



Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Misan
College of Engineering
Department of Electrical Engineering



Design and Implementation of SCADA system in industrial process

A graduation project submitted to the **Department of Electrical Engineering**, in partial fulfillment for the requirements for the award of the degree of Bachelor of **Electrical Engineering**

By

1-Ahmed Sabah Hanoun 2-Ali Jassim Muhaybis

3-Caesar Mohammed Omran 4-Fatima Saad Saghir

5-Athraa Kadhim Muftin

SUPERVISED BY

1- Asst. Lec. Al-Hussein Mohammed Jumaah

2- Lec. Hisham Dawood Salman

Maysan, Iraq

2025

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

((وَلَقَدْ أَتَيْنَا دَاوُودَ وَسُلَيْمَانَ عِلْمًا ۖ وَقَالَا الْحَمْدُ لِلَّهِ الَّذِي فَضَّلَنَا عَلَى كَثِيرٍ مِّنْ عِبَادِهِ

الْمُؤْمِنِينَ)) النمل آية (١٥)

صدق الله العلي العظيم

Supervisors Certification

I certify that the preparation of this project entitled “Design and Implementation of SCADA system in industrial process” prepared by **“Ahmed Sabah Hanoun” “Ali Jassim Muhaybis” “Caesar Mohammed Omran” “Fatima Saad Saghir” “Athraa Kazeam Maftean”** was under my supervision at college of engineering, Electrical Engineering Department, University of Misan in partial fulfilment of the requirements for the degree of Bachelor of Science in Electrical Engineering.

Ass.Lec.Al Hussein Mohammed jumaah

Signature: Date: / / 2025

Les. Hisham Dawood Salman

Signature: Date: / / 2025

In review of the available recommendation, I forward this project for debate by the Examination Committee.

Committee Certification

We certify, as an examining committee, that we have read this project report entitled "**Design and Implementation of SCADA system in industrial process**" examined the students "**Ahmed Sabah Hanoun**" "**Ali Jassim Muhaybis**" "**Caesar Mohammed Omran**" "**Fatima Saad Saghir**" "**Athraa Kazeam Maftean**" in its contents and found the project meets the standard for the degree of Bachelor of Science in Electrical Engineering.

Signature:

Signature:

Name:

Name:

Date:

Date:

Signature:

Signature:

Name:

Name:

Date:

Date:

Signature:

Name:

Date:

DEDICATION

Every challenging work needs self-efforts as well as guidance of elders especially those who were very close to our heart.

Our humble efforts we dedicate to our sweet and loving **Father & Mother**, whose affection, love, encouragement and prayers of day and night make me able to get such success and honor, along with all hard working and respected teachers

ACKNOWLEDGEMENT

*Praise be to **ALLAH** who enabled us to complete this work under his benediction.*

*Cordial thanks and deepest gratitude to our supervisor (**Asst. Lec. Al-Hussein Mohammed Jumaah,***

***Lec. Hisham Dawood Salman**) for being assistance and for the continuous encouragement and suggestions from him which helped us to successfully complete this project.*

*In addition, we would like to express my thanks to the head of the Electrical Engineering Department (**Dr. Mohammed Khalaf Jumaah**) for his continuous support and encouragement.*

We would like to express our thanks to our University for their encouragements and anyone helped us. Finally, sincere gratitude and appreciation and love to our families for their encouragements and supports and for all things they have given us.

ABSTRACT

This project presents the design and implementation of a Supervisory Control and Data Acquisition (SCADA) system for monitoring and controlling an industrial process using real-world hardware and software tools. The goal was to build a low-cost, flexible, and scalable control system by integrating components such as an XGB PLC, an Arduino microcontroller, a Variable Frequency Drive (VFD), and a Human Machine Interface (HMI). Communication between these devices was accomplished using reliable industrial protocols, namely Modbus RTU for serial communication and OPC UA for Ethernet-based data exchange.

The Arduino module was programmed to acquire real-time data from analog sensors and send it to the PLC via RS-485 using the Modbus RTU protocol. The PLC was programmed using XG5000 software to process the input signals, apply control logic, and communicate over TCP/IP with the KepServerEX platform. KepServer acted as a middleware OPC server, collecting data from the PLC and making it accessible to the HMI through OPC UA. The HMI, designed using Siemens TIA Portal, provided a graphical interface for visualizing system data and sending control commands to the PLC.

This system allowed real-time monitoring of sensor readings, motor control via the VFD, alarm notifications, and data logging with precise time and date stamps. It demonstrates how different

industrial technologies can be combined to implement a complete SCADA solution.

LIST OF ABBREVIATIONS

Abbreviations	Definition
PLC	Programmable Logic Controller
IDE	Integrated development environment
SCADA	Supervisory control and data acquisition
RS485	Is a standard defining the electrical characteristics of drivers and receivers for use in serial communications systems.
HMI	Human Machine Interface
Modbus	Modicon (now Schneider Electric) Bus
RTU	Remote Terminal Unit
TIA Portal	Totally Integrated Automation
TCP/IP	Transmission Control Protocol/Internet Protocol
OPC UA	Open Platform Communications Unified Architecture

LIST OF FIGURES

Figure (4.1)	Relay	26
Figure (4.2)	Arduino	26
Figure (4.3)	Rs-485	27
Figure (4.4)	Router	27
Figure (4.5)	WaterPump	28
Figure (4.6)	Pump	28
Figure (4.7)	Filter	29
Figure (4.8)	3ph-Induction Motor	29
Figure (4.9)	LSIS IC5 VFD	29
Figure (4.10)	BreadBoard	30
Figure (4.11)	PLC XGB	30
Figure (4.12)	Emergency Button	31
Figure (4.13)	Push Buttons	31
Figure (4.14)	Siemens VDC24	32
Figure (4.15)	Ultra Sonic Sensor	32
Figure (4.16)	Arduino IDE	33
Figure (4.17)	XG5000	36
Figure (4.18)	Refill operation	37
Figure (4.19)	Filter operation	38
Figure (4.20)	Emptying operation	38
Figure (4.21)	VFD operation	39
Figure (4.22)	Registers of VFD	40
Figure (4.23)	PLC to Router Connection Configuration	42
Figure (4.24)	KEPServer	41

Figure (4.25)	KEPServer data store	42
Figure (4.26)	SCADA Configurations	43
Figure (4.27)	SCADA Tags	43
Figure (4.28)	SCADA Screen	43
Figure (4.29)	actual project pictures showing all components	44

TABLE OF CONTENTS

Dedication	iv
Acknowledgement	II
Abstract	Vi
List of Figures	Vii
Table of Contents	Viii

Chapter 1: Introduction

1.1 Introduction.....	2
1.2 SCADA system.....	3
1.2.1 Field Data Interface device.....	3
1.2.2 Communication network.....	4
1.2.3 Central Host Computer.....	5
1.2.4 Operator workstation and software component.....	5
1.3 Scope of report	6

Chapter 2: SCADA system

2.1 Introduction about SCADA.....	8
2.2 Using SCADA system.....	8
2.3 SCADA system contents.....	9
2.4 SCADA hardware.....	10
2.5 SCADA software.....	10
2.6 Security.....	11

2.7 Alarm Handling.....	12
2.8 SCADA system benefits.....	13

Chapter 3: Methodology

3.1 Based on PLC.....	15
3.1.1 Introduction.....	15
3.1.2 Architecture of PLC.....	15
3.1.3 Advantages of PLC.....	17
3.1.4 PLC versus Relay.....	18
3.1.5 PLC operations.....	19
3.1.6 PLC terminology.....	19
3.1.7 Ladder logic and it mean order.....	20
3.2 Industrial Ethernet.....	21
3.2.1 Introduction.....	21
3.3 Modbus RTU Introduction.....	22
3.4 Router TP-Link.....	23
3.4.1 Introduction.....	23
3.5 AC drive (LSIS IC5 inverter).....	24

Chapter 4 : Practical

4.1 Hardware parts.....	26
4.2 Software parts.....	33
4.2.1 Arduino Programming.....	33
4.2.2 Arduino code.....	33

4.2.3 PLC Programming.....	36
4.2.3 PLC program Logic.....	37
4.2.4 Protocols and Resister.....	39
4.2.5 HMI Tags and Configurations.....	43
4.2.6 Final Overview of the Project.....	44
Chapter 5: Conclusion	
5.1 Conclusion.....	46
5.2 Future Work.....	47

Chapter 1

Introduction

Chapter 1

1.1 Introduction :

A SCADA (Supervisory Control and Data Acquisition) system is a type of industrial control system (ICS) used to monitor and control processes in industries like manufacturing, energy, water treatment, and more. It collects real-time data from sensors and devices spread across a system (like pipelines, factories, or power plants) and presents it in a central control interface. This allows operators to monitor performance, detect issues, and make adjustments remotely to optimize the system's operations. SCADA systems often have data historians, which are databases that store historical data for trend analysis and reporting. SCADA systems provide operators and management with access to real-time data and detailed reports, which helps improve decision-making processes. With access to accurate data, operators can make informed decisions about system adjustments, maintenance scheduling, and resource allocation, leading to more efficient operations. This data can be used for performance analysis, compliance reporting, and predictive maintenance. In industrial environments, SCADA systems are vital for improving operational efficiency, ensuring safety, minimizing downtime, and enabling better decision-making. By automating control, providing real-time data, and enabling remote monitoring, SCADA systems help industries optimize production, reduce costs, and maintain regulatory compliance. Whether it's in manufacturing, power generation, water treatment, or any other industrial sector, SCADA plays a central role in modern industrial automation and control.

1.2 SCADA System :

SCADA (Supervisory Control and Data Acquisition) is a system used for monitoring and controlling industrial processes, infrastructure, and facility-based operations. SCADA systems are crucial in industrial settings because they provide centralized control, real-time monitoring, and data analysis, which help improve efficiency, safety, and decision-making. SCADA is used because it helps industries operate more efficiently, safely, and with greater control. It automates processes, improves decision-making, provides real-time visibility, and enables proactive maintenance, all of which contribute to improved performance, lower costs, and safer operation. Industrial that Rely on SCADA is Power Grid Management, Water Treatment , Manufacturing, Oil and Gas, Transportation and Building Automation..etc.

1.2.1 Field Data Interface Devices :

In a SCADA (Supervisory Control and Data Acquisition) system, Field Data Interface Devices play a crucial role in bridging the communication between field devices (such as sensors, actuators, and controllers) and the central SCADA system. These devices are responsible for collecting real-time data from the field and transmitting it to the SCADA system for monitoring and control purposes. The relationship between Field Devices, RTUs and PLCs in a SCADA system plays an essential role in monitoring and control of industrial processes. Both RTUs and PLCs collect data and

execute control commands, PLCs are often used for more localized, complex control tasks, while RTUs typically handle communication over longer distances and remote locations. In some systems, RTUs and PLCs can work together at different layers of the system. For instance, PLCs might control specific processes locally (e.g., inside a plant), while RTUs are used to gather data from various remote locations and relay that information to the SCADA system. PLCs and RTUs often use different protocols, with RTUs primarily focused on remote communication, while PLCs are optimized for real-time, local control. The SCADA system receives the data from both RTUs and PLCs and presents it to operators. It also sends control signals back to RTUs or PLCs to make real-time adjustments to the process, such as changing settings or turning equipment on or off.

1.2.2 Communication Network :

The communication network is a critical component of a SCADA (Supervisory Control and Data Acquisition) system enabling the transmission of data between field devices (such as sensors, PLCs, and RTUs) and the central control stations. This network ensures continuous and secure data transfer in real-time, allowing operators to monitor processes and make immediate decisions. There are different types of Communication Network such as Wired networks (Internet/Ip and Fiber Optic) and Wireless Networks (Microwave , Satellite and Cellular Network). It is also important to know that it uses different types of protocols such as ModBus ,DNP3, IEC, OPC. These protocols allows us to connect different types of system

together. A common example is connecting an LS PLC To a TIA Portal via Rs232.

1.2.3 Central Host Computer :

The central host computer in a SCADA system is the core component responsible for processing and managing the data received from remote field devices (such as sensors, RTUs, and PLCs). It acts as the brain of the SCADA system, where data collection, analysis, storage, and visualization take place. The central host computer is typically located in the control room or a centralized facility, and it enables operators to monitor and control industrial processes in real time. It is important to note that Central Host Computer And HMI are not the same and they work together in a SCADA system. The central host computer sends processed data, reports, and system status to the HMI for visual display to the operator. In turn, HMI allows the operator to input commands which are sent back to the central host computer or directly to field device. The central host computer acts as the "brain" of the system, performing analysis and decision-making, while the HMI serves as the user interface for operators to view and interact with the system. The central host computer provides vital data to the HMI, which displays it for the operator, enabling continuous real-time monitoring and control of the system.

1.2.4 Operator Workstations and Software components :

An Operator Workstation is a computer or set of devices used by operators to view information and interact with the SCADA system. These workstations enable operators to monitor the system in real-

time, input control commands, and receive alerts and notifications about system status. It provides Data Display and Alarm Management and system interaction And reporting and analysis. A Software Computer is the computer that hosts the SCADA software, including control and monitoring applications This computer plays a key role in managing the SCADA system, as it coordinates between operator workstations, the central host computer, and field devices. Its function is so important for SCADA system. It runs the SCADA software and store data and analysis it, It also provides communication management and alarm management. Both the Operator Workstation and Software Computer work together to ensure the system operates efficiently, allowing operators to monitor, analyze, , and control processes effectively.

1.3 Scope of report :

This report examines the impact of SCADA system in industrial and how important it is. The report focused mainly on what SCADA actually are and the benefits of it. It also focused on other things used like the hardware and software parts. The report aims to make the reader have an idea of what the SCADA system are and how can we use it. In further chapters, We'll explain the advantages and disadvantages of SCADA system, PLCs and RTUs. Also mentioning the HMI and why we need it. The report also will include the project philosophy.

Chapter 2

SCADA system

Chapter 2

2.1 Introduction about SCADA

It is a system for collecting, monitoring and controlling data, and the word SCADA (SCADA) is an abbreviation for (Supervisory control and data acquisition). The SCADA system consists of a set of software and hardware components, which in turn allows institutions and commercial and industrial organizations to perform several tasks, such as Control and control of production and industrial processes locally or internationally. Monitoring, data collection, and processing. Direct handling of devices, sensors, sensors, pumps, and valves, through computer operating systems software. Record events and save them

2.2 Using SCADA System

SCADA systems are essential in various sectors for efficient monitoring and control. Key applications include:

- **Energy Management:** Monitoring and controlling power generation and distribution networks to enhance efficiency and reliability.
- **Water Management:** Tracking water supply networks, treatment plants, and wastewater processes to ensure quality and compliance.
- **Manufacturing:** Automating processes, monitoring production lines, and managing inventory for better operational efficiency.
- **Oil and Gas:** Managing pipeline operations, refining processes, and ensuring safety through real-time data.

- **Transport Systems:** Overseeing traffic signals, public transportation systems, and logistics for improved flow and safety.
- **Building Automation:** Controlling HVAC, lighting, and security systems to enhance comfort and energy efficiency.

SCADA systems ultimately enhance productivity and safety across various

2.3 SCADA System Contents

1. **Central Server:** Acts as a data aggregation point for analysis and system management.
2. **Human-Machine Interface (HMI):** Allows users to monitor and interact with the system through a graphical interface.
3. **Remote Terminal Units (RTUs):** Collect data from sensors and devices on-site and transmit it to the server.
4. **Programmable Logic Controllers (PLCs):** Control devices and processes based on input data and execute programmed instructions.
5. **Communication Network:** Enables data and information transfer between different components, either wired or wireless.
6. **Database:** Stores historical data and allows for analysis to facilitate decision-making.
7. **Security System:** Ensures data protection and confidentiality of operations against attacks or unauthorized access.

2.4 SCADA Hardware

A SCADA system consists of various hardware components that work together for effective monitoring and control. Key parts include:

2.4.1 Sensors: Collect data from the surrounding environment, such as pressure, temperature, and fluid levels.

2.4.2 Programmable Logic Controllers (PLC): Process the data from sensors and execute control commands.

2.4.3 Human-Machine Interfaces (HMI): Provide graphical user interfaces that allow operators to monitor and control systems.

2.4.4 SCADA Servers: Store and analyze data collected from workstations and manage communication among components.

2.4.5 Communication Devices: Include devices for transmitting data between SCADA devices, such as modems, networks, and wireless communication technologies.

2.5 SCADA Software

SCADA software comprises several essential components that facilitate monitoring and control of processes. Key parts include:

2.5.1 Data Acquisition: Collects real-time data from field devices and sensors.

2.5.2 Database Management: Stores historical data for analysis and reporting, often using SQL databases.

2.5.3 HMI Software: Provides visual interfaces for operators to interact with the system, displaying data trends and alerts.

2.5.4 Alarm Management: Alerts operators to abnormal conditions, allowing for quick response to potential issues.

2.5.5 Reporting Tools: Generate reports based on collected data for performance analysis and compliance.

2.5.6 Control Modules: Allow operators to issue commands to PLCs and other devices to manage processes.

These software components ensure efficient data management, user interaction, and system control in SCADA applications.

2.6 Security

Security is crucial in protecting personal and organizational data. Here are key points to consider:

2.6.1 Data Protection: Use encryption for sensitive information and ensure regular backups.

2.6.2 Access Control: Implement strong passwords, multi-factor authentication, and limit access based on roles.

2.6.3 Network Security: Keep software updated, use firewalls, and monitor network traffic for suspicious activity.

User Education: Train employees on recognizing phishing attempts and safe browsing practices
Incident Response: Develop a plan for responding to security breaches, including communication and recovery strategies.

Staying informed and proactive is essential to maintaining security in any environment.

2.7 Alarm Handling

Effective alarm handling is crucial for ensuring safety and security. Here are key steps:

2.7.1 **Prioritization:** Assess the severity of each alarm to determine response priorities.

2.7.2 **Initial Investigation:** Gather necessary information to understand the cause and context of the alarm.

2.7.3 **Prompt Response:** Take immediate action based on the alarm level, whether reporting it or implementing corrective measures.

2.7.4 **Documentation:** Record all details of the incident from the alarm to the response for future review and procedure improvement.

2.7.5 **Evaluation and Improvement:** After managing the alarm, assess the process and identify potential improvements to reduce future risks.

Effective alarm response contributes to stronger and more effective safety procedures

2.8 SCADA System Benefits

2.8.1 Continuous Monitoring: SCADA systems enable real-time monitoring of operations, facilitating immediate detection of issues.

2.8.2 **Efficiency Improvement:** Reduces the need for manual interventions, enhancing productivity and speeding up operations.

2.8.3 **Data Collection:** Gathers comprehensive information for performance analysis and informed decision-making.

2.8.4 **Incident Response:** Develop a plan for responding to security breaches, including communication and recovery strategies. Staying informed and proactive is essential to maintaining security in any environment.

Chapter 3

Methodology

Chapter 3

3.1 Based on PLC :

3.1.1 Introduction :

The programmable logic controller, or PLC, is ubiquitous in every kind of process and manufacturing industry today. PLCs were initially designed to replace electromechanical relay systems in order to offer a simpler solution for modifying the operation of a control system. Rather than having to rewire a large bank of relays, a quick download from a PC or programming device enables changes to the control logic in a matter of seconds. The majority of PLCs today are modular, allowing the user to add an assortment of functionality including discrete and analog inputs and outputs, PID control, position control, motor control, serial communication, and high-speed networking. Compared to older technologies such as relay banks, the PLC is far easier to troubleshoot and maintain, more reliable, more cost-effective, and far more versatile..

3.1.2 Architecture of PLC :

The term PLC architecture refers to the design specification of the various PLC hardware and software components and the how they interact with one another to form the overall PLC system. The architecture of a PLC is based on the same principles of that used in standard computer architecture. However, PLC architecture does differ because the design is based around providing high reliability, immunity to harsh industrial environment, ease of maintenance and access to large amounts of peripheral inputs and outputs. The 3 distinct types of PLC architecture available for use in industrial

automation are known as fixed, modular and distributed. The terminology surrounding PLC types can vary between PLC manufacturers, especially when talking about fixed PLCs. There is also crossover between PLC types with some fixed type PLCs having modular type features and some modular type PLCs having distributed type features.

1- Power Supply : The power supply accepts and regulates electrical voltage in the PLC system. It converts the electricity into a signal voltage used by the PLC processor and other modules to send and receive commands, monitor automated machines, and communicate with other systems.

2. Processor : The processor module typically holds one or more microprocessors acting as the PLC's brain. The CPU performs control functions, computes, and tells the other PLC components what to do and when to do it. The processor unit also contains the system memory required to store programs and additional command-line information.

3. Communication Card : Communication modules come in a wide variety of industry standards. They allow PLCs to communicate with each other and the systems they automate. From computer systems to environmental controls, the communication card acts as the voice and ears of the PLC system.

4. Input/Output (I/O Cards) : I/O cards connect your PLC unit to other devices in your facility. These devices include control valves, motor controls, pressure transmitters, and monitoring systems. There are two kinds of I/O cards — analog and digital. Analog PLC

components use continuous electrical signals for operation, while digital I/O cards use non-continuous electrical signals.

3.1.3 Advantages of PLC :

- Rugged and designed to withstand vibrations, temperature, humidity and noise
- PLC has a lot of contacts and low cost and safe
- It has a very faster scan time, it has a fast operating time
- A wide range of control application
- It has capable to communicate with a computer in the plant
- It has great computational capabilities
- It has shorter training time required
- It has a small physical size
- It has project cost can be accurately calculated
- It has supervisory control capability
- PLCs are easily programmed and it was relatively easily understood programming language
- Have interfacing for input and output already inside the controller
- One single programmable logic controller can easily run many machines so it is flexible
- It has high-speed counters

3.1.4 PLC versus relay :

There are several key differences between PLCs and relays:

Complexity: PLCs are more complex than relays, and they are capable of performing a wider range of tasks. Relays are simple devices that are designed to perform a specific task, while PLCs are digital computers that can be programmed to perform a wide range of tasks.

Programming: PLCs are programmed using industrial programming languages, while relays do not require programming. **Functionality:** PLCs can be used to control and monitor a wide range of processes and systems, while relays are typically used to control a specific aspect of a process or system.

Cost: PLCs are generally more expensive than relays, especially for smaller-scale applications. However, relays may be more cost-effective for simple tasks or systems.

Reliability: PLCs are generally more reliable than relays, as they are less prone to failure due to their complex design and the use of high-quality components.

PLCs are programmable controllers that are used for industrial automation and control, while relays are electrically operated switches that are commonly used in PLCs to control the flow of electricity to various devices.

3.1.5 PLC operations :

There are four basic steps in the operation of all PLCs; Input Scan, Program Scan, Output Scan, and Housekeeping. These steps continually take place in a repeating loop.

Four Steps In The PLC Operations :

- 1.) Input Scan Detects the state of all input devices that are connected to the PLC.
- 2.) Program Scan Executes the user created program logic.
- 3.) Output Scan Energizes or de-energize all output devices that are connected to the PLC.
- 4.) Housekeeping This step includes communications with programming terminals, internal diagnostics, etc...

3.1.6 PLC terminology :

1. **Input/Output (I/O):** The interfaces through which the PLC communicates with the external world. Inputs are signals coming from sensors, switches, etc., while outputs control devices like motors, lights, or valves.
2. **Ladder Logic (LD):** A graphical programming language used to program PLCs, resembling electrical relay logic diagrams. It's a common method for programming PLCs.
3. **Scan Cycle:** The process by which a PLC continuously reads inputs, processes the program, and updates outputs in a repeating cycle.

4. **Rung:** A line in a ladder diagram that represents a control logic operation, similar to an electrical circuit.

5. **Tags/Variables:** These are used in PLC programming to represent data points such as inputs, outputs, or internal memory variables.

6. **Timers and Counters:** Special instructions in PLC programs used for time delays or counting events. Common types include TON (on delay timer), TOF (off delay timer), and CTU (up counter).

7. **PLC CPU (Central Processing Unit):** The main processing unit of the PLC, responsible for executing the control program and managing inputs/outputs.

8. **Relay Outputs:** Physical relay switches controlled by the PLC outputs to turn devices on or off.

9. **Communication Protocols:** The methods used for communication between PLCs and other devices or systems, such as Modbus, Ethernet/IP, or Profibus.

3.1.7 ladder logic and it mean orders :

Ladder logic was originally a written method to document the design and construction of relay racks as used in manufacturing and process control. Each device in the relay rack would be represented by a symbol on the ladder diagram with connections between those devices shown. In addition, other items external to the relay rack such as pumps, heaters, and so forth would also be shown on the ladder diagram. Ladder logic has evolved into a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. Ladder logic is used to

develop software for programmable logic controllers (PLCs) used in industrial control applications. The name is based on the observation that programs in this language resemble ladders, with two vertical rails and a series of horizontal rungs between them. Ladder diagrams were once the only way to record programmable controller programs, but today, other forms are standardized in IEC 61131-3. For example, instead of the graphical ladder logic form, there is a language called Structured text, which is similar to C, within the IEC 61131-3 standard.

3.2 Industrial Ethernet :

3.2.1 Introduction :

Industrial Ethernet (IE) is the use of Ethernet in an industrial environment with protocols that provide determinism and real-time control. Protocols for industrial Ethernet include EtherCAT, EtherNet/IP, PROFINET, POWERLINK, SERCOS III, CC-Link IE, and Modbus TCP. Many industrial Ethernet protocols use a modified media access control (MAC) layer to provide low latency and determinism. Some microprocessors provide industrial Ethernet support. Industrial Ethernet can also refer to the use of standard Ethernet protocols with rugged connectors and extended temperature switches in an industrial environment, for automation or process control. Components used in plant process areas must be designed to work in harsh environments of temperature extremes, humidity, and vibration that exceed the ranges for information technology equipment intended for installation in controlled environments. The use of fiber-optic Ethernet variants reduces the problems of electrical

noise and provides electrical isolation. Some industrial networks emphasized deterministic delivery of transmitted data, whereas Ethernet used collision detection which made transport time for individual data packets difficult to estimate with increasing network traffic. Typically, industrial uses of Ethernet employ full-duplex standards and other methods so that collisions do not unacceptably influence transmission times.

3.3 Modbus RTU introduction :

Modicon originally developed the open serial protocol based on the master/slave architecture (now client/server) called Modbus RTU. People widely accept it as a serial-level protocol because of its ease of use and reliability. Building Management Systems (BMS) and Industrial Automation Systems (IAS) widely use Modbus RTU. Modbus RTU messages are a simple 16-bit structure with a Cyclic-Redundant Checksum. The simplicity of these messages ensures reliability. Thanks to its simplicity, the basic 16-bit Modbus RTU register structure can pack floating point numbers, tables, ASCII text, queues, and other types of data. This protocol primarily uses RS-232 or RS-485 serial interfaces for communication and is supported by all commercial SCADA, HMI, OPC server, and data acquisition software available in the marketplace. This makes it very easy to integrate Modbus-compatible equipment into new or existing monitoring and control applications.

3.4 Router TP-Link :

3.4.1 Introduction :

A TP-Link router is a device used to establish a network connection for multiple devices and allow them to communicate with each other and access the internet. TP-Link is one of the most recognized manufacturers of networking devices, offering a wide range of routers suited for different needs, from basic home setups to more advanced networking solutions for businesses. Below, I'll go into more detail about what a TP-Link router is, its features, and its types.

Key Components of a TP-Link Router

1. **WAN Port (Wide Area Network Port):** This port connects to the modem, which provides internet service. It allows the router to get access to the internet and share it across multiple devices.
2. **LAN Ports (Local Area Network Ports):** These are Ethernet ports where you can plug in devices directly using wired connections, such as computers, smart TVs, and gaming consoles. They provide stable and high-speed internet connections compared to wireless ones.
3. **Wireless Access Point (Wi-Fi):** TP-Link routers come with a built-in wireless access point that transmits and receives wireless signals (Wi-Fi). This feature allows devices like smartphones, laptops, and tablets to connect wirelessly to the network.
4. **Power Input:** This is where you plug in the power adapter to supply the router with power.

3.5 AC drive (LSIS IC5 inverter) :

The LSIS IC5 Inverter is a high-performance variable frequency drive (VFD) designed for controlling the speed and torque of AC induction motors in industrial applications. Below is a more detailed breakdown of the key features, specifications, and applications of the IC5 series inverter. The IC5 inverter supports both V/F (Voltage/Frequency) control and sensorless vector control (SLVC), offering high-performance motor control with precise speed regulation and better torque control. It ensures smooth acceleration and deceleration of the motor, providing better performance in applications where rapid speed changes are required. The IC5 inverter supports the Modbus RTU protocol for integration with a SCADA or PLC system, allowing for remote monitoring and control. Some models support additional communication protocols like CANopen and Profibus for industrial automation systems.

Chapter 4

Practical

Chapter 4

4.1 Hardware Parts :

1. **Relay:** A relay is an electrical switch that opens and closes under the control of an external circuit. It's often used to control a high-power device (like a motor or light) with a low-power signal. When an electric current flows through the relay's coil, it activates a switch mechanism that controls the connected load.



Figure 4.1 Relay

2. **Arduino:** Arduino is an open-source electronics platform based on simple software and hardware. It consists of a microcontroller that can be programmed to interact with sensors, actuators, and other devices. It's commonly used for DIY electronics projects and prototyping.



Figure 4.2 Arduino

3. **RS-485:** RS-485 is a standard for serial data transmission, which is widely used in industrial communication systems. It supports

long-distance communication and multiple devices connected in a network. It's particularly useful for environments with electrical noise or where a large number of devices need to communicate over long distances.

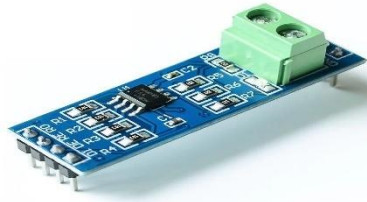


Figure 4.3 Rs-485

4. **Router:** A router is a networking device that forwards data packets between computer networks, typically between a local area network (LAN) and the internet. It directs traffic based on IP addresses, ensuring that data reaches its intended destination.



Figure 4.4 Router

5. **Water Pump:** A water pump is a mechanical device used to move water from one place to another. It can be used for a variety of purposes, such as supplying water to a building or draining water from a flooded area. There are different types of water pumps, including submersible, centrifugal, and diaphragm pumps.



Figure 4.5 WaterPump

6. **Pump:** A pump is any mechanical device used to move fluids (liquids or gases). Pumps work by either displacing the fluid, creating suction, or using pressure to move the fluid. The term "pump" could refer to water pumps, fuel pumps, air pumps, and many more.



Figure 4.6 Pump

7. **Water Filter:** A water filter is a device or system designed to remove contaminants and impurities from water. Water filters typically use physical barriers, chemical processes, or biological processes to improve the water quality for consumption or use.



Figure 4.7 Filter

8. **3-Phase Induction Motor:** A 3-phase induction motor is a type of AC motor that runs on a 3-phase electrical supply. It is commonly used in industrial and commercial applications because of its efficiency and reliability. The motor operates based on electromagnetic induction, with no need for external brushes or commutators.



Figure 4.8 3ph Induction motor

9. **VFD LSIS IC5:** A Variable Frequency Drive (VFD) is an electronic device that controls the speed and torque of an electric motor by varying the frequency and voltage supplied to the motor. The LSIS IC5 is a specific brand and model of VFD, often used to control 3-phase motors for applications like pumps, fans, and conveyor systems.



Figure 4.9 LSIS IC5 VFD

10.**Breadboard:** A breadboard is a tool used for prototyping electronic circuits without soldering. It allows components such as resistors, capacitors, and integrated circuits (ICs) to be connected and tested in a temporary setup before moving to a permanent version.

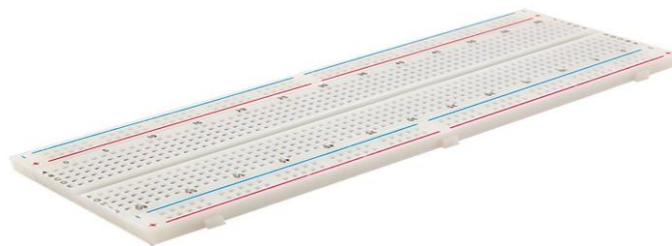


Figure 4.10 BreadBoard

11.**PLC XGB:** A PLC (Programmable Logic Controller) is a ruggedized computer used for automation and control in industrial settings. The XGB is a specific model or series of PLCs, commonly used in various industries to control machinery, assembly lines, and processes.



Figure 4.11 PLC XGB

12. Emergency Buttons: Emergency buttons are safety devices that can be pressed to shut down equipment or activate safety systems in case of an emergency. These are often used in factories, machinery, and electrical systems to quickly halt operations if a dangerous situation arises.



Figure 4.12 Emergency Button

13. Push Buttons: Push buttons are simple mechanical or electronic switches that are pressed to activate or deactivate a function. These are commonly found on control panels or machinery to start or stop operations or to signal an action.



Figure 4.13 Push Buttons

14. Siemens VDC 24: VDC 24 refers to a 24-volt DC (Direct Current) power supply. It's commonly used in control systems, sensors, actuators, and communication devices where a stable low voltage is required for operation. Many industrial devices operate on 24VDC for safety and efficiency reasons.



Figure 4.14 Siemens VDC24

15. ultrasonic sensor : is a device that uses ultrasonic waves (sound waves at frequencies higher than the human hearing range, typically above 20 kHz) to detect objects and measure distances. The sensor emits ultrasonic sound waves, which travel through the air and bounce back when they hit an object. The sensor then measures the time it takes for the waves to return, calculating the distance based on the speed of sound.

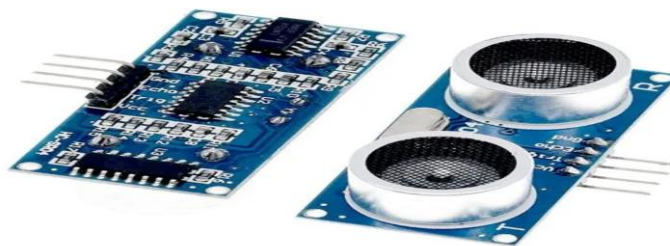


Figure 4.15 Ultrasonic Sensor

4.2 Software Parts :

4.2.1 Arduino Programming:

In our project, an Arduino module was used to read data from analog and digital sensors. It acted as a data acquisition device, converting sensor signals into numerical data that could be transmitted to the PLC. Communication between the Arduino and PLC was achieved using the Modbus RTU protocol over the RS-485 interface.

The Arduino was programmed using **Arduino IDE**, and custom code was developed to:

- Read sensor values (e.g., distance from ultrasonic sensor).
- Format the data into Modbus registers.
- Continuously respond to Modbus requests from the PLC.



Figure 4.16 Arduino IDE

4.2.2 Arduino code:

```
#include <ModbusRTU.h>

ModbusRTU mb;

#define DE_RE 2

#define TRIG_PIN1 5

#define ECHO_PIN1 4

#define TRIG_PIN2 7

#define ECHO_PIN2 6

const long TANK_HEIGHT = 24;
```

```

long duration1, distance1, level1;
long duration2, distance2, level2;
unsigned long lastUpdate = 0;
void setup() {
    Serial.begin(9600);
    pinMode(DE_RE, OUTPUT);
    pinMode(TRIG_PIN1, OUTPUT);
    pinMode(ECHO_PIN1, INPUT);
    pinMode(TRIG_PIN2, OUTPUT);
    pinMode(ECHO_PIN2, INPUT);
    digitalWrite(DE_RE, LOW);
    mb.begin(&Serial, DE_RE);
    mb.slave(1);
    mb.addHreg(0, 0);
    mb.addHreg(1, 0);
}
void loop() {
    if (millis() - lastUpdate > 1000) {
        digitalWrite(TRIG_PIN1, LOW);
        delayMicroseconds(2);
        digitalWrite(TRIG_PIN1, HIGH);
        delayMicroseconds(10);
        digitalWrite(TRIG_PIN1, LOW);
        duration1 = pulseIn(ECHO_PIN1, HIGH, 30000);
        if (duration1 == 0) duration1 = TANK_HEIGHT * 58;
        distance1 = duration1 / 58.0;
        level1 = TANK_HEIGHT - distance1;
        if (level1 < 0) level1 = 0;
    }
}

```

```

    if (level1 > TANK_HEIGHT) level1 = TANK_HEIGHT;
    digitalWrite(TRIG_PIN2, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN2, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN2, LOW);
    duration2 = pulseIn(ECHO_PIN2, HIGH, 30000);
    if (duration2 == 0) duration2 = TANK_HEIGHT * 58;
    distance2 = duration2 / 58.0;
    level2 = TANK_HEIGHT - distance2;
    if (level2 < 0) level2 = 0;
    if (level2 > TANK_HEIGHT) level2 = TANK_HEIGHT;
    mb.Hreg(0, (uint16_t)level1);
    mb.Hreg(1, (uint16_t)level2);
    Serial.print("Water Level 1: ");
    Serial.print(level1);
    Serial.print(" cm | Water Level 2: ");
    Serial.print(level2);
    Serial.println(" cm");

    lastUpdate = millis();
}

mb.task();
delay(10);}

```

4.2.2 PLC Programming:

The PLC used in our project was the **XGB series**, programmed via the **XG5000 software**. The PLC was the core of the control system, handling input signals, executing logic, and sending control commands to devices.

Key functions implemented in the PLC:

- Reading data from Arduino and VFD via Modbus RTU.
- Processing logic based on sensor input.
- Sending control signals to the VFD and relays.
- Communicating with KepServer through TCP/IP.

All logic was implemented using ladder diagrams inside the XG5000 environment.



Figure 4.17 XG5000

4.2.3 PLC program Logic:

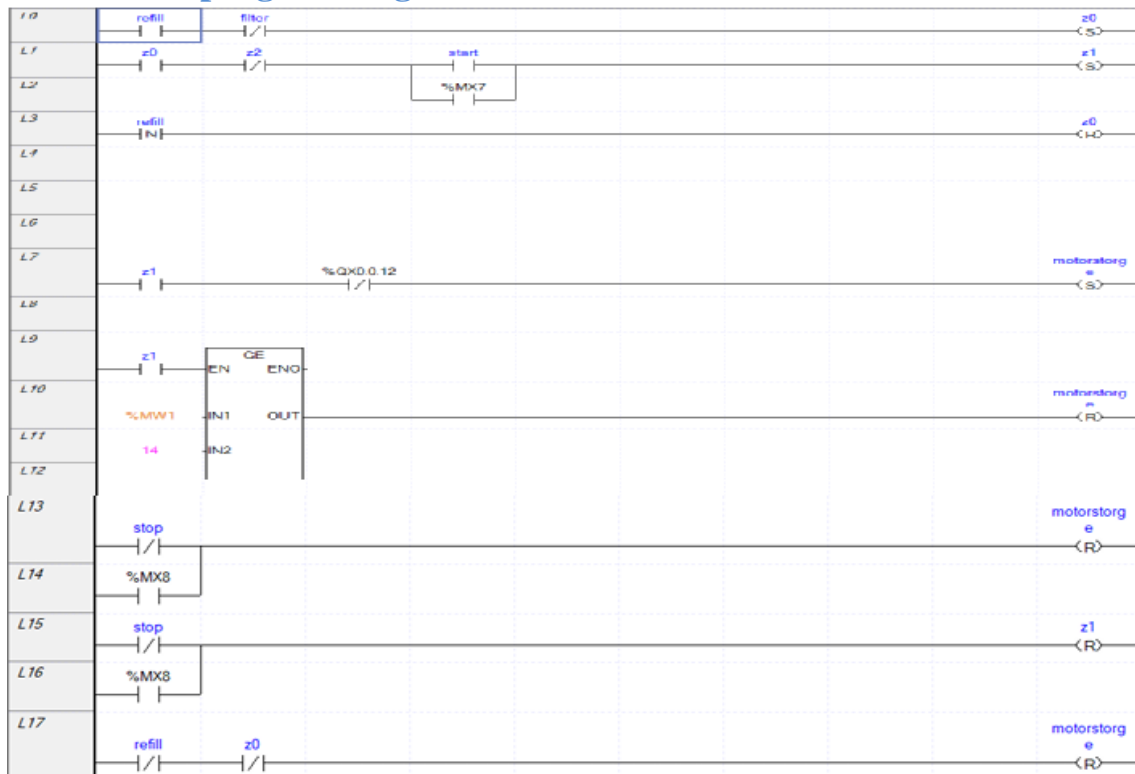


Figure 4.18 Refill operation

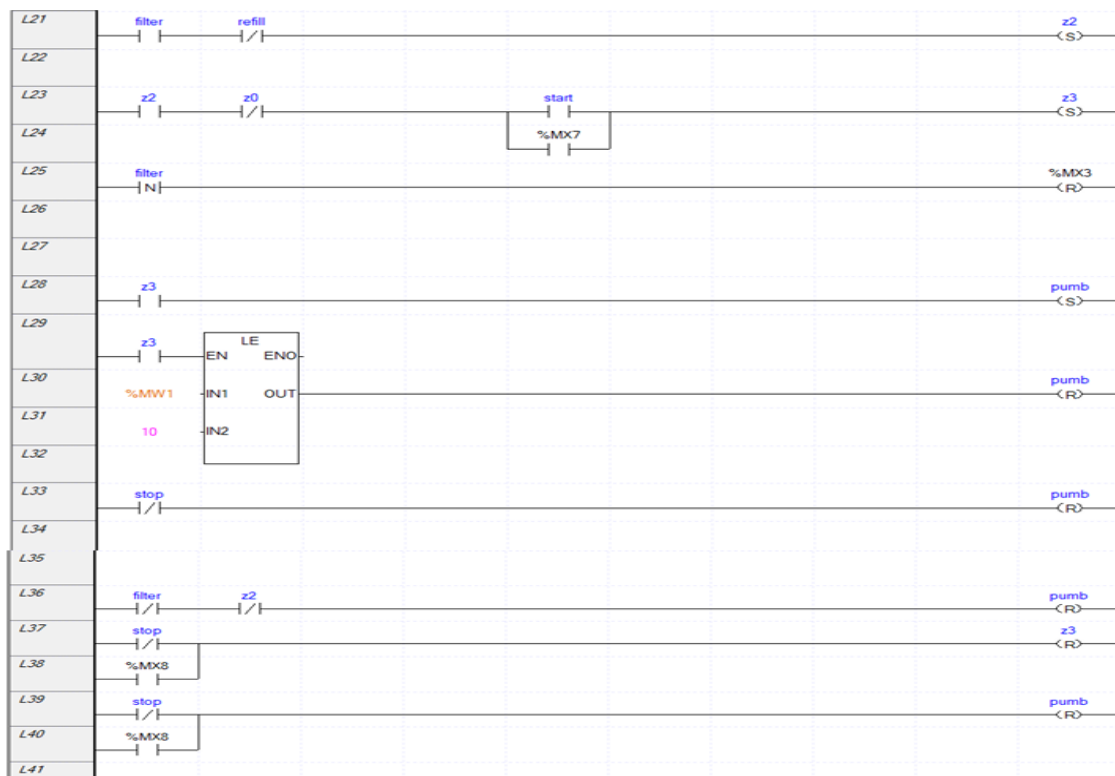


Figure 4.19 Filter operation

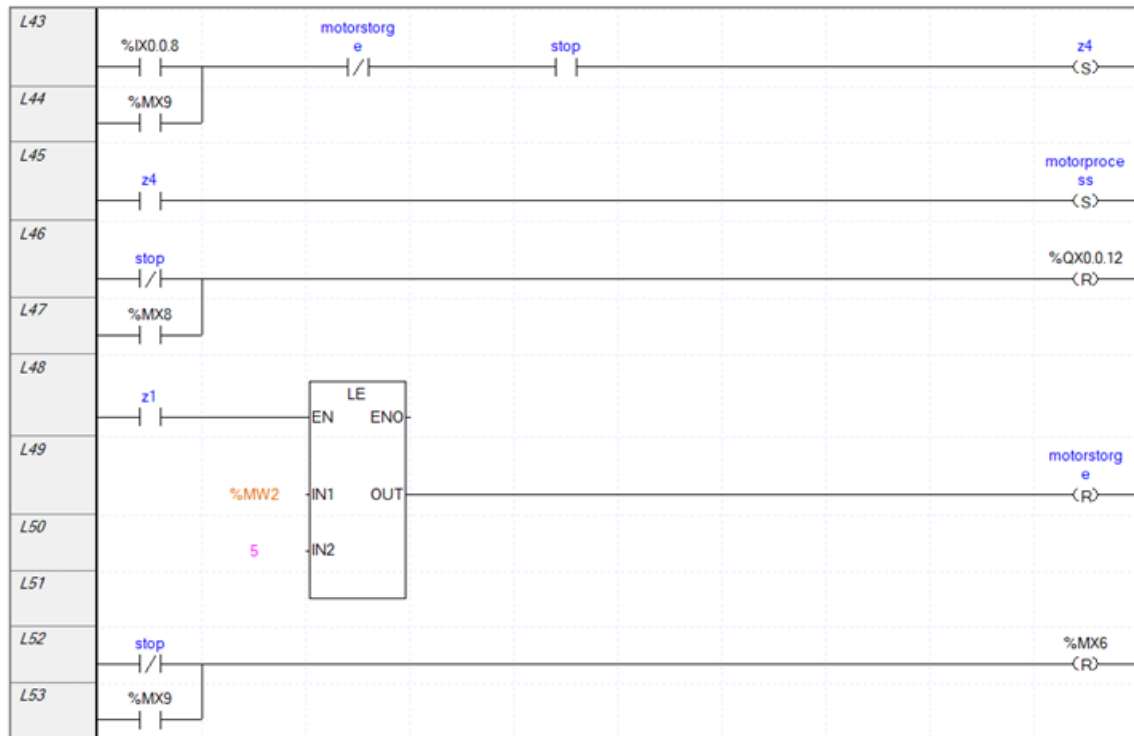


Figure 4.20 Emptying operation

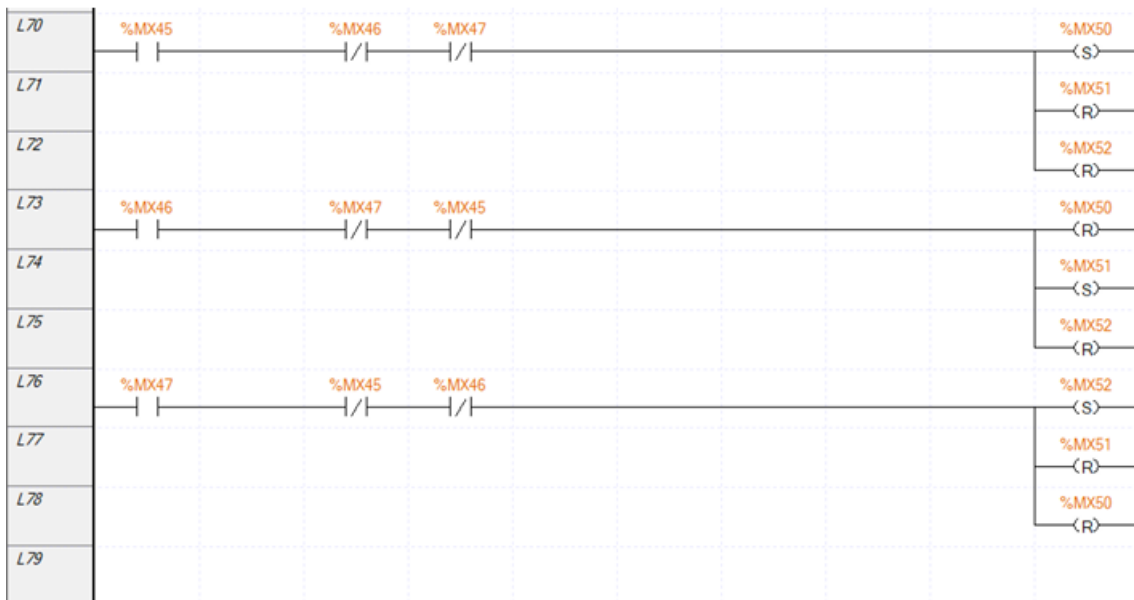




Figure 4.21 VFD operation

4.2.4 Protocols and Resister:

In this part of the project, communication protocols and register mapping played a critical role in ensuring data exchange between devices.

1- Modbus RTU (RS-485):

Used to connect the Arduino and VFD with the PLC. It allowed the reading of sensor data from Arduino and status/control registers from the VFD. Modbus provides reliable communication in industrial environments. When transferring sensor data from the Arduino to the PLC, we used holding registers because the sensor

data type was word (not bit). Specifically, we used address **40000** for the first sensor and **40001** for the second sensor.

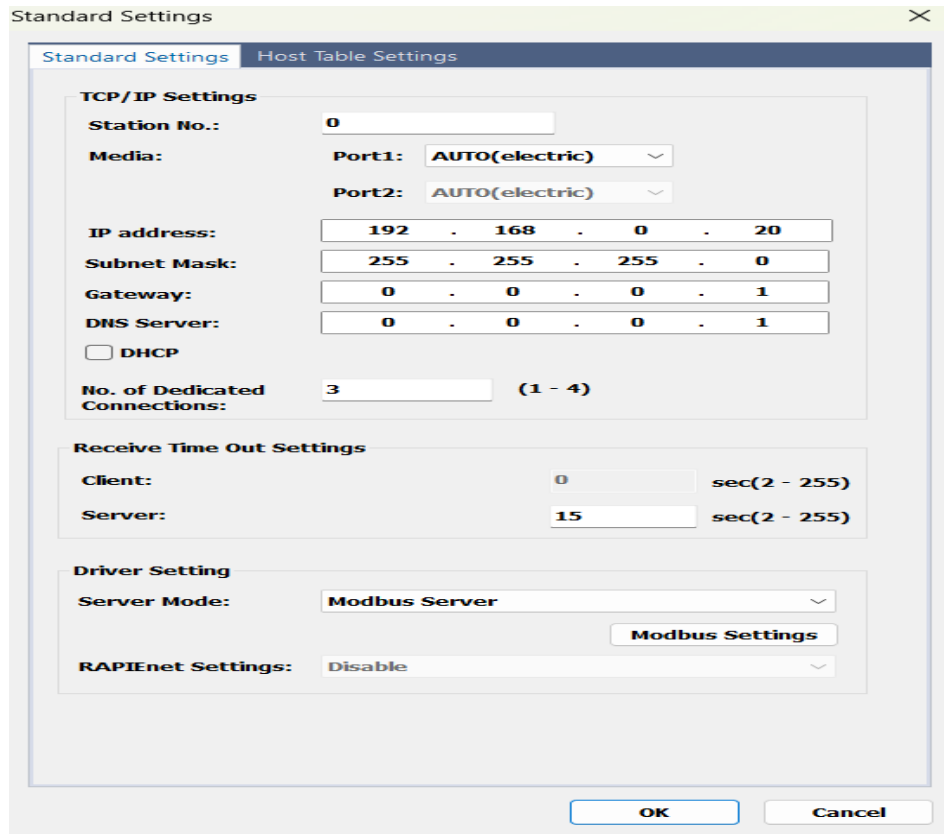
Address	Parameter	Scale	Unit	R/W	Description
0x0004	Parameter Read/Write enable			R/W	0: Parameter Lock 1: Parameter Read/Write Enable
0x0005	Frequency Reference	0.01	Hz	R/W	Starting freq – Max freq
0x0006	Operatin command (Option)			R/W	BIT 0 : Stop (S) BIT 1 : Forward Run (F) BIT 2 : Reverse Run (R) BIT 3 : Fault reset (0->1) BIT 4 : Emergency stop BIT 5 : Not used
0x0007	Accel time	0.1	sec	R/W	See function table
0x0008	Decel time	0.1	sec	R/W	See function table
0x0009	Output current	0.1	A	R	See function table
0x000A	Output frequency	0.01	Hz	R	See function table
0x000B	Output voltage	0.1	V	R	See function table
0x000C	DC Link Voltage	0.1	V	R	See function table
0x000D	Output power	0.1	kW	R	See function table
0x000E	Status of Inverter			R	BIT 0 : Stop BIT 1 : Forward running BIT 2 : Reverse running BIT 3 : Fault (Trip) BIT 4 : Accelerating BIT 5 : Decelerating BIT 6 : Speed arrival BIT 7 : DC Braking BIT 8 : Stopping Bit 9 : Not Used BIT 10 : Brake Open (I55: 3 or 4) BIT13: REM. R/S BIT14: REM. Freq.
0x000F	Trip information			R	BIT 0 : OCT BIT 1 : OV BIT 2 : EXT-A BIT 3 : EST BIT 4 : Option BIT 5 : GF(Ground Fault) BIT 6 : OH(Inverter overheat) BIT 7 : ETH(Motor overheat) BIT 8 : OLT(Overload trip) BIT 9 : HW-Diag BIT10: EXT-B BIT11: EEP

Figure 4.22 Registers of VFD

"In Figure 4.21, the registers related to the VFD type we used are shown. Each manufacturer has its own manual, which contains different registers specific to their device."

2- TCP/IP:

Used to connect the PLC to the network router, enabling communication with the KepServerEX software over Ethernet.



The screenshot shows the 'Standard Settings' dialog box with the 'Host Table Settings' tab selected. The 'TCP/IP Settings' section includes fields for Station No. (0), Media (Port1 and Port2 both set to AUTO(electric)), IP address (192.168.0.20), Subnet Mask (255.255.255.0), Gateway (0.0.0.1), and DNS Server (0.0.0.1). There is an unchecked checkbox for DHCP and a field for No. of Dedicated Connections (3) with a range of (1 - 4). The 'Receive Time Out Settings' section has Client (0) and Server (15) fields, both with a range of sec(2 - 255). The 'Driver Setting' section shows Server Mode set to Modbus Server, with a Modbus Settings button, and RAPIEnet Settings set to Disable. OK and Cancel buttons are at the bottom right.

Figure 4.23 PLC to Router Connection Configuration

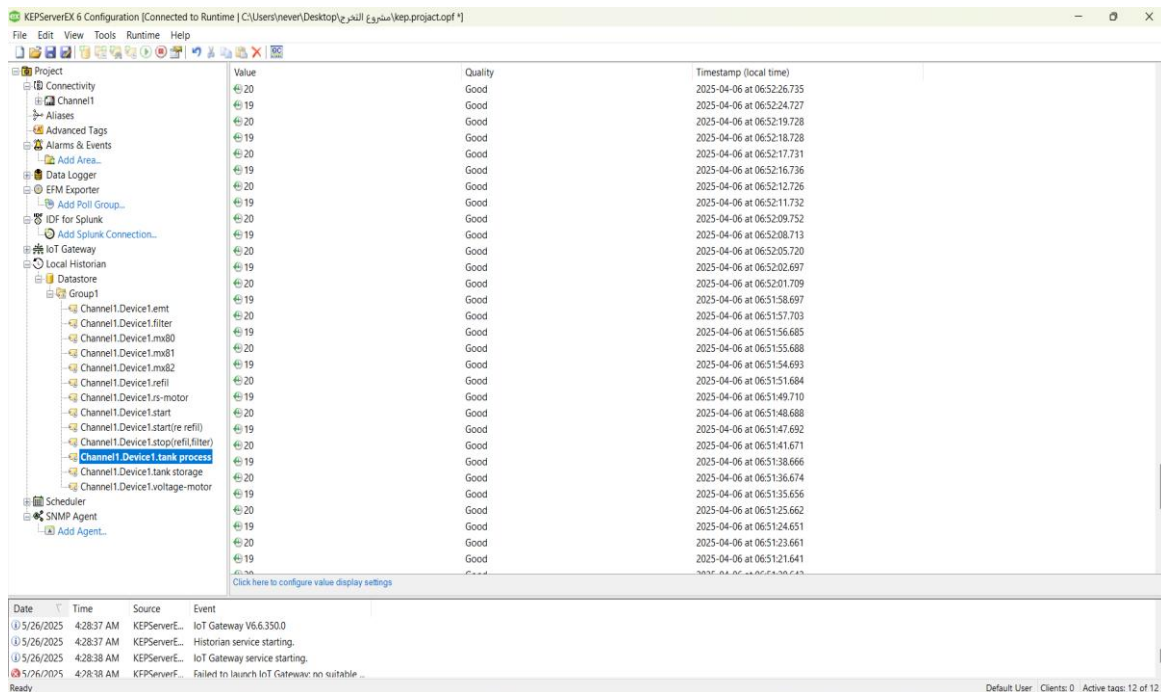


Figure 4.24 KEPServer

The KepServer software was connected to the router via the TCP/IP protocol in order to access the data stored in the PLC registers. Then,

through KepServer, we established a connection with the HMI using the OPC UA protocol.

We used KepServer because it provides an internal server that stores data permanently. This allows us to retrieve and read data accurately at any time and on any specific date.



The screenshot shows the KEPServerEX 6 Configuration window. The left pane displays a project tree with various components like Connectivity, Alarms & Events, Data Logger, and IoT Gateway. The right pane shows a table of data points with columns for Value, Quality, and Timestamp (local time). The 'Channel1.Device1.tank process' tag is highlighted in blue.

Value	Quality	Timestamp (local time)
20	Good	2025-04-06 at 06:52:26.735
19	Good	2025-04-06 at 06:52:24.727
20	Good	2025-04-06 at 06:52:19.728
19	Good	2025-04-06 at 06:52:18.728
20	Good	2025-04-06 at 06:52:17.731
19	Good	2025-04-06 at 06:52:16.736
20	Good	2025-04-06 at 06:52:12.726
19	Good	2025-04-06 at 06:52:11.732
20	Good	2025-04-06 at 06:52:09.752
19	Good	2025-04-06 at 06:52:08.713
20	Good	2025-04-06 at 06:52:05.720
19	Good	2025-04-06 at 06:52:02.697
20	Good	2025-04-06 at 06:52:01.709
19	Good	2025-04-06 at 06:51:58.697
20	Good	2025-04-06 at 06:51:57.703
19	Good	2025-04-06 at 06:51:56.685
20	Good	2025-04-06 at 06:51:55.688
19	Good	2025-04-06 at 06:51:54.693
20	Good	2025-04-06 at 06:51:51.684
19	Good	2025-04-06 at 06:51:49.710
20	Good	2025-04-06 at 06:51:48.688
19	Good	2025-04-06 at 06:51:47.692
20	Good	2025-04-06 at 06:51:41.671
19	Good	2025-04-06 at 06:51:38.666
20	Good	2025-04-06 at 06:51:36.674
19	Good	2025-04-06 at 06:51:35.656
20	Good	2025-04-06 at 06:51:25.662
19	Good	2025-04-06 at 06:51:24.651
20	Good	2025-04-06 at 06:51:23.661
19	Good	2025-04-06 at 06:51:21.641

Below the table, there is a log window showing events:

Date	Time	Source	Event
5/26/2025	4:28:37 AM	KEPServerE...	IoT Gateway V6.6.350.0
5/26/2025	4:28:37 AM	KEPServerE...	Historian service starting.
5/26/2025	4:28:38 AM	KEPServerE...	IoT Gateway service starting.
5/26/2025	4:28:38 AM	KEPServerE...	Failed to launch IoT Gateway: no suitable ...

Figure 4.25 KEPServer data store

This is a sample of the sensor data readings we used. It clearly shows the sensor values along with the corresponding date, including the day, month, year, and precise time

4.2.5 HMI Tags and Configurations:

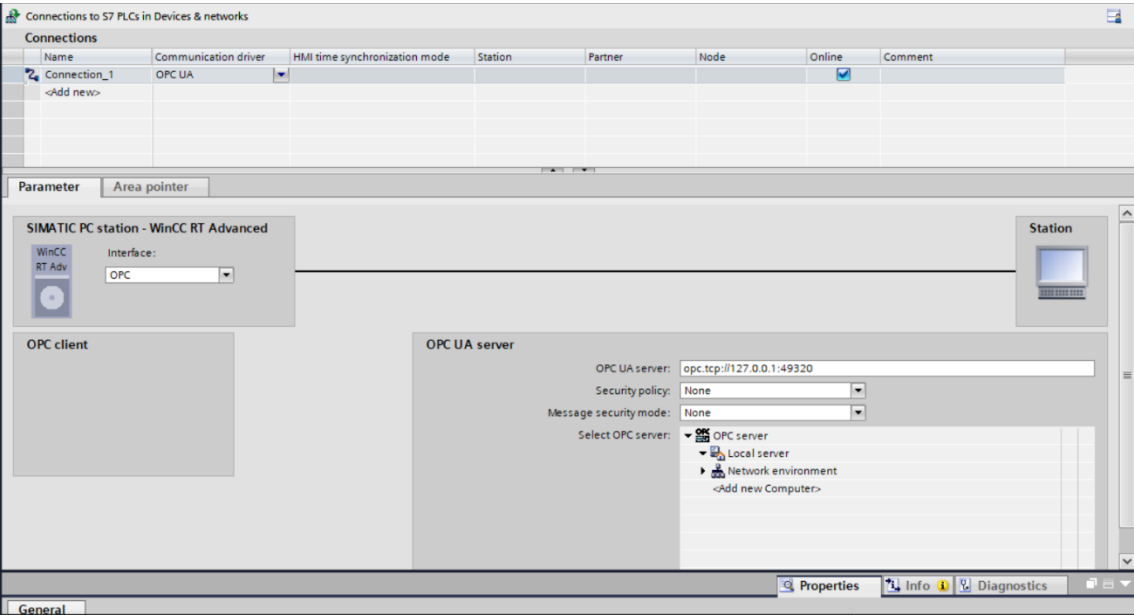


Figure 4.26 SCADA Configurations

HMI tags						
Name	Tag table	Data type	Connection	PLC name	PLC tag	Address
emt	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
filter	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
fw-motor	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
mx80	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
mx81	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
mx82	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
refil	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
rpm-motor	Default tag table	UInt16	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
rs-motor	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
sensor2	Default tag table	UInt16	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
sensor1	Default tag table	UInt16	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
star	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
stop	Default tag table	Boolean	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
voltage-motor	Default tag table	UInt16	Connection_1		<Undefined>	ns=KEPServerEX3=Channel1.Devic...
<Add new>						

Figure 4.27 SCADA Tags

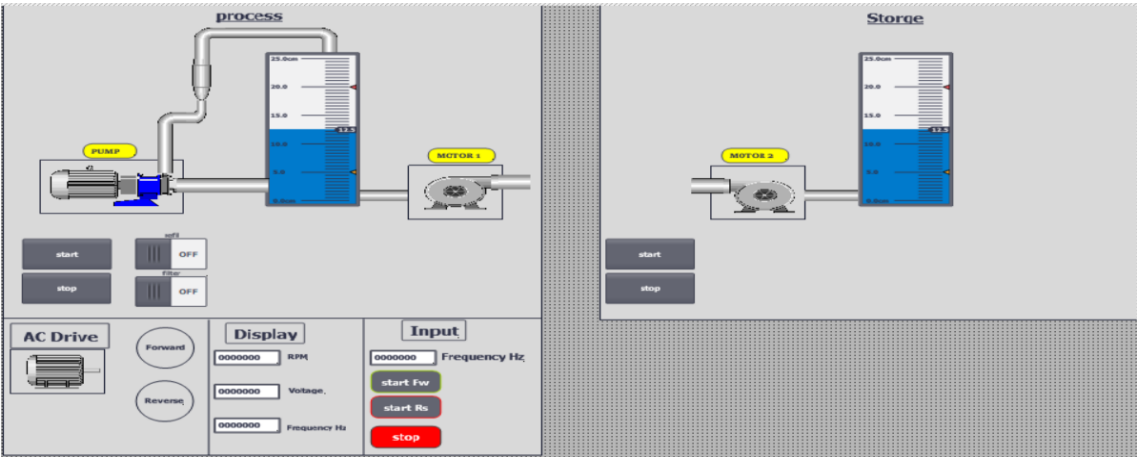


Figure 4.28 SCADA Screen

4.2.6 Final Overview of the Project:

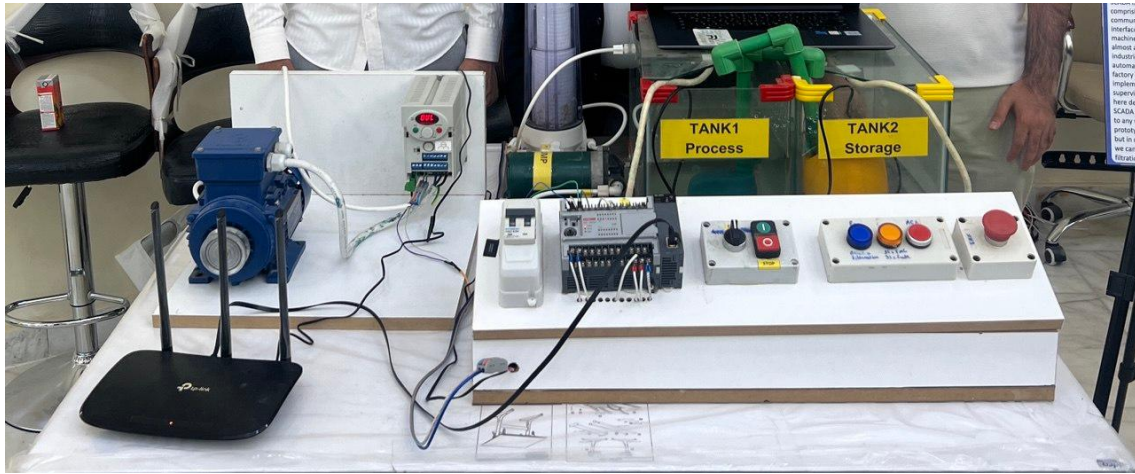


Figure 4.29 actual project pictures showing all components

This project successfully demonstrated the design, development, and implementation of a fully functional SCADA system using both hardware and software components commonly found in industrial environments. By integrating PLCs, VFDs, sensors, Arduino modules, routers, communication protocols, and SCADA software, we created a compact and practical automation system capable of real-time monitoring, control, and data exchange. The implementation involved several engineering disciplines including control systems, communication networks, software programming, and electrical design. We were able to monitor sensor readings, control motor speeds through the VFD, manage alarms, and visualize the entire process through a user-friendly HMI. The use of protocols like Modbus RTU and OPC UA ensured smooth and reliable communication across all system components.

Chapter 5

Conclusion

Chapter 5

5.1 Conclusion:

The successful completion of this project marked a significant achievement in the field of industrial automation. Our main objective was to design, program, and implement a complete SCADA system capable of real-time monitoring, data acquisition, control, and user interaction — and we accomplished this by integrating multiple components and technologies in a structured and systematic way. We used an XGB PLC as the central controller, which communicated with an Arduino to read sensor data and with a VFD to control a three-phase induction motor. The communication was established using Modbus RTU over RS-485, while KepServerEX served as a bridge to connect the PLC with the HMI via OPC UA protocol. The HMI, developed using TIA Portal, offered a real-time, user-friendly interface for operators to monitor system status and interact with the process. One of the key accomplishments of this project was the integration of heterogeneous devices using standard industrial communication protocols. This enabled smooth data flow and reliable control without requiring expensive or proprietary platforms. The system proved to be accurate, stable, and responsive during testing, with the ability to log data, generate alarms, and adapt to different scenarios. From an educational perspective, the project allowed us to gain hands-on experience in PLC programming, embedded system integration, serial and Ethernet communication, SCADA software configuration, and industrial system design. It also improved our ability to work as a team, manage a multidisciplinary project, and troubleshoot complex problems all of which are critical

skills in professional engineering practice In summary, the project not only met its technical and functional goals but also laid a strong foundation for our future in the field of control systems, automation, and industrial communication.

5.2 Future Work:

While the current SCADA system functions effectively, there are several ways it can be enhanced and extended to serve more advanced industrial applications:

- **Wireless Communication:** Currently, all data exchange relies on wired communication. Future versions could implement wireless technologies such as Wi-Fi, Zigbee, or LoRaWAN to reduce cabling, improve flexibility, and enable remote installations.
- **Cloud Integration and IoT:** The system could be extended to send data to cloud platforms like AWS IoT, ThingsBoard, or Blynk, enabling real-time monitoring and control from anywhere in the world. This would make the system part of a modern Industrial Internet of Things (IIoT) solution.
- **Mobile Application Interface:** A mobile-friendly dashboard or smartphone app could be developed to allow operators to receive alerts, view data trends, and issue control commands on the go.
- **Advanced Data Analytics:** Collected data could be used for predictive maintenance and process optimization by applying machine learning techniques or data visualization tools like Grafana or Power BI.

- **System Scalability:** The current project is a small-scale prototype. In future stages, more sensors, actuators, and PLCs could be added to manage larger industrial plants or distributed systems.

By addressing these future enhancements, the SCADA system can evolve from a student prototype into a reliable industrial-grade platform suitable for real-world deployment in smart factories and Industry 4.0 environments.

Reference:

- [1] J.H. Christensen, PLCopen Standard Presentation V1.0, 1998.
- [2] International Standard 61131: Programmable Logic Controllers. Part 3: Languages, 1993.
- [3] Archana B.Yadav & Pooja S.Shukla, Paper on "Augmentation to Water supply schme using PLC & SCADA", IEEE digital library.
- [4]
<http://literature.rockwellautomation.com/idc/groups/public/documents/webassets/browse-category.hcst>
- [5] S. M. U. Talha, S. S. Mohani, S. H. Ahmed and M. Ebrahim, "Design for an irrigation and monitoring system of an automated dam", Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 2, 2012.
- [6] K. A. Gupta, N. Armani, T. MANJUNATH and H. MANJUNATH, "De-sign and implementation of plc based industrial application prototypes", Indian Journal of Science and Technology, vol. 10, 2017.
- [7] Falliere. N, Murchu. L. O, Chien. E. 2010. W32.Stuxnet Dossier. Symantec Security Response, (Nov. 2010), 1—64
- [8] Dabidson. C. C, Andel. T. R, Yampolskiy. M, McDonald. J. T, and Gilsson. W. 2018. On SCADA PLC and Fieldbus Cybersecurity. Proceedings of the 13th International Conference on Cyber Warfare and Security, 140--148.

- [9] A. Vasic kaninova and M. Bakosova, "Cascade fuzzy logic control of a chemical reactor", Proc. 15. Int. Conference Process Control'05, 2005.
- [10] B. Reaves and T. Morris, "Discovery Infiltration and Denial of Service in a Process Control System Wireless Network" in IEEE eCrime Researchers Summit, Tacoma, WA, pp. 20-21, October 2009.
- [11] M. Banzi: Getting Started with Arduino (O'Reilly Media, America 2009).
- [12] C. Queiroz, A. Mahmood, and Z. Tari, "SCADASim - A Framework for Building SCADA Simulations, "IEEE Transactions on Smart Grid, vol. 2, no. 4 (2011): 589-597.
- [13] <https://www.arduino.cc/>
- [14] Mohd. Firdaus Bin Mahyidin. "Student Attendance Using RFID System". in University Malaysia, Pahang, May2008.