

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA MINISTRY OF  
HIGHER EDUCATION AND SCIENTIFIC RESEARCH

**University KASDI MERBAH OUARGLA**

Faculty of New Information and Communication Technologies Department of  
Electronics and Telecommunications



**Theses**

**Professional Master's**

Domain: Science and Technology

Sector: Electronics

Specialty: Instrumentation and systems

Presented by:

KAFI Ahmed Rami

BELAID Seyf ALLAH

**Theme:**

**Level and flow cascade control using  
Programmable Logic Controller “Siemens”**

**Before the jury:**

Dr GUMOUHE Samia	MCB	President	UKM Ouargla
Dr HAMMOUCHI Fateh	MCA	Examinator	UKM Ouargla
Dr CHAIB Ibtissam	MCB	Advisor	UKM Ouargla

**Academic year: 2024 / 2025**

# Thanks

We want to express our sincere gratitude to our respected supervisor, Miss Dr << **CHAIB Ibtissam** >>, for her invaluable guidance, continuous support, and encouragement throughout our thesis. Her expertise, patience, and insightful feedback played a crucial role in shaping the quality and direction of our work.

We also extend our heartfelt thanks to all the teachers who have taught us during our university journey. Your dedication, knowledge, and mentorship have greatly contributed to both our academic and personal development. Each of you has played a part in helping us reach this milestone, and for that, we are deeply grateful.

Thank you all for your support and contributions.

# Dedication

I dedicate this work to my beloved family, whose unconditional love and support have been my greatest strength throughout this journey.

To my late father, **KAFI Abdelmoutalib**, whose memory lives on in my heart, your wisdom, guidance, and sacrifices have shaped me. Though you are no longer here, your spirit continues to inspire me every step of the way.

To my dear mother, **DJERBAOUI Nacira**, who has stood by me through every challenge with endless patience and love. Your tireless efforts and belief in me have been the foundation that helped me overcome all obstacles and reach this important milestone.

To my siblings and extended family, thank you for your encouragement and understanding.

Your faith in me has been a constant source of motivation.

This achievement reflects our shared resilience and love. I am forever grateful to have such a wonderful family.

**KAFI Ahmed Rami**

# Dedication

This work is lovingly dedicated to my family, whose unwavering support and encouragement have been the foundation of my success.

To my mother, **MESSAK. Saida**,

Your endless love and sacrifices have been the source of my strength and motivation. Your belief in me has been a beacon of hope during difficult times.

To my wife, **Messaouda. ABIDL**,

Thank you for your constant encouragement, patience, and understanding. Your support has been invaluable in helping me stay focused and committed to this journey.

I am forever grateful to have such a caring and inspiring family by my side.

BELAID Seyf ALLAH

## Summary:

Thanks .....	I
Dedication .....	II
Dedication .....	III
Summary .....	IV
List of tables .....	VII
List of Figures .....	VIII
Abbreviation .....	X

### General introduction

General introduction.....	1
---------------------------	---

### Chapter I Overview of Regulation

I.1.Introduction.....	3
I.2.Regulation and control.....	3
I.2.1 Regulation Loop.....	3
I.2.2 Loop types.....	3
I.3. Performance criteria for a regulation system.....	4
I.3.1 Stability.....	4
I.3.2 Precision.....	5
I.3.3 Speed.....	5
I.4. Regulation in Flow and Level Control Using Cascade Control.....	6
I.4.1 Proportional action (P).....	6
I.4.2 Integral action regulator (I).....	7
I.5 Cascade Regulation.....	9
I.5.1.2 Working Principle of Cascade Control.....	9
I.5.1.3 Necessary Conditions for Cascade Control.....	10
I.5.2 Industrial Example of Cascade Control (Flow and Level) .....	10
I.6. Comparison with Simple Control.....	12
I.7. PID regulator.....	12
I.7.1 Definition.....	12
I.7.2 Function of the PID Regulator in Cascade Flow and Level Control.....	13
I.8. Conclusion.....	14

## Chapter II General Overview of Programmable Logic Controller

II.1. Introduction.....	15
II.2. General Information about PLCs.....	15
II.2.1. History.....	15
II.2.2. Definition of a PLC.....	15
II.3. Architecture of PLCs [1].....	16
II.3.1. External Appearance.....	16
II.3.2. Internal Structure .....	17
II.3.3 Graphical Languages .....	18
II.4. Description of the S7-315 Programmable Controller.....	19
II.5. Conclusion .....	22

## Chapter III Programming and Monitoring in TIA Portal and Simulation with Factory I/O

III.1. Introduction.....	23
III .2. Cascade level control system with flow .....	23
III .2.1. Description project .....	24
III.2.2 System Overview .....	24
III.3. Programing the cascade control system .....	25
III.3.1 Introduction to the TIA Portal .....	26
III .3.2. Advantages of working with the TIA Portal: .....	27
III.4 project View .....	27
III.5 Programming and Design of Cascade PID Control with TIA Portal .....	28
III.5.1. Project Creation .....	28
III.5.2. Inserting and Configuring a PLC .....	28
III.5.3. Signal Modules for PLC 315-2 PN/DP and SM334 .....	30
III.5.4. Creation of the Variable Table .....	32
III.5.5. Creation of a Global Data Block .....	32
III.5.6. Creation of the Organization Block “OB1” .....	33
III.5.6.1. User Program .....	33
III.5.6.2 Creation of the Organization Block for the PID Controller .....	34
III.5.6.3 Principle of Regulation .....	35
III.5.7 Continuous Control with the "CONT_C" Block.....	35
III.5.8 PID Configuration .....	36
III.6 . Simulation using PLCSIM and Factory IO .....	39
III.7. Factory I/O .....	40
III.7.1 Overview .....	40

III.7.2 Overview of project stimulation .....	41
III.8. Case Study .....	43
III.8.1 Experiment Procedure .....	43
III.8.2. Trend Analysis: .....	44
III.8.3. Observations: .....	44
III.8.4. Trend Analysis: .....	45
III.8.5. Observations: .....	45
III.9. Conclusion .....	46
General conclusion .....	47
Resume .....	49
Bibliography.....	50

## List of tables:

<b>Table I.1:</b> PID control actions overview .....	13
<b>Table II.1:</b> Position of the Operating Mode Switch .....	20
<b>Table II.2</b> Technical Spécifications – Siemens CPU 315-2 PN/DP .....	21
<b>Table III.1.</b> General Characteristics of PLC S7-300 CPU 315-2 PN/DP .....	30
<b>Table III.2.</b> General Characteristics of SM 334 (Analog Input/output Module) .....	31
<b>Table III.3:</b> parameters input/output of the Master PID block.....	37
<b>Table III.3:</b> parameters input/output of the Slave PID block.....	38

## List of Figures:

Figure I.1: Schematic diagram of a regulation loop .....	3
Figure I.2: open loop .....	4
Figure I. 3: closed loop.....	4
Figure I.4 Index response of a stable system.....	5
Figure I.5: Presents the system error curve .....	5
Figure I.6: The oscillatory damping response .....	6
Figure I. 7: proportional action .....	7
Figure I.8: Integral action regulator .....	9
Figure I.9: Derivation action regulator.....	9
Figure I.10: cascade loop .....	10
Figure I.11: PID regulator .....	12
Figure II .1 : Automate compact.....	16
Figure II .2: Automate modulaire (Modicon.....	16
Figure II.3: Modular PLC (Siemens).....	17
Figure II.4: Internal Structure of a PLC.....	17
Figure II .5: Programming Languages .....	18
Figure II.6: Plc S7- 300-315 2DP N.....	19
Figure. III.1: Circuit related to cascade level control with flow.	23
Figure III.2. TIA Portal.....	26
Figure III.3. Project View.....	27
Figure III.4. Project Creation.....	28

Figure III.5. Inserting and Configuring a PLC (1).....	29
Figure III.6. Inserting and Configuring a PLC (2).....	29
Figure III.7. Configuration of SM 334 (Analog Module).....	31
Figure III.8. Creation of the Variable Table.....	32
Figure III.9. Creation of a Global Data Block.....	33
Figure III.10. cyclic OB .....	34
Figure III.11. Creation of the Block PID Controller.....	34
Figure III.12: PID block.....	35
Figure III.13: Principle Diagram of the PID Block.....	36
Figure III.14. The master block Configuration.....	37
Figure III.15. The Slave Block Configuration.....	38
Figure III.16. save and compile.....	39
Figure III.17. download and simulation.....	39
Figure III.18. extended download.....	39
Figure III.19. load preview.....	40
Figure III.20. start plcsim.....	40
Figure III.21. Factory I/O.....	41
Figure III.22. Overview project stimulation.....	41
Figure III. 23. Driver Men.....	42
Figure III.24. simulation view.....	42
Figure III.25. PID LEVEL Control.....	44
Figure III.26. PID FLOW Control.....	45
Figure III.27. simulation factory view.....	46

## Abbreviation

OL	open loop
Cl	close loop
PID	Proportional-Integral-Derivative control
Kp	Proportional gain
Ki	Integral gain
Kd	Derivative gain
PLC	Programmable Logic Controller
LD	Ladder Diagram
FBD	Function Block Diagram
ST	Structured Text
IL	Instruction List
HMI	Human-Machine Interface
DP/NP	Distributed Peripheral / Not Peripheral (in Siemens context)
OPC	OLE for Process Control (communication protocol)

## **General introduction**

In recent decades, industrial automation has become a vital cornerstone in developing and optimizing manufacturing systems. With the rise in demand for higher efficiency, precision, and reliability, the role of automated control systems has grown considerably. Among these systems, the Proportional-Integral-Derivative (PID) controller continues to be one of the most widely applied strategies in industrial process control due to its simplicity, robustness, and effectiveness across various applications. [1]

However, in systems where multiple interdependent variables must be controlled, especially those with slow and fast dynamic responses, a single PID loop may be insufficient to achieve optimal performance. In such cases, cascade control architecture provides a more effective solution. Cascade control involves a primary (master) controller and a secondary (slave) controller, where the output of the primary controller becomes the setpoint of the secondary controller. This structure allows for faster correction of disturbances and improved overall system performance. [2]

The objective of this thesis is to design, implement, and evaluate a cascade control system for a level-flow process using Programmable Logic Controllers (PLCs) and simulation tools. Specifically, we employ Siemens' TIA Portal for programming a S7-300 PLC, and Factory I/O for process simulation and visualization. The system under consideration consists of a water tank where level control serves as the primary loop and flow control as the secondary loop. The cascade configuration enables the flow loop to respond quickly to disturbances, thereby stabilizing the slower level loop. [2]

The study begins with a comprehensive theoretical background on control systems, PID control principles, and cascade control theory. Next, we introduce the architecture and operational principles of Siemens PLCs, the TIA Portal software environment, and Factory I/O. The practical part of the work focuses on the configuration and integration of hardware and software components, programming of control logic in ladder diagram (LAD), and real-time data exchange between the PLC and the virtual environment.

The implementation is validated through a series of simulation experiments designed to test the system's behavior under different conditions. These include setpoint changes, external disturbances, the results confirm that cascade control improves response time, reduces overshoot, and enhances disturbance rejection compared to traditional control strategies.

Our thesis is organized into three main chapters. The first chapter presents a theoretical overview of control systems, with a focus on PID control and the cascade regulation strategy, particularly in level and flow applications. It discusses the advantages of cascade control and its role in enhancing system performance. The second chapter introduces Programmable Logic Controllers (PLCs), especially the Siemens S7-300 series, detailing their architecture, programming languages, and integration into industrial environments. The third and final chapter covers the practical implementation of the cascade control system using TIA Portal for programming and Factory I/O for simulation. It includes the configuration of hardware, development of ladder logic programs, and simulation experiments that validate the performance and reliability of the designed system under various operating conditions

# **Chapter I**

## **Overview of Regulation**

## I.1.Introduction

Control and regulation are key concepts in engineering and industrial automation. A control system is designed to manage or adjust the behavior of another system to achieve desired performance, while regulation focuses on maintaining a specific variable (like temperature or pressure) at a set value despite disturbances. These systems are widely used in industries such as oil, gas, power, and manufacturing, often implemented using methods like PID control or cascade control. They help improve efficiency, safety, and product quality. [3]

## I.2.Regulation and control

In a regulation system, the goal is to keep a variable close to a constant setpoint despite external changes. Maintaining minimal deviations ensures stability. If the setpoint changes, the controlled variable must adjust smoothly. This type of control is known as servomechanism or servo control [3].

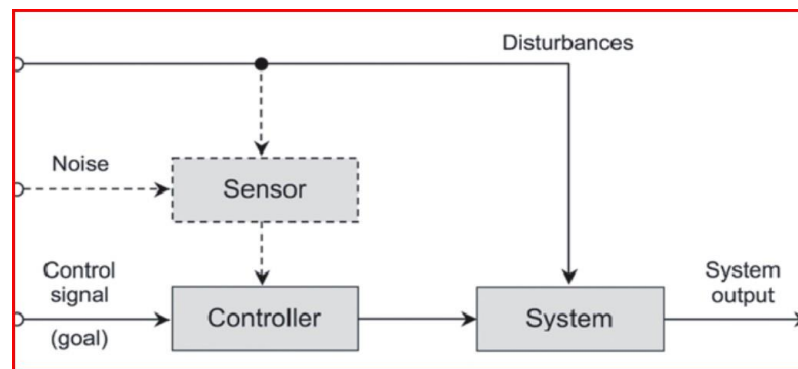


Figure I.1: Schematic diagram of a regulation loop

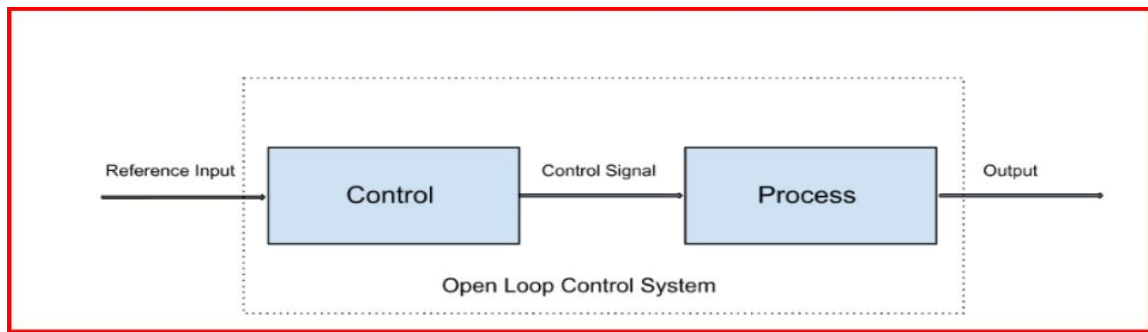
### I.2.1 Regulation Loop

The regulator gets a measurement signal from the sensor and a setpoint. It sends an output signal to the actuator to eliminate the difference. [3]

#### I 2.1.1 Loop types

##### Open loop (OL):

The figure shows an open loop. The magnitude of the correction is independent of the magnitude of the measurement. The regulator is in manual mode and allows the user to choose the value of the control signal applied to the actuator. [4]

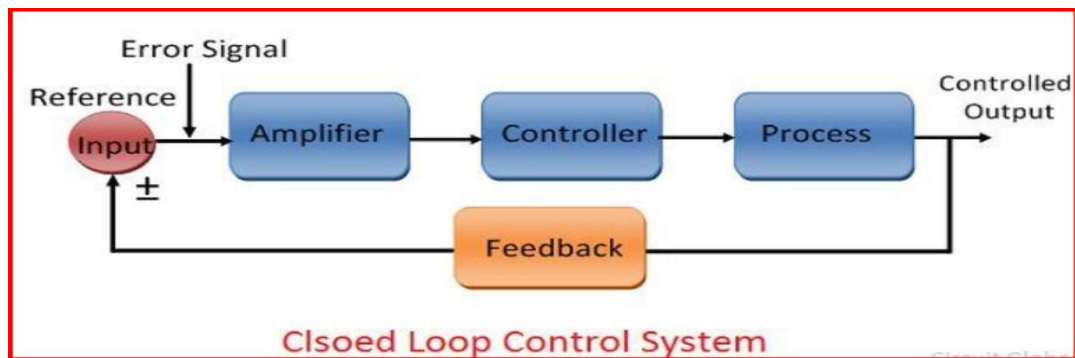


**Figure I.2: open loop**

### Close Loop (CL):

The closed-loop system measures and compares output to a set point. The error signal adjusts the input to keep the output near the target, ensuring accuracy and stability.

**Example:** A thermostat controls the temperature in a room by constantly reading the current temperature and comparing that to the ideal temperature. [4]



**Figure I. 3: close loop.**

## I.3. Performance criteria for a regulation system

For any industrial regulation project, the user's specifications represent the point of departure for an engineer's work. Within the variety of regulation issues, the following criteria can be taken into consideration:

### I.3.1 Stability:

Such a system, constituted by the process and the control loop, is said to be stable; if it is subjected to a setpoint variation, the measurement returns to a stable state; otherwise, it is unstable.

The transient regime is the time needed to regain stability for a stable system. .[6]

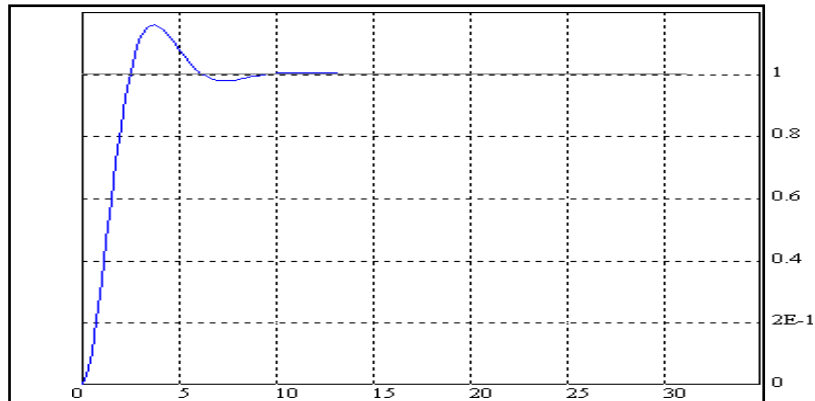


Figure I.4 Index response of a stable system.

### I.3.2 Precision:

The precision of a system, in relative value, regarding the setpoint  $c(t)$ , is defined by its error signal  $\varepsilon(t)$ . Depending on whether the system is in a static regime  $c(t) = \text{constant}$  (permanent regime) or in a dynamic regime,  $c(t) = f(t)$  (variable permanent regime), we speak of static precision or dynamic precision. For any system, we will then try to reduce this error [6].

$$\text{Precision error (\%)} = (\varepsilon/C) \cdot 100$$

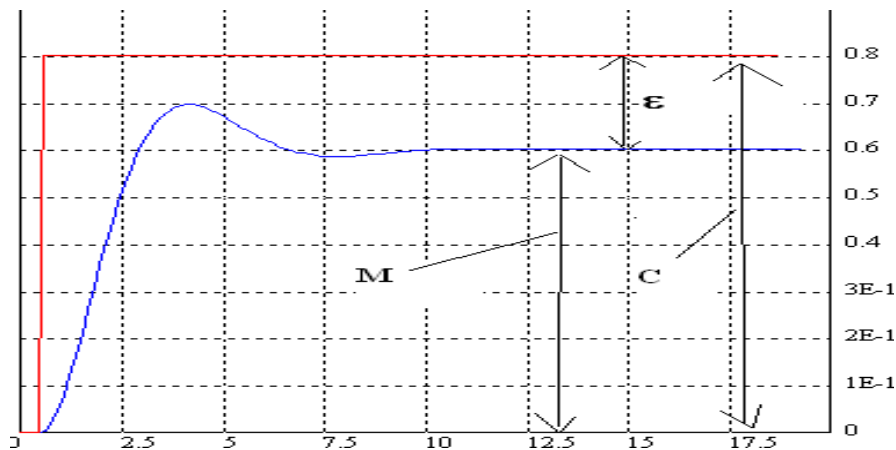


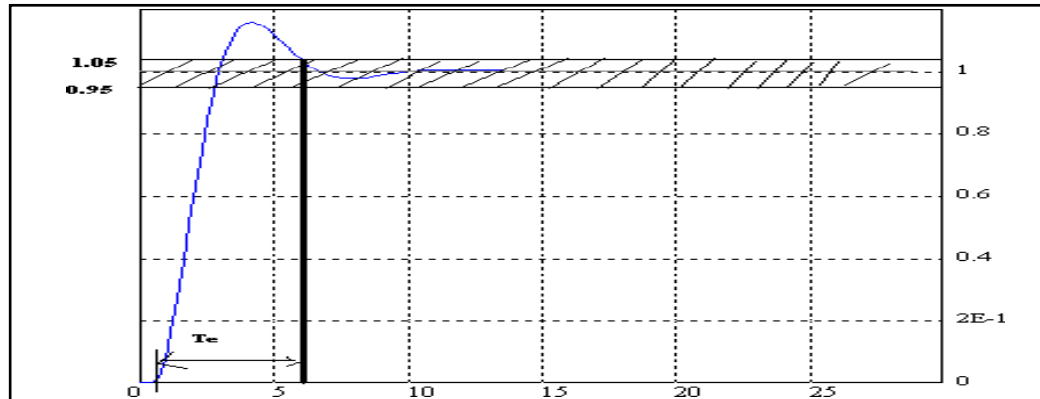
Figure I.5: Presents the system error curve

### I.3.3 Speed:

It practically translates the transitional duration. More precisely, it is expressed by the response time  $T_e$  or establishment time, which is the time taken by the measurement to reach its final value at +5% of its variation while remaining in this +5% zone.

Speed = response time  $T_e$

$T_e$ : corresponds to the time taken to reach 95% or 105% of the final value of  $\Delta M$  The following figure represents damped oscillatory responses[6]



**Figure I.6: The oscillatory damping response.**

#### I.4. Regulation in Flow and Level Control Using Cascade Control

Cascade control is a common strategy in process industries used to improve the accuracy and stability of complex systems. It uses two control loops: a primary (master) loop and a secondary (slave) loop. The primary loop manages the liquid level in a tank, while the secondary loop controls the inflow rate. The level controller gives a setpoint to the flow controller, which adjusts the control valve. This setup allows the secondary loop to quickly respond to disturbances, making the control system more stable and responsive. Cascade control is important in industries like oil and gas, petrochemicals, and water treatment, where precise level regulation is essential.

##### I.4.1 Proportional action (P)

proportional action if the control signal is proportional to the error signal, corrects any deviation instantly, hence rapidly, of the system. In order to describe the adjustment difference and to make the system faster, we increase the gain (we reduce the proportional band), but we are limited by the stability of the system.

The P regulator is used when you want to adjust a parameter whose precision is not important.

The equation for the control  $u$  of the proportional action P is as follows:

$$u(t) = K_p e(t) = K_p (y_c(t) - y(t)) \quad (1.1)$$

And the transfer function of the proportional action P is:  $C(s) = Kp$

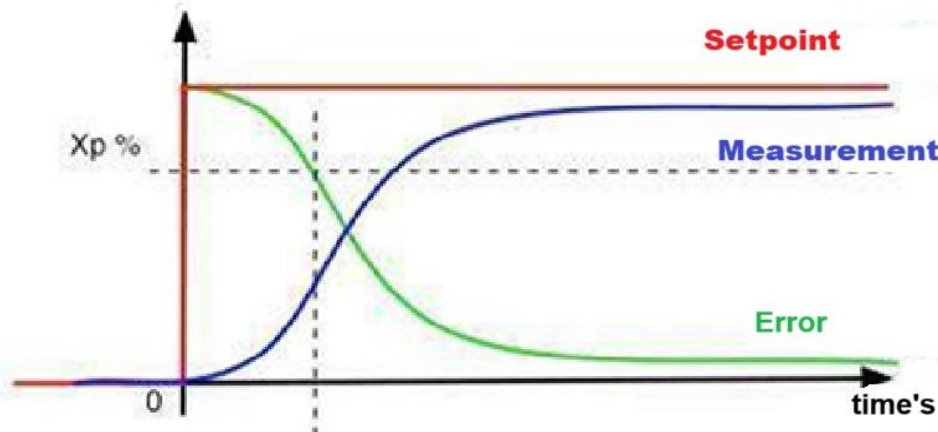


Figure I. 7: proportional action .

#### I.4.2 Integral action regulator (I)

An integral type controller completely eliminates the constant regulation error. As long as the error is not equal to zero, the value of the controlled variable is stabilized. The regulation stops when the output has reached the setpoint value or the regulated variable has reached a maximum threshold set by the system properties ( $U_{max}$ ,  $P_{max}$ , etc.).[6]

The mathematical statement of this integral action is that the controlled variable is proportional to the integral with respect to time of the error.

$$Y(t) = K_i \int e(t) dt$$

$$K_i = 1/T_i$$

The time constant  $T_i$ , often expressed in units of time, is referred to as the integration time constant.

The rate at which the controlled variable increases (or decreases) is a function of the error of regulation and the chosen integration time constant. The effect of  $T_i$  on the system is shown in the figure.

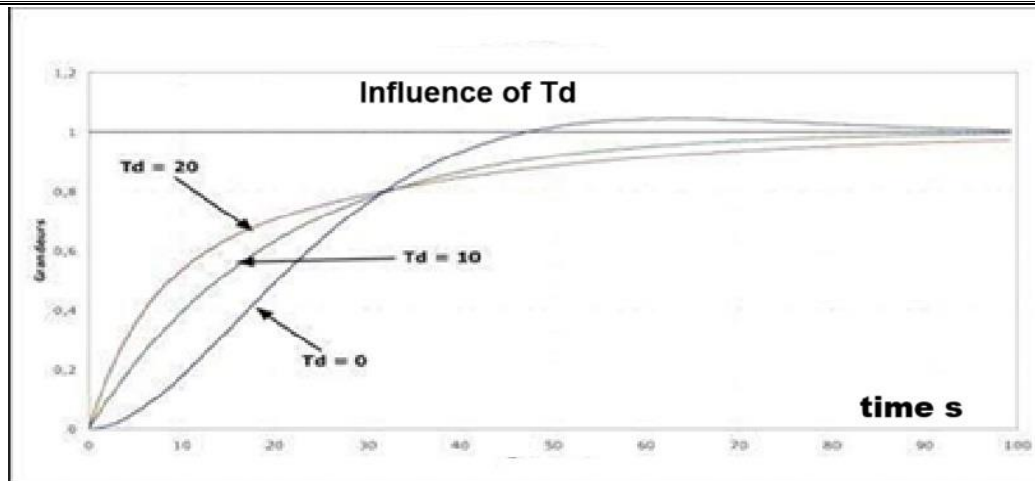


Figure I.8: Integral action regulator.

### I.4.3 Pure bypass regulator (D)

The derivative (D) regulator responds to the rate of change of the error rather than its magnitude, making it faster than a proportional (P) regulator. It can produce a strong reaction even for small errors if there's a rapid change. However, it becomes inactive when the error is constant, as there's no variation to respond to. For this reason, a D regulator is never used alone and is typically combined with a P regulator. [6]

The transfer function of a derived corrector is  $C(p)=T_d.P$  (I.13). The time constant  $T_d$ , of dimension the unit of time, is called the derivative time constant. It also acts as a gain.

This type of corrector is purely theoretical; physically, a system can't have a numerator of degree greater than that of the denominator. The effect of  $T_d$  on the system is shown in the next figure.

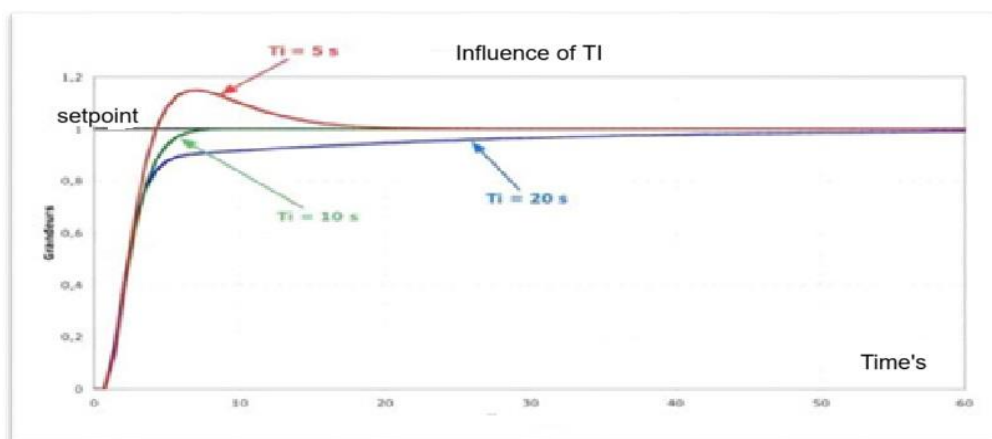


Figure I.9: Derivation action regulator

## I.5 Cascade Regulation

### I.5.1 Cascade Control in Process Automation

#### I.5.1.1 Cascade Control System Overview

Cascade control is an advanced process control strategy designed to enhance the stability and performance of complex systems. It involves the use of two interconnected control loops: a primary (master) loop that regulates the main process variable, and a secondary (slave) loop that controls an intermediate variable with a faster response.

The master controller sends its output as the setpoint for the slave controller, allowing the secondary loop to respond rapidly to disturbances before they significantly affect the primary variable.

This structure is particularly effective in systems with high inertia or those subject to fast disturbances, such as heat exchangers, industrial furnaces, or hydraulic systems. By anticipating changes and correcting them early, cascade control reduces steady-state error, improves response time, and increases the overall robustness and efficiency of the process.

[8]

#### I.5.1.2 Working Principle of Cascade Control:

control loops arranged in a hierarchical structure. The primary (outer) loop is responsible for maintaining the main process variable, such as temperature or level, at its desired setpoint.

It does this by continuously comparing the actual value of the process variable with the setpoint and sending a corrective signal—not directly to the final control element, but as a setpoint to the secondary (inner) loop.

The secondary controller manages an intermediate process variable, such as flow or pressure, which has a faster dynamic response and directly influences the primary variable. This inner loop receives its own measurement from a dedicated sensor and acts rapidly to keep the secondary variable at the setpoint received from the primary controller. As a result, any disturbances affecting the secondary variable—such as fluctuations in pressure or supply conditions—can be corrected immediately by the inner loop before they propagate and affect the main process. This hierarchical feedback structure enables faster disturbance rejection, improved control accuracy, and greater process stability, particularly in systems where the

primary variable responds slowly or is sensitive to upstream changes. The efficiency of cascade control relies on proper tuning and ensuring that the secondary loop operates significantly faster than the primary loop [10]

The Cascade Structure is a Secondary Loop Inside a Primary Loop

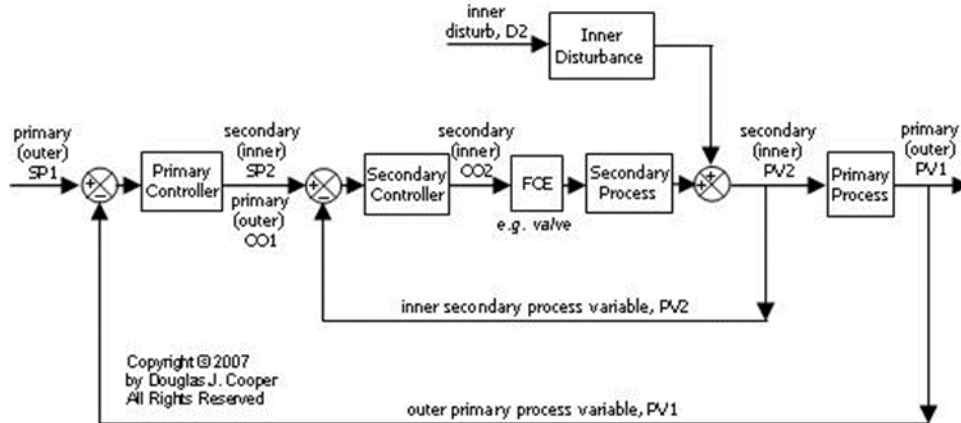


Figure I.10: cascade loop

### I.5.1.3 Necessary Conditions for Cascade Control:

For a cascade control system to work well, several conditions are important. First, the secondary variable must react faster than the primary variable to manage disturbances effectively. For example, in a furnace, the heating fluid flow should be adjusted quickly to keep the temperature stable. Second, the secondary variable needs to be measurable and influenced by the primary process, allowing for real-time corrections to enhance system performance. Third, both the primary and secondary controllers must be tuned carefully to ensure quick responses and stability. Lastly, a clear cause-and-effect relationship between the secondary and primary variables is essential for effective control. [5]

### I.5.2 Industrial Example of Cascade Control (Flow and Level):

An industrial example of a cascade control system can be seen in the regulation of liquid level and flow rate in a tank system. In many industrial processes, such as chemical reactors, water treatment plants, or fuel storage, maintaining a constant liquid level is crucial for proper operation. However, controlling the liquid level directly can be challenging due to fluctuations in input flow rates or disturbances in the process.

---

In this case, a cascade control system can be applied to improve the system's performance by using two control loops: a primary loop that controls the liquid level and a secondary loop that controls the flow rate

### **I.5.2.1 Primary (Master) Loop: Levels Control:**

The primary controller is responsible for maintaining the liquid **level** in the tank. The desired setpoint for the liquid level is set by the operator based on the system's requirements. The level of the liquid is measured by a level sensor, and the primary controller compares this value with the setpoint. The controller then generates an output, which becomes the setpoint for the secondary (flow) controller.

### **I.5.2.2 Secondary (Slave) Loop: Flow Control:**

The secondary loop controls the flow rate of the liquid into the tank. The flow control loop uses a flow sensor to measure the actual flow rate, comparing it with the setpoint from the primary controller. Based on this comparison, the secondary controller adjusts the flow control valve to maintain the desired flow rate. The secondary controller's quick response helps to ensure that any disturbances in the flow rate, such as fluctuations in the supply line pressure or pump speed, are corrected rapidly before they affect the liquid level.

### **I.5.2.3 Work principal**

- Disturbances: If there is a sudden increase in the inflow rate (e.g., due to pump speed fluctuations or changes in supply pressure), the secondary controller immediately adjusts the flow control valve to bring the flow rate back to the setpoint, preventing the tank level from rising too high.
- If the tank level drops too low, the primary controller sends a signal to the secondary controller to increase the flow rate to restore the level to the desired setpoint.
- The cascade structure ensures that the flow rate can be adjusted quickly without waiting for the slower primary loop to react, thereby maintaining a stable level in the tank.

### **I 5.2.4 Advantages of Cascade Control in This Example**

By using cascade control, the system can handle rapid disturbances in the flow rate more effectively. The secondary loop's fast response helps to prevent large deviations in the liquid level, which would be more difficult to manage with a single feedback loop. Additionally,

cascade control allows for more precise control of both variables, as each loop is optimized for its specific task: the secondary loop for flow and the primary loop for level. [12]

## I.6. Comparison with Simple Control

Cascade control and simple control are two methods for managing industrial processes, each with distinct features. Simple control uses one feedback loop, where a single controller keeps the process variable at a setpoint by adjusting control elements, like valves or pumps. It is easy to implement but may struggle with slow dynamics or disturbances. On the other hand, cascade control involves multiple interconnected loops, with a primary controller managing the main process and a secondary controller handling an intermediate variable. The primary controller's output sets the secondary controller's setpoint, allowing quicker responses to disturbances. While cascade control improves response time and regulation, it is more complex and requires careful tuning of controllers. Simple control is easier to implement, fitting systems with less complexity and disturbances, especially when fast response is not essential. [13]

## I.7. PID regulator

### I.7.1 Definition

**PID regulator (Proportional, Integral, Derivative)** This is a self-regulation system (closed loop), which seeks to reduce the error between the setpoint and the measurement

The aim of regulation is to maintain at a desired value (REFERENCE QUANTITY), a physical quantity (REGULATED QUANTITY) such as temperature, relative humidity,

$$\text{Error} = \text{setpoint} - \text{measure}$$

pressure, etc., subject to disturbances by measuring its value. After comparison between the regulated quantity and the reference quantity, a difference in adjustment results. Depending on this difference, the regulator forms a control signal (adjustment quantity) which will vary the adjustment power via an actuator (adjustment devices) [4].

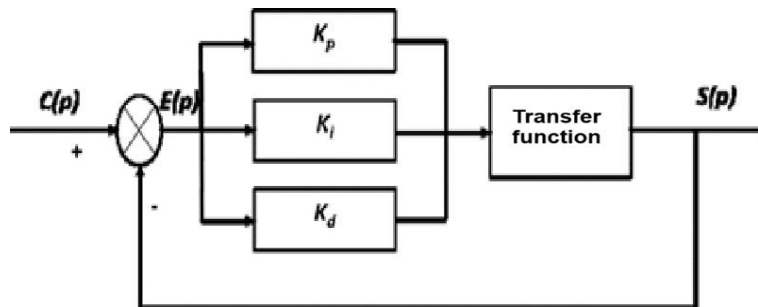


Figure I.11: PID regulator

In the form of a summary table, we summarize the advantages and influence of the basic actions of PID regulators.

**Table I.1: PID control actions overview**

Action	Proportional	Integrators	Derivative
Strengths	Instant action	Cancels the static error	Highly Dynamic action improves speed
Weaknesses	Does not a static error but allows it to be reduced	Slow acting slows down the system (destabilizing effect)	Noise sensitivity high load on the control unit
Stability	Decreases	increases	Decreases
Precision	increases	No influence	No influence
speed	increases	Decreases	Increases

## I.7.2 Function of the PID Regulator in Cascade Flow and Level Control

In a cascade control architecture, the PID (Proportional-Integral-Derivative) regulator plays a central role in achieving precise and stable regulation of both the flow (secondary loop) and the liquid level (primary loop). Each control loop is equipped with a dedicated PID controller, which continuously minimizes the error between the measured process variable and its corresponding setpoint [5].

### I.7.2.1 PID in the Flow Control Loop (Secondary Loop)

The flow loop is typically characterized by fast dynamics. The PID regulator in this loop ensures rapid response to disturbances such as upstream pressure variations or valve nonlinearities. The proportional term provides immediate corrective action, the integral term eliminates steady-state errors, and the derivative term anticipates future trends, thereby improving response speed and damping oscillations.

By maintaining the flow rate at the setpoint received from the level controller, the flow PID regulator ensures that any disturbance is corrected before it propagates to the level control loop[2]

### **I.7.2.2 PID in the Level Control Loop (Primary Loop)**

The level control loop responds more slowly due to the integrating nature of the process. The PID controller in this loop maintains the liquid level at the desired setpoint by dynamically adjusting the setpoint of the flow controller. The integral action is particularly critical here to eliminate long-term offset, while the proportional and derivative terms help enhance system stability and transient response.

By coordinating both loops through the cascade structure, the use of PID regulation ensures:

- Fast disturbance rejection through the inner (flow) loop,
- Robust and accurate level control in the outer loop,
- Minimized process variability and improved overall system performance. [1]

## **I.8. Conclusion:**

cascade control using PID regulators is a proven technique for improving the performance of flow and level control systems. By employing a secondary (inner) PID loop to quickly correct flow disturbances before they affect the primary (outer) level control loop, cascade structures offer faster response times, better disturbance rejection, and enhanced overall stability. Proper tuning, ensuring that the inner loop is significantly faster than the outer loop, is essential for effective operation. Cascade control thus provides a more robust and precise regulation method compared to traditional single-loop systems, making it a key strategy in modern industrial automation [2].

**Chapter II**  
**General Overview of Programmable**  
**Logic Controller**

## II.1. Introduction

The Programmable Logic Controller, PLC is a special type of computer—robust and responsive—with physical input and output interfaces. It is used to automate processes such as controlling machinery on an assembly line in a factory or managing automated handling systems. [13]

In this chapter, we will present the history and architecture of these powerful tools, as well as the programming languages used. We will also on the different SIMATIC ranges and focus from Siemens.

## II.2. General Information about PLCs

### II.2.1. History

Sequential automation has long been implemented using electromagnetic relays. The downside of this approach is that it is a wired system, which requires a complete rewiring for even the slightest modification in the sequence logic.

In 1966, the emergence of static relays made it possible to create various additional modules such as counters, timers, and step-by-step logic. However, this technology still had the same drawback—it remained a hard-wired solution.

In 1968, responding to demands from the North American automotive industry, the first programmable logic control devices were introduced: the **PLCs** (Programmable Logic Controllers) by Allen-Bradley and Modicon. [14]

### II.2.2. Definition of a PLC

NFC Standard 63-850 defines it as:

"An electronic device that includes memory programmable by an automation engineer using a suitable language. It stores instructions internally to execute automation functions such as: sequential and combinational logic, timing, counting, up/down counting, comparison, arithmetic calculations, control, regulation, etc., to command, measure, and control various types of machines or industrial processes via input and output modules (logical, digital, or analog)." Thus, a PLC is an electronic machine situated between two major categories: hard-wired logic and universal computers. It is characterized by its ability to operate in harsh industrial environments. [14]

## II.3. Architecture of PLCs [15]

### II.3.1. External Appearance

PLCs can be either **compact** or **modular**.

- **Compact type:**

This includes programming modules such as Siemens LOGO, Schneider's ZELIO, or Crouzet's MILLENIUM, which are considered micro-PLCs. These integrate the processor, power supply, and I/O (inputs and outputs). Depending on the model and manufacturer, they may offer additional features like high-speed counting or analog I/O, and they may support a limited number of extensions. These simple-function PLCs are generally used for controlling small automation systems. [14]

- **Modular type:**

The processor, power supply, and I/O interfaces are housed in separate units (modules) that are mounted on one or more racks containing the backplane bus (bus and connectors). These PLCs are utilized in complex automation systems where power, processing capability, and flexibility are essential. [14]

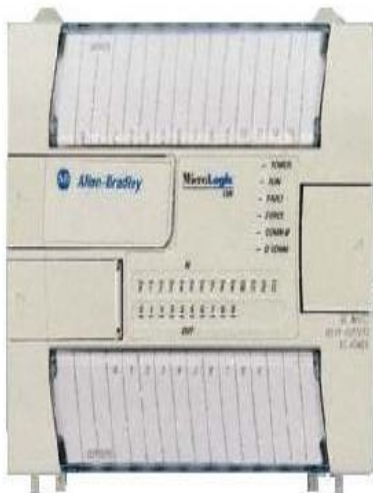


Figure II .1 : Automate compact



Figure II .2 : Automate modulaire (Modicon)

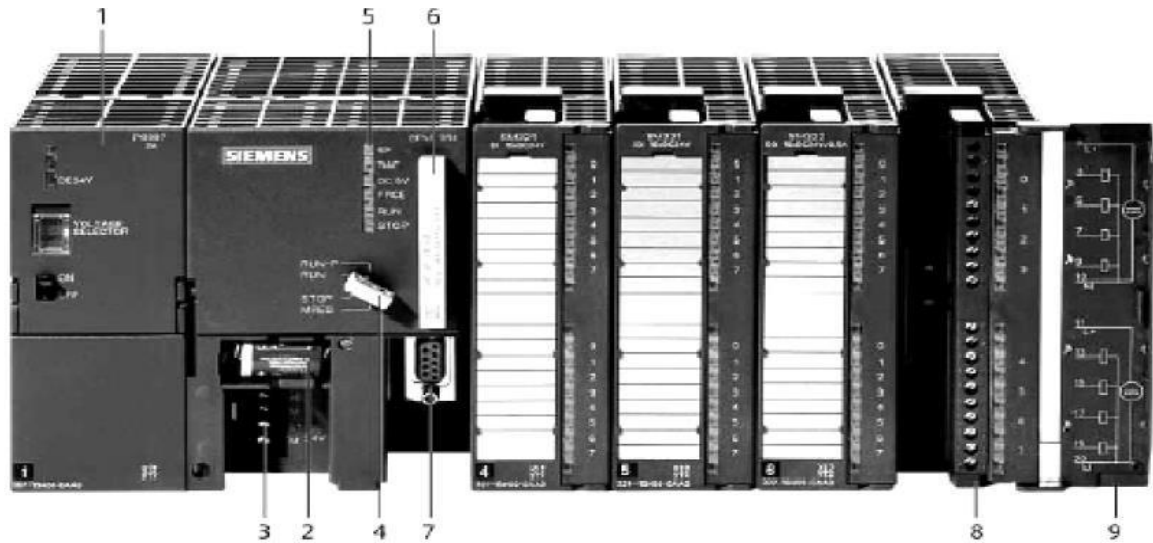


Figure II.3: Modular PLC (Siemens)[15]

1. Power Supply Module
2. Backup Battery
3. 24V DC Connection
4. Mode Selector Switch (key-operated)
5. Status and Fault Indicator LEDs
6. Memory Card
7. Multipoint Interface (MPI)
8. Front Connector
9. Front Panel Flap

### II.3.2. Internal Structure

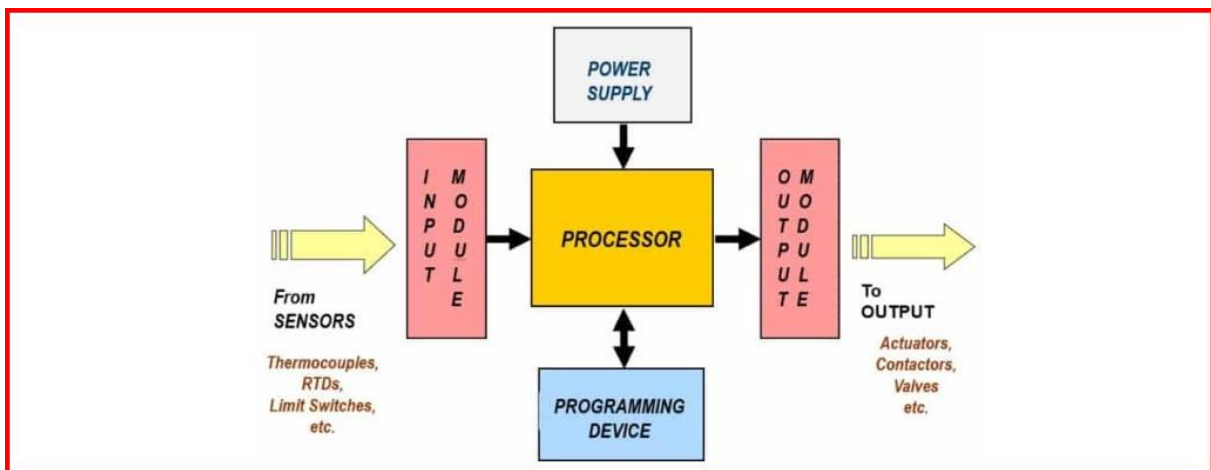


Figure II.4: Internal Structure of a PLC

The internal structure of a PLC consists of several key components. The Power Supply Module provides power, while the Central Processing Unit (CPU) manages logic and data tasks.

An Internal Bus connects different modules. Memory Units include ROM/PROM for the operating system, EEPROM for user programs, and RAM for temporary data, expandable with PCMCIA cards. Input/Output Interfaces receive signals and send commands.

Functions vary by PLC type, with Compact PLCs managing basic tasks and Modular PLCs offering advanced features. Main functional modules include I/O, High-Speed Counter, Axis Control, and Analog I/O Modules. Additional specialized modules are available for various tasks.[14]

### Programming Languages:

- IEC 61131-3 standard defines five PLC programming languages, divided into two categories .

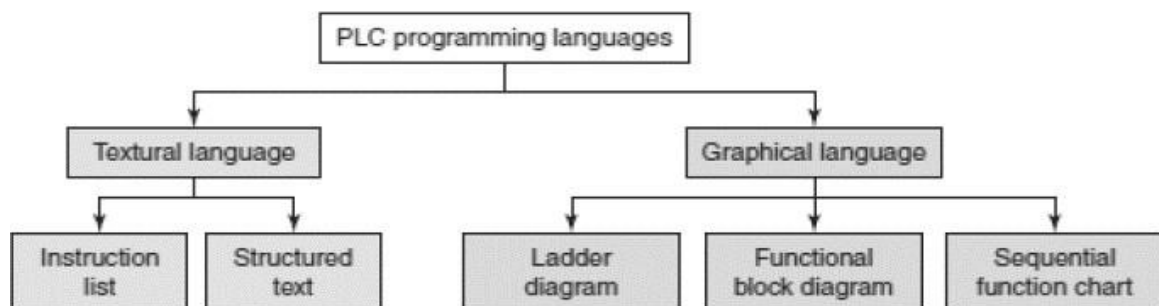


Figure II .5: Programming Languages

### III.3.3 Graphical Languages

- **SFC** (*Sequential Function Chart*) or **GRAFCET**
  - **LD** (*Ladder Diagram*) or relay schematic
  - **FBD** (*Function Block Diagram*) or block schematic
- **Textual Languages**
    - **ST** (*Structured Text*)
    - **IL** (*Instruction List*)

---

## II.4. Description of the S7-315 Programmable Controller

The S7-315 is a model in the Siemens SIMATIC S7 series, intended for medium to complex control tasks across different industries. Key features include:

1. Modular Design: It allows customization by adding or removing modules based on project needs.
2. Central Processing Unit (CPU): Executes user programs and manages I/O, with various models offering different memory and speed.
3. Memory and Performance: Supports real-time tasks efficiently with adequate program and data memory.
4. Input/Output Modules: Interacts with sensors and devices through various digital and analog I/O.
5. Communication and Networking: Works with multiple industrial protocols for network integration.
6. Flexibility and Expandability: Easily expandable to meet changing application needs.
7. Programming and Configuration: Utilizes STEP 7 software and supports multiple programming languages.
8. Applications: Commonly used in production lines, robotics, building automation, and process control for reliability and performance..[15]



**Figure II.6: Plc S7- 300-315 2DP NP**

Regarding the CPU 315, the summary table of the different positions of the operating mode switch is as follows: [15]

**Table II.1: Position of the Operating Mode Switch [15]**

Switch position	Mode name	Main function
stop	Stop mode	The user program is halted. No PLC cycle is executed. Outputs are disabled (or remain in their last state depending on configuration).
Run-p	Run with programming	The program is running, and the user can monitor variables, download blocks, or modify the program online (useful for debugging or commissioning).
Run	Run mode	The user program runs normally. No changes to the program are allowed. This is the standard production mode.

As for the components: battery, memory card, and MPI interface, they will be described in the following table along with the technical specifications of the CPU:

**Table II.2 Technical Specifications – Siemens CPU 315-2 PN/DP**

Parameter	Details
CPU reference	6ES7 315-2EH14-0AB0
Hardware Version	Typically HW V2.6 / V3.0
Programming software	STEP7 V5.5 or Tia portal v13 and high
Expandable Work Memory	Not expandable
Load Memory	Micro Memory Card (MMC), from 512 KB to 8 MB
Retentive Memory	Supported (battery-backed or non-volatile memory)
Backup Battery	Optional, for data and clock retention
<b>Instruction Execution Times</b>	
Operation Type	Execution Time (Typical)

Bit Logic Operation (e.g. AND, OR)	~0.06 $\mu$ s
Word Operation (e.g. MOVE)	~0.09 $\mu$ s
Integer Arithmetic (e.g. ADD_I)	~0.18 $\mu$ s
Floating-Point Arithmetic (e.g. ADD_R)	~0.45 $\mu$ s
<b>Timers</b>	
<b>Feature</b>	<b>Specification</b>
Number of Timers	512 timers
Time Range	10 ms to 9990 seconds
<b>Organization Blocks (OBs)</b>	
Block type	Details
OB1 (Main Program Cycle)	1 main cycle block
Time-triggered OBs (OB10–OB17)	Yes – Available
Cyclic Interrupt OB (OB35)	Supported
Startup OBs (OB100–OB102)	Supported
Error/Exception OBs (OB121–OB122)	Supported
<b>Program Blocks</b>	
<b>Type</b>	<b>Maximum</b>
Function Blocks (FBs)	Up to 2048 / each up to 64 KB
Functions (FCs)	Up to 2048 / each up to 64 KB
<b>Communication Interfaces</b>	
MPI / Profibus DP Port	1 port (supports MPI, DP-Slave, DP-Master modes)
PROFINET (Ethernet Port)	1 port – RJ45

**II.5. Conclusion**

The programmable logic controller (PLC) has become an essential component in automated production systems. Thanks to its mobility, the flexibility of its architecture, the ease of its programming, connectivity, and adaptation to industrial environments.

However, the wide range of possibilities it offers in implementation and its cost are not sufficient conditions on their own when developing an automation solution. Indeed, a thorough analysis of the problem to be solved, as well as adherence to installation guidelines, are essential.

**Chapter III**  
**Programming and Monitoring in TIA**  
**Portal and Simulation with Factory**  
**I/O**

### **III.1. Introduction**

In this chapter, we move from the theoretical aspect to the practical implementation of our project, which aims to design and develop an automated control system for the liquid level in an industrial tank, based on a **cascade PID control** structure between the variables "level" and "flow." This system controls the tank filling level by adjusting the inflow using a control valve that is operated via an analog signal.

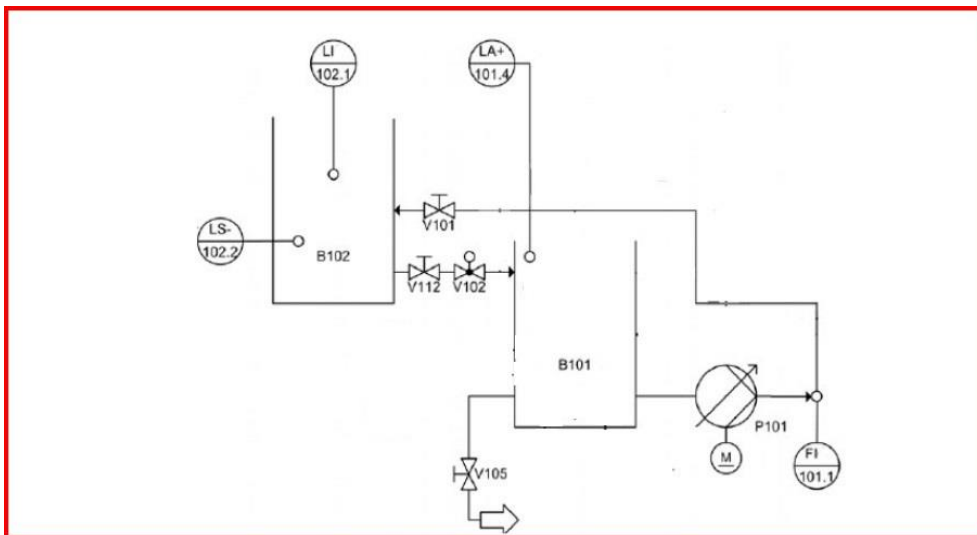
The control system is based on a **Siemens S7-300 PLC (CPU 315-2 PN/DP)** and is programmed using **TIA Portal V13**, Siemens' integrated engineering platform. To simulate the process and test the control logic without real hardware, we use **Factory I/O**, a 3D industrial simulation software that provides a realistic and interactive environment for validation.

### **III .2. Cascade level control system with flow**

The role of the cascade level control system with flow is to regulate the filling level of tank B102 by controlling the analog flow using the pump as the control element.

This allows the level control, which has a slow dynamic response, to react more quickly to external disturbances and setpoint changes.

After performing the necessary configurations, the synoptic diagram of the system becomes:



**Figure. III.1: Circuit related to cascade level control with flow [17]**

### **III .2.1. Description project**

For this regulation, aside from the information provided by the digital sensors mentioned above, the filling level of tank B102 is measured by an analog sensor at the measurement point . The flow rate of the fluid delivered by the pump through the piping system is calculated via an optoelectronic flow sensor (FIC101).

Indeed, the pump, used in analog mode, delivers the water stored in tank B101 to tank B102 through a piping system. The filling level of tank B102 must remain constant, even in the presence of disturbances. Therefore, the level value transmitted by the analog sensor must remain stable.

### **III.2.2 System Overview**

The system is composed of the following main components:

- **Analog Level Sensor** – Measures the liquid level in the main tank.
- **Analog Flow Sensor** – Measures the flow rate of the fluid entering or leaving the tank.
- **PLC (Programmable Logic Controller)** – Processes sensor data and executes the control algorithm.
- **PID Controllers** – Regulate the flow and level based on feedback and setpoints.
- **Control Valve** – Adjusts the flow rate to maintain the desired level.
- **Automatic Drain Valve** – Opens to discharge excess fluid in case of overflow or system disturbance.

The **level sensor** continuously measures the liquid level in the tank and sends an **analog signal** (typically 4–20 mA or 0–10 V) to the PLC.

The **flow sensor** measures the real-time flow rate of the liquid and also sends an analog signal to the PLC.

These signals are processed by the **analog input modules** of the PLC and scaled into engineering values (e.g., cm for level, L/min for flow).

### **III.3. Programing the cascade control system**

In this section, we present the implementation of the cascade control system using a Siemens TIA PORTAL S7-315 DP/PN programmable logic controller (PLC). The control logic is developed using **STEP 7** software in **FBD** (Function Block Diagram) programming languages, depending on clarity and suitability for the system's requirements.

The cascade control system consists of two control loops:

1. **Primary loop (Master loop):** Maintains the level in tank B102 constant by adjusting the fill valve.
2. **Secondary loop (Slave loop):** Controls the flow rate delivered by the pump, ensuring fast and stable response to the master loop's demand.

The system receives analog input signals:

- **Level measurement (LIC102):** 4–20 mA converted to 0–10 V via A1, connected to analog input channel AI0 of module X2.
- **Flow measurement (FIC101):** 0–1000 Hz square wave converted to 0–10 V via A2, connected to analog input channel AI1 of X2.

And it produces analog output:

- **valve control signal:** 0–10 V from the PLC analog output, adapted to 0–24 V by A4 and applied to valve

Additionally, digital signals are used to monitor or simulate disturbances such as valve V102 opening/closing.

The programming structure includes:

- Analog signal scaling and normalization
- PID control blocks (FB32)
- Data blocks (DB) for parameter storage
- Interconnection between master and slave PID controllers

### III.3.1 Introduction to the TIA Portal[18]

**Totally Integrated Automation Portal (TIA Portal)** is a comprehensive software suite developed by Siemens for designing, programming, and managing automation systems. It serves as a unified platform that integrates various automation components, streamlining the engineering process and enhancing productivity

#### Key Features of TIA Portal

- **Centralized Engineering:** Integrates tools for PLC programming, HMI design, and drive configuration into a single environment.
- **Multi-Language Support:** Supports programming languages like Ladder Logic (LAD), Function Block Diagram (FBD), Structured Text (ST), and Instruction List (IL).
- **Simulation and Testing:** Allows for virtual testing of programs to detect and resolve issues before deployment.
- **Advanced Diagnostics:** Provides tools for monitoring and troubleshooting systems in real-time.
- **Seamless Integration:** Ensures smooth communication between different automation devices, enhancing system efficiency.[18]

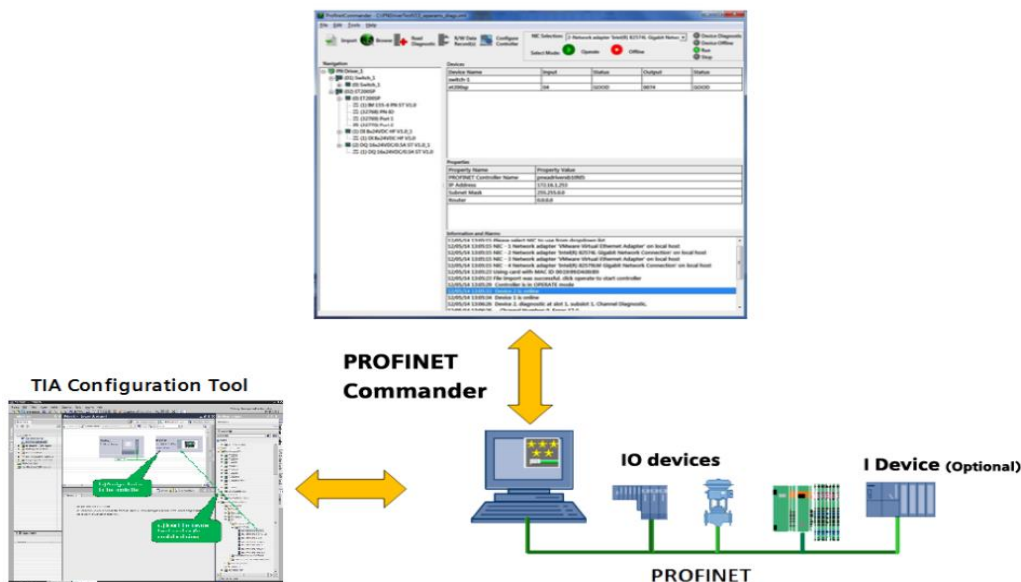


Figure III.2. TIA Portal [18]

### III.3.2. Advantages of working with the TIA Portal:

By working with the TIA Portal, we benefit from effective support in implementing our automation solution thanks to the following features:

- Seamless engineering based on a unified control concept.
- Centralized and consistent data management using powerful editors and transparent symbolic representation
- Comprehensive library concept
- Multiple programming languages. [16]

### III.4. Project View

The project view corresponds to a structured view of all the components within a project. In the project view, we have access to various editors through which we create and edit the corresponding project components.

The following figure shows the structure of the project view.

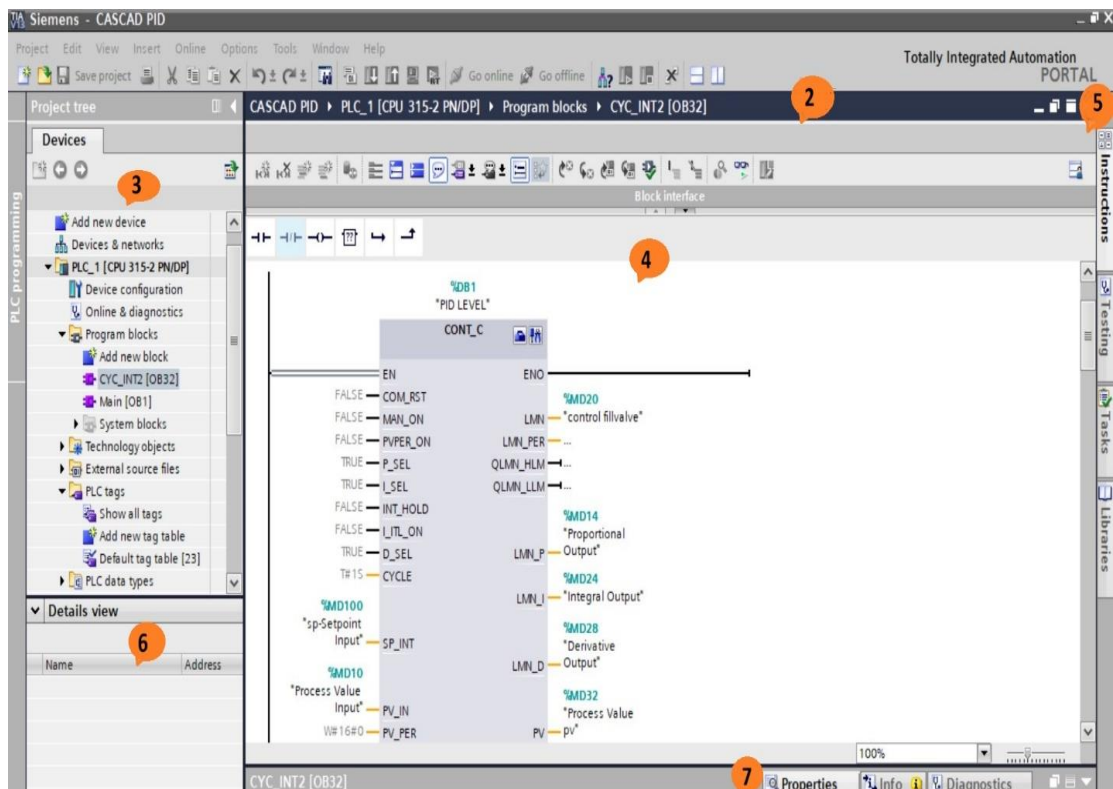


Figure III.3. Project View

**1 Menu Bar** includes commands for project tasks like creating and editing components.

**2 Toolbar** gives quick access to common commands for better efficiency.

**3 Project Navigator** shows all project components, allowing users to add, edit, or view properties.

**4 Work Area** is for editing objects like PLC blocks and HMI screens.

**5 Task Cards** provide context-specific tools.

**6 Detailed View** offers specific details about selected objects.

**7 Inspector Window** displays extra information, including properties, messages, and diagnostics.

### III.5. Programming and Design of Cascade PID Control with TIA Portal

#### III.5.1. Project Creation

The following steps describe how to create a new project.

Start the Totally Integrated Automation Portal by clicking Start > Programs > Siemens Automation > TIA Portal V13 to create a new project. Then, create the project "cascade PID" in a chosen directory.

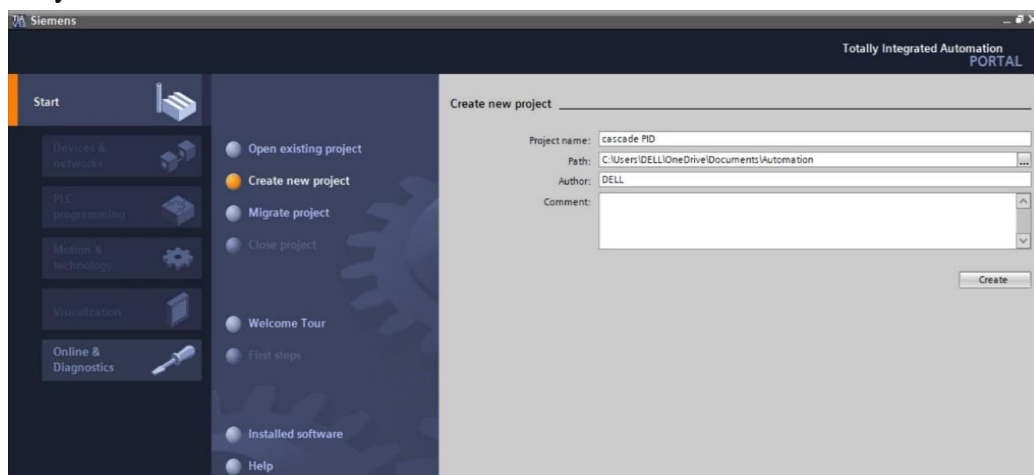


Figure III.4. Project Creation

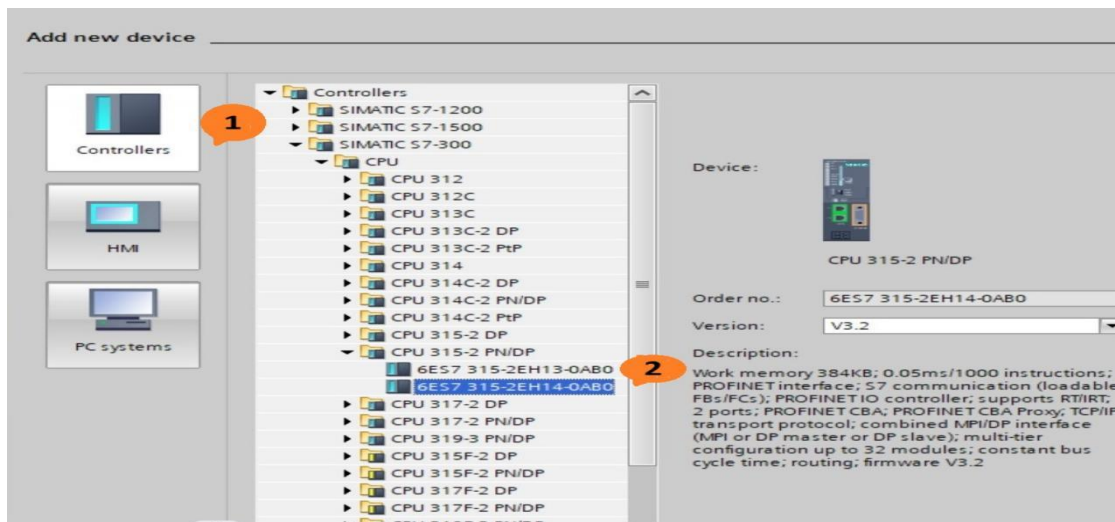
#### III.5.2. Inserting and Configuring a PLC

The following steps describe how to insert a PLC through the portal view and how to open its configuration in the project view.

##### Procedure

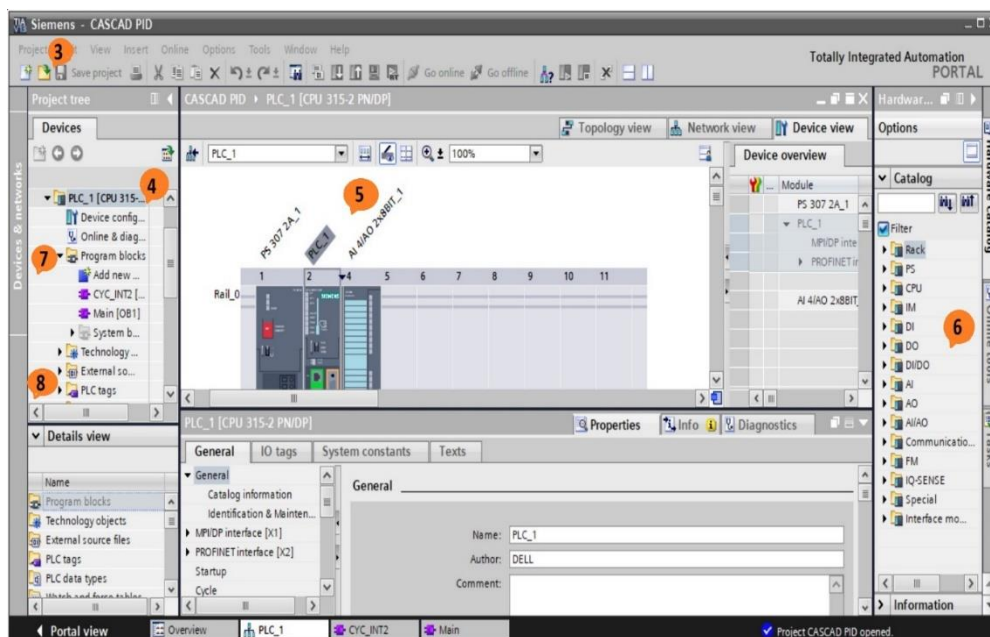
Follow the steps below to add a new device to the project:

1. Add a new device through the **portal view**.
2. Select the desired PLC, which in our case is the **SIMATIC S7-300 CPU 315-2 DP/NP**.



**Figure III.5. Inserting and Configuring a PLC (1)**

3. Enable the "Open device view" option.
4. The PLC will be added to the project, and the Device View will open automatically.
5. Configure the hardware setup by adding input/output modules, power supply, or communication modules.
6. After hardware configuration, proceed to the Program Blocks section to write your control logic.
7. Create PLC tags and data blocks for variables and parameters.
8. The project can then be simulated or tested on actual PLC hardware.



**Figure III.6. Inserting and Configuring a PLC (2)**

**III.5.3. Signal Modules for PLC 315-2 PN/DP and SM334**

1. he **S7-300 CPU 315-2 PN/DP** is a mid-range Siemens PLC used in industrial automation. It features two communication interfaces: **PROFIBUS DP** and **PROFINET**, allowing flexible integration with other devices. [15]

**Table III.1. General Characteristics of PLC S7-300 CPU 315-2 PN/DP [15]**

<b>Parameter</b>	<b>Description</b>
Product Name	SIMATIC S7-300 CPU 315-2 PN/DP
Manufacturer	Siemens
CPU Type	Modular PLC (Medium-performance range)
Memory	384 KB work memory
Processing Speed	Approx. 0.1 $\mu$ s per bit operation
Communication Ports	1 $\times$ MPI/DP (PROFIBUS), 1 $\times$ PROFINET
Power Supply	24 V DC (via separate power module)
Integrated I/O	No (requires external SM modules)
Programming Languages	LAD, FBD, STL
Mounting Type	DIN rail (modular rack system)
Expansion Capability	Yes, via signal modules and communication processors
Application Area	Industrial automation, process control, training systems

**SM334** is an analog I/O module for Siemens **S7-300 PLCs**, combining input and output functions in one unit. It handles standard signal ranges (0–10V, 4–20mA) and is used for reading sensor data and controlling analog devices in automation systems. [15]

Table III.2. General Characteristics of SM 334 (Analog Input/Output Module) [15]

Parameter	Description
Module Name	SM 334
Type	Analog Input/Output Module
Channels	4 analog inputs and 2 analog outputs (typical configuration)
Input Signal Types	Voltage (e.g., 0–10 V, ±10 V), Current (e.g., 0–20 mA, 4–20)
Output Signal Types	Voltage (e.g., 0–10 V, ±10 V), Current (e.g., 0–20 mA, 4–20)
Resolution	12 to 16 bits (depending on signal type and configuration)
Isolation	Electrical isolation between channels may be limited (check specific variant)
Mounting	DIN rail (fits into the S7-300 rack system)
Use Case	Reading analog signals from sensors and sending analog signals to actuators

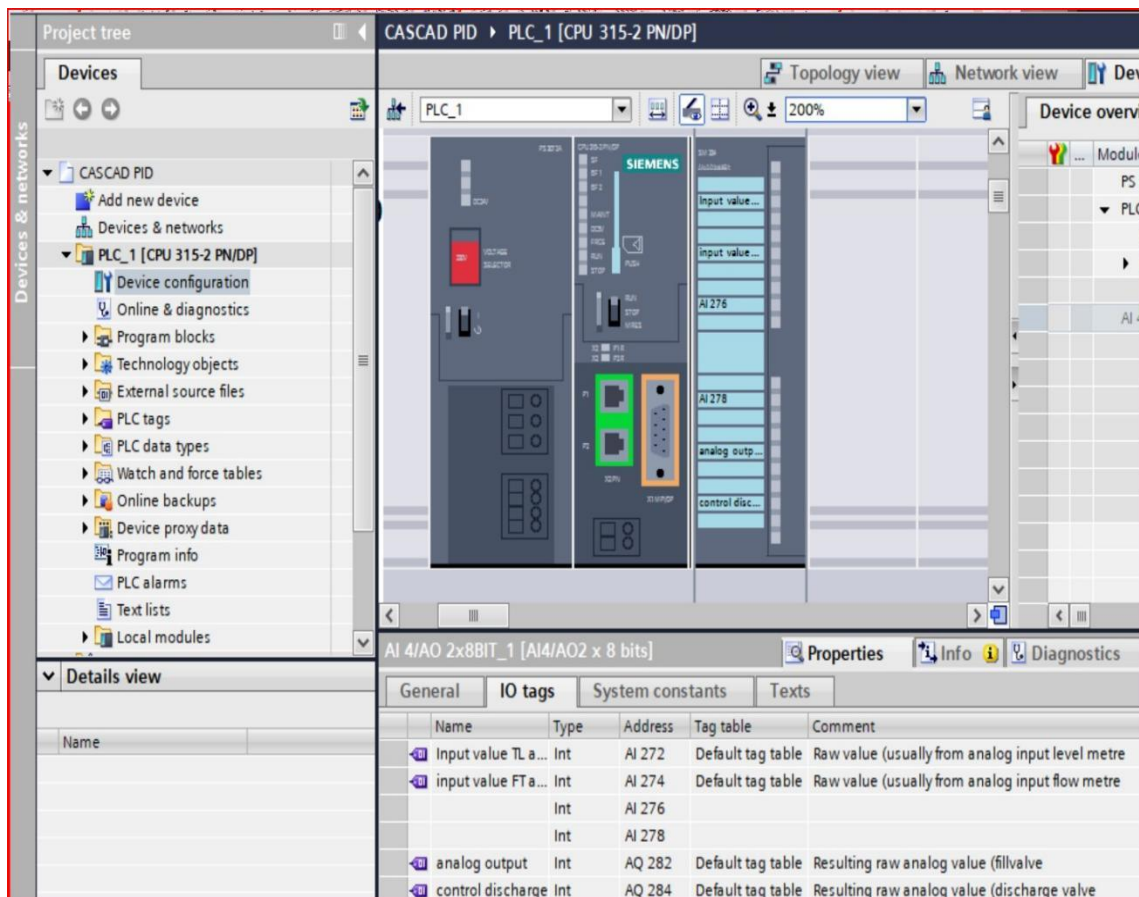


Figure III.7. Configuration of SM 334 (Analog Input/output Module)

### Program Creation:

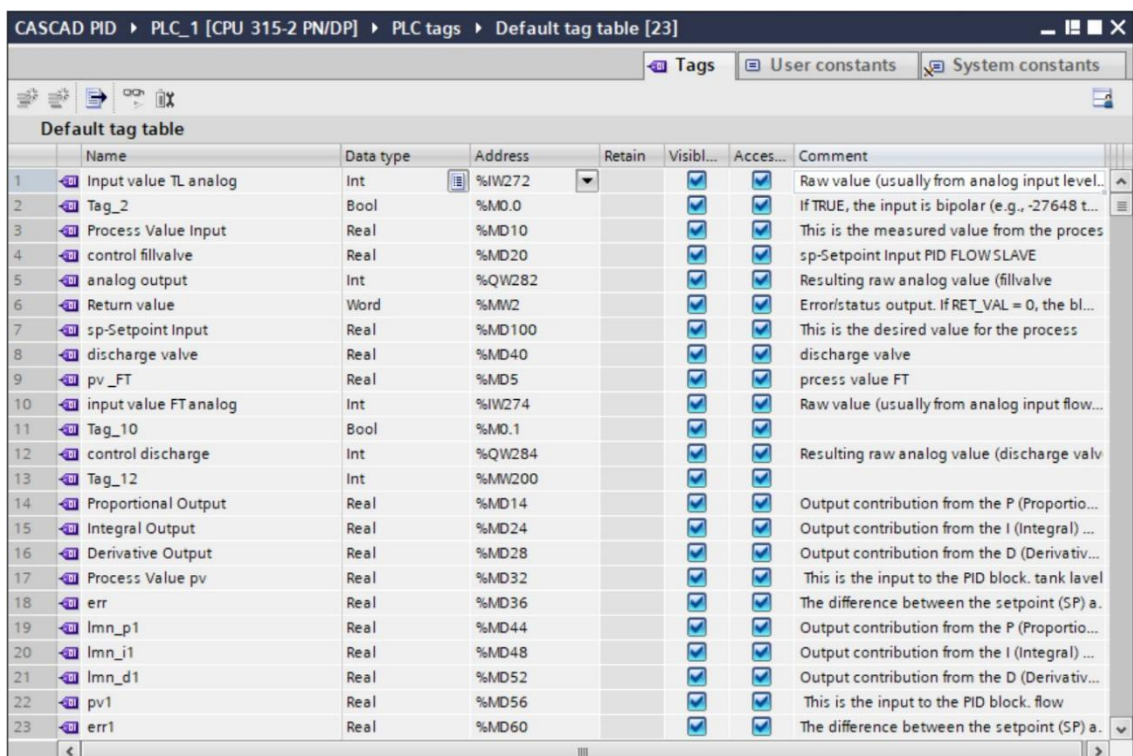
In this section, we will begin programming the cascade PID by following this workflow, which will be explained in detail later:

- Creation of the variable table
- Creation of the data block
- Creation of the organization block "OB32"
- Program compilation and simulation

### III.5.4. Creation of the Variable Table

A variable is a value in a program that can change. There are two types of variables: local variables, which are valid only in their specific block, and global (PLC) variables, which are valid throughout the entire PLC program.

Variables are managed in the TIA Portal and can be created in the program editor or variable table. The text includes instructions on how to fill in the variable table:



	Name	Data type	Address	Retain	Visibl...	Acces...	Comment
1	Input value TL analog	Int	%IW272		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Raw value (usually from analog input level...
2	Tag_2	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	If TRUE, the input is bipolar (e.g., -27648 t...
3	Process Value Input	Real	%MD10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	This is the measured value from the proces
4	control fillvalve	Real	%MD20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	sp-Setpoint Input PID FLOW SLAVE
5	analog output	Int	%QW282		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Resulting raw analog value (fillvalve
6	Return value	Word	%MW2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Error/status output. If RET_VAL = 0, the bl...
7	sp-Setpoint Input	Real	%MD100		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	This is the desired value for the process
8	discharge valve	Real	%MD40		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	discharge valve
9	pv_FT	Real	%MD5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	process value FT
10	input value FT analog	Int	%IW274		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Raw value (usually from analog input flow...
11	Tag_10	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	control discharge	Int	%QW284		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Resulting raw analog value (discharge valv
13	Tag_12	Int	%MW200		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Proportional Output	Real	%MD14		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Output contribution from the P (Proportio...
15	Integral Output	Real	%MD24		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Output contribution from the I (Integral) ...
16	Derivative Output	Real	%MD28		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Output contribution from the D (Derivativ...
17	Process Value pv	Real	%MD32		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	This is the input to the PID block. tank level
18	err	Real	%MD36		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	The difference between the setpoint (SP) a...
19	lmn_p1	Real	%MD44		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Output contribution from the P (Proportio...
20	lmn_i1	Real	%MD48		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Output contribution from the I (Integral) ...
21	lmn_d1	Real	%MD52		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Output contribution from the D (Derivativ...
22	pv1	Real	%MD56		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	This is the input to the PID block. flow
23	err1	Real	%MD60		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	The difference between the setpoint (SP) a...

Figure III.8. the Variable Table

### III.5.5. Creation of a Global Data Block

A global data block (which can be accessed by any code block in the program) is a table consisting of a number of shelves (up to a maximum of 256) where data can be stored in the form of Booleans, single or double integers, single or double real numbers, characters, etc.

The following figure shows the creation of a data block:

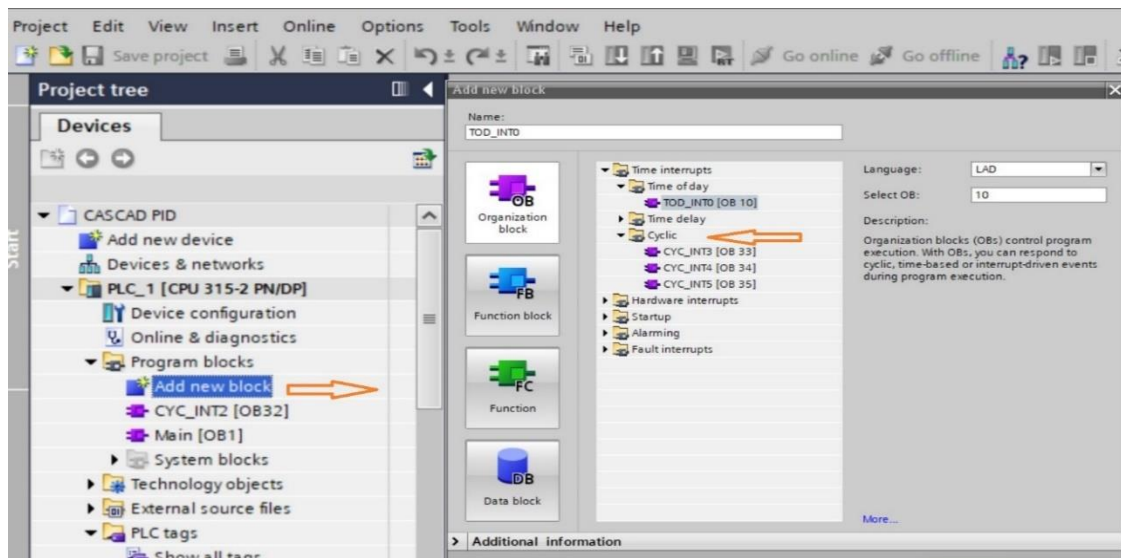


Figure III.9. Creation of a Global Data Block

### III.5.6. Creation of the Organization Block “OB32”

#### III.5.6.1. User Program

A user program in a Siemens PLC consists of one or more blocks, with at least one Organization Block (OB) required. These blocks handle automation tasks such as:

- Processing analog and binary data
- Responding to alarms (e.g., exceeding analog range)
- Handling runtime disturbances

#### Organization Blocks (OBs):

OBs act as the interface between the PLC’s operating system and the user program. They control key operations like:

- System startup behavior
- Cyclic execution of user logic
- Alarm-driven tasks

following figure shows how a cyclic OB is executed:

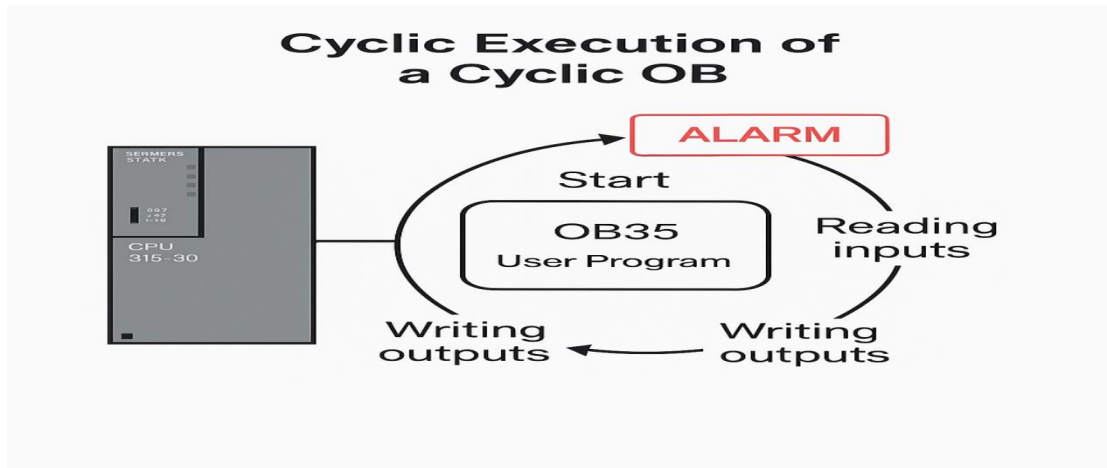


Figure III.10. cyclic OB

### III.5.6.2 Creation of the Organization Block for the PID Controller

Since we are performing a simulation on Factory I/O to control the liquid level using cascade PID\_C control, we will insert one PID block for the level and another PID block for the flow.

#### Procedure

Follow the steps below to add a PID control:

1. In instruction opening technology tab and open folder PID control.
2. Add block PID\_C in Editor Overview

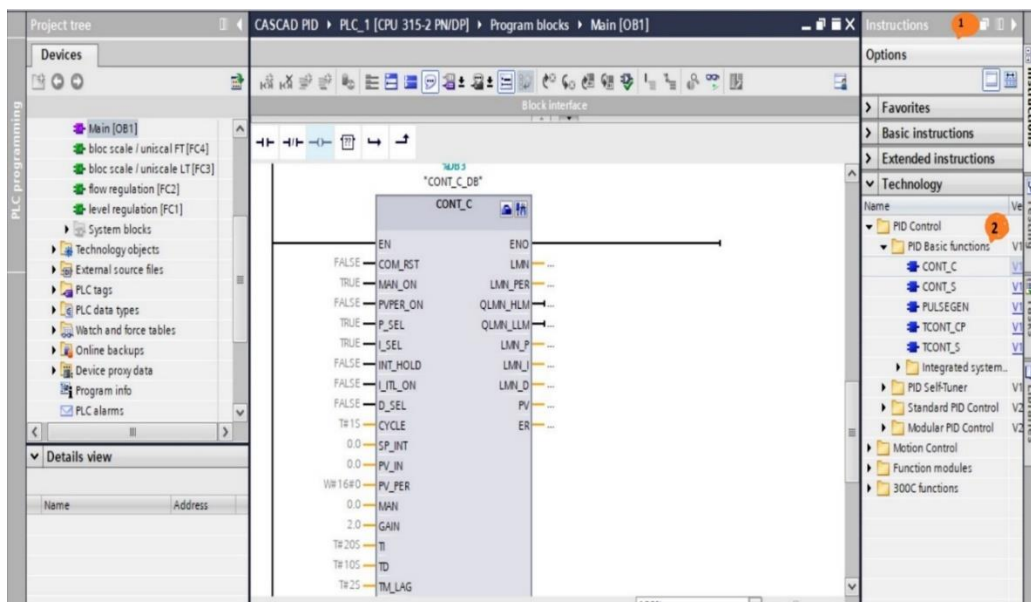


Figure III.11. Creation of the Block PID Controller

The following figure shows the structure of the control block and its input/output parameters, which we will explain later

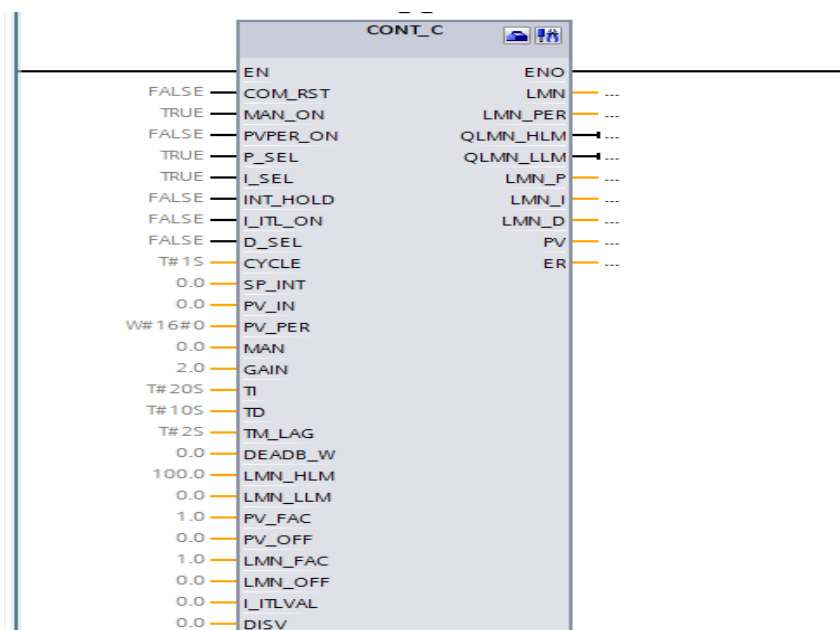


Figure III.12: PID block.

### III.5.6.3 Principle of Regulation

The variable to be controlled is continuously measured and then compared with a setpoint value provided as an input to the system. Based on the result of this comparison, the control system must adjust the process in order to bring the measured value closer to the setpoint.

### III.5.7 Continuous Control with the "CONT\_C" Block

The "CONT\_C" block is used to control industrial processes with continuous input and output values on SIMATIC S7 programmable logic controllers. Its configuration allows for enabling or disabling partial functions of the PID controller, making it adaptable to the controlled system.

The block can be used as a standalone PID controller but also supports cascade, blending, or ratio control in multi-loop regulation systems. Its operation is based on the PID algorithm of a sampled-data controller with analog output. If necessary, it can include a pulse-shaping stage to generate PWM (Pulse Width Modulation) output signals for two-position or three-position controls using proportional actuators.

The functional diagram of this block is shown in the following figure:[15]

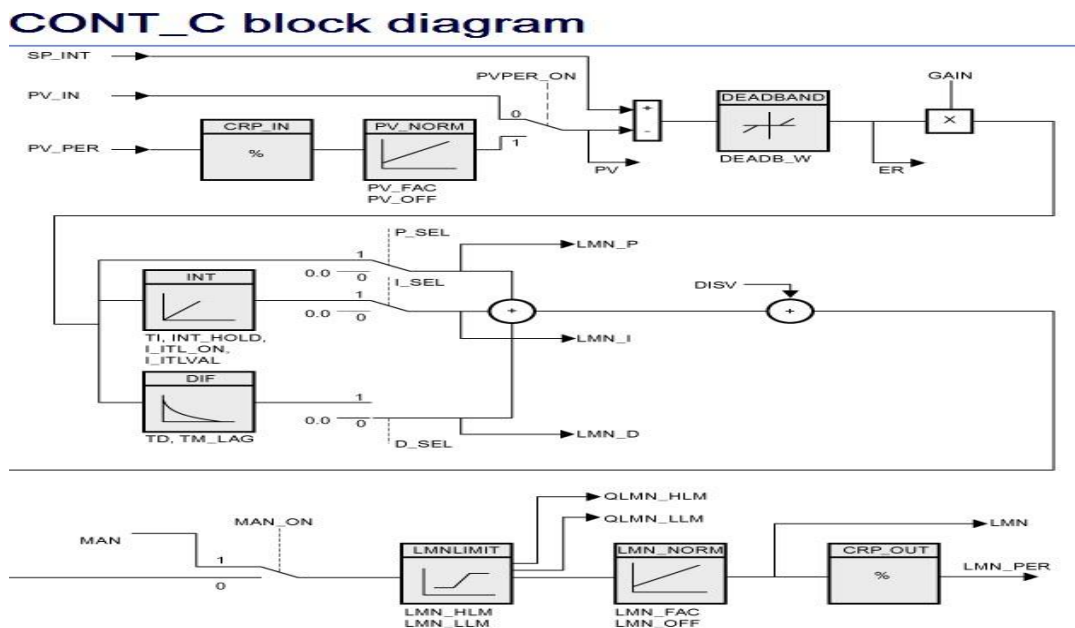


Figure III.13: Principal Diagram of the PID Block

### III.5.8 PID Configuration

Since our study focuses on **cascade control**, in the TIA Portal, you can configure a **PID Cascade** system using the **CONT\_C block**. We will program the control block accordingly for both loops.

In a cascade control structure:

- The **PID master controller** handles the **primary process variable** (level), comparing it to the main set point and generating a control signal.
- This control signal becomes the **set point** for the **PID slave controller**, which manages a **secondary variable** (flow ) that directly affects the primary variable.

The slave loop responds more quickly to disturbances, improving overall system stability and performance.

We will configure both the master and slave loops using CONT\_C blocks in TIA Portal. The setpoint of the slave block is defined by the output of the master block.

The Following figure and tables represent the required input and output parameters for the PID blocks in This Project.

- Master Block PID

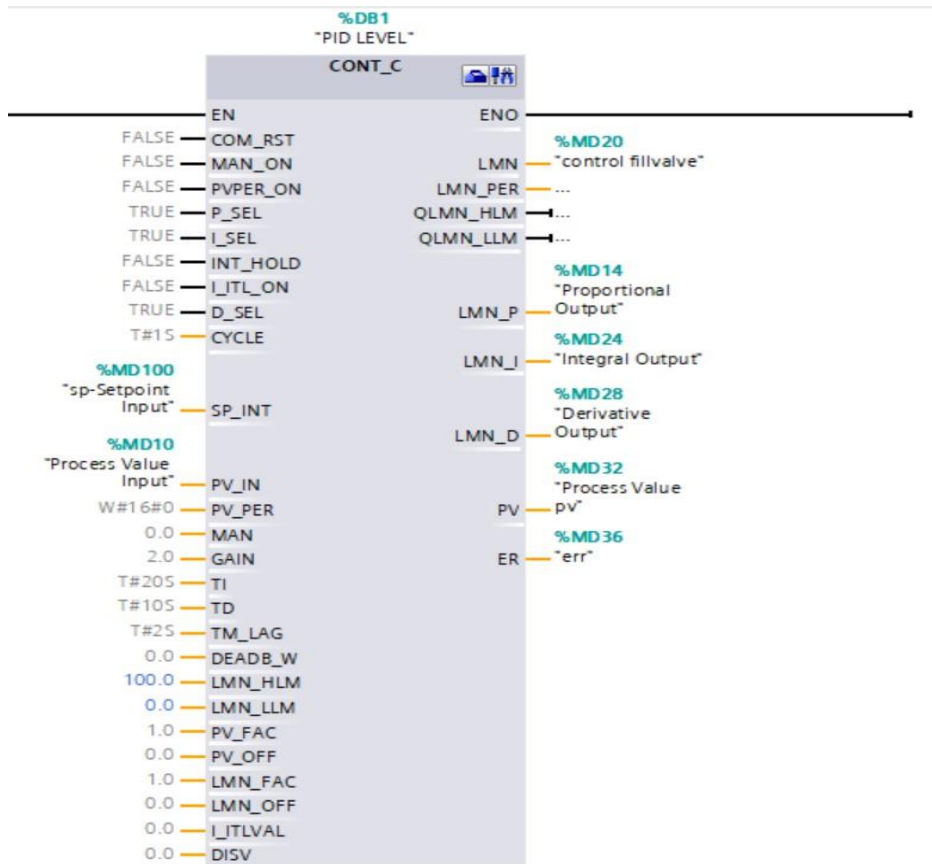


Figure III.14. the master block Configuration

"Table III.3: parameters input/output of the Master PID block

Adress	Type	Comment
MD5	Real	This is process value for the flow
MD40	Real	Output the procee(control valve)
MD20	Real	Sp-setpoint input slave flow pid
MD44	Real	Output contribution from proportional PID
MD48	Real	Output contribution from integral handles steady stat err
MD52	Real	Output contribution from derivative react to rate change err
MD56	Real	
MD60	Real	Err= sp-pv

- Slave Block PID

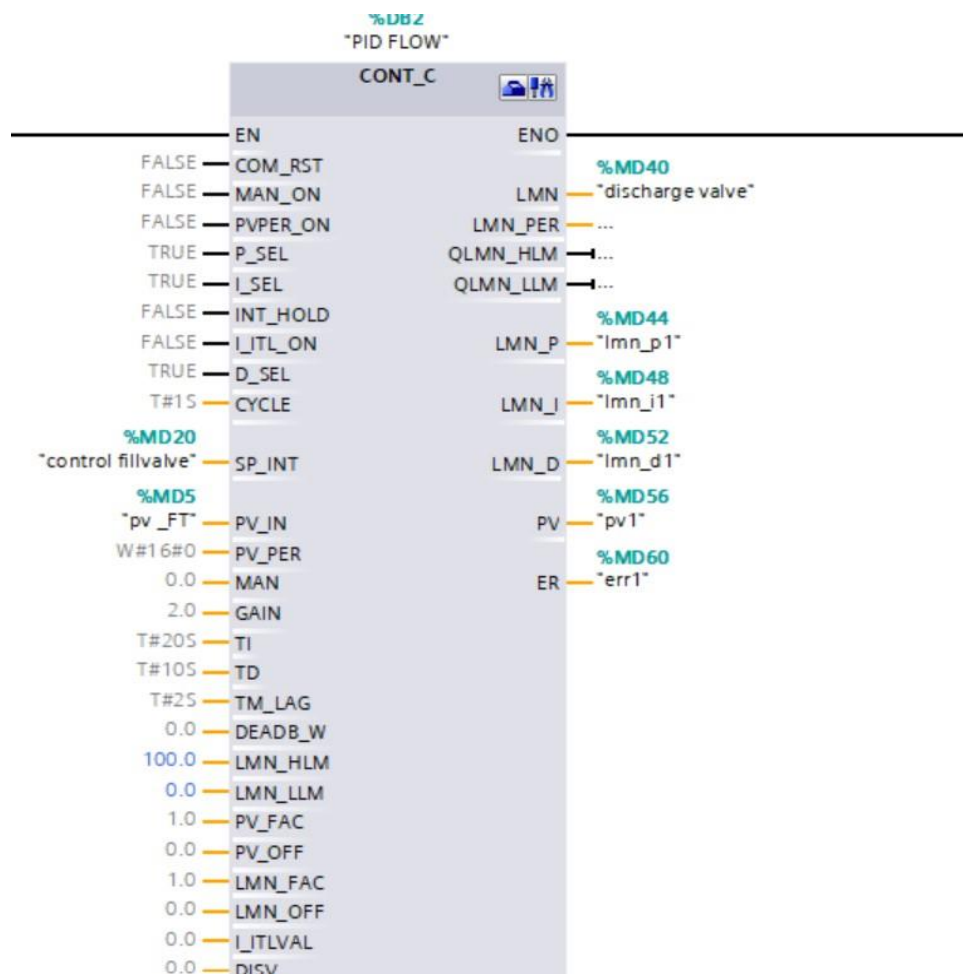


Figure III.15. the Slave Block Configuration

Table III.4: parameters input/output of the Slave PID block

Adress	type	comment
<b>MD5</b>	real	This is process value for the flow
<b>MD40</b>	real	Output the process (control valve)
<b>MD20</b>	real	Sp-setpoint input slave flow PID
<b>MD44</b>	real	Output contribution from proportional PID
<b>MD48</b>	real	Output contribution from integral handles steady stat err
<b>MD52</b>	real	Output contribution from derivative react to rate change err
<b>MD56</b>	real	
<b>MD60</b>	real	Err= sp - pv

### III.6 . Simulation using PLCSIM and Factory IO

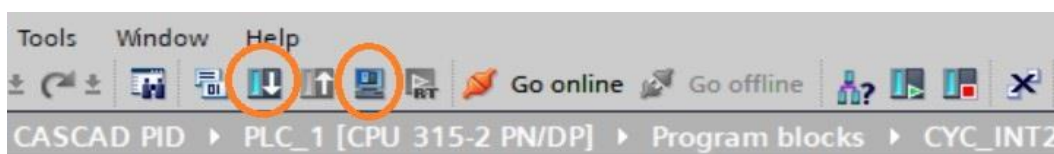
To start the simulation with PLCSIM, proceed as follows:

1. Save and Compile the hardware and software.



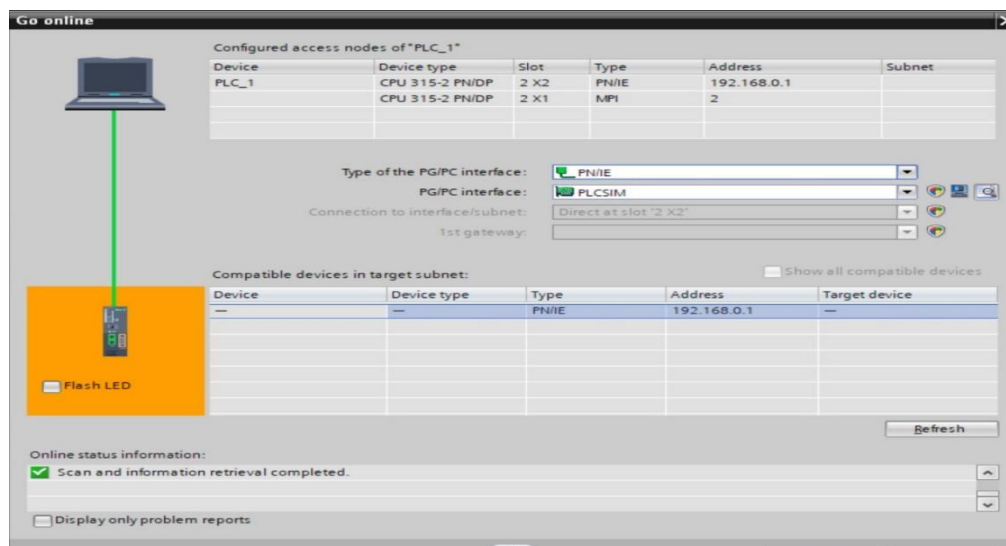
**Figure III.16.save and compile**

2. After compiling the program, Download to device and start the simulation by clicking on the "Start Simulation" icon in the toolbar.



**Figure III.17. download and simulation**

3. To configure the connection properties, select the correct interface in three steps: choose the type (PN/IE), select Ethernet Connection, and check "Show all compatible devices. " Click "start search," select your CPU from the list, confirm "Overwrite all," and proceed to "Load. "



**Figure III.18. extended download**

→ The "Start all" option will be selected next before the download operation can be completed with →"Finish".

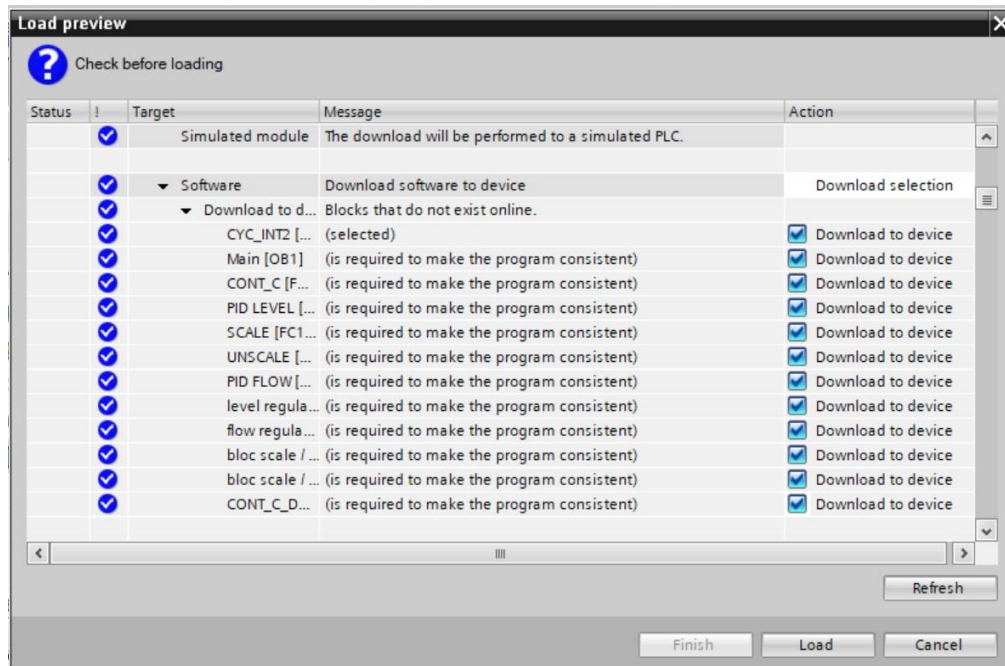


Figure III.19. load preview

After a successful download, the project view reopens, showing a loading report for troubleshooting. The PLCSIM simulator then launches. [13]

4. The PLCSIM simulator then launches (PLCSIM TCP/IP mode).

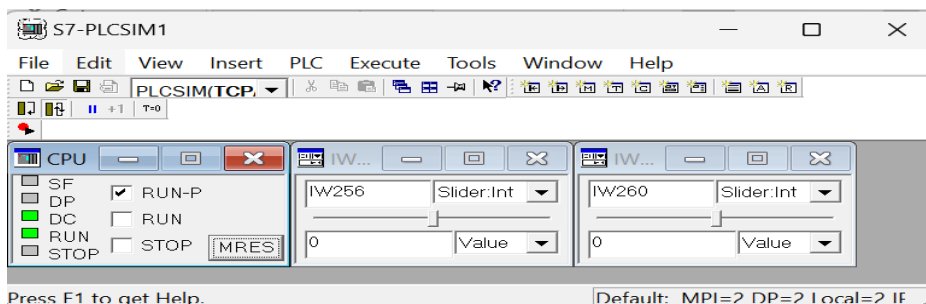


Figure III.20. start plcsim

## III.7. Factory I/O

### III.7.1 Overview

Factory I/O simulation software is available in 7 versions or editions: the starter edition, the Allen Bradley Edition for Allen Bradley PLCs, the Siemens Edition for Siemens PLCs, the Modbus and OPC Edition for Modbus and OPC communications, the MHJ Edition for the WinSPS-S7 virtual controller and the Siemens WinPLC-Engine PLC simulator, the Automgen Edition for the connection with the Automgen software, the Ultimate Edition which is the complete version

including an SDK for the developments communication drivers [18].

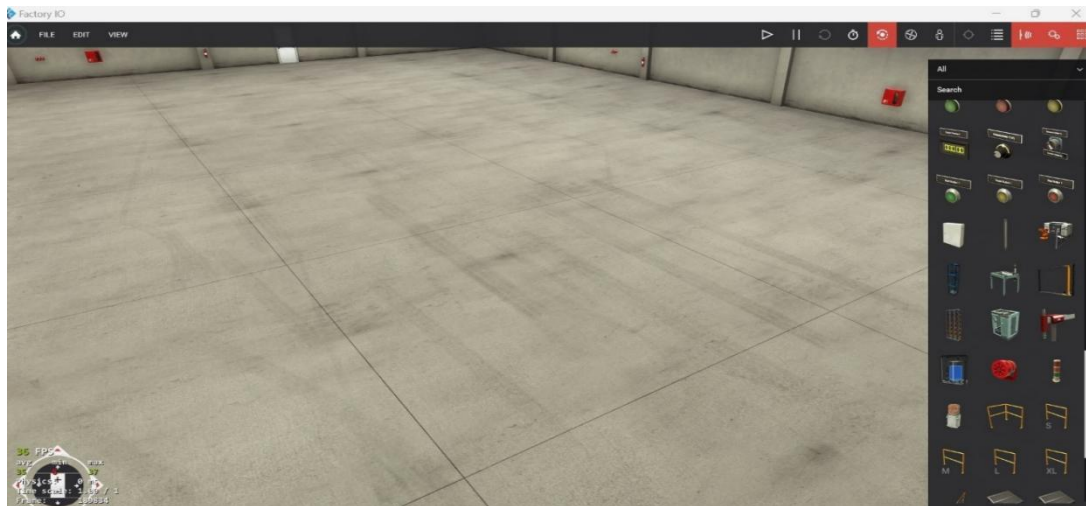


Figure III.21. Factory I/O

### III.7.2 Overview project stimulation

1. The first step is to go to the "Palette" tab, then select the water tank, which represents the system on which the cascade control will be applied. The tank includes two sensors: a level sensor to measure the liquid level in the tank and a flow sensor to measure the flow rate of the liquid. The system also includes two control valves: one for filling the tank and the other for draining it, as illustrated in the figure below.[19]

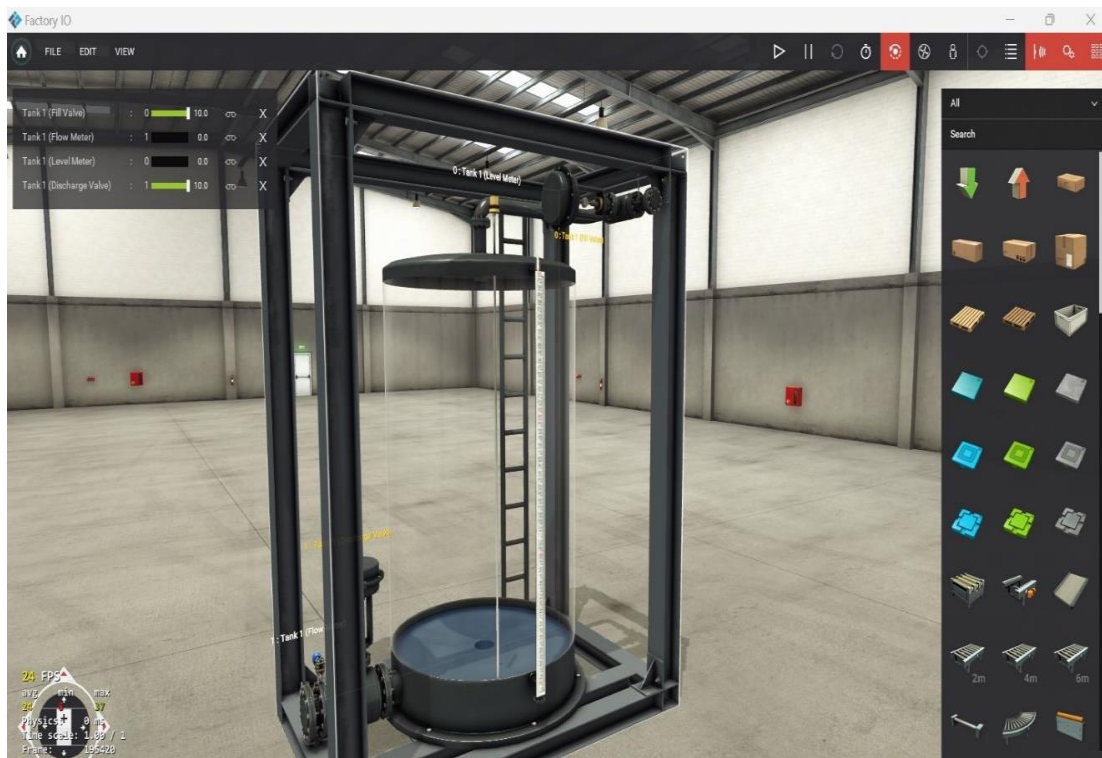


Figure III.22. Overview project stimulation

## Chapter III Programming and Monitoring in TIA Portal and Simulation with Factory

2. The second step is to configure the (Factory I/O) driver and establish the connection with PLCSIM.

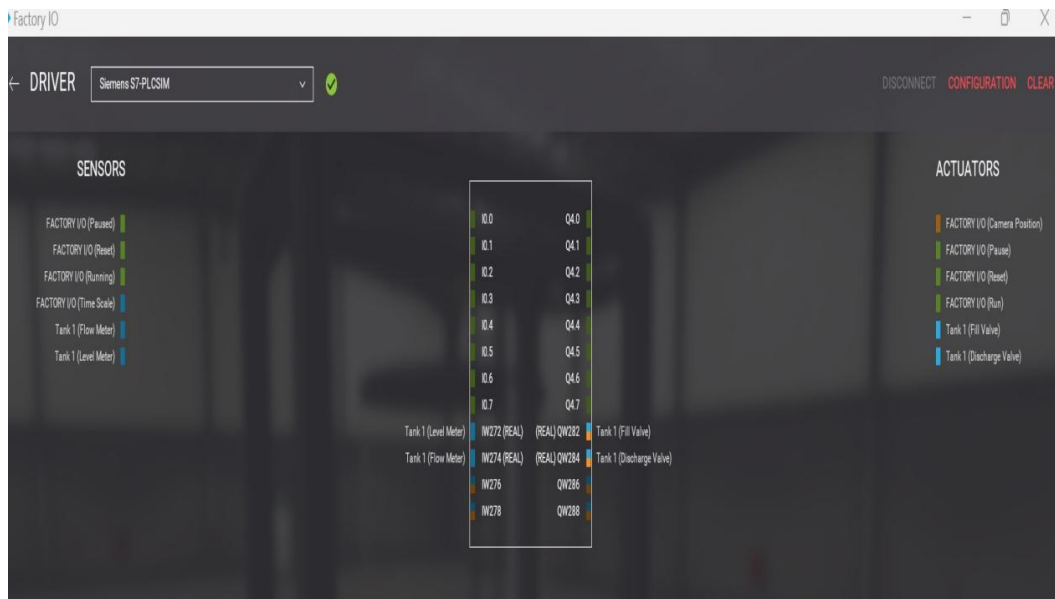


Figure III.23. Driver Menu

3. After a successful configuration, we start the simulation and monitor the results.

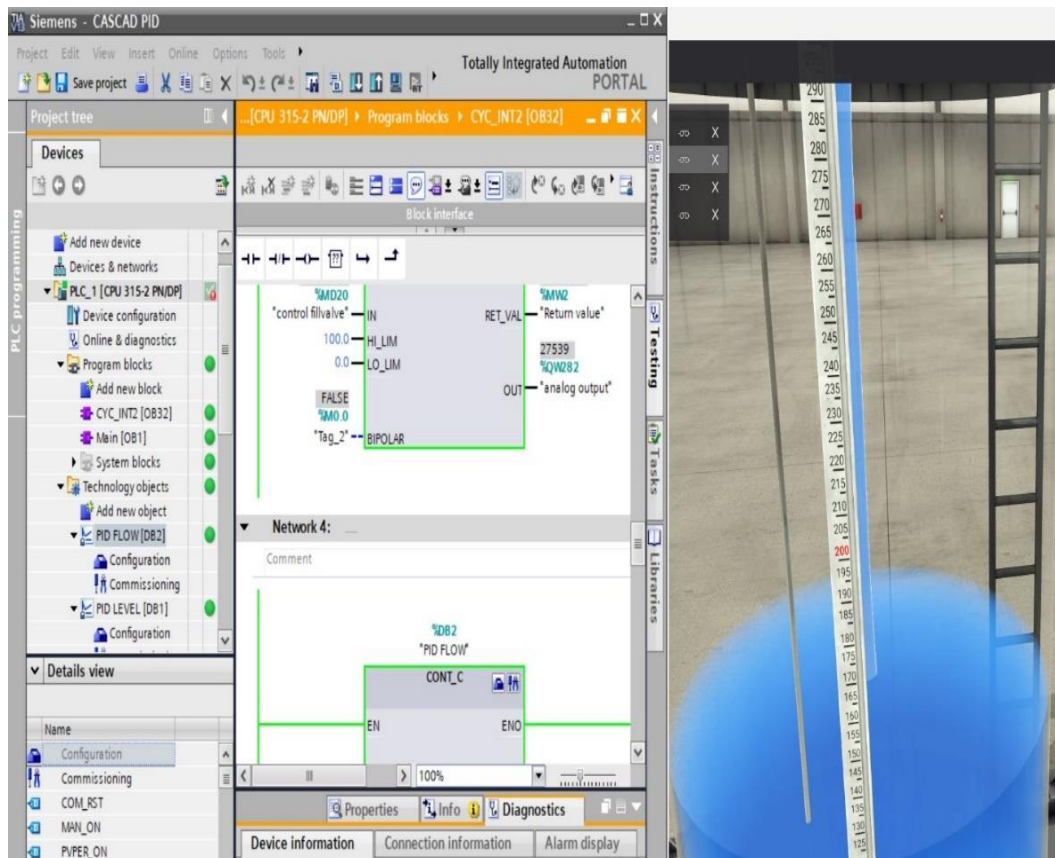


Figure III.24. simulation view

### **III.8. Case Study**

In order to achieve more precise cascade control, the following experiment is conducted: a fixed setpoint of 65% is applied to the level controller, and the PID parameters ( $K_o$ ,  $K_i$ ,  $K_d$ ) are iteratively adjusted. The tuning process relies on real-time Trend monitoring to observe system response and evaluate the effect of each parameter adjustment.

#### **III.8.1 Experiment Procedure**

To achieve more precise cascade control, the following experiment is conducted:

- A **fixed setpoint of 65%** is applied to the level controller.
- The PID parameters ( **$K_p$ ,  $K_i$ ,  $K_d$** ) are **iteratively adjusted** for both controllers.
- The tuning process is performed using **real-time Trend monitoring** within TIA Portal or HMI to observe the response of process variables (PV), setpoints (SP), and control outputs (CV).

##### **III.8.1.1 Tuning Strategy**

###### **Description:**

The tuning strategy is a method to find the best settings for a PID controller, which helps control processes accurately and responsively. It aims to reduce overshoot, settling time, and steady-state error to keep the controlled variable close to the desired level. In TIA Portal, tuning can involve manual adjustments, the Ziegler-Nichols technique, or autotuning features. Steps include analyzing the process, choosing a PID block, selecting a tuning method, refining parameters based on feedback, and validating performance.

###### **Step 1: Tune the Inner Loop (Flow Controller)**

- The level controller is temporarily disabled.
- A constant flow setpoint is applied (50%).
- Initial PID values:  $K_p = 0,8$   $K_i = 18s$ ,  $K_d = 2s$
- Adjust parameters based on response:
  - Increase  $K_p$  to reduce rise time.
  - Adjust  $K_i$  to eliminate steady-state error.
  - Introduce  $K_d$  only if overshoot or oscillation is observed.

###### **Step 2: Tune the Outer Loop (Level Controller)**

- The flow controller is fixed with the tuned parameters.
- The level controller is activated to generate the flow setpoint.

- Initial PID values:  $K_p = 1$ ,  $K_i = 20$ ,  $K_d = 2$
- Adjustments are made based on:
  - Rise time
  - Overshoot
  - Settling time

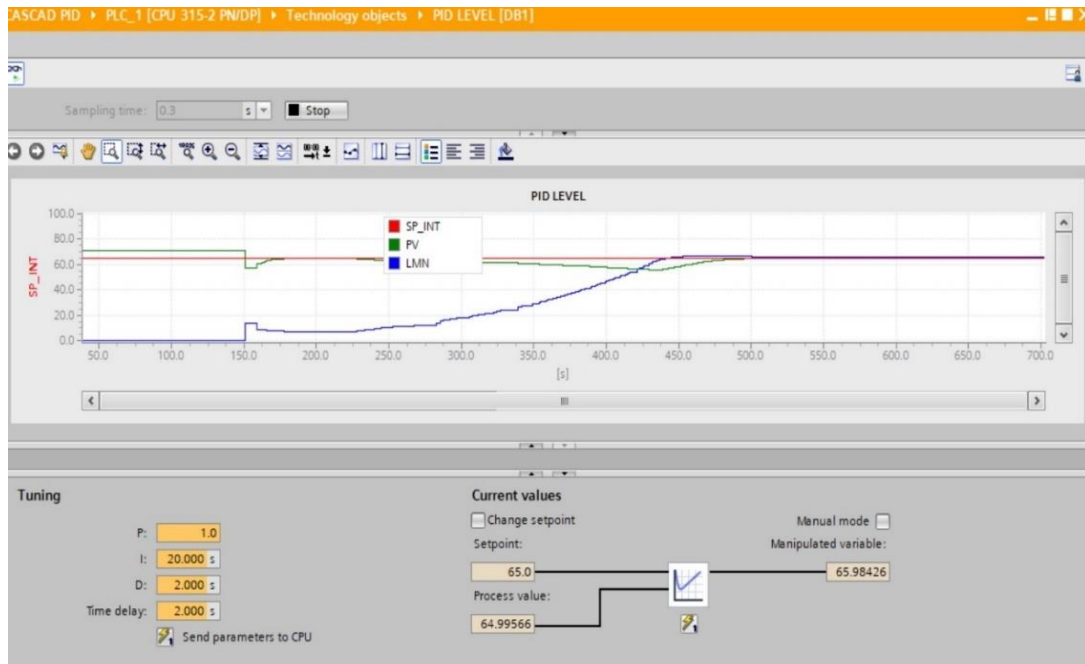


Figure III.25. PID LEVEL Control

### III.8.2. Trend Analysis:

- **SP\_INT (Red):** Represents the set-point of the level control, which is set to **65%**.
- **PV (Green):** The actual process value (measured level), which starts from 0 and rises gradually toward the setpoint.
- **LMN (Blue):** The PID output (manipulated variable), which is sent to the valve to regulate the level.

### III.8.3. Observations:

- At the start of the experiment, PV increases progressively as the valve opens.
- The process reaches the setpoint ( $\sim 65\%$ ) around  $t = 65s$ , with a slight **overshoot**, then stabilizes.
- At around  $t = 140s$ , a disturbance or setpoint change occurs, causing PV to drop, but the controller corrects it quickly.
- **LMN** was high at first and gradually reduced as the process stabilized.

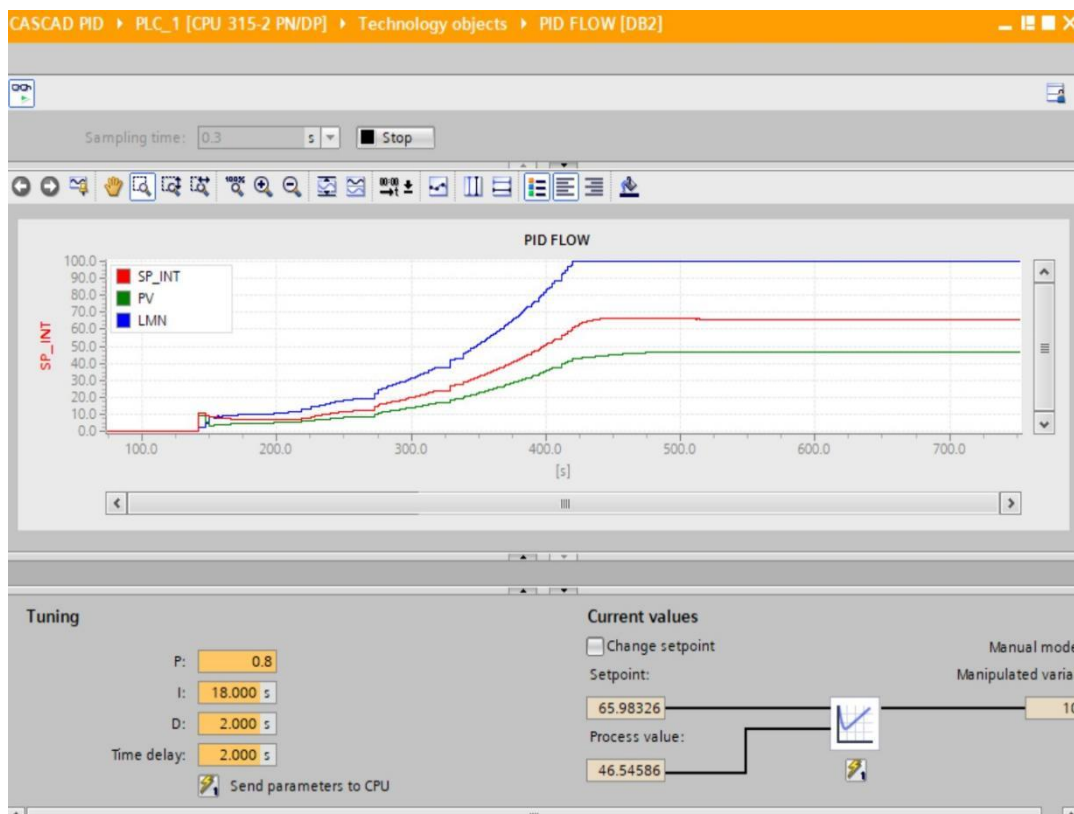


Figure III.26. PID FLOW Control

### III.8.4. Trend Analysis:

- **SP\_INT (Red):** Flow setpoint, initially high, then drops sharply to ~10.
- **PV (Green):** Actual flow value, follows SP with noticeable oscillation.
- **LMN (Blue):** Controller output to the actuator (valve or pump).

### III.8.5. Observations:

- At  $t = 0$ , a sudden decrease in SP causes the system to react strongly.
- PV rises rapidly, with some **initial oscillations**, then settles gradually.
- LMN fluctuates significantly in the first 50 seconds, then stabilizes.

#### NB:

- In cascade control, **flow** is the inner loop and should be stable and well-tuned before activating the **level** control (outer loop).
- Proper tuning of both loops ensures a smooth, responsive system with minimal errors and oscillations.

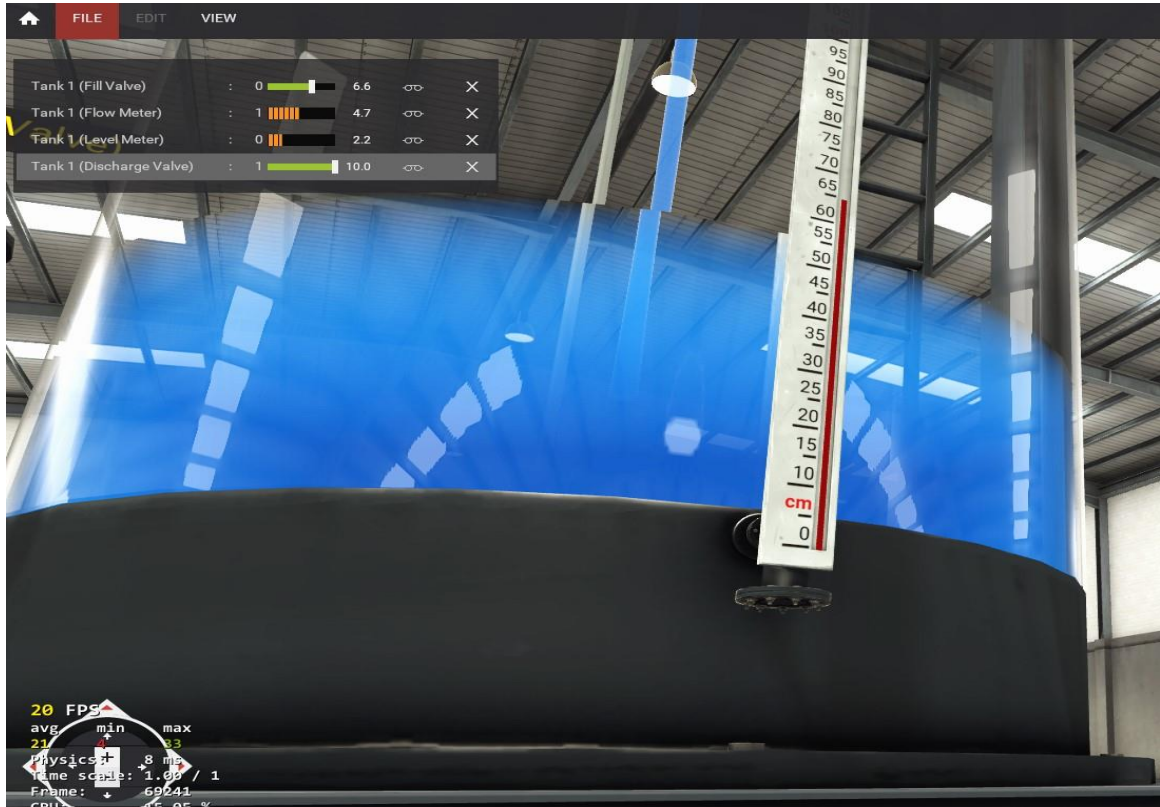


Figure III.27. Simulation factory view.

### III.9. Conclusion

This chapter presents an overview of the S7-300 programmable logic controller, specifically the CPU 315-2 PN/DP, along with the TIA Portal V13 programming environment and the Factory I/O simulation software. We also detailed the main steps involved in creating and configuring a complete automation project using these tools. Furthermore, we introduced the basic principles of PID control and explained how cascade PID regulation can be implemented for more precise and stable process control. Finally, a brief description of and its integration with Factory I/O was provided.

In the following chapter, we will describe the structure and components of our process station and present the development and implementation phases of our automated control system.

## General conclusion

Throughout this thesis, we explored the theoretical and practical aspects of cascade PID control implementation using Siemens S7-300 PLCs and the TIA Portal environment, with simulation and visualization handled by Factory I/O. The project focused on a level-flow process system—a common scenario in industrial process control—where we established a two-loop cascade control structure to enhance the precision and responsiveness of the system.

Our main contributions include:

1. **Theoretical Foundation:** We provided a thorough explanation of PID controllers, the rationale behind cascade control, and the conditions under which it offers superior performance, particularly for systems where process variables exhibit different time constants.
2. **System Design and Integration:** We developed a full-scale control architecture using Siemens TIA Portal, from hardware configuration and network setup to signal declaration and LAD programming of both primary and secondary PID loops.
3. **Real-Time Simulation and Testing:** Using Factory I/O, we simulated realistic industrial scenarios involving water level and flow regulation. The system's behavior was evaluated under different disturbances and setpoint tracking requirements.

The simulation results demonstrate the advantages of cascade control. The secondary loop (flow control) significantly reduces the effect of disturbances before they reach the primary loop (level control), leading to faster stabilization and improved overall control performance. Compared to a single PID loop, the cascade strategy resulted in lower overshoot, shorter settling time, and enhanced robustness.

From an educational and engineering standpoint, this project provided valuable insights into:

- Practical PID tuning and loop integration,
- Real-world simulation of industrial processes,
- The use of modern PLC-based tools for automated control system design.

Despite the positive outcomes, several limitations were identified. The tuning of cascade controllers, especially when performed manually, requires deep knowledge of process dynamics and may be time-consuming. Moreover, integrating additional real-time sensors and

actuators, or working with real industrial equipment, would further validate the effectiveness of the developed system.

In future work, we recommend:

- Incorporating advanced control strategies, such as fuzzy logic, model predictive control (MPC), or neural network-based PID tuning,
- Extending the system to include networked control using SCADA or IoT-based platforms,
- Applying the methodology to real hardware setups in industrial pilot plants.

**In conclusion**, this thesis presents a practical and structured approach to the design and implementation of a cascade control system using modern industrial automation tools. It also contributes to academic development in the field of process control and industrial instrumentation, both in Algeria and in broader contexts.

## **Abstract:**

This work presents the implementation of a cascade PID control system for a level-flow process using Siemens S7-300 PLC. The system is programmed in TIA Portal and simulated in Factory I/O. Cascade control improves response time and stability by using two loops: one for flow (inner) and one for level (outer). Results show better performance than single-loop control.

**Keywords:** Cascade control, PID, Siemens S7-300, TIA Portal, Factory I/O.

## **Resume:**

Ce travail présente la mise en œuvre d'un système de régulation en cascade PID pour un processus niveau-débit en utilisant un API Siemens S7-300. Le système est programmé dans TIA Portal et simulé avec Factory I/O. La régulation en cascade améliore la stabilité et le temps de réponse par rapport à une régulation simple.

**Mots-clés :** Régulation en cascade, PID, Siemens S7-300, TIA Portal, Factory I/O

## **ملخص :**

تناول هذا المشروع تصميم وتنفيذ نظام تحكم PID متسلسل للتحكم في مستوى وتدفق المياه باستخدام المتحكم المنطقي القابل للبرمجة Siemens S7-300. تم تطوير البرنامج باستخدام منصة TIA Portal ، بينما تمت المحاكاة باستخدام Factory I/O. يسمح التحكم المتسلسل باستخدام حلقتين (داخليّة للتدفق وخارجيّة للمستوى) بتحسين الاستجابة وتقليل تأثير الاضطرابات مقارنة بالتحكم التقليدي.

**الكلمات المفتاحية:** التحكم المتسلسل، منظم PID ، سيمنس S7-300 ، TIA Portal ، Factory I/O.

## Bibliography :

1. Åström, K.J., & Hägglund, T. (2006). *Advanced PID Control*. ISA – The Instrumentation, Systems, and Automation Society.
2. Seborg, D. E., Edgar, T. F., & Mellichamp, D. A. (2010). *Process Dynamics and Control* (3rd ed.). Wiley.
3. Dorf, R. C., & Bishop, R. H. (2017). *Modern Control Systems* (13th Edition). Pearson Education.
4. .ALLAOUI Ahmed Tarek BENALIA Brahim BOUHELAL Abdessalam Ajustement d'un régulateur SPWPID par la méthode méta-heuristique PSO] Univ Kasdi Marbah Ouargla Mémoire 2eme master 2019/2020
5. Bequette, B. W. (2003). *Process Control: Modeling, Design, and Simulation*. Prentice Hall.
6. GHANEM Ilyes, TEMMAM Zoubir. Mémoire [Etude et identification d'un procédé decontrôle de niveau] Univ Kasdi Marbah Ouargla Mémoire 2eme master 2017/2018
7. Bequette, B. W. (2003). *Process Control: Modeling, Design, and Simulation*. Prentice Hall.
8. Marlin, T. E. (2000). *Process Control*. McGraw-Hill.
9. Siemens AG. (n.d.). *Cascade Control in PCS 7*.  
[https://eng.libretexts.org/Bookshelves/Industrial\\_and\\_Systems\\_Engineerig](https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineerig)
10. Luyben, W. L. (2007). *Process Modeling, Simulation, and Control for Chemical Engineers*. McGraw-Hill.
11. Dorf, R. C., & Bishop, R. H. (2011). *Modern Control Systems*. Pearson
12. Ogata, K. (2010). *Modern Control Engineering*. Pearson.
13. [www.wikipedia.com](http://www.wikipedia.com)
14. Slim, BENSOUUD, "LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API)
15. [www.siemens.com](http://www.siemens.com)
16. Frank D. Petruzella, « Programmable Logic Controllers », McGraw-Hill, 2005
17. CHERGUI Younes, « Commande et supervision de la station FESTO
18. Présent et soutenu par, SOLTANI KHALED Le : juillet 2019, Simulation of a product packaging system by FACTORY I/O and TIA Portal V13 software
19. [www.factory i/o.com](http://www.factory.io.com)