الجمهـــوريـــــة الجزائـــريــــة الديمقـــراطيـــة الشـعبيـــة
République Algérienne Démocratique et Populaire
وزارة التـــعـلـــيـــــم العـــــالـــــــي والبـــحـــــث العــلـمـــي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
جامعة وهران 2 محمد بن أ حمد
Universitéd'Oran 2 Mohamed Ben Ahmed

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

Département de Génie Industriel

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industrielle
Spécialité : Maintenance des Automatismes et Instrumentation Industrielle

Thème

Supervisory and control of an automated level control using Tia portal and PLC S7-1200.

Présenté et soutenu par :

Nom Marsel **Prénom** Habib Ellah Nabil

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
		Université Oran 2	**Président**
AISSANI Nassima	MCA	Université Oran 2	**Encadreur**
		Université Oran 2	**Examinateur**

Année 2020/2021

بسم الله الرحمن الرحيم

إِنْ أُرِيدُ إِلَّا

الْإِصْلَاحَ مَا اسْتَطَعْتُ

وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ

عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ

سوره هود

ACKNOWLEDGEMENTS

My Praise and thanks go first of all to Almighty God for the strength and the will that has given me to complete my work.

I would like to warmly thank my supervisor, Madam Aissani Nassima, head of the Maintenance of Automation and Industrial Instrumentation specialty.

I would like to express my gratitude and deep appreciation to my supervisor, Madam Aissani Nassima, for her guidance, insightful comments, and suggestions. Her rigorous attention to the details over the course of this research from beginning to end has greatly improved my skills as professional automation engineer.

I would like to express my sincere gratefulness to my teachers at the Industrial Engineering Department of I.M.S.I during the University course for the last five years.

Finally, I have one last thought for my family, especially my parents who have unconditional trust and love in me and who have always supported me in what I do.

ملخص:

يختص هذا المشروع في برمجة وحدة تحكم **"S7-1200"** ببرنامج **"TIA Portal V16"** لأتمتة مستوى مياه الخزان والتحكم به بطريقة سريعة ودقيقة مع إنشاء واجهة آلة بشرية على **"Siemens TP900 comfort panel"** من أجل التشخيص، التحكم والإشراف على عملية ملء الخزان ومراقبة خلال عملية ملء الخزان عن طريق إطلاق إنذار **ESD** في حالة اكتشاف عدم المطابقة.

الكلمات المفتاحية: "جهاز التحكم المنطقي"، "S7-1200"، "**TIA Portal**"، "Siemens TP900 Comfort".

Résumé :

Ce projet consiste à programmer un contrôleur « S7-1200 » avec le logiciel « TIA Portal » pour automatiser et contrôler le niveau d'eau du réservoir de manière rapide et précise. Egalement, la création de l'interface homme machine sur "Panneau confort Siemens TP900" pour le diagnostic, le contrôle et l'affichage de la scène de process et la supervision en déclenchant une alarme ESD en cas de non conformité détectée.

Mots clés : ''Automate '', ''S7-1200'' ; ''TIA Portal''; ''Siemens TP900 Comfort''.

ABSTARCT:

This project consists of programming an "S7-1200" controller with "TIA Portal" software to automate and control the level of tank water quickly and precise. Also, the creation of the human machine interface on "Siemens TP900 comfort panel" for the diagnostic, control and the display of the process scene and the supervision by triggering an ESD alarm in the event of non conformity detected.

Keywords: ''Controller '', ''S7-1200''; ''TIA Portal''; ''Siemens TP900 Comfort''.

Table of contents

Table of contents

Chapter II: Tia Portal V16 & Factory I/O v2.4.2

Chapter III: Hardware Description (S7-1200 PLC & Process Parts)

Table of contents

List of Figures

Chapter I: Introduction to Automation

Chapter II: Tia Portal V16 & Factory I/O v2.4.2

List of Figures

List of Figures

List of Figures

General Introduction

Process control deals with the science of maintaining the output of a specific process within a desired range, the proportion of one ingredient to another, the temperature of materials, how well ingredients are mixed and the pressure under which materials are held can significantly impact the quality of an end product. That's why process control solutions are so important.

And the reasons behind the will to automate theses industrial processes are logically depend on how crucial are these parameters to our industry or application in terms of security and assurance.

In our case we have studied our tank water (inputs, outputs and the process characteristics itself), and using our automation tool, the Siemens S7-1200 programmable logic control with step7 professional or Totally Integrated automation portal version 16 we can create a specific program in order to control the tank water level by controlling the fill and drain pneumatic valves and get a feedback through a capacitive sensor, however so that the control to be more accurate and fast, we have developed two kind of control program:

1- **Semi Automatic** which we use a specific ranges to fill or drain our tank, it won't reached the exact setpoint, we use it for application with less hazard fluid.
2- **Full Automatic Mode** using the Siemens PID compact which we have the luxury to reach the exact desired setpoint faster and precisely by tuning the controller (proportional, integral and derivative)

Supervising the process is important as much as the controlling, because if we managed to automate the process without having access to monitoring it, so can say we lost the point of automation, which should help us to manufacture and produce product in simple, easy and secure way, and by supervising the installation from the central control room is more secure, which, we can stop the procedure in the event of non conformity detected.

For that we have create a human interface machine display with the Siemens TP900 comfort to monitor and control the level tank water using Wincc flexible integrated with Tia Portal.

This dissertation is divided into four chapters, and it is organized as follows:

§ **Chapter I:** We have presented in this chapter an Industrial automation and control, starting with a small introduction, history and how automation is related and integrate within the industry, then how industry automation is organized, finally describe most important controller in industry.

§ **Chapter II:** The second chapter is dedicated to a general presentation of the software we will be using to reach our goal which is automation the tank level control, so first of all we an overview about Tia portal, its basics, views...etc, then we talked about few general basic programming and how program organized within a specific blocks, after that we moved to the next software "the process scene simulator" which we described how it dedicated to help us reaching our goal.

§ **Chapter III:** The third chapter presents the real hardware (S7-1200, signal modules HMI...) we will be using for the project with a general description of how it works, architecture types, and also for the simulated hardware (tank, both valves, sensor ...)

§ **Chapter IV:** The last chapter absolutely dedicated for the realization, testing and developing the program that is meant to reach the point of the study.

Chaptre I :

Introduction to Automation

:

I.1 Industrial automation and control

I.1.1 Introduction

Industrial automation is a set of technologies that uses control systems and devices such as computer software and robotics to enable automatic operation of industrial processes and machinery without the need for human operators.

Industries implement automation to increase productivity, eliminate the possibility of human error, reduce costs, save time and achieve higher performances. A wide range of tools are required for industrial automation they include various control systems that incorporate different devices and systems impacting aspects of the manufacturing process, these automation devices include PLC's, HMI’s, SCADA and more.

This chapter aims to cover the fundamentals of automation, its historical development, principles, and applications in manufacturing and service industries using one of the most known controller or PID controller.

I.1.2 INTRODUCTION TO AUTOMATION

When talking about Service Automation, we should first talk about what led us to the idea of automating services. Automation isn’t a new concept and it dates back centuries to when humans invented basic automating tools like pulleys to make working easier. Automation is the use of various control systems for operating equipment such as machinery, processes in factories, steering of vehicles, aircraft, ships, and other applications.

Automation technology has matured to a point where a number of other technologies have emerged from it. For example, service automation and artificial intelligence. Advanced robotics is a specialized branch of automation in which automated machine possess certain levels of artificial intelligence in order to enable increased workforce capabilities or task functions.

Put simply, industrial robots and automated machinery are typically used to replace human workers in factory operations.

Automation provides benefits to almost every industry. Some examples are listed:

- Manufacturing, including food and pharmaceutical, chemical and petroleum, pulp and paper
- Transportation, including automotive, aerospace, and rail
- Utilities, including water and wastewater, oil and gas, electric power, and telecommunications
- Military and civil defense

- Facility operations, including security, environmental control, energy management, safety, and other construction automation.
- And many others. [1]

I.1.3 WHAT IS AUTOMATION?

Automation is the technology by which a process or procedure is performed with minimal human interference through the use of technological or mechanical devices. It is the technique of making a process or a system operate automatically. Automation crosses all functions within almost every industry from installation, maintenance, manufacturing, marketing, sales, medicine, design, procurement, management, etc. Automation has revolutionized those areas in which it has been introduced, and there is scarcely an aspect of modern life that has been unaffected by it.

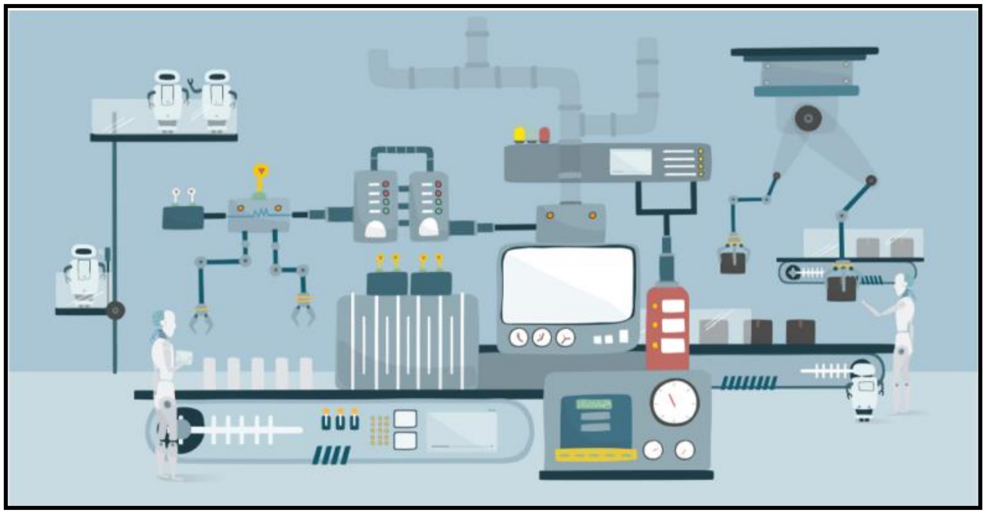

Figure I.1 Automation Processes.

Automation can be achieved by several different means including mechanical, electrical, electronic devices and computers, and hydraulic systems. Industries that involve very complicated systems such as modern factories, aircrafts and ships often use a combination of different automation techniques in order to maximise their process efficiencies.

Automation is a broad range of technologies that includes robotics and expert systems, telemetry and communications, electro-optics, Cybersecurity, process measurement and control, sensors, wireless applications, systems integration, test measurement, and more. Industrial automation in manufacturing entails the use of machines to carry out manufacturing processes with levels of speed, stamina, accuracy, and consistency with capabilities of that which is beyond the human worker.

The main benefits of automating processes in manufacturing include reduced production cost, reduced waste, improved quality and reliability, and drastically reduced workplace-related accidents.[1]

I.1.4 THE HISTORY OF AUTOMATION

Nowadays many associate the term ‘automation’ with advanced technologies such as artificial intelligence (AI), machine learning, and robotics. They aren’t wrong. However, the history of automation technology goes back centuries and is much deeper than just these versions of automation that appeared in the last two or three decades. The idea of automation isn’t necessarily a modern one as the theory behind utilising automation technology dates back to hundreds of years ago but has only recently begun to become more task-specific and refined to fit certain industries.

The earliest known mention of automation can be credited to Homer’s “The Iliad”. Towards the end of the first book, Homer presents the tale of Hephaestus. Representing the Greek god of blacksmiths, craftsmen, metals, fire, volcanos, artisans and sculptors, Hephaestus was tasked with manufacturing all of the weaponry needed for the gods of Mount Olympus. To help himself with this task, he created automatons. These automatons were self-operating machines forged from metal. They helped Hephaestus with his task and made it possible for him to complete the required equipment for the gods. Although this tale is likely just a tale, its ancient mention proves that the idea of automation has been around for a long time.

For many centuries there has been evidence of different civilisations attempting to use forms of automation to solve everyday problems they faced. However, automation didn’t really become popular until the Industrial Revolution. After this period, there was a massive demand for basic things like cotton, textiles, paper, household items which meant there needed to be drastic changes in the production process to enable to mass-production of items. With an immense amount of emphasis placed on extreme efficiency and production, innovations in automation allowed for the mechanised production of textiles and other things.

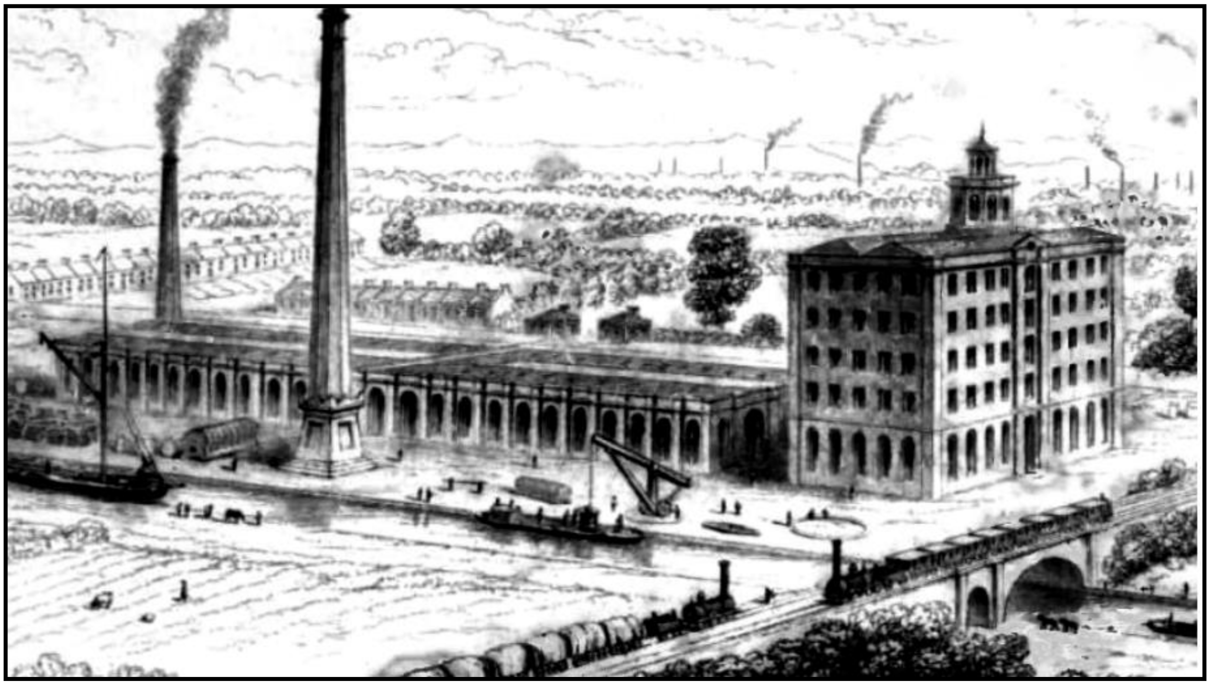

Figure II.2 The Bridgewater Foundry. [1]

In 1806, the first paper-making machine was invented. Before this invention, industries could not satisfy the demands of developing modern society for large quantities of printing and writing material. This paper-making machine allowed for an industrialised version of the historical process of hand paper-making. This industrial revolution also saw other significant innovations such as the development of the transportation and communication industry, leading to the increased globalisation of companies, products, and immigration of workforces.

In 1913, Henry Ford installed the first moving assembly line for the mass production of an entire car. This innovation reduced the time it took to build a car from more than twelve hours to less than three. Ford was already selling relatively inexpensive cars but he wanted to reduce the cost even further. To do this, he has to completely change the production process in order to maximise efficiency. While producing the Model T automobile, the streamlining process grew more sophisticated and in order to make it easier, he broke the production process into 84 steps. In addition to this, they trained each worker to specialise in just one step. Ford's assembly line was inspired by the continuous-flow production methods used by flour mills, breweries, canneries and industrial bakeries.

Overall, automation has a rich history that spans over many centuries. The main uses of automation prior to the 20th and 21st century have been in industrial fields, and have only more recently been integrated into the IT industry. The drivers of all automation technology, however, have always been similar. With industrial automation, the goal was always clear: to improve the efficiency of manufacturing a variety of items.

With IT automation, the goal is to improve efficiency by creating a process that is self-sufficient and replaces an IT worker's manual labor in data centers and cloud deployments. The parallels are clear, and they show why automation will always be prevalent in society.

Innovating things to reduce the burden on humans, increase business productivity and make our lives easier in terms of reducing manual labor is something that has been important to us for centuries, and will continue to make its impact in the future. [1]

I.2 Industrial Automation

Industrial automation is the use of various control devices like PC's/PLC's/DCS, used to have control on various operations of an industry without significant intervention from humans and to provide automatic control performance. In industries, control strategies use a set of technologies which are implemented to get the desired performance or output, making the automation system most essential for industries.

Figure III.3 Automated Process in industries. [2]

Industrial automation involves usage of advanced control strategies like cascade controls, modern control hardware devices as PLC's,sensors and other instruments for sensing the control variables, signal conditioning equipments to connect the signals to the control devices, drives and other significant final control devices, standalone computing systems, communication systems, alarming and HMI (Human Machine Interface) systems.

I.2.1 Need of Automated Industry [2]

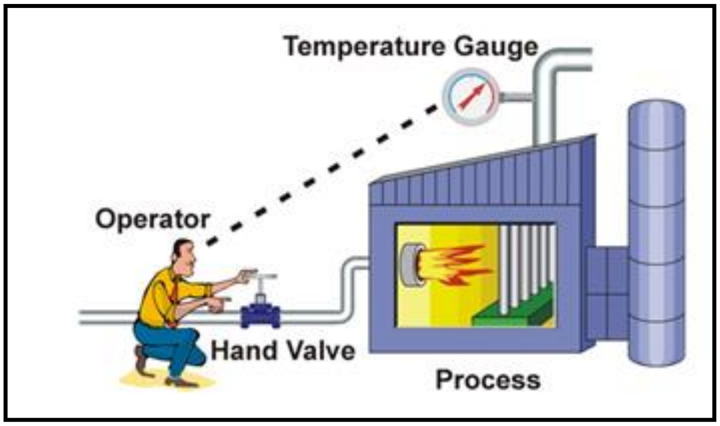

Figure IV.4 Manual Control. [2]

- **To reduce Periodic or Manual checking**

In some critical applications periodic checking of the process variable is necessary to perform industrial operations. Automation equipment reduces the periodic or manual operations and establishes the automatic working conditions.

- **To increase the Productivity**

Automating the manufacturing and other production processes increases the production rate by producing output at greater amounts for a given labour input

- **Reduce the Production Cost**

Using the automatic machines and equipments, human intervention to control the processes abruptly falls. This reduces the investment on the labor cost hence the production cost.

- **To improve Product Quality**

Continuously doing the same work may not be perfect in all the cases in terms of quality specifications with human efforts. With automation equipment ,one can get reliable and uniform product quality by using real time hardware control devices.

- **To increase the Flexibility**

Using the automation equipment various , process are handled simply without getting any complex environment particularly in manufacturing processes.

- **Operator Friendly and Improves the Safety**

Complexity of operating the equipments or processes is reduced with industrial automation. It changes the position of the operator as operator to the supervisory role.

I.2.2 Types of Industrial Automation Systems [2]

- **Fixed Automation**

In fixed automation, the sequence of processing operations is set by the equipment parameters. Each of the operation in a fixed or hard automation sequence is usually simple; it is the combination and coordination of many operations into one piece of equipment that makes the system more complicated. This type of automation is characterized by high initial investment cost and high production rates. It is, therefore, suitable for products with very high demand and volumes. Machine transfer lines, automatic assembly machines, and certain chemical processes instruments are examples of fixed automation.

- **Programmable Automation**

The production equipment is designed to be able to modify the sequence of operations to the different product configurations in this automation. The sequence of operation is controlled by a programming, which is a set of coded instructions allowing the system to read and interpret them. This automation is particularly appropriate for batch production process where production volume is medium to high. It is hard to change and recognize the system for a new product or sequence of operations. Numerically controlled machines, steel rolling mills, paper mills, and industrial robots are the examples of programmable automation.

- **Flexible Automation**

A flexible or soft automated system is a system that is capable of producing a wide range of products with essentially no time for changes from one product to another. It is a fully programmable automation. There is no loss of production time when reprogramming the automation system and changing the physical parameter of the product. As a result, the system can produce different combinations and schedules of products instead of requiring them to be manufactured in separate batches. Examples of this automation system are self-guided vehicles, automobiles and CNC machines.

I.2.3 Structure of Industrial Automation Or The Automation pyramid [3]

The automation pyramid is a pictorial example of the different **levels** of automation in a factory, It also serves as a visual example of how technology is being integrated into industry.

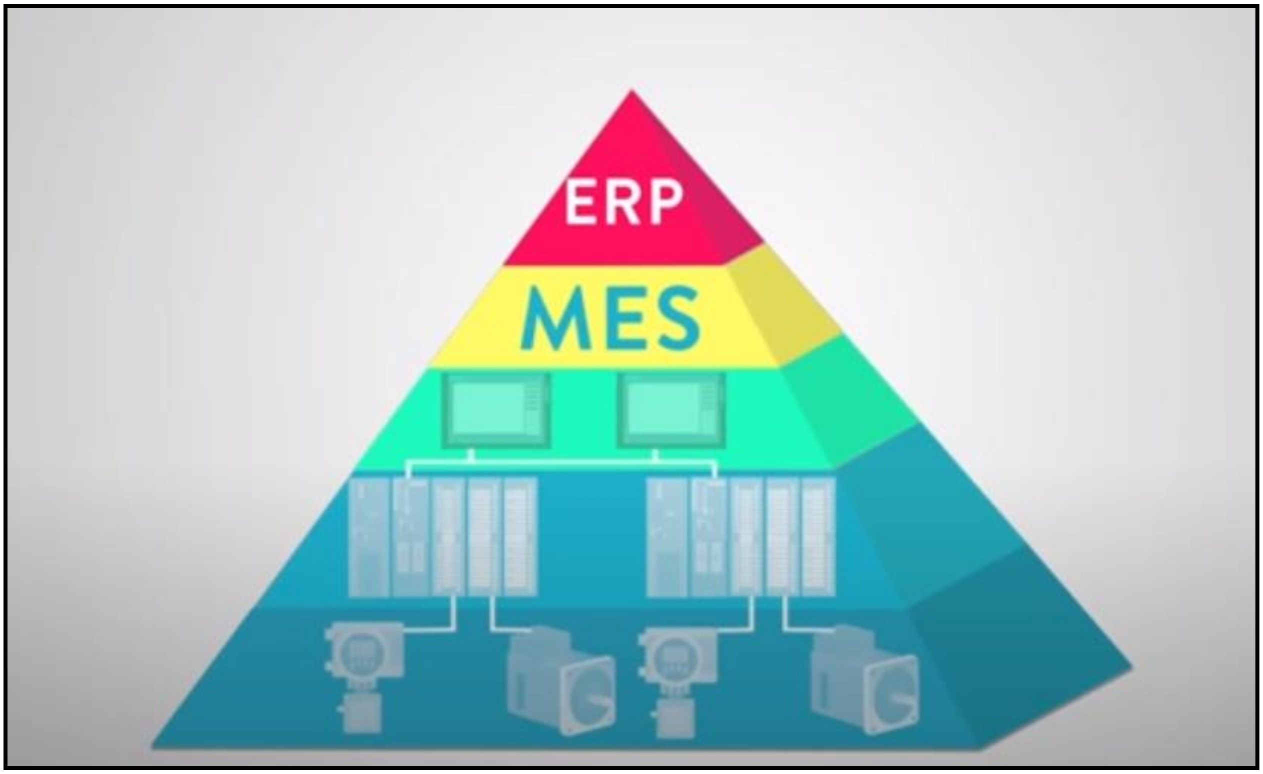

Figure V.5 Automation Pyramid. [3]

I.2.3.1 Field Level

Beginning on the bottom of the pyramid is what we will refer to as the "field" level. These are the devices, actuators, and sensors that you see in the field or on the production floor.

If you think of it this way, the field level is the production floor that does the physical work and monitoring. Electric motors, hydraulic and pneumatic actuators to move machinery, proximity switches used to detect that movement or certain materials, photoelectric switches that detect similar things will all play a part in the field level. [3]

I.2.3.2 The Control Level

The next level is referred to as the control level. This is where the PLC’s and PID’s come in to play.

The control level uses these devices to control and “run” the devices in the field level that actually do the physical work. They take in information from all of the sensors, switches, and other input devices to make decisions on what outputs to turn on to complete the programmed task.

A PID is usually integrated in to the PLC and stands for proportional–integral–derivative. That is what can keep a variable within a set of parameters.

A common industrial PID controlled item is a heater. Many systems in manufacturing plants have to be heated. We control this with a PID block within the PLC. When a set point is entered, the PID will determine when the PLC needs to turn the heater on and off to maintain a constant temperature. [3]

I.2.3.3 Supervisory Level

The third level of the automation pyramid is known as the supervisory level. Where the previous level utilizes PLCs, this level utilizes SCADA. SCADA is short for supervisory control and data acquisition.

SCADA is essentially the combination of the previous levels used to access data and control systems from a single location.

Plus it usually adds a graphical user interface, or an HMI, to control functions remotely. Water plants will often employ this technology to control remote water pumps in their systems.

The important thing to remember about SCADA is that it can monitor and control multiple systems from a single location. It isn’t limited to a single machine like HMI’s that we have discussed in previous posts. [3]

I.2.3.4 The Planning Level

The fourth level of the automation pyramid is called the planning level. This level utilizes a computer management system known as MES or manufacturing execution system.

MES monitors the entire manufacturing process in a plant or factory from the raw materials to the finished product.

This allows management to see exactly what is happening and allows them to make decisions based on that information. They can adjust raw material orders or shipment plans based on real data received from the systems we talked about earlier. [3]

I.2.3.5 The Management Level

The top of the pyramid is what is called the management level. This level uses the companies integrated management system which is known as the ERP or enterprise resource planning.

This is where a company's top management can see and control their operations. ERP is usually a suite of different computer applications that can see everything going on inside a company. It utilizes all of the previous levels technology plus some more software to accomplish this level of integration.

This allows the business to be able to monitor all levels of the business from manufacturing, to sales, to purchasing, to finance and payroll, plus many others. The integration of the ERP promotes efficiency and transparency within a company by keeping everyone in the same page. [3]

I.3 PID Controller

A PID controller is an instrument used in industrial control applications to regulate temperature, flow, pressure, speed and other process variables. PID (proportional integral derivative) controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable controller.

PID control is a well-established way of driving a system towards a target position or level. It's a practically ubiquitous as a means of controlling temperature and finds application in myriad chemical and scientific processes as well as automation. PID control uses closed-loop control feedback to keep the actual output from a process as close to the target or setpoint output as possible. [4]

I.3.1 History of PID Controller

The first evolution of the PID controller was developed in 1911 by Elmer Sperry. However, it wasn't until 1933 that the Taylor Instrumental Company (TIC) introduced the first pneumatic controller with a fully tunable proportional controller. A few years later, control engineers went eliminate the steady state error found in proportional controllers by resetting the point to some artificial value as long as the error wasn't zero. This resetting "integrated" the error and became known as the proportional-Integral controller. Then, in 1940, TIC developed the first PID pneumatic controller with a derivative action, which reduced overshooting issues. However, it wasn't until 1942, when Ziegler and Nichols tuning rules were introduced that engineers were able to find and set the appropriate parameters of PID controllers. By the mid-1950's, automatic PID controllers were widely adopted for industrial use. [4]

I.3.2 PID Controller Working Principle [5]

The working principle behind a PID controller is that the proportional, integral and derivative terms must be individually adjusted or "tuned." Based on the difference between these values a correction factor is calculated and applied to the input. For example, if an oven is cooler than required, the heat will be increased. Here are the three steps:

- **Proportional tuning**

Involves correcting a target proportional to the difference. Thus, the target value is never achieved because as the difference approaches zero, so too does the applied correction.

- **Integral tuning**

Attempts to remedy this by effectively cumulating the error result from the "P" action to increase the correction factor. For example, if the oven remained below temperature, “I” would act to increase the head delivered. However, rather than stop heating when the target is reached, "I" attempts to drive the cumulative error to zero, resulting in an overshoot.

- **Derivative tuning**

Attempts to minimize this overshoot by slowing the correction factor applied as the target is approached.

I.3.3 Types of PID Controller [6]

There are three basic types of controllers: on-off, proportional and PID. Depending upon the system to be controlled, the operator will be able to use one type or another to control the process.

- **On/Off Control**

An on-off pid controller is the simplest form of temperature control device. The output from the device is either on or off, with no middle state. An on-off controller will switch the output only when the temperature crosses the setpoint. One special type of on-off control is a limit controller. This controller uses a latching relay, which must be manually reset, and is used to shut down a process when a certain temperature is reached.

- **Proportional Control**

Proportional controls are designed to eliminate the cycling associated with on-off control. A proportional controller decreases the average power supplied to the heater as the temperature approaches setpoint. This has the effect of slowing down the heater so that it will not overshoot the setpoint but will approach the setpoint and maintain a stable temperature. This proportioning action can be accomplished by turning the output on and off for short time intervals. This "time proportioning" varies the ratio of "on" time to "off" time to control the temperature.

- **Standard PID Controller**

This standard PID controller combines proportional control with integral and derivative control (PID), which helps the unit automatically compensate for changes in the system. These adjustments, integral and derivative, are expressed in time-based units; they are also referred to by their reciprocals, RESET and RATE, respectively. The proportional, integral and derivative terms must be individually adjusted or "tuned" to a particular system using trial and error. PID controllers provide the most accurate and stable control of the three controller types.

I.3.4 PID Control Loop

A PID process loop controller is designed to generate an output that causes some corrective effort to be applied to a process so as to drive a measurable process variable towards the desired set-point value.

The controller uses an “actuator” to affect the process and a “sensor” to measure the results.

Often automation technicians and programmers are required to become familiar with configuring and tuning a PID loop control instruction and it can be one of most overwhelming when it comes to PLC programming topics. [7]

I.3.5 PID Control Loop Application or “Car cruise control” [7]

- **Proportional Control Loop**

the “P” or proportional is described as in the farther you are from the desired speed, the more you press the gas pedal and on the other hand, the closer you are, the less you press on it.

This works well but when you get at the desired speed, based on this rule you would let off the gas completely. And the end result is your car slows down and stays a little below the desired speed.

Proportional control is the main ingredient of any control but maybe a little inaccurate.

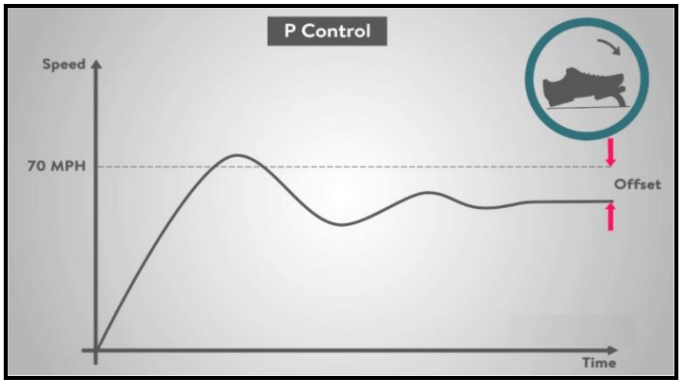

Figure VI.5 Proportional Controller

- **Proportional-Integral Control Loop**

For "I "or integral, you wait for a little, and if there is no improvement you push a little more on the pedal.

If you are stuck below the desired speed for a long time without progress, you push the gas pedal a little further. If you still do not make it to the desired speed for some time, you again push the pedal a little further down.

Once you get to the desired speed you leave the pedal where it is. Integral control gives you accuracy but you have to wait.

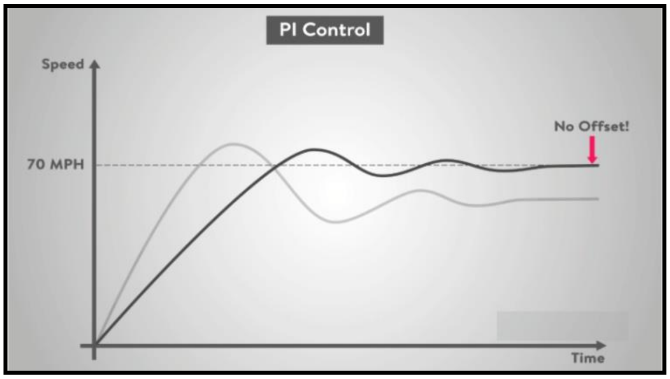

Figure VII.5 Proportional-Integral Controller

I.3.6 PID Control Loop

For "D" or derivative, you react to sudden changes. Let's say a strong wind gust pushes your car. Suddenly your speed surges fast upward toward the desired speed. You become startled so you release the gas pedal.

As the speed surge ends and the speed stabilizes, you will then return the pedal to where it was.

Derivative control manages sudden surges and may prevent overshooting your target speed.

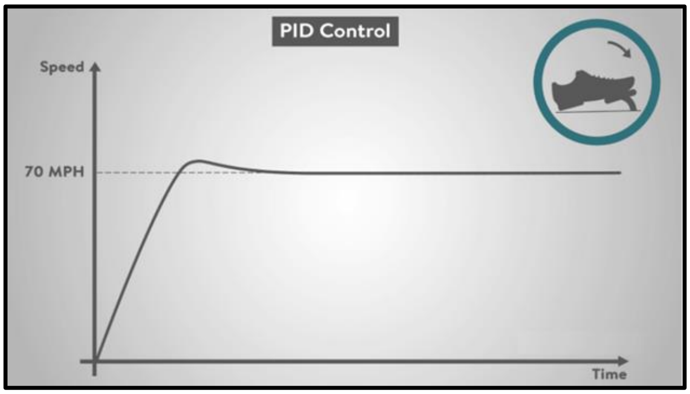

Figure VIII.6 Proportional-Integral-Derivative Controller

I.4 Conclusion

We have presented in this chapter the fundamentals of automation and we have back in time to explore the historical development of automation and to know why industry automation was so important for the modern revolution industry.

Finally we have explore one of the most ubiquitous controller in automation since was created and its types and how could we use it in different applications.

Chapter II :

Tia Portal V16 & Factory I/O v2.4.2

.

II.1 Introduction

In this chapter we will present a global description on our software that we will use to realize and simulate our process level control.

We will be talk about its valuable role and how could allows us to understand how industrial processes can be automated and control.

II.2 Description of Totally Integrated Automation Portal

II.2.1 TIA Portal basics [8]

II.2.1.1 TIA Portal overview

- Introduction

The Totally Integrated Automation Portal (TIA portal) integrates various SIMATIC products in a software application with which you can increase your productivity and efficiency. The TIA products work together within the TIA portal and support you in all areas required for the creation of an automation solution. A typical automation solution en compasses:

- ✓ A PLC that controls the process with the aid of the program.
- ✓ An HMI device with which you operate and visualize the process.

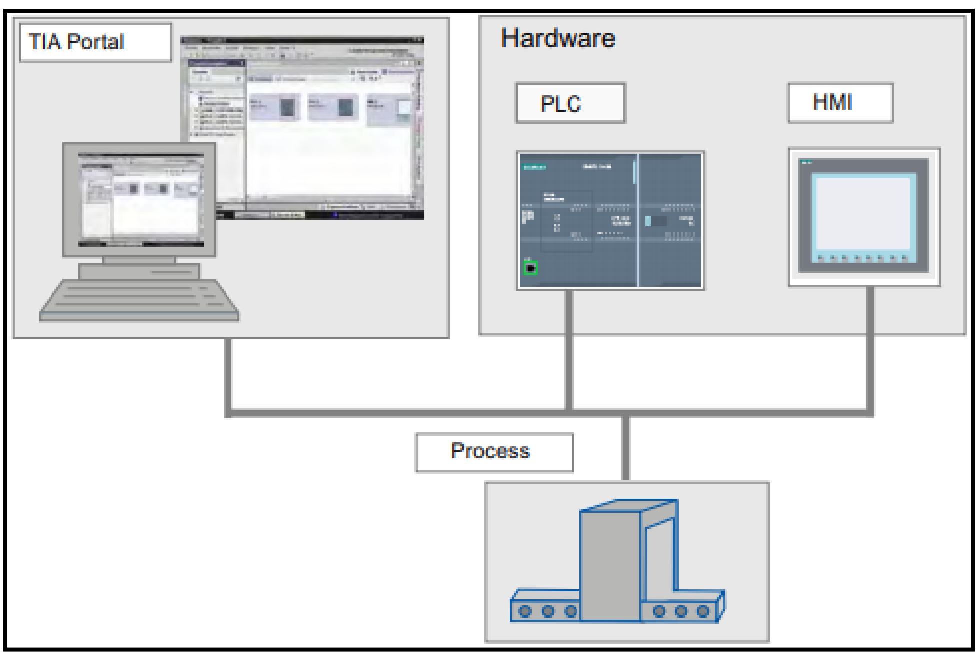

Figure II.1 Automate and supervise a process

- Tasks

The TIA portal supports you in creating an automation solution. The most important configuration steps are:

- ✓ Creating the project.
- ✓ Configuring the hardware.
- ✓ Networking the devices.
- ✓ Programming the PLC.
- ✓ Configuring the visualization.
- ✓ Loading the configuration data.
- ✓ Using the online and diagnostic functions.

- Benefits

The TIA portal offers the following advantages:

- ✓ Common data management.
- ✓ Easy handling of programs, configuration data and visualization data.
- ✓ Easy editing using drag-and-drop operation.
- ✓ Easy loading of data to the devices.
- ✓ Easy operation.
- ✓ Graphic supported configured and diagnostics.

II.2.1.2 Engineering concept

- Engineering system

You can use the TIA Portal to configure both the PLC and the visualization in a uniform engineering system. All data are stored in one project. The components for programming (STEP 7) and visualization (Wincc) are not separate programs, but rather editors of a system that accesses a common data base. All data are stored in a common project file.

You use a common user interface for all tasks to access all programming and visualization functions at all times.

II.2.1.3 Data management

- Central data management

In the TIA Portal, all data are stored in one project. Modified application data, such as tags, are automatically updated within the complete project even across several devices.

- Symbolic addressing across project parts

If you use a process tag in several blocks of various PLCs and in HMI screens, the tag can be created or modified from any place in the program. In this case it is of no importance in which block of which device you make the modification.

The TIA Portal offers the following options for defining PLC tags:

- ✓ ● Definition in the PLC tag table
- ✓ ● Definition in the program editor
- ✓ ● Definition by means of a link with the inputs and outputs of the PLC

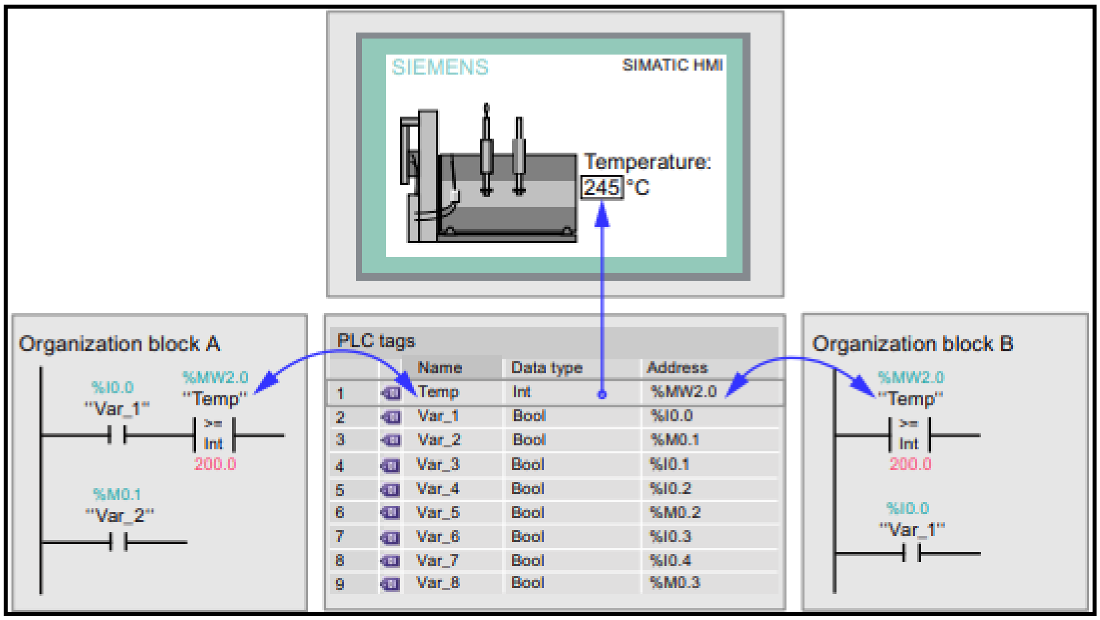

Figure II.2 Data Management

All defined PLC tags are listed in the PLC tag table and can be edited there. Modifications are performed centrally and updated continuously. The consistent data management eliminates the need for synchronization between the project participants within a project, for example, between the programmer and the HMI designer.

- Library concept

Project parts can be used again via the library both within the project and also in other projects.

- ✓ Elements such as blocks, PLC tags, PLC tag tables, interrupts, HMI screens, individual modules or complete stations can be stored both in local and in global libraries.
- ✓ Devices and defined functions can be reused.
- ✓ The global library allows for an easy exchange of data between projects.

II.2.2 Views in the TIA portal [9]

II.2.2.1 Layout of the user interface

- Views

  Three different views are available for your automation project:

  - ✓ The portal view is a task-oriented view of the project tasks.
  - ✓ The project view is a view of the components of the project, as well as the relevant work areas and editors.
  - ✓ The library view shows the elements of the project library and the open global libraries.

  You can change over between the two views using a link

1- Portal view

The portal view provides you with a task-oriented view of the tools. Here, you can quickly decide what you want to do and call up the tool for the task in hand. If necessary, the view changes automatically to the project view for the selected task.

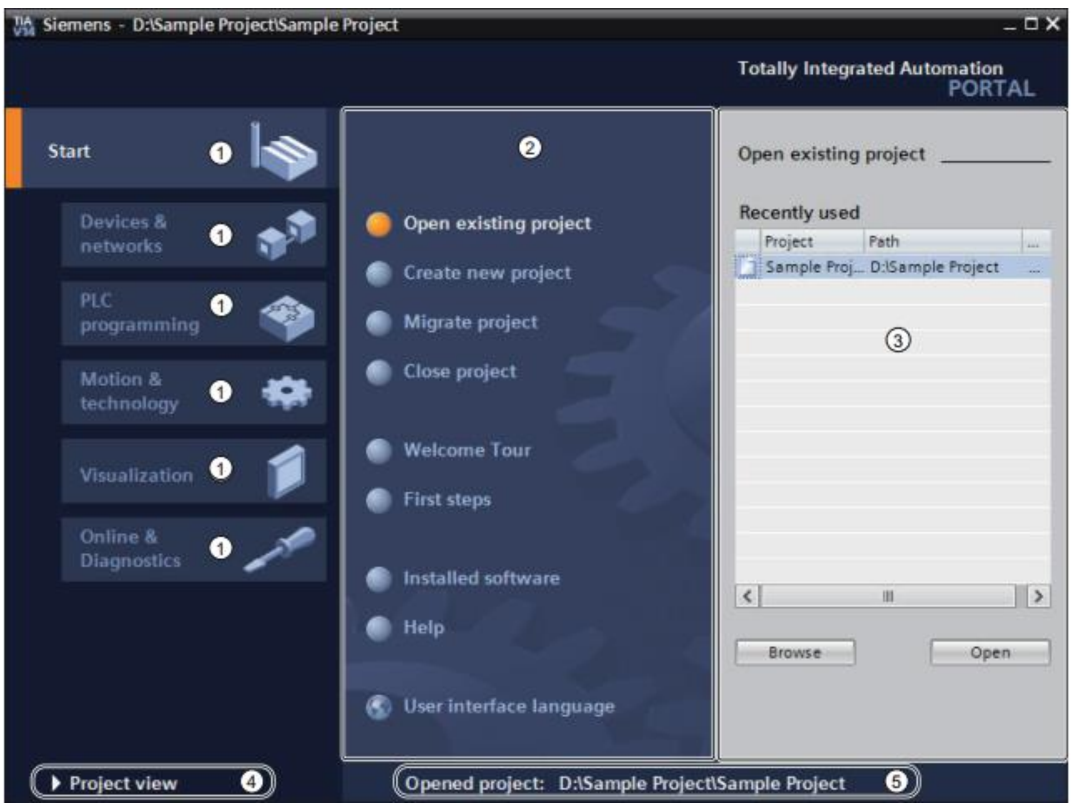

Figure III.3 Layout of the portal view

① Portals for different tasks
② Actions for the selected portal
③ Selection panel for the selected action
④ Switch to project view
⑤ Display of the project that is currently open

Portals

The portals provide the basic functions for the individual task areas. The portals that are provided in the portal view depends on the products that have been installed.

Actions for the selected portal

Here, you will find the actions available to you in the portal you have selected. You can call up the help function in every portal on a context-sensitive basis

Selection panel for the selected action

The selection panel is available in all portals. The content of the panel adapts to your current selection.

Switch to project view

You can use the "Project view" link to change to the project view

2- Project view

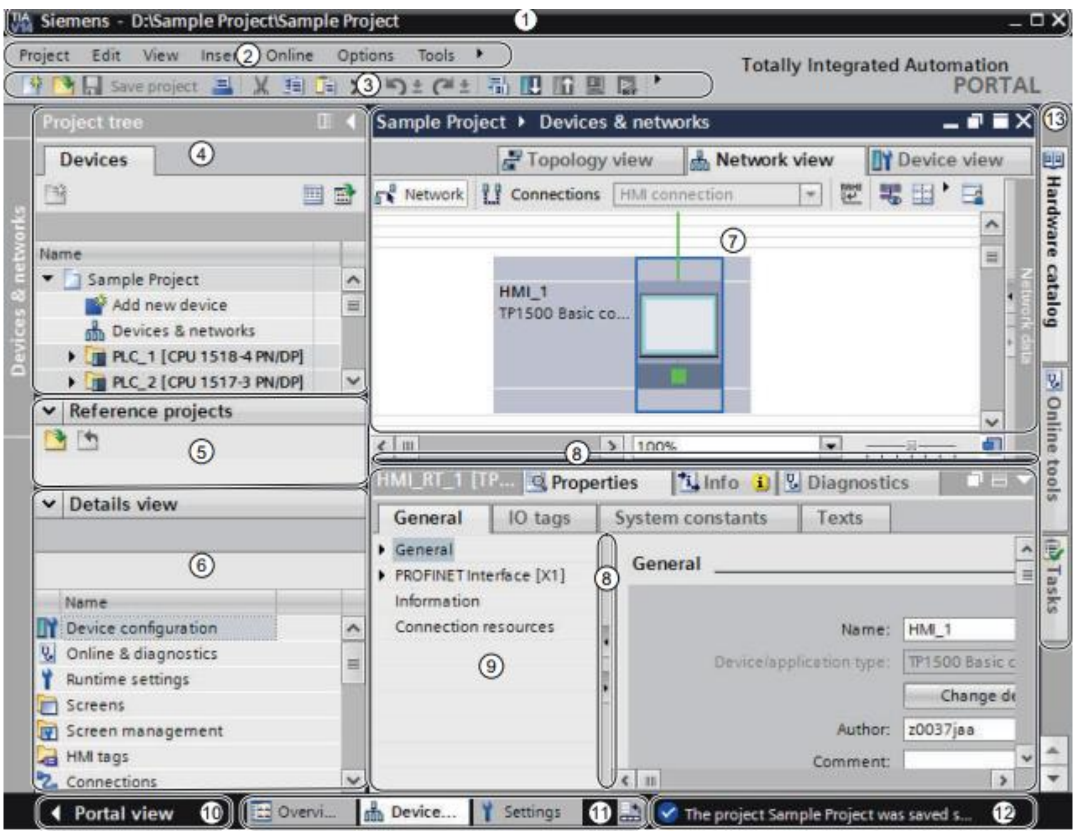

Figure IV Layout of the project vie

① Title bar
② Menu bar
③ Toolbar
④ Project tree
⑤ Reference projects
⑥ Details view
⑦ Work area
⑧ Dividers
⑨ Inspector window
⑩ Changing to the portal view
⑪ Editor bar
⑫ Status bar with progress display
⑬ Task cards

Title bar
The name of the project is displayed in the title bar.

Menu bar
The menu bar contains all the commands that you require for your work.

Toolbar
The toolbar provides you with buttons for commands you will use frequently. This gives you faster access to these commands.

Dividers
Dividers separate individual components of the program interface. The arrows on the dividers allow you to display and hide the adjacent sections of the user interface.

Changing to the portal view
You can use the "Portal view" link to change to the portal view.

Editor bar
The Editor bar displays the open editors. This allows you to quickly switch between open elements. If you have opened numerous editors, you can display the editors of the same type as a group.

Status bar with progress display

In the status bar, you will find the progress display for processes that are currently running in the background. This also includes a progress bar that shows the progress graphically. Hover the mouse pointer over the progress bar to display a tooltip providing additional information on the active background process. You can cancel the background processes by clicking the button next to the progress bar.
If no background processes are currently running, the status bar displays the last generated alarm.

3- Library view

The library view provides an overview of the elements in the project library and the open global libraries. You can switch to the library view using the "Libraries" task card.

II.2.2.2 Project tree

The library view provides an overview of the elements in the project library and the open global libraries. You can switch to the library view using the "Libraries" task card.

Function and structure of the project tree

Using the project tree features gives you access to all components and project data. You can perform the following tasks in the project tree:

- ✓ Add new components.
- ✓ Edit existing components.
- ✓ Scan and modify the properties of existing components.

You can select the objects of the project tree either with the mouse or via the keyboard by typing the first letter of the desired object. If more than one object begins with the same letter, the next lower object is selected. The project tree must be the focused user interface element in order for you to select an object with its initial letter.

① Title bar
② Toolbar
③ Table header
④ Project
⑤ Devices
⑥ Ungrouped devices
⑦ Security settings
⑧ Cross-device functions
⑨ Unassigned devices
⑩ Common data
⑪ Documentation settings
⑫ Languages & resources
⑬ Online access
⑭ Card Reader / USB memory

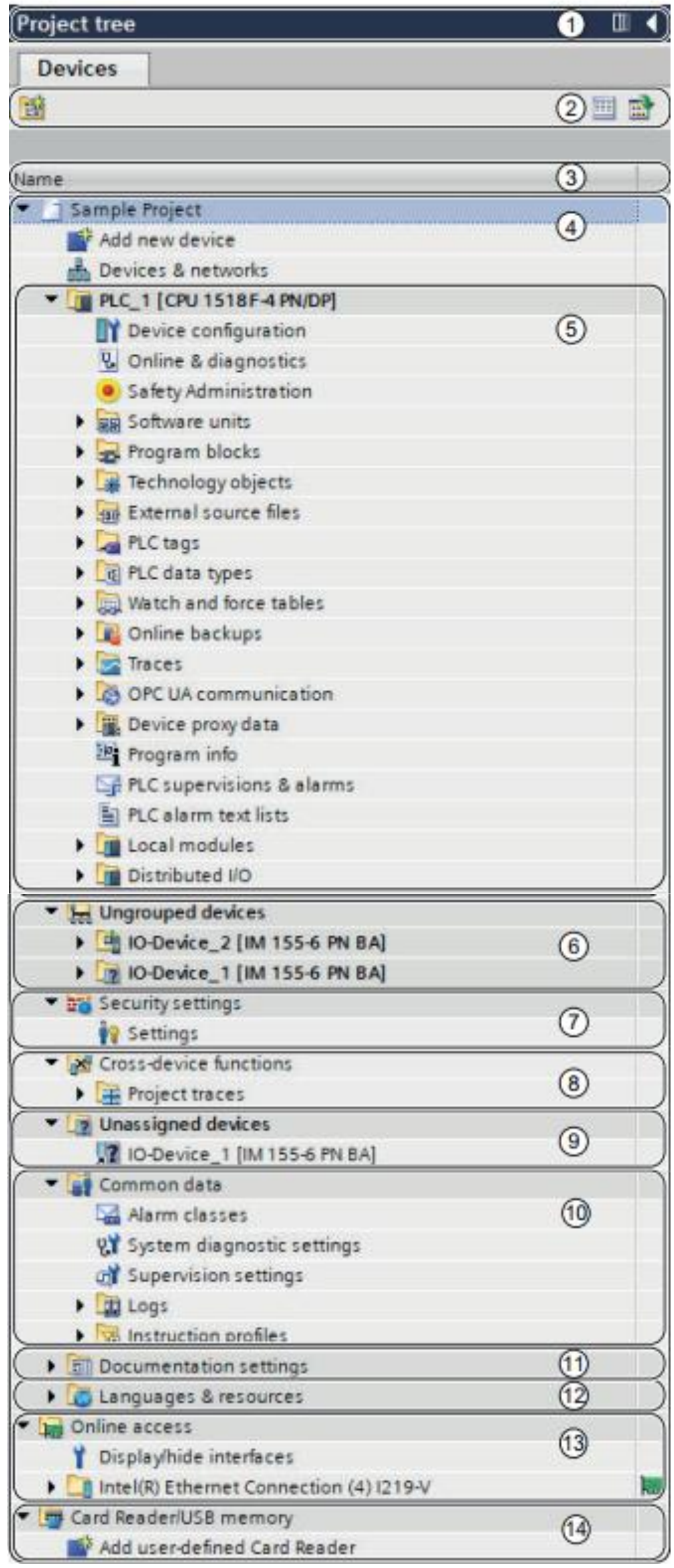

Figure II.5 Layout of project tree.

II.2.3 General Programming [10]

II.2.3.1 Operating system and user program

SIMATIC controllers consist of operating system and user program.

- The operating system organizes all functions and sequences of the controller that are not connected with a specific control task (e.g. handling of restart, updating of process image, calling the user program, error handling, memory management, etc.). The operating system is an integral part of the controller.
- The user program includes all blocks that are required for the processing of your specific automation task. The user program is programmed with program blocks and loaded onto the controller.

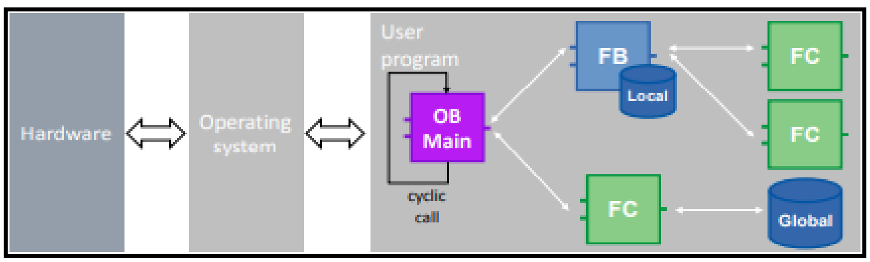

Figure II.6 Operating system and user program.

For SIMATIC controllers the user program is always executed cyclically. The “Main” cycle OP already exists in the “Program blocks” folder after a controller was created in STEP 7. The block is processed by the controller and recalled in an infinite loop.

II.2.3.2 Description Of Programming Blocks

In STEP 7 (TIA Portal) there are all familiar block types from the previous STEP 7 versions:

- ✓ Organization blocks
- ✓ Function blocks
- ✓ Functions
- ✓ Data blocks

- **Organization blocks (OB)**

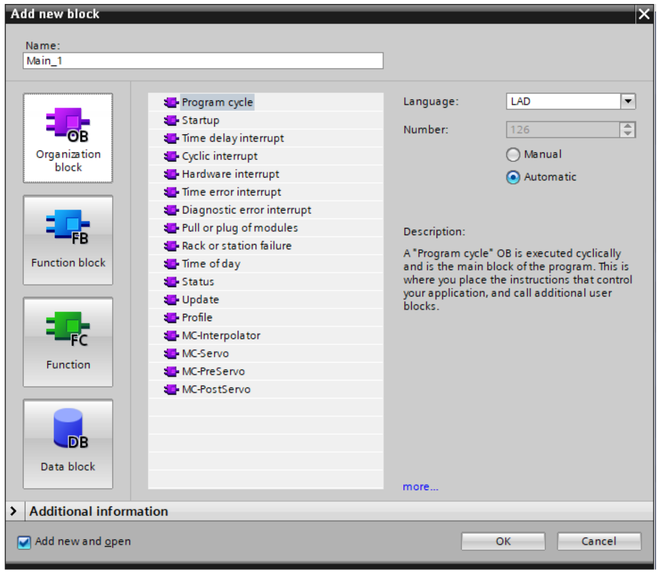

Figure II.7 “Add new block” dialog (OB)

OBs are the interface between the operating system and the user program. They are called by the operating system and control, for example, the following processes:

- ✓ Startup behavior of the controller
- ✓ Cyclic program processing
- ✓ Interrupt-controlled program processing
- ✓ Error handling

- **Functions (FC)**

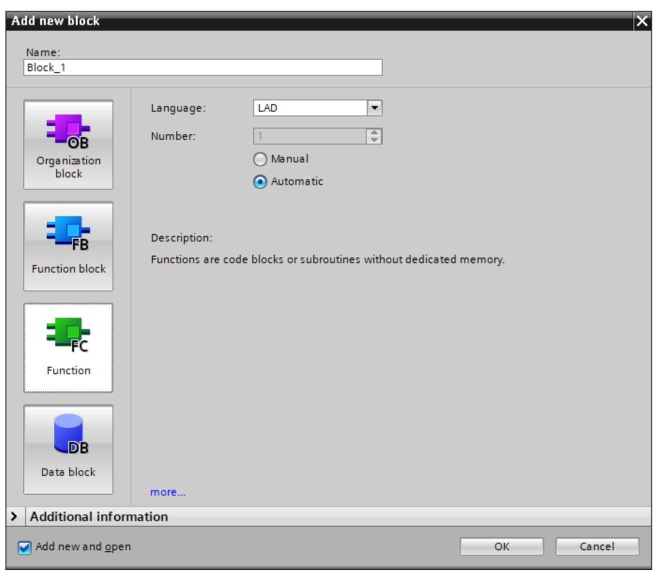

Figure II.8 “Add new block” dialog (FC)

FCs are blocks without cyclic data storages. This is why the values of block parameters cannot be saved until the next call and have to be provided with actual parameters when called, in other word **Functions are code blocks or subroutines without dedicated memory.**

- **Function blocks (FB)**

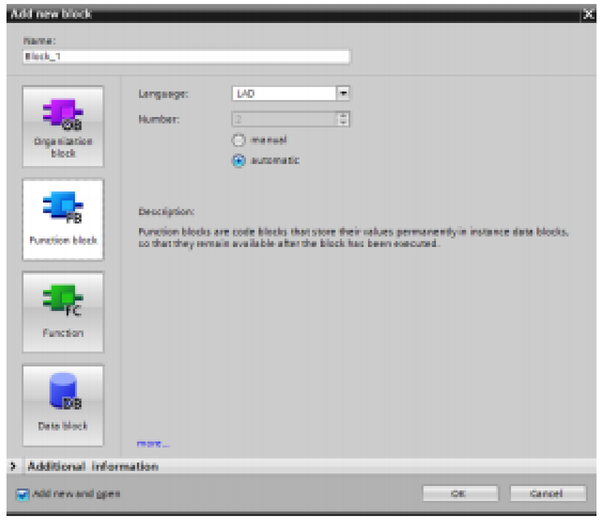

Figure II.9 “Add new block” dialog (FB)

FBs are blocks with cyclic data storage, in which values are permanently stored. The cyclic data storage is realized in an instance DB so that they remain available after the block has been executed.

- **Data Block**

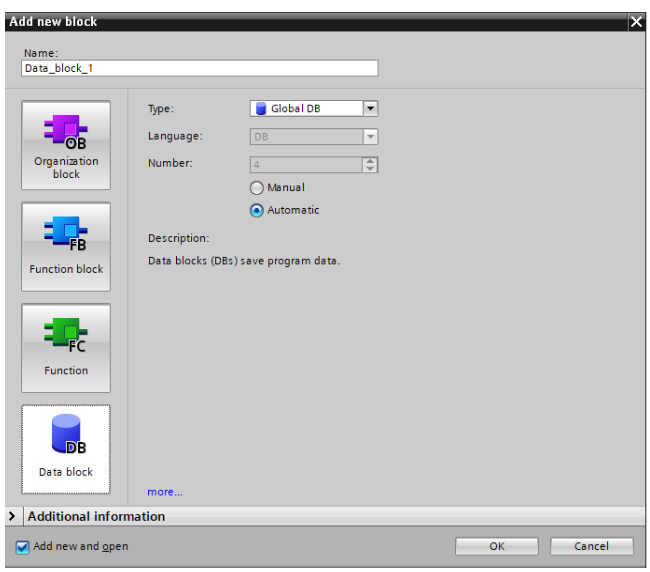

Figure II.10 “Add new block” dialog (DB)

Variable data is located in data blocks that are available to the entire user program.

II.2.4 Programming Languages [10]

For the programming of a user program, various different programming languages are available. Each language has its own advantages, which can be variably used, depending on the application. Every block in the user program can therefore be created in any programming language.

II.2.4.1 Ladder logic [11]

The graphic programming language Ladder Logic (LAD) is based on the representation of circuit diagrams. The elements of a circuit diagram, e.g. normally open contacts and normally closed contacts, are combined to form networks. The code section of a logic block represents one or more networks.

II.2.4.2 Function Block Diagram Programming Language [11]

The programming language Function Block Diagram (FBD) is based on graphic logic symbols also known in Boolean algebra. Complex functions such as math functions can also be displayed directly in combination with the logic boxes. The programming language FBD is supplied with the standard STEP 7 software package.

II.2.4.3 Statement List Programming Language [11]

The programming language STL is a text-based programming language with a structure similar to machine code. Each statement represents a program processing operation of the CPU. Multiple statements can be linked to form networks.

II.2.4.4 Structured Control Language [11]

The programming language SCL (Structured Control Language) is available as an optional package. This is a high-level text-based language whose global language definition conforms to IEC 1131-3. The language closely resembles PASCAL and, other than in STL, simplifies the programming of loops and conditional branches due to its high-level language commands, for example. SCL is therefore suitable for calculating equations, complex optimization algorithms, or the management of large data volume. S7 SCL programs are written in the source code editor.

II.2.4.5 S7-GRAPH (Sequential Control) [11]

The graphic programming language S7-GRAPH is available as optional package. It allows you to program sequential controls. This includes the creation of sequencers and the specification of corresponding step contents and transitions. You program the contents of the steps in a special programming language (similar to STL). Transitions are programmed in a Ladder Logic Editor (a light version of LAD).

II.3 Description Of Factory I/O Software

Factory I/O or 3D factory simulation is the next generation PLC training which allows us to create our own 3D scene and help us to get a clear idea of how industrial processes are automated.

II.3.1 3D factory simulation [12]

Factory I/O is a 3D factory simulation for learning automation technologies. Designed to be easy to use, it allows to quickly build a virtual factory using a selection of common industrial parts.

Factory I/O also includes many scenes inspired by typical industrial applications, ranging from beginner to advanced difficulty levels. The most common scenario is to use Factory I/O as a PLC training platform since PLC are the most common controllers found in industrial applications. However, it can also be used with microcontrollers, SoftPLC, Modbus, among many other technologies.

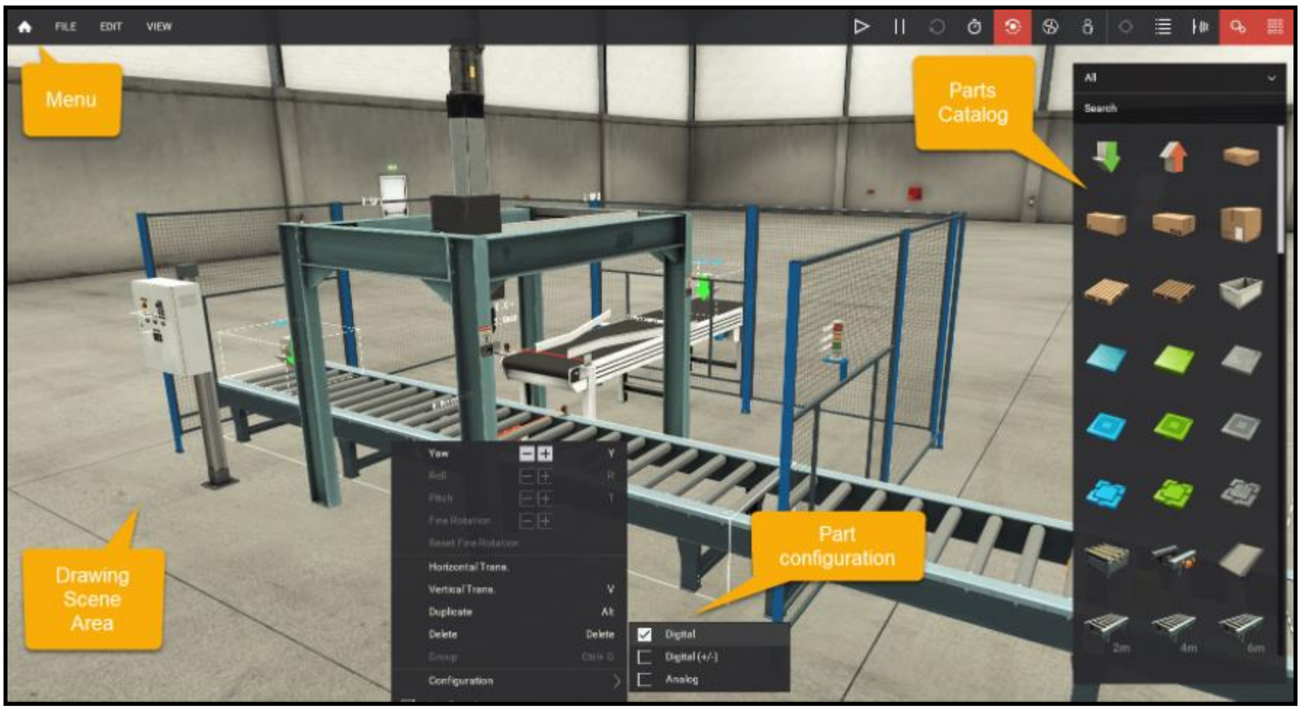

Figure II.11 Factory I/O Faceplate

II.3.2 Installing [12]

1. Run the installer. If prompted by the User Account Control, click **Yes**. Follow the installer instructions.

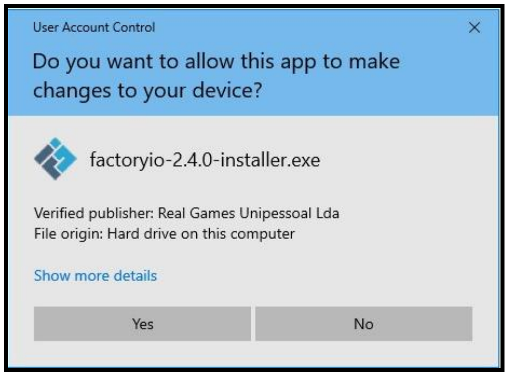

Figure II.12 installation window.

2. On the components window leave the Advantech drivers checked if you intend to use Factory I/O with Advantech USB 4750 and 4704 DAQ boards (requires the Ultimate Edition). Leave the OPC Core Components option checked if you are going to use Factory I/O with the OPC Client Data Access driver (requires either the Ultimate or Modbus & OPC Client DA Edition).

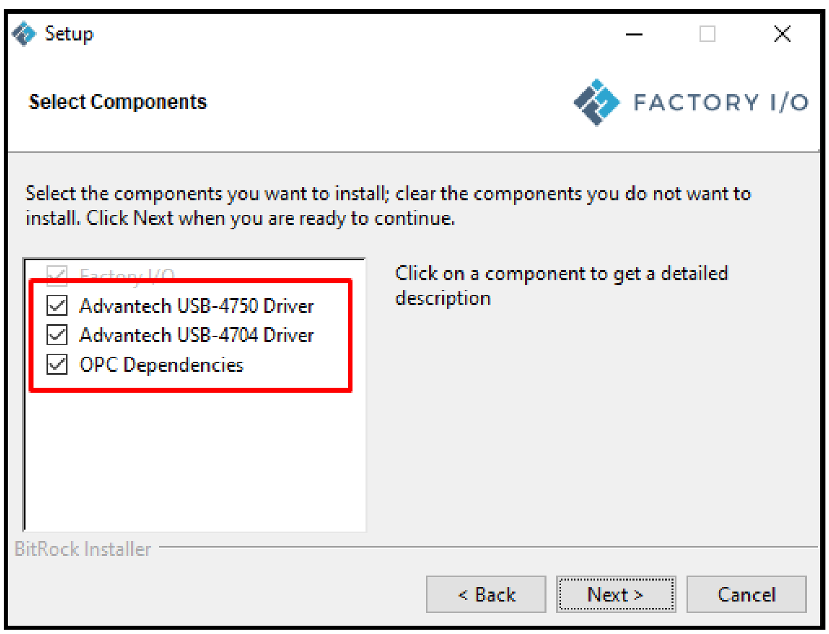

Figure II.13 Select Components

3. Now it's time to launch Factory I/O. If no license is found, Factory I/O will automatically start a 30-day full feature trial (requires internet connection). Click on File > Options >

Licensing, enter a serial key to activate a standalone license or enter a share code to checkout a floating license. Next, click on Activate.

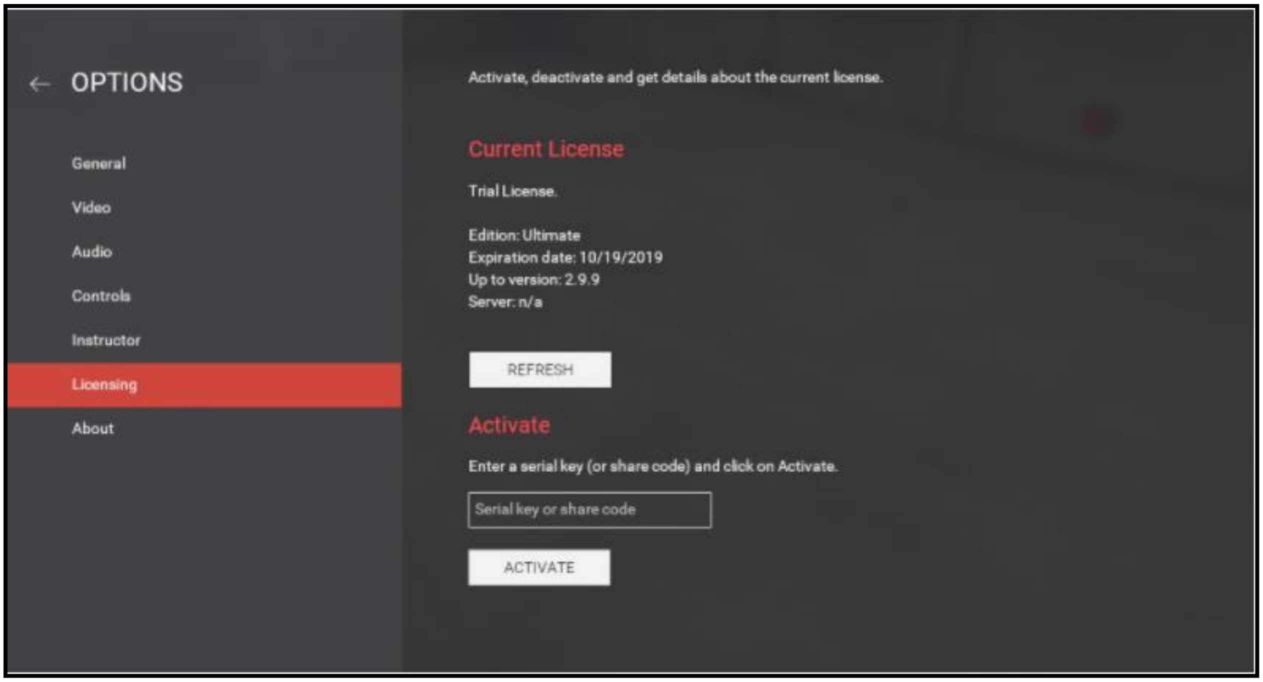

Figure II.14 Licensing and Activation

II.3.3 Getting started with the software

To be able to manipulate on the software it must learn a few basics including working with cameras, creating/editing scenes and controlling them with external technologies. At the end of this guide you will be able to create a virtual factory and use it together with your own PLC. [13]

II.3.3.1 Navigating [14]

One of the most important skills to learn in Factory I/O is how to use cameras. Cameras are used to navigate in the 3D space and are the key to interacting with parts or building new scenes. You can use three types of cameras:

1. **Orbit (1)**,
2. **Fly (2)**
3. **First Person (3)**

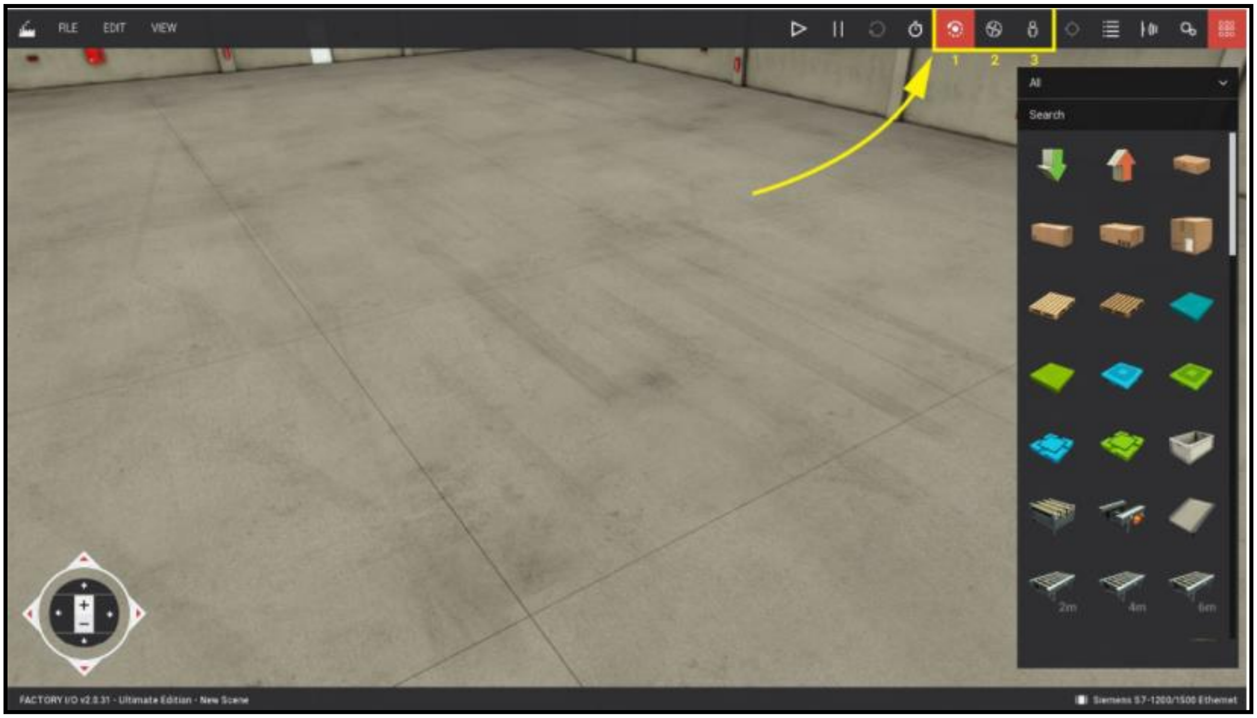

Figure II.15 Navigation system **[15]**

II.3.3.2 Opening a Scene [16]

To open a scene choose **Open** from the **File Menu (Ctrl + O)** and select it from the list by **Left-clicking**.

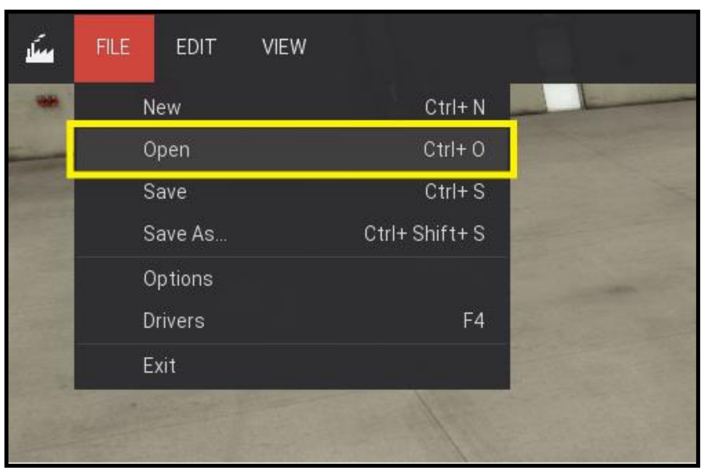

Figure II.16 Opening a scene

Factory I/O comes with 21 Scenes, which can be accessed under the **Scenes** tab. You may also use any of these scenes as a starting point for a new one by opening and saving it with a custom name.

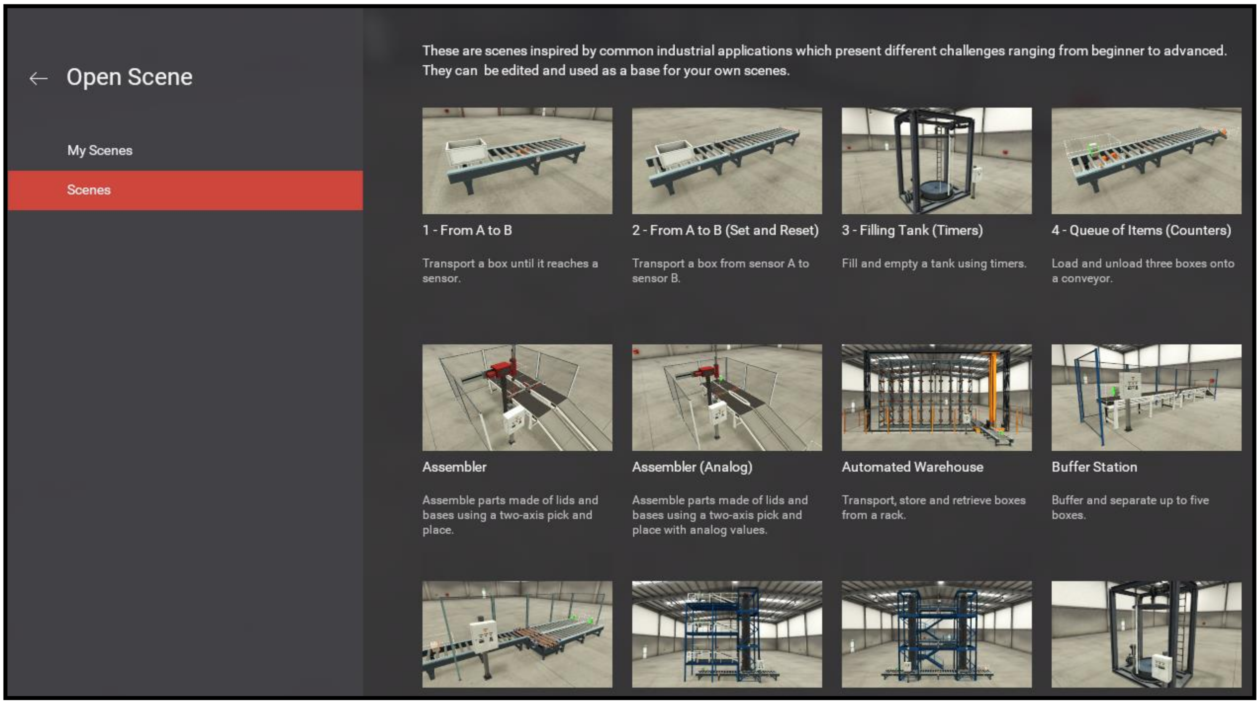

Figure II.17 Scenes faceplate

II.3.3.3 Creating a Scene [17]

Factory I/O includes a large selection of parts inspired by the most common industrial equipment. You create a virtual factory by placing and arranging these parts together. By following the next steps you will learn to create a simple sorting system.

1. Click on **File**, choose **New** (**Ctrl + N**) to create an empty scene.
2. On the **Toolbar**, select the **Orbit** camera - you should use the Orbit camera when editing. If the Palette is not visible click on the **Palette** icon.

3. Choose **Heavy Load Parts** category from the **Palette** and **Left-click** and **Drag** a Roller Conveyor (2m) into the scene. When a new part is created it automatically becomes selected, indicated by a white bounding box. When you drag a selected part it will move on the horizontal plane; to move it vertically press the **V** key and drag.

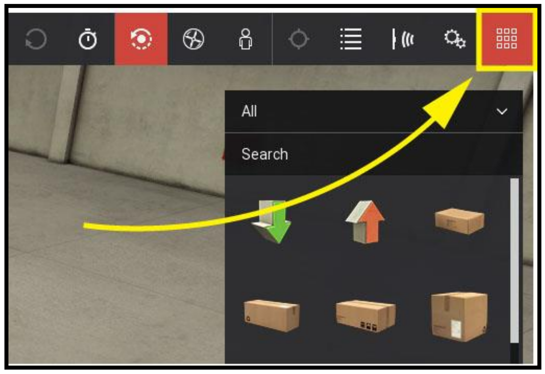

Figure II.18 Palette icon

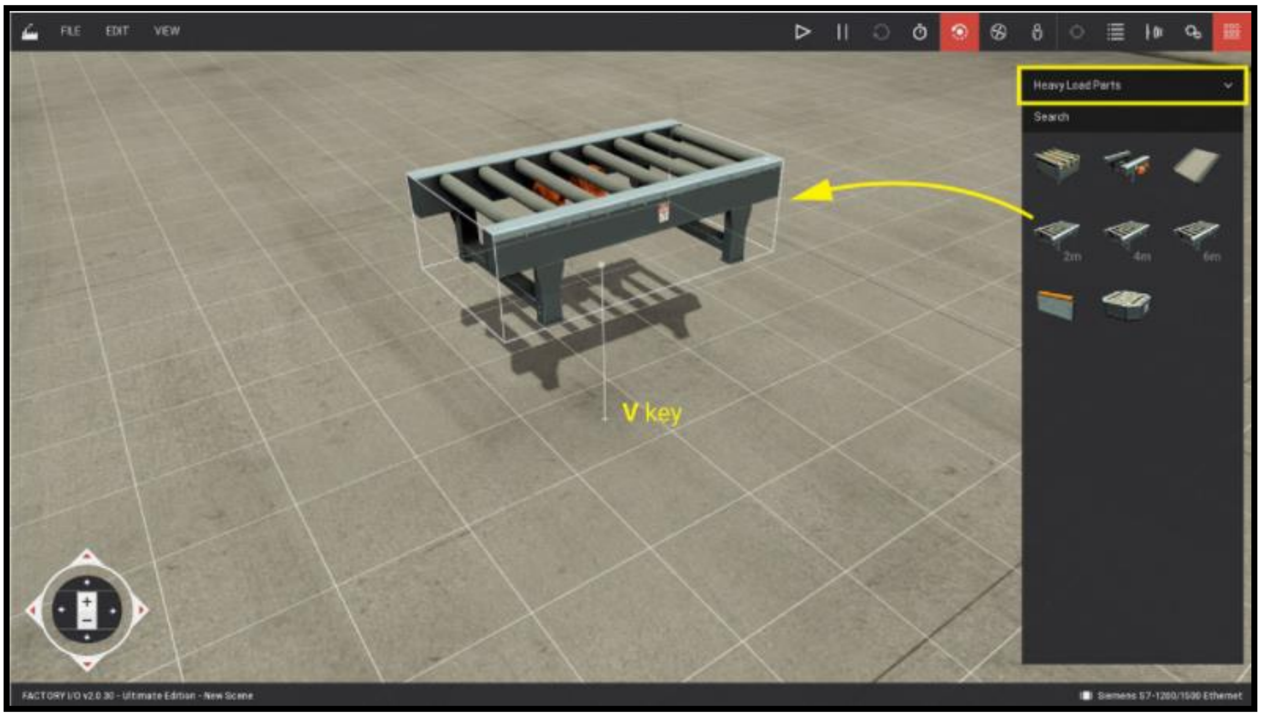

Figure II.19 Create a scene

4- We can select other **"Items"** **(**Figure II.20**)** from the palette such as Sensors, Panel, buttons…etc. Until is the scene done.

5- Finally and after we create the scene click on **File** and **Save (Ctrl + S)**, give it a name (and optionally a description) and click on **Save**.

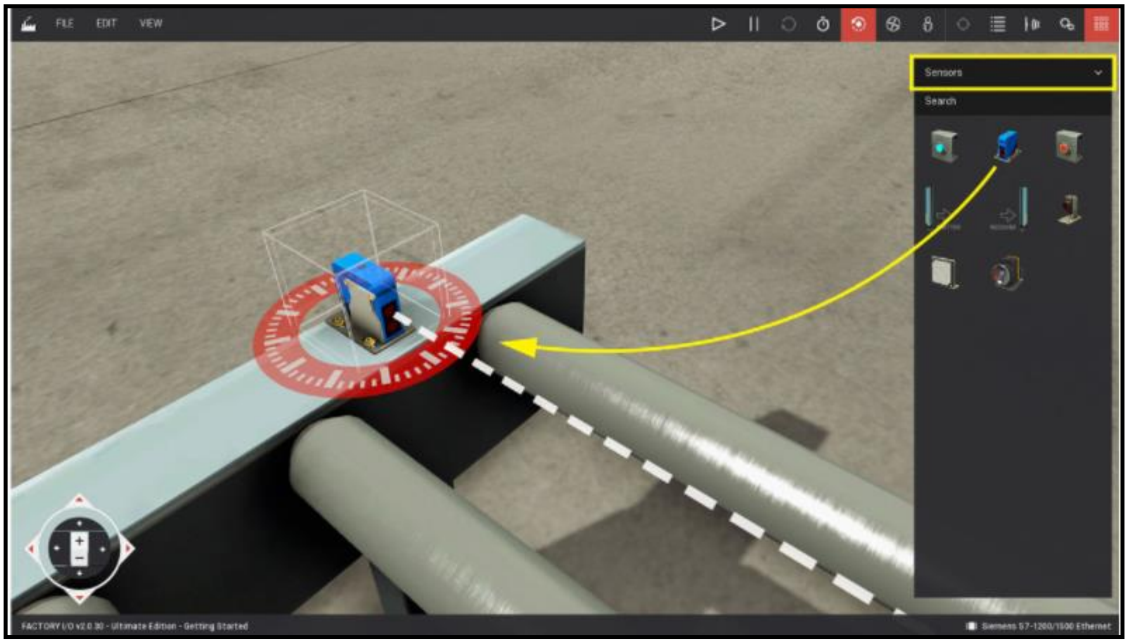

Figure II.20 select a sensor

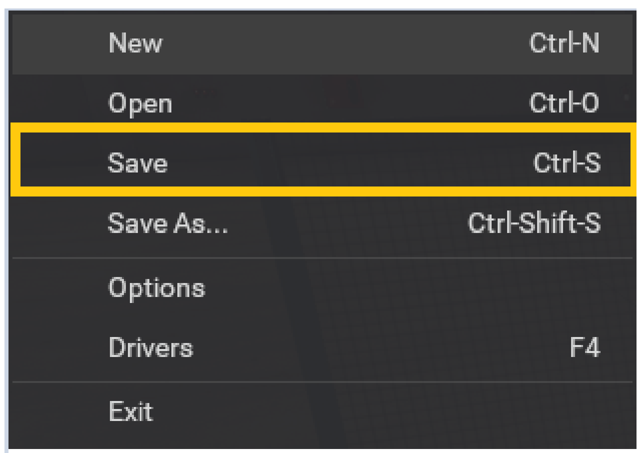

Figure II.21 Save a scene

II.3.4 Control With a PLC [18]

Now that you have created your factory it's time to control it with a PLC. But first, you should learn what I/O Drivers are and how to use them.

An I/O Driver is a built-in feature of Factory I/O responsible for "talking" to an external controller. Factory I/O includes many I/O Drivers, each one for a specific technology. You select a driver in Factory I/O based on the controller you want to use. Next, you configure this driver, so it knows how to "talk" to the controller and how to read and write I/O from it.

So, we will show up step by step how to use a Siemens S7-1200 PLC. However, most of the steps described here apply to other drivers as well.

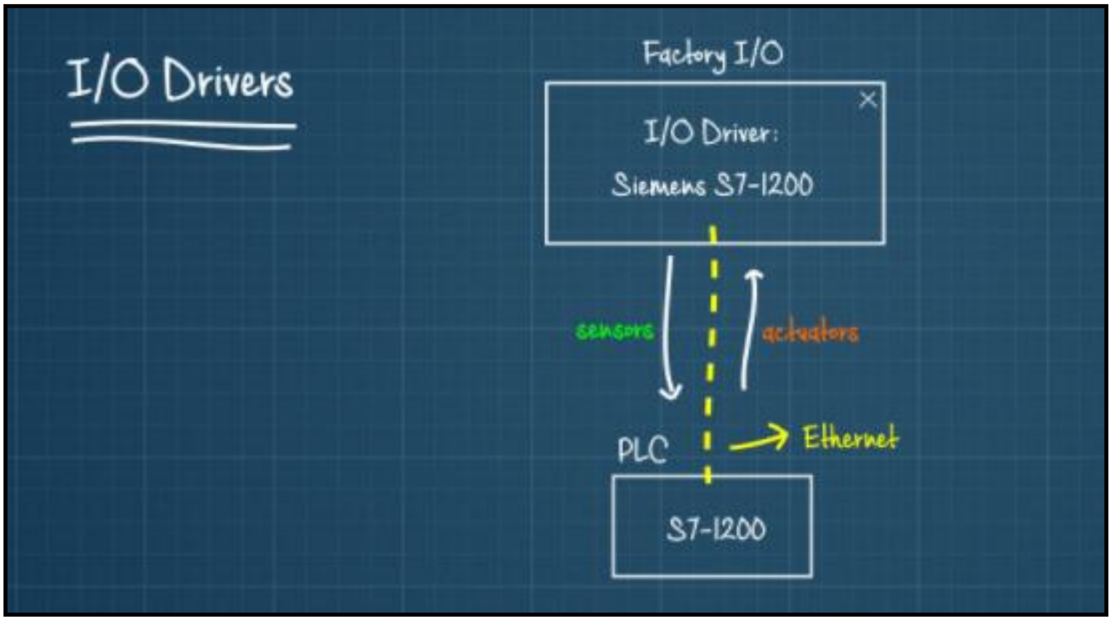

Figure II.22 S7-1200/Factory i/o connection's freehand drawing.

1- Open the Drivers window by clicking on **File** and next on **Drivers (F4)**. Alternatively, you may open the Drivers window by **Left-clicking** on the current driver displayed on the status bar.

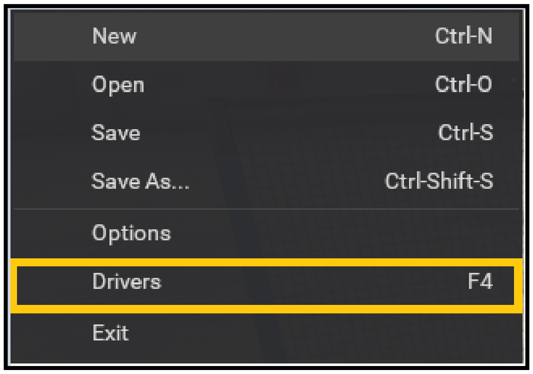

Figure II.23 Drivers option

2- Select **Siemens S7-1200/1500 Ethernet** driver from the list by **Left-clicking** on it.

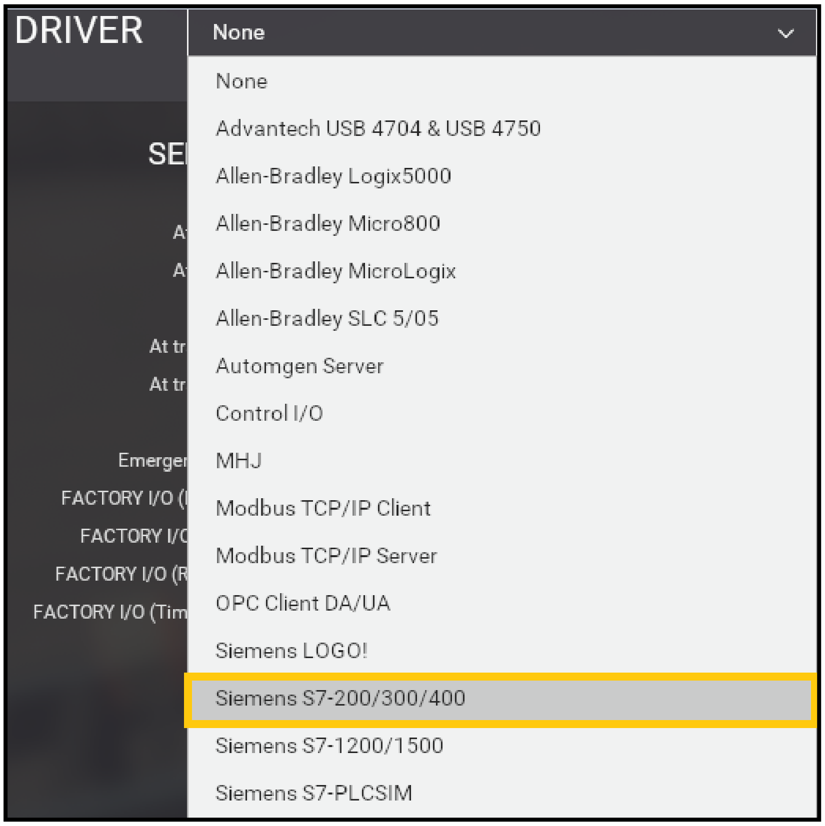

Figure II.24 Select Siemens S7-1200 driver

3- Click the **Configuration** button to setup the driver according to your PLC model and IP address.

Figure II.25 Configuration option

4- Choose S7-1200 model and fill in the **Host** field with your PLC's IP address. Proceed to the next step by clicking on the **Back Arrow** (or the **ESC** key).

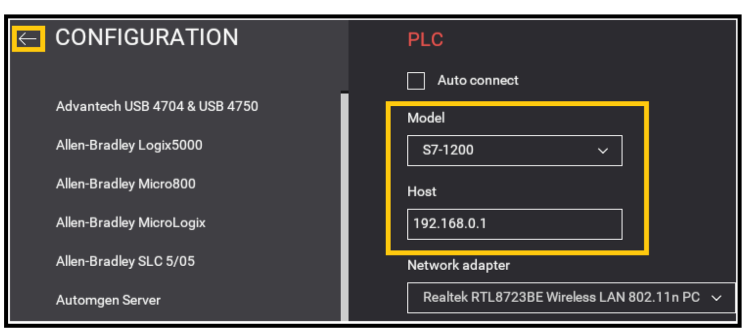

Figure II.26 Set Configuration

5- Press the **Connect** button to connect to the PLC. A successful connection will be indicated by a green sign displayed next to the drivers list.

Figure II.27 validate the connection

6- Start mapping tags by dragging and dropping each one onto the intended port. To remove a tag from a port, just drag it back to the list. Once you have mapped all tags, set the simulation to Run Mode and test your PLC logic.

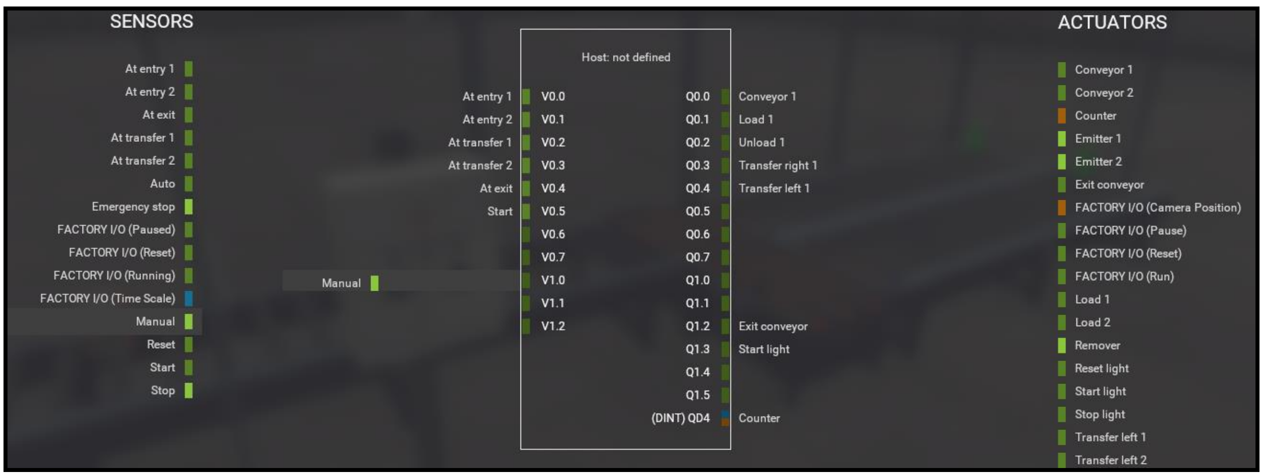

Figure II.28 Mapping the driver's tags

II.4 Conclusion

Using these two software we can manage to demonstrate how a tank process level control is automated by the Siemens S7-1214c DC/DC/DC programmable logic controller, supervise and control by the TP-900 comfort Human Machine Interface.

Chapter III :
Hardware Description
(S7-1200 PLC & Process Parts)

III.1 Introduction

The field of automatic control has been undergoing a transformation over the past twenty years. Twenty years ago, the engineering undergraduate had a course in feedback control theory and those interested in control engineering secured a position in the aerospace or chemical industries. Due to various factors, the number of control engineering positions in the aerospace industry has been declining, but the number of control engineering positions in manufacturing has been dramatically increasing to the point that the majority of control engineering positions is now in manufacturing and involves PLCs. [20]

III.2 PLC Architecture [21]

Typically a PLC system has the basic functional components of processor unit, memory, power supply unit, input/output interface section, communications interface and the programming device.

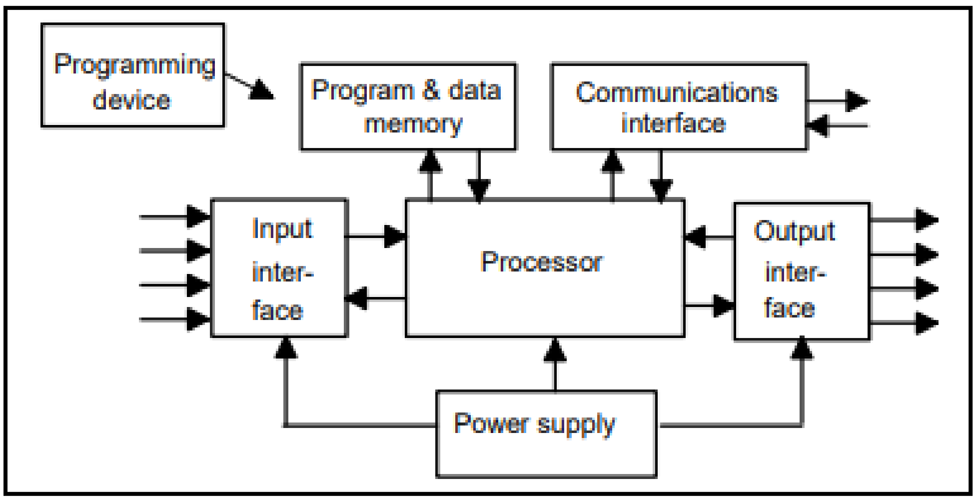

Figure III.1 CPU Architecture. **[21]**

The processor unit or central processing unit (CPU) is the unit containing the microprocessor and this interprets the input signals and carries out the control actions, according to the program stored in its memory, communicating the decisions as action signals to the outputs.

The power supply unit is needed to convert the mains AC. voltage to the low DC. voltage (5 V) necessary for the processor and the circuits in the input and output interface modules.

The programming device is used to enter the required program into the memory of the processor. The program is developed in the device and then transferred to the memory unit of the PLC.

The memory unit is where the program is stored that is to be used for the control actions to be exercised by the microprocessor and data stored from the input for processing and for the output for outputting.

The input and output sections are where the processor receives information from external devices and communicates information to external devices.

The communications interface is used to receive and transmit data on communication networks from or to other remote PLCs

III.3 Internal Architecture [21]

The basic internal architecture of a PLC. It consists of a central processing unit (CPU) containing the system microprocessor, memory, and input/output circuitry. The CPU controls and processes all the operations within the PLC. It is supplied with a clock with a frequency of typically between 1 and 8 MHz. This frequency determines the operating speed of the PLC and provides the timing and synchronization for all elements in the system. The information within the PLC is carried by means of digital signals. The internal paths along which digital signals flow are called buses. In the physical sense, a bus is just a number of conductors along which electrical signals can flow. It might be tracks on a printed circuit board or wires in a ribbon cable. The CPU uses the data bus for sending data between the constituent elements, the address bus to send the addresses of locations for accessing stored data and the control bus for signals relating to internal control actions. The system bus is used for communications between the input/output ports and the input/output unit.

III.3.1 Buses [21]

The buses are the paths used for communication within the PLC. The information is transmitted in binary form.

III.3.1.1 Data Bus

Carries the data used in the processing carried out by the CPU. A microprocessor termed as being 8-bit has an internal data bus which can handle 8-bit numbers.

III.3.1.2 Address Bus

Used to carry the addresses of memory locations. So that each word can be located in the memory, every memory location is given a unique address.

III.3.1.3 Control Bus

Carries the signals used by the CPU for control, e.g. to inform memory devices whether they are to receive data from an input or output data and to carry timing signals used to synchronize actions.

III.3.1.4 System Bus

Used for communications between the input/output ports and the input/output unit.

III.3.2 Memory areas [22]

The CPU provides the following memory areas to store the user program, data, and configuration:

1- Load memory is non-volatile storage for the user program, data and configuration. When a project is downloaded to the CPU, it is first stored in the Load memory area. This area is located either in a memory card (if present) or in the CPU. This non-volatile memory area is maintained through a power loss. You can increase the amount of load memory available for data logs by installing a memory card.
2- Work memory is volatile storage for some elements of the user project while executing the user program. The CPU copies some elements of the project from load memory into work memory. This volatile area is lost when power is removed, and is restored by the CPU when power is restored.
3- Work memory is volatile storage for some elements of the user project while executing the user program. The CPU copies some elements of the project from load memory into work memory. This volatile area is lost when power is removed, and is restored by the CPU when power is restored
4- Work memory is volatile storage for some elements of the user project while executing the user program. The CPU copies some elements of the project from load memory into work memory. This volatile area is lost when power is removed, and is restored by the CPU when power is restored.

III.3.3 PLC selection criteria

PLC selection criteria consists of:

1- Application requirements.
2- What input/output capacity is required?
3- What type of inputs/outputs are required?
4- What size of memory is required?
5- What speed is required of the CPU?
6- Electrical requirements.
7- Speed of operation.
8- Communication requirements.
9- Software.
10- Physical environments.

III.3.4 Types of Siemens PLC

- **S7-200 CPU**

The S7-200 CPU combines a microprocessor, an integrated power supply, input circuits, and output circuits in a compact housing to create a powerful Micro PLC. [23]

Figure IIII.2 S7-200 CPU

- **S7-300 CPU**

This is a mini modular PLC for the low and mid-range applications. Can expand the I/O without any issues when the tasks increase but there is limit foe expansion. It has inbuilt motion control applications. [24]

Figure IIIII.3 S7-300 CPU

- **S7-400 CPU**

S7-400 designed for system solutions for manufacturing and process automation. This process controller is suitable for data-intensive activities that are exceptionally common for the process industry.

S7-400 CPU available in Standard, Failsafe and High availability. [25]

Figure IVII.4 S7-400 CPU

- **LOGO**

LOGO! offers solutions for domestic and installation engineering (e.g. for stairway lighting, external lighting, sun blinds, shutters, shop window lighting etc.) [26]

Figure VII.5 LOGO

- **S7-1500 CPU**

SIMATIC S7-1500 is the modular automation system for the medium and upper performance ranges. Different versions of the controllers allow the performance to be matched to the respective application. [27]

Figure VIII.6 S7-1500 CPU

III.4 Description of the powerful and flexible S7-1200

III.4.1 Introduction of the Programmable Logic Controller [28]

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 provides a PROFINET port for communication over a PROFINET network. Additional modules are available for communicating over PROFIBUS, GPRS, RS485, RS232, IEC, DNP3, and WDC networks.

① Power connector
② Memory card slot under top door
③ Removable user wiring connectors (behind the doors)
④ Status LEDs for the onboard I/O
⑤ PROFINET connector (on the bottom of the CPU)

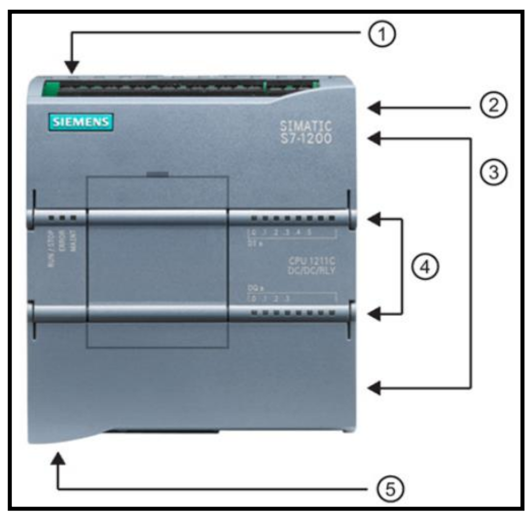

Figure VIIII.7 S7-1200 CPU

III.4.2 Expansion capability of the CPU [28]

The S7-1200 family provides a variety of modules and plug-in boards for expanding the capabilities of the CPU with additional I/O or other communication protocols.

- ✓ Signal Boards SB allows you to add analog or digital inputs or outputs without changing the size of the CPU. (Signal Boards can be integrated in CPU 1211C / 1212C and 1214C).

Figure VIIIII.8 S7-1200 Signal Board

- ✓ CM communication modules for serial RS 232 / RS 485 communication (up to 3 CM are possible for CPU 1211C / 1212C and 1214C)

Figure IXII.9 S7-1200 Communication Modules

- ✓ 2 MB SIMATIC memory cards up to 32 MB for storing program data and for easy replacement of CPUs in the event of maintenance.

Figure XII.10 S7-1200 Memory cards

- ✓ Signal module (SM) (digital SM, analog SM, thermocouple SM, RTD SM, technology SM)

Figure XIII.11 S7-1200 Signal Modules

III.4.2.1 S7-1200 Status LED [29]

The CPU and the I/O modules use LEDs to provide information about either the operational status of the module or the I/O.

The CPU provides the following status indicators:

- ✓ STOP/RUN
  - Solid yellow indicates STOP mode.
  - Solid green indicates RUN mode.
  - Flashing (alternating green and yellow) indicates that the CPU is in STARTUP mode.
- ✓ ERROR

- Flashing red indicates an error, such as an internal error in the CPU, a error with the memory card, or a configuration error (mismatched modules)
- Defective state (Solid red indicates defective hardware, all LEDs flash if the defect is detected in the firmware)

✓ MAINT (Maintenance) flashes whenever you insert a memory card. The CPU then changes to STOP mode. After the CPU has changed to STOP mode, perform one of the following functions to initiate the evaluation of the memory card:

- Change the CPU to RUN mode
- Perform a memory reset (MRES)
- Power-cycle the CPU

Figure XIIII.12 CPU LED's

III.4.3 Description of the project hardware [30]

We have used the S7-1214c DC/DC/DC 6ES7214-1AG40-0XB0 for our automated process control and that because when it comes to **flexibly** and **efficiently** performing automation tasks S7-1200 controllers are the ideal choice in the lower to medium performance range. They feature a comprehensive range of technological functions and integrated communication as well as especially compact and space-saving design.

1. CPU S7-1214

- **Features**

✓ DC/DC/DC compact 6ES7214-1AG40-0XB0
✓ Supply 24 VDC, 1.5A.
✓ Digital Input 14×24 VDC 6mA
✓ Digital Output 10×24 VDC 0.5A
✓ Analog Input 2×10 Bit 0-10 VDC
✓ Program/data memory 100 KB

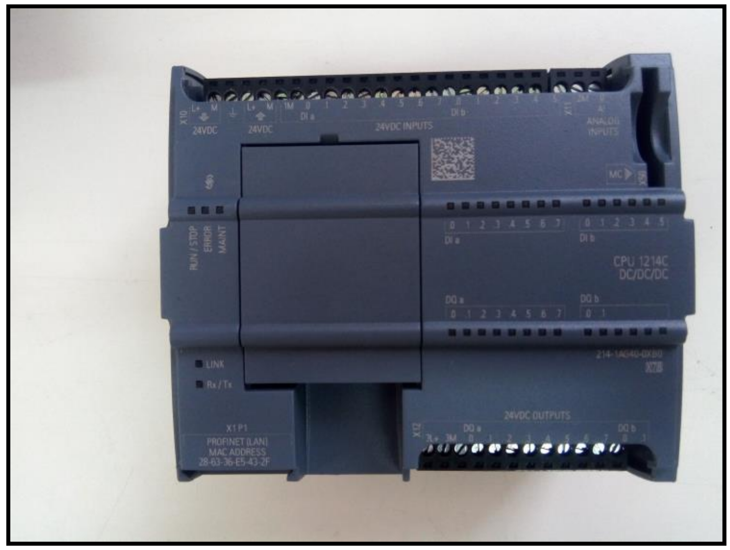

Figure XIIIII.13 CPU_S7-1214c DC/DC/DC 6ES7214-1AG40-0XB0

2. Signal Module 1231 Analog Input

- **Features**

✓ Supply 24 VDC, 65mA
✓ Analog Input 4×16 Bit ±10VDC / 0-20mA

Figure XIVII.14 Analog Input 16Bit

3. Signal Module 1231 Analog Output

- **Features**

✓ Supply 24 VDC, 45mA
✓ Analog Output 4×14 Bit ±10 VDC / 0-20mA

Figure XVII.15 Analog Output 14Bit

4. Power Supply (SITOP PSU 100S)

- **Features**

✓ Stabilized power supply
✓ input: 120/230 V AC, 50/60 HZ
✓ 02 output: DC 24 V/10 A

Figure XVIII.16 SITOP PSU 100S

5. Profinet Cable

A PROFINET network is based on standard IEEE 802.3 Ethernet. Therefore, we can utilize standard Ethernet cables to build your PROFINET network.

- **Features**

✓ Port RJ45
✓ 4-wire shielded, green-colored cable
✓ 100 Mbps Fast Ethernet at a distance of 100 meters

Figure XVIIII.17 Ethernet RJ45

6. Switch Scalance X208

- **Features**

✓ Supply 24 VDC
✓ 6GK5208-0BA10-2AA3
✓ Managed IE switch, 8x 10/100 Mbit/s RJ45 ports
✓ LED diagnostics
✓ PROFINET IO device

Figure XVIIIII.18 Switch Scalance X208

III.5 Description of the Human Interface Module

III.5.1 Introduction [31]

Visualization is part of the standard repertoire for most machines these days. The cost factor plays a crucial role in this case, especially for small machines and simple applications. HMI devices with basic functions are often fully sufficient for simple applications.

III.5.2 The Siemens HMI Brand

III.5.2.1 Simatic Comfort Panel [32]

SIMATIC HMI Comfort Panels are designed for implementation of high-performance visualization applications on the machine-level. High performance, functionality and numerous integrated interfaces offer the greatest convenience in high-end applications.

Siemens offers simatic comfort displays panels with 4", 7", 9", 12", 15", 19" and 22".

Figure XIXII.19 Siemens Comfort Touch Panels

III.5.3 The Siemens TP-900 Comfort HMI

- **Features**

1. SIMATIC HMI TP900 Comfort a comfort panel
2. Supply voltage 24VDC [19.2-28.8 VDC]
3. touch operation, 9" widescreen TFT display
4. 16 million colors
5. PROFINET interface RJ45, MPI/PROFIBUS DP interface RS485
6. 12 MB configuration memory
7. Windows CE 6.0configurable from WinCC Comfort V11

Figure XXII.20 Siemens TP900 Comfort Touch Panel

8. Horizontal image resolution 800 pixel
9. Vertical image resolution 480 pixel

III.6 Description of the Simulate Industrial Process parts

III.6.1 Tank Water

- **Features**

- Height: 3 m
- Diameter: 2 m
- Discharge pipe radius: 0.125 m

Figure XXIII.21 Simulated Tank Water

III.6.2 Control Valves

- The control valve receives a signal from a controller such as a PLC in order to operate.
- Control valves are controlled by pneumatic actuators that can be positioned with signals between 0 and 10 V.
- Control valves allow water to whether come in or come out of the tank with :
-

  - Max. input flow: 0.25 m^3/s
  - Max. output flow: 0.3543 m^3/s

Figure XXIIII.22 Fill Control Valve

Figure XXIIIII.23 Drain Control Valve

III.6.3 Differential Pressure Flow Meters [33]

Differential pressure flow meters use laminar plates, an orifice, nozzle or Venturi tube to create an artificial constriction then measure the pressure loss of fluids as they pass that constriction. According to Bernoulli's principle, the pressure drop across the constriction is proportional to the square of the flow rate. The higher the pressure drop, the higher the flow rate. These rugged, accurate meters are ideal for a wide range of clean liquids and gases.

Figure XXIVII.24 Analog Flow Meter

III.6.4 Capacitive Level Sensor

Capacitive proximity sensors are non-contact devices that can detect the presence or absence of virtually any object regardless of material. They utilize the electrical property of capacitance and the change of capacitance based on a change in the electrical field around the active face of the sensor. [34]

Capacitive sensing technology is often used in other sensing technologies such as:

- Liquid level
- Flow
- Pressure

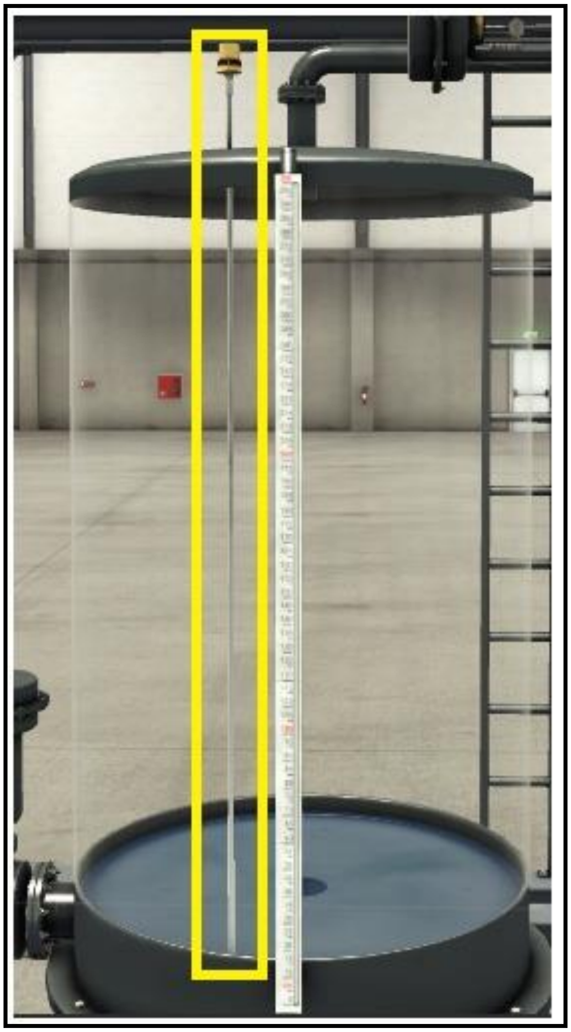

Figure XXVII.25 Capacitive Level Sensor

III.7 Conclusion

We have represented in this chapter all the different hardware we have in the lab such as programmable logic controller and its expand modules plus the panel display and all the extra accessory we need to realize the project and also the simulated hardware within the 3D simulator.

Finally we will move to the realization part of this project in the fourth chapter.

Chapter IV :

Realization & Parameterization of the Process Level Control

IV.1 Introduction

This chapter I dedicated for my application to explain how can we automate a process level control and be able to supervise and control it using Siemens Tia portal's software and S7-1214c DC/DC/DC CPU.

We will explain step by step the creation, configuration and testing of our automated, supervision and control of the project.

IV.2 Tia Portal V16 Configuration and Parameterization

To create my project's program, I have to set up some parameters in Tia Portal (Totally Integrated Automation Portal) first, so I will start with:

- **"RUN"** Tia Portal V16 software.

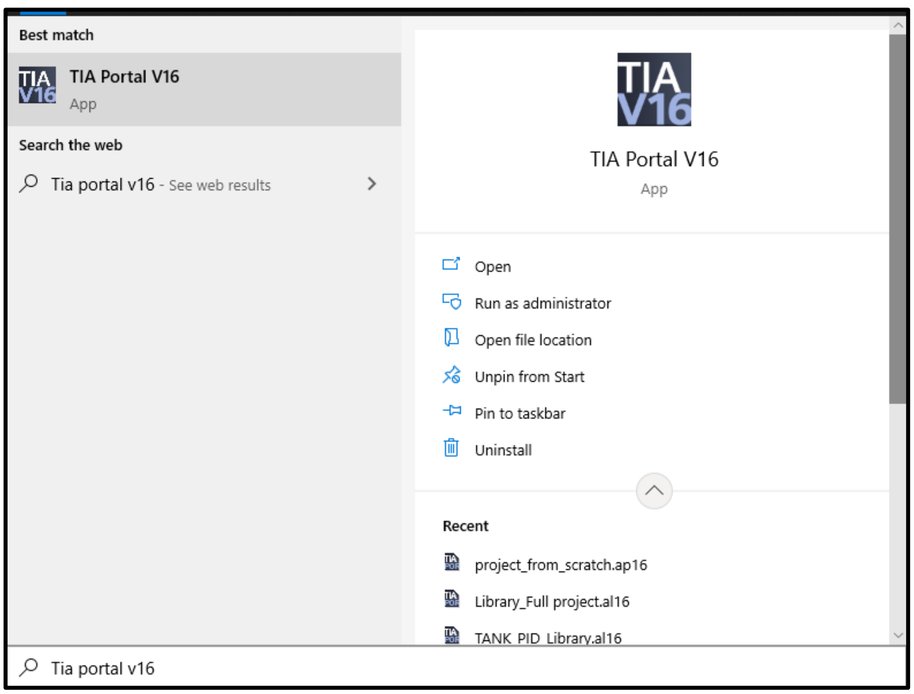

Figure IV.1 RUN Window

- Create a Project named “PLC_IMSI”

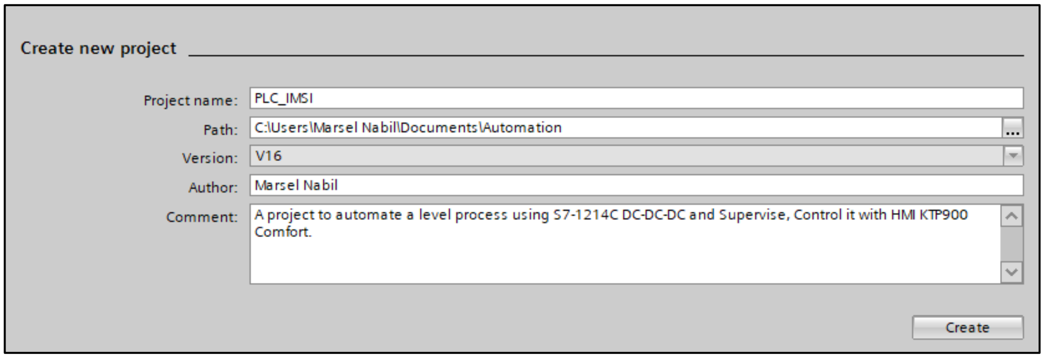

Figure IIV.2 Project Name

- First step is, to configure the “Hardware” by adding the required devices whether it’s a HMI, PLC (S7-200, S7-300, S7-400, S7-1200 or S7-1500 series), or PC stations (SCADA), but most important is to be sure of the device’s reference and the version of our actual real hardware.

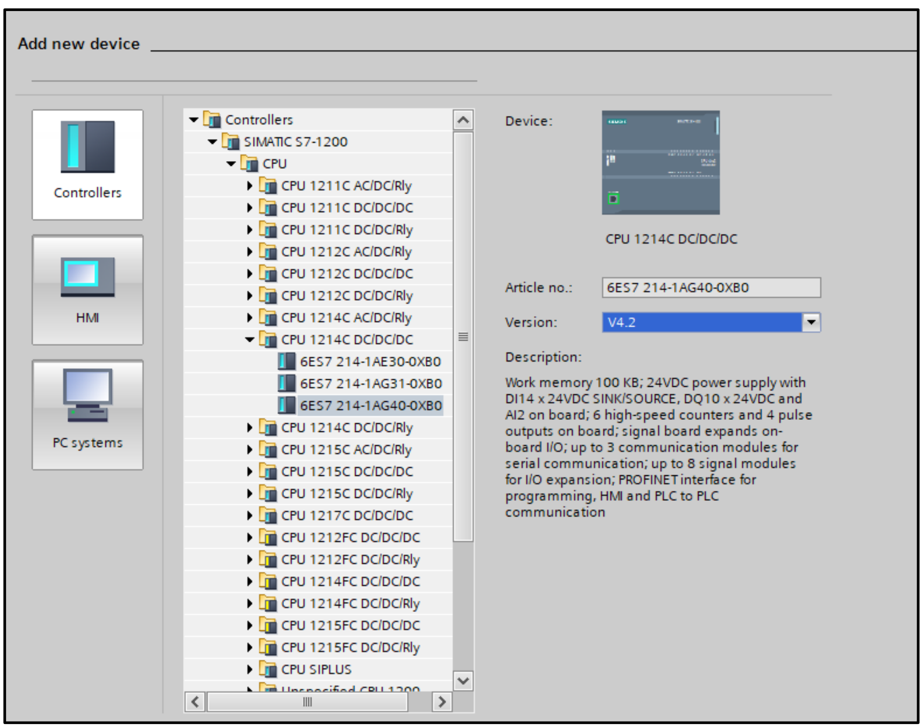

Figure IIIV.3 Add CPU 1214C DC/DC/DC

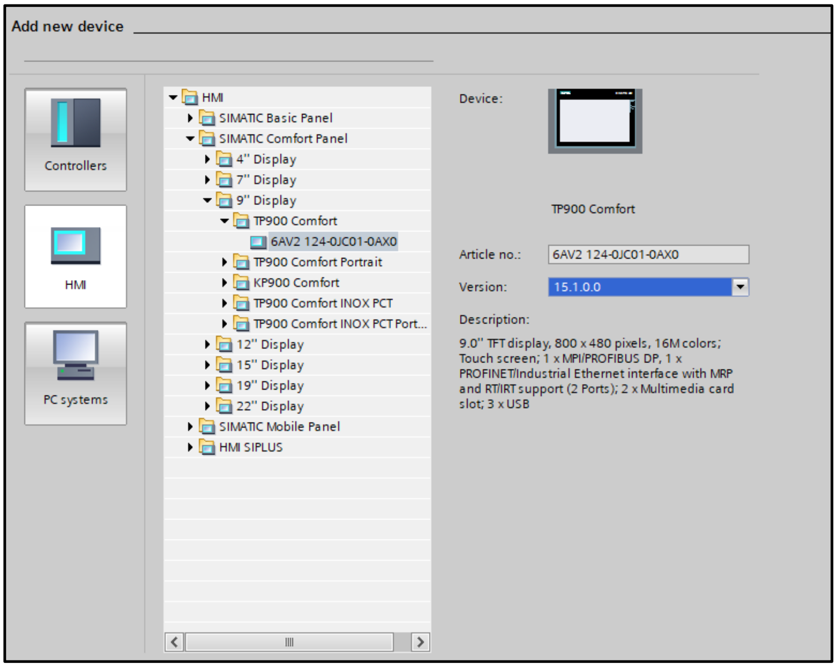

Figure IVV.4 Add TP900 Comofort HMI.

- After adding the CPU and the HMI required, we can drag and drop also from the catalog the analogic expand modules or Signal modules to our “Rack”.

  Project view >Devices & Networks >Device view.

  1. SM1231 AI 4×16 BIT 6ES7 231-5ND32-0XB0.
  2. SM1232 AQ 4×14BIT 6ES7 232-4HD32-0XB0.

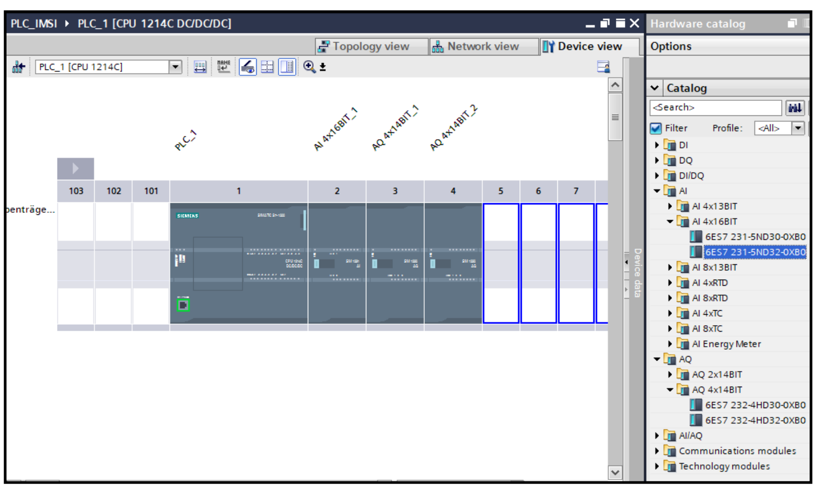

Figure VV.5 Add Expand Modules.

- After the hardware configuration, we move to the software parameterizations (Create a ProfiNet Network), in the Network view by adding a subnet named “IMSI_PN”

Project view >Devices & Networks >Network view.

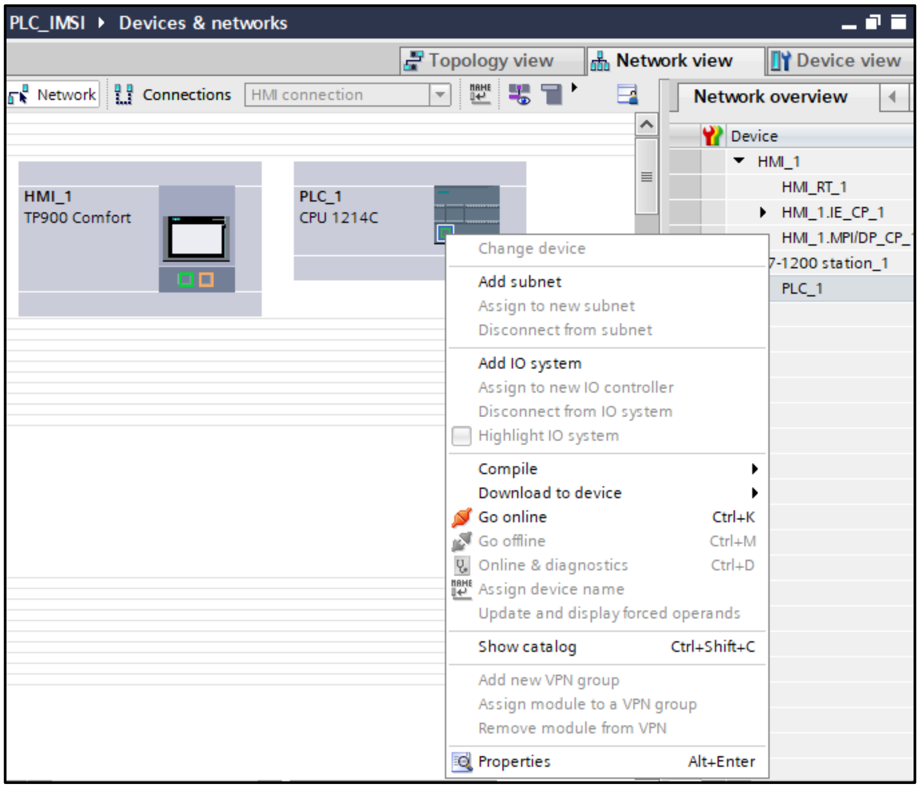

Figure VIV.6 Create Profinet Network

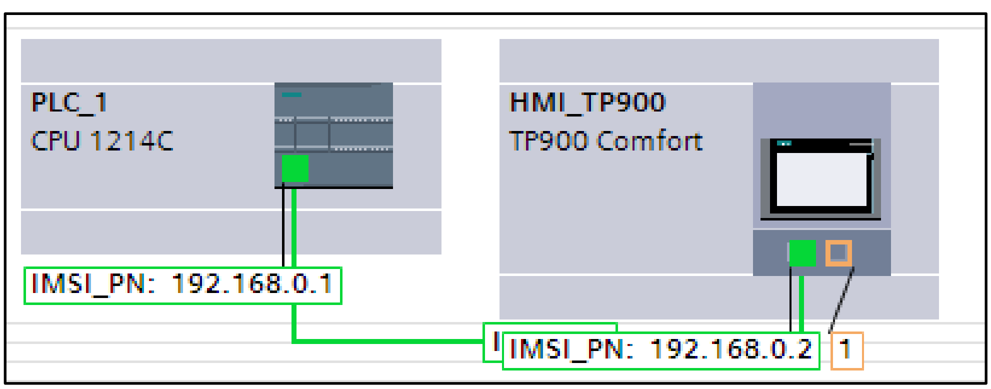

Figure VIIV.7 Profinet Connection

IV.3 Process Level control's programming

- At this point, we have configured & parameterized our project, and now we are ready to start programming the process level control.

  We will start with the System's program in the Main OB1 which consider as an important part of our program or any program, so it will contain the START, STOP, RESET, ESD (Emergency shutdown), Auto/Manual mode buttons and switches for the real sensors/actuators, and the HMI using memory bits.

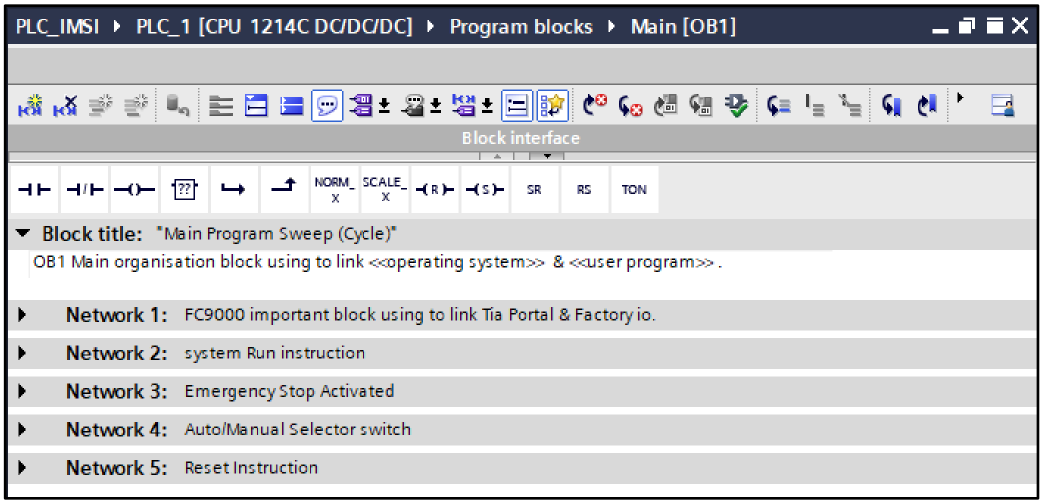

Figure VIIIV.8 Main OB Networks.

- After that, we move to the application itself, for that we will divide it into three parts:

  1. Manual Mode.
  2. Semi-Automatic Mode.
  3. Automatic Mode or PID Compact Mode.

IV.3.1 Manual Mode Blocks

- We create another organization block, called OB123 with a high class priority, which does not need to be called by OB1, we program a manual Fill & Drain Mode using Move instruction, so by set the **"Fill_PB"** to 1 will enable Move instruction to move 10 volts to the analogical fill valve and by releasing the push button, 0 volts will move the fill valve.

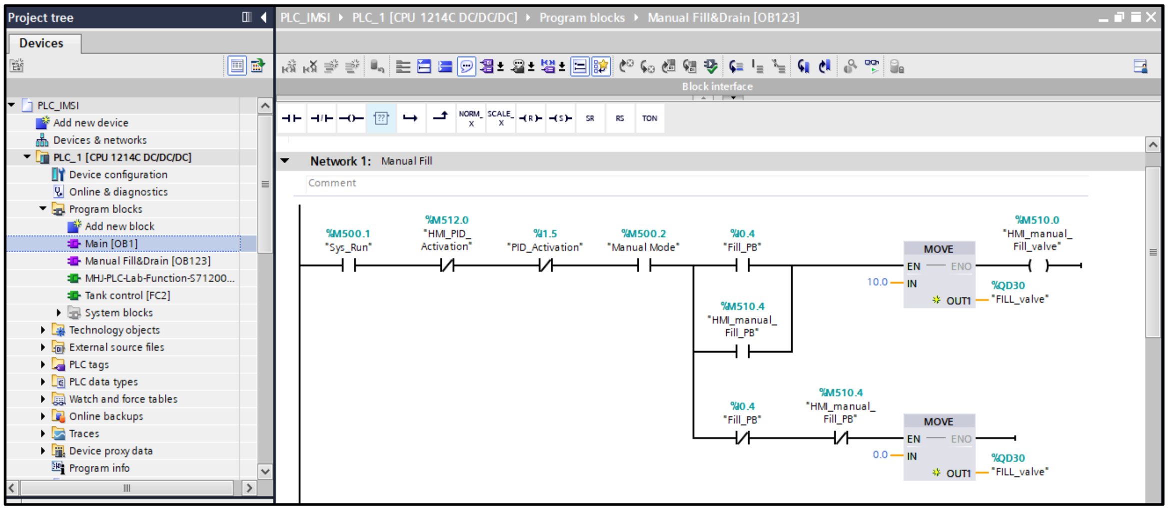

Figure IXV.9 Manual Fill Network.

- Then, we add the same LADDER program for the Manual Drain, except we will move the volts to the drain valve.

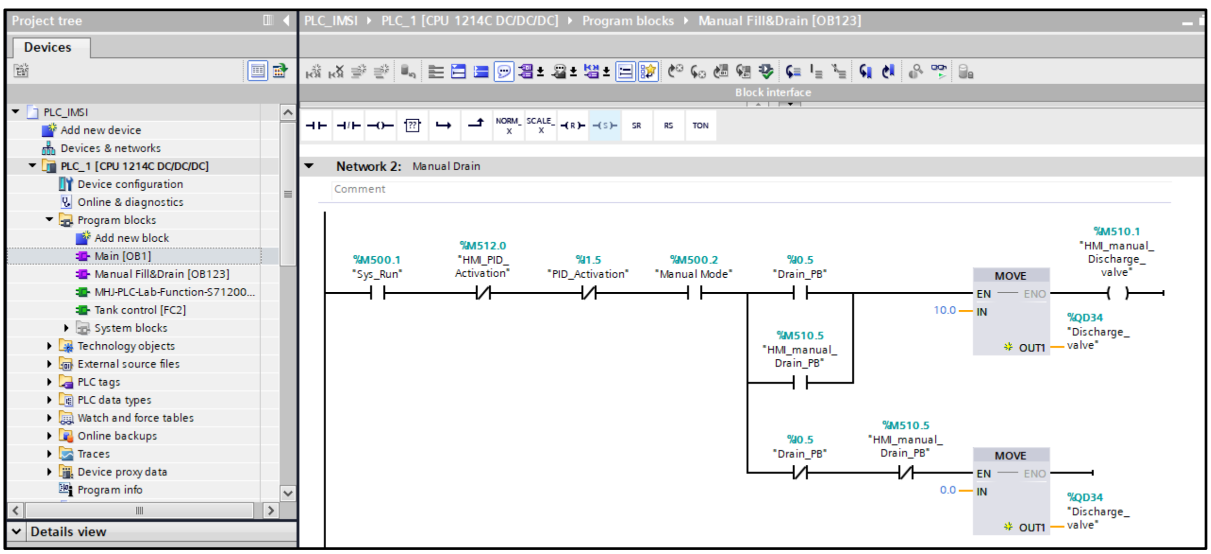

Figure XV.10 Manual Drain.

IV.3.2 Auto Fill & Drain or "Semi-Automatic"

- By adding another organization block [OB125], we will be able to program three Auto networks.
    1. **"Fill or Drain Mode"** network, because it's a semi-automatic we have to choose whether we are aim to **Fill** or **Drain** the tank.

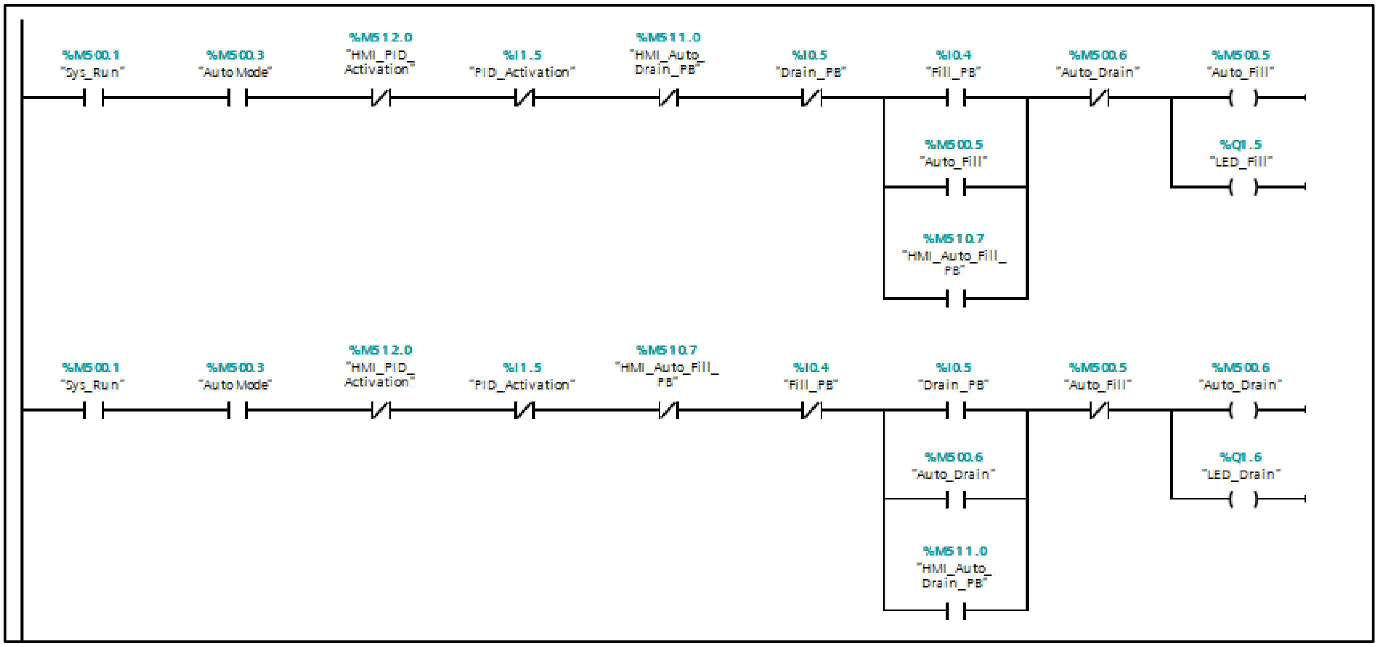

Figure XIV.11 Fill or Drain Mode Network.

2. **"Auto Fill Settings"** network for filling the tank at a precise points of level 25% for 75 cm, 50% for 150 cm, 75% for 225 cm and obviously 100% for 300 cm which is the high of the tank level, so we will manage to exceed these points by using as usually Move instruction.

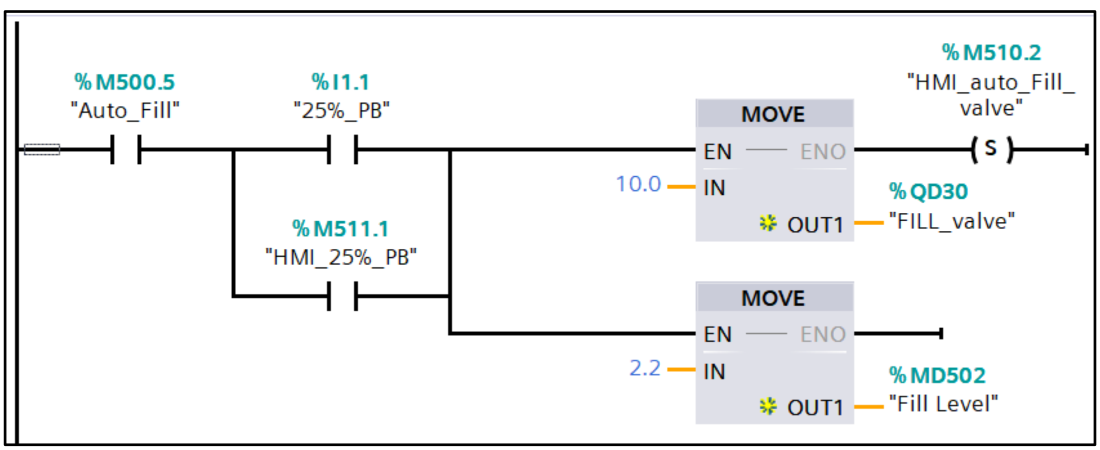

Figure XIIV.12 25% Auto Fill Network.

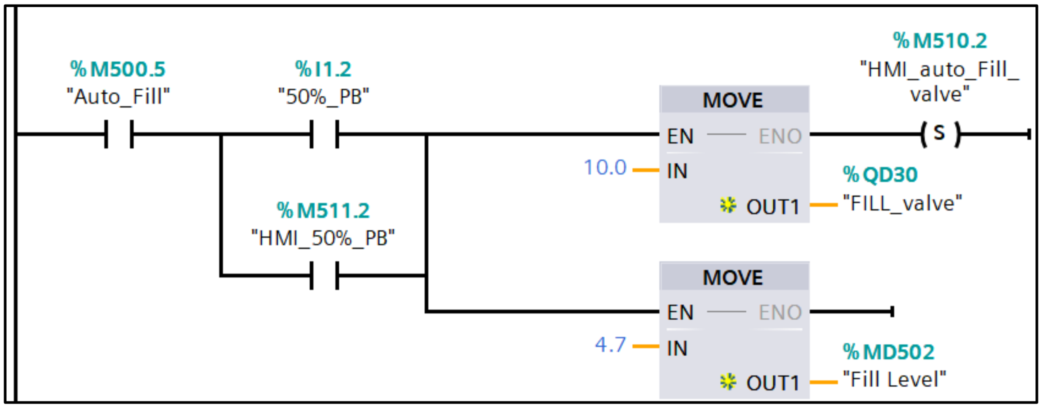

Figure XIIIV.13 50% Auto Fill Network.

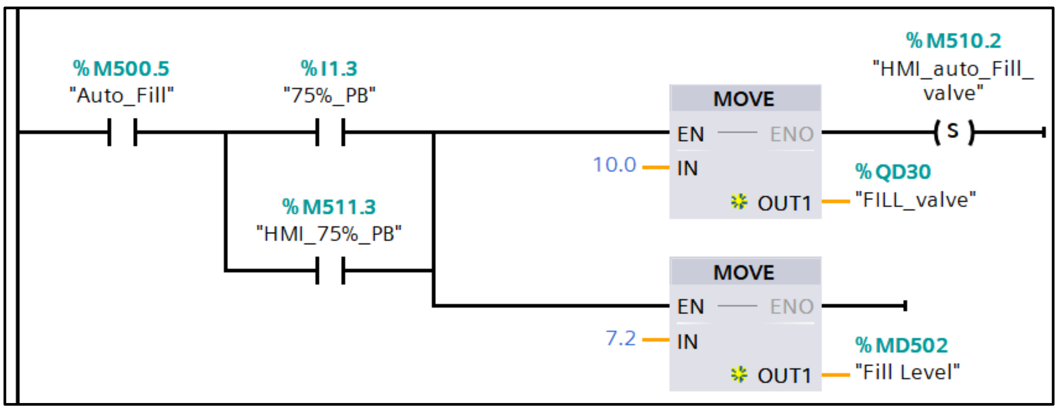

Figure XIVV.14 75% Auto Fill Network.

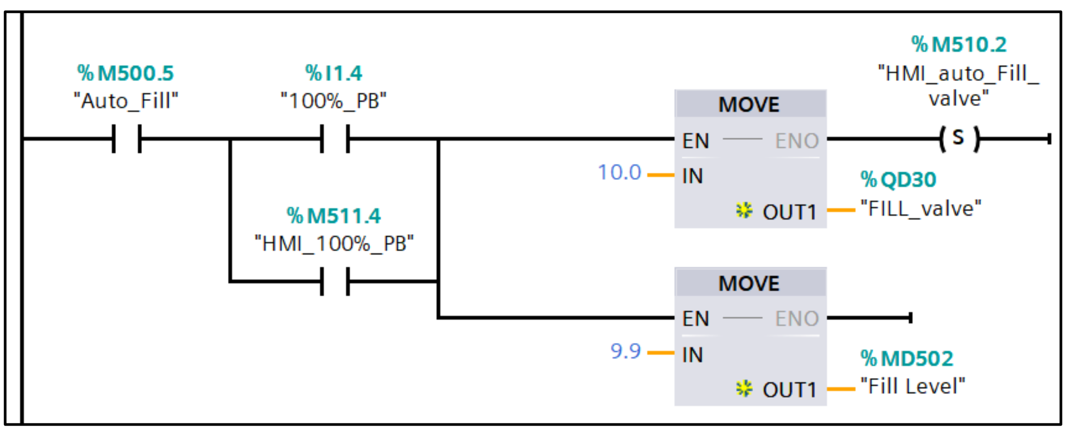

Figure XVV.15 100% Auto Fill Network.

3. Adding a comparator operation “**Greater** or **Equal** “to make sure that the fill valve is going to close when the level water required is exceeded.

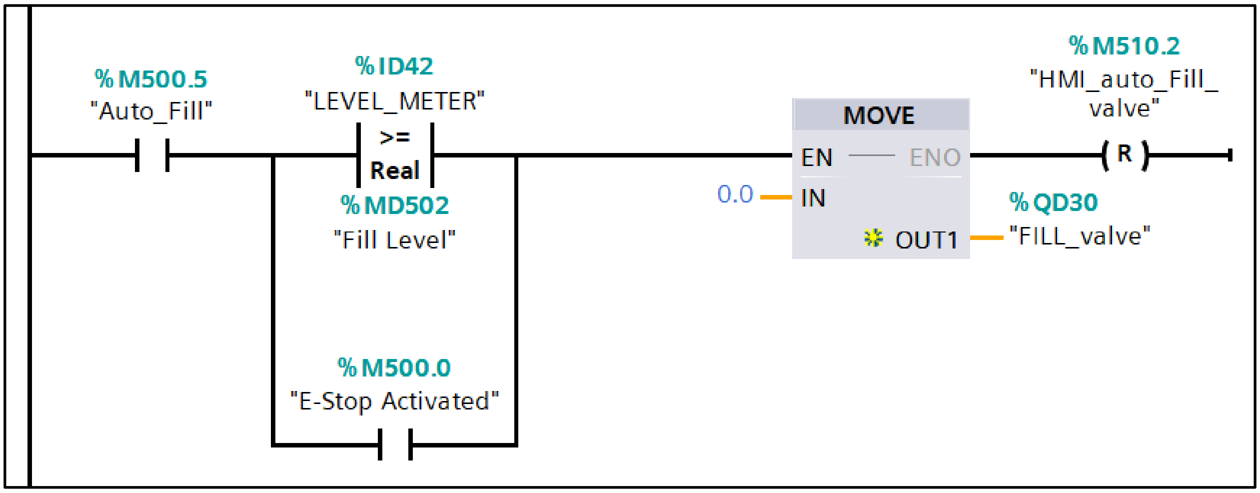

Figure XVIV.16 Auto-Fill Comparison Network.

5. Finally, a third network for the “**Auto Drain Settings**”, which basically the same as the **“Auto Fill”** except we will be aim to control the **“Discharge valve”** or **“Drain valve”.**

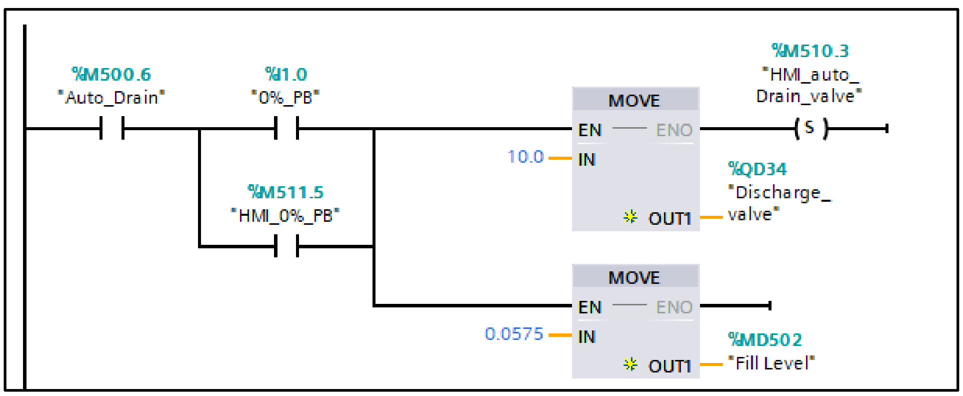

Figure XVIIV.17 0% Auto Drain Network.

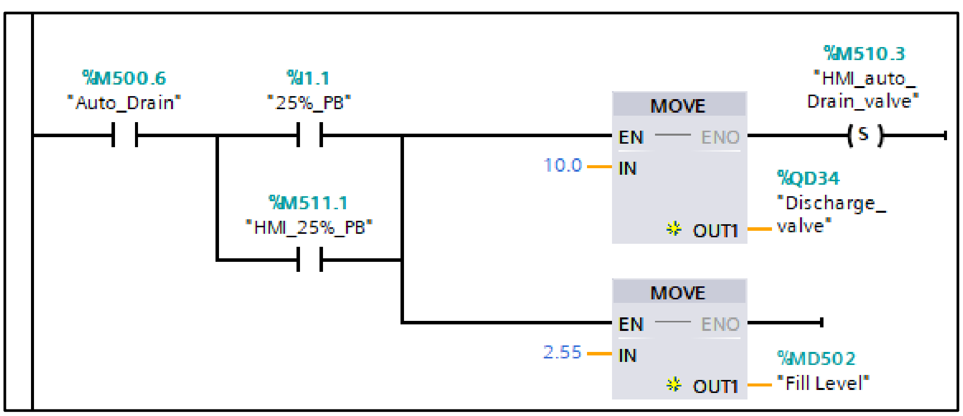

Figure XVIIIV.18 25% Auto Drain Network.

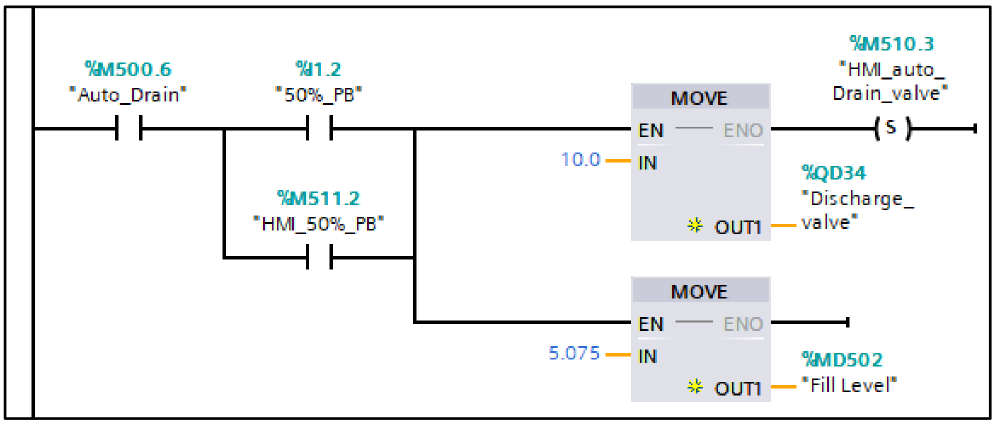

Figure XIXV.19 50% Auto Drain Network.

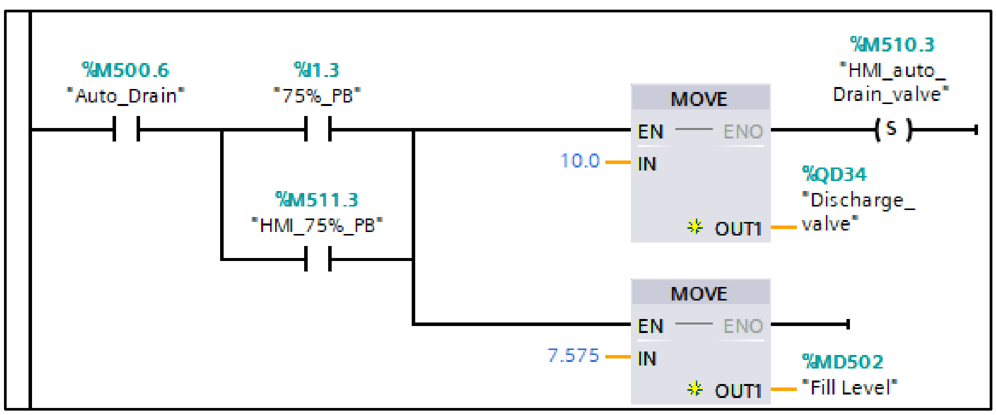

Figure XXIV.20 75% Auto Drain Network.

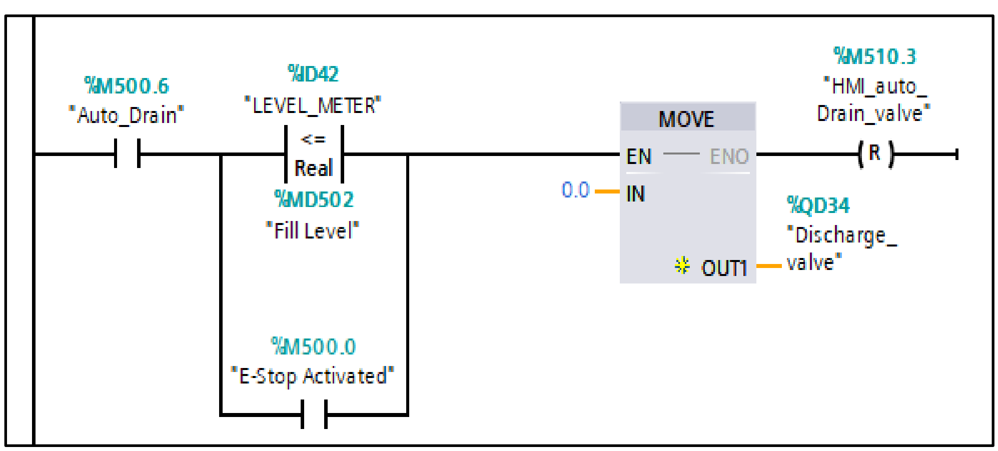

Figure XXV.21 Auto Drain Comparison Network.

IV.3.3 Automatic Mode or PID Compact Mode

- For the PID compact mode, we will use specifically an organization block called **“Cyclic interrupt”** or [OB 30], which allows us to start programs at periodic intervals, independently of cyclic program execution [OB main]. Plus, the intervals can be defined in the properties, in our case we set it for 100ms.

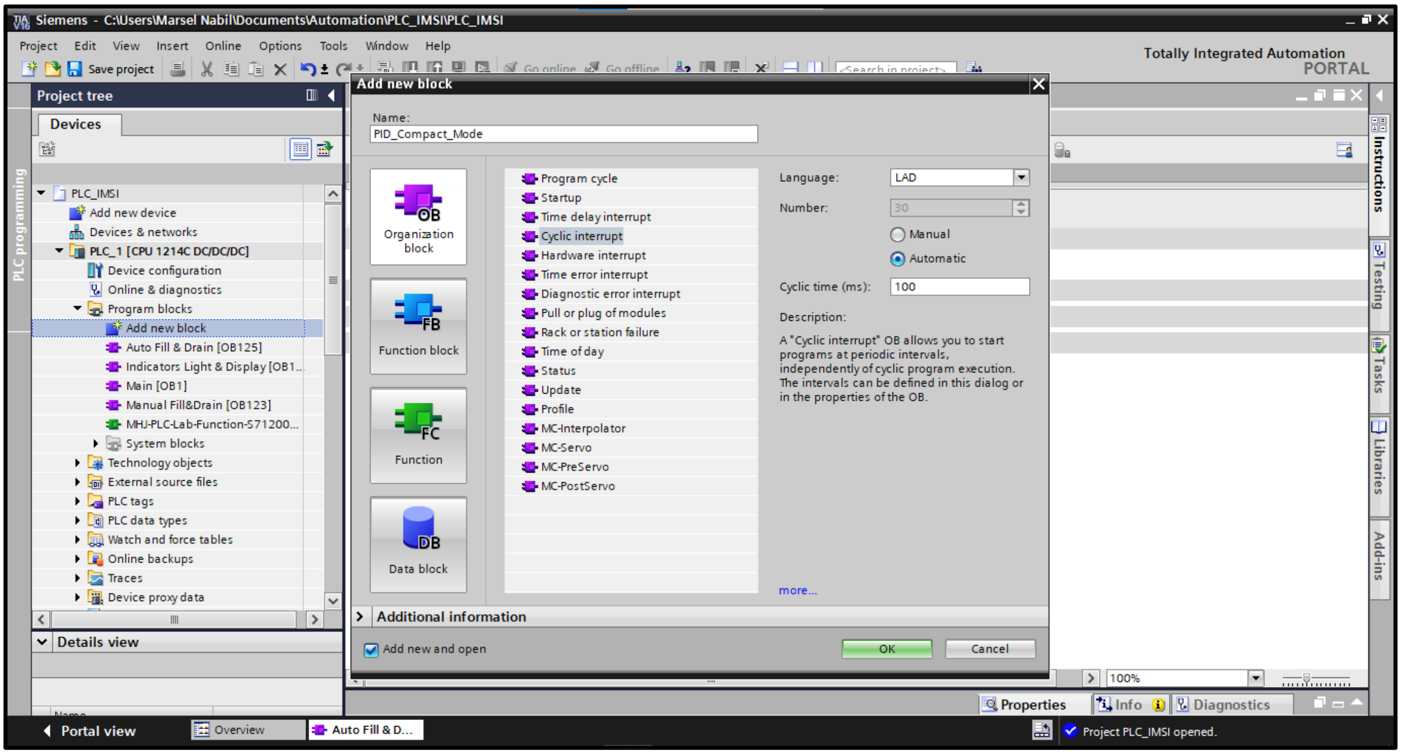

Figure XXIIV.22 PID Compact Block.

- Using this cyclic interrupt block, we will manage to create our automatic network to automate process level control with PID controller for both valves by drag and drop the PID compact V2.3 from the technology instruction.

1. "PID_Compact_Fill_Valve Block"

To complete the PID compact parameterization, we have to open the configuration window and set some standard parameters up such as:

a) The control type (Temperature, pressure, length, Flow, Torque) with the right measure units.
b) Inputs/Outputs parameters, means that we are going to use Analogue (Input PER& Output PER) or digital values (Input& Output).
c) Process values high limits (300 cm) & low limits (0 cm).

d) And the most crucial parameter, the PID parameters (Proportional gain, Integral action time and the Derivative action time).

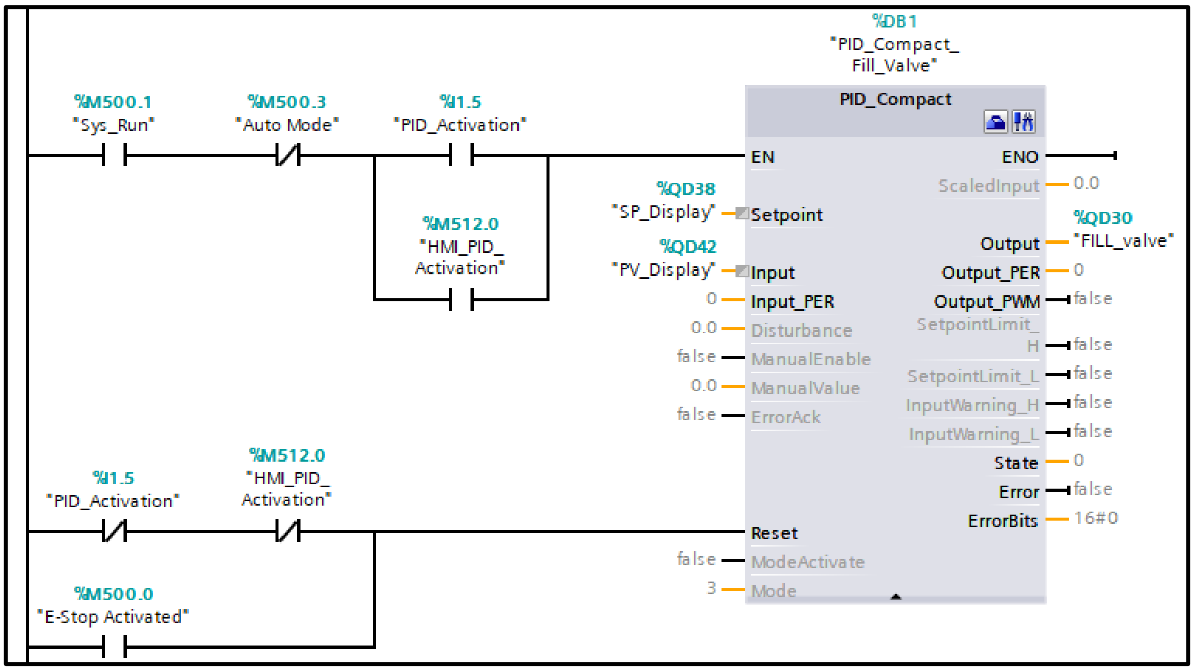

Figure XXIIIV.23 PID Compact Fill Valve Network.

2. "PID_Compact_Drain_Valve block"

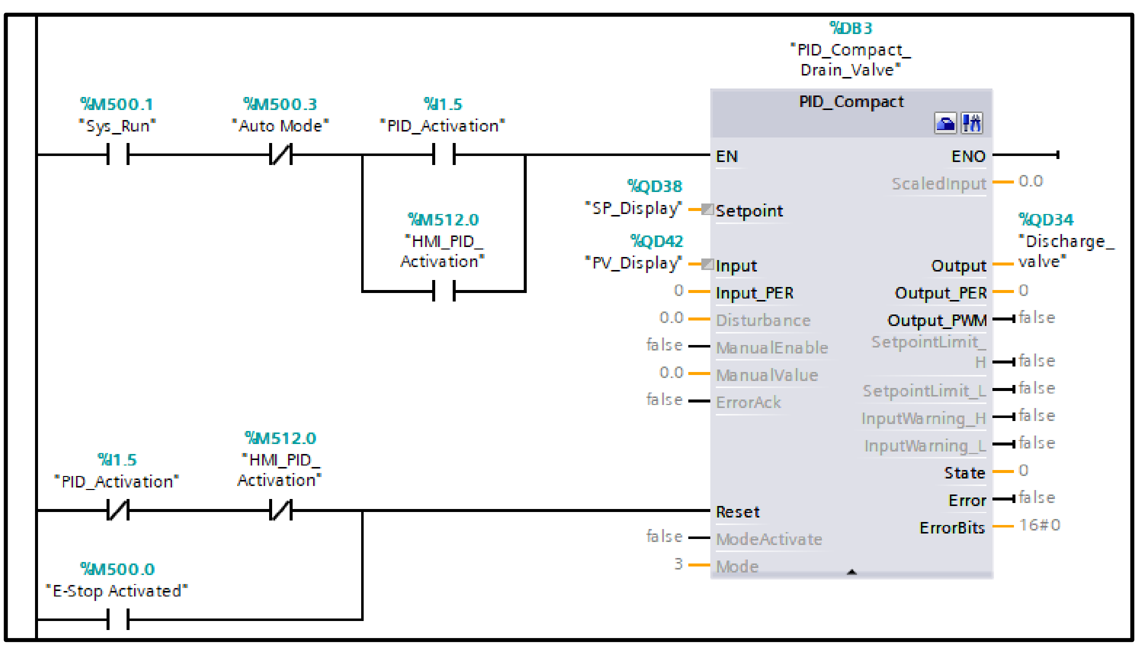

Figure XXIVV.24 PID Compact Drain Valve Network.

IV.3.4 Setpoint & Process Value Function

We add a block for our tank's inputs/outputs analogue values, a function block named SP & PV [FC] which we will manage to use it by calling from the main block [OB01] to normalize and scale SP and PV for our hardware (potentiometer & Capacitive sensor)

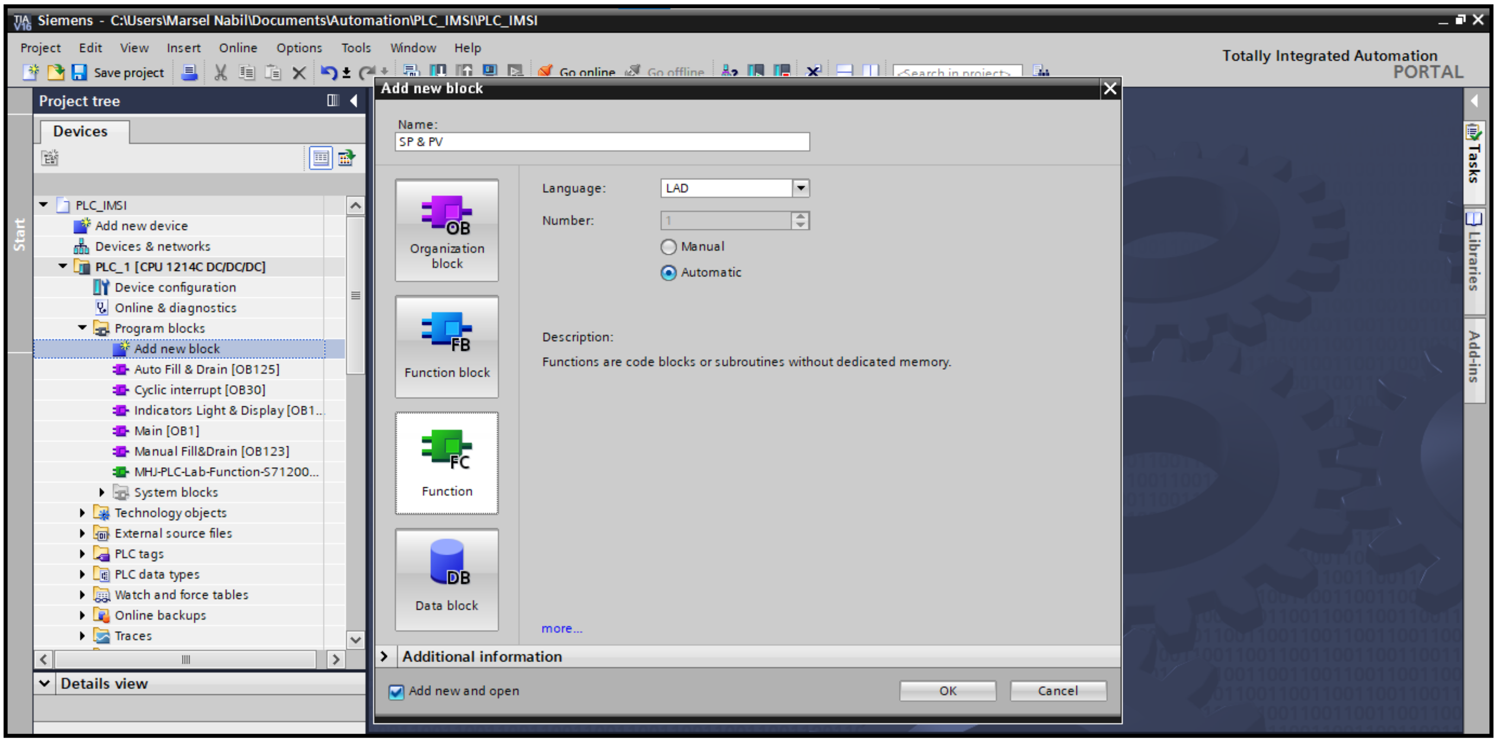

Figure XXVV.25 Create SP & PV Function.

IV.3.4.1 Network 01 Setpoint SP

ID38 (double integer) refers to the SP potentiometer [0-10v], so we use NORM_X instruction to normalize analogue values to [0-100%] then scale it SCALE_X depending on our tank's level which in our case [0-300cm] finally display it using QD38.

IV.3.4.2 Network 02 Process Value PV

ID42 refers to the analogue capacitive sensor [0-10v], so by detecting the water level meter the input analogue module will normalize and scale PV values then display it on the PV display.

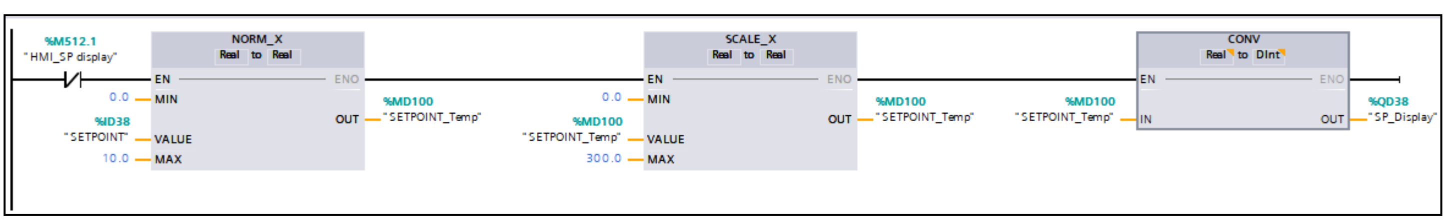

Figure XXVIV.26 Setpoint Network.

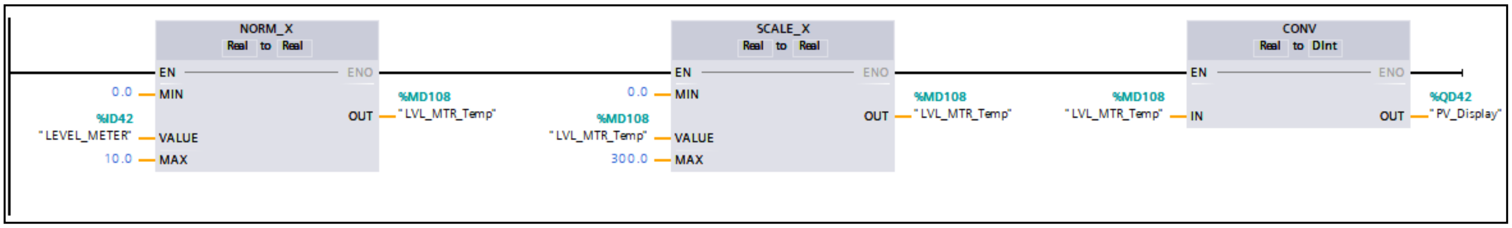

Figure XXVIIV.27 Process Values Network.

- Note:

We used CONV instruction to convert values from Real to Dint (Double integer) because our digital display hardware displays only integer values.

IV.4 Process Level Control's Design

IV.4.1 Tank scene design

- Using Factory i/o software we will simulate a process level control of a water tank and design its panel control.

- Starting with a new empty scene we add a tank water process from the stations list option.

- Then add the panel control and its accessory such as (Push Button, Potentiometer, ESD Button, Alarm Siren, Digital Display…)

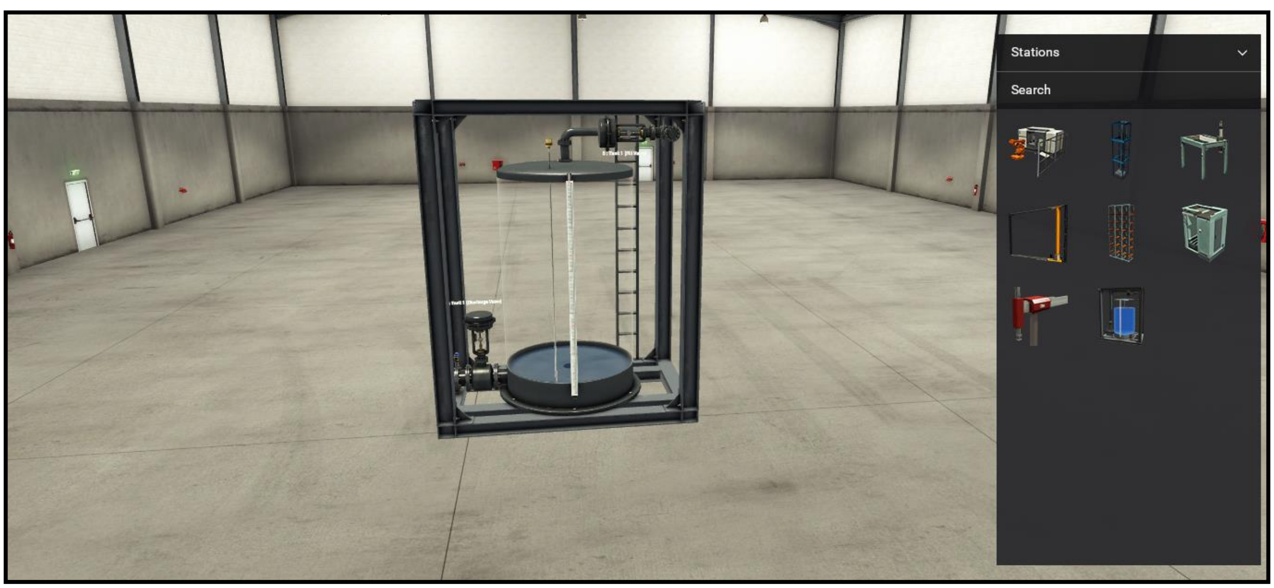

Figure XXVIIIV.28 Process Level Control.

Figure XXIXV.29 Tank Scene Design.

Figure XXXV.30 Tank Panel Control.

- After that we set inputs/outputs variables to the real hardware in our simulation software by selecting the **"DRIVERS"** option.

IV.5 Connect & Commissioning

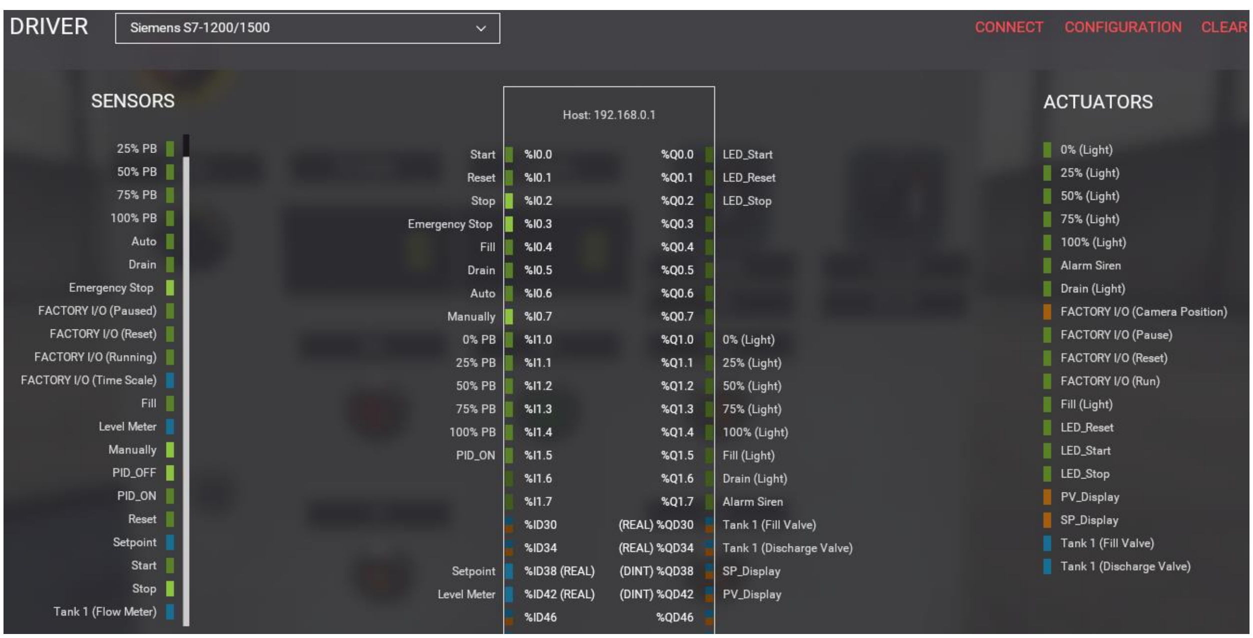

Figure XXXIV.31 Input/Output Driver Design.

- Finally, we can connect our project and start commissioning the project by selecting the right driver from the list **"Siemens S7-1200/1500"** then click **"CONNECT".**

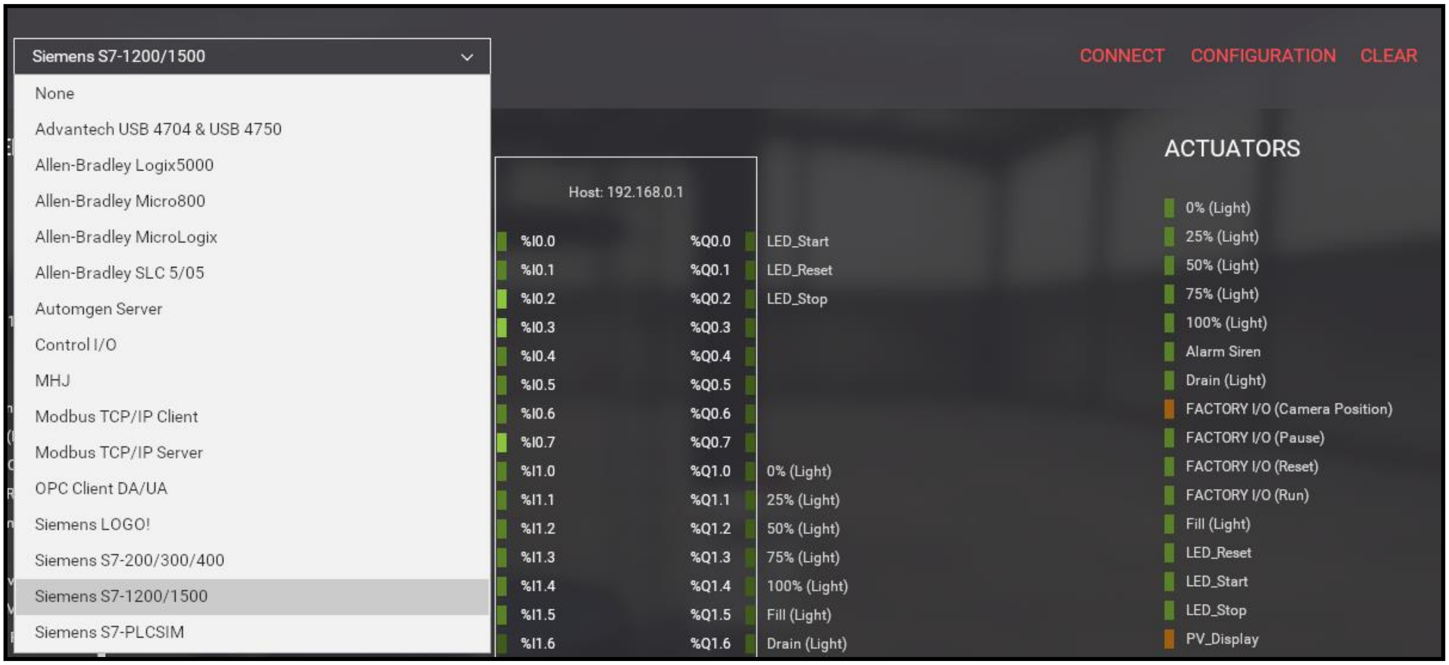

Figure XXXIIV.32 Siemens S7-1200/S7-1500 Driver Selection.

IV.6 Automate Process Level Control from HMI

In order to automate our tank process level control using "Human Machine Interface" we will manage to create for each control mode a specific screen and a standard template for the system buttons such as Start, Stop, Reset and ESD buttons and because are so crucial it will be design in a way that we will have access to it from any mode screen.

IV.6.1 HMI Tags & PLC Tags

We add the tags we need to the **"HMI tags"** and make sure that are connected with PLC tags, and most important, the acquisition cycle must be set to 100ms unless we will have a respond time issue.

PLC_IMSI ▸ HMI_TP900 [TP900 Comfort] ▸ HMI tags

HMI tags | System tags

HMI tags

Name	Tag table	Data type	Connection	PLC name	PLC tag	Address	Access mode
Auto Mode	Default tag table	Bool	HMI_Connection_1	PLC_1	"Auto Mode"		<symbolic access>
Auto_Drain	Default tag table	Bool	HMI_Connection_1	PLC_1	Auto_Drain		<symbolic access>
Auto_Fill	Default tag table	Bool	HMI_Connection_1	PLC_1	Auto_Fill		<symbolic access>
Drain_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	Drain_PB		<symbolic access>
E-Stop Activated	Default tag table	Bool	HMI_Connection_1	PLC_1	"E-Stop Activated"		<symbolic access>
Fill_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	Fill_PB		<symbolic access>
FILL_valve	Default tag table	Real	HMI_Connection_1	PLC_1	FILL_valve		<symbolic access>
HMI_0%_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_0%_PB		<symbolic access>
HMI_100%_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_100%_PB		<symbolic access>
HMI_25%_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_25%_PB		<symbolic access>
HMI_50%_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_50%_PB		<symbolic access>
HMI_75%_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_75%_PB		<symbolic access>
HMI_Auto	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_Auto		<symbolic access>
HMI_auto_Discharge_valve	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_auto_Drain_valve		<symbolic access>
HMI_Auto_Drain_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_Auto_Drain_PB		<symbolic access>
HMI_Auto_Fill_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_Auto_Fill_PB		<symbolic access>
HMI_auto_Fill_valve	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_auto_Fill_valve		<symbolic access>
HMI_Drain_PID	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_Drain_PID		<symbolic access>
HMI_Fill_PID	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_Fill_PID		<symbolic access>
HMI_manual_Discharge_valve	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_manual_Discharge_...		<symbolic access>
HMI_manual_Drain_PB	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_manual_Drain_PB		<symbolic access>
HMI_manual_Fill_BP	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_manual_Fill_PB		<symbolic access>
HMI_manual_Fill_valve	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_manual_Fill_valve		<symbolic access>
HMI_PID_Activation	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_PID_Activation		<symbolic access>
HMI_SP display	Default tag table	Bool	HMI_Connection_1	PLC_1	"HMI_SP display"		<symbolic access>
HMI_START	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_START		<symbolic access>
HMI_STOP	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_STOP		<symbolic access>
LED_Fill	Default tag table	Bool	HMI_Connection_1	PLC_1	LED_Fill		<symbolic access>
Manual Mode	Default tag table	Bool	HMI_Connection_1	PLC_1	"Manual Mode"		<symbolic access>
PID_Activation	Default tag table	Bool	HMI_Connection_1	PLC_1	PID_Activation		<symbolic access>
PV_Display	Default tag table	DInt	HMI_Connection_1	PLC_1	PV_Display		<symbolic access>
Reset Mode	Default tag table	Bool	HMI_Connection_1	PLC_1	"Reset Mode"		<symbolic access>
Screen_displayed	Default tag table	UInt	<Internal tag>		<Undefined>		
Screen_selection	Default tag table	UInt	<Internal tag>		<Undefined>		
SETPOINT_Temp	Default tag table	Real	HMI_Connection_1	PLC_1	SETPOINT_Temp		<symbolic access>
SP_Display	Default tag table	DInt	HMI_Connection_1	PLC_1	SP_Display		<symbolic access>
<Add new>							

Figure XXXIIIV.33 PLC-HMI Tags Connection.

IV.6.2 Control Mode Screens

1. **Creation**

By expanding the screens file in our HMI_TP900 comfort project arborescent we can add our control mode screens (HOME, SemiAuto_Fill&Drain, Manual_Fill&Drain, PID_controller.)

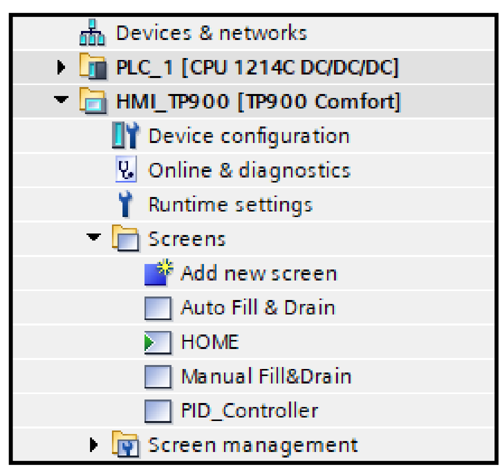

Figure XXXIVV.34 HMI Project Tree.

2. Design the screens

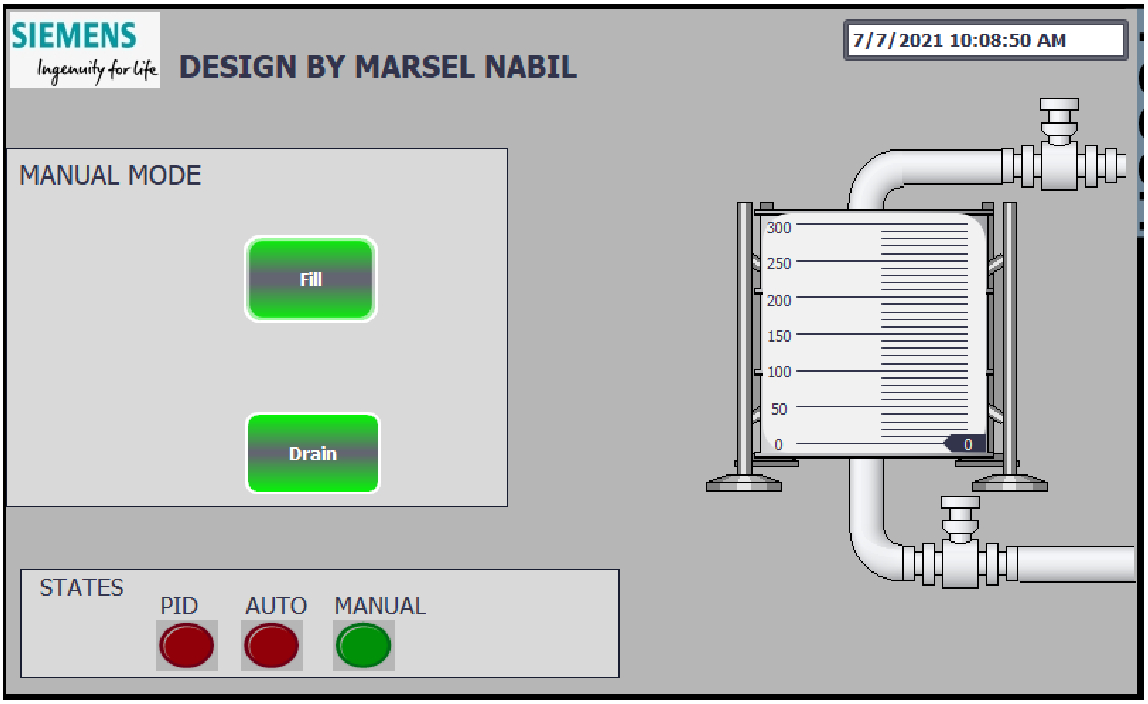

Figure XXXVV.35 Manual Mode Screen.

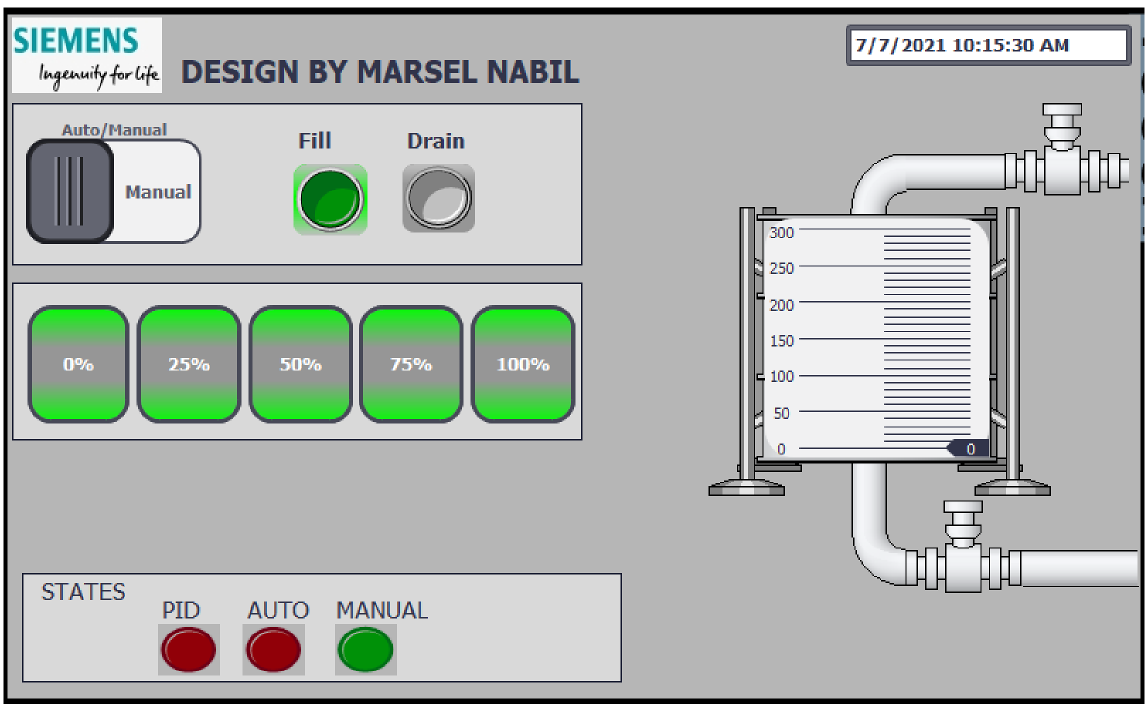

Figure XXXVIV.36 Semi Automatic Mode Screen.

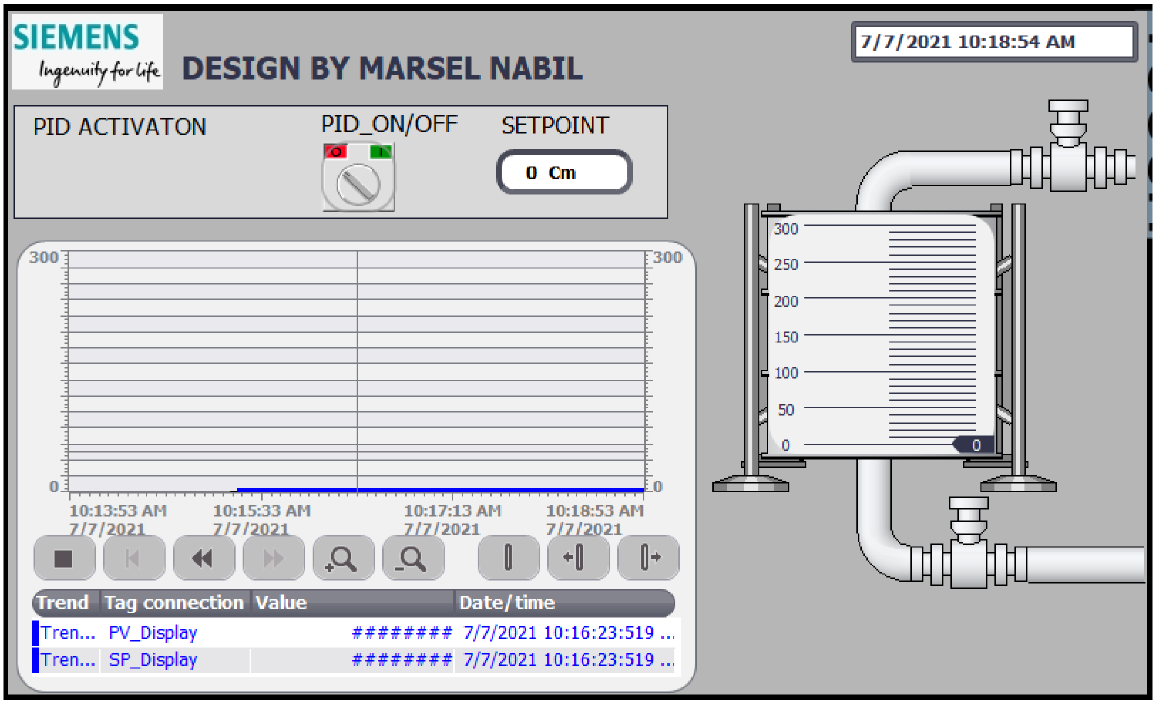

Figure XXXVIIV.37 PID Mode Screen.

IV.7 Realization of The Project

1. Connect the CPU module and expand modules on the S7-1200 rack using the **bus connector** on the right of the **SITOP** power supply.

Figure XXXVIIIV.38 Module's Bus Connector.

2. Use copper wires to supply the CPU module and the Analog Signal modules from the power supply 24 VDC.

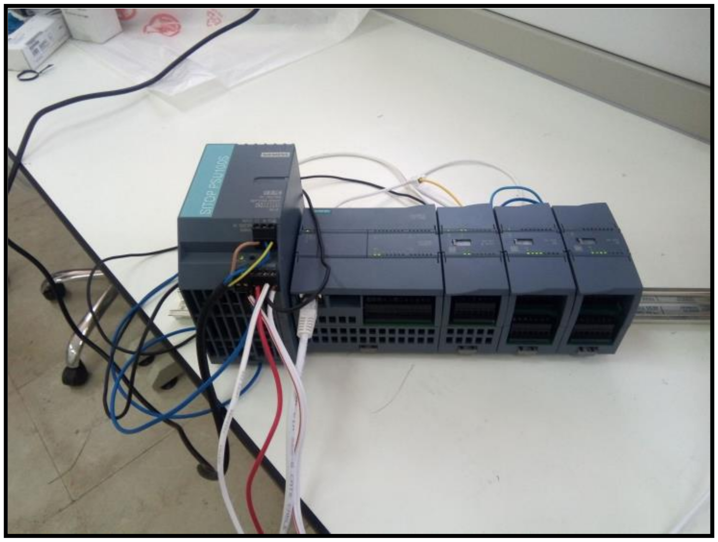

Figure XXXIXV.39 Project PLC Station.

3. Supply the TP900 comfort panel the Scalance X208 switch with 24 VDC, then create the profinet network using Ethernet cable port RJ45.
4. Run the 3D simulator and download the project Software/Hardware configuration on the hardware.

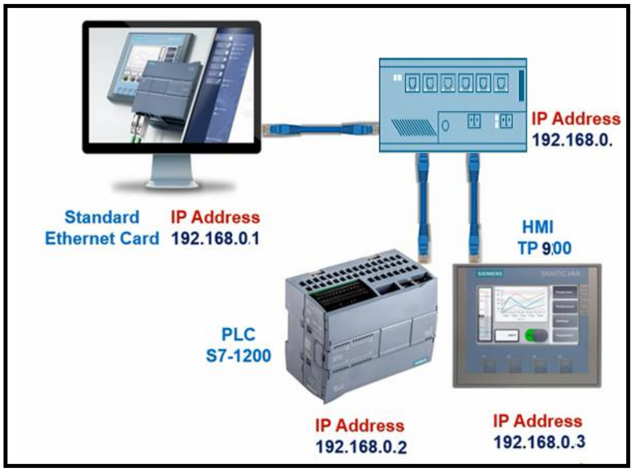

Figure XLV.40 Project Network Scheme.

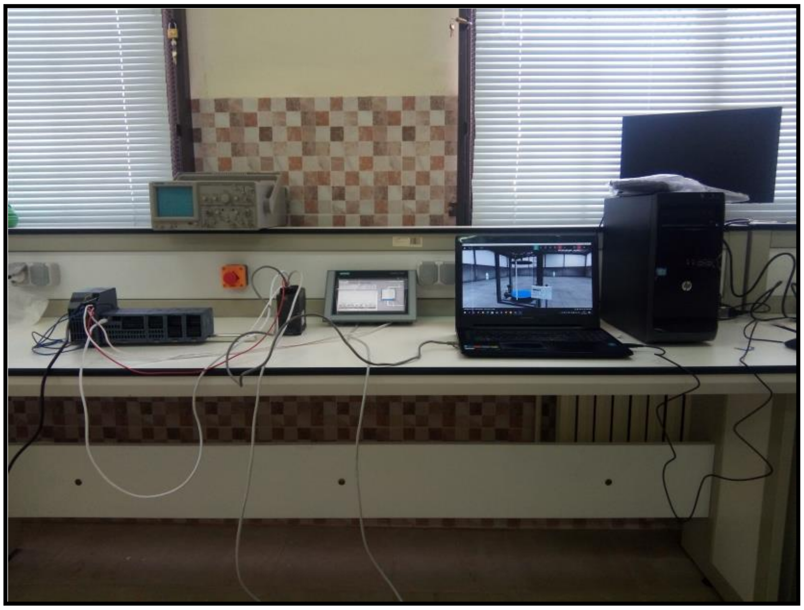

Figure XLIV.41 Project Hardware Connection.

IV.8 Conclusion

In this chapter, we have presented our practical realization starting with the Siemens s7-1200 and the TP900 comfort HMI going through programming in “LADDER” language and “winCC flexible” supervision with TIA PORTAL.

We then explained the commissioning and verification and finally we have given the corresponding goal we have expecting which is "automating the process level control.

Conclusion & Perspective

We have came to the conclusion that automation has been attributed to enhance and increase higher rates and productivity, more efficient use of materials, better product quality, reduced factory lead times and more important improved safety.

And because higher output and increased productivity have been two of the biggest reasons in justifying the use of automation but despite the claims of high quality from good workmanship by humans, automated systems typically perform the manufacturing process with less variability than human workers, resulting in greater control and consistency of product quality. Also, increased process control makes more efficient use of materials, resulting in less scrap.

In return back to our process, process level control of tank water, we have seen that using manually mode required more focus and skill to reach the desire setpoint, and that was the case before and that the reason human start looking for another solution, better and efficient one, so that got us to the semi automatic mode which is basically depends on relays and contactor, because like we saw in the fourth chapter, the auto fill and drain mode to be specific, we moved a zero volt to the fill or drain valve when the level meter approach the desire setpoint, and that because of the left water in the inflow pipe, so technically we still in the second revolution industry here and that why we called it Semiautomatic.

The automatic mode or PID compact mode, here we are in the next era, the automation era which we have seen how simply it is to reach the desire setpoint by simply command from the central room using human interface machine or locally with the control panel (potentiometer) and that due to the PID compact instruction integrated within the powerful S7-1200 PLC using the Siemens solution for programming programmable logic controls Tia portal and WinCC flexible for programming TP900 touch panel HMI.

Bibliography

[1] https://www.cybiant.com/resources/an-introduction-to-automation/,10-07-2021

[2] https://www.elprocus.com/an-overview-on-industrial-automation/

[3] https://realpars.com/automation-pyramid/

[4] https://www.omega.co.uk/prodinfo/pid-controllers.html

[5] https://www.omega.co.uk/prodinfo/how-does-a-pid-controller-work.html

[6] https://www.omega.co.uk/prodinfo/pid-controller-types.html

[7] https://realpars.com/pid-loop/

[8] SIMATIC TIA Portal STEP 7 Basic V10.5 Getting Started, 12/2009

[9] "STEP 7 and WinCC Engineering V16 System Manual" Introduction to the TIA Portal_Programming and Operating Manual, 11/2019

[10] Programming Guideline for S7-1200/S7-1500

[11] SIMATIC Programming with STEP 7 Manual

[12] https://www.mhj-tools.com/?page=factory-io

[13] https://docs.factoryio.com/installing/

[14] https://docs.factoryio.com/getting-started/

[15] https://docs.factoryio.com/getting-started/navigating/

[16] https://docs.factoryio.com/getting-started/opening-a-scene/

[17] https://docs.factoryio.com/getting-started/creating-a-scene/

[18] https://docs.factoryio.com/getting-started/controlling-with-a-plc/

[20] "Kelvin T. Erickson" Programmable Logic Controllers: An Emphasis on Design and Application Second Edition

[21] W. Bolton,"Programmable Logic Controllers" Newne, Fourth édition

[22] Hans Berger,"Automating with SIMATIC S7-1500 Configuring, Programming and Testing with STEP 7 Professional"

[23] S7-200 Programmable Controller System Manual

[24] S7-300 Module data, Manual

[25] https://forumautomation.com/t/what-are-the-models-of-siemens-plc/8961, 09-07-2021

[26] LOGO! Manual Edition 06/2003

[27] Hans Berger,"Automating with SIMATIC S7-1500 Configuring, Programming and Testing with STEP 7 Professional"

[28] SIMATIC S7-1200 Easy Book Manual

[29] https://support.industry.siemens.com/cs/mdm/91696622?c=44253821835&lc=en-AF 09-07-2021

[30] https://new.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html, 09-07-2021

[31] SIMATIC HMI, Multi Panel, MP 370 09/02

[32] https://new.siemens.com/global/en/products/automation/simatic-hmi/panels/basic-panels.html

[33] https://www.instrumart.com/categories/4140/differential-pressure-flow-meters 10-07-2021

[34] https://automation-insights.blog/2017/06/07/what-is-a-capacitive-sensor/