

People's Democratic Republic of Algeria  
Ministry of Higher Education and Scientific Research  
University M'Hamed BOUGARA – Boumerdès



Institute of Electrical and Electronic Engineering  
Department of Control and Power Engineering

Project Report Presented in Partial Fulfilment of  
the Requirements of the Degree of

**‘MASTER’**

**In Power Engineering**

Title:

**SCADA System For Power System  
Laboratory**

Presented By:

- **LALMI Ahmed**

Supervisor:

**Dr. Brahim Metidji**

Registration Number:...../2022

## *Acknowledgment*

In the Name of Allah, the Most Merciful, the Most Compassionate all praise be to Allah, the Lord of the worlds; and prayers and peace be upon Mohamed His servant and messenger.

First, gratefulness and praise be to God, the lord of the universe, the most gracious, the Most Merciful and the Most Beneficent for guiding me through all these years. I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my final year project supervisor, **Mr Metidji** for his help, guidance and passion through all the steps of my work. His expertise and insight have been influential in performing this research work. His teaching and work ethics are an inspiration I am infinitely grateful to my family, particularly my parents for their patience, unwavering support, continuous encouragement, and belief in me throughout my whole life. I would have never made it this far without them beside me, every step of the way.

I would like to extend my thanks to all my teachers who have done their best to transmit their knowledge and also all the staffs and the entire employees for their kindness and their contribution.

Furthermore, I would like also to express my greatest thankfulness to my friends who have been there to support and encourage me.

## *Abstract*

The present report shows the design and implementation of a SCADA system for power system laboratory enabled to acquire data from various power electronics modules connected into the same local area network. The main focus was to develop a reliable SCADA software that is able to monitor and analyze the electrical flow within the building to provide a more in-depth analysis of the electrical network. The software uses the famous *ModbusTCP* communication protocol to poll metric data from digital power analyzer units used in the laboratory, it is possible to display all measured values and operating states inside the system in real time on the existing measuring instruments, or even select, record and export the results to CSV or Excel files. The presented application is written in python3 programming language, and it uses Qt for the GUI and other libraries to manage communications and databases.

# Contents

<b>Acknowledgment</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>viii</b>
<b>General Introduction</b>	<b>1</b>
<b>1 SCADA System: Theoretical Background</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 What is a SCADA System? . . . . .	3
1.3 History of Automation . . . . .	4
1.4 Evolution of SCADA Systems . . . . .	5
1.4.1 First Generation of SCADA Systems: Monolithic Systems . . . . .	5
1.4.2 Second Generation of SCADA Systems: Distributed SCADA . . . . .	5
1.4.3 Third Generation of SCADA Systems: Networked SCADA . . . . .	6
1.4.4 Fourth Generation of SCADA Systems: Modern SCADA . . . . .	6
1.5 Fundamental Components of SCADA Systems . . . . .	7
1.5.1 Remote Terminal Unit (RTU) . . . . .	7
1.5.1.1 Components of RTU . . . . .	8
1.5.2 Intelligent Electronic Devices (IEDs) . . . . .	9
1.5.2.1 IED Block Diagram . . . . .	9
1.5.2.2 IED Advanced Functionalities . . . . .	10
1.5.3 Data Concentrator and Merging Units . . . . .	12
1.5.4 SCADA Communication Channel . . . . .	13
1.5.4.1 SCADA Network Topologies . . . . .	14
1.5.4.2 SCADA Communication Protocols . . . . .	15
1.5.4.3 Modbus Protocol . . . . .	17
1.5.4.4 Modbus Message Frame . . . . .	17
1.5.4.5 Modbus TCP . . . . .	18
1.5.4.6 MBAP Header Description . . . . .	19
1.5.4.7 Modbus Functions Codes Description . . . . .	19
1.5.5 SCADA Master Station . . . . .	20
1.5.5.1 Master Station Software Components . . . . .	20
1.5.5.2 Advanced SCADA Application Functions . . . . .	20

1.5.5.3	Master Station Hardware Components . . . . .	21
1.6	SCADA In Power System . . . . .	21
1.7	Advantages of SCADA Systems in Power Systems . . . . .	21
<b>2</b>	<b>SCADA Implementation for Power System Laboratory</b>	<b>23</b>
2.1	Introduction . . . . .	23
2.2	Hardware Components . . . . .	23
2.2.1	Adjustable Three-phase Power Supply . . . . .	24
2.2.2	Air Power Line . . . . .	25
2.2.3	Three-phase Resistive Load . . . . .	26
2.2.4	Three-phase Capacitive Load . . . . .	27
2.2.5	Three-phase Inductive Load . . . . .	28
2.2.6	Digital Power Analyzer . . . . .	29
2.3	Communication Channel . . . . .	30
2.3.1	Modbus TCP: . . . . .	30
2.3.1.1	Implementation of ModbusClient . . . . .	30
2.4	SCADA Software . . . . .	31
2.4.1	The Startup Page . . . . .	32
2.4.2	The Designer Window . . . . .	33
2.4.3	The Block Parameters . . . . .	35
2.4.4	Running a Project . . . . .	36
2.4.5	Diagram File Format . . . . .	37
2.4.6	The Database Viewer . . . . .	38
2.4.7	Plotting . . . . .	40
<b>3</b>	<b>Results and Discussion</b>	<b>42</b>
3.1	Introduction . . . . .	42
3.2	Communication Channel Test . . . . .	42
3.3	Transmission Line Tests . . . . .	44
3.3.1	Resistive Load . . . . .	45
3.3.2	Inductive Load . . . . .	49
	<b>Conclusion</b>	<b>52</b>
	<b>Appendix</b>	<b>53</b>
	<b>References</b>	<b>55</b>

# List of Figures

1.1	The monitoring and controlling process	4
1.2	Monolithic SCADA system	5
1.3	Distributed SCADA	6
1.4	Networked SCADA	6
1.5	Components of SCADA system	7
1.6	RTU internal architecture diagram	9
1.7	Structure of IED	10
1.8	MBDC-200: an autonomous Modbus data concentrator and collector	12
1.9	Merging units and IEDs in modern SCADA networks	13
1.10	Star topology	14
1.11	Bus topology	14
1.12	Ring topology	15
1.13	General MODBUS frame	17
1.14	MODBUS over TCP/IP	18
1.15	Use of SCADA in power systems	22
2.1	ELE-PAC3 : Adjustable three-phase power supply	24
2.2	ELE-APL : Air Power Line Module	25
2.3	ELE-RES : Three-phase resistive load module	26
2.4	ELE-CAP : Three phase capacitive load	27
2.5	ELE-IND : Three-phase inductive load	28
2.6	ELE-IND : Three-phase inductive load	29
2.7	SCADA software startup page	33
2.8	SCADA software Designer Tab	33
2.9	Available Modules list in the SCADA Software	34
2.10	The list of the different operations that can be performed to a block	35
2.11	The parameters of power meter block	35
2.12	The three different connection state for a block	36
2.13	Example of a running project	36
2.14	SCADA database viewer tab	38
2.15	Sample of the stored data into the database	39
2.16	Accessing the database using SQLite3 through terminal	40
2.17	Editing and viewing databases using DB Browser for SQLite3	40
2.18	Example of the generated graphs using Matplotlib	41
3.1	Configuring IP Address and Port for <i>PyModSlave</i>	43
3.2	Setting the output values range for Modbus server	43
3.3	Dashboard for simulation test	44
3.4	Bus Monitor output	44

---

3.5	Three-phase load signal line diagram . . . . .	45
3.6	TL with resistive load test . . . . .	46
3.7	Adding the circuit schematic to the SCADA designer . . . . .	46
3.8	Monitoring the system in SCADA dashboard . . . . .	47
3.9	Power analyzer reading . . . . .	48
3.10	Checking the reading with the multimeter . . . . .	48
3.11	Simulation result of signal phase . . . . .	49
3.12	Variable capacitive load connections . . . . .	50
3.13	Graph that shows the change in the voltages in an inductive load . . . . .	50
3.14	UML diagram for SCADA software . . . . .	53

# List of Tables

- 1.1 Industrial Automation Protocols . . . . . 16
- 1.2 MBAP Header content . . . . . 19
- 1.3 Modbus Functions and Registers . . . . . 20
  
- 2.1 Transmission Line module impedance values . . . . . 25
- 2.2 Available Modbus requests functions . . . . . 31
  
- 3.1 Measured and calculated values for resistive load test with  $V_s = 120V$  . . . . . 49

# List of Abbreviations

<b>ADC</b>	Analog to Digital Converter
<b>AGC</b>	Automation Generation Control
<b>API</b>	Application Programming Interface
<b>DA</b>	Distribution Automation
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>DMS</b>	Distribution Management System
<b>DNP3</b>	Distributed Network Protocol 3
<b>EMS</b>	Energy Management System
<b>GUI</b>	Graphical User Interface
<b>HMI</b>	Human Machine Interface
<b>IEC</b>	International Electrotechnical Commission
<b>IED</b>	Intelligent Electronic Device
<b>IoT</b>	Internet of Things
<b>LAN</b>	Local Area Network
<b>MU</b>	Merging Unit
<b>OPC UA</b>	Open Platform Communications Unified Architecture
<b>OSI</b>	Open Systems Interconnection Model
<b>PLC</b>	Programmable Logic Controller
<b>PMU</b>	Phasor Measurement Unit
<b>RTU</b>	Remote Terminal Unit
<b>SA</b>	Substation Automation
<b>SCADA</b>	Supervisory Control And Data Acquisition
<b>SOE</b>	Sequence Of Events
<b>SQL</b>	Structured Query Language
<b>THD</b>	Total Harmonic Distortion
<b>UML</b>	Unified Modeling Language
<b>WAN</b>	Wide Area Network

# General Introduction

Global electricity demand is increasing rapidly, necessitating the need for more dependable, environmentally friendly, and efficient transmission and distribution systems. Traditional grids and substations are no longer suitable for environmentally friendly power delivery and sustainable development. As a result, utilities are transitioning to the next-generation grid, which incorporates innovations in various fields of technology, allowing end users to have more flexible options while also empowering utilities to reduce peak demand and carbon dioxide emissions in order to become more efficient in all aspects.

In today's world of limited resources and rising energy demands, optimizing available resources is critical. Coal, water, and nuclear fuels are all depleting or raising environmental concerns. Renewable energy sources must also be used wisely. As a result, there is a need to optimize energy use and reduce waste. Automation of power systems is one solution toward this goal, and every sector of the power system, from generation to transmission to distribution to the customer, is currently being automated to achieve optimal energy and resource use [1].

Power system automation regulates power plant operations by optimizing against parameter variations to provide high efficiency and reliability depending on the demand of operation. Because of significant advancements in computer hardware and software, power system automation has become increasingly sophisticated. However, advances in sensors, amplifiers, recorders, control elements, valves, and measurement technologies have enabled the measurement and monitoring of many power operating parameters, providing a detailed picture of a plant's state in real time.

Many utilities still rely on manual labor to perform electrical distribution tasks that SCADA systems can easily automate. Automation facilitates smoother operations while minimizing disruptions, in addition to lowering labor costs. Modern SCADA systems include built-in redundancy and backup systems to ensure adequate reliability, and they can be much faster and more consistent than manual processes.

The problem with most SCADA systems available in the market and in the power system field is the level of difficulty for a student to access them and work and learn anywhere else outside the big power companies, and most of the used software programs are proprietary and extremely expensive for individuals. With noticeable movement towards the open industrial protocols and the development in the computer technologies, it is possible now days to develop a full interactive software application than can monitor and control power systems network for research centers and laboratories.

This report introduces theoretical and practical concepts regarding SCADA and SCADA software program for education purpose, illustrated and implemented on a power system equipment made by *Micro-tech Lab System Design*.

The report is divided into three chapters, the first chapter covers the theoretical background of SCADA systems, how SCADA works, fundamental components and the system usage in the power field. The second chapter describes the method and implementation of SCADA system that have been

introduced, the hardware components with their specifications and the software with all its features are being exposed. The last chapter represents the results for different tests which will show the system efficiency and reliability.

# Chapter 1

## SCADA System: Theoretical Background

### 1.1 Introduction

The following chapter will introduce the SCADA system and its impact in automating power system monitoring and control. The major components of the SCADA system are going to be discussed in detail.

### 1.2 What is a SCADA System?

SCADA is an acronym for the term “Supervisory control and data acquisition”, it is an automation technology that combines software and hardware to create a complete system that gives the user the sufficient information to identify the state of machines or processes and send a set of control instructions to the field remotely; Hence it is unnecessary for an operator to physically be presented with the operating machine.

SCADA systems are widely utilized in the oil and gas industry, as well as huge power systems, from generation to distribution. It enables the operator to alter set points on remote process controllers, open or close valves or switches, monitor alarms, and collect measurement data. When the size of the process grows to a wide variety of miles from one end to the other, the benefits of SCADA in terms of lowering the expense of routine visits to monitor facility functioning become apparent. If the facilities are really faraway and require a significant amount of effort to reach (e.g., a helicopter flight), the value of these benefits will increase even more [2, p. 9].

SCADA implementation entails two main functions: data collection for monitoring and supervisory control of processes, which leads to complete automation. By automating the monitoring and control actions, a process can be completely automated.

When the monitoring part is automated, an operator in a control room can "see" the remote process on the operator console, complete with all the necessary information displayed and updated at the appropriate time intervals. The following steps will be taken:

- Gather data from the field.
- Convert the data into a format that can be transmitted.
- Separate the data into packets.
- Send data packets over the communication medium.

- Gather data at the control center.
- Decode the information.
- Display the data on the operator's display screens at the appropriate points.

The following steps will be involved in automating the control process to ensure that the control command issued by the system operator is translated into the appropriate action in the field:

- The control command is initiated by the operator.
- Combine the control command with a data packet.
- Send the packet via the communication medium.
- The control command is received and decoded by the field device.
- Control action is initiated in the field by activating the appropriate device.

As shown in figure 1.1, the set of equipment measuring elements assists in acquiring data from the field, and the set of equipment controlling elements implements control commands in the field [1, p. 5].

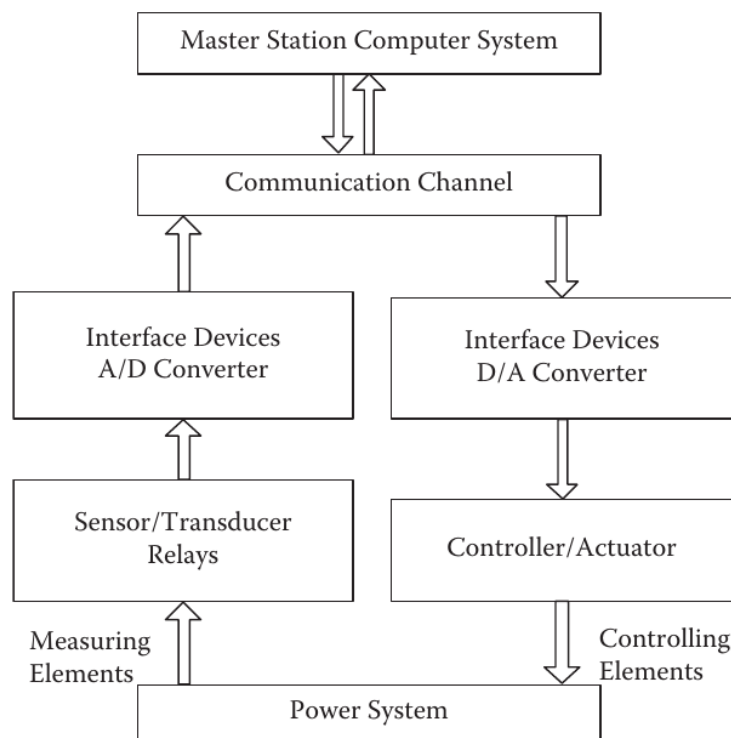


Fig. 1.1: The monitoring and controlling process

## 1.3 History of Automation

The history of automation systems may be traced back to the First Industrial Revolution (1750-1850), when human muscle power was supplanted by machine power. Process control was developed during the second industrial revolution (1850–1920), as machines took over the normal functions of the human intellect and constant presence. In 1821, Michael Faraday devised the electric motor, and in 1861–1862, James Clark Maxwell connected electricity and magnetism. Rapid breakthroughs in

electricity and electric power supply occurred in the latter 19th century, with contributions from giants such as Siemens, Westinghouse, Nikola Tesla, Alexander Graham Bell, Lord Kelvin, and many others [1, p. 2].

SCADA technology evolved slowly throughout the third industrial revolution, which occurred in the late 1970s. During this time, computers were brought into industry to automate processes. Hard-wired relay logic circuits were replaced with programmable logic controllers (PLCs). TCP/IP-based systems, such as ModbusTCP ProfiNET and OPC UA, have superseded domain-specific, proprietary field buses like CAN and Modbus [3]. that software advancements led in improved human-machine interfaces Report writers were created to deliver the information that was required at the time it was wanted. The systems grew in size. The introduction of powerful minicomputers had a significant impact on the development of SCADA, as it did with most industrial technologies [2, p. 27].

## 1.4 Evolution of SCADA Systems

### 1.4.1 First Generation of SCADA Systems: Monolithic Systems

These originated during the Mainframe era, when all data was kept on a single mainframe computer and all computations were performed on that computer. The systems were not interconnected in this generation. RTU suppliers usually created the communication protocols for transferring and receiving data from RTUs. They are responsible for scanning and regulating data within remote devices. Because all data was stored on a single mainframe unit, these systems were dubbed Monolithic.

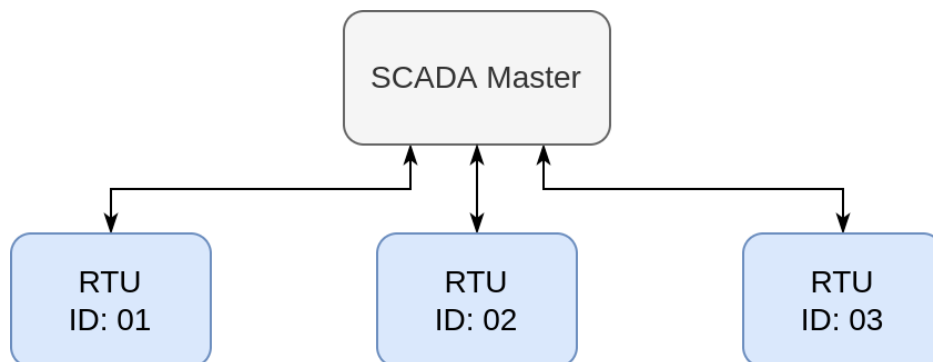


Fig. 1.2: Monolithic SCADA system

### 1.4.2 Second Generation of SCADA Systems: Distributed SCADA

Local Area Networks (LANs) were used commercially to share control functions that were distributed over various systems connected to each other. Distributed SCADA systems was the moniker given to these systems. When compared to monolithic systems, these distributed SCADA systems were small and inexpensive. Due to redundancy and the use of dependable LAN protocols, the processing performance of SCADA systems improved.

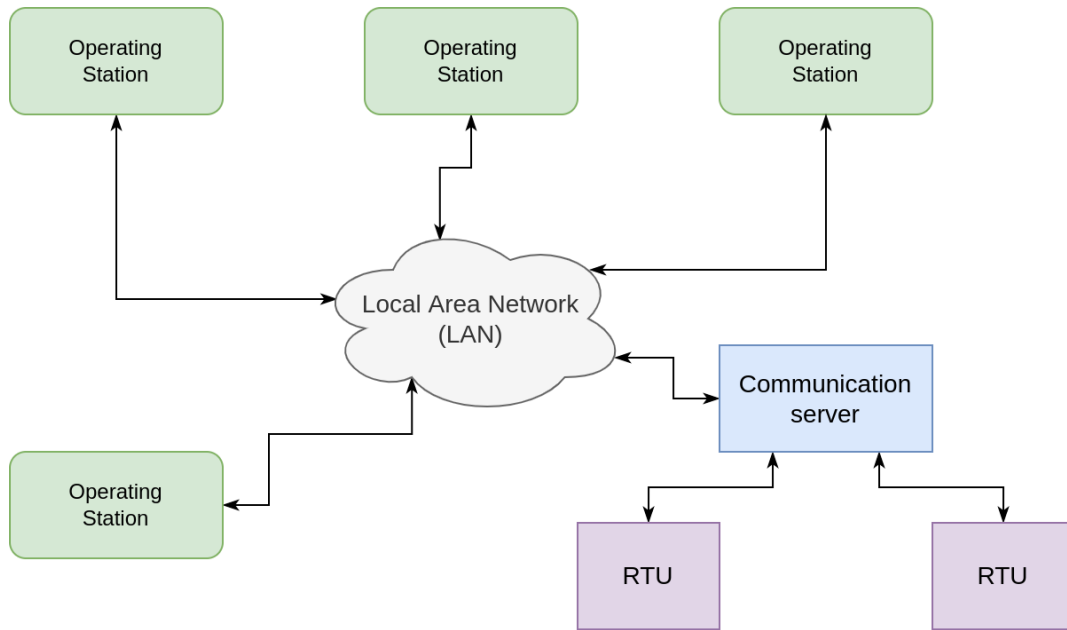


Fig. 1.3: Distributed SCADA

### 1.4.3 Third Generation of SCADA Systems: Networked SCADA

SCADA systems may be able to overcome the restrictions of proprietary systems as open source Internet protocols for communication evolve. Third-generation SCADA systems were typically networked and communicated over data lines or the phone utilizing Wide Area Network (WAN) Systems. Data was transmitted using Ethernet or Fiber Optic connections, and the devices were monitored and controlled using Programmable Logic Controllers (PLCs). Network SCADA systems, unlike monolithic and distributed SCADA systems, were exposed to risk since they were connected over the Internet.

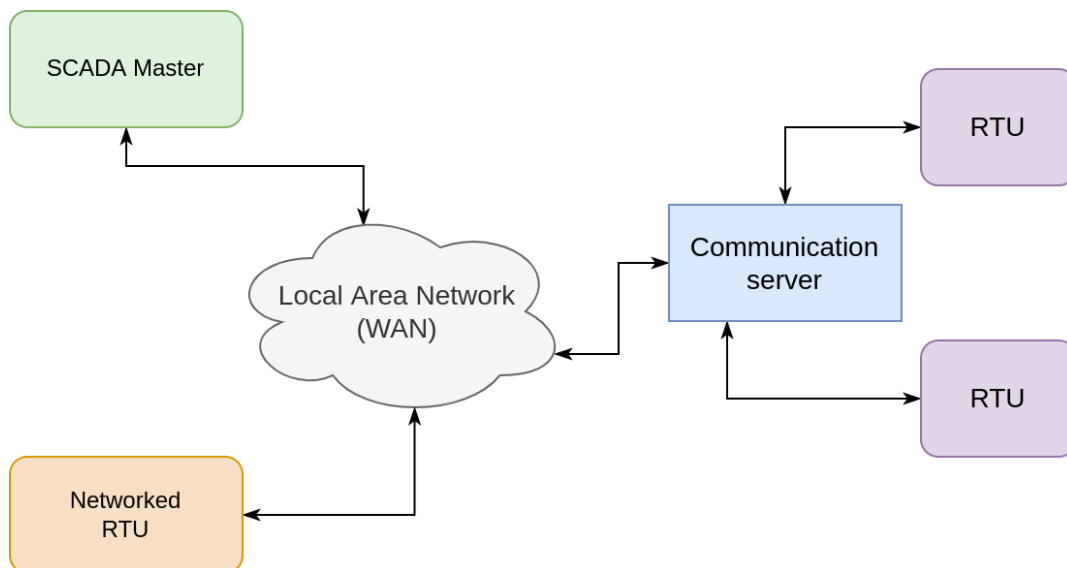


Fig. 1.4: Networked SCADA

### 1.4.4 Fourth Generation of SCADA Systems: Modern SCADA

SCADA systems have grown more dynamic and cost-effective as a result of the evolution of cloud-based computing approaches and numerous open source IoT platforms. SCADA systems of the present

generation are simple to maintain and integrate with other components. They grew more adaptable and scalable, allowing them to manage data of any size or complexity, as well as computing obstacles [4][5].

## 1.5 Fundamental Components of SCADA Systems

Modern SCADA system consist of three main components, The first is the remote terminal unit (RTU) or data concentrator, which is the control system's link to the field and is responsible for collecting data from field devices and relaying control commands from the control station to the field devices. Traditional RTUs' hardwired input and output (I/O) points are being replaced by wireless I/O points (IEDs). Today, the SCADA system designer must be able to handle legacy systems with only RTUs, hybrid systems with RTUs and IEDs, and new systems with solely IEDs with ease.

The communication system is the second component, and it transports monitored data from the RTU to the control center as well as control commands from the master station to the RTU or data concentrator for transmission to the field. Because the power system field is widely distributed across the landscape, and critical information that is time bound must be communicated to the master station and control decisions to the field, the communication system is extremely important in SCADA in general and power automation in particular.

The third component of the SCADA system is the master station with the SCADA software where the operator monitors the system and makes control decisions to be transmitted to the field. The human machine interface (HMI), which is the interaction between the operator and the machine, can be replaced by the software's graphical user interface (GUI). These three components are included in all automation systems, in varying quantities based on the process needs.

The sections that follow will explain how the components of the SCADA system work together to monitor and regulate the process to achieve optimum system performance [1, p. 23].

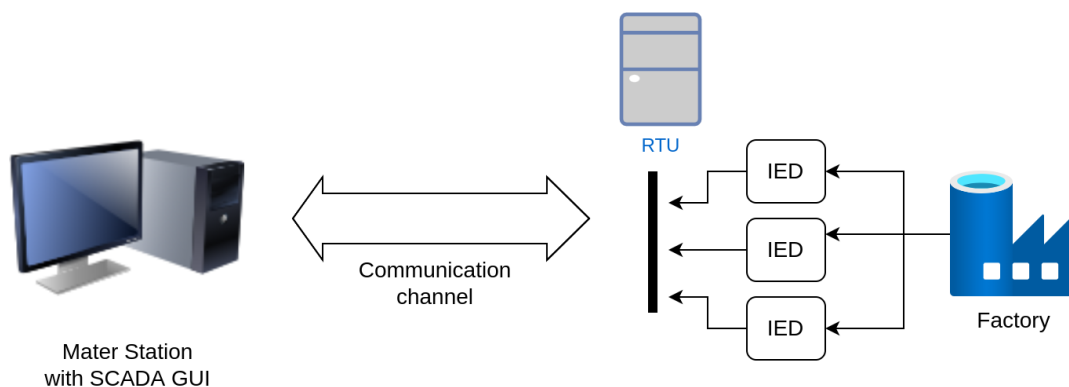


Fig. 1.5: Components of SCADA system

### 1.5.1 Remote Terminal Unit (RTU) [1, p. 24]

The RTU is the SCADA system's eyes, ears, and hands. RTUs used to be slaves to the master station, but now they have inherent computational and optimization capabilities. RTU receives data from field devices, analyses it, and sends it to the master station via the communication system to aid in power system monitoring as the master station's "eyes" and "ears." The RTU receives control commands from the master station and transmits them to the field devices at the same time, validating the parallel to the master station's "hands".

The RTUs were programmed using microprocessor logic. This gave supervisory systems more flexibility and opened up new operational and performance possibilities. The ability to build modular systems with largely preprogrammed user interfaces that are easy to adapt to individual processes, as well as the support of industry standard communication protocols (IEEE 1815 or DNP3, IEC 60870-5-101 and 103) make data exchange with the control station relatively simple.

#### 1.5.1.1 Components of RTU

To perform the functions of monitoring and controlling field equipment, the RTU has the following essential components, which are presented in figure 1.6:

- **Communication Subsystem:** The communication subsystem connects the SCADA communication network to the RTU's internal logic. This subsystem receives messages from the master, interprets them, and initiates actions within the RTU, which in turn initiates some field action. RTU also sends an appropriate message to the master station when the task is completed. It also collects data from the field, processes it, and transmits it to the master station. RTU can report to either a single or multiple masters.
- **Logic Subsystem:** The logic subsystem, which includes the main processor and database, is responsible for all major processing, time keeping, and control sensing. In most cases, the logic subsystem also handles analog to digital conversions and computational optimization.
- **Termination Subsystem:** The termination subsystem connects RTU to external equipment such as communication lines, primary sources, and substation devices. The RTU logic must be protected from the harsh substation environment.
- **Power Supply Subsystem:** The power supply converts primary power, which is typically supplied by the substation battery, to the supply requirements of the other RTU subsystems.
- **Test/HMI Subsystem:** This subsystem includes a variety of components, including built-in hardware/firmware tests and visual indicators, as well as built-in or portable test/maintenance panels or displays, within the RTU.

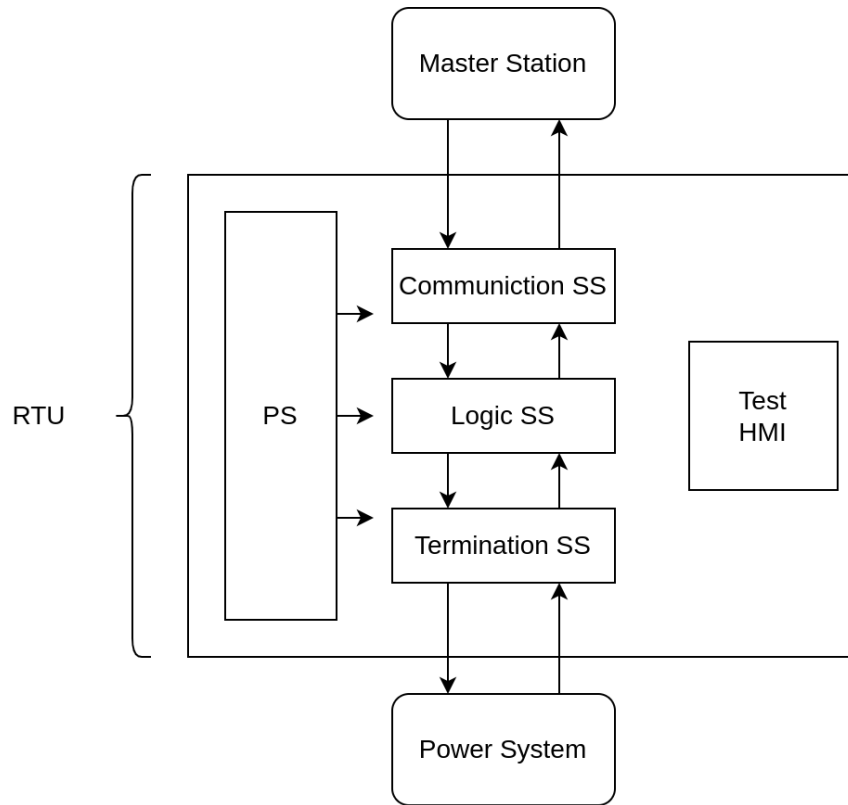


Fig. 1.6: RTU internal architecture diagram

## 1.5.2 Intelligent Electronic Devices (IEDs)

An IED is any device that contains one or more processors capable of receiving or sending data/control from or to an external source (e.g: electronic multifunction meters, digital relays, controllers). IED technology can assist utilities in increasing reliability, gaining operational efficiencies, and enabling asset management programs such as predictive maintenance, life extensions, and improved planning [6].

They provide numerous useful features such as protection and metering, as well as superior computing power. It can also record a sequence of events for use in post-event analysis and fault detection, as well as record waveforms and measure power quality.

As a result of the various functions it provides, it is an important component in the automation of substations and efficient monitoring in power delivery systems. Simply put, power system automation appears to be impossible without IEDs. The integration and interoperability features of IEDs have caused a shift from RTUs to IEDs [7].

### 1.5.2.1 IED Block Diagram

Intelligent electronic devices' architectural design ensures excellent features such as flexibility, adaptability, multipurpose and modular nature, and robust communication capabilities. IED hardware is designed to use draw out type cards, which has the advantage of allowing replacement without disconnecting the terminal wires and removing the IED from the panel.

The IED software architecture is designed so that the commissioning engineer can independently evaluate and program the available functions. The required function can be selected, while the other functions are deactivated and are not visible to personnel, saving time during commissioning. Each

selected function is an independent embedded unit with dedicated logical inputs and outputs, setting, and event reporting features that is generally associated with the IED.

Figure 1.7 depicts the functional blocks in an IED, demonstrating the device's versatility. In addition to analog and digital inputs and outputs, the IED has waveform capture and disturbance analysis capabilities. Metering and demand value recording are additional features, in addition to the IEDs' programmable logic capability, which eliminates the need for an additional PLC. The device's self and external circuit monitoring make it extremely reliable and reduces downtime [1, p. 38].

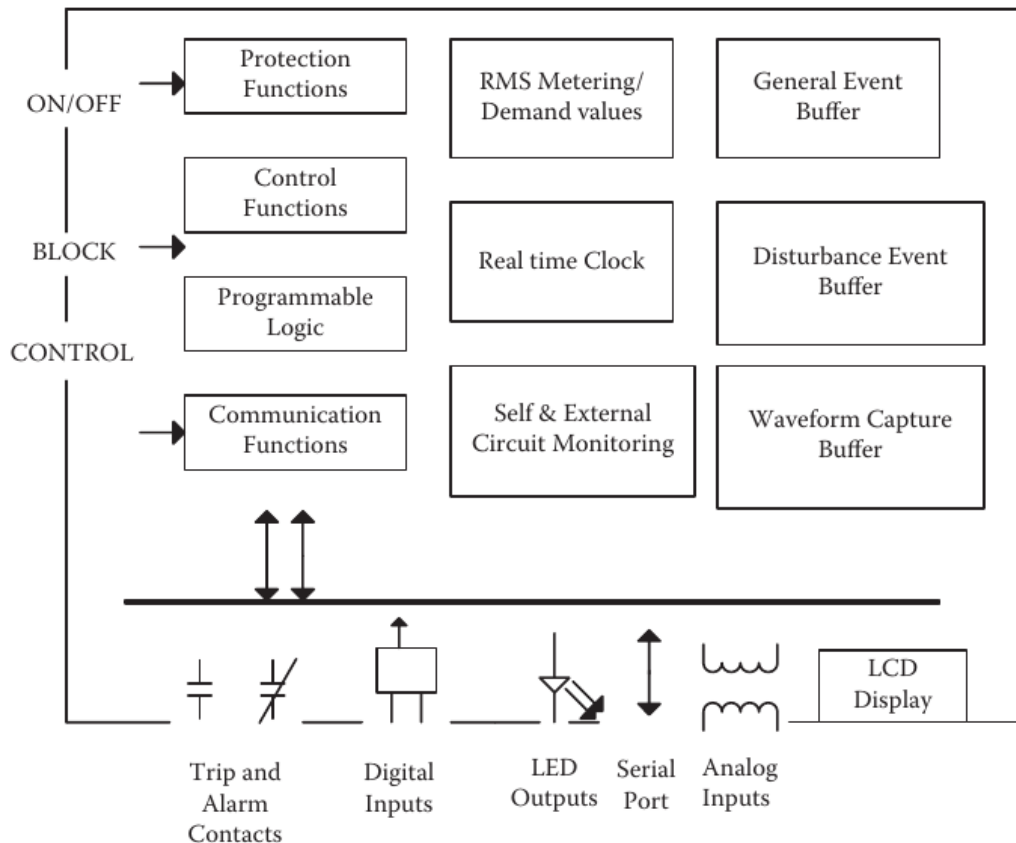


Fig. 1.7: Structure of IED

### 1.5.2.2 IED Advanced Functionalities [8]

#### *Protection function including Phasor estimation:*

As IEDs are primarily an improvement on microprocessor-based relays, the primary function of a relay IED is protection. The new IEDs are vastly improved, with more precise measurement principles and less auxiliary equipment required. Because the original primary transformer currents are available for further analysis in a modern relay IED, auxiliary CTs can be eliminated in a transformer differential relay. Similarly, the CT mismatch, inrush, and CT saturation problems can be solved without the use of external devices using appropriate techniques and numerical comparison algorithms.

The generalized numerical relay theory, which is primarily obtained from open system relaying, comprised of a minimal set of modern digital and numerical relay hardware modules and functions. The majority of modern digital and numerical relay equipment can be recreated using this concept and the amount of information readily available. The following data processing modules constitute the generalized numerical relay:

1. Isolation and analog signal scaling: Acquired current and voltage waveforms from instrument transformers are scaled down to usable voltage levels for use in digital and numerical relays.
2. Analog anti-aliasing filtering: Aliasing occurs when the high-frequency components of the inputs appear to be parts of the fundamental frequency components. Low pass filters are often used to eliminate this phenomenon.
3. Analog to digital conversion: Because digital processors can only handle numerical or logical data, input waveforms must be sampled at discrete intervals. To accomplish this, each analog signal is routed through a "sample and hold" module before being routed to an analog to digital converter (ADC) by a multiplexer one at a time.
4. Phasor estimation algorithm: A microprocessor-based software algorithm calculates the amplitude and phase of the waveforms fed into the relay. This is critical in modern monitoring systems, where phasor metrics are becoming an essential part of system monitoring. This IED feature is used to compute the phasor of a voltage or current in relation to a reference phasor. (The IEDs are time-synchronized using a single GPS source.) This is known as a phasor measurement unit (PMU), and the phasor data is collected by a phasor data concentrator (PDC) at a suitable point, usually in the control center.
5. Relay algorithm and trip logic: The protection algorithm's equations and parameters, as well as the associated trip logic, are applied in the software of the relay's microprocessor. The microprocessor computes the phasors representing the inputs, obtains the status of the switches, calculates protective relays, and finally provides output signals for controlling the circuit breakers. Communications, self-testing, target display, time clocks, and many other tasks may also be supported by the processor.

***Programmable logic and breaker control:***

A modern relay IED eliminates the need for external programmable logic controllers (PLCs) because the IED can manage the logical inputs and outputs of the protection functions, which can be directly connected to the IED's flip flops and/or gates.

***Metering and power quality analysis:***

Metering capabilities of IEDs were quickly accepted by power utilities, and significant cost savings were realized by merging the non-revenue metering function into the IEDs. It should be noted that the primary CTs and PTs used for protection may not be precise enough for normal current measurement in revenue metering. Measuring the voltage and current root mean square (RMS) values, as well as the real and reactive power, are all standard metering functions.

Metering, in addition to these basic functions, includes values for commissioning and testing, which reduces commissioning and testing times on the job site. Metered values are the positive, negative, and zero sequence components of voltage and current phase shifts, including the standard RMS values. To expedite the commissioning process, the phase mismatch, differential, and restriction values can be easily computed.

Another metering activity that can be accomplished with IEDs is load profiling, in which the power factor profile, ampere demand, long-term RMS voltage value, and so on can be monitored and used for load profiling for long-term expansion planning.

***Self-monitoring and external circuit monitoring:***

IEDs have internal card-level diagnostic capabilities, as well as self-monitoring software that can detect up to 98 percent of problems such as hardware failure, memory failure, and power supply problems. In addition to internal monitoring, modern IEDs have interface monitoring and external

circuit monitoring capabilities. Interface monitoring entails inspecting the inputs to the IEDs, which can be verified using simple methods. For example, the three-phase input currents to the relay should add up to three times the neutral current, if present. If there is any deviation, the analog channel of any of the currents may be defective. The relay can prevent false trips. External circuit monitoring will include checking the circuit breaker coil for any interruptions in the trip-close path, which can also indicate a failure of the instrument transformer.

### ***Event reporting and fault diagnosis:***

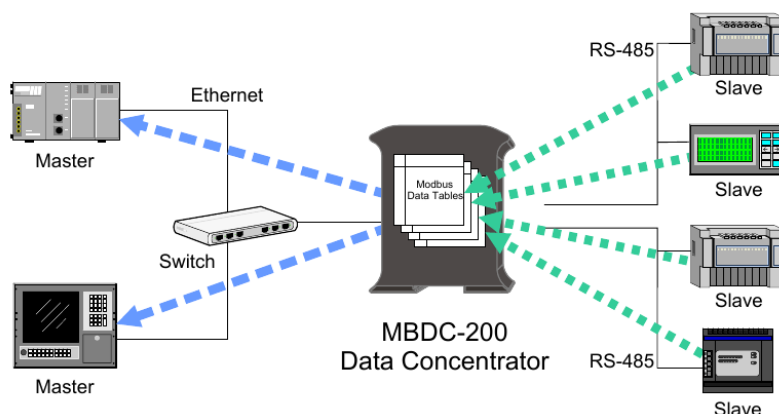
Because they can capture waveforms during a fault, relay IEDs replace digital fault recorders, whereas electromechanical relays do not. Relay IEDs, which eliminate the need for sequence of events (SOE) recorders, make event reporting simple. The relay IEDs store captured data in nonvolatile memory, so disturbance event reports (pick up, trip, and auto-reclose), along with the overall event reports like setting changes, must be saved and managed separately.

All events are timestamped by the IEDs, which require GPS synchronization as well as a battery backup for the real-time clock. Once the events have been correctly time-tagged, they can be reported in the order that they occurred, eliminating the need for additional sequencing in the control room. Because the values are stored in the IED and can be recovered later, even in the event of a blackout, performing fault diagnosis after a fault is simple.

## **1.5.3 Data Concentrator and Merging Units**

A data concentrator is the core of data and energy management in an automated metering infrastructure. It provides the technology to measure and collect energy usage data. The concentrator can also be programmed to analyze and communicate this information to the central utility database. Not only can the utility providers could use this information for billing services, but can improve customer relationships through enhanced consumer services such as real-time energy analysis and communication of usage information [9].

Data concentrators in a substation collect data from IEDs and other field inputs and can provide temporary or permanent information to the higher hierarchy. As previously discussed, the hard-wires coming from the field carrying analog values and status points that terminate at the IED are converted to all digital values. Figure 1.9 depicts traditional RTUs and newer IEDs with data concentrators in a substation. The IEDs send the required information to the higher hierarchy via a communication protocol to the data concentrator. As shown in Figure 1.8, the data concentrator communicates via a LAN [1, p. 46].



**Fig. 1.8:** MBDC-200: an autonomous Modbus data concentrator and collector [10]

Merging units are an essential component of the process bus concept, allowing the digitization of analogue, binary, and command information into the IEC 61850 format, allowing for end-to-end digitization and the creation of a fully digital substation [11].

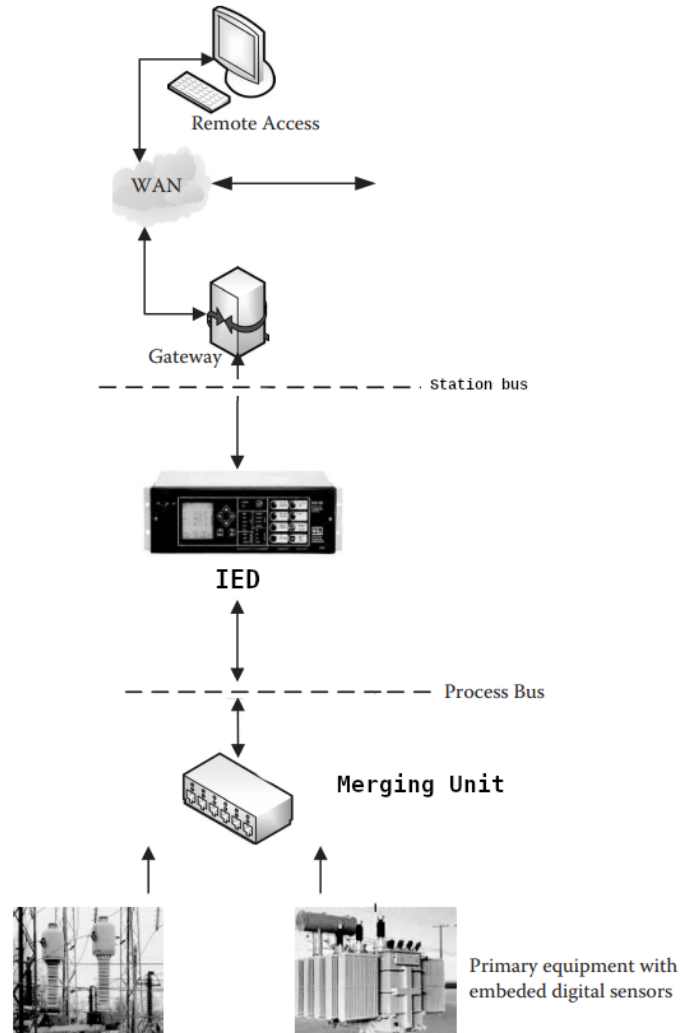


Fig. 1.9: Merging units and IEDs in modern SCADA networks

### 1.5.4 SCADA Communication Channel

A SCADA system cannot exist without a properly designed communication network system. All aspects of supervisory control and data acquisition rely entirely on the communication system to provide a conduit for data flow between the supervisory controls, data acquisition units, and any controllers that may be connected to the system. A communications network in a SCADA system connects the remote terminal units (RTUs) or IEDs to the SCADA Master station [12].

The channel allows a remote control center to access field data in real time to assess the state of the system, such as the generation by each unit, voltage and current vectors from buses, system loading, and circuit breaker and isolator positions. The communication channel also transports control commands from the control center to the appropriate field equipment for implementation, ensuring the stability and security of the power system [1, p. 75].

### 1.5.4.1 SCADA Network Topologies [13]

Direct connect and multi-drop are the two types of data link connections that can be used to build networks. There are only two devices connected via network media in a direct connection, which can be metallic, wireless, or fiber. At each device, each interface consists of a separate transmit and receive connection. Because there are only two devices, each of them can control the connection on which they are transmitting at all times, and both can know implicitly to which other device they are connected. Several individual direct connections were made at the same time. Many direct connection originating from one device is called a star network topology. Figure 1.10 illustrates the star topology. Many star networks can be connected in a parallel or vertical hierarchy.

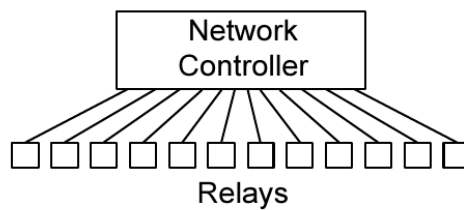


Fig. 1.10: Star topology

For direct connections in a star topology, any protocol, including those designed for multi-drop applications, can be used. To support direct connection, almost all IEDs have a simple RS-232 serial port connection. In a direct connection, any of the other communication methods can also be used.

Star network designs are compatible with a wide range of IED capabilities. Simple, sluggish IEDs can coexist with more complex, fast-talking IEDs. Because each device has a dedicated direct connection, devices from different manufacturers and with different protocols can coexist in the same star network.

The term "open architecture" corresponds to networks that are fully compatible between hardware and software interfaces, and thus between vendors. The star network is the only true open architecture design that can support multiple protocols, baud rates, and network interfaces.

Several devices can be physically connected in a bus or ring network in a multi-drop network topology, and control of the transmit and receive connection must be negotiated. Figure 1.11 depicts IEDs connected in a bus topology, while Figure 1.12 depicts IEDs connected in a ring topology. Only one device can communicate at a time with a multi-drop connection.

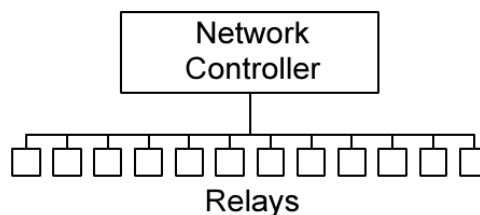
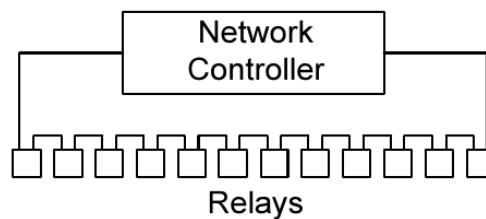


Fig. 1.11: Bus topology



**Fig. 1.12:** Ring topology

So that data does not collide on the conductor, software and hardware are used to decide which device has permission to transmit. Because several devices are connected, addressing within the protocol is required to identify the source and destination of the data being communicated. This addressing adds overhead in the form of processing time and the amount of information that must be transmitted, resulting in a decrease in data transfer rate. Devices reimburse for this by increasing their communication speed and the amount of communications processing they perform.

It is difficult to troubleshoot communication issues on a multi-drop network. Messages from various sources must be intercepted and deciphered. With the help of a simple LED indicator, direct connections can be quickly and easily verified.

IEDs have different memory and computational capacities, and thus different protocol support capabilities. On a multi-drop network, interactions must be performed at the lowest common denominator, and all devices must support the same baud rate and physical network connection.

It is critical to remember that if the mediation of control of data transmission fails, none of the multi-dropped devices will be able to communicate. This can occur as a result of IED communications hardware failing to reveal control, IED communications software failing to correctly process mediation schemes, or network corruption.

RS-485, a twisted pair network “daisy-chain” connection between multiple devices, is a common multi-drop option. For use on a multi-drop network, RS-232 connections can be converted to RS-485. Other copper and fiber connections are being developed to support both standard and proprietary interfaces such as Ethernet. A broadcast multi-drop network is a common network with distinct functions and purposes. One-sided conversations can simply be sent to multiple unresponsive receiving devices. Time-synchronization messages from the inter-range instrumentation group (IRIG) are frequently sent to relays in this manner. Separate connections on the IED are frequently required for this purpose.

#### 1.5.4.2 SCADA Communication Protocols

The communication protocol defines the format in which data is transferred from one device to the other, making communication easier because both devices can decode the data received and multiple data can be transferred using the same channel using the techniques discussed previously.

A protocol specifies the exact procedure for adding additional layers to data, which differs from protocol to protocol. In the beginning, this resulted in a large number of protocols evolving in response to specific user requirements. Interoperability was jeopardized, however, as proprietary protocols proliferated and international organizations stepped in to define common protocol formats.

The operation of each layer of the communication protocol is clearly defined in the Open System Interconnection (OSI) reference model published by the International Organization for Standardization [ISO] in 1984 and updated in 1994. TCP/IP (Transmission Control Protocol/Internet Protocol) was

also developed concurrently, and some SCADA protocols are based on TCP/IP. The International Electrotechnical Commission (IEC) adopted the OSI model for SCADA communication as Enhanced Performance Architecture (EPA), and open SCADA communication protocols were developed using EPA. IEC 61850 was recently developed with full OSI layers for power system SCADA applications [1, p. 89].

There are numerous popular protocols used in the power system SCADA today. DNP3 (IEEE 1815) is used in North America for communication from the master station to the field equipment, while IEC 870-5-101 (T101) serial and 104 (TCP/IP) are used in Europe and by European vendors. IEC 61850, DNP3 (IEEE1815), and Modbus are also commonly used for communication between field equipment [14].

Some of the more popular and widely used communication protocols are listed in Table 1.1:

**Table 1.1:** Industrial Automation Protocols

Protocol	Developer	First Introduced	Maximum Speed
Industrial Ethernet	Xerox	1976	10 Gbps
Modbus	Modicon	1970	38 Kbps
Modbus TCP/IP	Modicon	1999	100 Mbps
Modbus RTU	Modicon	1979	19.2 Kbps
Modbus Plus	Modicon	1991	57.6 Kbps
HART	Emerson	1986	1.2 Kbps
SDS	Honeywell	1994	1 Mbps
Fieldbus	Fieldbus Foundation	1995	100 Mbps
CAN Open	CAN Automation	1993	20 Kbps
AS-i	Factory Automation Suppliers, Germany	1994	167.5
BSAP	Bristol Babcock	-	250 Kbps
CC Link	Mistubishi Electric	1997	1 Gbps
CIP	ODVA	1992	-
EtherCAT	Beckoff	2015	-
EGD	GE Fanuc	1998	-
Ethernet/IP	Rockwell Automation	2001	-
HostLink	Omron	-	9600 Bps
Optomux	Opto22	1982	38.4 Kbps
Pie-P	-	-	-
Profibus	Profibus International	1989	500 Kbps
Profinet IO	Profibus International	-	-
Rapienet	LSIS	-	100 Mbps to 1 Gbps
SERCOS III	Sercos International	2003	100 Mbps
SERCOS Interface	VDW/ZVEI	1987	-
GE-SRTP	GE Fanuc	-	-
MPI	Seimens	1991	-
ControlNet	Rockwell Automation	1997	5 Mbps
DF-I	Rockwell Automation	-	-
DeviceNet	Rockwell Automation	1994	500 Kbps
DirectNet	Koyo	1999	38.4 Kbps

### 1.5.4.3 Modbus Protocol [15], [16], [17], [1, p. 102]

Gould-Modicon invented Modbus as a proprietary protocol for PLC communication. Thousands of vendors now use it as a 'open' protocol. It is primarily used for local serial connections and is unsuitable for use as a telemetry protocol. Modbus is a straightforward master-slave protocol that employs the cyclic polling method. It applies to the OSI stack layers 1, 2, and 7.

Unlike the many other buses protocols, no interface is defined in the Modbus protocol. As a result, the user has the option of using EIA-232, EIA-422, EIA-485, or a 20 mA current loop, all of which are compatible with the protocol's transmission rates.

Although the Modbus is slower than other buses, it has the advantage of widespread acceptance among instrument manufacturers and users. Around 20 to 30 manufacturers produce equipment that uses the Modbus protocol, and many systems are in use in industry. As a result, it can be considered a de facto industrial standard with proven capabilities. According to a recent survey published in the well-known American Control Engineering magazine, the Modbus protocol is used for interfacing in more than 40 percent of industrial communication applications.

However, as a protocol for electrical substation communications, Modbus is severely limited. The Modbus protocol operates on the master/slave principle, with one master and up to 247 slaves. A transaction is initiated only by the master.

Messages are sent in frames, and each frame contains four fields: address, control, message, and error check. A Modbus application data unit is depicted in Figure 1.13.

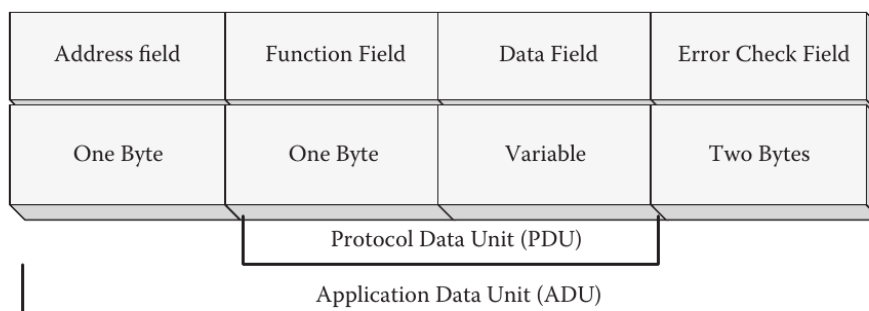


Fig. 1.13: General MODBUS frame

### 1.5.4.4 Modbus Message Frame

The address of the slave device is in the address field of a query or request sent by the master, which uses one byte. Modbus addresses can range from 1 to 247. The address 0 is used for the broadcast message to all devices. So that the master knows which slave is responding, the slave enters its own address in the response field. The function field will contain the slave device's control task, such as read or write one byte or read event counters. The length of the data field will vary, and the master request will contain the information needed to complete the function. The CRC code will be in the error check field, and the slave will check the data for errors before accepting and executing the command.

The slave's response frame will have similar fields and will send the information requested by the master. The master can address individual slaves or send a message to all slaves at once during communication. When addressed individually, the slaves respond to the master as a query expecting a response, whereas no response is sent when a broadcast message is sent to all slaves. A master

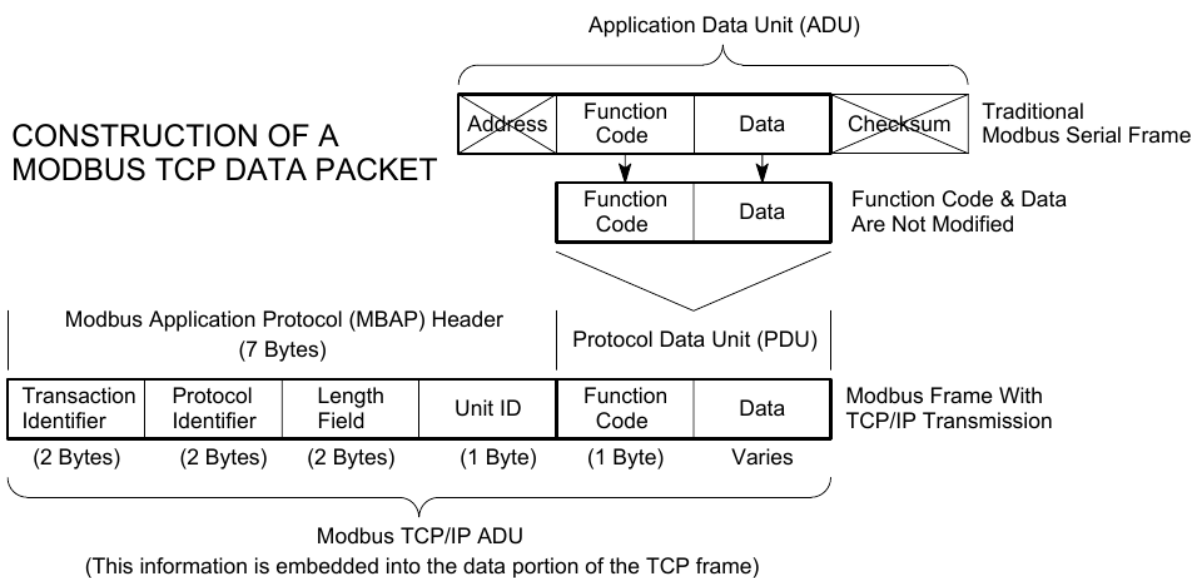
can handle up to 247 slaves in a SCADA system, but most masters will only have a few slaves. The Modbus protocol has fixed characteristics such as frame format, frame sequence, error checking (CRC), and exceptions, whereas the transmission mode, transmission media, and characteristics are selectable.

In a Modbus network, there are two asynchronous transmission modes: ASCII and RTU. RTU mode is smaller and faster, and it is used for routine operations. The RTU mode employs the message frame discussed above. The system is tested in American Standard Code for Information Interchange (ASCII) mode. The message frame has seven characters and the address field has two.

### 1.5.4.5 Modbus TCP

Modbus can also be implemented using TCP/IP over Ethernet, which is now widely used in station LANs. On TCP/IP, a dedicated header is used to identify the MODBUS Application Data Unit. It is known as the MBAP header (MODBUS Application Protocol header). This header differs from the MODBUS RTU application data unit used on serial lines in the following ways:

- The MODBUS slave address field: which is normally used on the MODBUS Serial Line, is replaced by a single byte 'Unit Identifier' within the MBAP Header. The 'Unit Identifier' is used to communicate via bridges, routers, and gateways that use a single IP address to support multiple independent MODBUS end units.
- All MODBUS requests and responses are designed so that the recipient can confirm that a message has been completed. The function code alone is sufficient for function codes where the MODBUS PDU has a fixed length. The data field includes a byte count for function codes that carry a variable amount of data in the request or response.
- When MODBUS is transmitted over TCP, extra length information is carried in the MBAP header to allow the recipient to recognize message boundaries even if the message has been split into multiple packets for transmission. The presence of explicit and implicit length rules, as well as the use of a CRC-32 error check code (on Ethernet), results in a negligible chance of undetected corruption of a request or response message.



**Fig. 1.14:** MODBUS over TCP/IP

### 1.5.4.6 MBAP Header Description

According to the official Modbus TCP/IP implementation guide the MBAP Header contains the following fields:

- *Transaction Identifier* - It is used for transaction pairing, the MODBUS server copies in the response the transaction identifier of the request.
- *Protocol Identifier* – It is used for intra-system multiplexing. The MODBUS protocol is identified by the value 0.
- *Length* - The length field is a byte count of the following fields, including the Unit Identifier and data fields.
- *Unit Identifier* – This field is used for intra-system routing purposes. It is typically used to communicate to a MODBUS+ or a MODBUS serial line slave through a gateway between an Ethernet TCP-IP network and a MODBUS serial line. This field is set by the MODBUS Client in the request and must be returned with the same value in the response by the server.

**Table 1.2:** MBAP Header content

Fields	Length	Description	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client ( request)	Initialized by the server ( Response)
Unit Identifier	1 Bytes	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

### 1.5.4.7 Modbus Functions Codes Description

The Modbus database schema has a straightforward structure that distinguishes only four basic data types:

- Discrete Inputs
- Coils (Outputs)
- Input Registers (Input Data)
- Holding Registers (Output Data)

A device's Modbus registers are organized around the four basic data reference types mentioned above, and this data type is further identified by the leading number of the reference address, as shown in table 1.3:

**Table 1.3:** Modbus Functions and Registers

CODE	FUNCTION	REFERENCE
01 (01H)	Read Coil (Output) Status	0xxxx
03 (03H)	Read Holding Registers	4xxxx
04 (04H)	Read Input Registers	3xxxx
05 (05H)	Force Single Coil (Output)	0xxxx
06 (06H)	Preset Single Register	4xxxx
15 (0FH)	Force Multiple Coils (Outputs)	0xxxx
16 (10H)	Preset Multiple Registers	4xxxx
17 (11H)	Report Slave ID	Hidden

## 1.5.5 SCADA Master Station

SCADA master stations range in size from small control rooms in substations to large transmission SCADA master stations that manage a country's power flow. The master station is a collection of computers, servers, peripherals, and I/O systems that assist the operator in monitoring the state of the field and initiating control actions when necessary [18].

The master station components are divided into two categories; hardware and software.

### 1.5.5.1 Master Station Software Components

The master station software components are divided into two categories: basic SCADA functions and advanced application functions specific to the SCADA implementation, such as generation SCADA, transmission SCADA, or distribution SCADA applications.

The basic SCADA software performs the fundamental functions of a SCADA system and is shared by all SCADA applications such as:

1. **Data Acquisition and Control:** This covers the fundamental SCADA functions of data acquisition and control. This software includes the fundamental modules for project engineering and commissioning.
2. **Database:** Because historical data is critical, the power system SCADA requires a dedicated database management system. Established databases provide quick access to data on any query and can be customized to meet the needs of the master station.
3. **Reporting and Accounting:** Because of the hierarchical nature of the power system, numerous reports and accounts must be prepared for submission to various agencies as well as for internal use. As a result, this function is critical in power applications. The reporting and accounting software can be predefined, but it should be adaptable to future needs.
4. **HMI Functions:** The software on the operator's console is critical, and it must be user friendly in order for the control center to function properly [1, p. 47].

### 1.5.5.2 Advanced SCADA Application Functions

The Symphony® Plus SCADA is an example of an advanced SCADA platform used for geographically distributed systems, the following features are listed in the main product web page:

1. **Advanced alarm management system :** Ensures each alarm is meaningful, and that it alerts, informs, guides operators to take proper action.

2. Geographical Information System (GIS) : Helps operators in effectively locating an abnormal situation on a plant network to take immediate actions to respond to the situation.
3. Advanced analytics and applications : Analytics Engine with performance algorithms to analyze and predict plant maintenance issues.
4. Flexible and scalable platform: Flexible and highly scalable, adapts easily to the growing needs.
5. Security: Designed to include the latest cybersecurity standards, assuring a safe and secure system through its lifecycle.

### 1.5.5.3 Master Station Hardware Components

The main hardware in a master station will be the computer and server systems used to execute the various tasks that the master station will perform. The computer servers must be chosen in accordance with the needs of the master station.

The number of servers and workstations depends on the field and application of each specific SCADA system, and it differs a lot from one area to another. The servers are linked by a high-speed, dual-redundant LAN. In this client-server environment, data from one server can be obtained by another. The dedicated server systems will have unique capabilities and features that make them ideal for a specific application. Some of these features include faster CPUs, increased high-performance RAM, power supply redundancy, network connections [1, p. 48].

## 1.6 SCADA In Power System

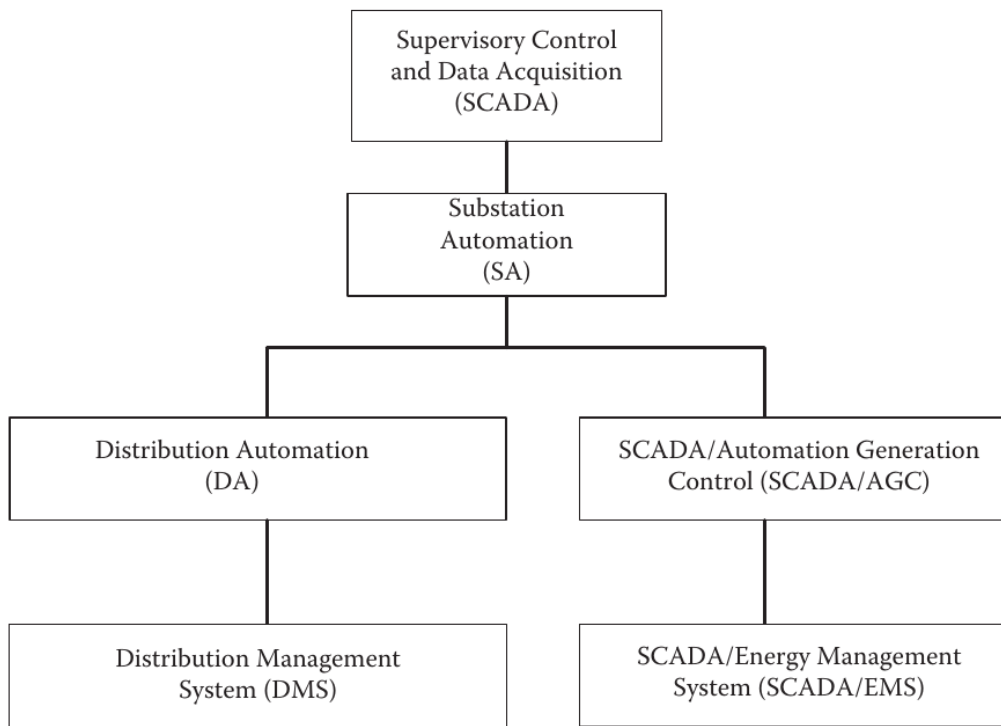
SCADA systems are used in all aspects of power system operations, from generation to transmission, distribution, and use of electrical energy.

Figure 1.15 depicts the application of SCADA in power systems, with the first SCADA block showing the basic functions. The figure's right section represents generation SCADA, represented by SCADA/AGC (Automatic Generation Control), which is used in generation control centers around the world. Furthermore, the transmission SCADA is illustrated as SCADA/EMS (Energy Management Systems), with basic functions supplemented by energy management system functions. This is implemented in the transmission control centers. Because of the complexity of each application, the EMS software applications are the most expensive component of the SCADA/EMS.

The distribution functions are superimposed on the basic SCADA functions on the left side of the figure, beginning with the SCADA/distribution automation system and progressing to the distribution management system functions. The systems become more complex and expensive as one scans the figure from top to bottom (i.e., the basic SCADA system is the simplest and least expensive, the SCADA/AGC is more involved and a little more expensive, and the SCADA/EMS is much more complex and expensive). The same can be said for distribution. The SCADA/DA system is more complex and costly than the basic SCADA system. The SCADA/DMS system is far more complicated and costly [1, p. 7].

## 1.7 Advantages of SCADA Systems in Power Systems

The advantages of automating a system are numerous, and power systems are no exception. Some benefits are as follows:



**Fig. 1.15:** Use of SCADA in power systems

- Increases reliability through automation: The system can be run with fewer severe contingencies, and downtimes are resolved quickly.
- Alarms and system-wide monitoring enable operators to quickly spot and address problems.
- Faster decision-making, as a wealth of information about system conditions is made available to the operator to assist the operator in making accurate and appropriate decisions.
- Automation protects workers by enabling problem areas to be detected and addressed automatically.
- Operators can use powerful trending capabilities to detect future problems, provide better routine maintenance of equipment and spot areas for improvement.
- Sequence of events (SOE) recording: Crucial events in the system are time stamped for post-event analysis.
- Historians provide the ability to view data in various ways to improve efficiency.
- Human influence and errors are reduced because the values are accessed automatically, and meter reading and related errors are avoided.
- System operation can be optimized by running optimization algorithms and selecting appropriate performance parameters.
- Improved active and reactive power management because values are accurately captured in the automation system and appropriate action can be taken.
- Proper load balancing on the system, avoiding unnecessary overloading of equipment, and ensuring components have a longer service life.
- Power quality (PQ) monitoring devices can be networked and centrally monitored to detect harmonics, voltage sags, swells, and unbalances.

# Chapter 2

## SCADA Implementation for Power System Laboratory

### 2.1 Introduction

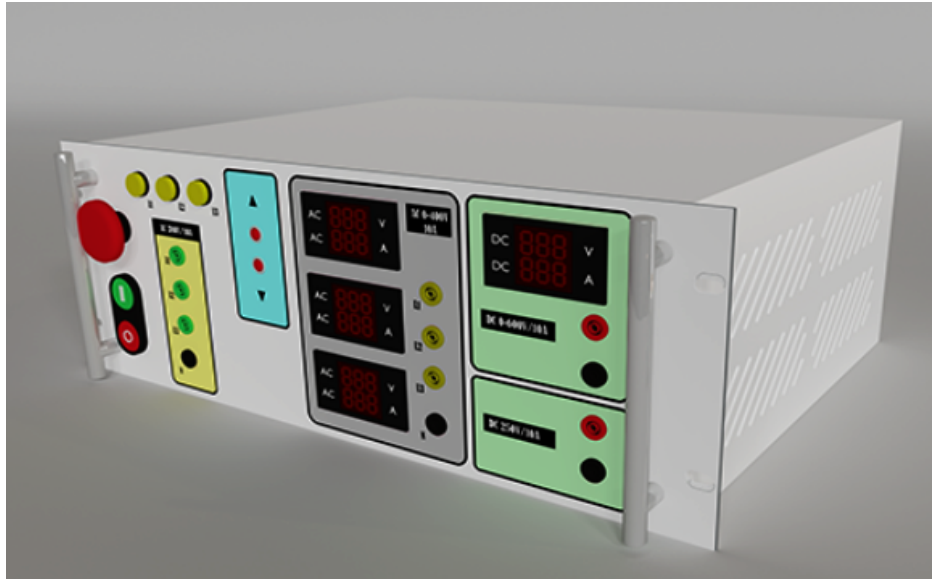
SCADA systems are used in a wide range of automation applications around the world, including the gas and petroleum industries, power automation, building automation, and small manufacturing unit automation. Although SCADA systems are widely used in industry, they are proprietary to each company, so very few technical details are available to students and researchers. The establishment of the SCADA power system laboratory is extremely important because it provides research facilities in the form of hardware and software for adaptive and intelligent control of integrated power systems.

The SCADA laboratory should include all the components mentioned in the previous chapter, as well as live monitoring and control. The laboratory was designed to give students and practicing engineers hands-on experience with SCADA systems and their applications to the management, supervision, and control of an automated system, with a focus on an electric power system.

### 2.2 Hardware Components

The SCADA laboratory that we are working on is made to simulate real world situations on power transmission when the monitoring of the power at the sending end and the receiving end is important for detecting power quality, power losses, system efficiency and the economical dispatch, we have implemented several modules that allow the students to understand the working principle of transmission lines and power transmission in general and the SCADA system is built to give them hands-on experience with supervisory control and data gathering. The main components implemented are:

### 2.2.1 Adjustable Three-phase Power Supply



**Fig. 2.1:** ELE-PAC3 : Adjustable three-phase power supply

*ELE-RAC3* is three-phase variable power-supply to provide power to devices connected to laboratory and test panels. For AC power the module can give a fixed 380V with maximum of 10A, or you can get an adjustable value from the range of 0 to 400V with maximum of 10A current, the same idea goes with DC power with ranges of 0 to 250V and 0.5A output current.

Figure 2.1 shows the 4U 19" power supply module which contains 4 mm safety sockets (L1, L2, L3, N, PE, L- and L+) in the foreground, with built-in 3 Digital Voltmeter 0...450V, 3 Digital Ammeter 0...10A and a 3 Phase control lights indicators.

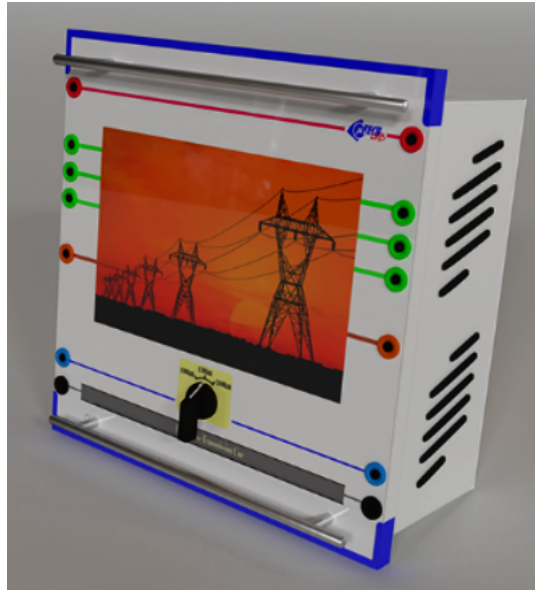
In terms of protection the power supply has 3 thermo-magnetic device circuit breakers, circuit breakers, Motor protection switch adjustable from 1.6...6A and Under voltage trip.

#### Application

Power supplies are necessary in almost all laboratory experiments, such as:

- Electrical power distribution and wiring.
- Electric Machines and drive.
- Power electronics.
- Instrumentation and programmable devices.
- Renewable Energy.
- Instrumentation and protection.

### 2.2.2 Air Power Line



**Fig. 2.2:** ELE-APL : Air Power Line Module

*ELE-APL* is a three-phase line simulation of a high voltage overhead power line with different lengths for measuring various operational states (open-line, matched, short-circuit). Scale factor 1/1000 for current and voltage. Automatic modification of wire length as soon as the corresponding overlay is in place.

The module has a rotary switch with three states of transmission line length: 100Km, 150Km and 300Km. The impedance of the transmission line is related to the distance, and it is going to be changed as presented in the table 2.1, the modules contain several Inputs/outputs 4 mm safety sockets.

**Table 2.1:** Transmission Line module impedance values

Distance	Resistance	Inductance	Capacitance phase to phase	Capacitance line to ground
100 Km	2.4 $\Omega$	77 mH	2 x 100 nF	2 x 0.33 $\mu$ F
150 Km	3.6 $\Omega$	115 mH	2 x 150 nF	2 x 0.55 $\mu$ F
300 Km	7.2 $\Omega$	230 mH	2 x 300 nF	2 x 1.1 $\mu$ F

The maximum power consumption is rated at 1kW with rated voltage 400V and 2A current with 50Hz frequency.

#### Applications

Transmission lines are the core of the power transmission between the generating site to the electrical substations. Therefore, this module can be used in many cases such as:

- Electrical power distribution
- Electric Machines and drive.
- Power electronics
- Renewable Energies

- Instrumentation
- Protection

### 2.2.3 Three-phase Resistive Load



**Fig. 2.3:** ELE-RES : Three-phase resistive load module

*ELE-RES* Resistive Load consists of a module housing nine wire-wound power resistors arranged in three identical banks. Each bank consists of three resistors connected in parallel that can be switched on or off with toggle switches to obtain various resistance values. This allows the total (necessary) resistance of each bank to be increased or decreased by steps. Six 4 mm safety jacks on the module front panel provide access to each resistor bank. The three resistor banks can be connected separately for operation in three-phase circuits. Also, the three resistor banks can be connected together for operation in single-phase circuits.

The Resistive Load is commonly used in conjunction with other basic load modules, like the Inductive Load and the Capacitive Load to experiment with the effects of different types of load on a circuit.

The three banks of resistors allow setting three-phase loads in 8 independent steps each or a single load of 22 different values

#### **Specs:**

- Three banks of manually switchable resistive loads
- Resistance Values (Each Bank): 1100/2200/4400  $\Omega$
- Nominal Voltage: 230 V AC/DC
- Resistance Accuracy:  $\pm 5\%$
- Load at Nominal Voltage (Each Bank): 84W
- Designed for A4 mounting frames

### 2.2.4 Three-phase Capacitive Load



**Fig. 2.4:** ELE-CAP : Three phase capacitive load

The capacitive load is an A4 module that provides a universal capacitive electrical load for a wide range of experiments. Each bank consists of three capacitors connected in parallel that can be switched on or off with toggle switches to obtain various capacitance values. This allows the total (equivalent) capacitance of each bank to be increased or decreased by steps. The module is suitable for both three-phase and single-phase experiments.

The Capacitive Load is commonly used in conjunction with other basic load modules, like the Inductive Load and the Resistive Load to experiment with the effects of different types of load on a circuit.

The three banks of capacitors allow setting three-phase loads in 8 independent steps each or a single load of 22 different values.

#### Specs:

- Three banks of manually switchable capacitive loads
- Capacitance Values (Each Bank): 3.3/6.6/13.2  $\mu\text{F}$
- Nominal Voltage: 230 V/50 Hz
- Maximum Voltage: 440 V
- Capacitance Value Accuracy:  $\pm 5$
- Load at Nominal Voltage Reactive Power: 1000VA
- Designed for A4 mounting frames

### 2.2.5 Three-phase Inductive Load



**Fig. 2.5:** ELE-IND : Three-phase inductive load

*ELE-IND* Inductive Load is a module that provides a universal inductive electrical load for a wide range of experiments. Each bank consists of three inductors connected in parallel that can be switched on or off with toggle switches to obtain various inductance values. This allows the total (equivalent) inductance of each bank to be increased or decreased by steps. The module is suitable for both three-phase and single-phase experiments.

The Inductive Load is commonly used in conjunction with other basic load modules, like the Resistive Load and the Capacitive Load to experiment with the effects of different types of load on a circuit.

The three banks of inductors allow setting three-phase loads in 8 independent steps each or a single load of 22 different values

#### **Specs:**

- Three banks of manually switchable inductive loads
- Inductance per Bank: 0.5H/1H/2H (8 pos)
- Nominal Voltage: 230 V/50Hz
- Accuracy of Inductance Value:  $\pm 5\%$
- Load at Nominal Voltage (Each Bank) Reactive Power: 1000VA

### 2.2.6 Digital Power Analyzer

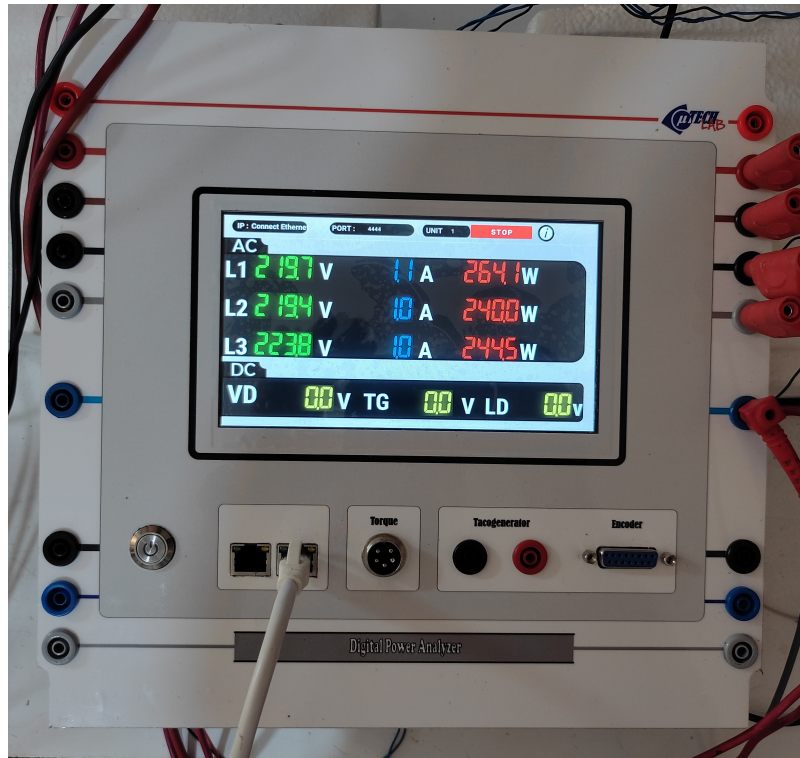


Fig. 2.6: ELE-IND : Three-phase inductive load

The MES-PAN station is a three-phase energy measurement, analysis, and data acquisition platform. This versatile peripheral device is used for the measurement, observation, analysis of electrical and mechanical parameters within electrical energy systems and power electronics circuits.

This measurement and data acquisition station is the keystone of the electrical energy technology-training program and enables deep learning on a wide range of topics related to electrical energy.

The energy measurement unit is considered an Intelligent Electronic Device, it has a 16 bit ADC that is able to read analog measurements and display voltages, currents, power and phase values. This device can be connected to the LAN using the Ethernet port, the user then can monitor and analyze the data in real time.

The integration of a large format touch screen (203 mm) allows the digital display of the values of currents, voltages, Active/Reactive power, power factor, energies in DC and AC and single-phase and three-phase. Thus, the voltage / current curves (Multichannel oscilloscope function) and the harmonic analysis of the Voltages/Currents and the THD.

#### Specs:

- Computerized instrumentation
- Integrated 8-inch touch display (203mm).
- RS232 / USB, Ethernet or WIFI connection for PC or Modem.
- Measurement of single-phase or three-phase alternating voltages up to 400V eff.
- Measurement of single-phase or three-phase alternating currents up to 10A eff (120A eff optional).

- Measurement of a direct voltage up to 600V.
- 20A direct current measurement.
- Measurement of the rotational speed of holding machines equipped with an encoder.
- Display of instantaneous values of electrical quantities (currents, voltages, active power, reactive power, power factor, etc.)
- Internal or external storage of measurement data.
- Box adaptable to the LabVOLT or DINA4 cabinet.
- 220V power supply

## 2.3 Communication Channel

The in-plant equipment are typically interconnected via a Local Area Network (LAN), using Ethernet. The illustration of the SCADA hierarchy discussed in the first chapter includes modems to a remote site, allowing remotely located controllers to operate over the same high-speed communication network.

Speaking of local area networks, or LANs, they are smaller networks, usually within an office base. LAN is used to connect computing resources, typically inside one building. The computing resources are our IEDs and the local server computer, and the router. Connections between the devices are physical, computers are connected to a switch with Ethernet cables. Each device has a unique IP address.

The routers will act as a Dynamic Host Configuration Server, or DHCP server. This means that the router will give addresses to workstations automatically. We have to make sure that the address pool has enough addresses for all the IEDs. We made certain that there are enough addresses outside the range for any hosts that need static addresses.

### 2.3.1 Modbus TCP:

For the Software side, we have implemented ModbusTCP protocol using “*pyModbusTCP*” python library, which gives us the access to modbus/TCP servers (IEDs) through the ModbusClient object.

The source code is available at the following Github repository <https://github.com/sourceperl/pyModbusTCP.git>.

#### 2.3.1.1 Implementation of ModbusClient

Initiate module from constructor (raise ValueError if host/port error):

```
from pyModbusTCP.client import ModbusClient

try:
    c = ModbusClient(host='localhost', port=502)
except ValueError:
    print("Error with host or port params")
```

**Listing 2.1:** Importing ModbusClient class and initiating client object

*ModbusClient* class takes as arguments, the host IP address (IED IP), the port number, the other parameters are optional. And it returns Object *ModbusClient*.

```
def __init__(self, host='localhost', port=502, unit_id=1, timeout=30.0,
            debug=False, auto_open=True, auto_close=False):
```

**Listing 2.2:** The parameters of *ModbusClient* object

The *ModbusClient* can perform the basic Modbus requests functions listed in the table 2.2.

```
def get_measurements(self):
    """Main polling function, this will handle the modbus request"""
    c = ModbusClient(self.host, self.port, self.unit_id, auto_open=True,
                    auto_close=True)
    # read 12 holding registers starting from address 0
    self.data = c.read_holding_registers(0, 12)
    return self.data
```

**Listing 2.3:** The method used for polling the measurements from the IEDs

**Table 2.2:** Available Modbus requests functions

Domain	Function name	Function code	ModbusClient function
Bit	Read Discrete Inputs	2	<a href="#">read_discrete_inputs()</a>
	Read Coils	1	<a href="#">read_coils()</a>
	Write Single Coil	5	<a href="#">write_single_coil()</a>
	Write Multiple Coils	15	<a href="#">write_multiple_coils()</a>
Register	Read Input Registers	4	<a href="#">read_input_registers()</a>
	Read Holding Registers	3	<a href="#">read_holding_registers()</a>
	Write Single Register	6	<a href="#">write_single_register()</a>
	Write Multiple Registers	16	<a href="#">write_multiple_registers()</a>

More details on the code working principle will be explained in the next section and the SCADA software code in the appendix.

## 2.4 SCADA Software

The SCADA software was created with Python, in-built modules, and some additional libraries. The Python language was chosen over other high-level languages such as C, C++, or Java to capitalize on the fact that Python is widely used in the scientific community for a wide range of applications. Furthermore, many of these applications are being developed in research laboratories and universities, with a large number of them being completely open source. The greatest advantage of an open-source application is that it can be developed and modified concurrently by several independent groups, resulting in better software and greater community engagement. Bug reporting and subsequent bug fixes result in cleaner code. [19]

The SCADA software used in the lab is a custom Python desktop application based on PyQt graphical user interface library. It operates as both as supervisory SCADA server (also known as a master

terminal unit) and HMI software. The software supports multiple operating systems and can be used on any compatible Windows, MacOS or Linux machine. Sqlite3 database engine is used in the background as a historian, which records all IED events, sensor data, and alarm information when the application runtime is live. The communication between the master station and the IEDs is managed by ModbusTCP protocol as discussed in the previous section.

PyQt is a Python binding for Qt, which is a set of C++ libraries and development tools that include platform-independent abstractions for Graphical User Interfaces (GUI), as well as networking, threads, regular expressions, SQL databases, SVG, OpenGL, XML, and many other powerful features. Developed by RiverBank Computing Ltd, PyQt is available in two editions:

- PyQt4: an edition that is built against Qt 4.x and 5.x
- PyQt5: an edition that is only built against Qt 5.x which is the one that we have used.[20]

Qt classes employ a signal/slot mechanism for communicating between objects that is type safe but loosely coupled making it easy to create reusable software components.

Qt also includes Qt Designer, a graphical user interface designer. PyQt is able to generate Python code from Qt Designer. It is also possible to add new GUI controls written in Python to Qt Designer.

Python is a simple but powerful object-oriented language. Its simplicity makes it easy to learn, but its power means that large and complex applications can be created. Its interpreted nature means that Python programmers are very productive because there is no edit/compile/link/run development cycle.

Much of Python's power comes from its comprehensive set of extension modules providing a wide variety of functions including HTTP servers, XML parsers, database access, data compression tools and, of course, graphical user interfaces. Extension modules are usually implemented in either Python, C or C++.

The application has a list of pre-built examples ready to be used or re-configured, also the user can create his own designs, save them or export them as portable document format (PDF).

### 2.4.1 The Startup Page

When the user launches the SCADA software, a startup window will appear, as shown in figure 2.7; this page only provides a list of pre-built projects by which the user can choose and tinker. These examples are organized according to the topics that students will study during their semesters, beginning with the most basic and simple and progressing to the most complex and advanced. The project names are interactive, which means that the user can double-click them to open a new window containing the actual project design. The startup page also includes another tab labeled "DB viewer," that will be explained in a deeper level later.

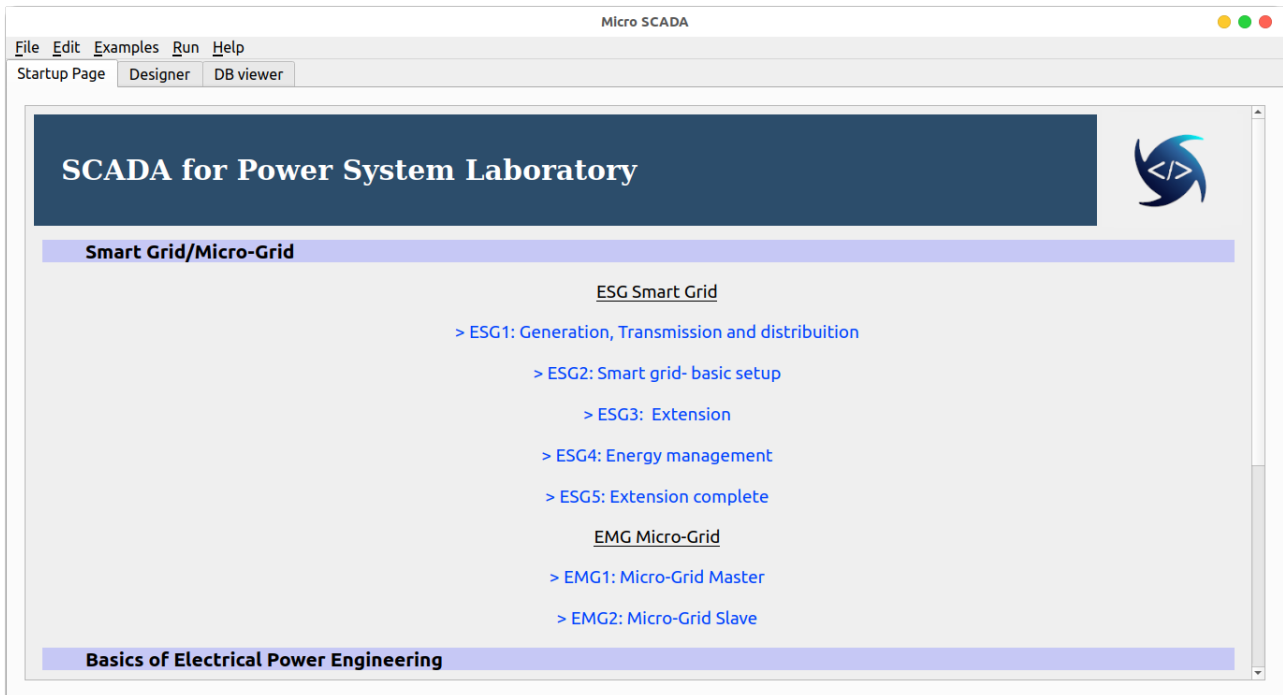


Fig. 2.7: SCADA software startup page

## 2.4.2 The Designer Window

After the user opens a new project, the following window will pop up, basically it is composed of two main elements: the modules list and the designer main view area.

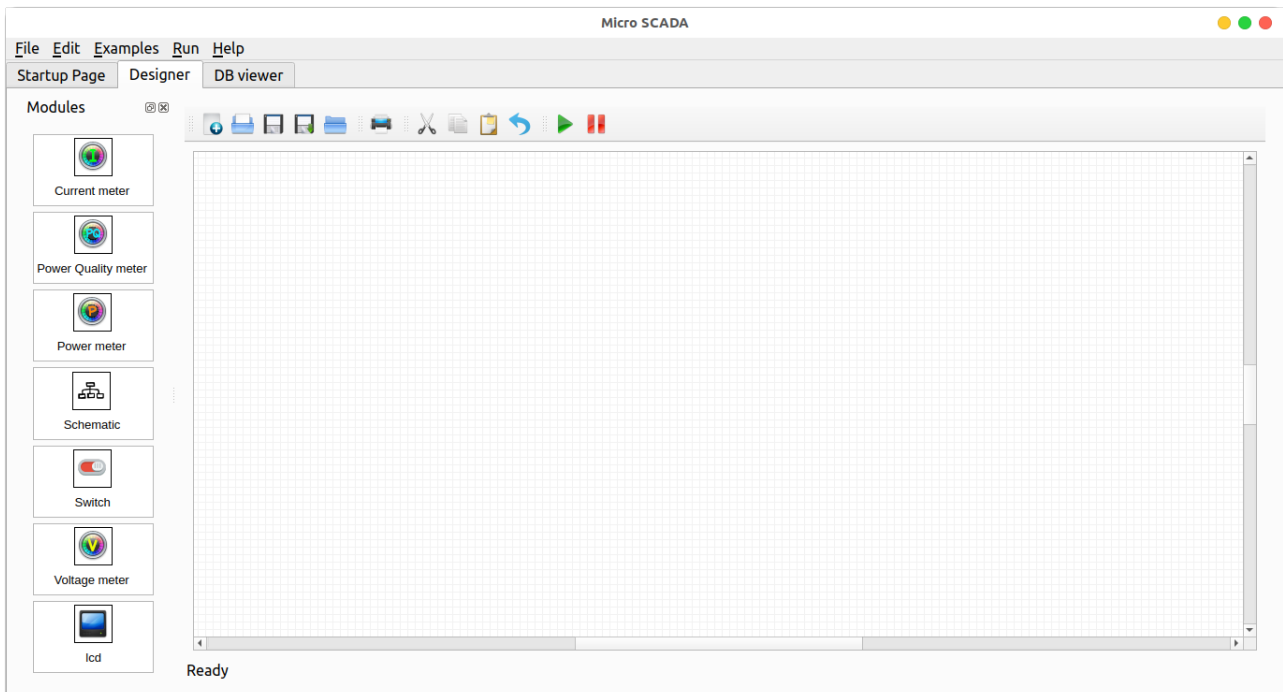
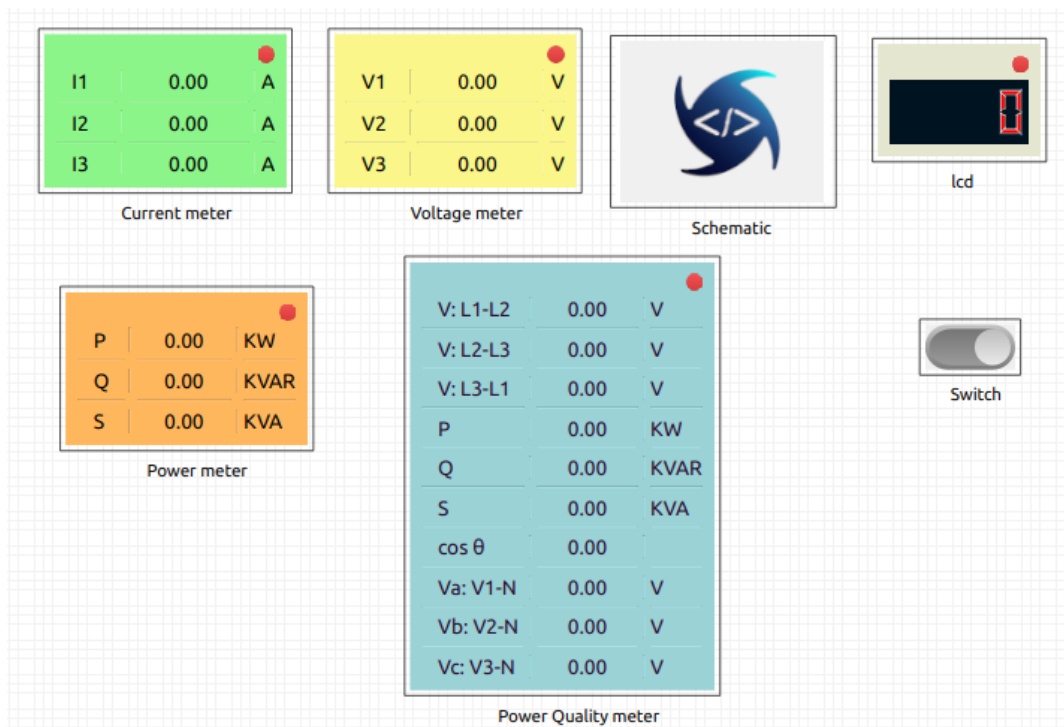


Fig. 2.8: SCADA software Designer Tab

A list of all the available modules that the operator can use is presented on the left side of the screen, those items accept the drag event which means the user can drag them using the mouse and drop them

in the main designer view. They are built as *QGraphics* Widgets contained in *QGraphicsItems* Which the designer can process and use. Currently, the available items are:

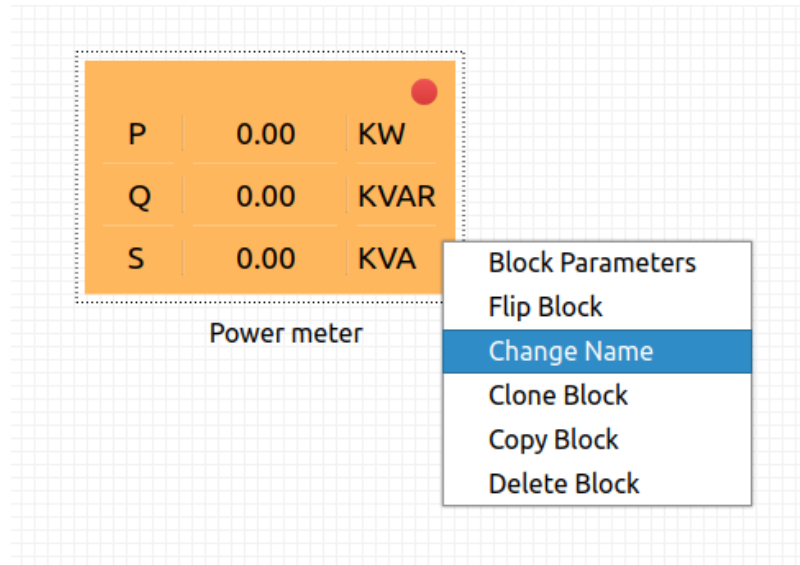
- Current meter: a module that can display three-phase currents.
- Voltage meter: a module that can display three-phase voltage.
- Schematic: a module that gives the user the ability to load his schematic diagram as *png* or *jpeg* image, with specific width and height.
- LCD module: a general purpose module that displays a single digital value.
- Power meter: a module that can display real power as well as reactive and phase shift between current and voltage.
- Power Quality: this module shows all the precedent data with the power and phases.
- Switch: this module can be linked with a physical switch to be controlled on and off.



**Fig. 2.9:** Available Modules list in the SCADA Software

The main designer view is the central white area with a grid background; it accepts drop events, which means that if the user drags a proper module from the modules list and releases it on the designer view, the software will load the relevant module to the designer. The choice here is done based on the module type, since each type causes the SCADA software to call the appropriate functions to load the required objects to the view.

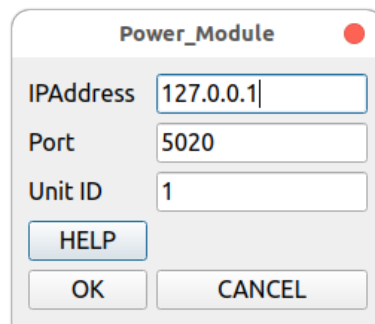
The designer view has more features and accepts more events, the user can rearrange the modules using drag and drop, Also he can select, copy, clone or delete modules. By default, after the module is loaded a label name is presented at the bottom of that block, this label can be changed by choosing “Change Name” from the list that will appear on the item right-click event.



**Fig. 2.10:** The list of the different operations that can be performed to a block

### 2.4.3 The Block Parameters

Each module has a set of parameters that the user can change by double-clicking on the item block or choose block parameters from the list that appears in figure 2.10, which will bring up the parameters' dialog as it is represented in figure 2.11. The modules that are directly connected to the field should have an IP address, a port, and a unique Unit ID. The new parameters will load after the user submitted a correct input and pressed the OK button.



**Fig. 2.11:** The parameters of power meter block

On the right top of each block there is a small status indicator; the red circle indicates that this module is offline, the yellow circle indicates that there is a problem with the network connection (wrong parameters, or the server is down...etc), and the green circle denotes that the module is online, This is clearly shown in figure 2.12

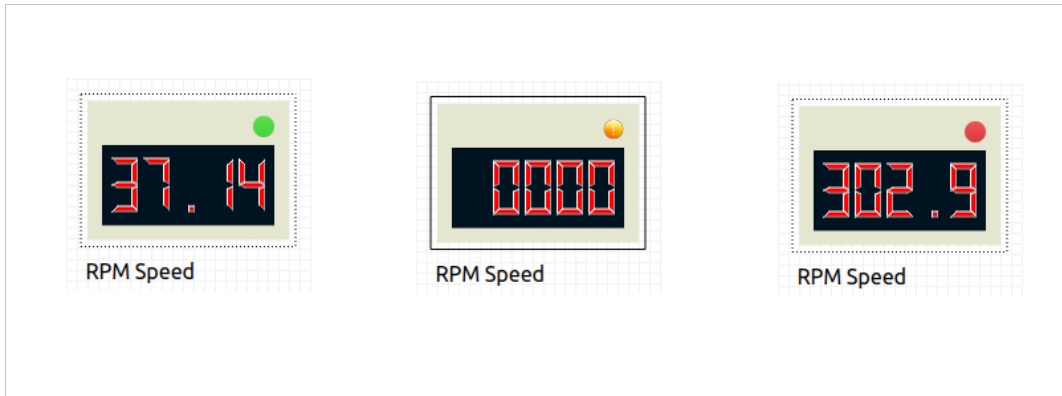


Fig. 2.12: The three different connection state for a block

### 2.4.4 Running a Project

When the design is ready and all the modules are loaded, the user can run the project by clicking on the run button from the toolbar, if all the modules are configured correctly and the slave units (IEDs) are connected to the same local area network (LAN), the app will start an infinite loop of polling data from the IEDs, and it could be stopped manually from the Stop button. The master station computer that hosts the software is considered as a client and each IED is a separate server.

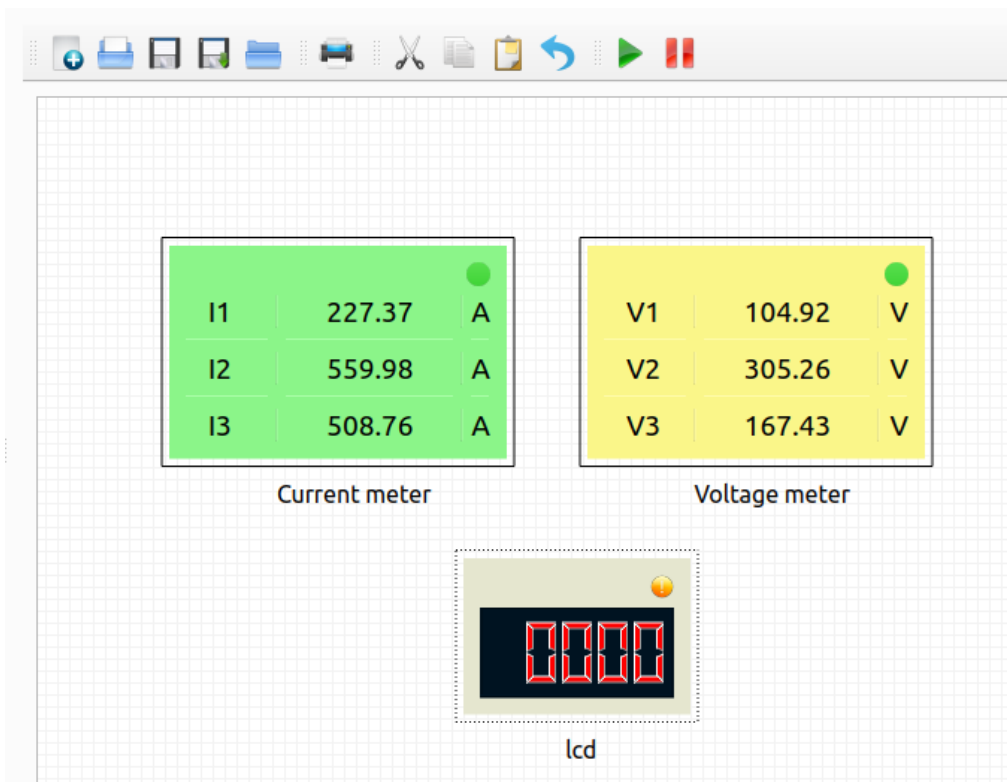


Fig. 2.13: Example of a running project

In the case of measurement modules which perform the monitoring, the client initiates a read registers or read coil request to the server, along with the target registers address. The server will respond with the content of the required registers in the form of an array, the returned data is processed and presented on the screen. However, the switch modules can send write requests to the server every time

the user switches them ON or OFF, and this is how the control process is done. The displayed data will be saved in the database so that the user can access it for historian reports and further analysis.

## 2.4.5 Diagram File Format

The top toolbar offers more features like opening a new design, save the current project or save it as specific file format, by default; the design will be saved in diagram format (*dgm*), which is just a form of XML.

XML stands for Extensible Markup Language which is a markup language and file format that can be used to store, transmit, and reconstruct arbitrary data. It specifies a set of rules for encoding documents in a human-readable and machine-readable format.[21]

The logical structure of an XML file requires that all data in the file be encapsulated within an XML element called the root element or document element. This element identifies the type of data contained in the file, The root element contains other elements that define the different parts of the XML document; In our case the root element contains some metadata elements and many <block> which, in turn, consist of other elements as depicted below:

```

<root>
  <uscada>
    <uscada_Version>0.9</uscada_Version>
  </uscada>
  <Date>
    <SavingDate>18.06.2022 - 21:18:56</SavingDate>
  </Date>
  <Panel_design>
    <Template>sim.tmf</Template>
    <Ts>0.01</Ts>
    <AddObj></AddObj>
    <ParScript></ParScript>
    <Tf>10</Tf>
    <Priority></Priority>
  </Panel_design>
  <block>
    <name>lcd</name>
    <type>lcd</type>
    <icon>lcd</icon>
    <params>LCD_Module|IPAddress: 127.0.0.1|Port: 5020|Unit ID: 1 </params>
    <help>This block implements an LCD module.</help>
    <width>162</width>
    <height>115</height>
    <flip>0</flip>
    <posX>-240.0</posX>
    <posY>-80.0</posY>
  </block>
</root>

```

**Listing 2.4:** sample dgm file generated from the SCADA software

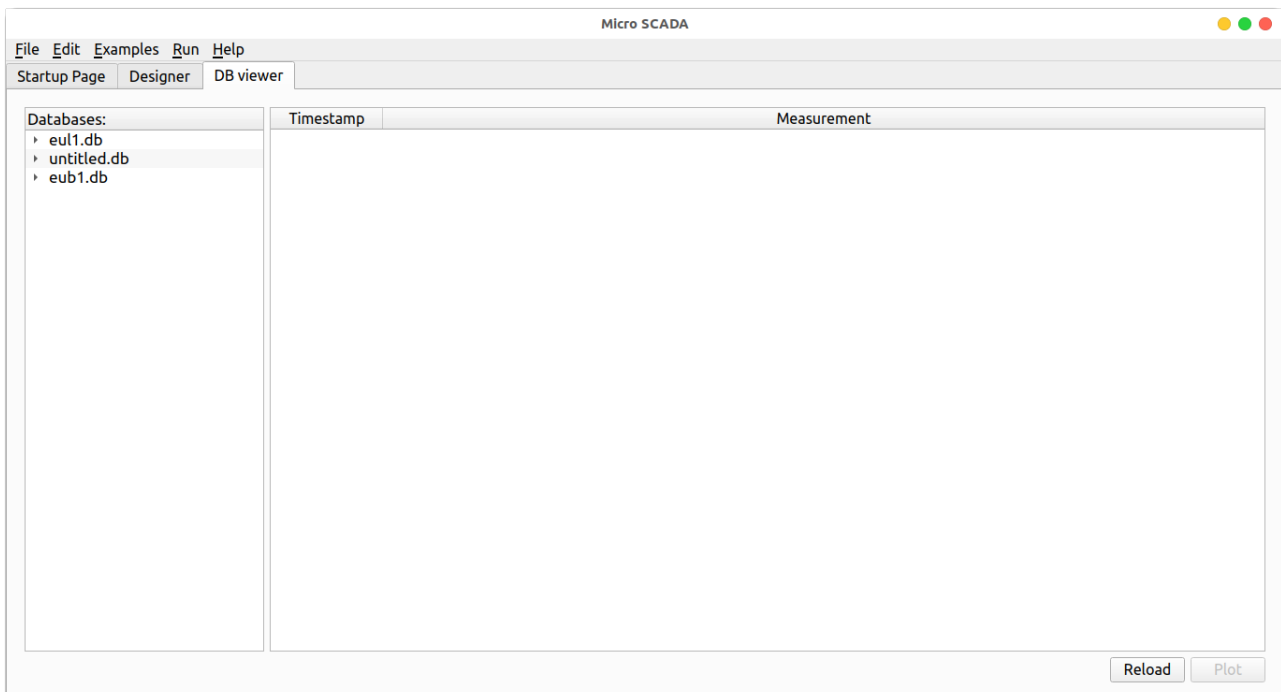
Each block element should have the following list of tags:

- <name>: The name of the block, the blocks with the same name will be differentiated by a number (e.g: block0, block01 ...)
- <type>: The type of the block, we have current panel type, voltage panel type, LCD type ...
- <icon>: The icon name for the block, this name used by back end functions to show the right icon for each block.
- <params>: the list of parameters for the block, separated by 'l'.
- <width>, <height>: specifies the geometry of the block.
- <flip>: contains boolean value to indicate if this block is flipped or not.
- <posX>, <posY>: stores the position of the block in the scene so that the next time the program loads this block, it knows where it should be placed.

The SCADA designer will prompt the user with a dialog box asking if the user wants to save his work or not, every time he tries to close the software with an unsaved project. The designs also could be printed or exported to PDF using the print icon from the top toolbar.

## 2.4.6 The Database Viewer

The third tab in the main window contains the database viewer; which is a simple file navigator that shows the databases list found in the current directory on the left side of the screen on a tree view, the sub-items for each database are tables, the user can navigate and display the stored values by double-clicking on the table name from the list.



**Fig. 2.14:** SCADA database viewer tab

For each new project, the SCADA software will create a new database and assign a new table to each independent measurement unit. The content of the table will be displayed on the right side as shown in figures 2.14 and 2.15.

	id	I1	I2	I3	IP	Port	timestamp
1	1	533.05	238.63	577.98	127.0.0.1	5020	21:53:09 18-06-2022
2	2	78.67	492.16	169.49	127.0.0.1	5020	21:53:09 18-06-2022
3	3	78.67	492.16	169.49	127.0.0.1	5020	21:53:10 18-06-2022
4	4	90.83	446.39	84.66	127.0.0.1	5020	21:53:10 18-06-2022
5	5	90.83	446.39	84.66	127.0.0.1	5020	21:53:11 18-06-2022
6	6	139.35	356.59	37.5	127.0.0.1	5020	21:53:11 18-06-2022
7	7	139.35	356.59	37.5	127.0.0.1	5020	21:53:12 18-06-2022
8	8	423.43	568.6	275.43	127.0.0.1	5020	21:53:13 18-06-2022
9	9	423.43	568.6	275.43	127.0.0.1	5020	21:53:13 18-06-2022
10	10	130.25	100.28	589.76	127.0.0.1	5020	21:53:14 18-06-2022
11	11	130.25	100.28	589.76	127.0.0.1	5020	21:53:14 18-06-2022
12	12	467.41	654.11	329.19	127.0.0.1	5020	21:53:15 18-06-2022
13	13	467.41	654.11	329.19	127.0.0.1	5020	21:53:15 18-06-2022
14	14	301.95	154.8	359.17	127.0.0.1	5020	21:53:16 18-06-2022
15	15	301.95	154.8	359.17	127.0.0.1	5020	21:53:16 18-06-2022
16	16	319.75	113.69	418.67	127.0.0.1	5020	21:53:17 18-06-2022
17	17	319.75	113.69	418.67	127.0.0.1	5020	21:53:17 18-06-2022
18	18	437.22	441.73	0.76	127.0.0.1	5020	21:53:18 18-06-2022
19	19	437.22	441.73	0.76	127.0.0.1	5020	21:53:18 18-06-2022

Fig. 2.15: Sample of the stored data into the database

The tables have measurement columns that show the stored metric data for the specific module along with IP address column, Port column and timestamp column. The order of the table can be changed by pressing on the headers of the table.

The software uses *SQLite3* library to manage databases; which is an DB-API to the C library that provides a lightweight disk-based database that does not require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. [22]

The user can access the database, since it is stored in the local drive, and he has the permission to read, write all the stored values like any *Sqlite3* database, either from terminal window as shown in figure 2.16 or using the “*DB Browser for sqlite3*” as shown in figure 2.17, which is the official GUI tool.

```

sqlite3
[~/scritper~/.../uscada/src]>>sqlite3 eub1.db
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
sqlite> .tables
Current Voltage lcd
sqlite> .schema Current
CREATE TABLE Current (id integer, `I1` real, `I2` real, `I3` real, `IP` text, `Port` integer
, `timestamp` timestamp, PRIMARY KEY(id));
sqlite> SELECT I2 FROM Current WHERE I1 > 100 limit 5;
238.63
356.59
356.59
568.6
568.6
sqlite> SELECT IP FROM Current LIMIT 1;
127.0.0.1
sqlite> SELECT IP, Port FROM Current LIMIT 1;
127.0.0.1|5020
sqlite>
    
```

Fig. 2.16: Accessing the database using SQLite3 through terminal

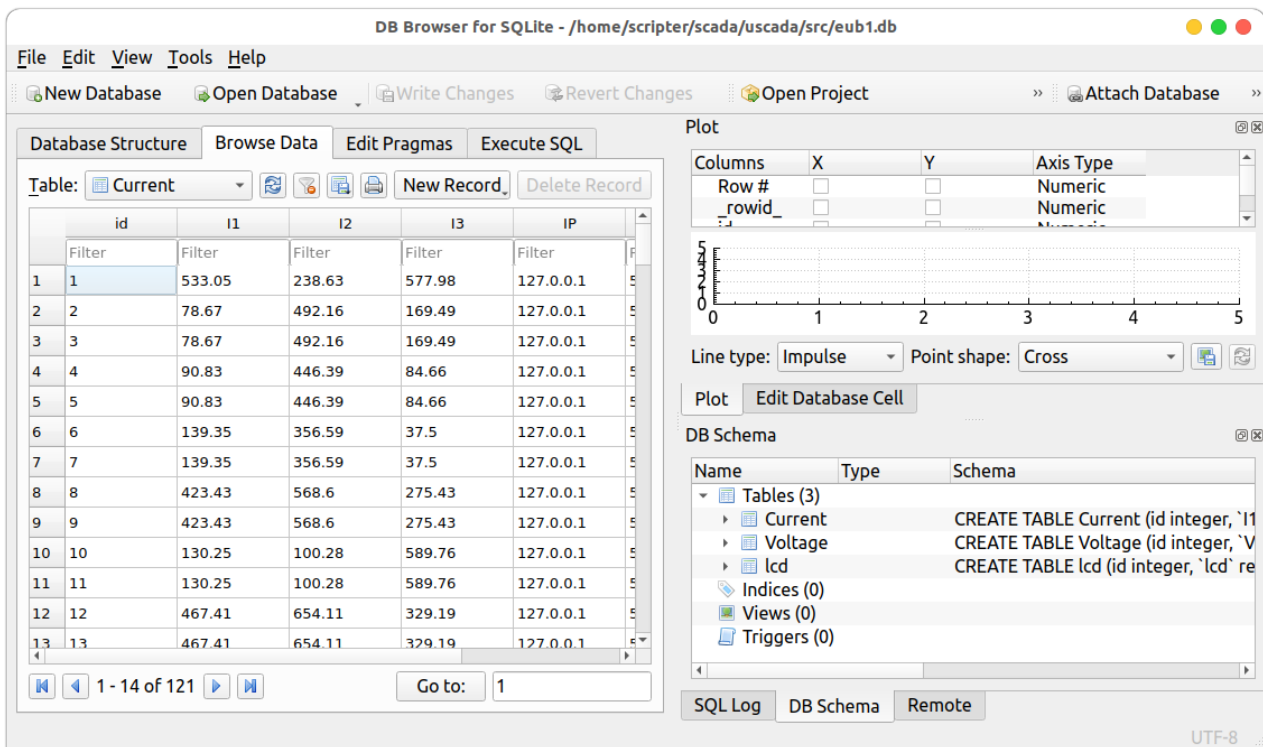


Fig. 2.17: Editing and viewing databases using DB Browser for SQLite3

### 2.4.7 Plotting

Another cool feature the SCADA software has is the ability to plot the stored data, the plot button is located on the bottom right of the database viewer tab. The button will be enabled when the table is filled with measurement values.

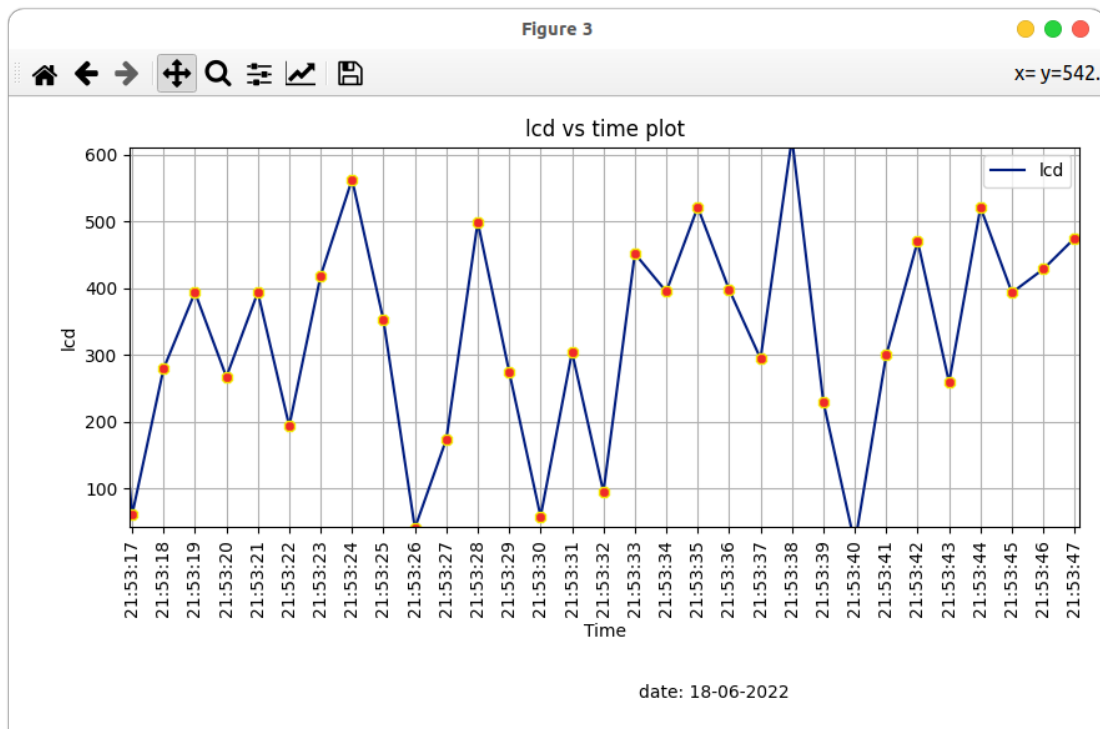
The plotting is done via “Matplotlib”, which is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is a cross-platform library for creating 2D plots from array

data. Matplotlib is written in Python and uses NumPy, Python's numerical mathematics extension. It provides an object-oriented API for embedding plots in Python GUI toolkits such as *PyQt*, *WxPython* or *Tkinter*. It is also compatible with Python and IPython shells, Jupyter notebooks, and web application servers.

Matplotlib has a procedural interface called *Pylab* that is intended to look like MATLAB, a proprietary programming language developed by MathWorks. Matplotlib and NumPy can be thought of as the open source equivalents of MATLAB.

Matplotlib was created in 2003 by John D. Hunter. The most recent stable version is 2.2.0, which was released in January 2018. [23]

The generated graphs will have the metric data on the y-axis vs their timestamp on the x-axis and this can be zoomed in or out easily, the Figure window has many options to change the title, the axis labels, the legend or to manipulate the type and style of the plotted curves. The various plots we can utilize using *Pyplot* Matplotlib's function are Line Plot, Histogram, Scatter, 3D Plot, Image, Contour, and Polar. These results could be saved as Portable Network Graphics Image format (*PNG*), which then could be used in making reports and further analysis.



**Fig. 2.18:** Example of the generated graphs using Matplotlib

# Chapter 3

## Results and Discussion

### 3.1 Introduction

The real or active power (watts, kW, MW), the reactive power (voltage-ampere-reactive, VAR, KVAR, MVAR), the voltage (V, kV, M), and the power angle (degrees) are the four electrical parameters that are constantly measured and monitored in electrical systems. With these readings, a typical operation necessitated a number of switching operations and device readjustment at the power grid, data collection from grid equipment and the load side, consumed energy, and energy forecasting for both typical and unusual events. [24]

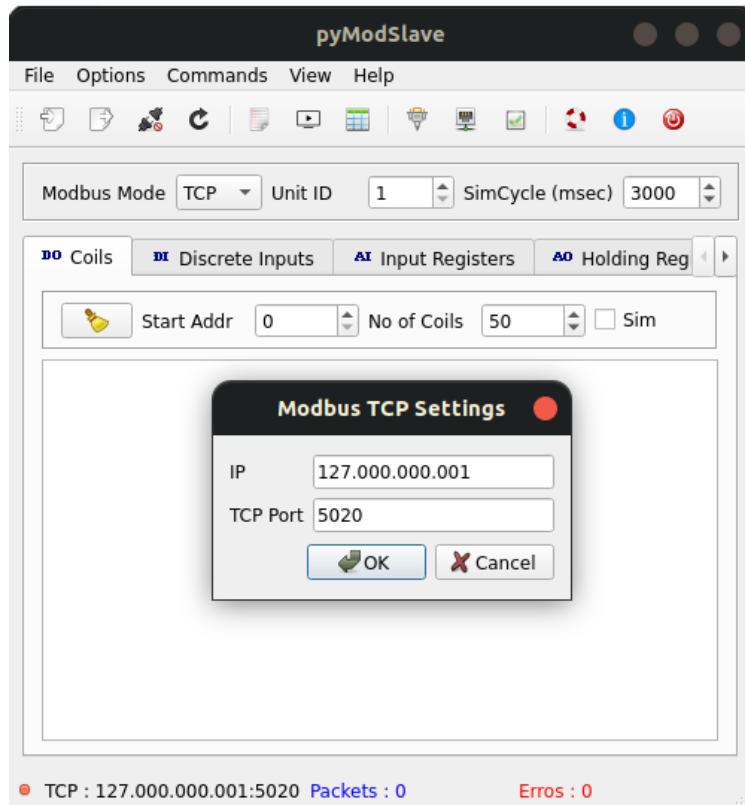
In This chapter we will present the overall picture of the working system while tracking the electrical network parameters and perform some simulation and real tests, then we can compare our results and note the changes and errors. The end goal is to illustrate the accuracy of the system.

### 3.2 Communication Channel Test

The SCADA system depends mostly on the reliability of the communication channel. So first we are going to expose our system to a simulation test where we are going emulate IEDs with virtual units that can acts the same as real slaves to our SCADA software master.

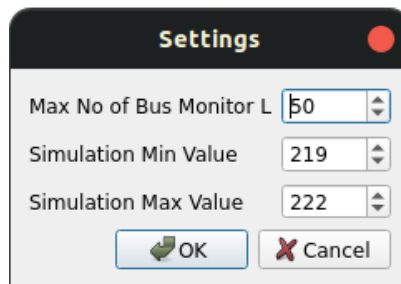
The main function of an IED is to get measured data and respond to requests through the Modbus protocol, for this purpose we can use *pyModSlave*; which is a piece of open source software that can be used to create a set of servers on the network with defined or random data.

*PyModSlave* has a very simple interface, to initiate a server we have to provide a valid IP address, port and unit ID, then we can specify the type and range of the data that we want the server to provide. Figure 3.1 illustrates the configuration of *pyModSlave* server runs on “127.0.0.1:5020” with unit ID 1, we will conduct our test on the local host with multiple live servers, and we will assign a new IP for each server.



**Fig. 3.1:** Configuring IP Address and Port for *PyModSlave*

The unit that we are going to simulate should give us random values in the range of 220 for the first unit which could be considered as generator RMS voltages, we could do this by setting maximum and minimum values just as shown in figure 3.2.



**Fig. 3.2:** Setting the output values range for Modbus server

With the same method we initiated other four servers to offer currents, phases and power readings.

In the SCADA software we started new project, and since we just want to test the communication we loaded only the relevant modules with this test, we have provided each module with the correct IP and port number (default unit ID is 1), then we launched the test as shown in figure 3.3

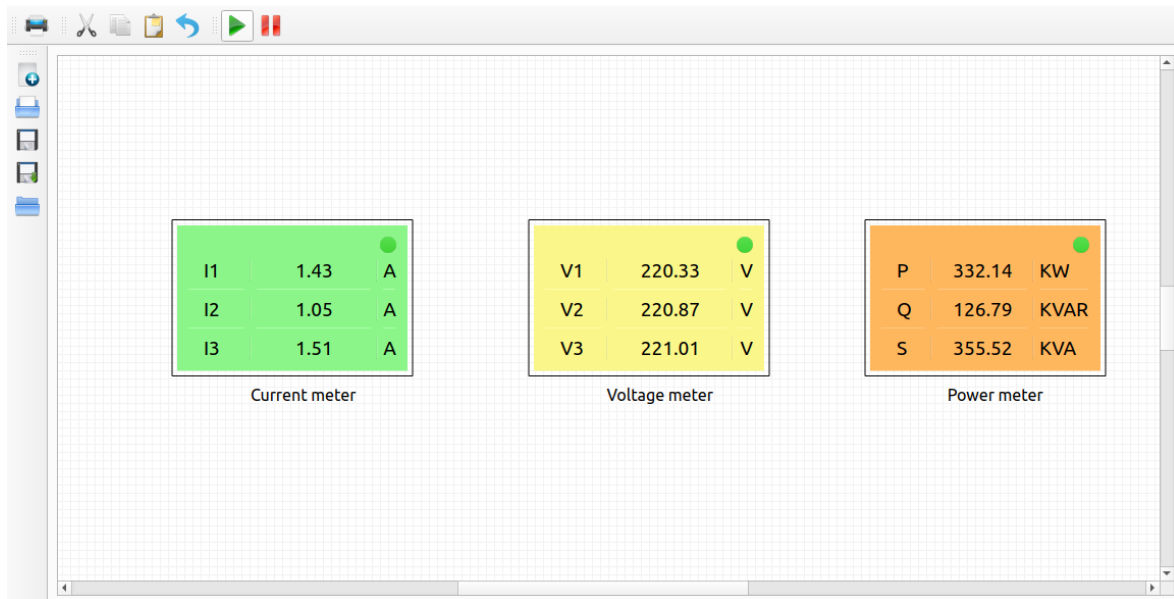


Fig. 3.3: Dashboard for simulation test

PyModSlave has a very nice feature which is “Bus Monitor” that allow as to see the requests and response in real time, export these results to a text file, figure 3.4 shows a communication log during our test.

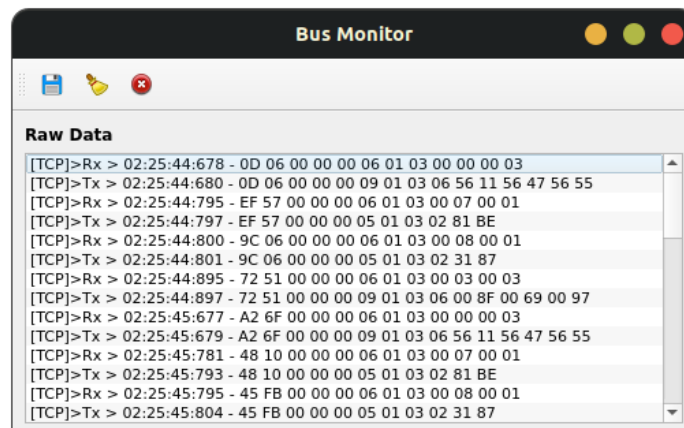


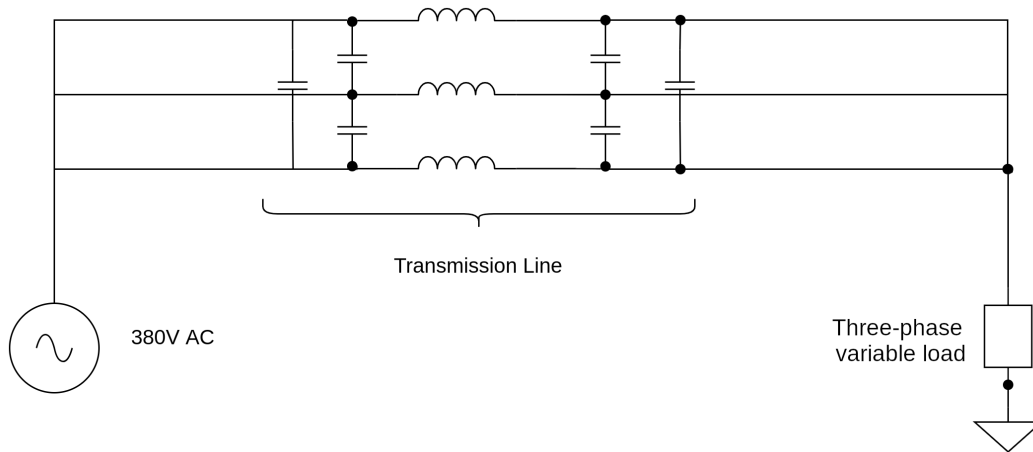
Fig. 3.4: Bus Monitor output

From figure 3.4 we can get the average Modbus reading time for a single request, we can notice that the server takes 2 ms to respond to each received request which is considered to be very fast for monitoring and events tracking. Usually in the industrial field machine states and voltages levels are tracked for every one second.

### 3.3 Transmission Line Tests

The purpose of these tests is to determine the effectiveness of the SCADA software with the Air line transmission laboratory equipment, the unit description and specification has been mentioned in the second chapter, we are going to change the type of load each time record, compare the output values in the SCADA dashboard and the SCADA graphs.

Figure 3.5 represents the circuit schematic that we are going to study. The diagram contains a three-phase variable AC generator connected to the transmission line then it reaches an adjustable resistive load.



**Fig. 3.5:** Three-phase load signal line diagram

### 3.3.1 Resistive Load

We will plug the receiving end of the transmission line to the *Digital power analyzer* so that we get the measurements of the second stage of the system, this module should be connected to the LAN within the same master station network, the IP address, port number and unit ID should be shown at the top right of module screen if successful connection established.

The hardware connections are presented in figure 3.6.

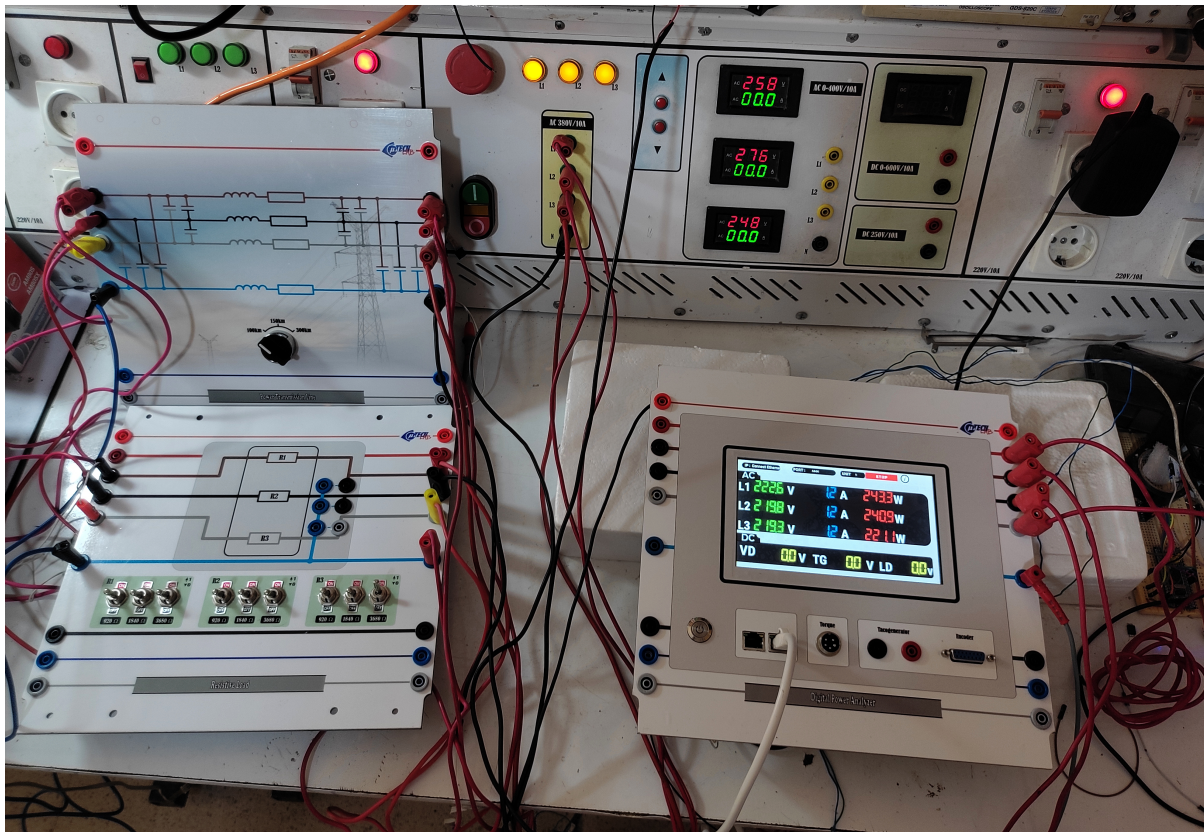


Fig. 3.6: TL with resistive load test

In the master station machine, where the SCADA software is hosted we will start a new project for this test and create our monitoring dashboard.

The single line diagram of the circuit could be loaded by adding a *schematic* module to the designer. Then we have to provide the PNG image path for the diagram and the required width and height, as illustrated in the figure 3.7.

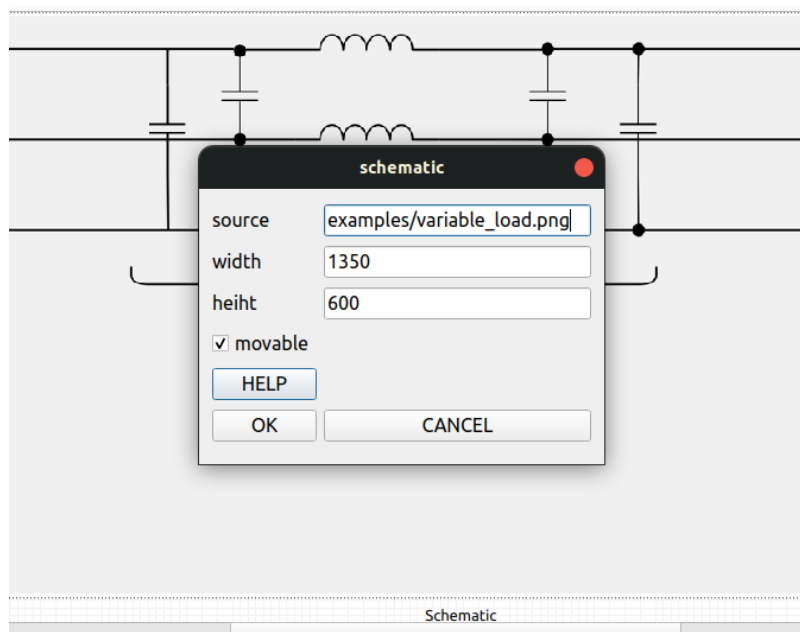
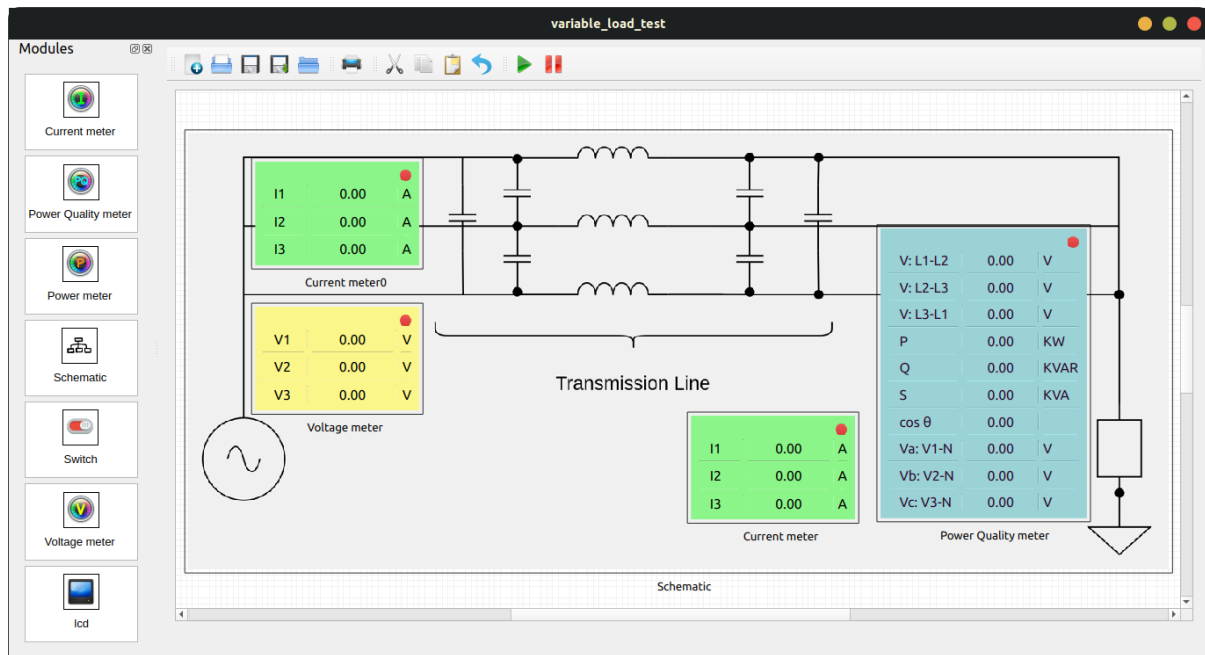


Fig. 3.7: Adding the circuit schematic to the SCADA designer

The targeted data for this experiment are the voltages, currents and power measurements, So that we have loaded the relevant modules and placed them on top of the single line diagram schematic, the modules parameters (IP, port, ID) must be the same values shown by the power analyzer, in order to get proper data transmission.

After checking the connections, the network configurations, and the protocol parameters, we can start polling the measurement data from the IED and track them from the SCADA software in real time.

The SCADA software is configured to send a read request every one second to the IED, the returned data for 1 minute test are stored in a local database, those values are accessible through *DB Viewer* tab.



**Fig. 3.8:** Monitoring the system in SCADA dashboard

The toggle switches on the load allow us to get different resistance values for each new combination, this should affect the voltage drop, currents and power factor.

After we turned the power supply ON, we get the following results from the power analyzer:



Fig. 3.9: Power analyzer reading

Then, we have used digital multimeter to check the accuracy of the measurement, on the first phase :

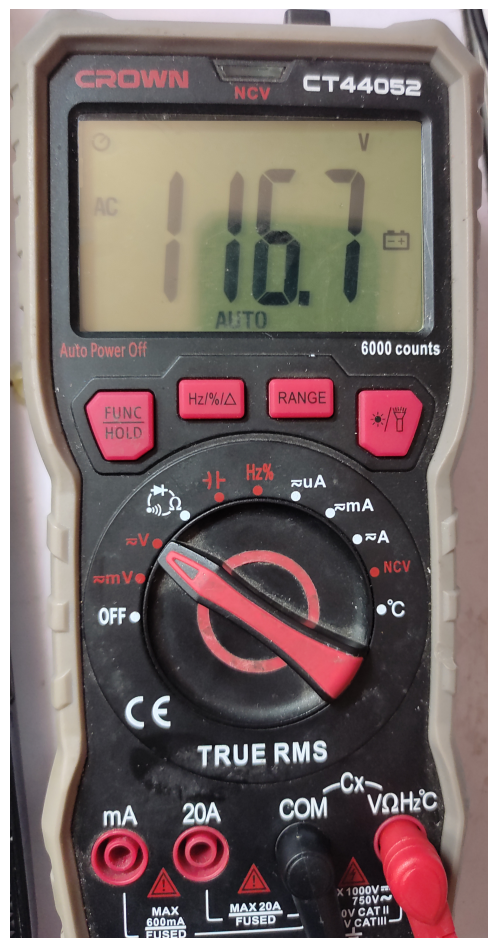


Fig. 3.10: Checking the reading with the multimeter

We used *EasdyEDA* online circuit simulator to simulate each phase, as reference to compare the

experimental readings with the simulated readings, the full records are presented in table 3.1.

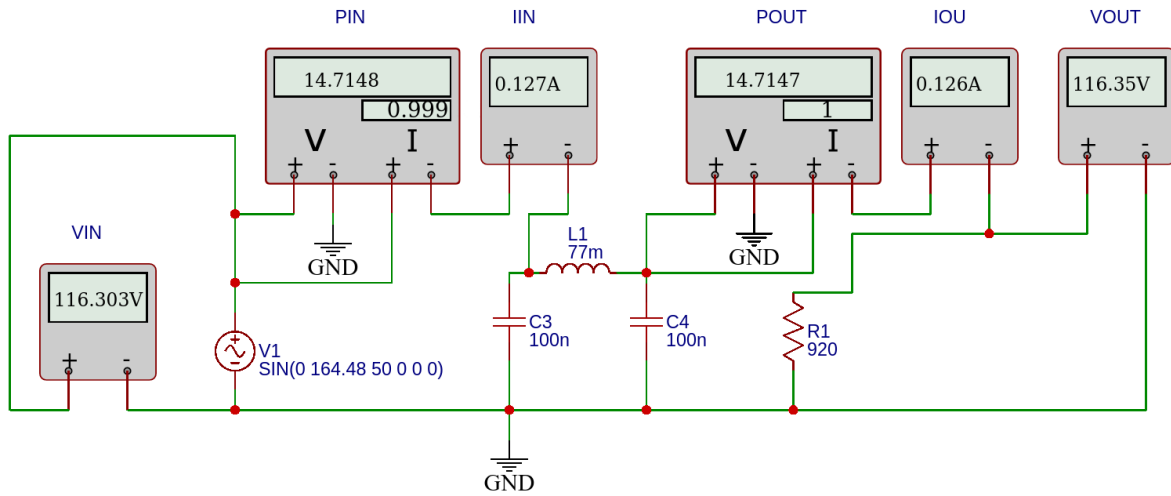


Fig. 3.11: Simulation result of signal phase

Table 3.1: Measured and calculated values for resistive load test with  $V_s = 120V$

Test	Experimental			Simulation		
	V(V)	I(A)	P(W)	V(V)	I(A)	P(W)
525.71	84.824	0.162	13.70	84.825	0.161	13.68
613.33	84.848	0.139	11.76	84.849	0.138	11.73
736	84.869	0.116	9.813	84.869	0.115	9.786
920	84.885	0.092	7.849	84.885	0.092	7.832
1226.67	84.898	0.069	5.883	84.898	0.069	5.875
1840	84.908	0.046	3.922	84.908	0.046	3.918
3680	84.914	0.023	1.956	84.914	0.023	1.959

We can notice that the differences between the theoretical and experimental values is a very small and can be ignored which means that our system is working just as expected and can be used to get very accurate results.

### 3.3.2 Inductive Load

In this test the only difference we are going to make is changing the load type to a series connected RL load, keeping the same connections and wiring used in the experiment before, figure 3.12 shows the overall system.

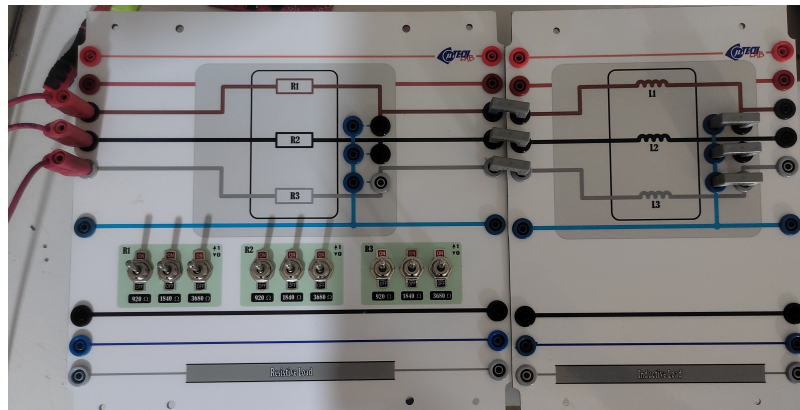


Fig. 3.12: Variable capacitive load connections

In the SCADA software we have to change the schematic diagram, and keep the same blocks used in the experiment before since the target data are the same.

Then we recorded the measurements using the SCADA software and from the database viewer we plotted the voltages curve to just keep track of the current fluctuations. The results for this test are represented using the SCADA software plotting function in figure 3.13.

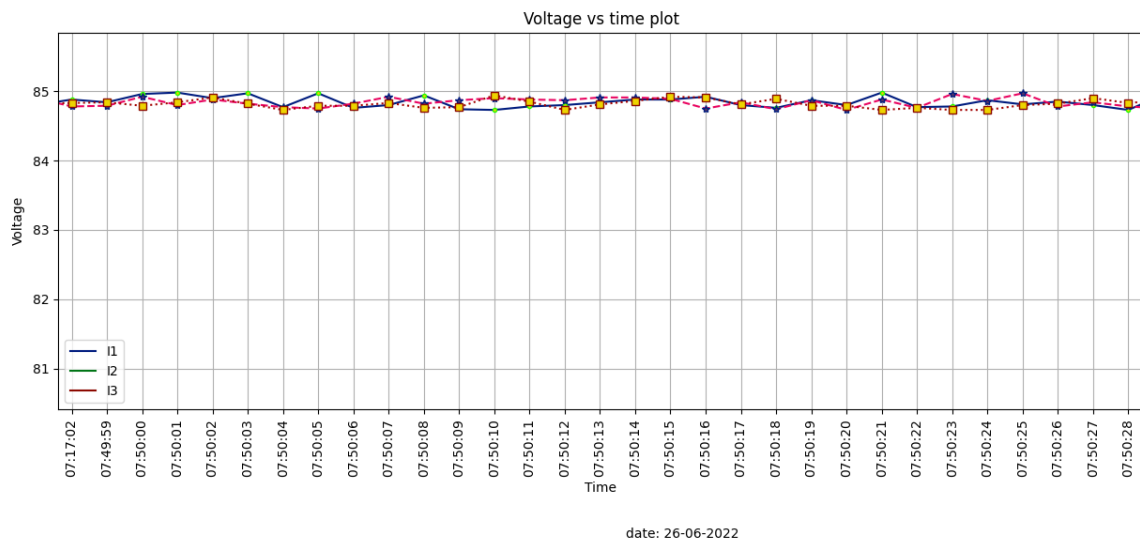


Fig. 3.13: Graph that shows the change in the voltages in an inductive load

The previous tests clearly show the ability of the designed SCADA system to perform accurate monitoring and easy data gathering and manipulations, with a very small error. The GUI of the application is quite simple, clean and easy to use.

The resulting charts and graphs provide an excellent visual representation of the monitored data. They can be exported and used to generate reports, analyze power consumption, and help determine the most cost-effective system dispatch.

The SQLite3 database management system has been proven to be a great option for small applications and local applications.

Although the software is still in development, several unexpected errors such as connection loss problems, bad inputs, and unrecognized events and gestures have been handled in the back end using the “*try except*” statements. As a result, the software has been proven to work flawlessly with far fewer bugs and problems.

Using LAN Ethernet communication to exchange data between the different components of SCADA system makes it more reliable, consistent and extremely fast.

# Conclusion

The SCADA Laboratory was designed and built to help Electrical Engineering students and professionals understand real-time monitoring and control of systems. The Laboratory is the first of its kind, and students will gain hands-on experience with the Electric Power System's online monitoring and control. The laboratory was conceived and designed following extensive industry and utility research and studies.

The laboratory contains SCADA system components, a master station, an IED, various communication channels, and a variety of field equipment. The data is being collected with time stamps, which will allow the sequence of events to be monitored.

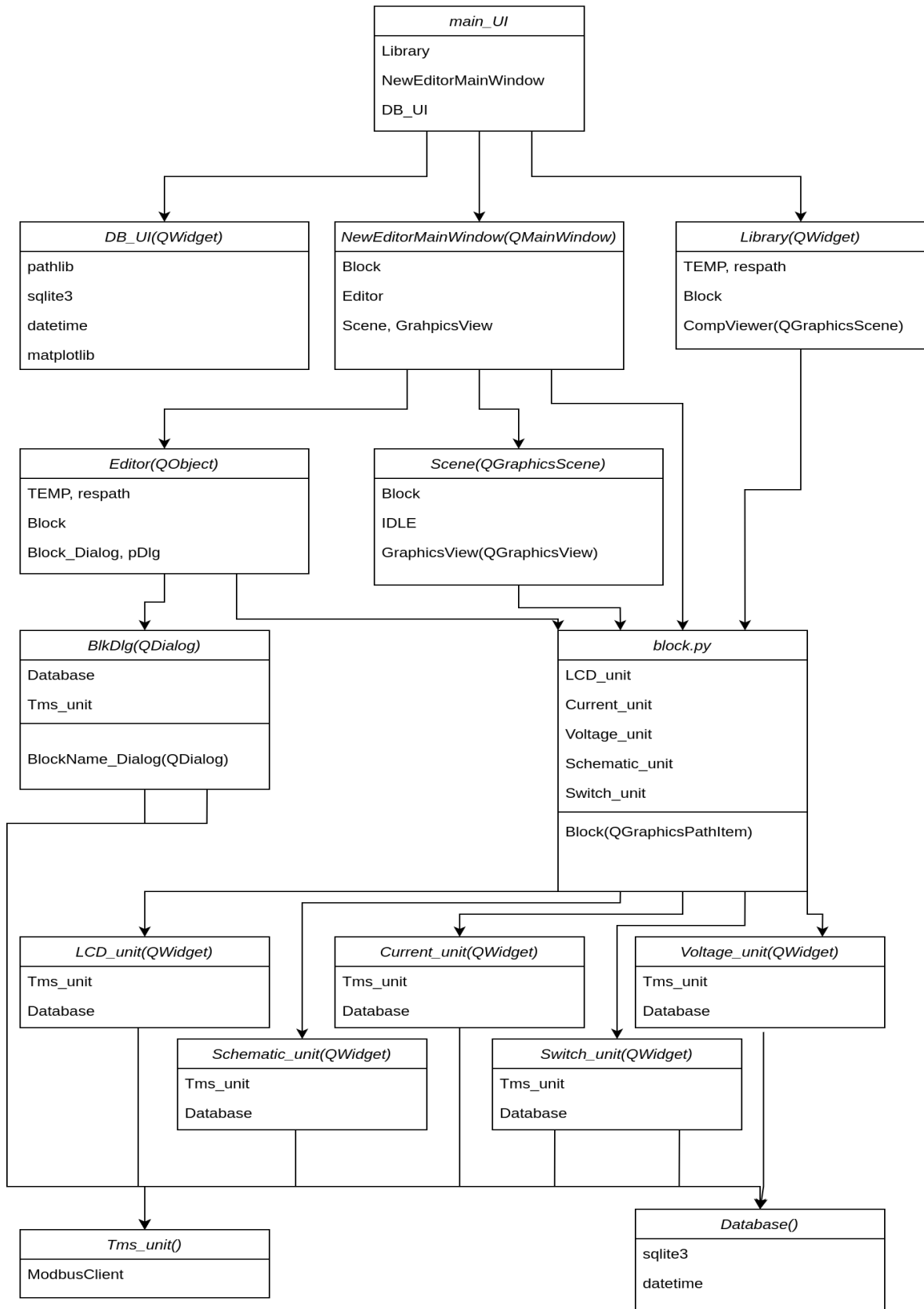
Overall, this laboratory will give undergraduate and postgraduate students a better understanding of industrial SCADA systems, which are currently proprietary to each company. Furthermore, it automates the process of report generation, plotting graphs and events recording.

## Future Works

Since the SCADA software is written in Python that gives it the possibility to improve and progress in the future, Actually there is a lot of functions and features that could be added, things like:

- Alarms and notifications: adding alarms to the system is very simple, since alarms are just going to be scheduled events that wait for certain conditions, we could think of many cases where alarms are needed, like monitoring the voltage drop to not exceed a specific value or may be short circuits and faults alarms...etc.
- Power quality detection and classification: the most recent techniques used for power quality detection are using the power of machine learning and artificial intelligence, Python has the best tool for such a topic.
- Fault detection: detecting short circuits and over voltage faults is not an easy task, it depends more on the hardware and relaying system, but if that part was handled correctly the software side could be adjusted to fit to the system.
- Web based SCADA: SCADA systems that are connected to the web have more control range and capabilities, Experiments conducted in a real laboratory are undoubtedly the essential learning experience. However, remote laboratory facilities allow the students to access the laboratory infrastructure at nonworking hours. From the point of view of the teaching institution that offered services, this pleases the students very much.

# Appendix



**Fig. 3.14:** UML diagram for SCADA software

# References

- [1] M. S. Thomas and J. D. McDonald, *Power system SCADA and smart grids*. CRC press, 2017.
- [2] S. A. Boyer, *SCADA: supervisory control and data acquisition*. International Society of Automation, 2009.
- [3] S. D. Anton, D. Fraunholz, C. Lipps, F. Pohl, M. Zimmermann, and H. D. Schotten, “Two decades of scada exploitation: A brief history,” in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, IEEE, 2017, pp. 98–104.
- [4] M. V. P. MCA, M. Vedavathi Katneni, M. V. K. Manda, and M. MCA, “Super scada systems: A prototype for next gen scada system,” *IAETSD JOURNAL FOR ADVANCED RESEARCH IN APPLIED SCIENCES*, 2018.
- [5] *A brief history of the scada system*, <https://bit.ly/3x51G4N>, Accessed: 2022-06-05.
- [6] J. McDonald, “Substation automation basics-the next generation,” *Electric Energy T&D Magazine*, p. 24, 2007.
- [7] *Intelligent electronic devices (ied) in scada*, <https://bit.ly/38u1ekk>, Accessed: 2022-06-05.
- [8] T. Sezi and B. K. Duncan, “New intelligent electronic devices change the structure of power distribution systems,” in *Conference Record of the 1999 IEEE Industry Applications Conference. Thirty-Forth IAS Annual Meeting (Cat. No. 99CH36370)*, IEEE, vol. 2, 1999, pp. 944–952.
- [9] P. Prakash, “Data concentrators: The core of energy and data management,” *White paper, Texas Instruments*, 2013.
- [10] *Mbdc-200 modbus data concentrator and gateway*, DSMBDC200-1101, Accessed: 2022-06-05, proconX Pty Ltd, 2011.
- [11] *Merging units*, <https://bit.ly/3xh9Btn>, Accessed: 2022-06-05.
- [12] *Communication network*, <https://bit.ly/3zzyzFE>, Accessed: 2022-06-05.
- [13] D. Dolezilek and E. Schweitzer, “Sel communications and integration white paper,” *Schweitzer Engineering Laboratories*, 2000.
- [14] D.-j. Kang and R. J. Robles, “Compartmentalization of protocols in scada communication,” *International Journal of Advanced Science and Technology*, vol. 8, pp. 27–36, 2009.
- [15] C. Strauss, *Practical electrical network automation and communication systems*. Newnes, 2003.
- [16] M. Organization, *MODBUS Messaging on TCP/IP Implementation Guide: V1. 0b*. Modbus Organization, 2006.
- [17] *White paper: Introduction to modbus tcp/ip - acromag*, BusWorks® 900EN Series, Accessed: 2022-06-05, Acromag, Inc, 2005.
- [18] T. Agarwal and Z. Fatima, “Master station architecture of a scada system,” *IETE Journal of Education*, vol. 43, no. 3, pp. 121–126, 2002.
- [19] S. V. Iyer, *Simulating Nonlinear Circuits with Python Power Electronics: An Open-Source Simulator, Based on PythonTM*. Springer, 2018.
- [20] *Python and pyqt: Building a gui desktop calculator*, <https://bit.ly/3z1pC2v>, Accessed: 2022-06-05.

- 
- [21] *Extensible markup language (xml) 1.0 (fifth edition)*, <https://bit.ly/3tiWANq>, Accessed: 2022-06-05.
- [22] *Sqlite3 — db-api 2.0 interface for sqlite*, <https://bit.ly/3tHLIsI>, Accessed: 2022-06-05.
- [23] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [24] H. Rafique, “5.15 energy management in network systems,” in *Comprehensive Energy Systems*, I. Dincer, Ed., Oxford: Elsevier, 2018, pp. 581–647, ISBN: 978-0-12-814925-6. DOI: <https://doi.org/10.1016/B978-0-12-809597-3.00531-9>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128095973005319>.