**CHAPTER ONE**

**INTRODUCTION**

**1.1 BACKGROUND HISTORY**

In today's industrial landscape, efficiency, reliability, and precision are paramount. The Supervisory Control and Data Acquisition (SCADA) process control system stands at the forefront of achieving these goals, serving as a critical component in the management and automation of complex industrial processes. By integrating Programmable Logic Controllers (PLCs), SCADA systems offer an unparalleled level of control and monitoring capabilities, ensuring seamless operation and enhanced productivity across various sectors, including manufacturing, energy, water treatment, and transportation.

SCADA systems function as the central nervous system of industrial operations, enabling real-time data collection, processing, and analysis. They facilitate the remote control of equipment and processes, thereby minimizing downtime, reducing operational costs, and improving overall system reliability. PLCs, which are robust, flexible, and scalable, play a vital role in this ecosystem. They are the backbone of local automation, interfacing with sensors and actuators to execute precise control logic and communicate crucial data back to the SCADA system.

This integration of SCADA systems with PLCs allows for the effective management of complex processes by providing a comprehensive view of operational data and control mechanisms. Through the use of advanced communication protocols and software tools, such as Visual Studio for SCADA application development, these systems can be tailored to meet specific industry requirements, offering custom solutions that enhance process efficiency and accuracy.

The following sections of this report delve deeper into the components and architecture of SCADA systems, the role and functionality of PLCs, and the development of SCADA applications using Visual Studio. By exploring these aspects, this report aims to provide a thorough understanding of the implementation and benefits of SCADA process control systems in modern industrial environments.

**1.2 AIM AND OBJECTIVE**

The aim of this project is to design, develop, and implement a robust SCADA process control system using Programmable Logic Controllers (PLCs) to enhance the monitoring, control, and automation of industrial processes. The system is intended to improve operational efficiency, ensure real-time data acquisition and analysis, and provide a scalable solution adaptable to various industrial applications.

**1.2.1 SPECIFIC OBJECTIVES**

i. System Design and Architecture

- Develop a comprehensive system architecture that integrates PLCs with a SCADA application to manage industrial processes effectively.

- Define the hardware and software requirements for the SCADA system and ensure compatibility with existing industrial infrastructure.

ii. PLC Programming and Configuration

- Program and configure PLCs to interface with sensors and actuators, enabling precise data acquisition and control of industrial processes.

- Implement control logic within the PLCs to handle automated responses and ensure seamless operation of connected equipment.

iii. SCADA Application Development

- Develop a user-friendly SCADA application using Visual Studio that provides real-time data visualization, control interfaces, and historical data logging.

- Ensure the application supports multiple communication protocols (e.g., Modbus, OPC, Ethernet/IP) for reliable data exchange between PLCs and the SCADA system.

iv. Integration and Communication

- Establish and test communication links between PLCs and the SCADA application to ensure reliable data transmission and control.

- Implement security measures to protect the communication network from cyber threats and ensure data integrity.

v. System Testing and Validation

- Conduct comprehensive testing of the SCADA system, including simulation and field tests, to validate its performance under various operational conditions.

- Identify and resolve any issues or bugs to ensure the system operates reliably and efficiently.

vi. Training and Documentation

- Develop detailed documentation covering the system architecture, PLC programming, SCADA application development, and integration procedures.

- Provide training for operators and maintenance personnel to ensure they are proficient in using and maintaining the SCADA system.

vii. Deployment and Maintenance

- Deploy the SCADA system in the target industrial environment and monitor its performance during the initial operation phase.

- Establish a maintenance schedule and provide ongoing support to address any operational issues and ensure long-term system reliability.

viii. Scalability and Future Enhancements

- Design the SCADA system with scalability in mind, allowing for easy expansion and integration of additional PLCs and sensors as needed.

- Explore and implement future enhancements, such as IoT integration, advanced data analytics, and cloud-based monitoring, to continuously improve the system’s capabilities and performance.

**1.3 SCOPE AND LIMITATION**

**Scope**

1. System Architecture and Design

- Develop a comprehensive SCADA process control system integrating PLCs to manage and automate industrial processes.

- Define the hardware and software components required, including sensors, actuators, PLCs, communication networks, and SCADA software.

2. PLC Programming and Configuration

- Program and configure PLCs to interface with various sensors and actuators, ensuring accurate data acquisition and control of processes.

- Implement control logic within PLCs to automate responses and ensure the seamless operation of industrial equipment.

3. SCADA Application Development

- Develop a SCADA application using Visual Studio, featuring real-time data visualization, control interfaces, alarm management, and historical data logging.

- Ensure the SCADA application supports multiple communication protocols (e.g., Modbus, OPC, Ethernet/IP) for reliable data exchange between PLCs and the central system.

4. Integration and Communication

- Establish robust communication links between PLCs and the SCADA system, ensuring reliable data transmission and control.

- Implement cybersecurity measures to protect the system from potential threats and ensure data integrity.

1. System Testing and Validatio - Conduct thorough testing, including simulation and field tests, to validate the performance and reliability of the SCADA system.

- Identify and resolve any issues or bugs to ensure optimal system functionality.

6. Deployment and Maintenance

- Deploy the SCADA system in the target industrial environment, providing real-time monitoring and control of processes.

- Develop a maintenance schedule and provide ongoing support to address operational issues and ensure long-term system reliability.

7. Training and Documentation

- Create detailed documentation covering system architecture, PLC programming, SCADA application development, and integration procedures.

- Provide training for operators and maintenance personnel to ensure proficient use and upkeep of the SCADA system.

8. Scalability and Future Enhancements

- Design the SCADA system to be scalable, allowing for the easy addition of PLCs and sensors to accommodate future growth.

- Explore and implement advanced features such as IoT integration, predictive maintenance, and cloud-based monitoring for continuous improvement.

**Limitations**

1. System Complexity

- The integration of multiple hardware and software components increases system complexity, requiring specialized knowledge for design, implementation, and maintenance.

2. Initial Cost

- The initial setup cost of a SCADA system, including hardware, software, and installation, can be high, which may be a barrier for smaller enterprises.

3. Cybersecurity Risks

- Despite implementing robust security measures, SCADA systems are still vulnerable to cyber attacks, which can compromise system integrity and data security.

4. Maintenance Requirements

- Regular maintenance is required to ensure the reliability and performance of the SCADA system, including updates to software and replacement of hardware components.

5.Integration Challenges

- Integrating the SCADA system with existing infrastructure and different types of PLCs and sensors can present compatibility and communication challenges.

6. Data Accuracy and Reliability

- The accuracy and reliability of data collected by the SCADA system depend on the quality and calibration of sensors and PLCs. Malfunctioning components can lead to incorrect data and control actions.

7. Training and Skill Requirements

- Effective operation and maintenance of the SCADA system require skilled personnel with specialized training, which can be a limitation for organizations with limited technical expertise.

8. Scalability Constraints

- While the system is designed to be scalable, adding new components may require significant reconfiguration and testing to ensure seamless integration and performance.

**1.4 CONTRIBUTION TO KNOWLEDGE**

Research into SCADA (Supervisory Control and Data Acquisition) process control systems significantly contributes to the body of knowledge in multiple ways, encompassing technological, operational, and strategic dimensions. Below are several key contributions:

1. Technological Advancements

A. Development of Advanced Control Algorithms

Research in SCADA systems often leads to the development of more efficient and reliable control algorithms. These algorithms optimize the operation of industrial processes, improving response times and reducing errors.

B. Enhanced Communication Protocols

Investigation into SCADA communication protocols enhances data transmission efficiency and security. Improved protocols like OPC UA, Modbus TCP/IP, and DNP3 contribute to more reliable and secure data exchange between field devices and control centers.

C. Integration with Emerging Technologies

SCADA research facilitates the integration of emerging technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and Machine Learning (ML). This integration allows for advanced predictive maintenance, anomaly detection, and data analytics, pushing the boundaries of what SCADA systems can achieve.

2. Operational Improvements

A. Increased System Reliability and Efficiency

Research efforts contribute to the development of more robust SCADA systems that enhance the reliability and efficiency of industrial processes. By identifying and mitigating potential failure points, these systems ensure continuous and efficient operations.

B. Real-Time Data Processing and Decision Making

Improving real-time data acquisition and processing capabilities allows for better decision-making and quicker response to operational issues. This is crucial in environments where timely interventions can prevent significant disruptions.

C. Enhanced User Interfaces

Research into human-machine interfaces (HMI) within SCADA systems leads to more intuitive and user-friendly control panels. Better interfaces improve operator efficiency and reduce the likelihood of human errors.

3. Strategic and Management Insights

A. Cost Reduction

Research in SCADA systems often identifies ways to reduce operational and maintenance costs. By optimizing process control and improving resource management, companies can achieve significant cost savings.

B. Cybersecurity Improvements

With the growing threat of cyber attacks, SCADA research focuses on developing advanced security measures to protect critical infrastructure. Contributions in this area help secure industrial processes against unauthorized access and cyber threats.

C. Regulatory Compliance

Research helps organizations understand and comply with regulatory requirements for safety, environmental protection, and operational standards. This ensures that SCADA systems not only improve efficiency but also adhere to legal and safety standards.

4. Academic and Educational Contributions

A. Knowledge Dissemination

Academic research in SCADA systems contributes to the dissemination of knowledge through publications, conferences, and seminars. This helps in spreading advanced concepts and innovations to a broader audience.

B. Curriculum Development

Research findings are often integrated into academic curricula, providing students with up-to-date knowledge and practical skills relevant to modern industrial automation and control systems.

C. Interdisciplinary Collaboration

SCADA research fosters collaboration across various fields such as electrical engineering, computer science, cybersecurity, and industrial engineering. This interdisciplinary approach broadens the scope and impact of research findings.

5. Industry Standards and Best Practices

A. Setting Industry Benchmarks

Research in SCADA systems helps establish industry standards and best practices. These benchmarks guide companies in implementing effective and reliable SCADA solutions.

B. Case Studies and Lessons Learned

Documenting and analyzing case studies from SCADA implementations provides valuable lessons and best practices. This information helps other organizations in planning and executing their SCADA projects more effectively.

**1.5 AREAS OF APPLICATION**

SCADA (Supervisory Control and Data Acquisition) process control systems are essential for monitoring and controlling industrial processes across a wide range of sectors. Their ability to collect real-time data, analyze it, and execute control actions makes them indispensable in many applications. Below are key areas where SCADA systems are applied:

i. Manufacturing and Production

A. Assembly Lines

- Monitoring and controlling the operations of assembly lines.

- Ensuring consistent quality and detecting faults in real-time.

B. Process Industries

- Managing chemical, pharmaceutical, food and beverage production processes.

- Controlling temperature, pressure, and flow rates to maintain optimal conditions.

ii. Energy and Utilities

A. Power Generation

- Monitoring and controlling operations in power plants (thermal, hydro, nuclear).

- Managing electrical grids to ensure stability and efficiency.

B. Oil and Gas

- Overseeing exploration, drilling, and production operations.

- Monitoring pipeline pressures, flow rates, and detecting leaks.

C. Renewable Energy

- Managing wind farms, solar power plants, and hydroelectric facilities.

- Optimizing energy production and integration into the grid.

iii. Water and Wastewater Management

A. Water Treatment Plants

- Monitoring and controlling water purification processes.

- Ensuring safe water quality by managing chemical dosing and filtration.

B. Wastewater Treatment

- Overseeing wastewater treatment processes.

- Controlling aeration, sedimentation, and disinfection stages.

C. Distribution Networks

- Managing water supply networks to ensure consistent delivery.

- Detecting leaks and managing pressure in pipelines.

iv. Transportation

A. Railways

- Monitoring and controlling railway signaling systems.

- Managing train schedules and ensuring passenger safety.

B. Highways and Traffic Management

- Overseeing traffic light systems and road condition monitoring.

- Managing toll collection and traffic flow.

C. Airports

- Managing runway lighting, baggage handling systems, and environmental controls.

- Monitoring fuel supply and aircraft movements.

v. Building and Facilities Management

A. Smart Buildings

- Controlling HVAC (Heating, Ventilation, and Air Conditioning) systems.

- Managing lighting, security, and energy consumption.

B. Industrial Facilities

- Overseeing operations of large industrial complexes.

- Monitoring environmental conditions and optimizing resource usage.

vi. Telecommunications

A. Network Management

- Monitoring and controlling telecom networks.

- Ensuring uptime and performance of communication systems.

B. Data Centers

- Managing power, cooling, and security systems in data centers.

- Monitoring server performance and environmental conditions.

vii. Agriculture

A. Precision Farming

- Monitoring soil moisture, weather conditions, and crop health.

- Controlling irrigation systems to optimize water usage.

B. Livestock Management

- Monitoring the health and productivity of livestock.

- Controlling feeding systems and environmental conditions in barns.

viii. Environmental Monitoring

A. Air Quality Control

- Monitoring and controlling air pollution levels in urban and industrial areas.

- Ensuring compliance with environmental regulations.

B. Weather Stations

- Collecting and analyzing weather data.

- Providing real-time data for weather forecasting and disaster management.

ix. Mining and Metals

A. Mineral Extraction

- Monitoring and controlling extraction processes.

- Managing equipment and ensuring safety in mines.

B. Smelting and Refining

- Overseeing the smelting and refining processes.

- Controlling temperature and chemical composition to ensure product quality.

x. Healthcare

A. Hospital Facilities Management

- Controlling HVAC systems, lighting, and security in healthcare facilities.

- Monitoring critical systems like power supply and medical gas distribution.

B. Pharmaceutical Manufacturing

- Ensuring the consistency and quality of pharmaceutical products.

- Controlling environmental conditions in cleanrooms and production areas.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Review of past related work**

**Design of Low Cost Multi Channel Data Acquisition System for Meteorological Application BY: Nisha Kashyap, Department of Electronics and Communication Engineering National Institute of Technology, Rourkela, Odisha May 2015.**

The primary objective of the present work is to design a low cost multi-channel data acquisition system which can be used for meteorological application. The proposed multi-channel data acquisition system acquires ambient temperature, barometric pressure, altitude, humidity and light intensity data from environment and stores the data for future use. The sensors are connected with a low-cost microcontroller (ATmega328) unit which performs the data acquisition and data logging operation. A real time clock is used to keep current time with the measurement. Different communication interfaces such as serial communication, wireless communication and Ethernet communication are used in the data acquisition system. In serial communication, the sensor data are logged in COM port of PC. LabVIEW based application is developed which provides graphical user interface for the user. VISA protocol is used to communicate the COM port data with LabVIEW. Other communication protocol such as wireless communication and Ethernet communication protocol is used to transmit the sensor data over a communication channel. The data is uploaded in Ethernet which can be viewed using a web browser. In wireless communication Xbee transmitter and Xbee receiver modules are used to transmit data over a longer distance. The salient features of this developed system is that the system is low cost, uses open source software’s like Arduino and python and the system is economical (that is - INR 6500). This developed system performs satisfactorily under different condition. The system is tested for 24 hour in April 2015 and provides satisfactory results.

**Bluetooth Based Data Acquisition System BY: Suresh Gurjar, Department of Electrical Engineering National Institute of Technology Rourkela Rourkela 769008 INDIA. May 2015**

Data acquisition systems are devices used to collect information to document or analyse some physical phenomenon such as voltage, force or temperature. Data acquisition systems available in the market are very expensive, bulky and power hungry. However, PC based data acquisition system offers a lot of benefit in terms of processing speed, display resolution and connectivity capabilities. The Project aims at designing and implementing a portable, economical and power efficient real-time data acquisition system. The proposed system comprises of a hardware circuitry and a Graphical User Interface (GUI) based on MATLAB environment. The hardware device consists of an 8-bit microcontroller interfaced with a serial ADC chip and a Bluetooth serial module. The Bluetooth HC-05 module is used to provide a wireless connectivity between the hardware and the PC. For testing purpose, the sampling rate of ADC is set to 833 Hz, capturing 50 values per 0.06 second. Whereas on the PC side, the GUI receives the sampled values transmitted by the hardware device and plots the real-time signal waveform. It has been found that the GUI plots the signal waveform with good quality and efficiency. The proposed system can be deployed in number of industrial applications such as remote device controlling and ECG data acquisition with some adjustments in the hardware. The wireless connectivity reduces the complexity of cables and probability of occurrences of the accidents in industrial areas.

**Real-Time Measurement And Monitoring System Using Zigbee And Labview By: Muhammad Hairul Bin Yahaya, Faculty of Electronic and Computer Engineering Universiti Teknikal Malaysia Melaka JUNE 2015.**

Wireless communication technologies become popular in real-time industrial environments. This project describes the fundamental work of real time measurement and monitoring system with Arduino, ZigBee, and Graphical Programming language (LabVIEW). Thickness of the item is taken as a parameter that will convert into the number of items. By acquiring the analogue data from the ultrasonic sensor, the suitable signal conditional is fed to Arduino. It is connected to the ZigBee transmitter module, which transmits the data to the ZigBee receiver. As a result, the data received will be viewed on a PC via LabVIEW. The methodology describe in the proposed work is useful for designing Wireless Sensor Network (WSN) system in the industrial application such as real levelling storage. The experimental of transmission range and observation of data required in LabVIEW was reported. The combination of measurement, wireless transmission and monitoring system can be improved of accuracy of measurement, especially for sustainability of better working place.

**Principles and Applications of Computerized Process Control in the Department Of Chemical Engineering, Rivers State University Of Science And Technology,**

**Port Harcourt, Nigeria. by DONGESIT EFFIONG SAMPSON (2016).**

Automatic process control, supervised control and data acquisition, integration of information systems or enterprise resource integration (ERI) activities are widely applied by process industries to remain competitive

and improve efficiency and profit margins. Computerized process control is more efficient than manual control as it

safes time and effort and with the digital inclusion new technologies that enhance the Nigerian vision for digital

human capital in the 21stCentury are introduced. In Nigeria, most industries use control systems which lack the features necessary for effective communication and efficient operation, thus imposing a limitation on the use of the system to achieve high operational efficiency, high quality product, yields and profit. An efficient use of a good process control computer system enhances efficiency, increased production rate and reduced operating cost

**A Simplified Approach for Using PLC and SCADA System in 330 kV distribution Substations in Nigeria by Ngang Bassey Ngang Anthony Lordson Amana UkoOfe**

**(2018).**

Introduction of SCADA system in Generation, Transmission and Distribution sectors of our electricity industry has changed the traditional way of doing our routine work in the industries.SCADA the substations and sends control signals to the remote control equipment on a continuous basis. It gets the historical data of the substation and generates the alarms in the event of electrical accidents or faults.Most substations now utilize the SCAD system to monitor measurements, of relevant parameters for correction and regulation of end devices.Certain measurements, supervision,control, operation and protection functions are performed using the technology The problem of not controlling power substation properly by the conventional approach has made someareas not to have power supply as at the time they are meant to have it. That is the reason SCADA control of power substation applying programmable Logic Controllers is now adopted by power systems Engineers. It has improved the overall efficiency of operating power system components. The design is done by taking cognizance of the fact that a membership function for SCADA control is required. You can now proceed by designing a SCADA rule for controlling of power substation, training these rules in ANN, designing a visual basic model for SCADA control of power substation using PLC and designing a Simulink model for SCADA control of power substation using PLC. The result obtained when compared to the conventional one was 10% better.

**2.2 Block diagram of the SCADA Process control system**

**PSU**

**PLC**

**FCE**

**COM Driver**

**HMI**

**FLOWRATE**

**SENSOR**

**INDICATORS AND ALARM SYSTEM**

**LEVEL**

**SENSOR**

1. Power Supply Unit (PSU): The power supply unit converts the incoming electrical power supply (AC or DC) to the appropriate voltage and current required by the PLC and other components in the SCADA system.
2. PLC Unit: The PLC unit is the central processing unit of the SCADA system. It contains the CPU, memory, and interfaces for communication with other devices. It receives input signals from the I/O modules, processes them using the program stored in its memory, and generates output signals to control the connected devices.
3. COM Driver: These modules interface with the external world and connect to the PLC unit. They receive input signals from sensors and switches, and convert them into a format that the PLC can understand. Similarly, they take output signals from the PLC and convert them into signals suitable for controlling the actuators and other devices.
4. Human Machine Interface (HMI) Device: The HMI device provides a graphical user interface for monitoring and interacting with the PLC. It typically consists of a display screen, buttons, and/or a touchscreen. The HMI allows users to view and modify system parameters, monitor the status of inputs and outputs, and interact with the PLC program.
5. FEC(Final Control Element): These are the devices that interact with the physical world. Sensors detect physical variables such as temperature, pressure, or proximity and convert them into electrical signals. Actuators, on the other hand, receive electrical signals from the PLC and perform physical actions, such as moving a motor or turning on/off a valve.

The components in a PLC SCADA control system are interconnected using appropriate cables and connectors to establish the necessary communication and power connections.

## 2.3 Component Review of PLC based SCADA system

**2.3.1 Power Supply Unit**

This unit converts the 220V AC to 12V AC , 30V DC required by the circuit. It was implemented with the following components:

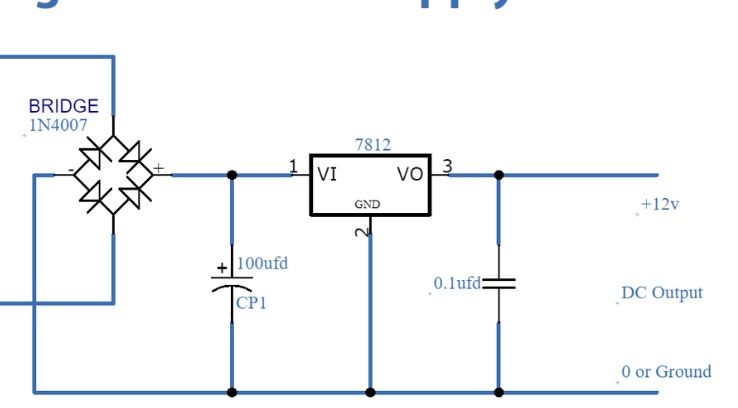
• 220V/12-0-12V step down transformer

• Bridge Diode

• Capacitor

• Voltage regulator

Below is the circuit diagram of the power supply unit:

****

# **Fig. 2.2: Power Supply Unit**

**Voltage Regulator:**

As we require a 12V we need LM7812 Voltage Regulator IC.

7812 IC Rating:

• Input voltage range 15V- 45V

• Current rating IC = 1.5A

• Output voltage range Vmax=12.2v, Vmin=11.8v

**Transformer:**

Selecting a suitable transformer is of great importance. The current rating and the secondary voltage of the transformer is a crucial factor.

• The current rating of the transformer depends upon the current required for the load to be driven.

• The input voltage to the 7812 should be at least 2V greater than the required 15V output, therefore it requires an input voltage at least close to 17V.

• So, we chose a 15-0-15V transformer with current rating 1500mA (Since 15\*√2 = 21.21V).

**Rectifying circuit:**

The bridge rectifier converts the ac voltage input to dc voltage at its output. The diodes IN4007 was used. The best is using a full wave rectifier due to the following good qualities:

• Its DC saturation is less, as in, both cycle diodes conduct.

• Higher Transformer Utilization Factor (TUF).

• 1N4007 diodes are used as it is capable of withstanding a higher reverse voltage of 1000v whereas 1N4001 is 50V.

• Center Tap Full Wave Rectifier.

The choice of the bridge rectifier depends on:

i. Peak inverse voltage.

ii. The forward current rating

The diode forward current rating is the maximum that the diode can conduct before failing. The diode should be selected in such a way that the current passing through it should be less than the forward current rating. The peak inverse output is the reverse voltage that the diode has to block when not conducting.

Peak Inverse Voltage (PIV) = ­

where Vrms  = transformer output = 15Vac

* Peak inverse voltage = x 15 = 16.97v
* The diodes used has forward current ≥ 1500mA and PIV ≥ 21.21V

**Selection of the Filter Capacitor (C1):**

The filter capacitor smoothens the dc voltage from the bridge rectifier.

The choice depends on:

i. The capacitor breakdown voltage

ii. The ripple percentage required

Capacitor breakdown voltage (Vc) is gotten by taking KVL from the bridge rectifier output to capacitor terminal. Using Fig. 3.1

Vpeak – Vd – Vc = O

Where Vpeak = bridge rectifier output

Vd = drop across bridge rectifier diodes

Vc = capacitor terminal voltage

21.21v – 1.4v – Vc = 0 ------------- (3.1)

For full wave rectification, on each half section, 0.7 is drop across each of the two conducting diodes. Which gives 1.4v (2 x 0.7v)

equation (3.1) becomes

19.81 – Vc = 0

where Vc = 19.81V

In practices, the rule is to use a capacitor with breakdown voltage double of the terminal voltage:

V1c = 2 x Vc

V1c = 2 x 19.81

= 39.62V

A standard value of 50V was used.

**Capacitor Capacitance:**

for full wave

Where VΔ is the difference between the maximum peak voltage and the minimum peak voltage.

Maximum peak = 21.21v

Minimum peak = 21.21 - % ripple

% ripple = 21.21 – minimum peak

taking minimum peak = 19.21v

% ripple = (21.21 – 19.21) v = 2v



A standard value of 2500uf capacitor was used.

**2.3.2 Arduino Based PLC**

An Arduino-based PLC leverages the flexibility of the Arduino microcontroller platform to emulate the functionality of a traditional Programmable Logic Controller (PLC). This approach provides an affordable and customizable option for automation tasks that are typically handled by dedicated PLCs.

* **Understanding PLCs and Their Functions**

A traditional PLC is a digital computer designed for controlling machinery and automation processes in industrial environments. Key functions include:

- Input Monitoring: Reading data from sensors, switches, and other input devices.

- Control Logic Execution: Performing programmed logic, typically using boolean logic, timers, counters, and other control functions.

- Output Control: Activating devices like motors, relays, lights, or alarms based on input conditions and programmed logic.

The PLC operates in a loop, continuously reading inputs, executing logic, and updating outputs based on the results.

* **Components of an Arduino-based PLC**

An Arduino-based PLC incorporates the core functions of a PLC but uses the Arduino microcontroller as the processing unit. The design typically includes the following components:

- Arduino Board (e.g., Arduino Nano, Uno, or Mega): Serves as the central processing unit (CPU). Arduino can be programmed to read inputs, execute logic, and control outputs.

-Input Signal Conditioning: Circuits or components (like operational amplifiers or resistors) that adapt input signals to levels compatible with the Arduino, ensuring signals from sensors or switches are accurately read.

- Output Drivers (e.g., ULN2803 or MOSFETs): Components to drive high-power loads, as Arduino's output pins can only supply limited current.

-Power Supply: A stable power source to supply both the Arduino and peripherals (like sensors and relays).

-Relay Modules or Solid-State Relays (SSRs): Interfaces between the Arduino and high-power AC or DC loads, allowing safe control of these loads.

Diagram of Arduino-Based PLC Components:

* Inputs: Push buttons, sensors, or switches connected to input conditioning circuits.
* Arduino Board: Running control logic based on inputs and managing outputs.
* Outputs: Relays, motors, or actuators connected to output drivers.
* **Input Signal Conditioning**

Input conditioning is necessary because the Arduino’s input pins require certain voltage levels (typically 0–5V) and current limits to interpret signals properly. Input signal conditioning includes:

- Level Shifting: Ensuring signals match the 5V or 3.3V levels expected by Arduino.

- Filtering: Removing noise or unwanted frequencies from sensor signals to improve stability.

- Protection Circuits: Safeguarding the Arduino from over-voltage and transient spikes, commonly using diodes or resistors.

For analog signals, operational amplifiers like the LM324 are used for amplifying, offsetting, and filtering to adjust signal levels.

* **Control Logic and Arduino Programming**

Control logic is the core function of a PLC. In an Arduino-based PLC:

- Sketch Programming: The control logic is written in the Arduino language (based on C/C++), uploaded via the Arduino IDE.

- Logic Functions: Include basic operations (AND, OR, NOT), timers, counters, and more advanced decision-making.

- Loop Operation: Arduino's `loop()` function continuously executes the logic, mimicking the cyclic scan nature of traditional PLCs.

- Decision-Making: The Arduino reads inputs, applies logic, and updates outputs based on programmed conditions.

A typical PLC operation can be programmed as follows:

```cpp

void loop() {

int inputStatus = digitalRead(inputPin);

if (inputStatus == HIGH) {

digitalWrite(outputPin, HIGH); // Activate output

} else {

digitalWrite(outputPin, LOW); // Deactivate output

}

}

More complex control operations involve timers and counters:

```cpp

if (millis() - lastActivationTime >= delayInterval) {

// Perform delayed action or time-based logic

}

* **Output Driving and Load Control**

Arduino’s output pins can source only limited current (typically 20-40 mA), which is insufficient for most industrial loads. Therefore, an output driver is necessary to interface with higher-current loads. Common output drivers include:

- ULN2803 Darlington Array: An 8-channel transistor array that can handle up to 500mA per channel, often used to drive relays or small motors.

- MOSFETs: For driving loads requiring even higher currents, MOSFETs are efficient, fast-switching transistors.

- Relays: To control high-power devices, relay modules are used, which allow the Arduino to control AC or DC circuits indirectly.

The output driver converts the low-current Arduino output signal into a higher-current output that can power larger devices.

* **Power Supply and Protection Circuits**

A reliable power supply and protection circuits are crucial for stable operation:

- Power Supply: The Arduino and peripherals (like relays) often require separate power supplies to avoid interference. Typically, a 5V DC regulator supplies the Arduino, while a 12V or 24V power source supplies the relays or motors.

- Isolation and Protection: Flyback diodes, optocouplers, or isolation relays prevent feedback from inductive loads (e.g., motors) that could damage the Arduino or create noise in the circuit.

* **Communication and User Interface (Optional)**

To monitor and control the Arduino-based PLC, communication interfaces can be added:

-Serial Communication: Provides simple monitoring via a computer.

-LCD/OLED Displays: Shows real-time status or error messages on a small screen.

-Ethernet or Wi-Fi Modules: Connects the PLC to networks, allowing remote monitoring or control.

* **Advantages and Limitations of Arduino-based PLCs**

**Advantages:**

- Cost-Effective: Low-cost components make this solution affordable.

- Customizability: Fully programmable, ideal for custom control applications.

- Scalability: Can be expanded by adding more input/output modules or shields.

**Limitations:**

- Processing Speed: Lower than traditional PLCs, limiting suitability for high-speed applications.

- Environmental Robustness: Arduino-based PLCs may lack the durability and environmental protections of industrial-grade PLCs.

- Limited I/O and Memory: Arduino boards have limited I/O pins and memory compared to dedicated PLCs, which may restrict complex applications.

**Applications**

Arduino-based PLCs are ideal for small-scale or low-cost automation projects, including:

-Home Automation: Controlling lighting, security, and appliances.

- Educational Training: Teaching basic automation and control principles.

- Small Industrial Tasks(SCADA): Simple control tasks like conveyor operation, pump control, and status indication in small industrial setups.

### **2.3.3 **Ultrasonic Level Sensor****

An ultrasonic sensor measures the distance to the surface of the liquid or solid material using ultrasonic waves. By placing the sensor at the top of a container, it can calculate the fill level based on the time it takes for the ultrasonic waves to reflect back.

#### Specifications

* **Range**: Typically 2 cm to 4 meters, depending on model
* **Accuracy**: ±1-3 mm
* **Output**: Analog voltage or PWM, which can be processed by the Arduino
* **Examples**: HC-SR04, JSN-SR04T

#### Pros and Cons

* **Pros**: Non-contact, works with liquids and solids, easy to set up
* **Cons**: Affected by dust, temperature, and foam on liquid surfaces; limited by range and container shape

#### Integration with Arduino

Ultrasonic sensors like the **HC-SR04** connect directly to Arduino pins, and the Arduino can calculate distance based on the time-of-flight of the sound waves. The calculated distance is then used to determine the fill level.

**2.3.4 HMI**

HMI (Human-Machine Interface) in a PLC (Programmable Logic Controller) system refers to the interface that allows humans to interact with the machine or industrial process controlled by the PLC. The HMI enables operators or users to monitor, control, and manage the system through graphical displays, input devices, and other interaction methods.

Key Components of HMI in PLC:

Display Screen: Shows real-time data from the PLC, such as system status, alarms, and process parameters (e.g., temperature, pressure, speed).

Control Buttons/Touchscreen: Allows users to input commands, adjust settings, or respond to system alerts.

Communication Protocol: Links the HMI and PLC, enabling them to exchange data. Common protocols include Modbus, Ethernet/IP, and Profibus.

Software: HMI software (e.g., Wonderware, Siemens WinCC, Allen-Bradley FactoryTalk) runs on the interface, helping to visualize and manipulate data.

Common Functions:

- Monitoring: Provides a real-time view of machine or process status.

- Control: Enables operators to adjust variables (e.g., start/stop machines, change setpoints).

- Alarms and Notifications: Alerts users to abnormal conditions or malfunctions in the system.

**2.3.4 INDICATORS AND ALARMS SYSTEM**

In industrial automation, **Indicators** and **Alarms** play crucial roles in monitoring and responding to various system conditions. This report outlines the design of an indicators and alarms system integrated into an **Arduino-based PLC with SCADA (Supervisory Control and Data Acquisition)**. The purpose of this system is to provide real-time monitoring, notification, and corrective actions for abnormal conditions, such as high temperatures, low fluid levels, or device malfunctions.

The **Arduino** serves as the PLC to gather data from sensors, process logic, and control indicators and alarms, while the **SCADA** interface provides a user-friendly dashboard for real-time monitoring and control.

##### Alarms Setup

Alarms notify operators of urgent issues that may require immediate intervention:

* **Alarm LED (Flashing Red)**: Indicates a severe or emergency condition.
* **Audible Alarm (Buzzer)**: Provides an intermittent sound during critical conditions.
* **SCADA Notifications**: Generates on-screen alerts and logs within the SCADA dashboard.

The alarms are designed to activate under conditions that require immediate operator attention, such as exceeding flowrate thresholds or critical fluid levels.

**CHAPTER THREE**

**MATERIALS AND METHODS**

**3.1 Design and Analysis**

In this section, the various block/units and their circuit diagrams, design analysis, criteria and assumptions made for component selection are presented. Figure 3.1 shows the functional blocks of the designed circuit.

Power Supply

FLOW SENSOR

Switch Unit

PLC

LEVEL SENSOR

T

Pump

HMI

Human Machine Interface

Legend

Power

Data

**Figure 3.1:** Bock diagram of the SCADA Process control system

**3.1.1 Power Supply Unit**

This unit converts the 220V AC to 24V DC required by the circuit. It was implemented with the following components:

• 220V/15-0-15V step down transformer

• Bridge Diode

• Capacitor

• Voltage regulator

**Transformer:**

Selecting a suitable transformer is of great importance. The current rating and the secondary voltage of the transformer is a crucial factor.

• The current rating of the transformer depends upon the current required for the load to be driven.

• The input voltage to the 7815 and 7805 IC should be at least 2V greater than the required 15V output, therefore it requires an input voltage at least close to 17V.

• So, we chose a 15-0-15V transformer with current rating 1500mA (Since 15\*√2 = 21.21V).

**Rectifying circuit:**

The bridge rectifier converts the ac voltage input to dc voltage at its output. The diodes IN4007 was used. The best is using a full wave rectifier due to the following good qualities:

• Its DC saturation is less, as in, both cycle diodes conduct.

• Higher Transformer Utilization Factor (TUF).

• 1N4007 diodes are used as it is capable of withstanding a higher reverse voltage of 1000v whereas 1N4001 is 50V.

• Center Tap Full Wave Rectifier.

The choice of the bridge rectifier depends on:

i. Peak inverse voltage.

ii. The forward current rating

The diode forward current rating is the maximum that the diode can conduct before failing. The diode should be selected in such a way that the current passing through it should be less than the forward current rating. The peak inverse output is the reverse voltage that the diode has to block when not conducting.

Peak Inverse Voltage (PIV) = ­

where Vrms  = transformer output = 15Vac

* Peak inverse voltage = x 15 = 16.97v
* The diodes used has forward current ≥ 1500mA and PIV ≥ 21.21V

**Selection of the Filter Capacitor (C1):**

The filter capacitor smoothens the dc voltage from the bridge rectifier.

The choice depends on:

i. The capacitor breakdown voltage

ii. The ripple percentage required

Capacitor breakdown voltage (Vc) is gotten by taking KVL from the bridge rectifier output to capacitor terminal. Using Fig. 3.1

Vpeak – Vd – Vc = O

Where Vpeak = bridge rectifier output

Vd = drop across bridge rectifier diodes

Vc = capacitor terminal voltage

21.21v – 1.4v – Vc = 0 ------------- (3.1)

For full wave rectification, on each half section, 0.7 is drop across each of the two conducting diodes. Which gives 1.4v (2 x 0.7v)

equation (3.1) becomes

19.81 – Vc = 0

where Vc = 19.81V

In practices, the rule is to use a capacitor with breakdown voltage double of the terminal voltage:

V1c = 2 x Vc

V1c = 2 x 19.81

= 39.62V

A standard value of 50V was used.

**Capacitor Capacitance:**

for full wave

Where VΔ is the difference between the maximum peak voltage and the minimum peak voltage.

Maximum peak = 21.21v

Minimum peak = 21.21 - % ripple

% ripple = 21.21 – minimum peak

taking minimum peak = 19.21v

% ripple = (21.21 – 19.21) v = 2v



A standard value of 2500uf capacitor was used.

**Voltage Regulator:**

As we require a 5V,15V we need LM7805 and LM7815Voltage Regulator IC.

7805 IC Rating:

• Input voltage range 7V- 35V

• Current rating IC = 1.5A

• Output voltage range Vmax=5.2v, Vmin=4.8v

7815 IC Rating:

• Input voltage range 15V- 45V

• Current rating IC = 1.5A

• Output voltage range Vmax=15.2v, Vmin=14.8v

## 3.1.2 **THE CONTROLLER UNIT**

This Unit is the Heart of the Entire System. It performs the entire logic of the system. Below is the requirements of the controller unit.

**Requirements of the Controller**

* Should be able to interface analog signal(ADC).
* Should have USB Module required to interface the HMI system
* Should be stable and efficient.
* Should have enough Input/output Pins to accommodate the entire system.
* Should be easily programmed.
* It should available and cost effective.

**Selection of the Controller**

The Atmega328p Microcontroller From Microchip Corporation was selected. Below are the reasons of the selection.

* It is a 28pin Microcontroller, it has enough input/output pins.
* A pin in the microcontroller can supply 5V 20mA.
* It has internal ADC Module Required to interface the LM324 IC
* It Has UART Serial communication Module
* It has an easy programming interface.
* It is stable, cost effective and available.

The Atmega328p Microcontroller was biased with a 16Mhz Crystal and two (2) 15pf Capacitors.

**To determine the time the microcontroller executes one instruction**

One machine cycle is the time taken to execute an instruction

Machine cycle = 4 pulses of crystal oscillator.

Hence, time taken = Oscillator Frequency

4

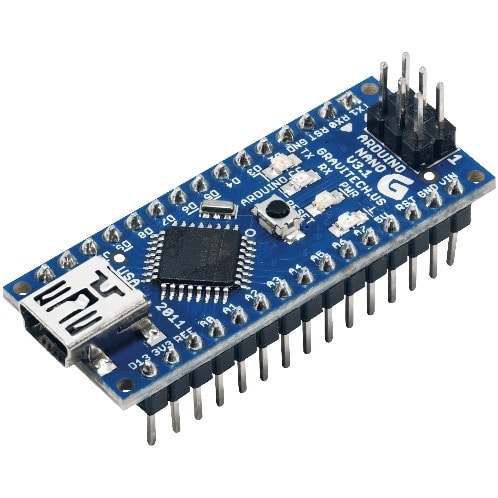
For one machine cycle

= 16MHZ = 4MHZ

4

1/F= 1/4MHZ = 1μs

The controller executes one instruction in 1 micro seconds.



**Figure 3.2:** The circuit diagram of micro controller unit

**3.1.3 SELECTION OF FLOW METER**

Selecting a flow meter for an Arduino-based SCADA system requires consideration of several factors, including the type of fluid (liquid, gas, or steam), flow rate range, accuracy requirements, and the communication interface with Arduino.

**Type of Flow Meter**

- Turbine Flow Meter: Good for low-viscosity liquids (e.g., water). They offer high accuracy but may require frequent calibration and maintenance.

- Electromagnetic Flow Meter: Best for conductive liquids like water and slurries. They’re highly accurate but usually more expensive and require conductive fluids to operate.

- Ultrasonic Flow Meter: Can measure liquid and gas. Non-invasive (clamp-on models) and suitable for measuring corrosive fluids. These are pricier but reduce maintenance.

- Coriolis Flow Meter: High accuracy for both liquids and gases, even at varying temperatures and pressures. However, these are usually costly.

- Differential Pressure Flow Meter: Suitable for gas and steam. Common for industrial applications but needs additional transmitters to calculate flow rate.

**Flow Rate and Accuracy**

- Determine the expected flow rate and accuracy requirements. Lower-cost sensors may have larger tolerances, while higher precision may require a premium meter.

- Check the operating range of the flow meter and ensure it aligns with your application.

**Communication Interface**

- Look for a flow meter with an output that is compatible with Arduino:

-Analog Output (0-5V or 4-20mA): Arduino can read these values with an analog input or an analog-to-digital converter (ADC) for 4-20mA.

- Pulse Output (frequency-based): Common in turbine flow meters. Arduino can interpret pulses and calculate flow rates.

- Digital Communication (I2C, UART, or SPI): Useful if you need to measure multiple parameters or want a more digital interface.

**Other Selection Criteria**

- Fluid Properties: Ensure the sensor material is compatible with the fluid to avoid corrosion.

- Installation and Maintenance: Consider non-invasive models if ease of maintenance is a priority.

- Cost: Depending on your budget, some flow meters like electromagnetic or ultrasonic are more expensive but may offer the reliability needed for long-term SCADA monitoring.

In this project, the Sea Water Flow sensor was used



**Figure 3.3:** The Sea Water Flow sensor

**3.1.4 SELECTION OF LEVEL SENSOR**

When choosing a level sensor for an Arduino-based SCADA system, you'll want to consider the type of liquid or material you’re monitoring, the required measurement range, accuracy, and compatibility with Arduino.

**Requirements of the Level Unit**

- Compatible with Arduino:

- Analog Output (0-5V or 4-20mA): Readable by Arduino’s analog input pins. For 4-20mA, use a resistor to convert the signal to a voltage that Arduino can read.

- I2C or Serial: For more advanced sensors (e.g., some radar or ultrasonic sensors), Arduino can read data over I2C or UART.

- Ensure the sensor’s measurement range matches the height of the project tank or container.

- Select a sensor that provides adequate accuracy for your SCADA application. Ultrasonic sensors, for instance, typically offer good accuracy for small to medium tanks.

## HC-SR04 Ultrasonic Sensor Pinout

The Ultra Sonic HC-SR04 emits ultrasound at 40,000Hz that travels in the air. If there is an object or obstacle in its path, then it collides and bounces back to the Ultra Sonic module. The formula **distance = speed \* time** is used to calculate the distance. Suppose, an object is placed at a distance of 10 cm away from the sensor, the speed of sound in air is 340 m/s or 0.034 cm/µs. It means the sound wave needs to travel in 294 µs. But the Echo pin double the distance (forward and bounce backward distance). So, to get the distance in cm multiply the received travel time value with echo pin by 0.034 and divide it by 2. The distance between Ultra Sonic HC-SR04 and an object is:

distance = (speed \* time)/ 2

Speed of Sound:

speed = 340m/s = 0.034cm/us

time = 10 /0.034us = 294us

distance = (speed \* time)/2

distance = (0.034 \* 294)/2

31

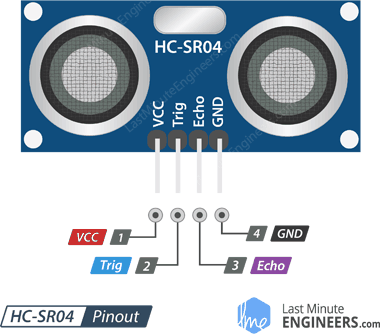


Fig 3.4: Pinout of Ultrasonic Sensor

## 

**3.2.7 THE ALARM UNIT**

This Unit Gives Audible Sound For when the process level reaches the SET Point. Below is the requirement of the ALARM unit.

**Requirements of the ALARM Unit**

* Should be able to give a good audible alert.
* Should have feasible connection interface.
* Reasonable power requirements i.e. input voltage and current demand.
* Should be cost effective and readily available.

**Selection of the ALARM**

The unit was implemented with a DC Buzzer Alarm. Below is the choice of the selection of the buzzer alarm:

* It has a wide operating voltage i.e. 5-24V
* It gives a clear and audible sound.
* It is easily biased with a resistor.
* It is cheap and very available.

The Unit is Directly Controlled by the Controller Unit. It is connected to Pin 19 of the PIC16F877A Microcontroller. It was biased a 1K Resistor. Below is the circuit diagram of the Buzzer Unit:



**FIG 3.6 The Alarm-Buzzer Unit Circuit**

**Buzzer Operating Current**

The buzzer operates with a voltage of 5-25V;

Current: 2mA to 10mA;

A pin of the microcontroller can supply 5V (as specified in Pic16f877a microcontroller datasheet)

Supply 5mA at 5V to the Buzzer:

From Ohms law: V=IR;

R=V/I

V=5V; I=5mA

R = 5/(5x10-3) = 1000;

Therefore R=1K ohms.