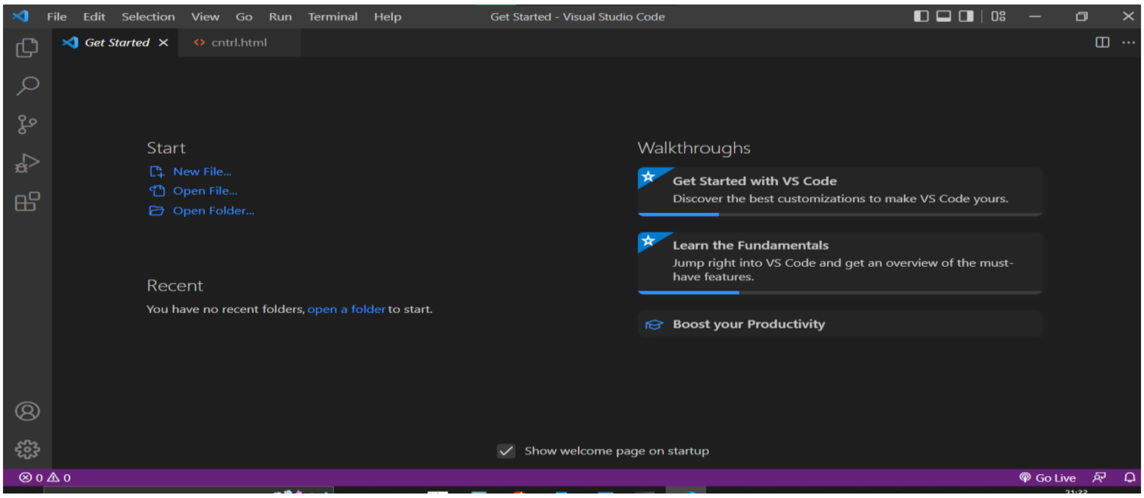


Figure 45: Laddar logic

3.3.2 Microsoft visual studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.



You have no recent folders, open a folder to start.

Discover the best customizations to make VS Code yours.

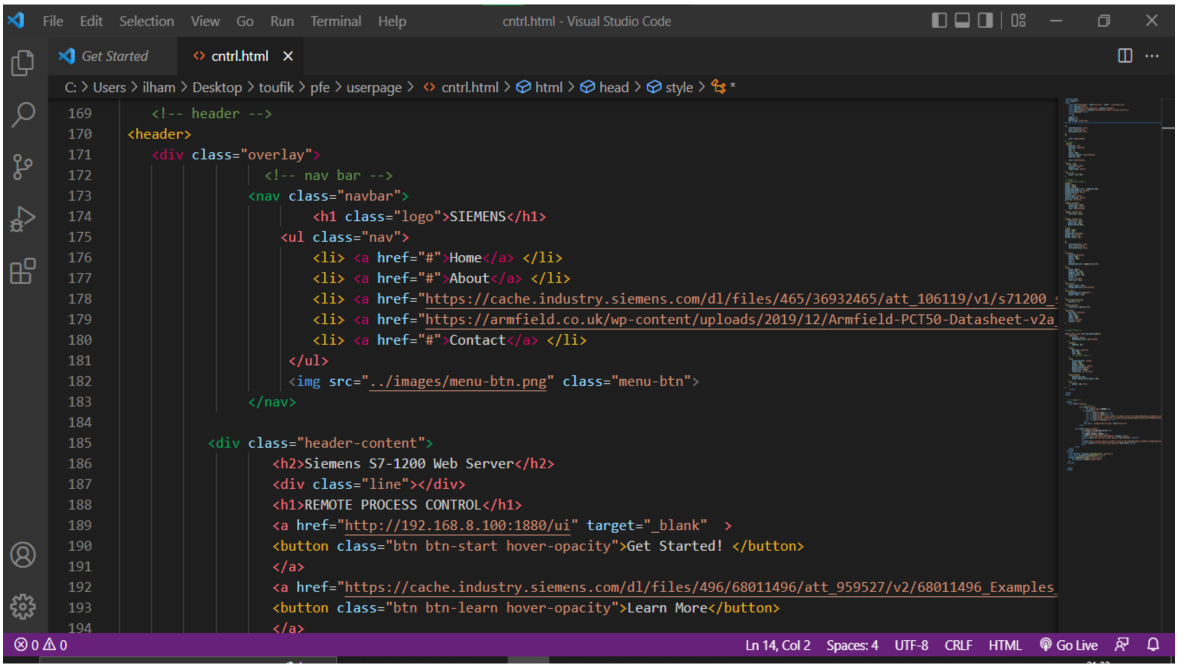
Jump right into VS Code and get an overview of the must-have features.

Figure 46: Microsoft visual studio

3.3.2.1 HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and final major HTML version that is a World Wide Web Consortium (W3C) recommendation. The current specification is known as the HTML Living Standard. It is maintained by the Web Hypertext Application Technology Working Group (WHATWG), a consortium of the major browser vendors (Apple, Google, Mozilla, and Microsoft).

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves, and rationalizes the markup available for documents and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications because it includes features designed with low-powered devices in mind [64].



File Edit Selection View Go Run Terminal Help
cntrl.html - Visual Studio Code
Get Started
cntrl.html
C: > Users > ilham > Desktop > toufik > pfe > userpage > cntrl.html > html > head > style > *
⊗ 0 ⚠ 0
Ln 14, Col 2 Spaces: 4 UTF-8 CRLF HTML Go Live

```html
   <!-- header -->
<header>
   <div class="overlay">
                 <!-- nav bar -->
               <nav class="navbar">
                       <h1 class="logo">SIEMENS</h1>
                   <ul class="nav">
                       <li> <a href="#">Home</a> </li>
                       <li> <a href="#">About</a> </li>
                       <li> <a href="https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_
                       <li> <a href="https://armfield.co.uk/wp-content/uploads/2019/12/Armfield-PCT50-Datasheet-v2a_
                       <li> <a href="#">Contact</a> </li>
                    </ul>
                    <img src="../images/menu-btn.png" class="menu-btn">
               </nav>

          <div class="header-content">
                  <h2>Siemens S7-1200 Web Server</h2>
                  <div class="line"></div>
                  <h1>REMOTE PROCESS CONTROL</h1>
                  <a href="http://192.168.8.100:1880/ui" target="_blank"  >
                  <button class="btn btn-start hover-opacity">Get Started! </button>
                  </a>
                  <a href="https://cache.industry.siemens.com/dl/files/496/68011496/att_959527/v2/68011496_Examples_
                  <button class="btn btn-learn hover-opacity">Learn More</button>
                  </a>
```

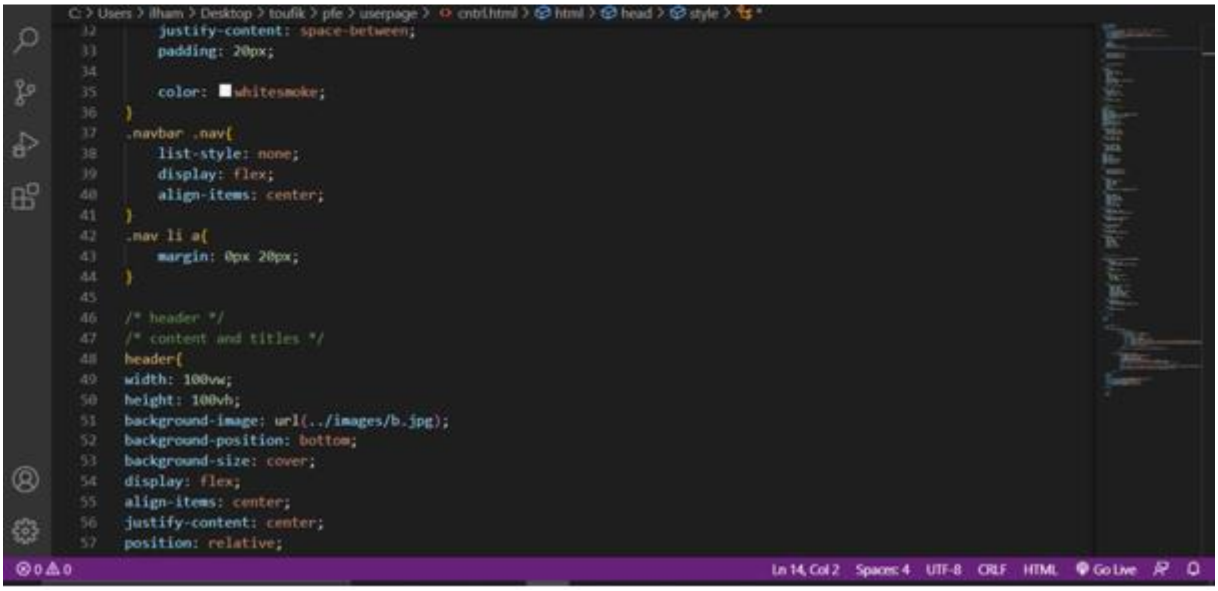
Figure 47: HTML code

3.3.2.2 CSS3

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

CSS3 is the latest version of the CSS specification. CSS3 adds several new styling features and improvements to enhance the web presentation capabilities [65]



C: > Users > ilham > Desktop > toufik > pfe > userpage > cntrl.html > html > head > style

```
        justify-content: space-between;
        padding: 20px;

        color: whitesmoke;
    }
    .navbar .nav{
        list-style: none;
        display: flex;
        align-items: center;
    }
    .nav li a{
        margin: 0px 20px;
    }

    /* header */
    /* content and titles */
    header{
    width: 100vw;
    height: 100vh;
    background-image: url(../images/b.jpg);
    background-position: bottom;
    background-size: cover;
    display: flex;
    align-items: center;
    justify-content: center;
    position: relative;
```

Figure 48: CSS code

3.3.2.3 JavaScript

JavaScript often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices [66].

3.3.3 Node-red

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things.

Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. Elements of applications can be saved or shared for re-use. The runtime is built on Node.js. The flows created in Node-RED are stored using JSON. Since version 0.14, MQTT nodes can make properly configured TLS connections.

In 2016, IBM contributed Node-RED as an open source JS Foundation project [67].

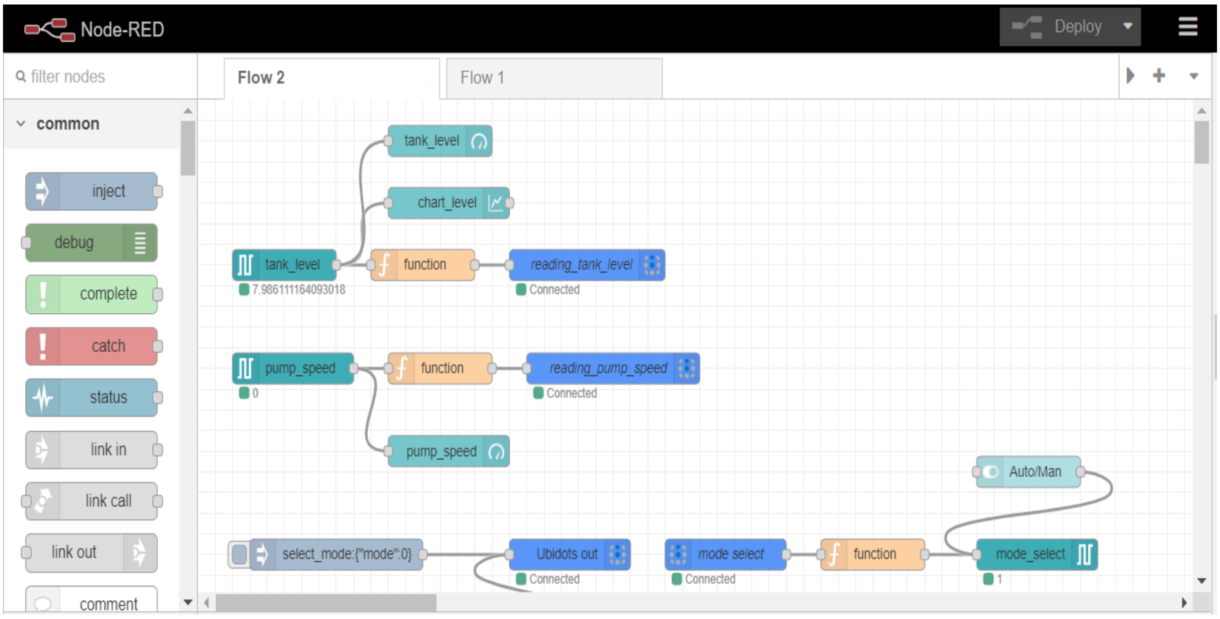
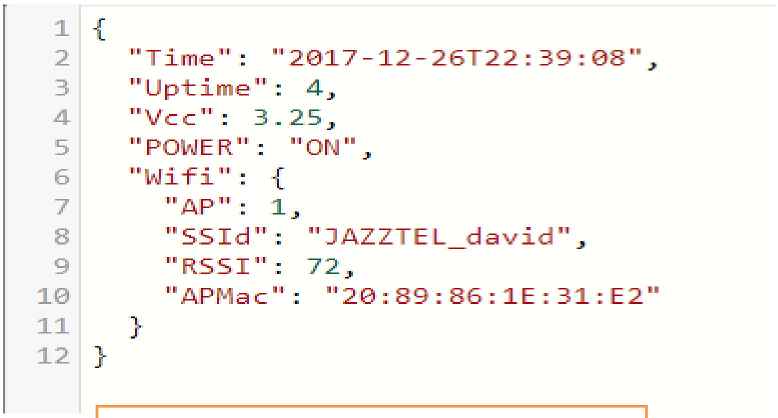


Figure 49: Node Red flow

3.3.3.1 JSON

JSON (JavaScript Object Notation, pronounced) is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays. It is a common data format with diverse uses in electronic data interchange, including that of web applications with servers [68].



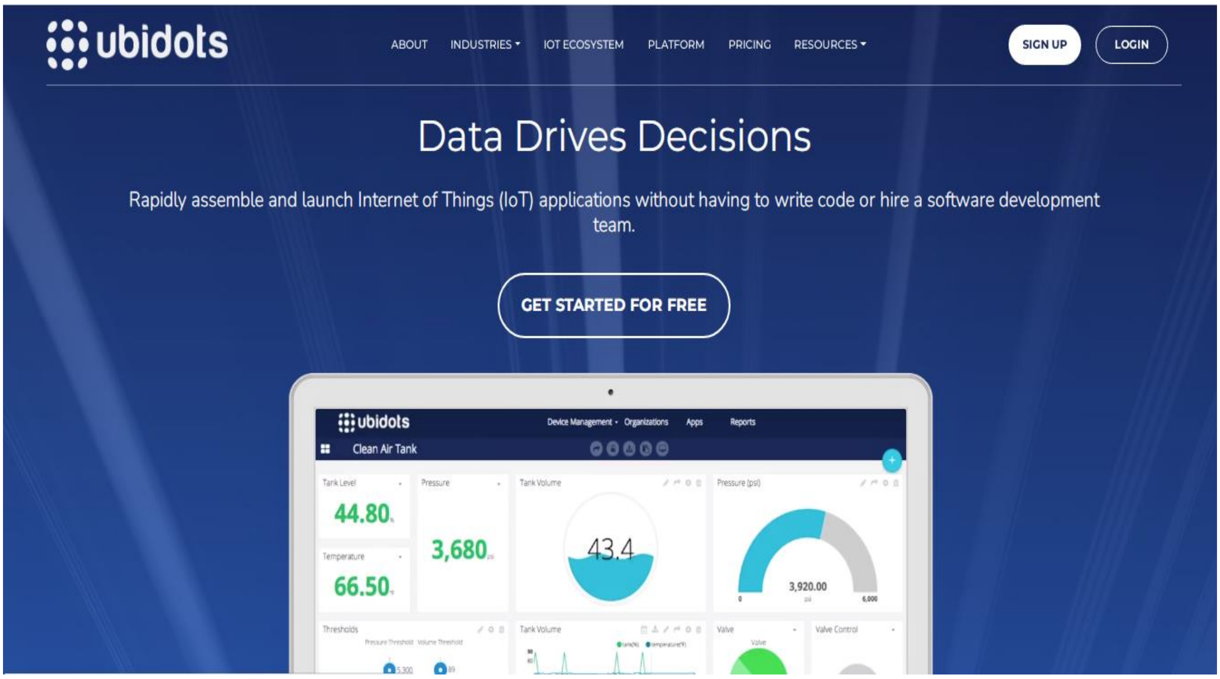
```
{
  "Time": "2017-12-26T22:39:08",
  "Uptime": 4,
  "Vcc": 3.25,
  "POWER": "ON",
  "Wifi": {
    "AP": 1,
    "SSId": "JAZZTEL_david",
    "RSSI": 72,
    "APMac": "20:89:86:1E:31:E2"
  }
}
```

Figure 50: JSON message

3.4 Cloud Service (Ubidots)

Ubidots is an IoT platform that empowers everything from prototypes, education, and research projects to scalable production-level IoT projects. It has different plans and a free version for non-commercial projects that includes most of the features of the paid version.

The Ubidots platform allows you to send and receive data to the cloud from internet-enabled devices. You can use a variety of protocols, including MQTT, HTTP (REST API), and TCP/UDP for data transfer. One of the features of Ubidots is that it can function as a broker, in other words, as data middleware. It also allows you to visualize and modify the transmitted data using drag and drop widgets, which can be integrated into Dashboards. This will enable you to have an IoT application that you can show to your team or customers in a short period of time [69].



Data Drives Decisions

Rapidly assemble and launch Internet of Things (IoT) applications without having to write code or hire a software development team.

Figure 51: Ubidots platform Home Page

Chapter 4: Model implementation

4.1 Wiring diagrams and layout of equipments

4.1.1 Power supply wiring diagram

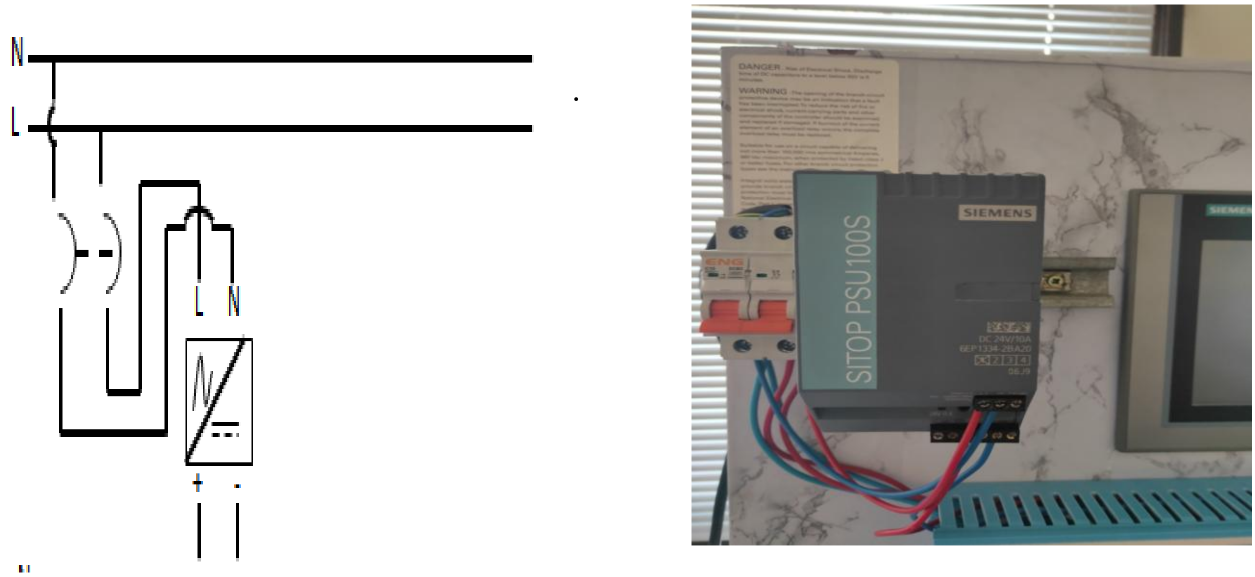


Figure 52: SITOP wiring

4.1.2 PLC and analog modules wiring diagram

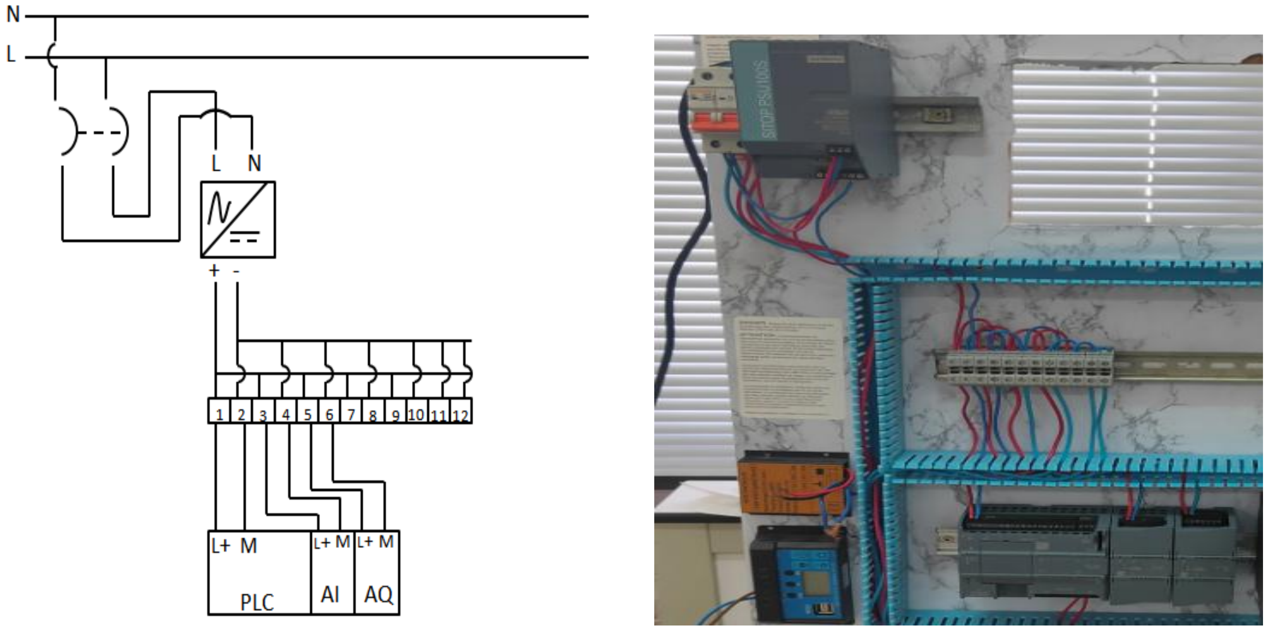
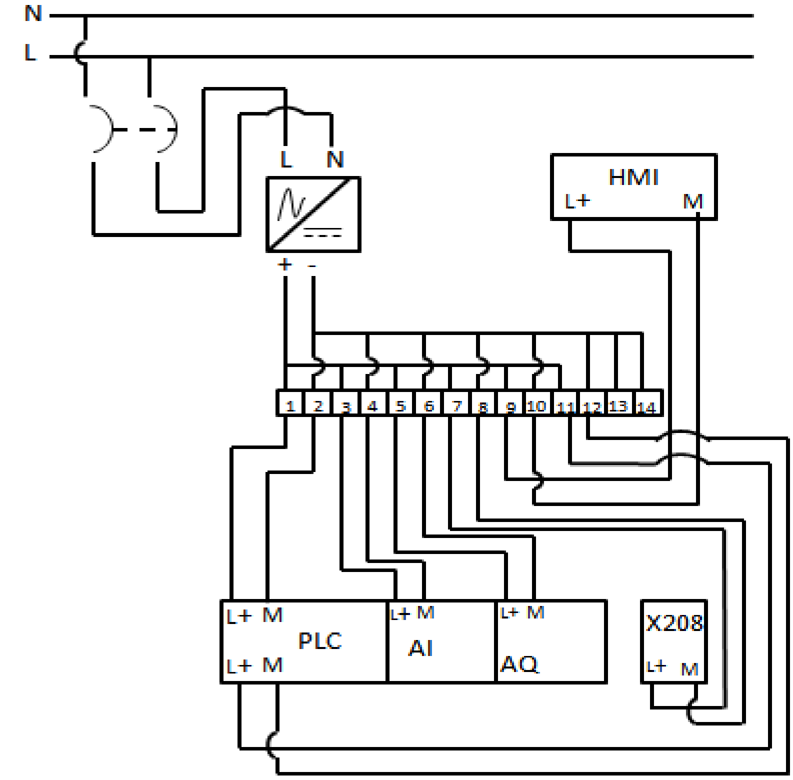


Figure 53: PLC and analog module wiring

4.1.3 PLC, Analog modules, HMI and Scalance wiring diagram



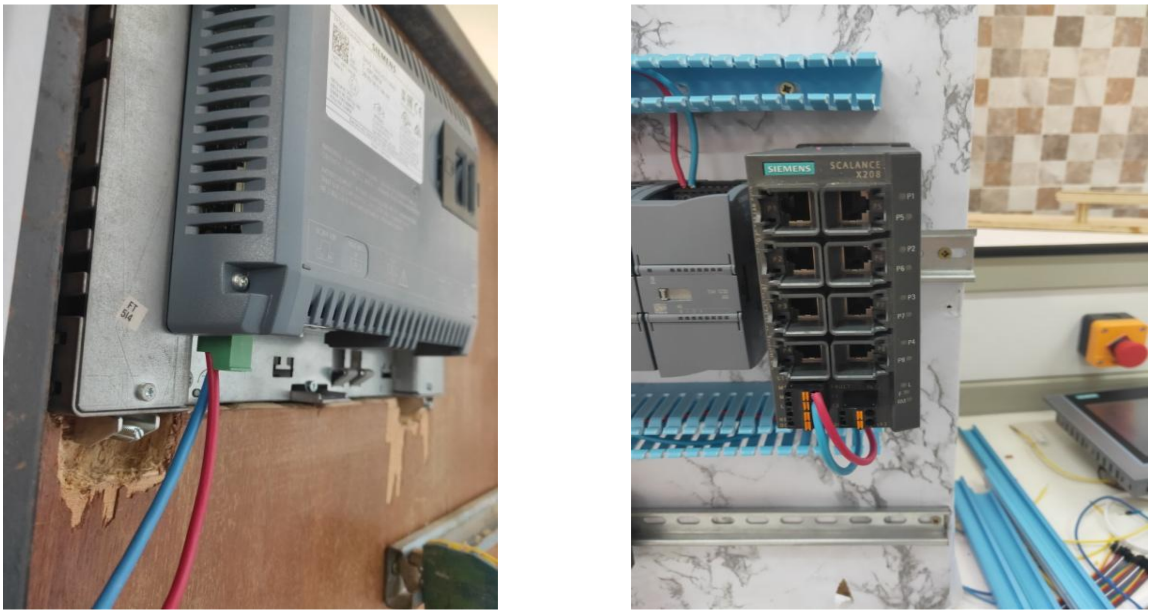
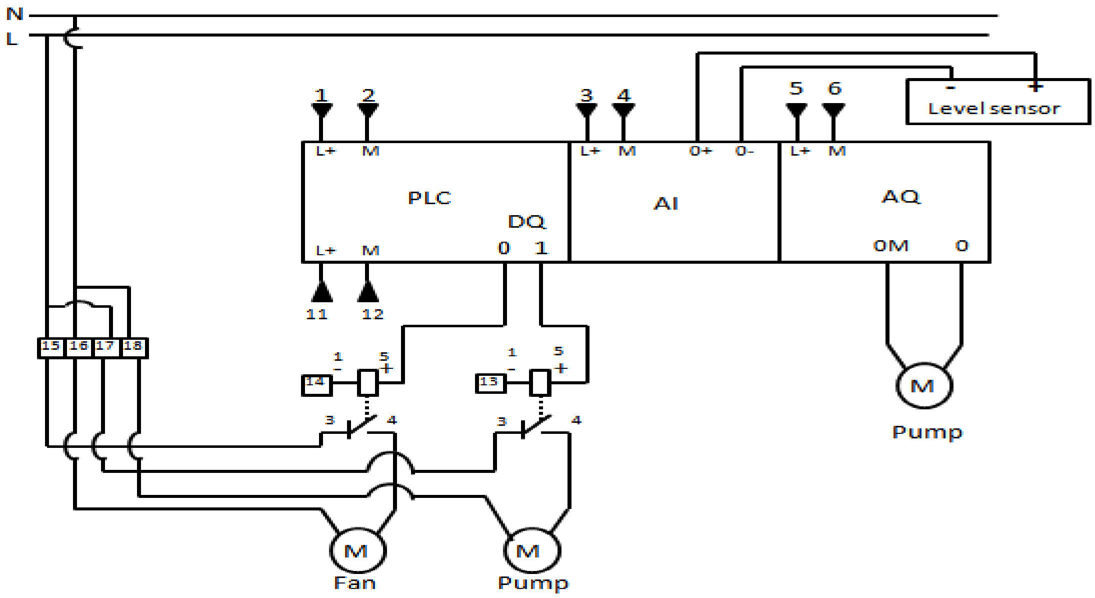


Figure 54: scalance, HMI wiring

4.1.4 Level sensor and actuators wiring diagram



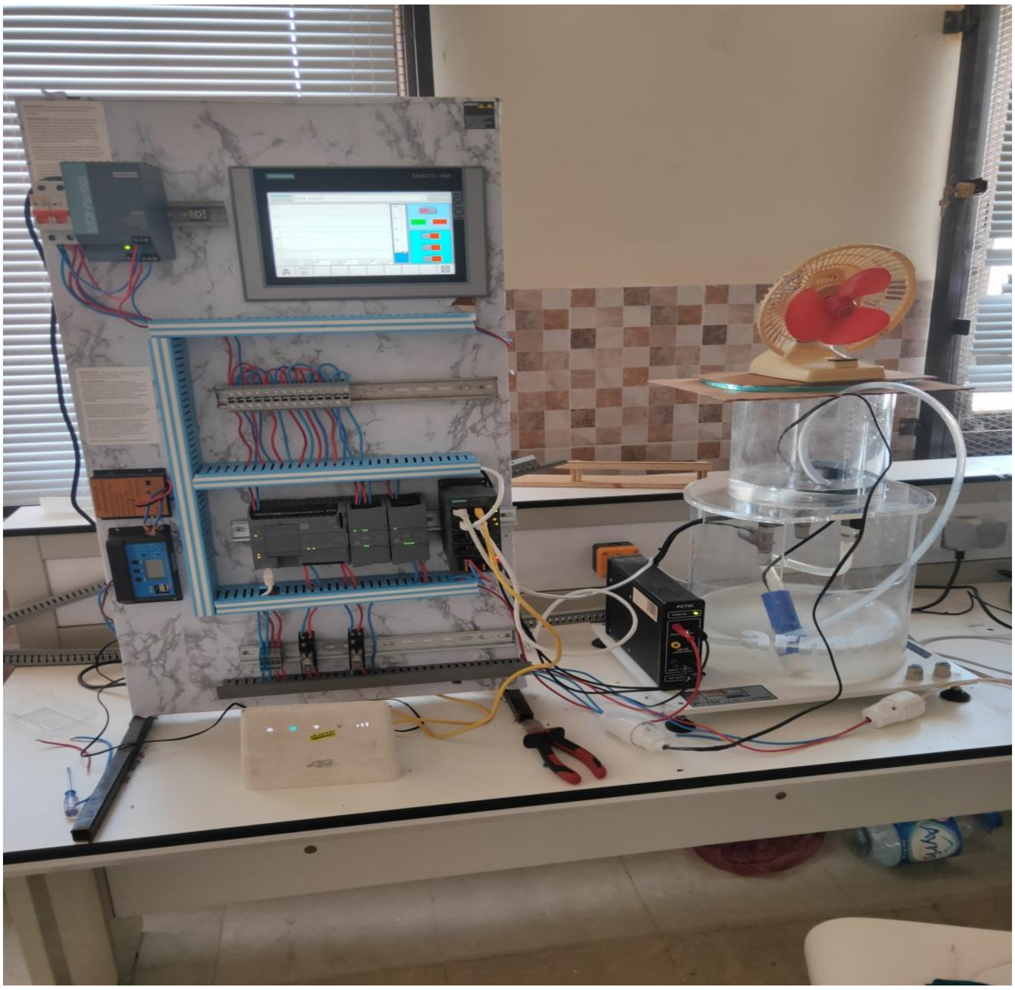
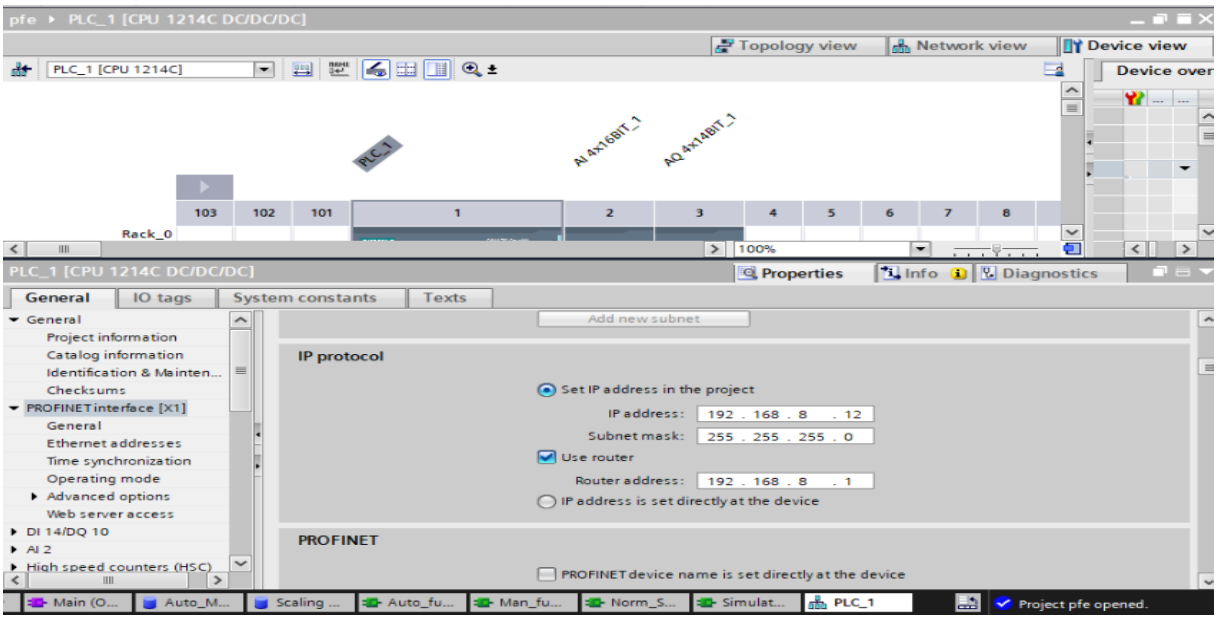


Figure 55: complete system wiring

4.2 PLC programming

4.2.1 PLC configuration

SIMATIC S7-1200 (CPU 1214C DC/DC/DC version 4.2) configuration



IP protocol

● Set IP address in the project

IP address: 192 . 168 . 8 . 12

Subnet mask: 255 . 255 . 255 . 0

☑ Use router

Router address: 192 . 168 . 8 . 1

○ IP address is set directly at the device

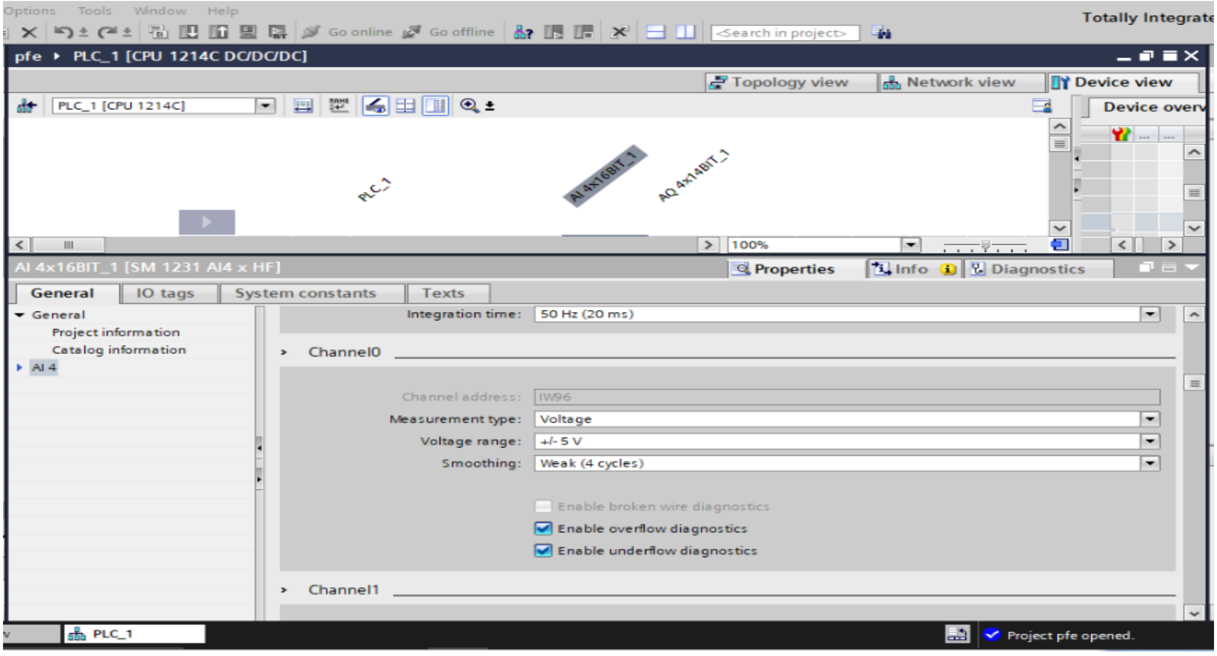
PROFINET

□ PROFINET device name is set directly at the device

Project pfe opened.

Figure 56: SIMATIC s7-1200 device configuration

4.2.2 Analog input module configuration: The analog input module needs to be configured to read -5/5v signal because the level sensor will send 0/5v analog signal



Integration time: 50 Hz (20 ms)

Channel0

Channel address: IW96

Measurement type: Voltage

Voltage range: +/- 5 V

Smoothing: Weak (4 cycles)

□ Enable broken wire diagnostics

☑ Enable overflow diagnostics

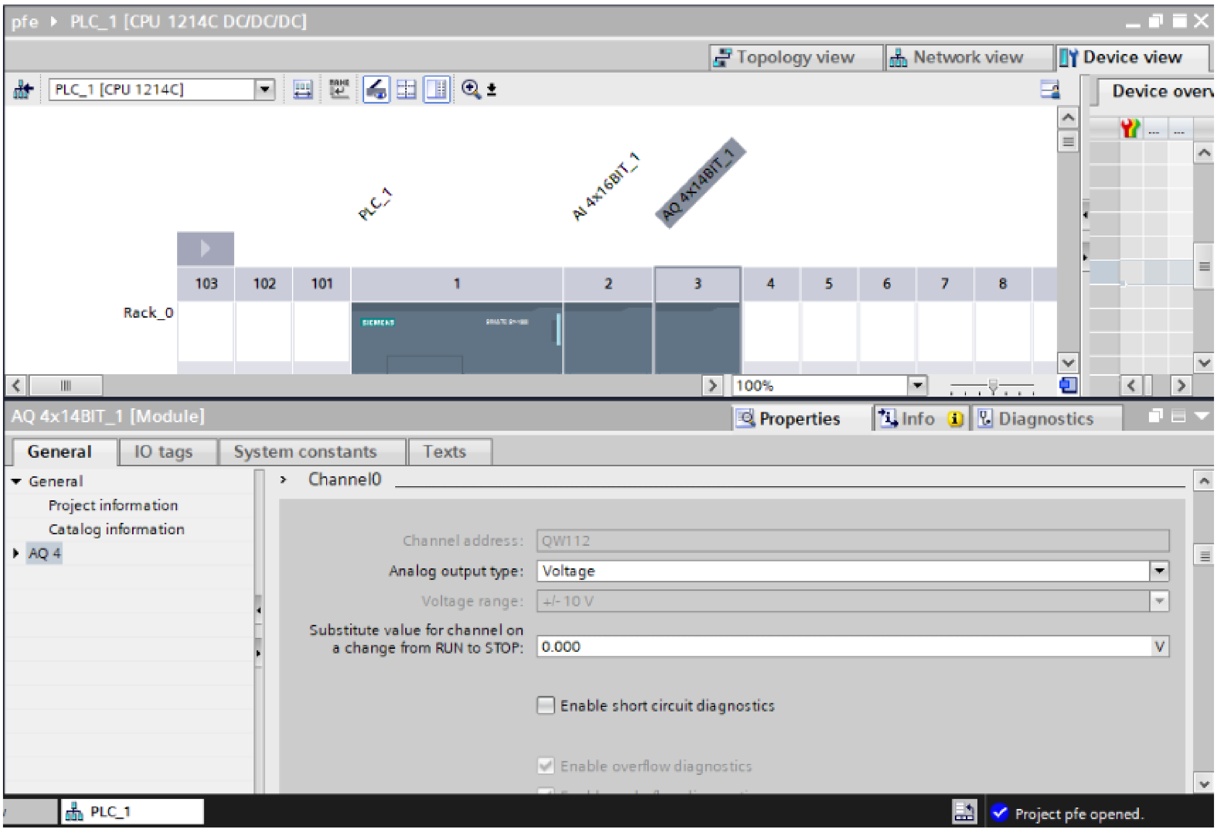
☑ Enable underflow diagnostics

Channel1

Project pfe opened.

Figure 57: Analog input module configuration

4.2.3 Analog output module configuration: The analog output module needs to be configured to send 0/10v signal because the pump needs to receive a voltage of 0 to 5v.



AQ 4x14BIT_1 [Module]

Channel0

Channel address: QW112

Analog output type: Voltage

Voltage range: +/- 10 V

Substitute value for channel on a change from RUN to STOP: 0.000

☐ Enable short circuit diagnostics

☑ Enable overflow diagnostics

Figure 58: Analog output module configuration

4.2.4 Data blocks

In contrast to logic blocks, data blocks contain no instructions. Rather, they serve as memory for user data. Data blocks thus contain variable data that is used by the user program. You can define the structure of global data blocks as required. Global data blocks store data that can be used by all other blocks (see Figure 1). Only the associated function block should access instance data blocks. The maximum size of data blocks varies depending on the utilized CPU.

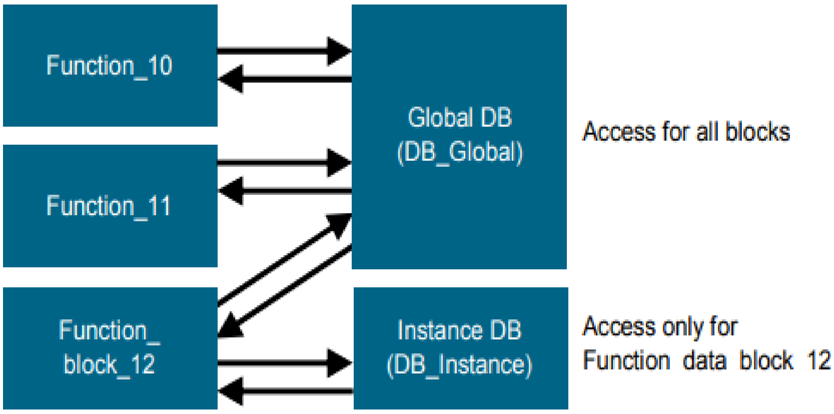
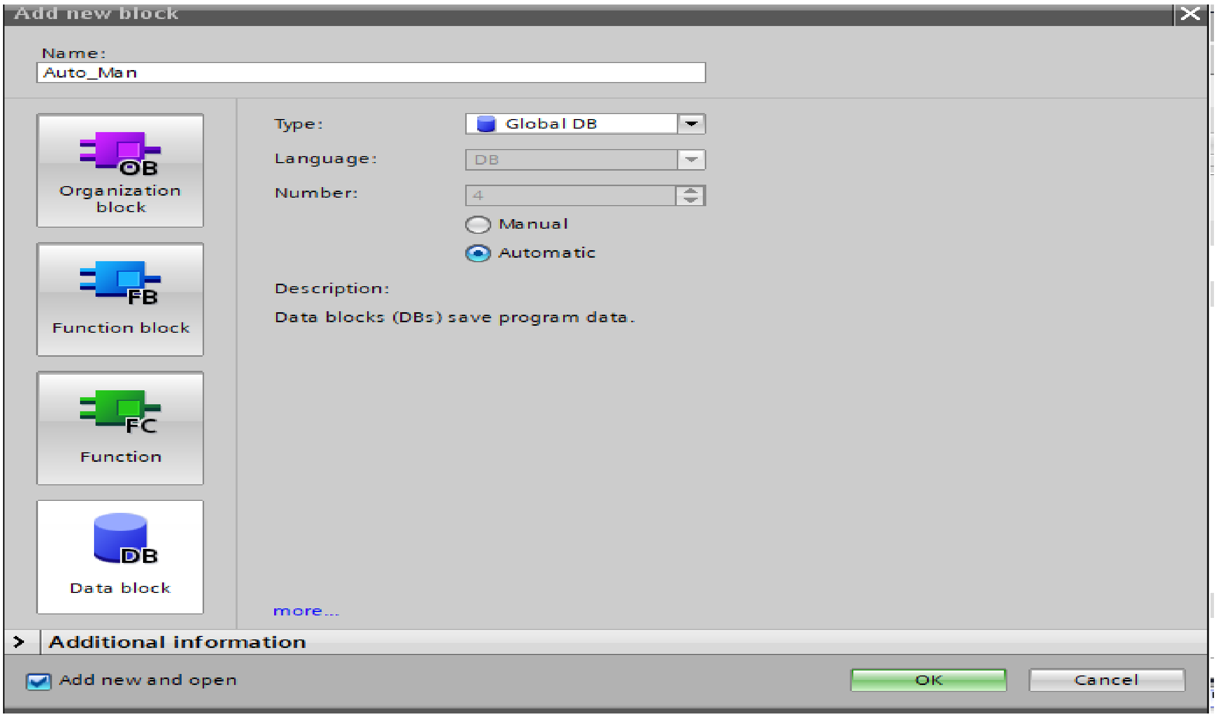


Figure 59: Data Blocks

4.2.4.1 Adding global data blocks



Name: Auto_Man

Type: Global DB

Language: DB

Number: 4

○ Manual

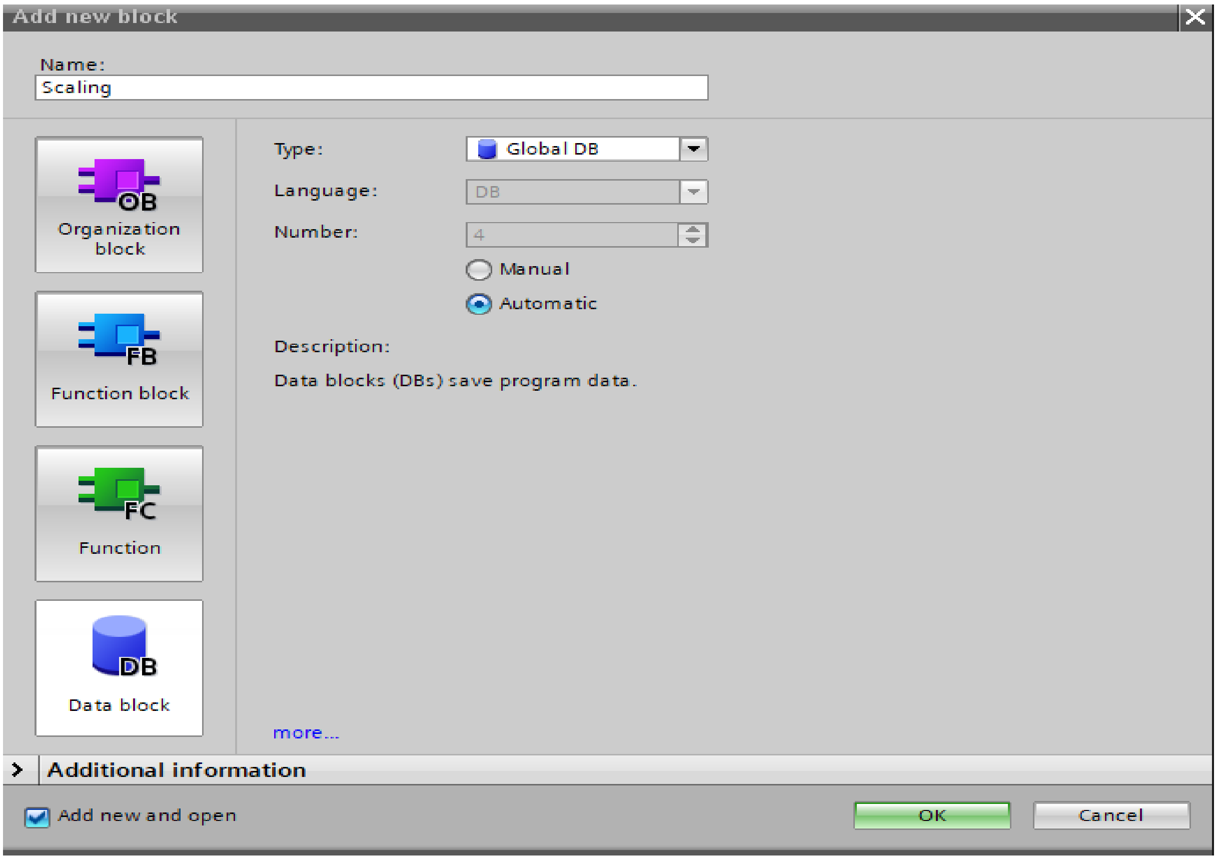
● Automatic

Description:

Data blocks (DBs) save program data.

☑ Add new and open

Figure 60: Adding a global data block (Auto_Man)



Name: Scaling

Type: Global DB

Language: DB

Number: 4

○ Manual

● Automatic

Description:

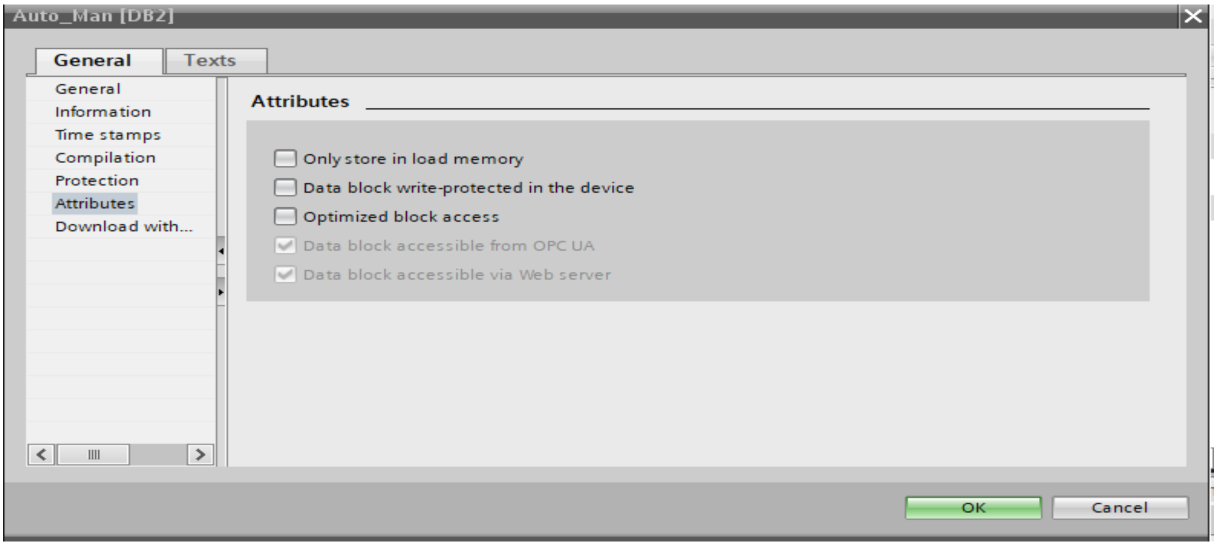
Data blocks (DBs) save program data.

☑ Add new and open

Figure 61: Adding a global data block (Scaling)

4.2.4.2 Standard data blocks

unchecking optimized block access



Attributes

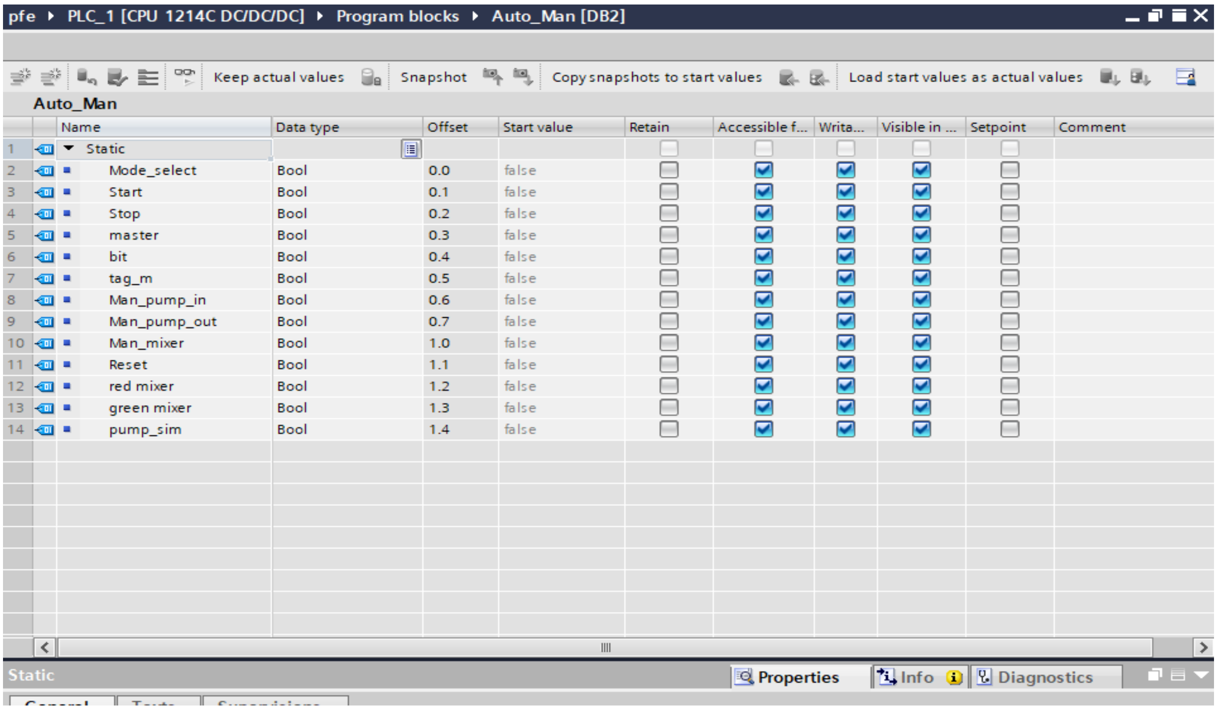
□ Only store in load memory
□ Data block write-protected in the device
□ Optimized block access
☑ Data block accessible from OPC UA
☑ Data block accessible via Web server

Figure 62: Standard data block

Note:

We are using standard data blocks because they give us the ability to know exactly where variables are stored so that we can configure S7 contrib nodes.

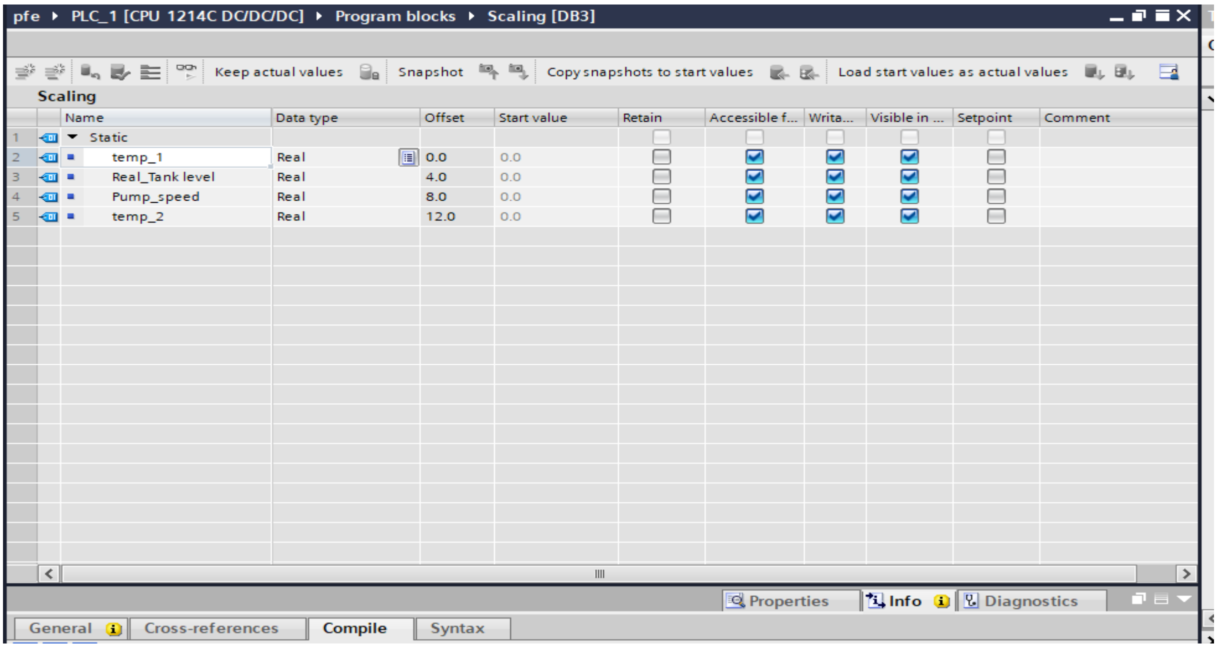
4.2.4.3 Declaring Variables


pfe ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Auto_Man [DB2]
Keep actual values
Snapshot
Copy snapshots to start values
Load start values as actual values
Static
Properties
Info
Diagnostics

Auto_Man

	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	▼ Static				□	□	□	□	□	
2	Mode_select	Bool	0.0	false	□	☑	☑	☑	□	
3	Start	Bool	0.1	false	□	☑	☑	☑	□	
4	Stop	Bool	0.2	false	□	☑	☑	☑	□	
5	master	Bool	0.3	false	□	☑	☑	☑	□	
6	bit	Bool	0.4	false	□	☑	☑	☑	□	
7	tag_m	Bool	0.5	false	□	☑	☑	☑	□	
8	Man_pump_in	Bool	0.6	false	□	☑	☑	☑	□	
9	Man_pump_out	Bool	0.7	false	□	☑	☑	☑	□	
10	Man_mixer	Bool	1.0	false	□	☑	☑	☑	□	
11	Reset	Bool	1.1	false	□	☑	☑	☑	□	
12	red mixer	Bool	1.2	false	□	☑	☑	☑	□	
13	green mixer	Bool	1.3	false	□	☑	☑	☑	□	
14	pump_sim	Bool	1.4	false	□	☑	☑	☑	□	

Figure 63: Data block 2 variables



pfe ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Scaling [DB3]

Keep actual values | Snapshot | Copy snapshots to start values | Load start values as actual values

Scaling

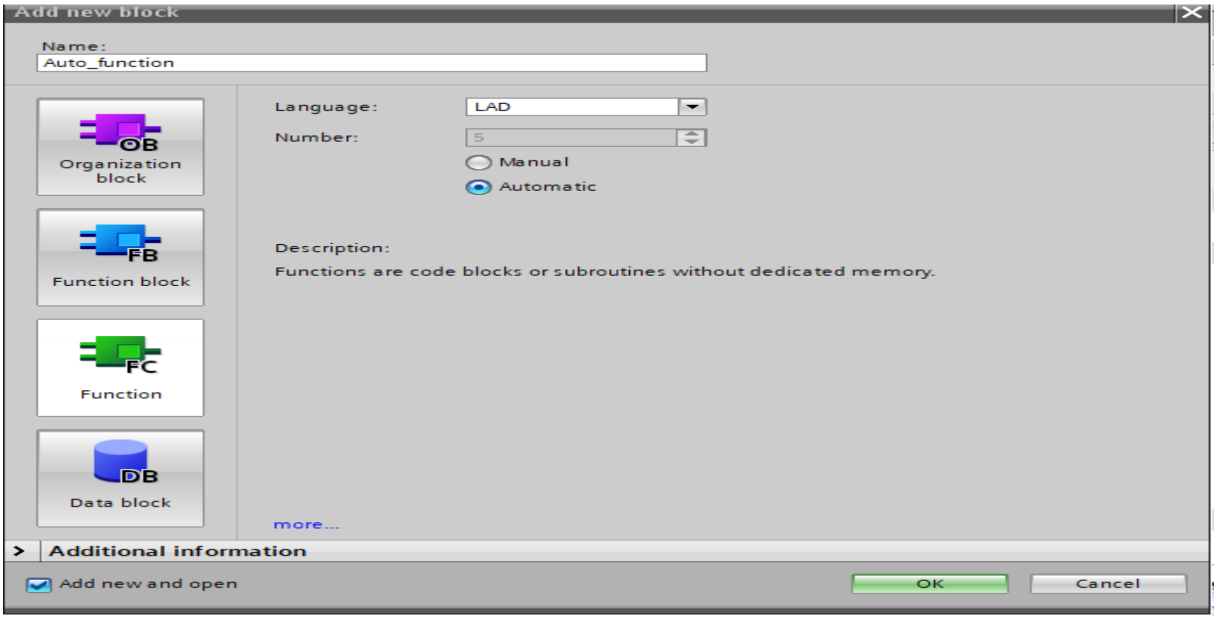
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	▼ Static				□	□	□	□	□	
2	▪ temp_1	Real	0.0	0.0	□	☑	☑	☑	□	
3	▪ Real_Tank level	Real	4.0	0.0	□	☑	☑	☑	□	
4	▪ Pump_speed	Real	8.0	0.0	□	☑	☑	☑	□	
5	▪ temp_2	Real	12.0	0.0	□	☑	☑	☑	□	

Figure 64: Data block 3 variables

4.2.5 Functions

Functions (FCs) are logic blocks without memory. They **have no data memory** in which values of block parameters can be stored. Therefore, all interface parameters must be connected when a function is called. To store data permanently, global data blocks must be created beforehand. A function contains a program that is executed whenever the function is called from another logic block.

4.2.5.1 Adding Functions



Add new block

Name:
Auto_function

Language: LAD

Number: 5

○ Manual

● Automatic

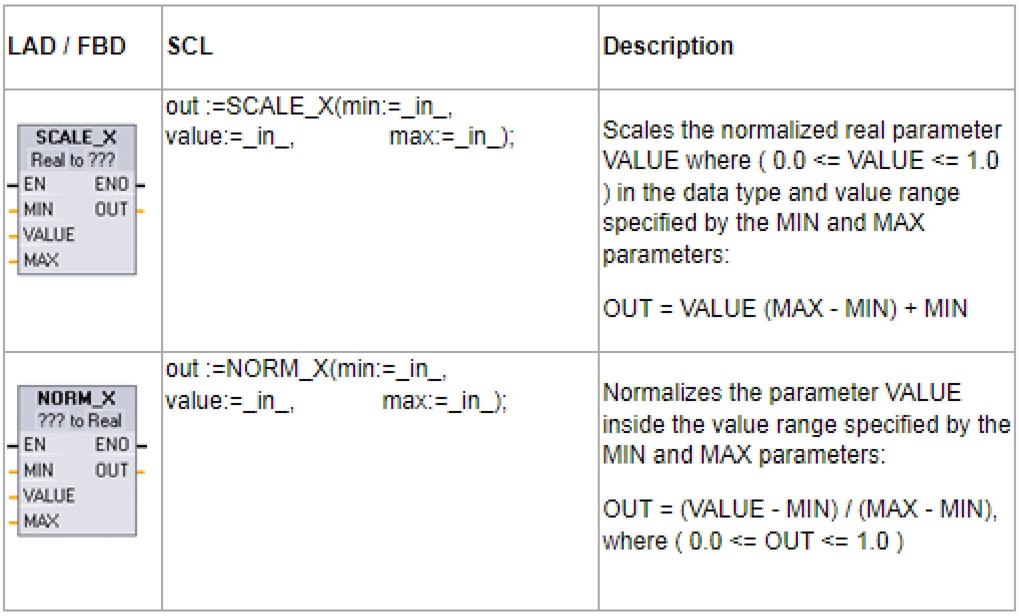
Description:
Functions are code blocks or subroutines without dedicated memory.

Figure 65: Adding a new function (Auto_function)

In the same way I added the Manual mode, scaling and simulation functions

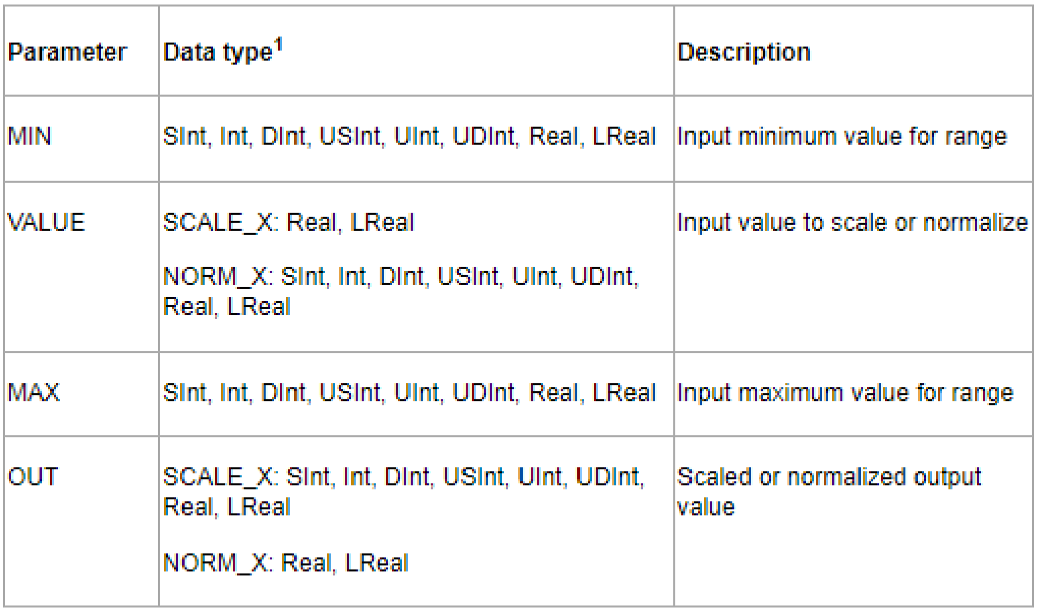
4.2.5.2 Scaling function:

The analog processing is done by using NORM_X (Normalize) and SCALE_X (Scale) functions



LAD / FBD	SCL	Description
SCALE_X Real to ??? EN ENO MIN OUT VALUE MAX	out :=SCALE_X(min:=_in_, value:=_in_, max:=_in_);	Scales the normalized real parameter VALUE where (0.0 <= VALUE <= 1.0) in the data type and value range specified by the MIN and MAX parameters: OUT = VALUE (MAX - MIN) + MIN
NORM_X ??? to Real EN ENO MIN OUT VALUE MAX	out :=NORM_X(min:=_in_, value:=_in_, max:=_in_);	Normalizes the parameter VALUE inside the value range specified by the MIN and MAX parameters: OUT = (VALUE - MIN) / (MAX - MIN), where (0.0 <= OUT <= 1.0)

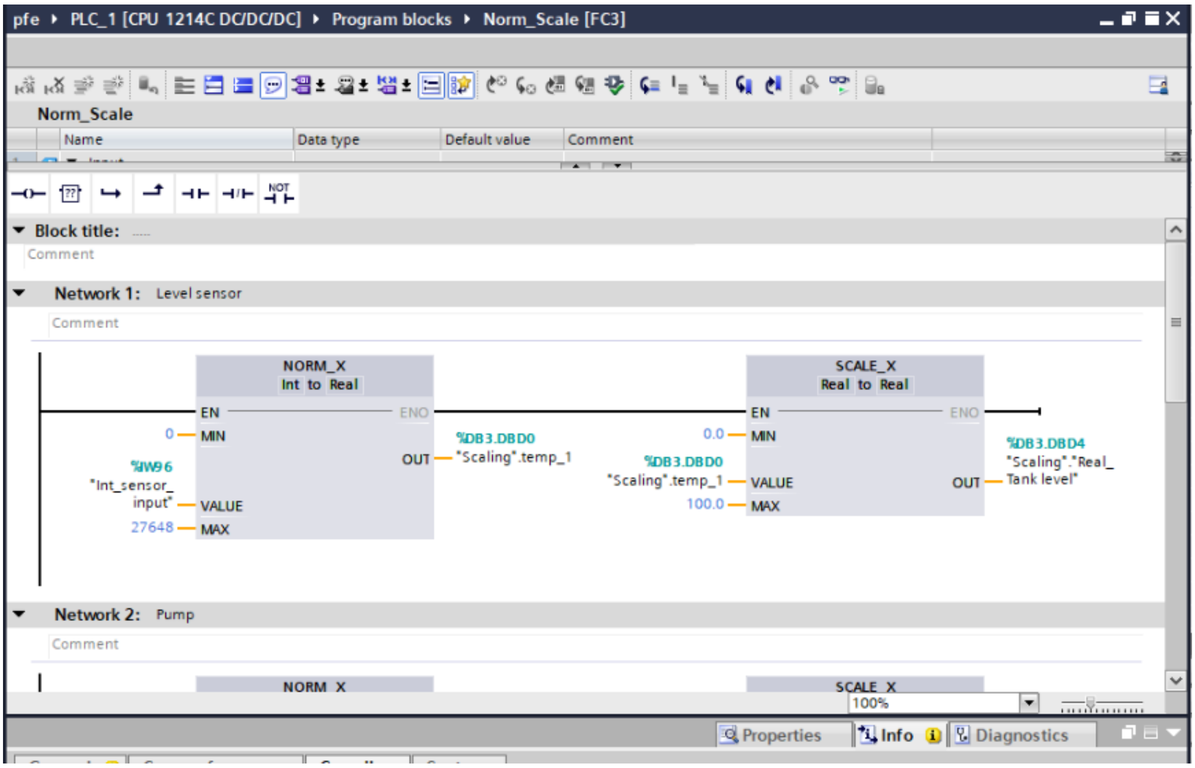
Table 5: Norm and scale description



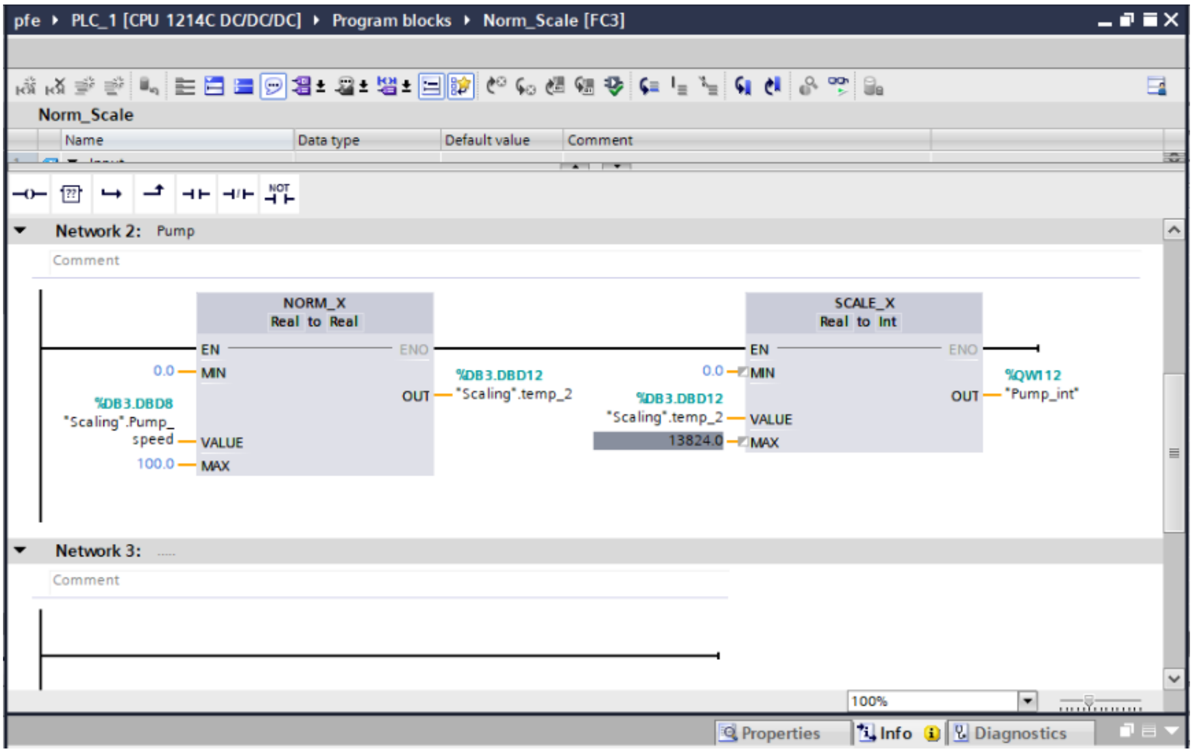
Parameter	Data type[1]	Description
MIN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Input minimum value for range
VALUE	SCALE_X: Real, LReal NORM_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Input value to scale or normalize
MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Input maximum value for range
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal NORM_X: Real, LReal	Scaled or normalized output value

Table 6: Norm and Scale parameter description and data type

Network 1: Analog processing of the level sensor signal

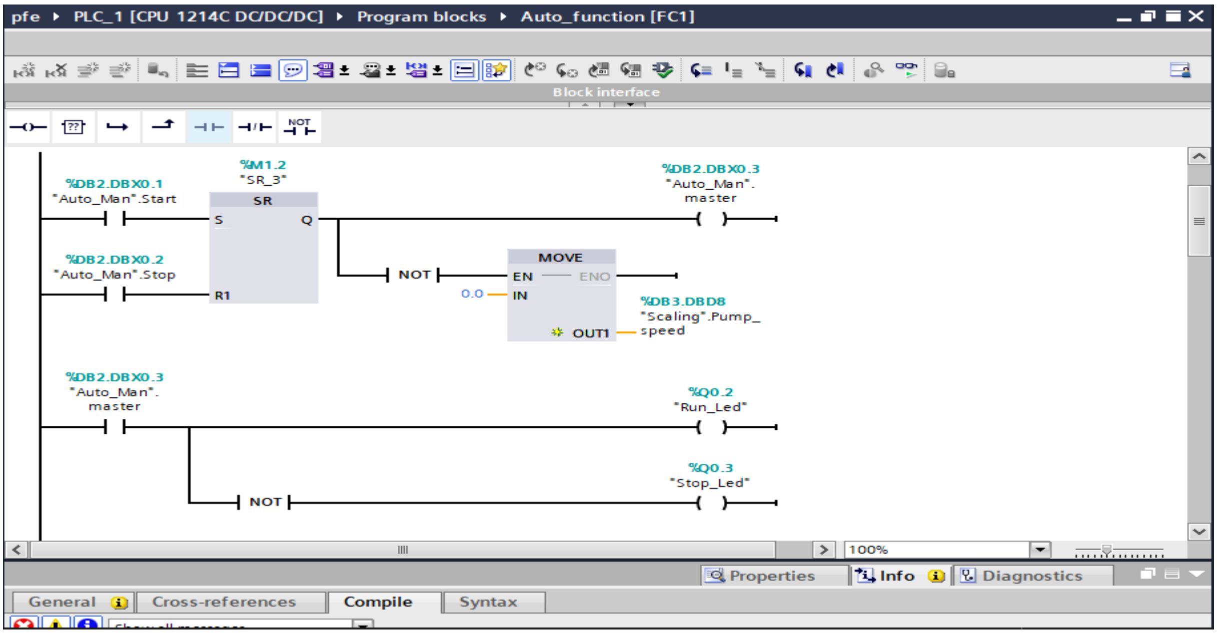


Network 2: sending a 0/5v signal to the pump



4.2.5.3 Automatic mode function

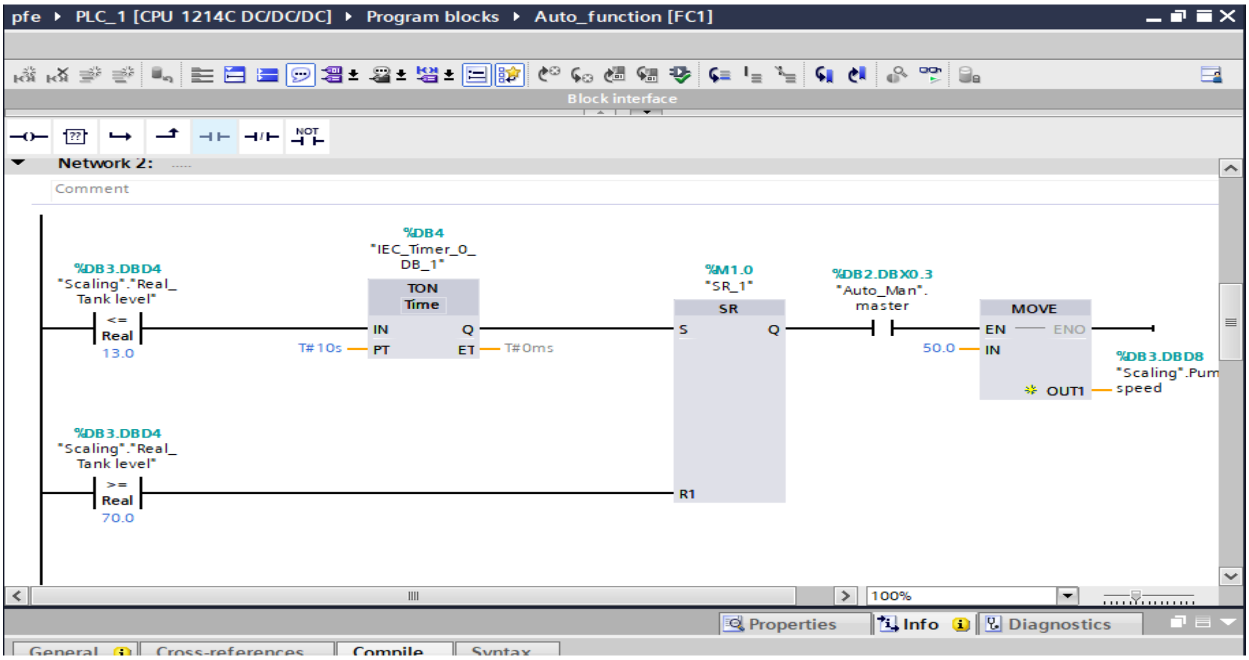
Network 1: the logic illustrates how the automatic mode function is controlled by two Boolean variables (start and stop)



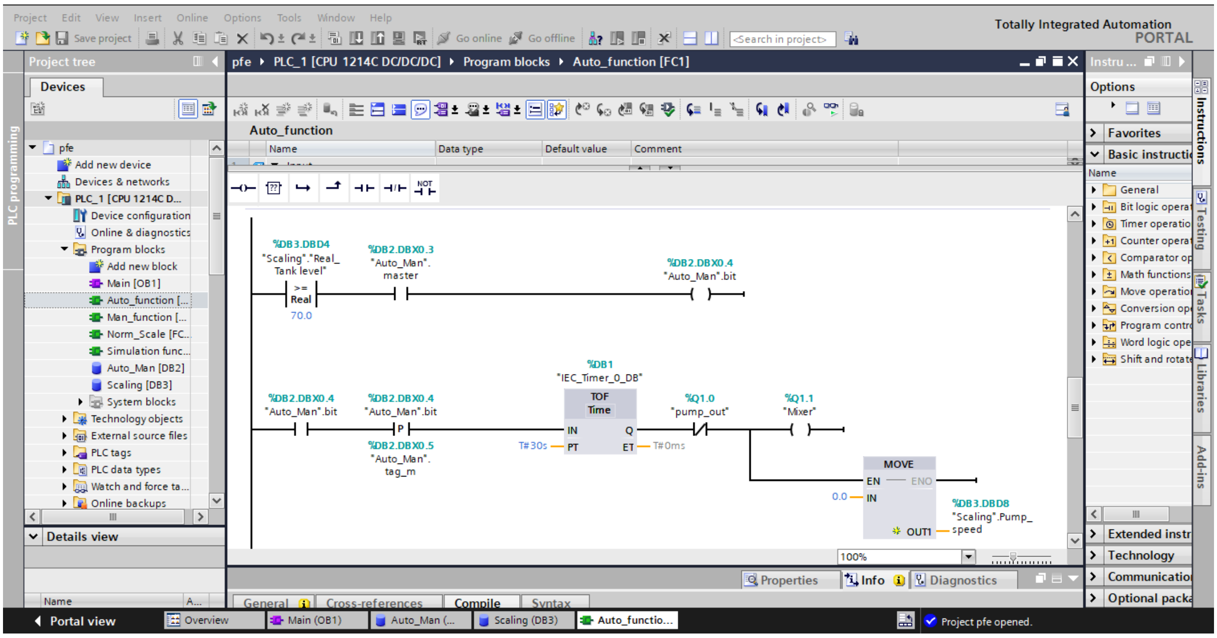
Network 2: this logic is illustrating how the pump is operating based on two conditions.

1-if the tank level is lower than 13% the pump will start operating after 10s with a speed of 50% of its capability

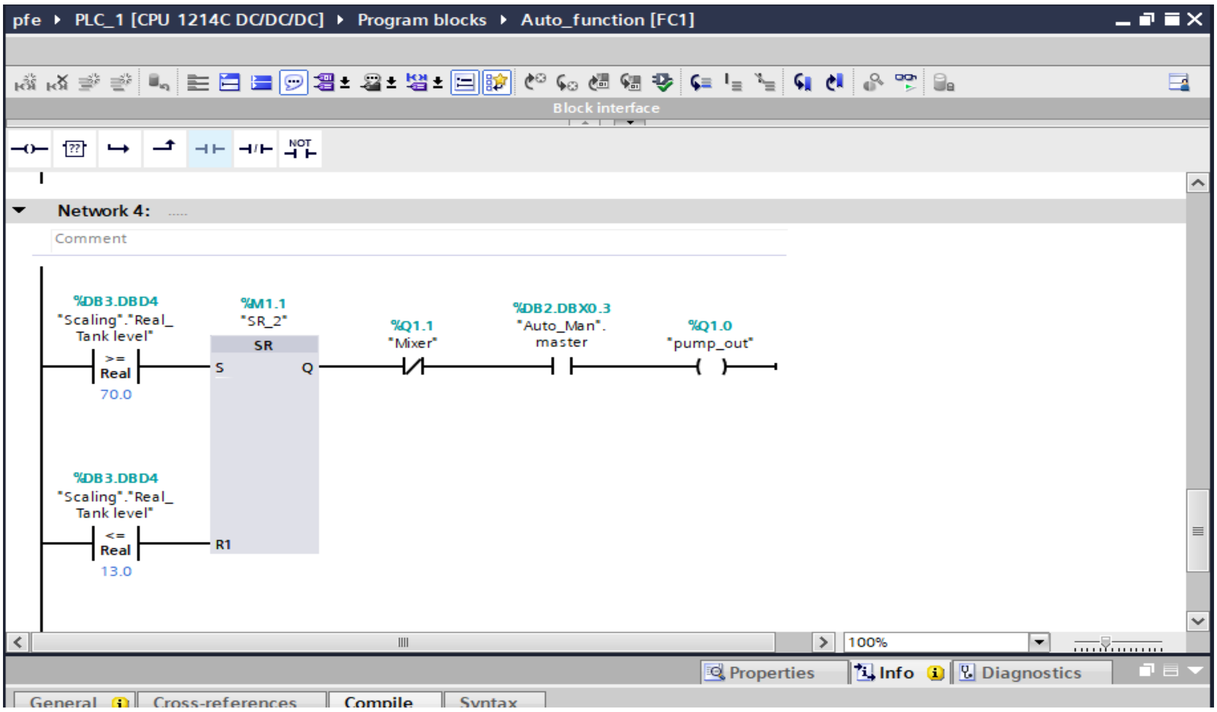
2-if the tank level is equal or higher than 70% the pump will stop working.



Network 3: when the pump is stopped the mixer (electric fan) will start turning for 30s and then stops

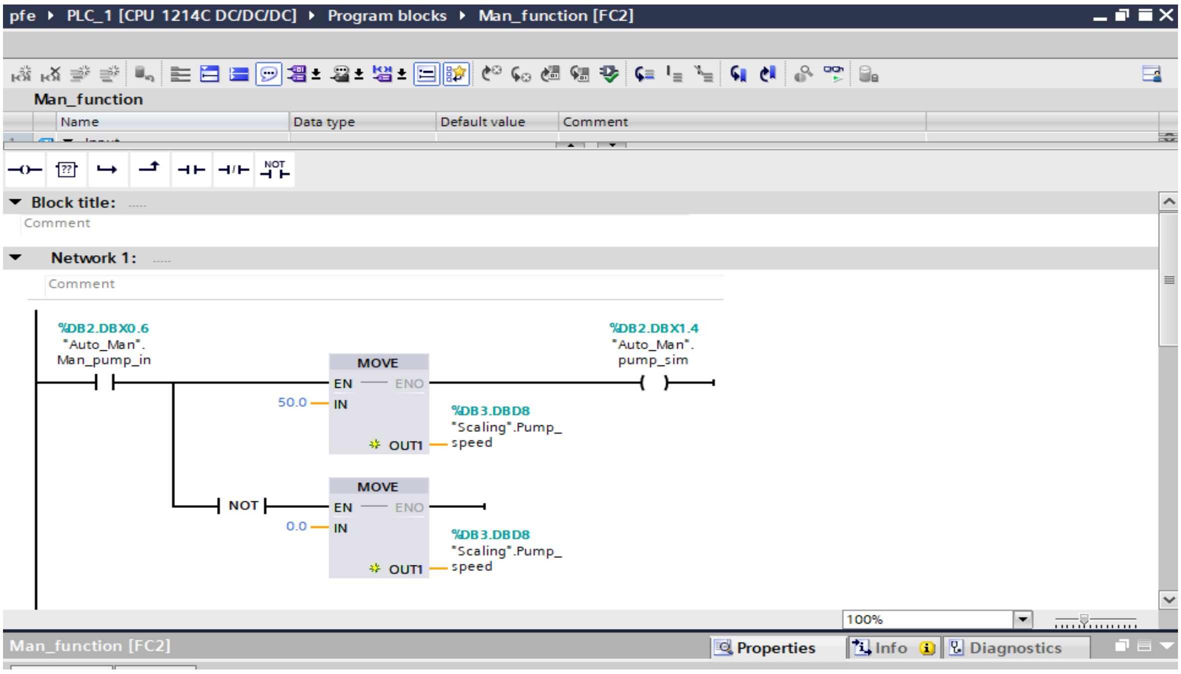


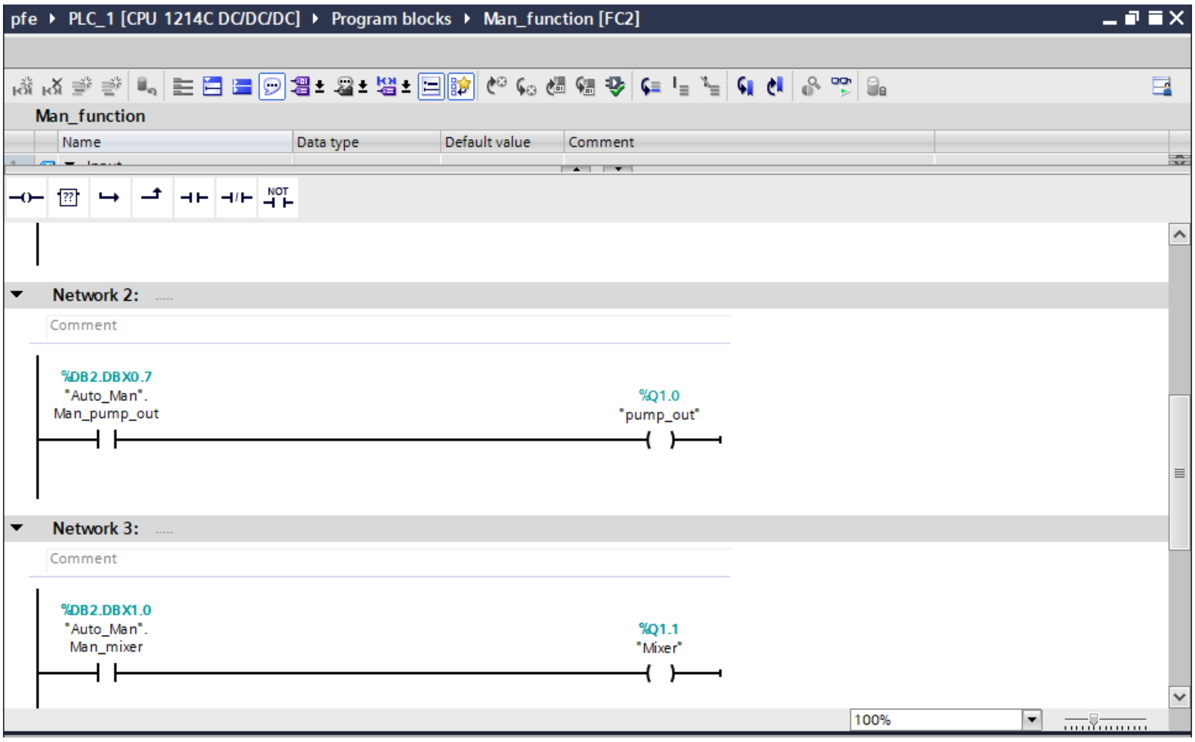
Network 4: when the mixer is stopped the pump will start draining water back to sump tank. It will stop working if the level is lower or equal to 13%



4.2.5.4 Manual mode Function

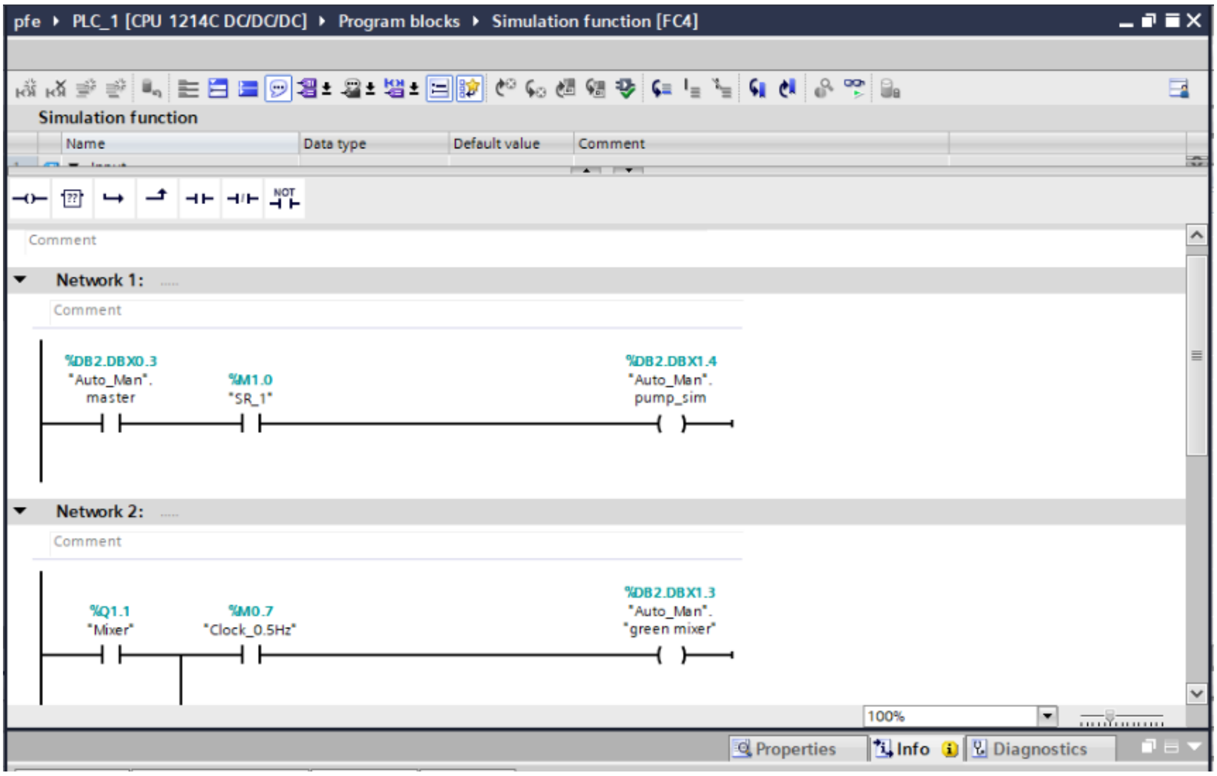
In this function all networks have the same working principle for example network 2 when the Boolean variable (man_pump_out) is set to 1 the pump will start draining water back to the sump tank and when it’s set to 0 the pump will stop

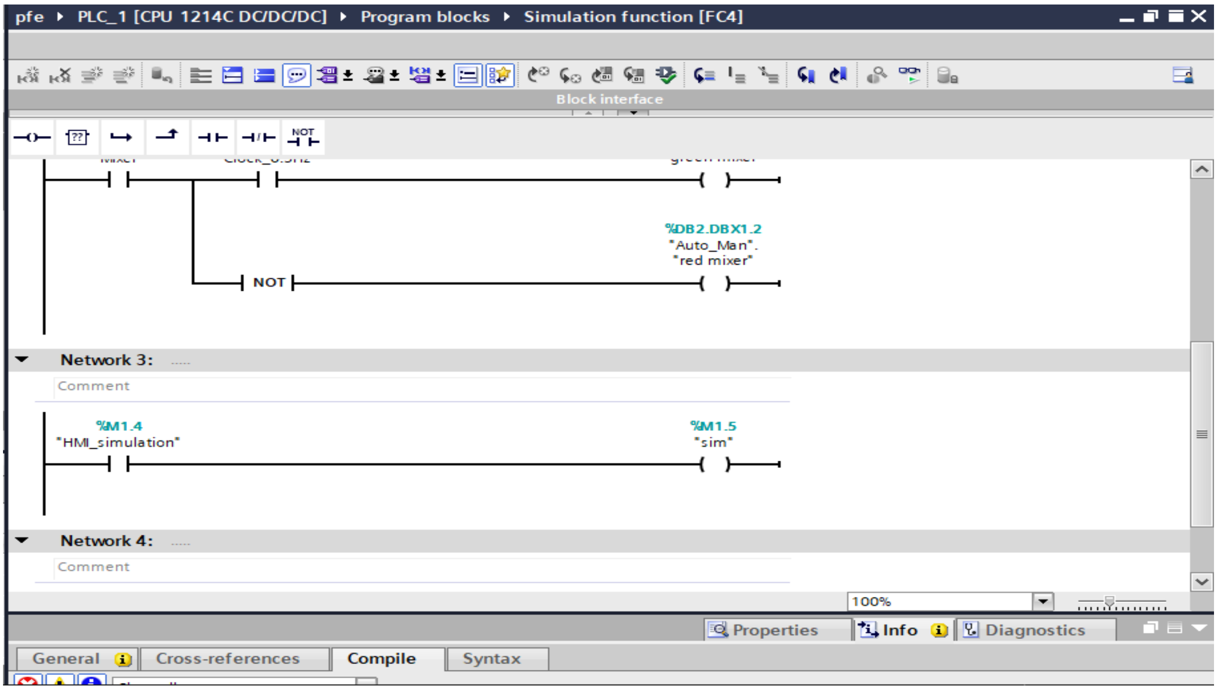




4.2.5.5 Simulation Function

In network 1 and 2 output variables (pump sim and green mixer, red mixer) are used for the styling of the pump and mixer in the scada control interface





Network 3: when HMI simulation is set to 0 the operator will not be able to control the system using the HMI because switches and buttons will not appear on the screen

4.2.6 Organization block

There are plenty of organization blocks and the main purpose of all of them is to execute user program in the given time intervals or based on different events. So, in short terms, OB's help you to organize your user program based on different criteria.

OB1: main organization block is being executed all the time and all over again. It can be interrupted by other organization blocks

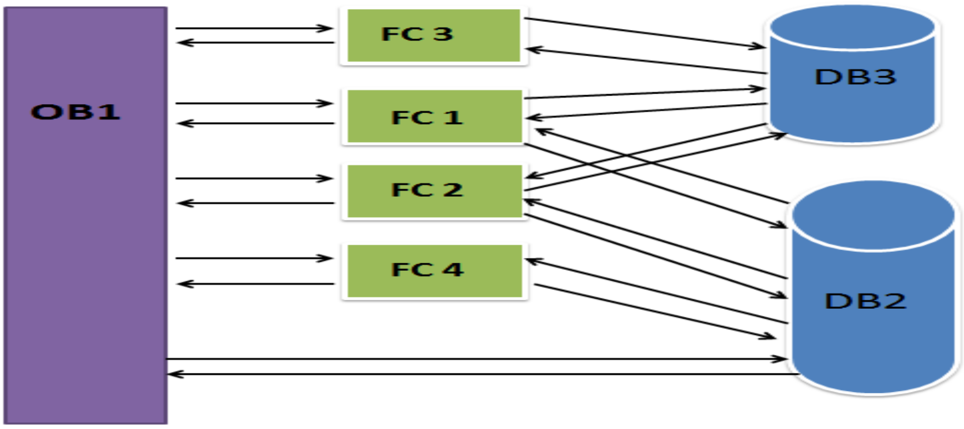
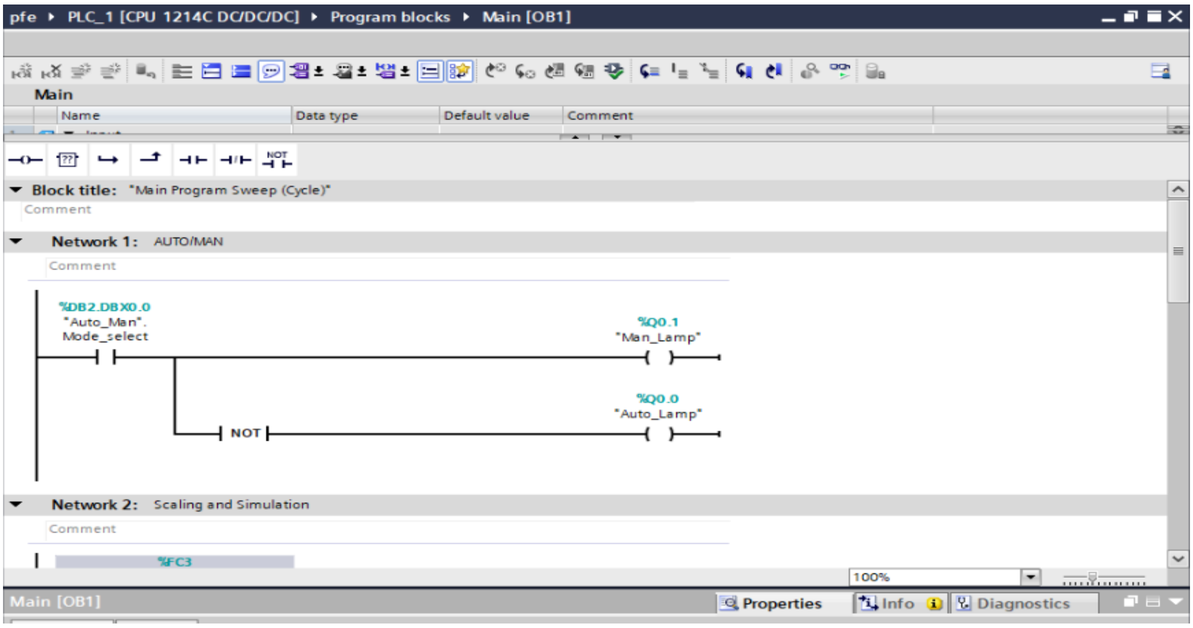


Figure 66: PLC Programming blocks structure

4.2.6.1 Calling functions in OB1

First network is for mode selection (Automatic – Manual)


pfe ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Main [OB1]

Main

Name	Data type	Default value	Comment

Block title: "Main Program Sweep (Cycle)"

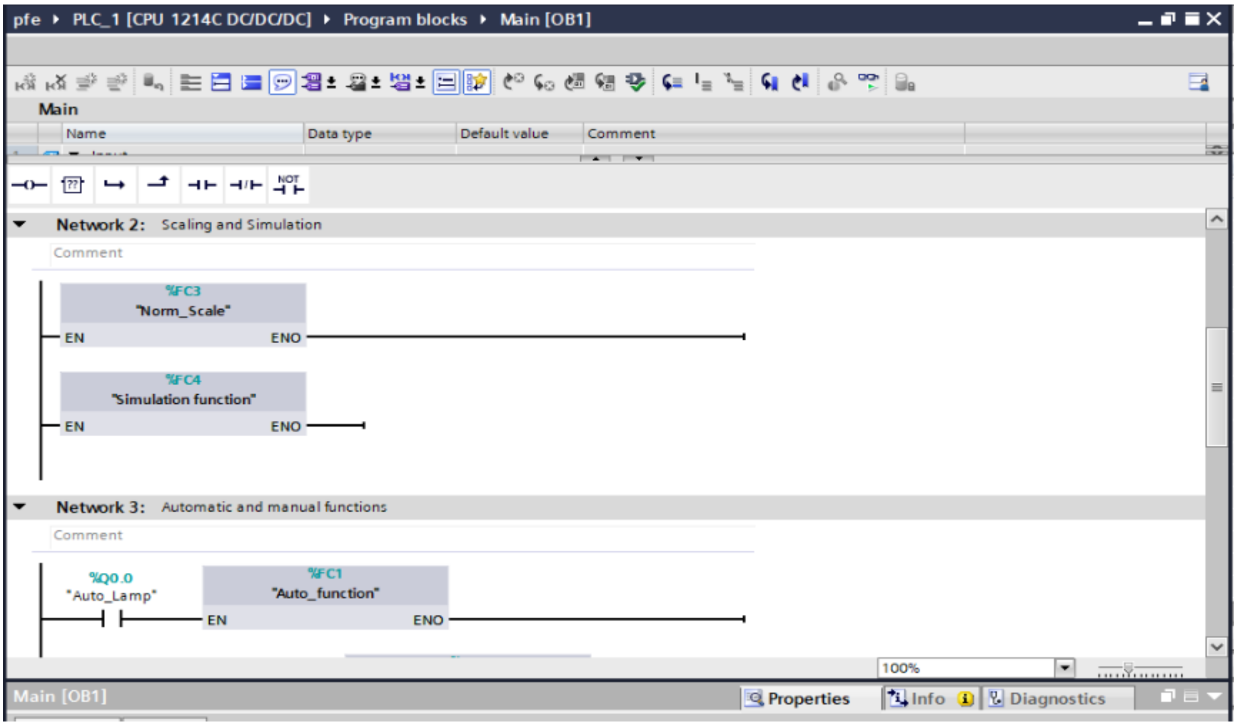
Comment

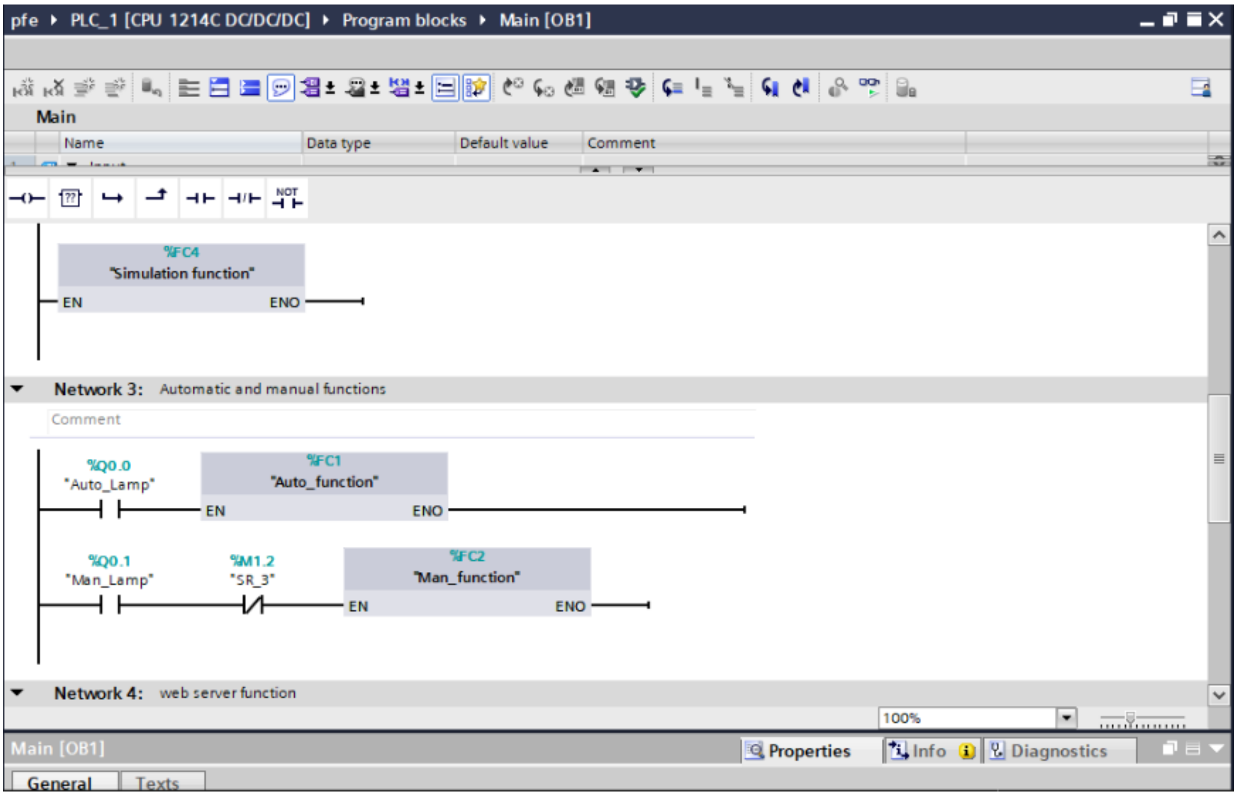
Network 1: AUTO/MAN

Comment

Network 2: Scaling and Simulation

Comment


pfe ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Main [OB1]
Main
Name
Data type
Default value
Comment
Network 2: Scaling and Simulation
Comment
%FC3
"Norm_Scale"
EN
ENO
%FC4
"Simulation function"
EN
ENO
Network 3: Automatic and manual functions
Comment
%Q0.0
"Auto_Lamp"
%FC1
"Auto_function"
EN
ENO
100%
Main [OB1]
Properties
Info
Diagnostics


pfe ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Main [OB1]
Main
Name
Data type
Default value
Comment
%FC4
"Simulation function"
EN
ENO
Network 3: Automatic and manual functions
Comment
%Q0.0
"Auto_Lamp"
%FC1
"Auto_function"
EN
ENO
%Q0.1
"Man_Lamp"
%M1.2
"SR_3"
%FC2
"Man_function"
EN
ENO
Network 4: web server function
100%
Main [OB1]
Properties
Info
Diagnostics
General
Texts

4.2.7 Control interface

4.2.7.1 Operator HMI

The HMI allows the operator to see what is going on in the plant in real time, including custom mimic displays, alarms, trends, etc

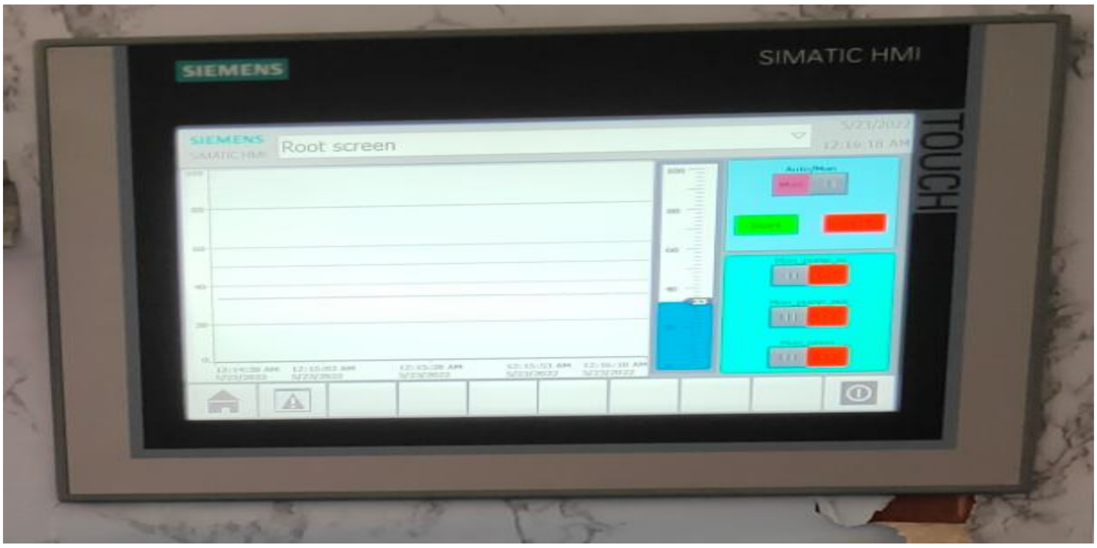
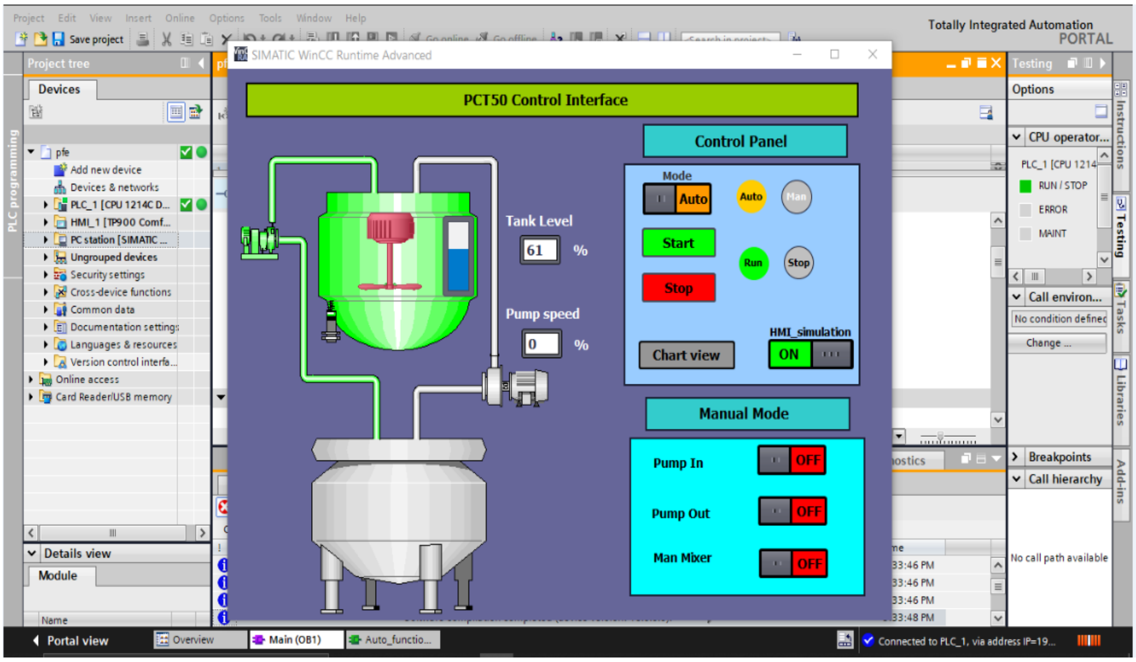


Figure 67: HMI interface

4.2.7.2 Master HMI

Without an HMI, operators would be unable to see or act upon the data gathered by PLCs and individual sensors. An HMI program allows operators to understand this data, as well as provide operators extensive reports related to network alarms.



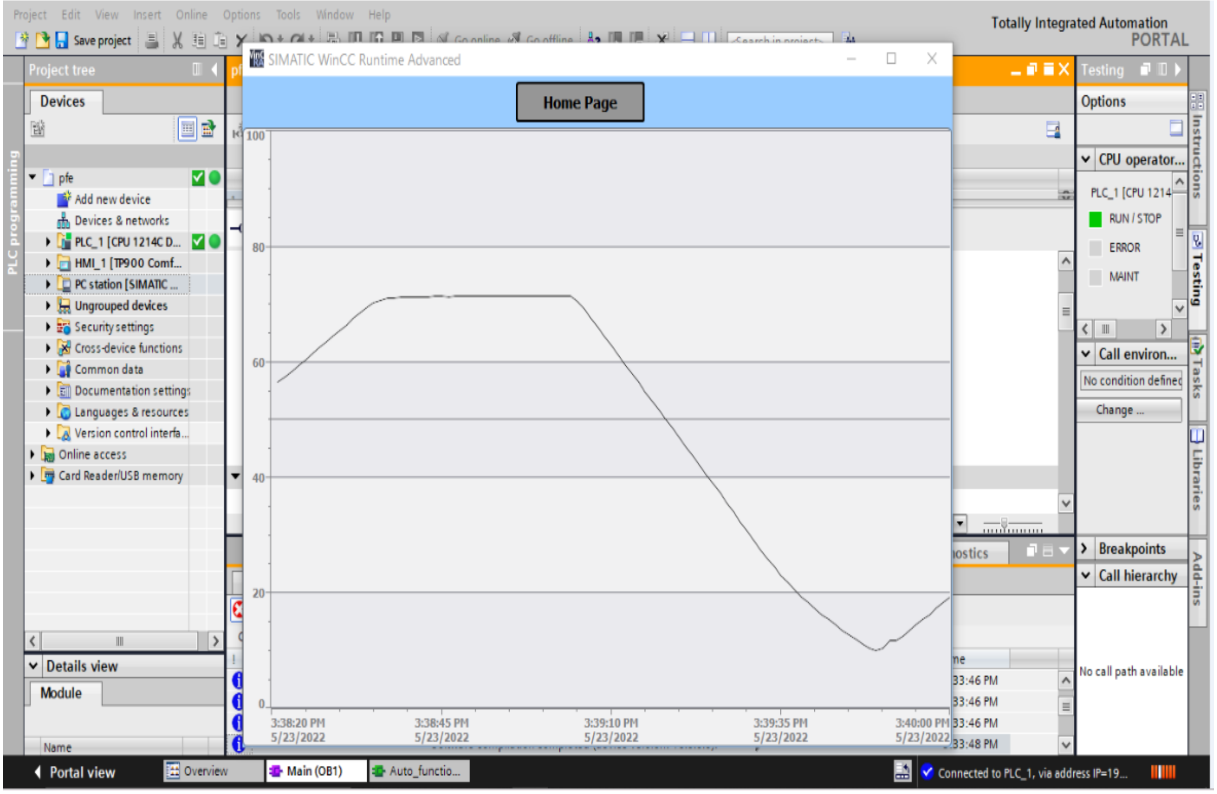


Figure 68: SCADA control interface

4.3 Web page Programming

4.3.1 HTML5 document structure

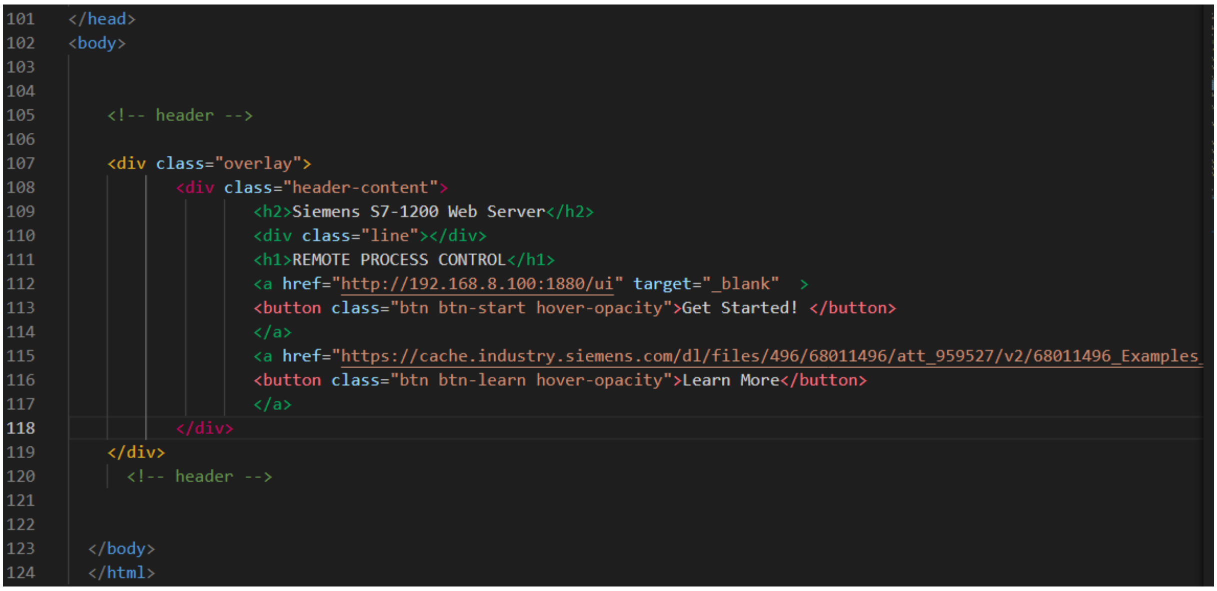
<!DOCTYPE> informs the browsers that it is actually an HTML 5 document. Although there are other types of DOC Types, this is the most commonly used declaration.

The DOCType Declaration is followed by **<html></html>** opening and closing tags. These tags contain everything inside the document, including the Head and Body

<head> </head> opening and closing tags, follow the opening Html tag. These tags contain information about the body, title of the page, definitions, labels, etc. You can only use certain markup elements in the HTML 5 head. Some of these elements include style, title, base, link, script, and meta. In HTML 5, these elements are collectively known as HTML Head Elements.

After the closing head tag is the **<body> </body>** opening and closing body tags. They contain all the content which appears on the browser, as well as the related HTML 5 codes.

4.3.2 HTML code



```
101  </head>
102  <body>
103
104
105      <!-- header -->
106
107      <div class="overlay">
108              <div class="header-content">
109                      <h2>Siemens S7-1200 Web Server</h2>
110                      <div class="line"></div>
111                      <h1>REMOTE PROCESS CONTROL</h1>
112                      <a href="http://192.168.8.100:1880/ui" target="_blank"  >
113                      <button class="btn btn-start hover-opacity">Get Started! </button>
114                      </a>
115                      <a href="https://cache.industry.siemens.com/dl/files/496/68011496/att_959527/v2/68011496_Examples_
116                      <button class="btn btn-learn hover-opacity">Learn More</button>
117                      </a>
118              </div>
119      </div>
120        <!-- header -->
121
122
123    </body>
124    </html>
```

The **HTML class** attribute specifies one or more **class** names for an element. **Classes** are used by CSS and JavaScript to select and access specific elements

The <div> tag defines a division or a section in an HTML document. The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript. The <div> tag is easily styled by using the class or id attribute. Any sort of content can be put inside the <div> tag

The **HTML** <**h2**> tag defines the second level heading in the **HTML** document. This tag is also commonly referred to as the <**h2**> **element**.

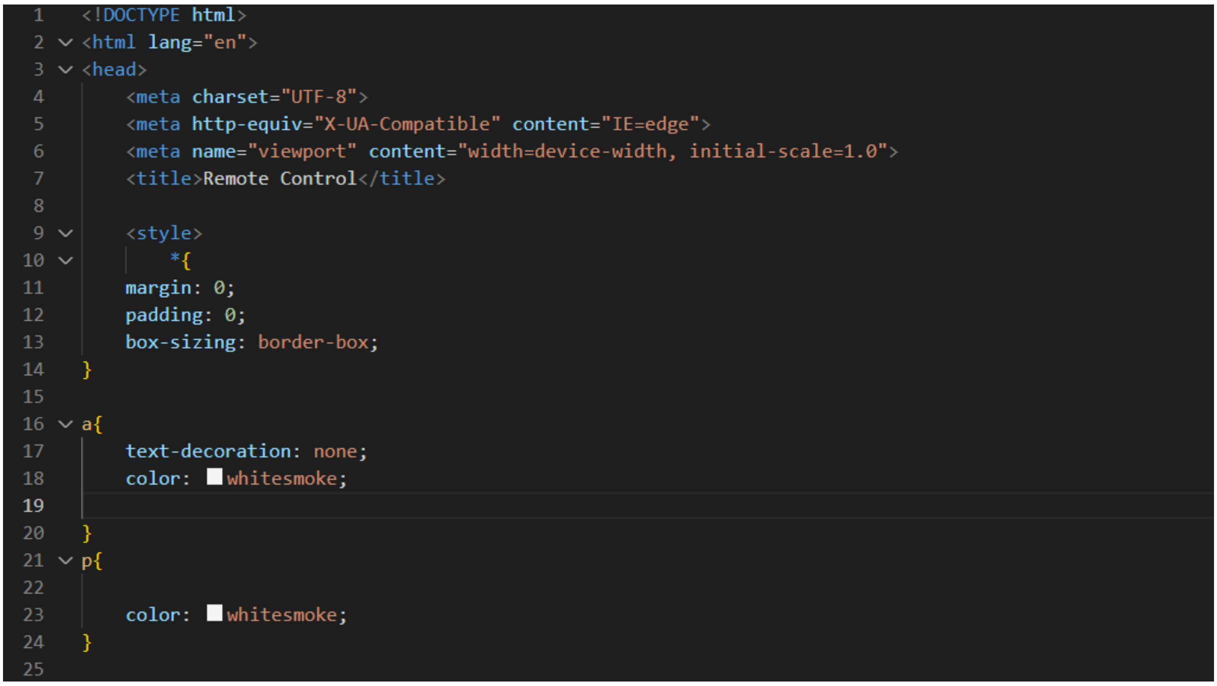
The HTML <**h1**> tag defines the highest level or most important heading in the HTML document. This tag is also commonly referred to as the **<h1> element.**

The <a> HTML element (or anchor element), with its **href** attribute, **creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address**. Content within each <a> should indicate the link's destination.

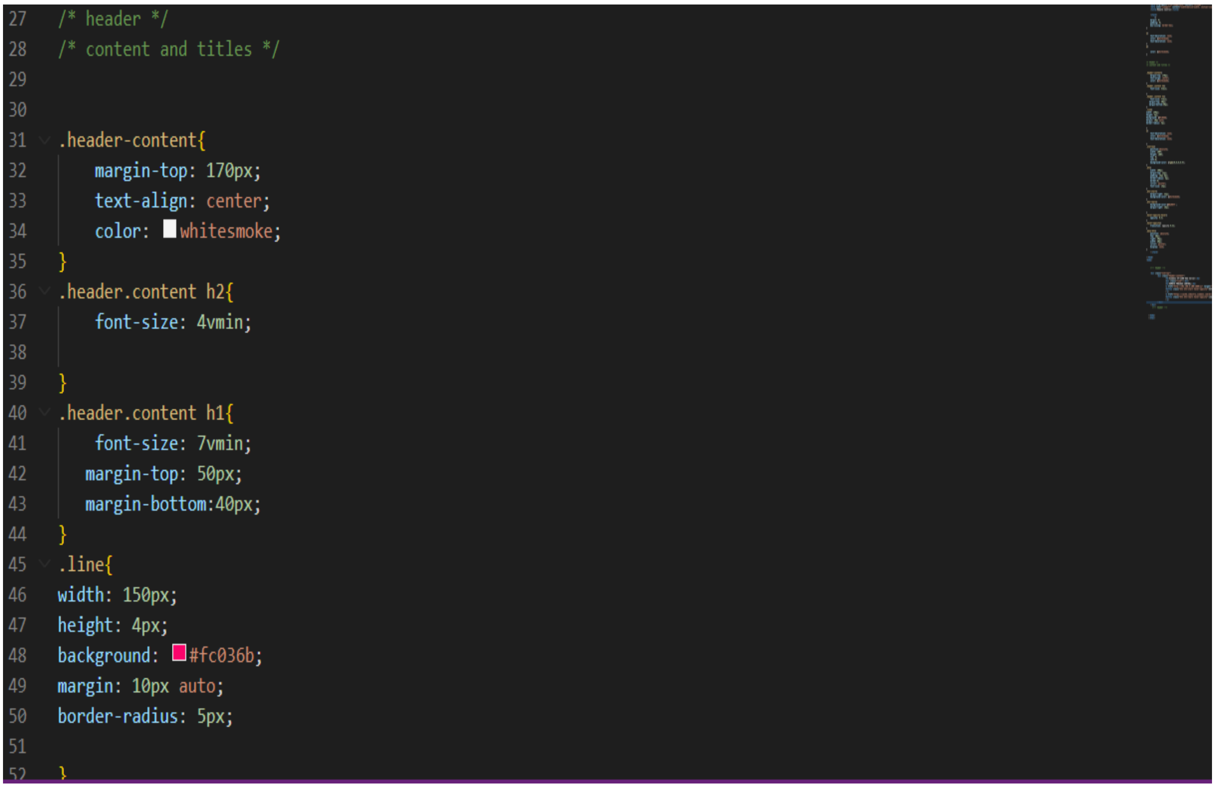
The <**button**> tag defines a clickable **button**. Inside a <**button**> **element** you can put text and tags.

Line 121: the href instruction is used to specify the URL of the page the link goes to and in my case it is the node red dashboard link which is my PC LAN IP address followed by :1880/ui to directly access node-red dashboard.

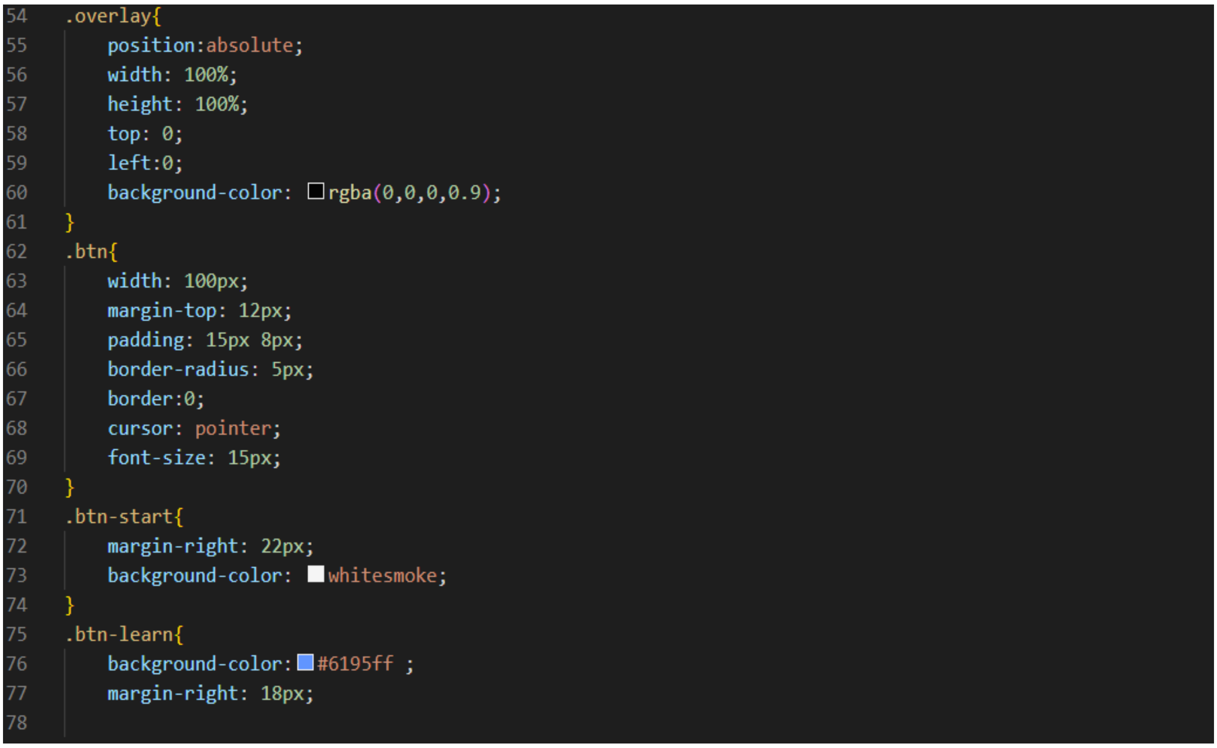
4.3.3 CSS code



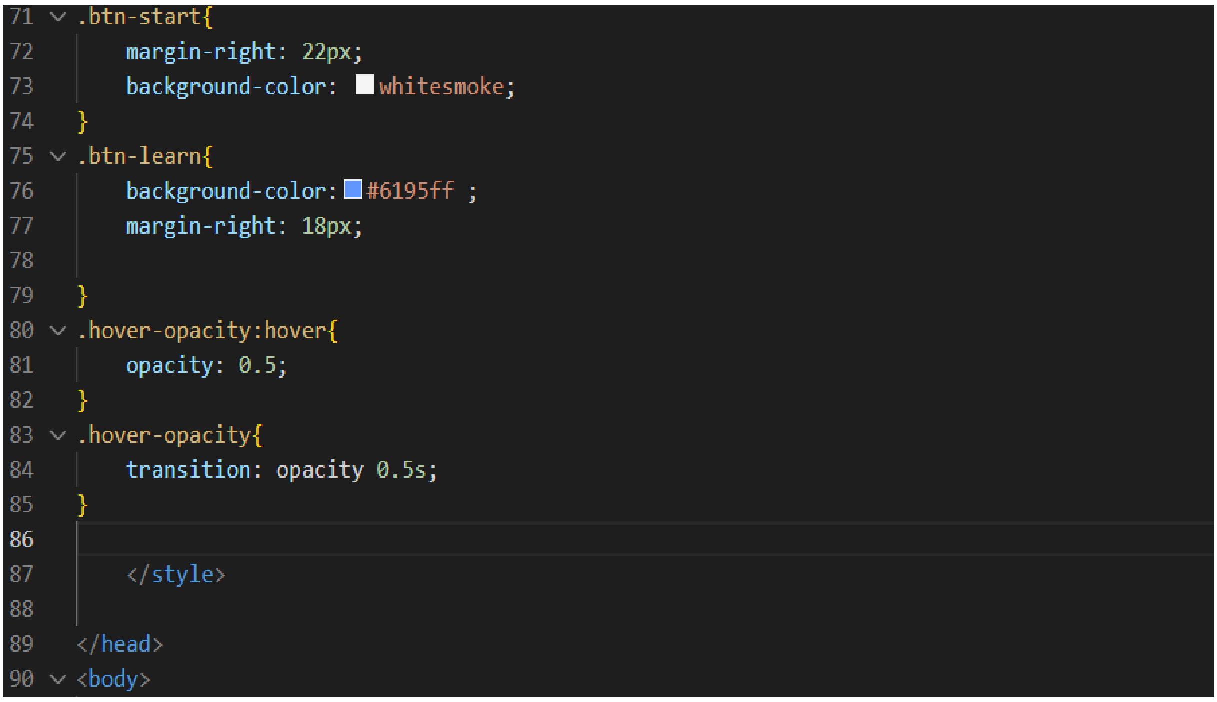
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Remote Control</title>

    <style>
        *{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

a{
    text-decoration: none;
    color: whitesmoke;

}
p{

    color: whitesmoke;
}

```



```
/* header */
/* content and titles */


.header-content{
    margin-top: 170px;
    text-align: center;
    color: whitesmoke;
}
.header.content h2{
    font-size: 4vmin;

}
.header.content h1{
    font-size: 7vmin;
   margin-top: 50px;
   margin-bottom:40px;
}
.line{
width: 150px;
height: 4px;
background: #fc036b;
margin: 10px auto;
border-radius: 5px;

}
```



```css
.overlay{
    position:absolute;
    width: 100%;
    height: 100%;
    top: 0;
    left:0;
    background-color: □rgba(0,0,0,0.9);
}
.btn{
    width: 100px;
    margin-top: 12px;
    padding: 15px 8px;
    border-radius: 5px;
    border:0;
    cursor: pointer;
    font-size: 15px;
}
.btn-start{
    margin-right: 22px;
    background-color: ⬜whitesmoke;
}
.btn-learn{
    background-color:🟦#6195ff ;
    margin-right: 18px;

```



```css
.btn-start{
    margin-right: 22px;
    background-color: ⬜whitesmoke;
}
.btn-learn{
    background-color:🟦#6195ff ;
    margin-right: 18px;

}
.hover-opacity:hover{
    opacity: 0.5;
}
.hover-opacity{
    transition: opacity 0.5s;
}

    </style>

</head>
<body>
```

Margins are used to create space around elements, outside of any defined borders.

The CSS **padding** properties are used to generate space around an element's content, inside of any defined borders.

The **box-sizing** property allows us to include the padding and border in an element's total width and height.

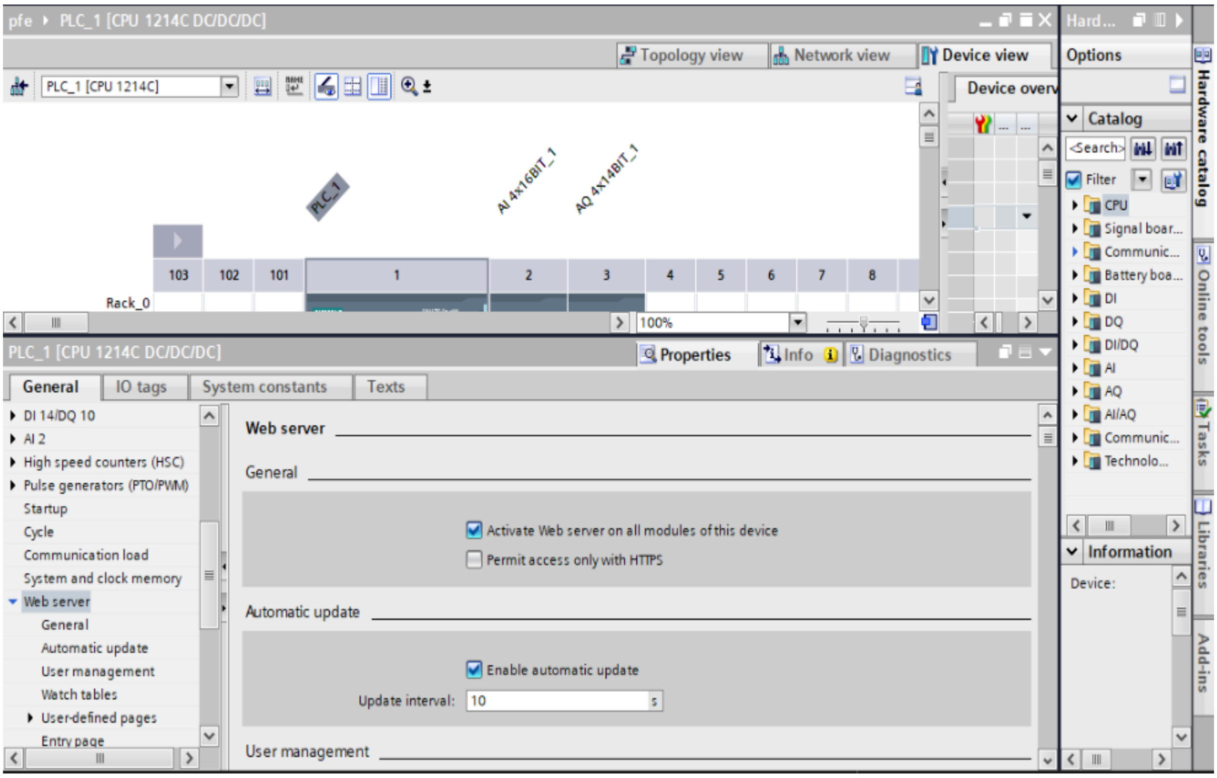
If you set **box-sizing: border-box** on an element, padding and border are included in the width and height

The **text-decoration** property is mostly used to remove underlines from links.

An element with **position: absolute** is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed). However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

The **:hover** CSS pseudo-class matches when the user interacts with an element with a pointing device, but does not necessarily activate it. It is generally triggered when the user hovers over an element with the cursor (mouse pointer).

4.3.4 SIMATIC S7-1200 web-server configuration



Web server

General

☑ Activate Web server on all modules of this device

□ Permit access only with HTTPS

Automatic update

☑ Enable automatic update

Update interval: 10 s

User management

Figure 69: Activate web server

4.3.5 Generating data blocks DB333 and DB334

Web control DB and fragment DBs

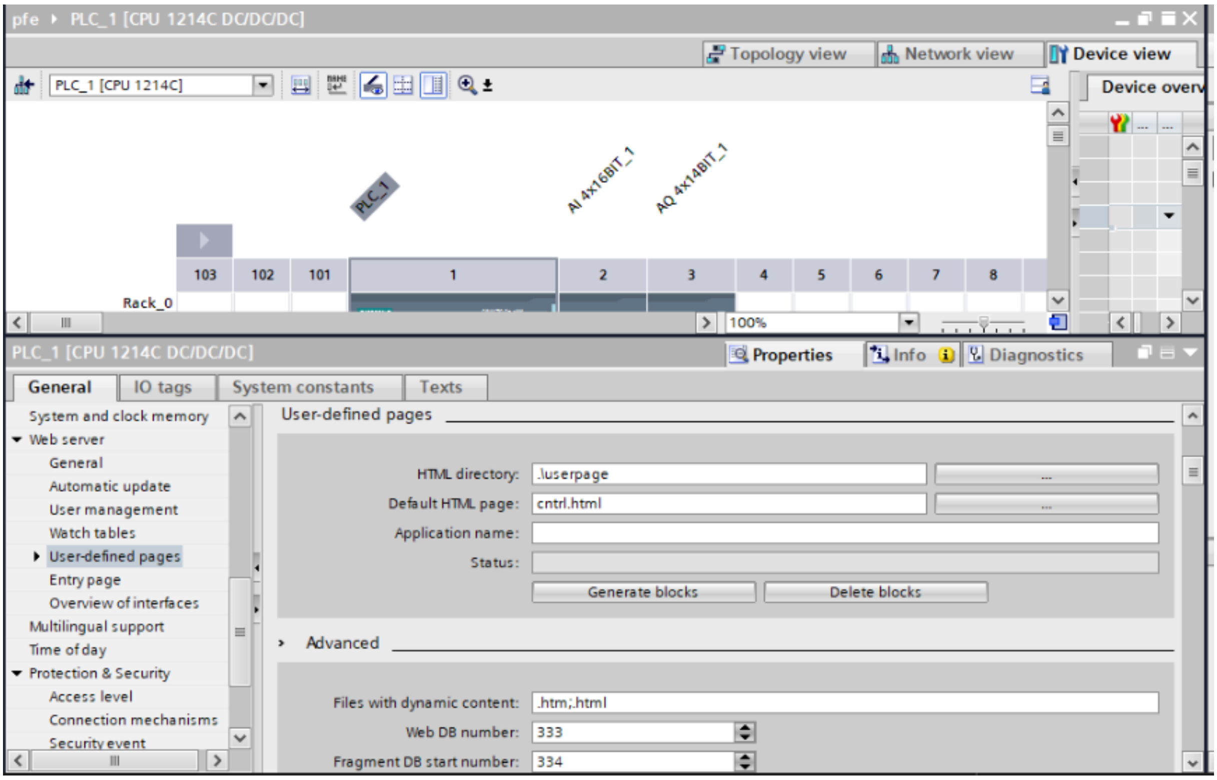
The basis of the web page designed is an HTML file (or several connected HTML files with images):

To enable the CPU to interpret the HTML file, it is stored in data blocks together with other required files:

The Web Control DB (default: DB333) contains the following:

- Status and control variables of the web page
- Communication status (e.g. whether a request from the web browser to the web server is pending)
- Error information

Additionally to the Web Control DB there are also fragment DBs starting by default with DB334. These DBs contain the coded web pages and media data (e.g. images).



HTML directory: .\userpage

Default HTML page: cntrl.html

Application name:

Status:

Files with dynamic content: .htm;.html

Web DB number: 333

Fragment DB start number: 334

Figure 70: Generating Data Blocks

4.3.6 Calling WWW function in OB1

The instruction WWW initializes the web server of the CPU or synchronizes user-defined web pages in short “user pages” with the user program in the CPU.

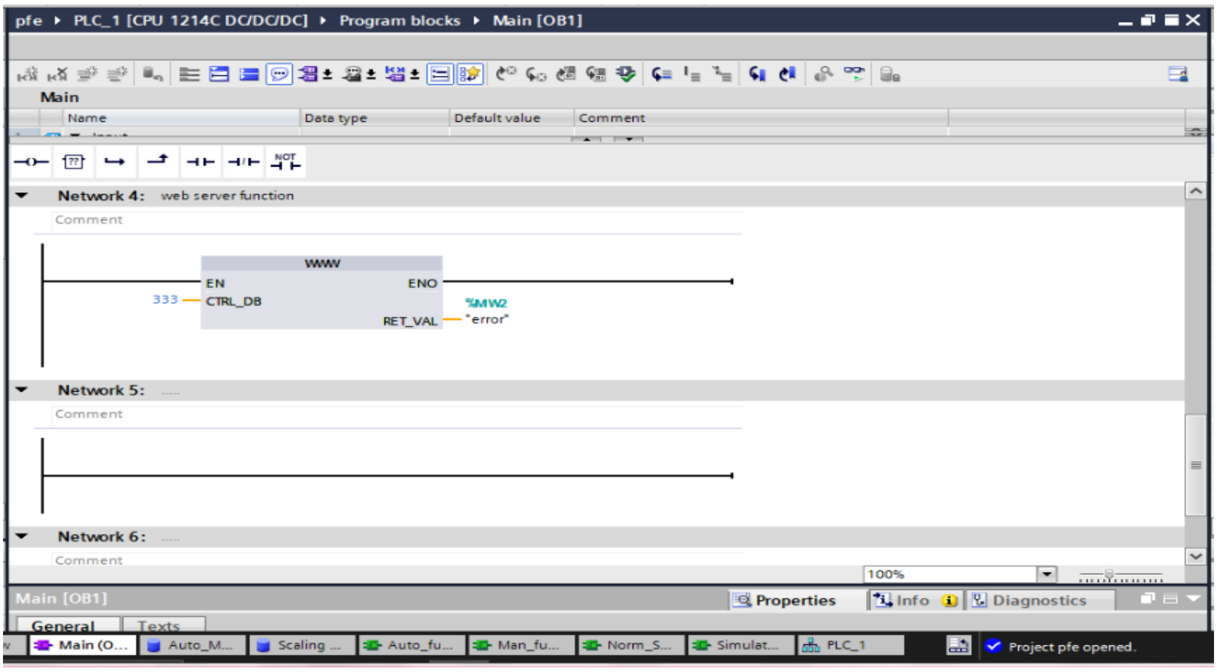


Figure 71: WWW function

4.4 Node Red programming

4.4.1 Node –red –contrib –S7

It is a node to interact with Siemens S7 PLCs.

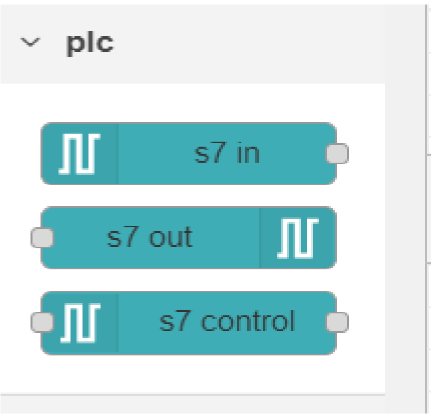
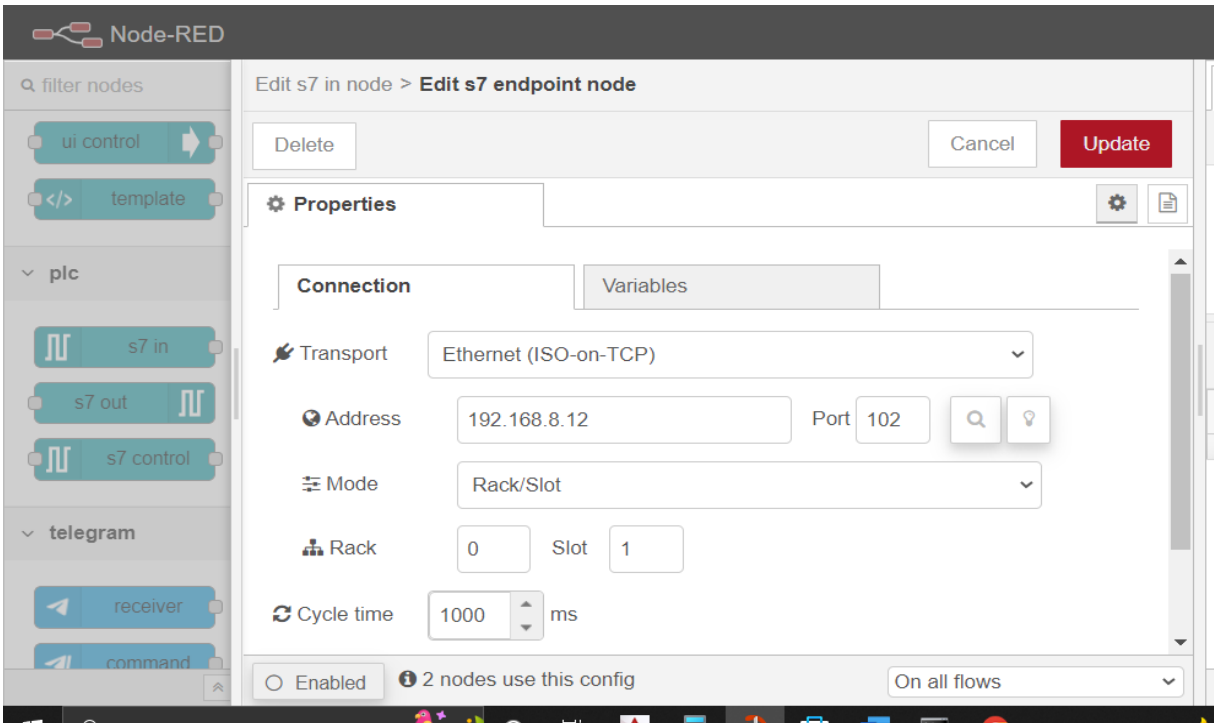


Figure 72: S7 contrib nodes

4.4.1.1 S7 nodes configuration



Edit s7 in node > **Edit s7 endpoint node**

Transport: Ethernet (ISO-on-TCP)

Address: 192.168.8.12 Port: 102

Mode: Rack/Slot

Rack: 0 Slot: 1

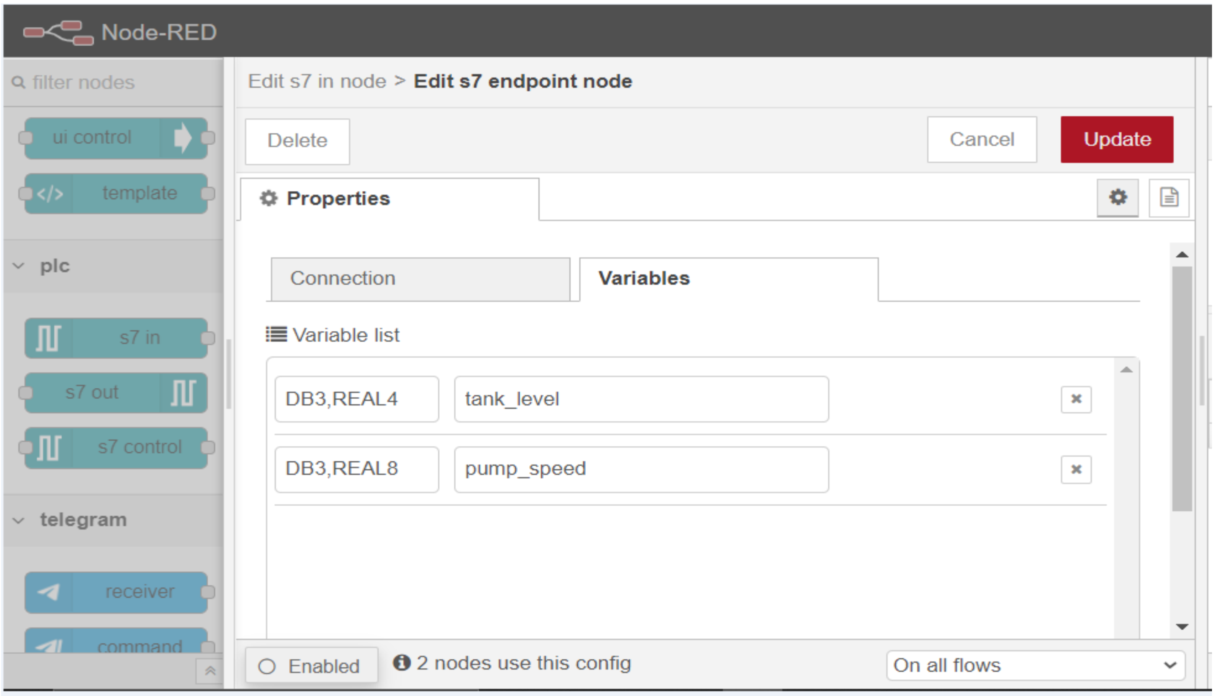
Cycle time: 1000 ms

2 nodes use this config

On all flows

Figure 73: S7 contrib node configuration

4.1.1.2 Variables configuration



Edit s7 in node > **Edit s7 endpoint node**

Variable list

DB3,REAL4 tank_level

DB3,REAL8 pump_speed

2 nodes use this config

On all flows

Figure 74: S7 contrib node variables

4.4.2 Node –red-dashboard

It is a set of nodes in Node-RED to quickly create a live data dashboard.

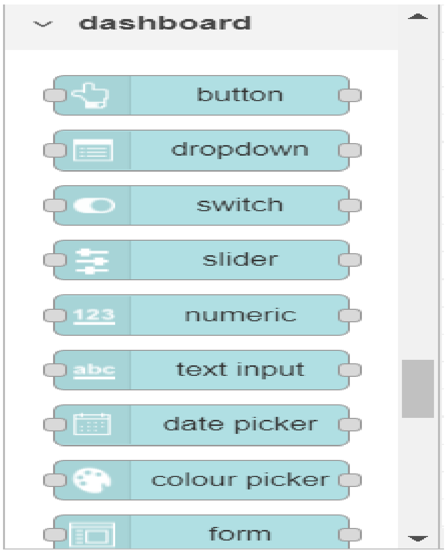


Figure 75: dashboard nodes

4.4.3 The Function node allows JavaScript code to be run against the messages that are passed through it.

4.4.4 The switch node allows your flow to execute one of several possible paths depending on the rules you have specified.

4.4.5 The Change node can be used to modify a message's properties and set context properties without having to resort to a Function node.

4.4.6 The Inject node allows you to inject messages into a flow

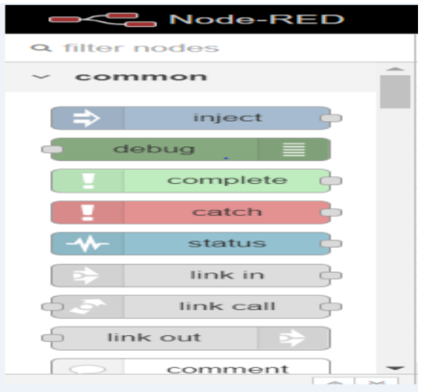


Figure 76: common nodes

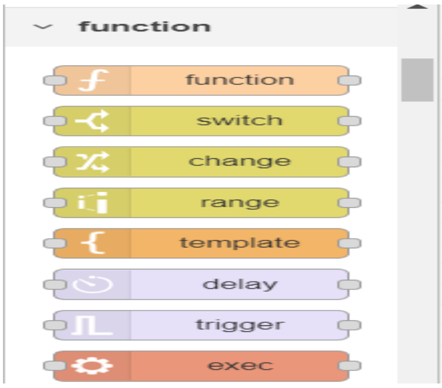


Figure 77: function nodes

4.4.7 Ubidots-nodered

It is used to interact with the Ubidots cloud service. It publishes and subscribes to one or multiple variables. It also provides the ability to subscribe to up to 10 Ubidots custom topics.

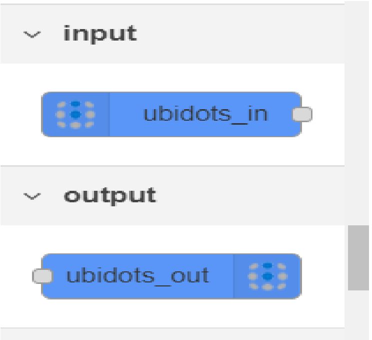
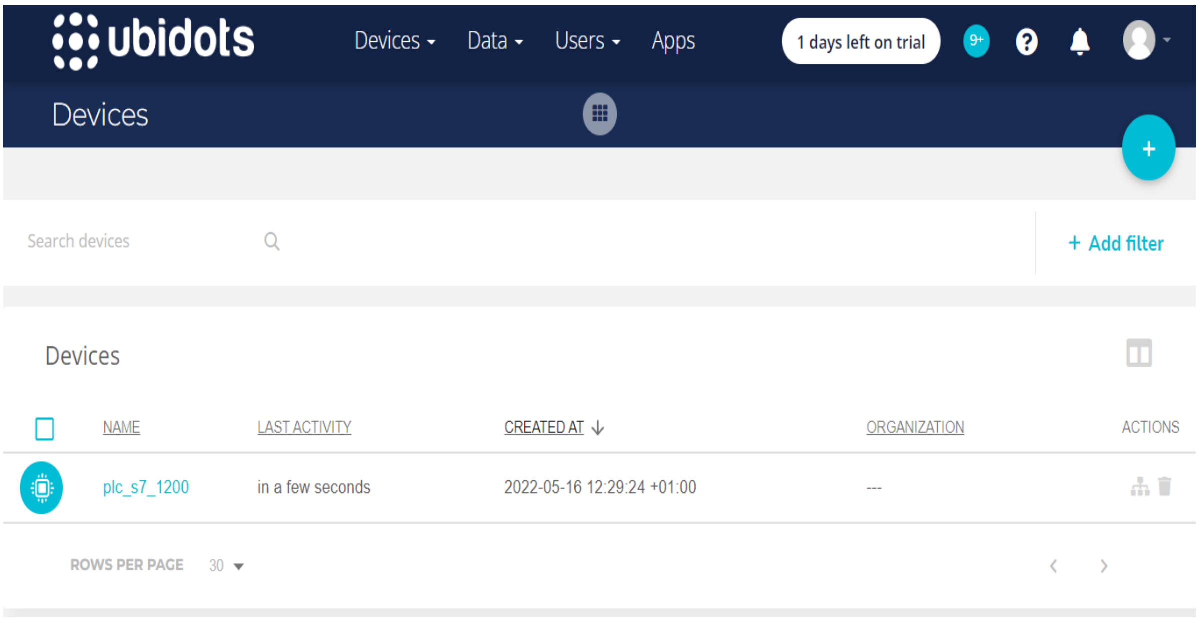


Figure 78: Ubidots nodes

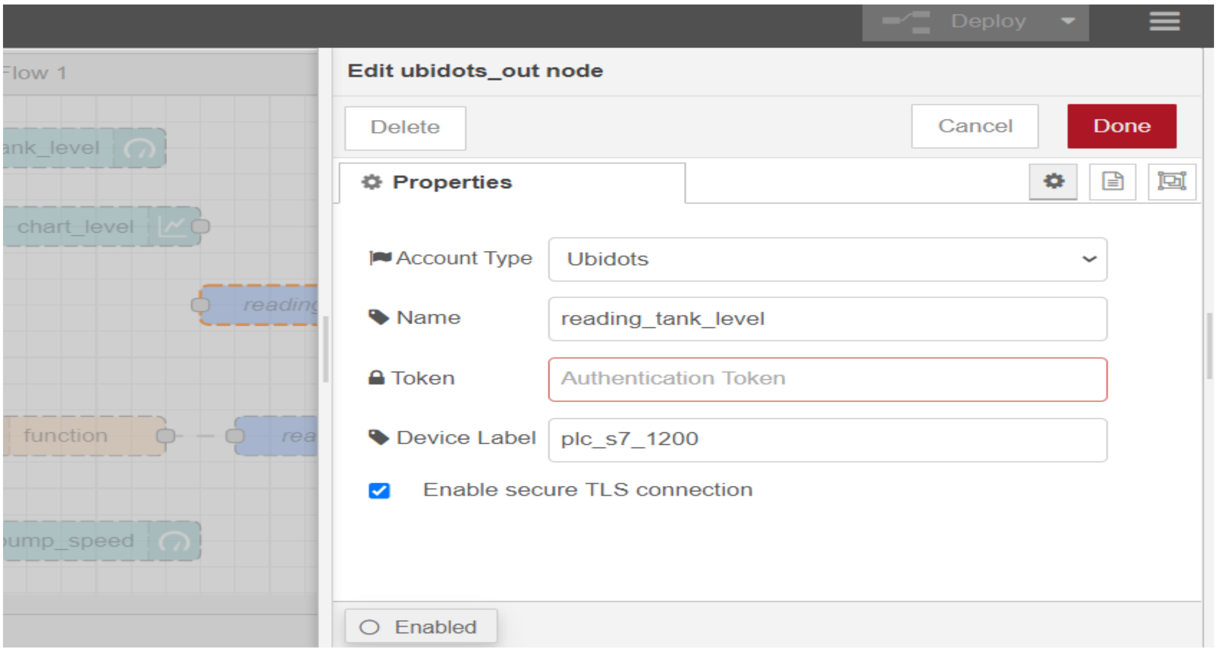
4.4.7.1 Creating a new device in Ubidots IoT platform



NAME	LAST ACTIVITY	CREATED AT	ORGANIZATION	ACTIONS
plc_s7_1200	in a few seconds	2022-05-16 12:29:24 +01:00	---	

Figure 79: creating a new device in Ubidots platform

4.4.7.2 Ubidots out node configuration



Account Type: Ubidots

Name: reading_tank_level

Token: Authentication Token

Device Label: plc_s7_1200

☑ Enable secure TLS connection

Figure 80: ubidots out node properties

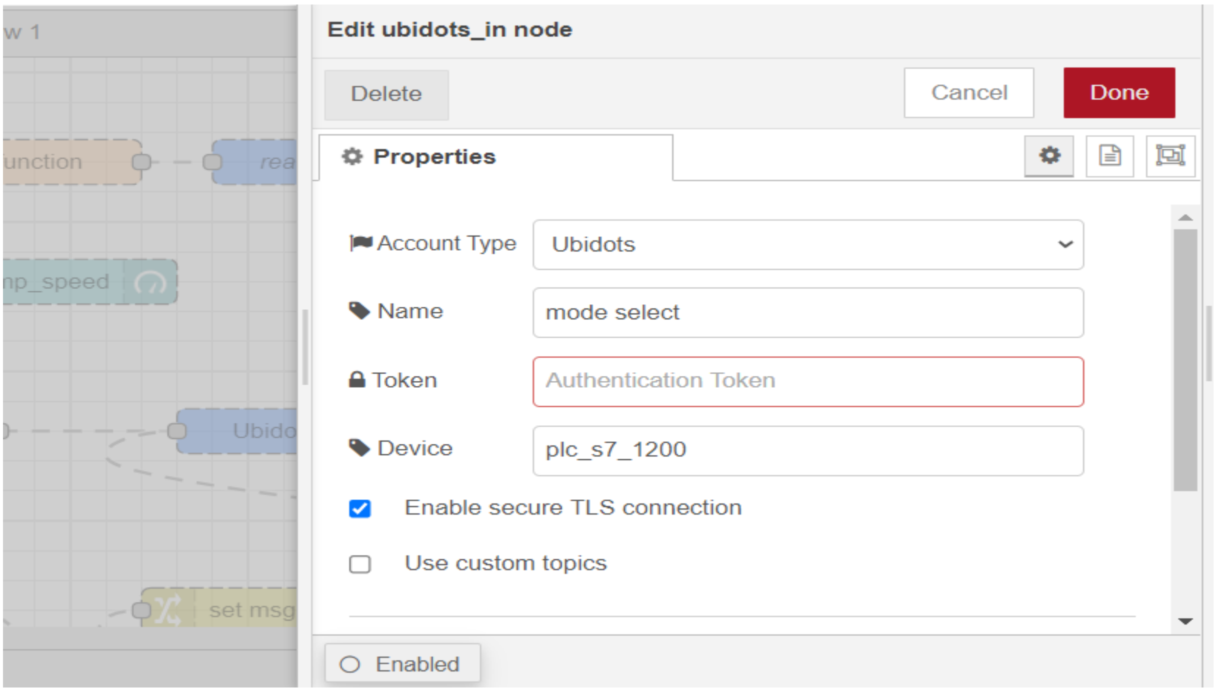
Account type: Defaults to “Ubidots”, which is valid for both Industrial and STEM account type. If you have a STEM account, choose “Ubidots for Education”

Name: Label of node in Node-Red workspace. If empty, defaults to: "Ubidots out". This library uses an MQTT implementation hence, it's necessary to name this MQTT client. We recommend you to choose a non-easy-to-copy name, to prevent cases where the name crashes between different clients in the broker.

Token: The Token necessary to authenticate the connection to your account. To obtain your token, login on ubidots.com, under "My Profile" click on "API Credentials".

TLS: By default all data is sent encrypted via TLS. Uncheck if data should be sent unencrypted

4.4.7.3 Ubidots in node configuration



Account Type: Ubidots
Name: mode select
Token: Authentication Token
Device: plc_s7_1200
☑ Enable secure TLS connection
□ Use custom topics

Figure 81: ubidots in node properties

Account type: Defaults to “Ubidots”, which is valid for both Industrial and STEM account type. If you have a STEM account, choose “Ubidots for Education”

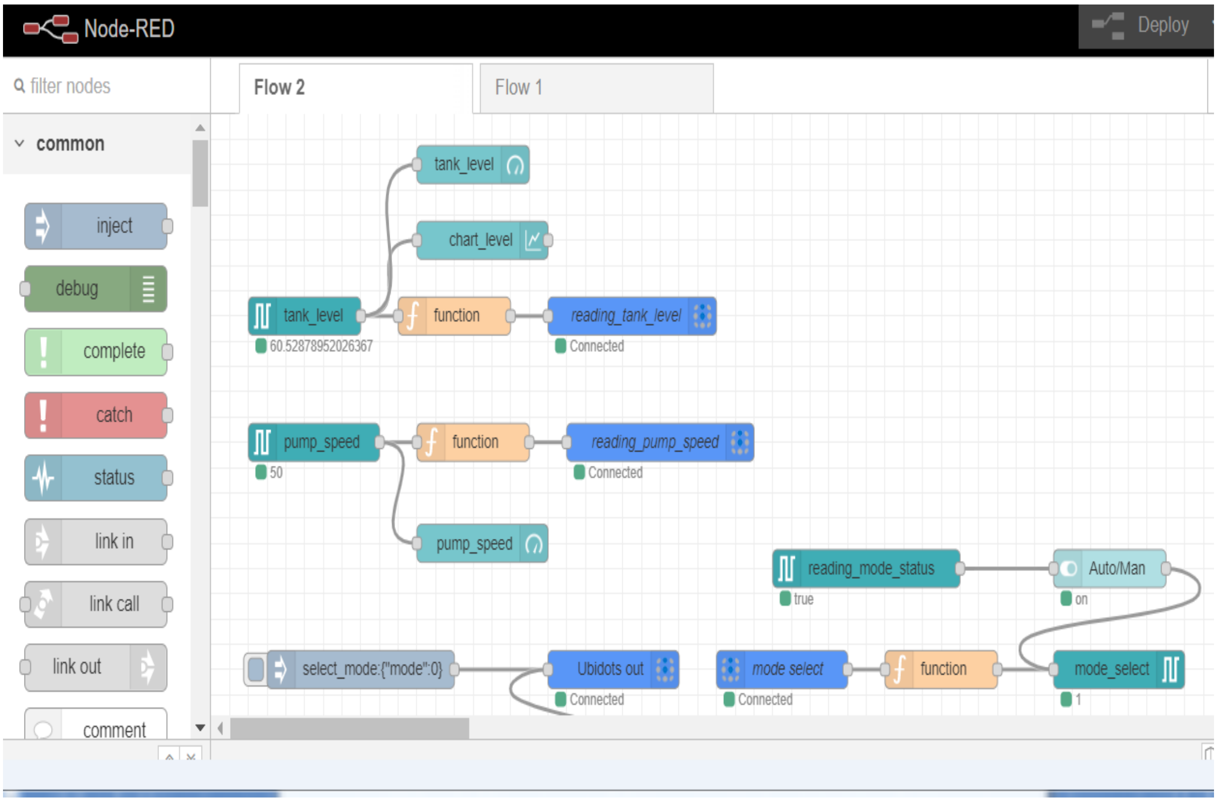
Name: The Label of the node in the Node-Red workspace. If empty, defaults to: "Ubidots in". This library uses an MQTT implementation hence, it's necessary to name this MQTT client. We recommend you to choose a non-easy-to-copy name, to prevent cases where the name crashes between different clients in the broker.

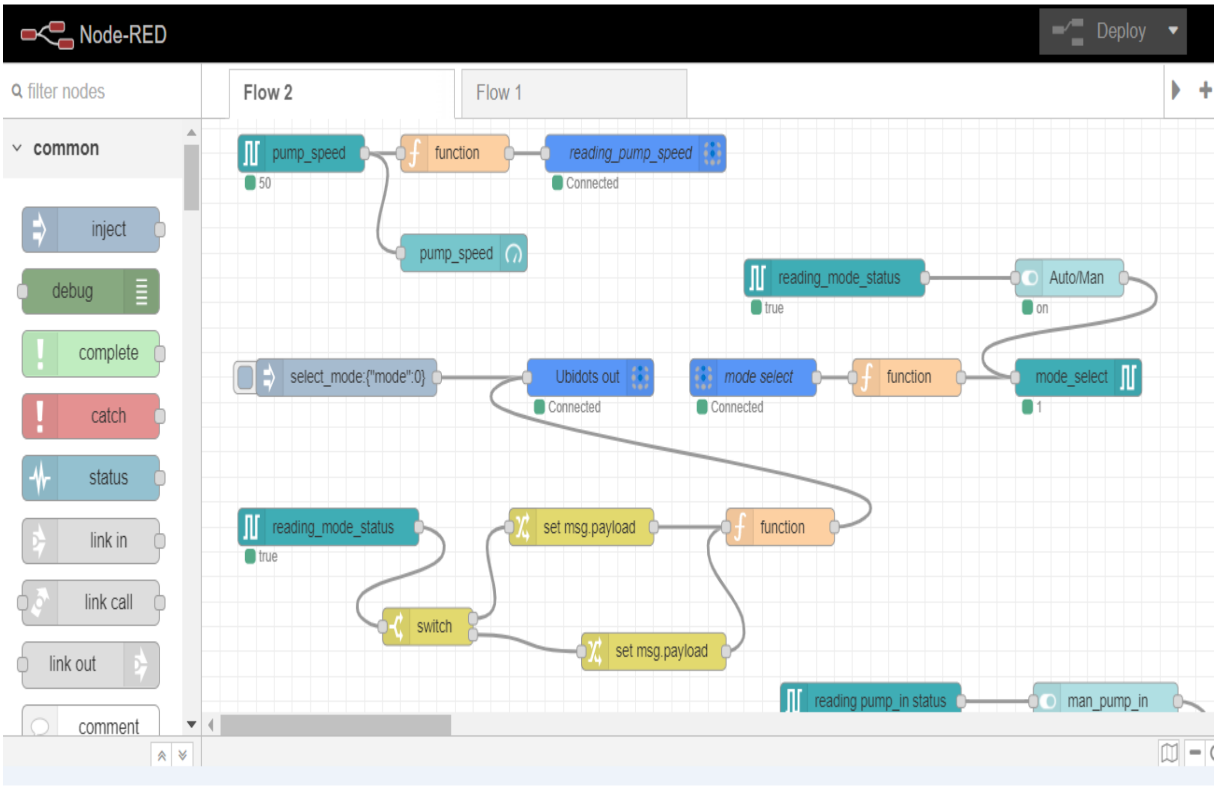
Token: The Token necessary to authenticate the connection to your Ubidots account. To obtain your token, login on ubidots.com, under "My Profile" click on "API Credentials".

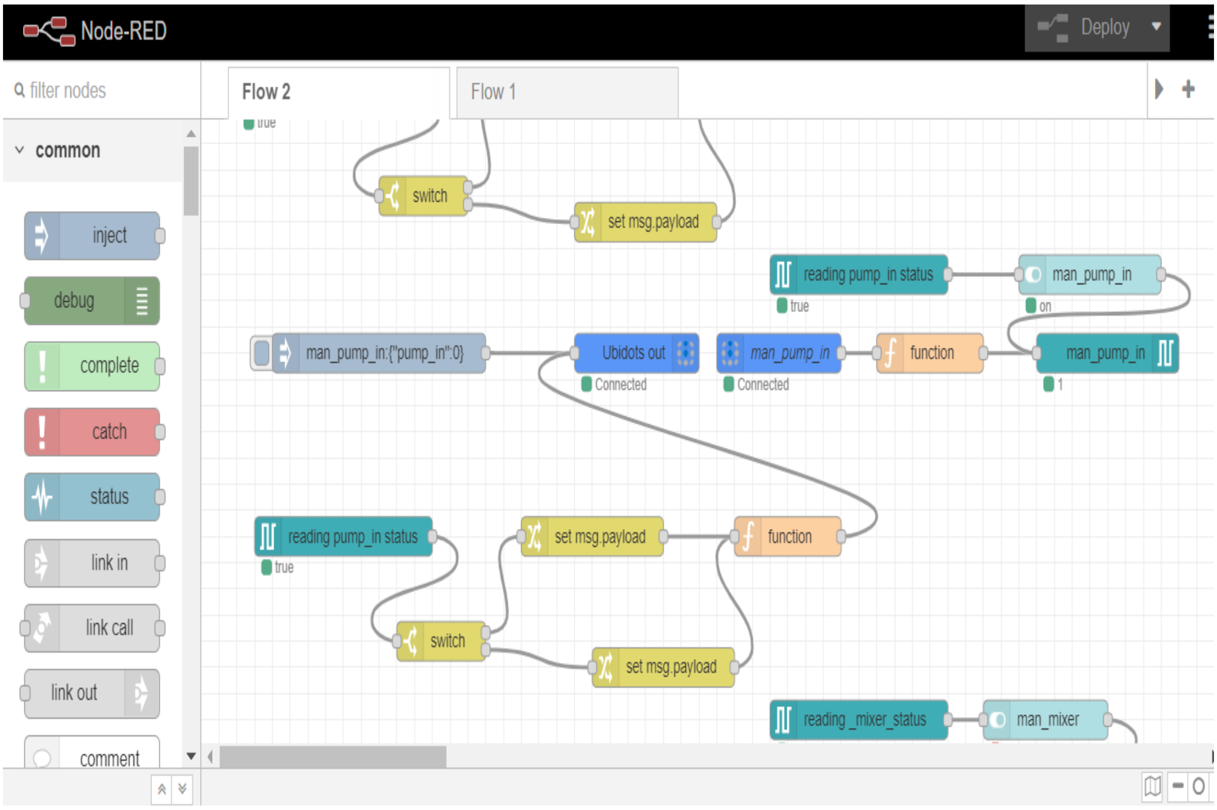
Device label: The label of the Device to which the node subscribes.

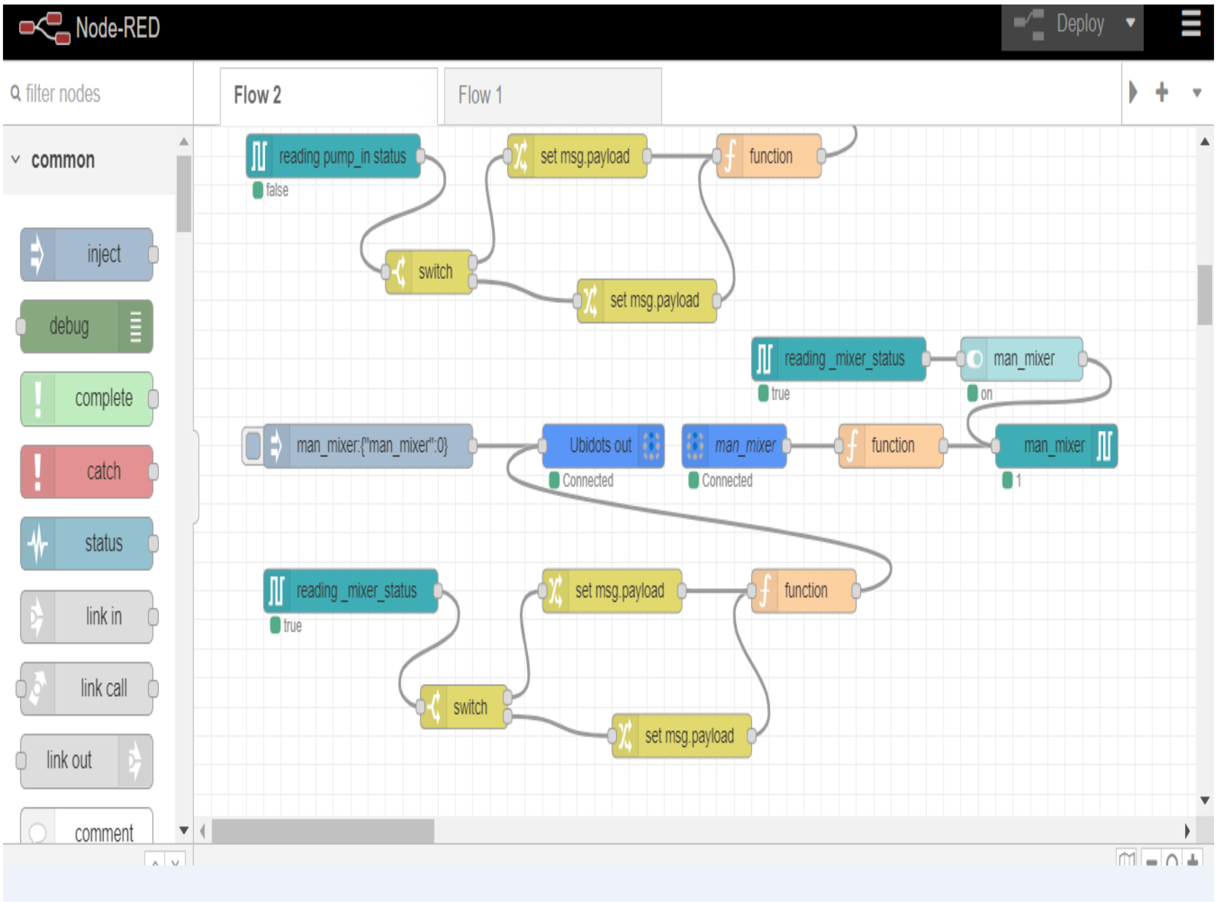
TLS: By default, all data is sent encrypted via TLS. Uncheck if data should be sent unencrypted

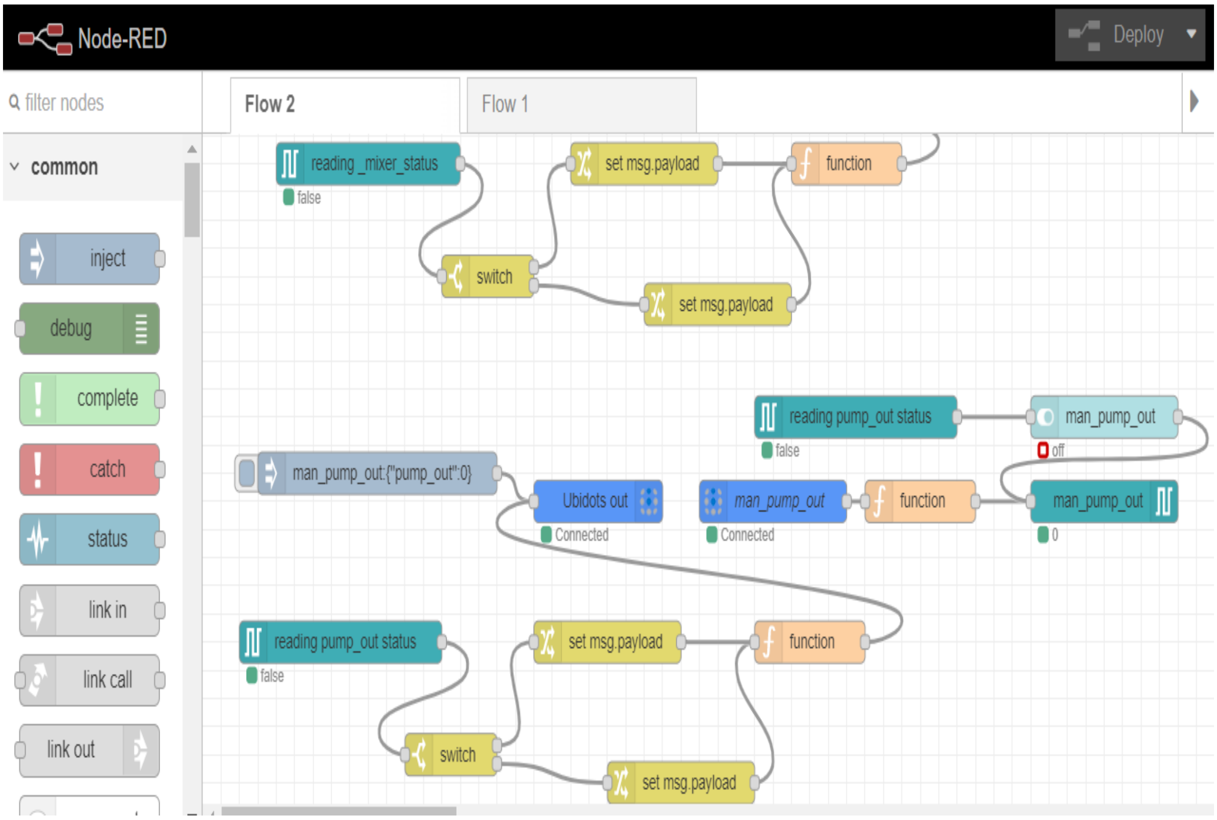
4.4.8 Node Red Program

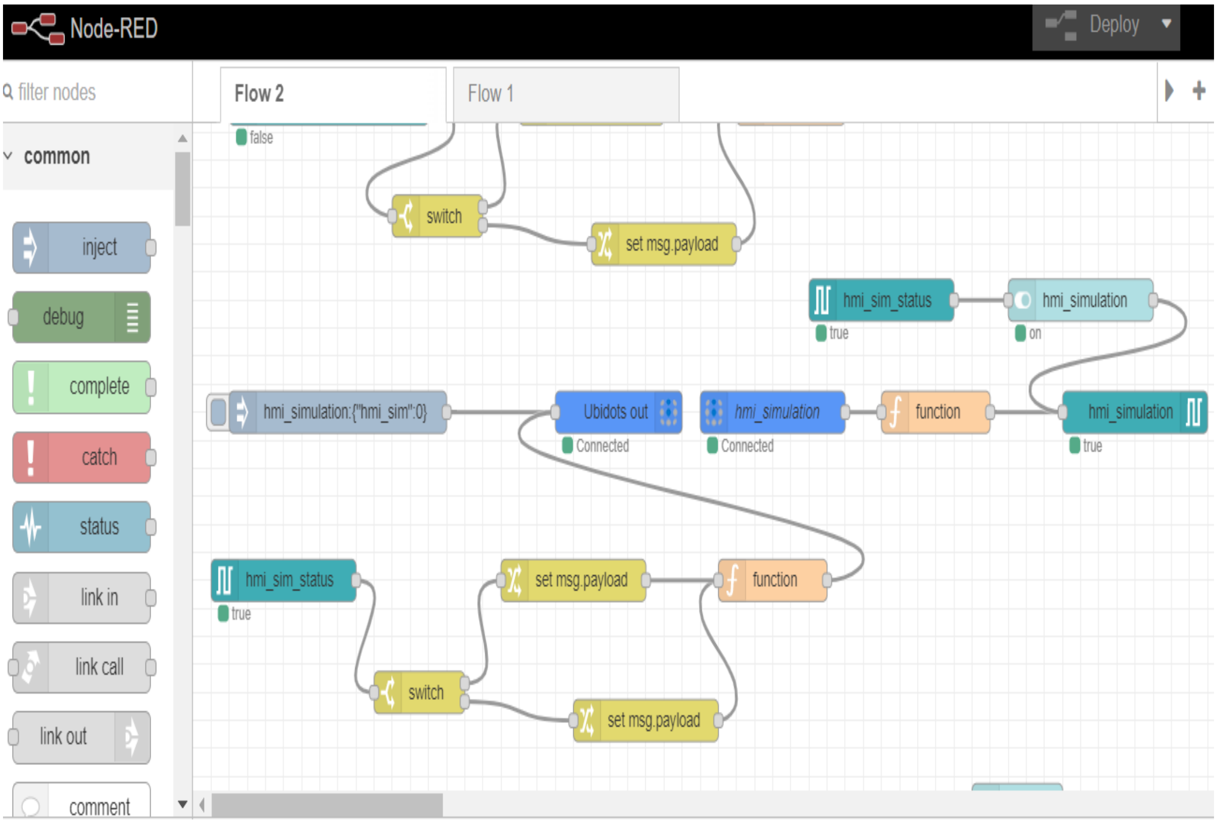


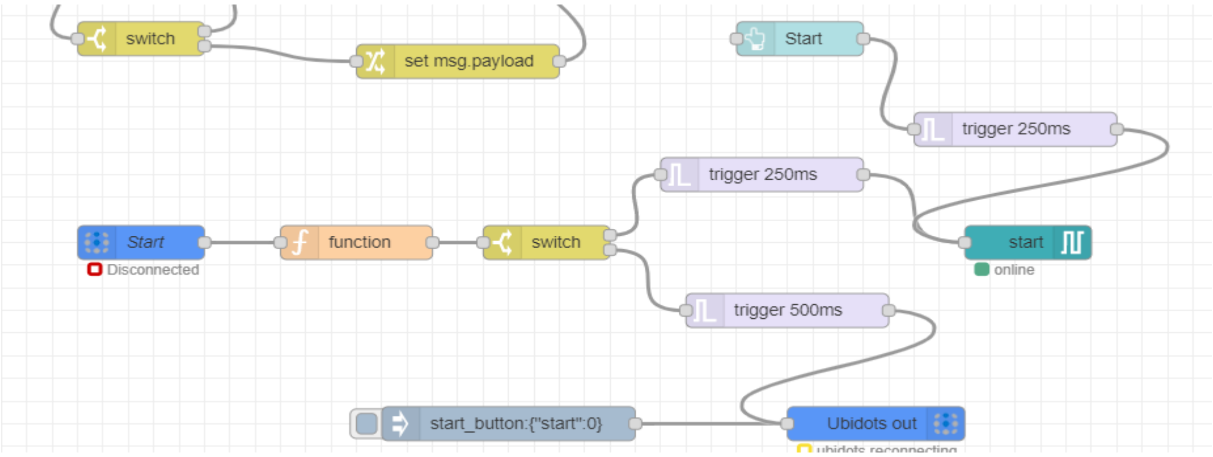


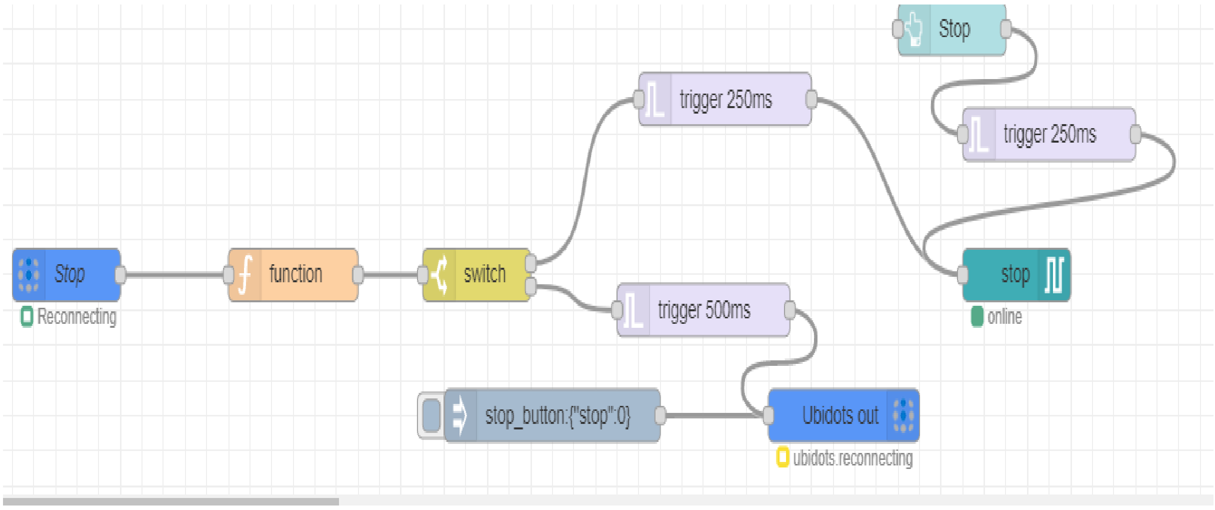

Node-RED
Deploy
filter nodes
Flow 2
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
true
switch
set msg.payload
reading pump_in status
true
man_pump_in
on
man_pump_in:{"pump_in":0}
Ubidots out
Connected
man_pump_in
Connected
function
man_pump_in
1
reading pump_in status
true
set msg.payload
function
switch
set msg.payload
reading _mixer_status
man_mixer

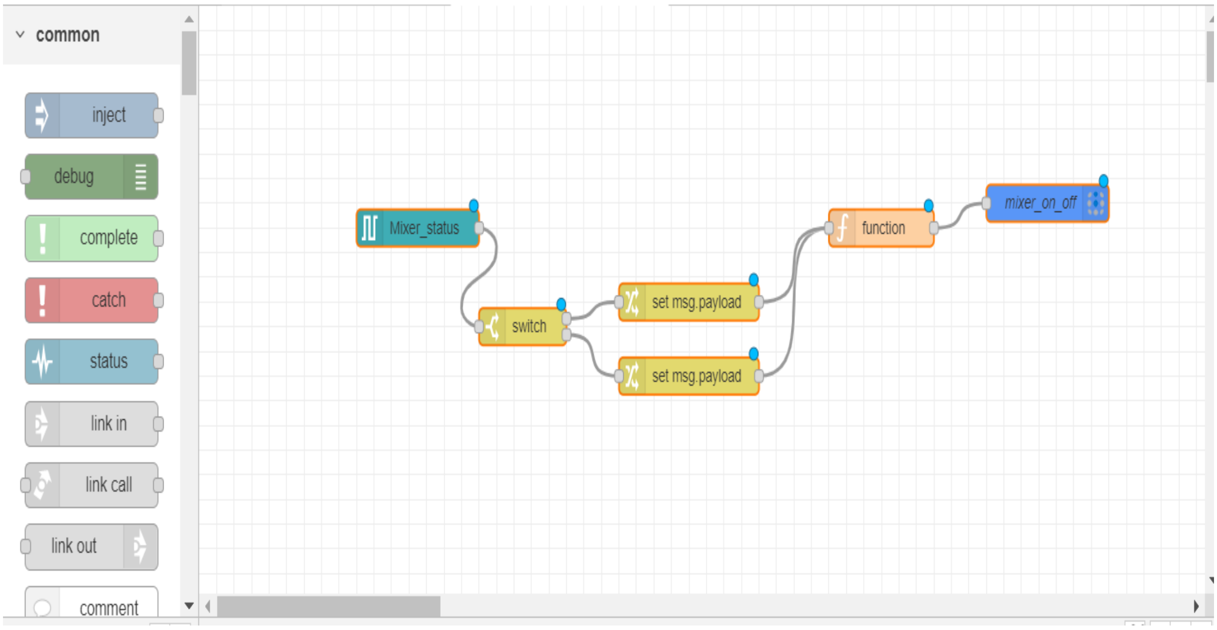

Node-RED
Deploy
filter nodes
Flow 2
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
reading pump_in status
false
set msg.payload
function
switch
set msg.payload
reading _mixer_status
true
man_mixer
on
man_mixer:{"man_mixer":0}
Ubidots out
Connected
man_mixer
Connected
function
man_mixer
1
reading _mixer_status
true
set msg.payload
function
switch
set msg.payload


Node-RED
Deploy
filter nodes
Flow 2
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
reading_mixer_status
false
set msg.payload
function
switch
set msg.payload
reading pump_out status
false
man_pump_out
off
man_pump_out:{"pump_out":0}
Ubidots out
Connected
man_pump_out
Connected
function
man_pump_out
0
reading pump_out status
false
set msg.payload
function
switch
set msg.payload


Node-RED
Deploy
filter nodes
Flow 2
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
false
switch
set msg.payload
hmi_sim_status
true
hmi_simulation
on
hmi_simulation:{"hmi_sim":0}
Ubidots out
Connected
hmi_simulation
Connected
function
hmi_simulation
true
hmi_sim_status
true
set msg.payload
function
switch
set msg.payload


switch
set msg.payload
Start
trigger 250ms
trigger 250ms
Start
Disconnected
function
switch
start
online
trigger 500ms
start_button:{"start":0}
Ubidots out
ubidots.reconnecting


Stop
trigger 250ms
trigger 250ms
Stop
Reconnecting
function
switch
stop
online
trigger 500ms
stop_button:{"stop":0}
Ubidots out
ubidots.reconnecting


common
inject
debug
complete
catch
status
link in
link call
link out
comment
Mixer_status
switch
set msg.payload
set msg.payload
function
mixer_on_off

Each connection to a PLC is represented by the **S7 Endpoint** configuration node. You can configure the PLC's Address, the variables available and their addresses, and the cycle time for reading the variables.

A single variable can be selected from the configured variables, and a message is sent every cycle, or only when it changes if *diff* is checked. msg.payload contains the variable's value and msg.topic has the variable's name.

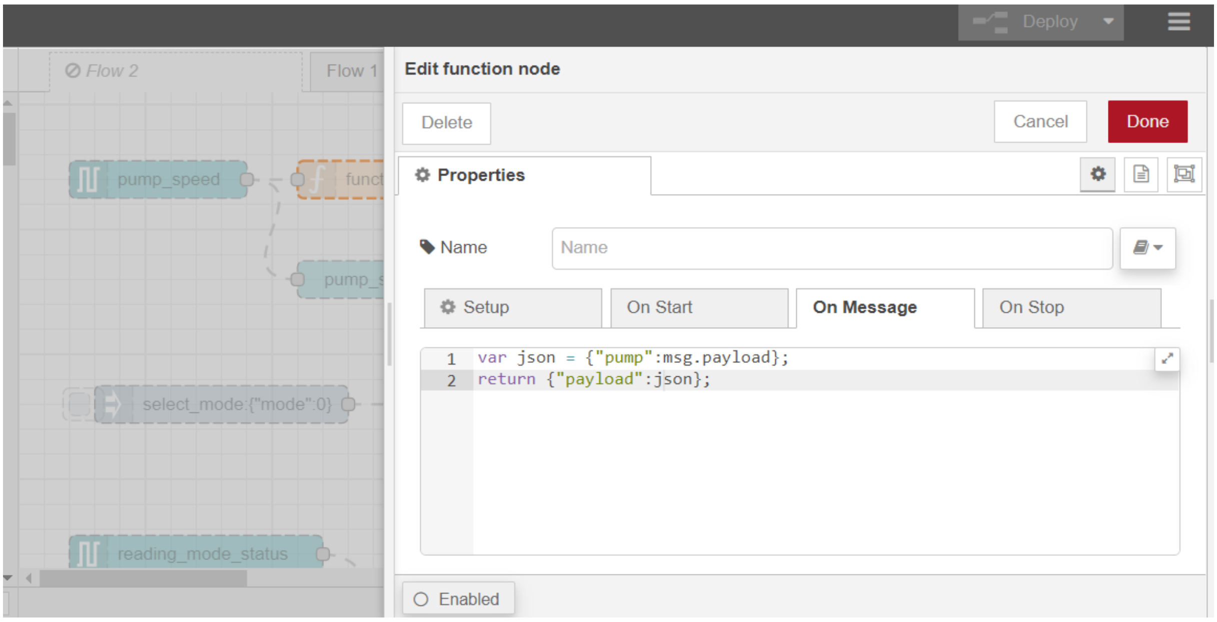
There are two different nodes: One for subscribing (reading) information from Ubidots and another one for publishing (sending) information to Ubidots.

Ubidots Out:

This node is used to publish to an Ubidots Variable. It will receive a value from a previous node and publish it to your variable on Ubidots.

msg.payload: The payload will contain all the values that are sent to the Device. It's structured to use the key of the object as the variable label and the value of the key as the value to send to Ubidots, e.g. `{"variable-label": 42}`

Sending a Json message that contains a key and a value (the value was received from S7 in node) using function node



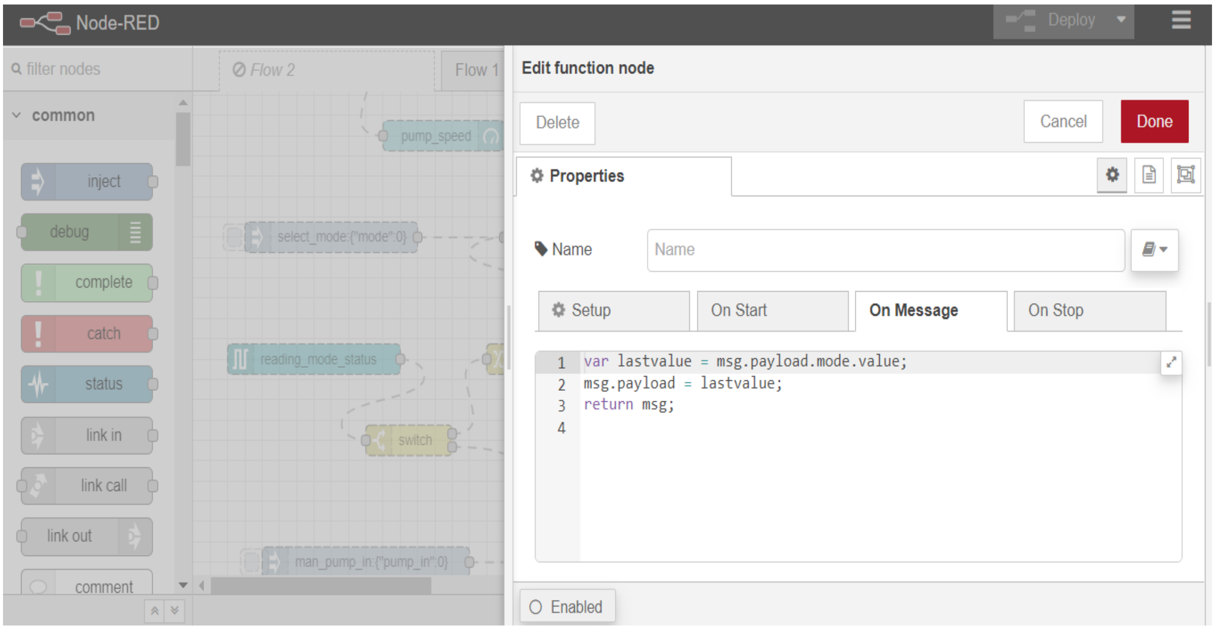
```
var json = {"pump":msg.payload};
return {"payload":json};
```

Ubidots In

This node (Subscriber) is used to subscribe to one or more (up to 10) Ubidots Variable(s) or custom topics. It will listen to new values and pass it to further nodes in the msg.payload.

The node uses the MQTT library to establish the connection and suscribe to the variable or topic.

Reading key value from ubidots in node and sending a msg object using a function node



Name: Name

```
var lastvalue = msg.payload.mode.value;
msg.payload = lastvalue;
return msg;
```

4.5 Remote Process Control

4.5.1 Remote monitoring and control of the system in LAN

A local area network (LAN) is a group of computers and peripheral devices that share a common communications line or wireless link to a server within a distinct geographic area. A local area network may serve as few as two or three users in a home office or thousands of users in a corporation's central office.

4.5.1.1 Accessing SIMATIC S7-1200 web server

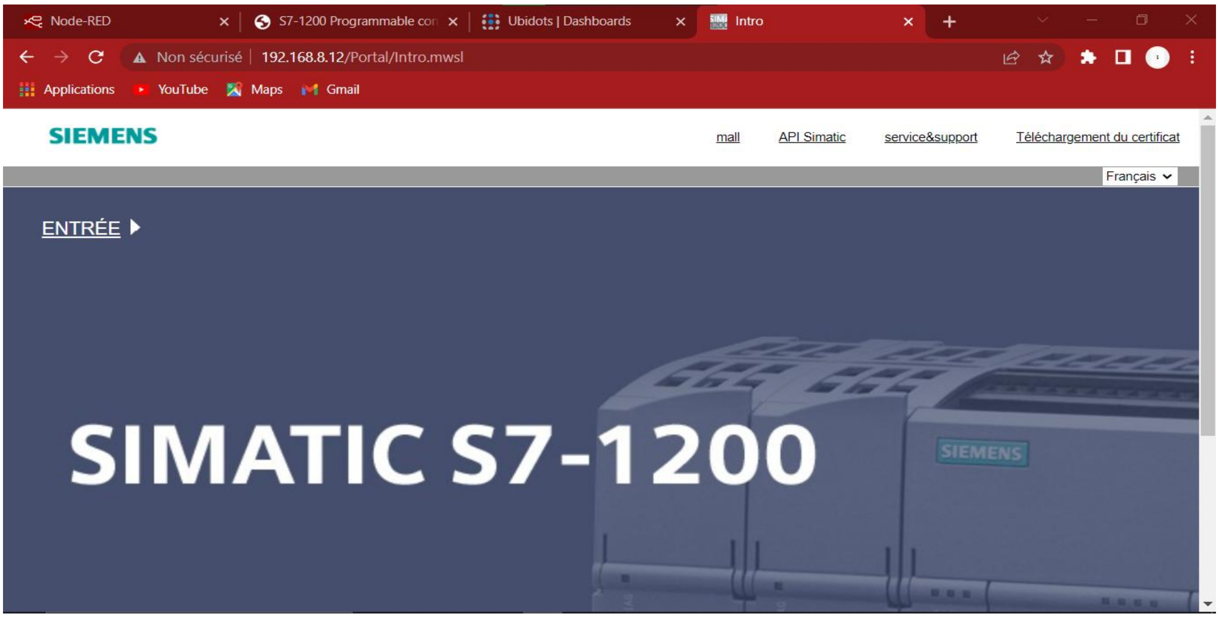
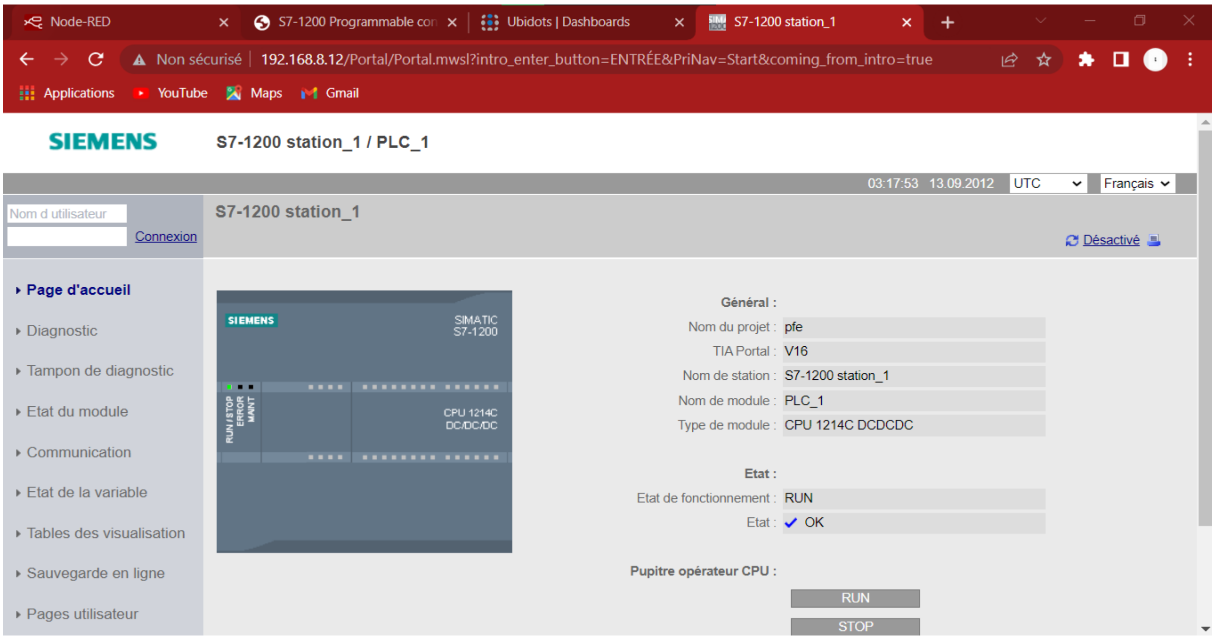


Figure 82: welcome page of the web server

To access the PLC web server we need to browse to its IP address using a web browser



Général :

Nom du projet : pfe

TIA Portal : V16

Nom de station : S7-1200 station_1

Nom de module : PLC_1

Type de module : CPU 1214C DCDCDC

Etat :

Etat de fonctionnement : RUN

Etat : ✓ OK

Pupitre opérateur CPU :

Figure 83: Web server Home page

PLC web server provides many features like Diagnostic buffer – module information etc we need to choose user pages to be able to access the user defined page

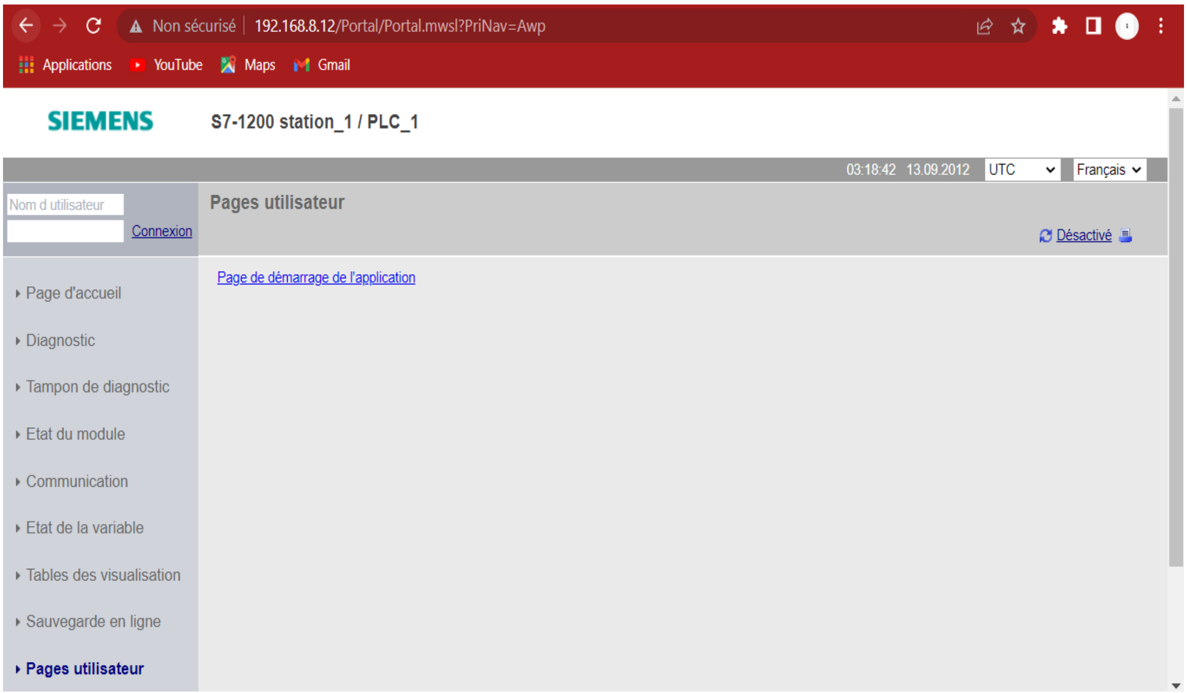


Figure 84: user page

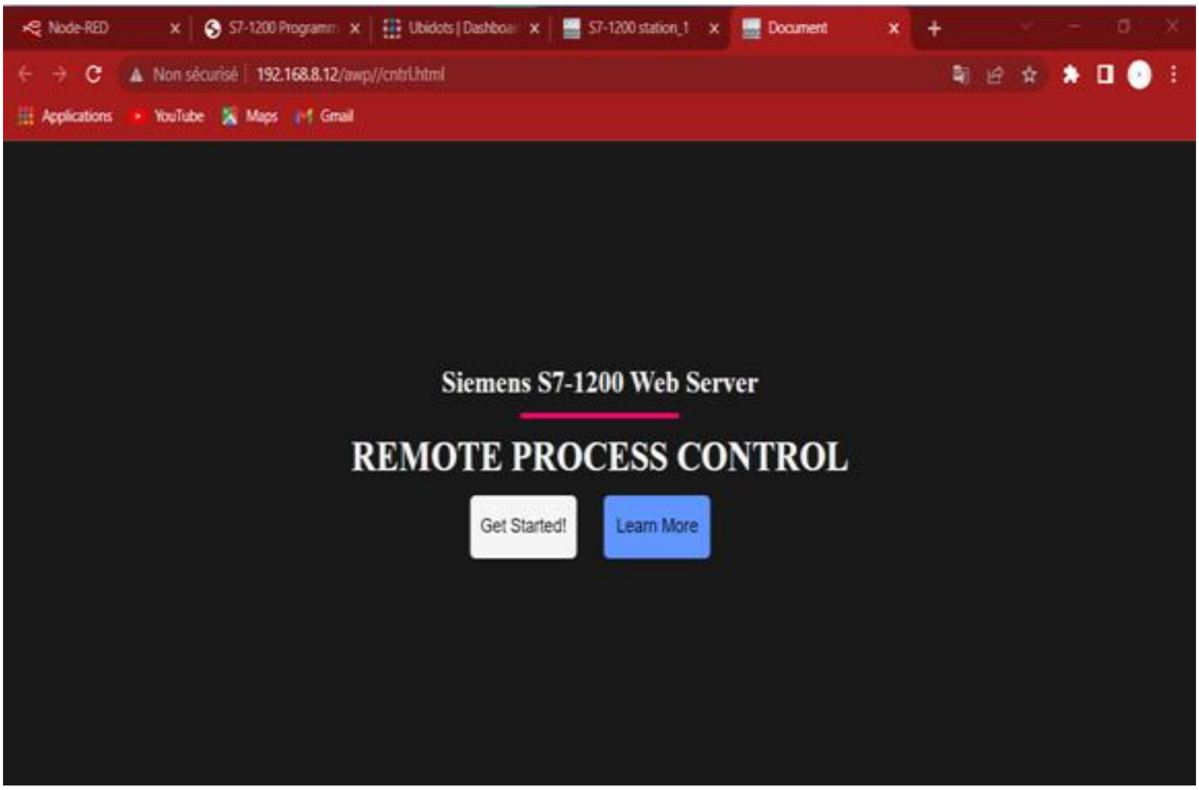
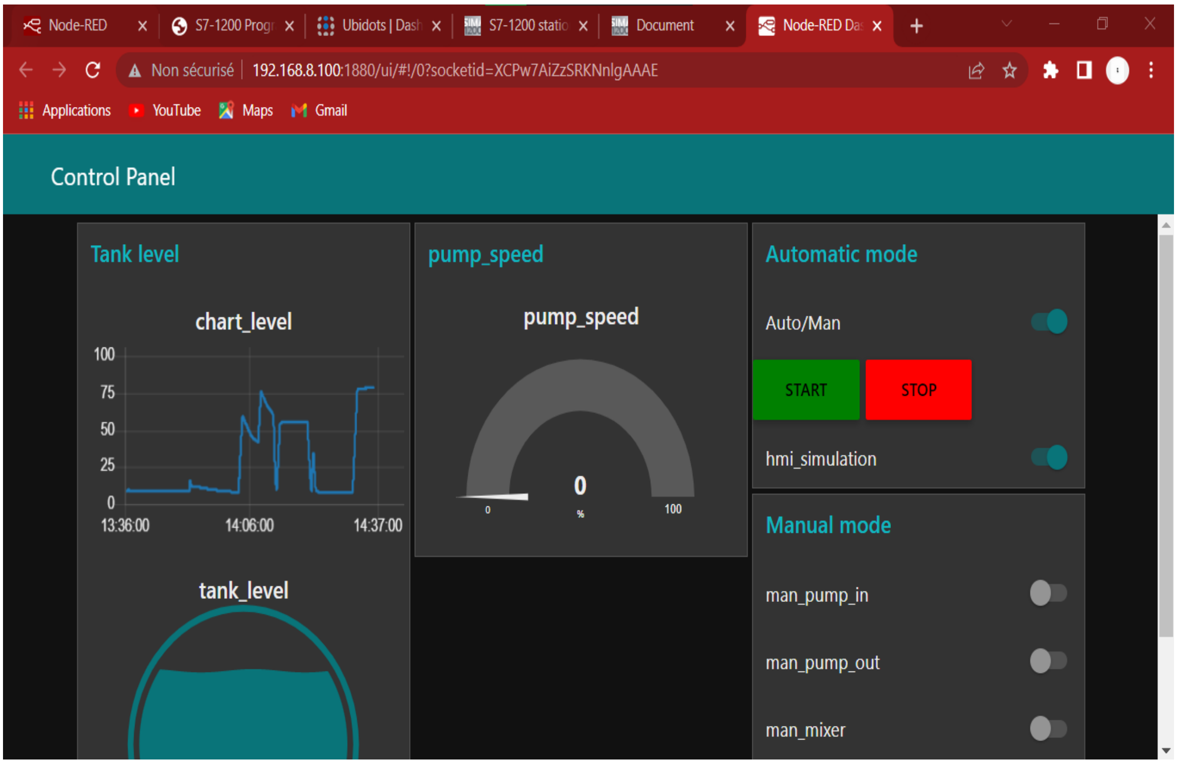


Figure 85: user defined page

After clicking on Get Started a new page will open with node-red dashboard on it.



This dashboard was designed using node red as explained previously.

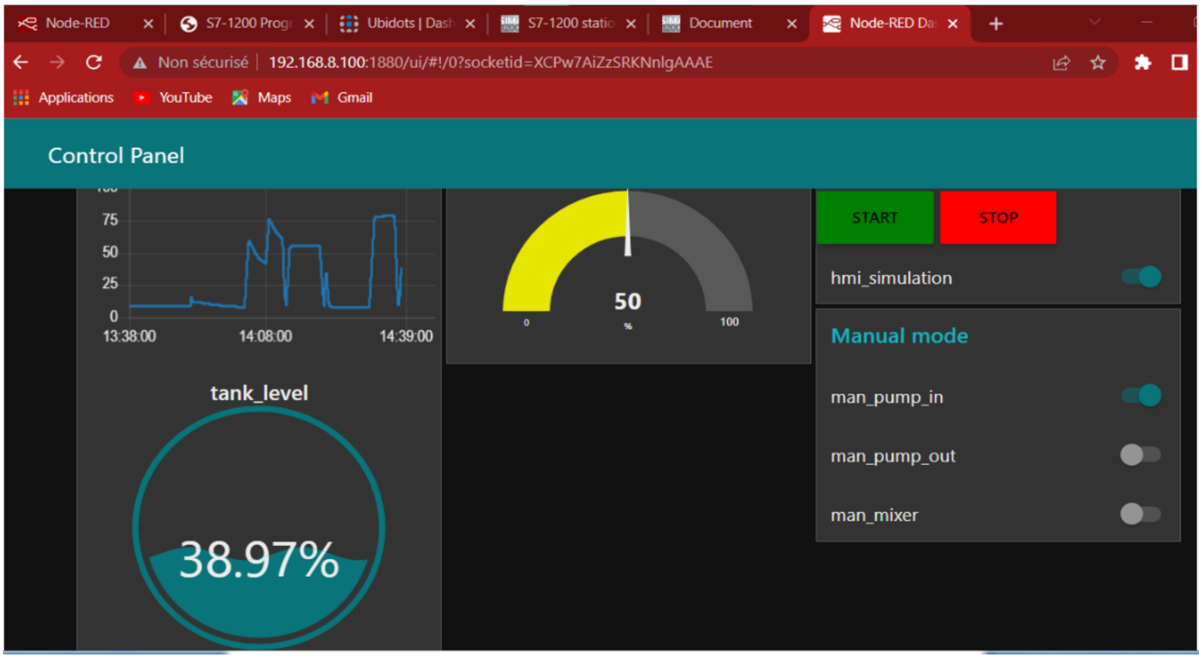


Figure 86: node red dashboard

4.5.2 Remote Monitoring and Control of the system from any place in the world

A wide area network (also known as WAN), is a large network of information that is not tied to a single location. WANs can facilitate communication, the sharing of information and much more between devices from around the world through a WAN provider.

4.5.2.1 Accessing Ubidots IoT platform dashboard

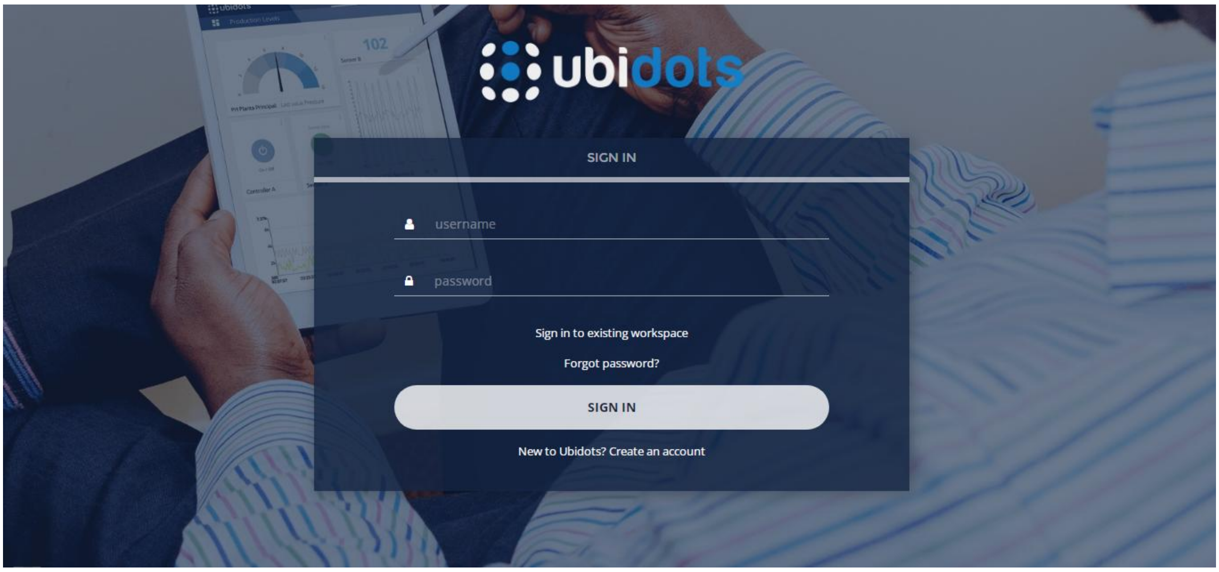
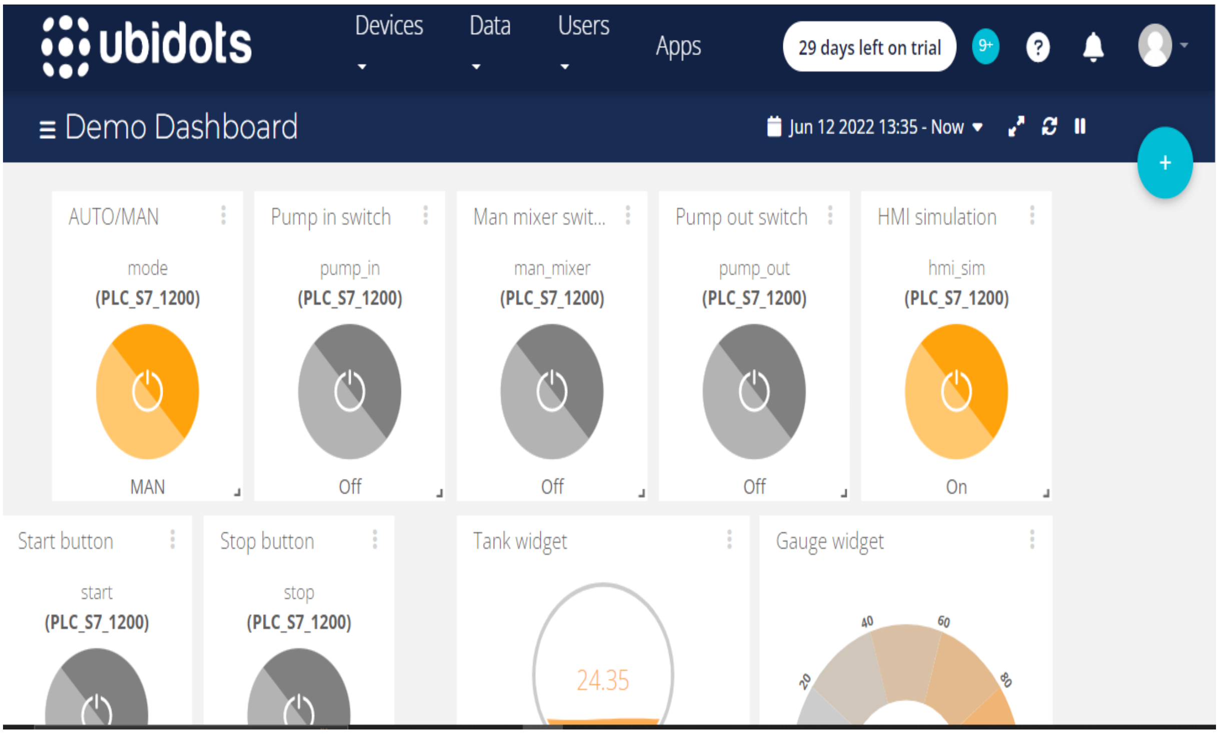

ubidots
SIGN IN
username
password
Sign in to existing workspace
Forgot password?
SIGN IN
New to Ubidots? Create an account

Figure 87: Ubidots log in page



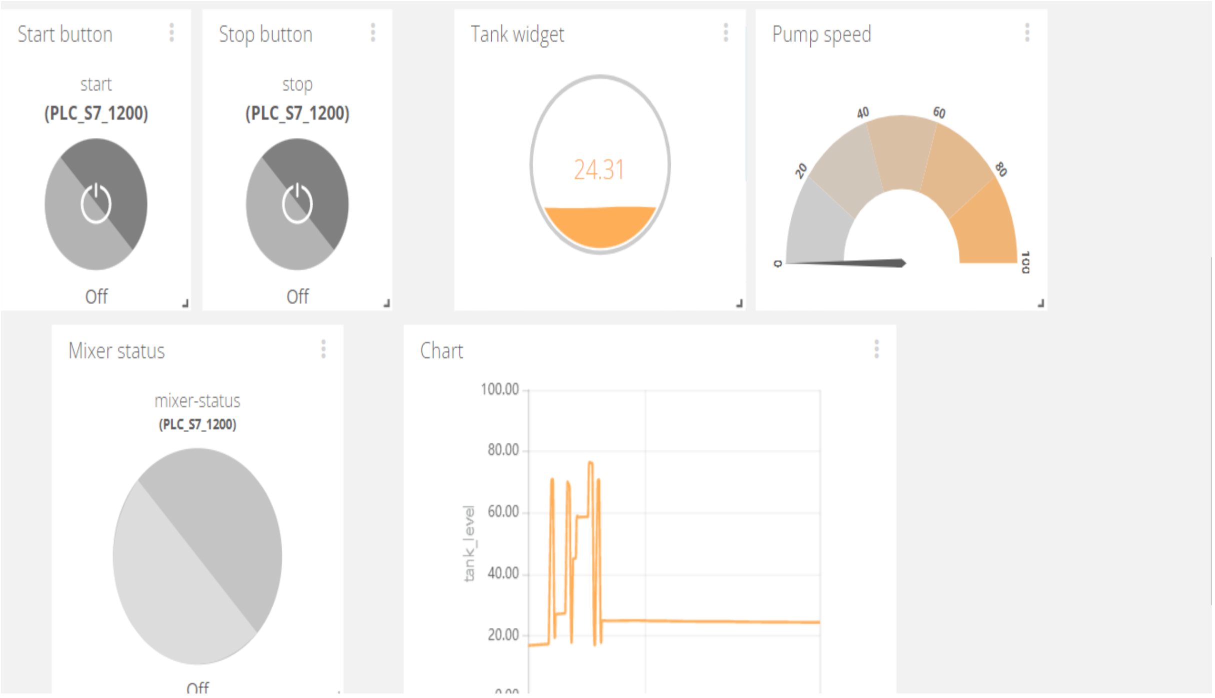


Figure 88: Ubidots dashboard

4.6 Results and Disscussion

After the connection of the physical and digital system that was shown in the previous chapter, the system was put into action and yielded relatively good results but still had some flaws. These results and flaws are discussed throughout this chapter.

4.6.1 Sequencing

First the sequencing of the operations was perfect. The outputs from the PLC were excited as desired according to different input patterns.

4.6.2 PLC web server with node red dashboard

4.6.2.1 Manual mode

-The two pumps started and stopped when the switch was ‘ON’ and ‘OFF’ respectively.

- The mixer started and stopped when the switch was “ON “and “OFF” respectively

4.6.2.2 Automatic mode

-The Actuators started when the start button was pressed and they stopped when stop button was pressed

4.6.3 Ubidots IoT platform dashboard

4.6.3.1 Manual mode

-The two pumps started and stopped when the switch was ‘ON’ and ‘OFF’ respectively.
- The mixer started and stopped when the switch was “ON “and “IFF” respectively

4.6.3.2 Automatic mode

-The Actuators started when the start button was pressed and they stopped when stop button was pressed

4.6.4 Remote monitoring and control

Benefits of Remote monitoring and control:
-Improved Operational Efficiency
- Detailed Data Collection
- Cost Savings
- Proactive maintenance

4.6.5 Components

Different equipment may give similar results for a specific task but some of them may provide larger system flexibility than others. For example using an Electro-valve to drain water back to sump tank may give the system better flexibility.

4.6.6 IoT connectivity

4.6.6.1 4G Signal

Weak 4G signal affects the real time monitoring and control of the system especially when it is done from a far place

4.6.6.2 Bandwidth

Bandwidth consumption is one of important challenges of connectivity in IoT. When bandwidth relies on cellular network (4G LTE) then it becomes expensive.

4.6.6.3 Security

Hacking is always an issue when it comes to IoT connectivity. In reality, any device connected to the Internet is at risk of hacking

4.7 Conclusion

After the project was fulfilled in all stages (design-programming-simulation-implementation) several facts were concluded.

1- Industrial automation is cost-effective in the long term but its initial costs are very high since in order to achieve optimum performance high quality components are required. Recall The discussion in section **4.6.5**

2- Remote Monitoring and control of the industrial systems has several benefits that lead to a better performance. Recall The discussion in section **4.6.4**

3- IoT connectivity has some problems that need to be solved. Recall The discussion in section **4.6.5**

4- IoT is one of the driving forces behind the Industry 4.0 movement, as it will enable enhanced automation, data collection, and analytics, in addition to optimizing workflows and processes.

Final Statement:

Industries potentially use SCADA systems to control, monitor, and analyze their processes in real-time. SCADA systems in industries are proving highly efficient as they interact directly with the devices like sensors, pumps, valves, motors, and record events to allow better decision-making. It is mainly because, a SCADA system maintains the efficiency of the industrial process by communicating any issue in the system, which further helps in reducing the machine downtime.

The Internet of Things is a disruptive technology that comprises of sensors, gateways, and interactive dashboards to enhance business productivity at all aspects. The current industrial SCADA system is limited to factory floors, and the data fetched from the factory equipment is only viewed inside the plant. IoT can play a major role here by enhancing the data accessibility and operability of various operations across industries. Therefore, the implementation of IoT technology with the existing SCADA systems can certainly empower the industries with its most amazing features like scalability, interoperability, and enhanced security of the overall industrial processing. Moreover, IoT can coexist with the SCADA system, as it is here to change the entire landscape of SCADA. The foremost step here would be to identify the right IoT platform that can provide different varieties of adapters and connectors to integrate with the SCADA system [70].

SCADA and IIoT (Industrial Internet of Things) concepts and their overall system architecture face cyber security challenges since they are a likely target for advanced hacking groups.

Here are the major benefits of adopting the Internet of Things in a SCADA system:

- Remote monitoring and control
- Predictive analysis.
- Overall equipment effectiveness
- Data analytics
- Cloud assistance

Reference

[1] Keyur K Pate, Sunil M Patel**.** Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges

https://www.researchgate.net/publication/330425585_Internet_of_Things-IOT_Definition_Characteristics_Architecture_Enabling_Technologies_Application_Future_Challenges

[2] https://data-flair.training/blogs/iot-tutorial/

[3] Sandra Khvoynitskaya IoT expert article https://www.itransition.com/blog/iot-history

[4] Bruno Dorsemaine. "Internet of Things: A Definition & Taxonomy". In: *2015 9th* International Conference on Next Generation Mobile Applications, Services and Technologies. IEEE.2015, pp. 72–77.

[5] Roberto Minerva, Abyi Biru , and Domenico Rotondi. "Towards a Definition of the Internet of Things (IoT)".In: IEEE Internet Initiative 1 (2015), pp.1–86.

[6] Dr**.** Ovidiu Vermesan. "Internet of Things Strategic Research Road map". In: Internet of Things-Global Technological and Societal Trends 1.2011 (2011), pp.9–52.

[7] Internet of Things Defined–Tech Definitions by Gartner. https://www.gartner. com/it-glossary/internet-of-things.htm

[8] Gerd Kortuem. "Smart Objects as Building Blocks for the Internet of Things". In: IEEE Internet Computing 14.1 (2009), pp.44–51

[9] Dr. Ovidiu Vermesan SINTEF, Norway, Dr. Peter Friess EU, Belgium, "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems", river publishers' series in communications, 2013

[10] Dr. Ovidiu Vermesan SINTEF, Norway, Dr. Peter Friess EU, Belgium, "Internet of Things–From Research and Innovation to Market Deployment", river publishers' series in communications, 2014.

[11] http://www.reloade.com/blog/2013/12/6characteristics-within-internet-things-iot.php

[12] https://www.imda.gov.sg/-/media/imda/files/industry-development/infrastructure/technology/internetofthings.pdf%3Fla%3Den

[13] https://www.jigsawacademy.com/top-uses-of-iot/

[14] K. Rose, S. Eldridge, L. Chapin, The internet of things: an overview

[15] D. Lukač, "The fourth ICT-based industrial revolution", 23rd Telecommunications

Forum Telfor, IEEE, 2015, pp. 835–838.

[16] https://www.dbresearch.com/PROD/RPS_EN-PROD/PROD0000000000451959/Industry_4_0%3A_Upgrading_of_Germany%E2%80%99s_industrial_ca.pdf?undefined&realload=LyjOCwkLGcH~/pKfzijLAgPPPolSWvqmYq8DifTo7fAH8go0tUoxYKezD9RUzYFA

[17] Hugh Boyes, Bil Hallaq, Joe Cunningham, Tim Watson The industrial internet of things (IIoT): An analysis framework
https://www.sciencedirect.com/science/article/pii/S0166361517307285

[18] M. Hermann, T. Pentek, B. Otto, Design Principles for Industry 4.0 Scenarios: A Literature Review, Technische Universität Dotmund, 2015, p. 11 working paper

[19] **H**. Boyes, A security framework for cyber-physical systems, WMG CSC Working Paper, Coventry, University of Warwick, 2017

Cyber security for Industry 4.0 pp 73–101 SCADA System Forensic Analysis Within IIoT
https://link.springer.com/chapter/10.1007/978-3-319-50660-9_4

[20] S.A. Boyer, SCADA: Supervisory Control and Data Acquisition, Instrument Society Of America, Research Triangle Park, North Carolina, 1993 ISBN 1-55617-210-9. p.9

[21] Dpstele.com, What Is SCADA? Real-world Remote Monitoring Elements and Applications**.** https://www.dpstele.com/scada/what-is.php

[22] M. Fahrion, Evolving from SCADA to IoT. Remote Monitoring and Control Conference

[23] David. Floyer, Defining and Sizing the Industrial Internet, Wikibon
http://wikibon.org/wiki/v/Defining_and_Sizing_the_Industrial_Internet

[24] Industrial Internet insights report for 2015
https://www.ge.com/news/taxonomy/term/4386

[25] https://www.impulse-embedded.co.uk/solutions/what-is-an-industrial-iot-gateway.htm

[26] https://www.techtarget.com/whatis/definition/MindSphere

[27] Wesley Chai, Kate Brush, Stephen J. Bigelow. What is Paas? Platform as a service definition and guide

https://www.techtarget.com/searchcloudcomputing/definition/Platform-as-a-Service-PaaS#:~:text=Platform%20as%20a%20service%20(PaaS,software%20on%20its%20own%20infrastructure.

[28] https://www.ibm.com/cloud/architecture/architectures/industry-40/

[29] https://www.ibm.com/cloud/learn/introduction-to-private-cloud

[30] Alexander S. Gillis. Amazon Web Services (AWS) https://www.techtarget.com/searchaws/definition/Amazon-Web-Services

[31] https://aws.amazon.com/types-of-cloud-computing/

[32] https://behrtech.com/blog/mqtt-in-the-iot-architecture/

[33] MQTT version 5.0 OASIS https://doc.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[34] https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP

[35] https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/

[36] https://www.thetechplatform.com/post/what-is-http-protocol-architecture-and-components-of-http
[37] https://www.engineersgarage.com/ethernet-technology-iot-part-23/

[38] https://www.kelltontech.com/kellton-tech-blog/internet-of-things-protocols-standards

[39] Kamil, M.S., 2009. The Performance Evaluation of OFDM Based WLAN. Diva Portal, pp.1-48. https://www.diva-portal.org/smash/get/diva2:830377/FULLTEXT01.pdf

[40] Bradley, M., 2017. Introduction to Wi-Fi Wireless Networking. https://www.lifewire.com/introduction-to-wi-fi-wireless-networking-818265

[41] Guo, Y., Zhang, S., Xiao, D., 2012. Overview of Wi-Fi Technology. The 2nd International Conference on Computer Application and System Modeling (2012) https://www.atlantis-press.com/article/2794.pdf

[42] N.S Arinze, D. Abonyi Evaluation of Wireless Fidelity performance in Nigeria’s mobile Networks https://www.researchgate.net/publication/339601913

[43] https://www.iotforall.com/wifi-role-iot

[44] Sean Michael Kerner 4G (fourth-Generation wireless) https://www.techtarget.com/searchmobilecomputing/definition/4G

[45] https://pelion.com/blog/education/4g-lte-iot-connectivity/

[46] Sebastian Jokinen. Industrial IoT Comparison of Cloud Integration https://www.theseus.fi/bitstream/handle/10024/507949/Jokinen_Sebastian_Thesis.pdf?sequence=2

[47] https://en.wikipedia.org/wiki/JSON

[48] Heath, Nick (March 13, 2014*).* "How IBM's Node-RED is hacking together the Internet of things" https://www.techrepublic.com/article/node-red/

[49] https://nodered.org/blog/2016/06/14/version-0-14-released

[50] Lewis, Karen, "Node-RED visual programming for the Internet of Things (IoT) is now a JS Foundation Project" https://www.ibm.com/blogs/internet-of-things/open-source-iot/

[51] S7-1200 Programmable controller. System Manual https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf

[52] https://support.industry.siemens.com/cs/mdm/91696622?c=58501434891&lc=en-AE

[53] Analog Values for SIMATIC S7-1200, https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/basics-programming-s7-1200/sce-031-500-analog-values-s7-1200-r1709-en.docx

[54] PCT50 Level Control, https://armfield.co.uk/product/pct50-level-control/

[55] Industrial Ethernet Switches SCALANCE X-200, https://cache.industry.siemens.com/dl/files/962/102051962/att_979080/v1/BA_SCALANCE-X-200_76.pdf

[56] SITOP PSU100S manual, https://media.automation24.com/manual/us/67476474_PSU100S_Handbuch_englisch_en-US.pdf

[57] https://www.pannam.com/blog/what-is-industrial-hmi/

[58] Siemens HMI TP900 Comfort, https://ie.rs-online.com/web/p/hmi-displays/8643973

[59] Bipolar circuit breaker, https://diruptor.com/en/our-breakers/bipolar-circuit-breaker/#:~:text=This%20bipolar%20circuit%20breaker%20is,230%20Volts%2050%2F60%20Hz)

[60] https://forum.huawei.com/enterprise/en/i-recommend-huawei-huawei-cpe-b310s/thread/754511-100181?page=1

[61] Bradley Mitchell, Ryan Perian, Ethernet cables and how they work https://www.lifewire.com/what-is-an-ethernet-cable-817548

[62] Julyann Tu, Siemens: Programming Software Selection Guide https://www.awc-inc.com/sub/siemens-programming-software-selection-guide/#:~:text=Totally%20Integrated%20Automation%20(TIA)%20Portal,%2C%20communication%2C%20and%20other%20features.

[63] Laddar logic, https://en.wikipedia.org/wiki/Ladder_logic

[64] HTML5, https://en.wikipedia.org/wiki/HTML5

[65] CSS, https://en.wikipedia.org/wiki/CSS

[66] JavaScript, https://en.wikipedia.org/wiki/JavaScript

[67] Node-Red**,** https://en.wikipedia.org/wiki/Node-RED

[68] JSON, https://en.wikipedia.org/wiki/JSON

[69] Jordan Garcia, connecting to Ubidots with NetBurner https://www.netburner.com/learn/connecting-to-ubidots-with-netburner/

[70] https://www.biz4intellia.com/blog/8-benefits-of-adopting-internet-of-things-(IoT)-in-scada-system/

[71] K Rajeswar CXO level advisor on technology Industry 4.0 wave - Relevance of SCADA in an IOT world and journey towards a true digital enterprise