electronics

*Article*

# Smart IoT SCADA System for Hybrid Power Monitoring in Remote Natural Gas Pipeline Control Stations

**Muhammad Waqas and Mohsin Jamil ***

Department of Electrical & Computer Engineering, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, 240 Prince Phillip Drive, St. John's, NL A1B 3X5, Canada; mwaqas@mun.ca
* Correspondence: mjamil@mun.ca; Tel.: +1-709-351-1521

**Abstract:** A pipeline network is the most efficient and rapid way to transmit natural gas from source to destination. The smooth operation of natural gas pipeline control stations depends on electrical equipment such as data loggers, control systems, surveillance, and communication devices. Besides having a reliable and consistent power source, such control stations must also have cost-effective and intelligent monitoring and control systems. Distributed processes are monitored and controlled using supervisory control and data acquisition (SCADA) technology. This paper presents an Internet of Things (IoT)-based, open-source SCADA architecture designed to monitor a Hybrid Power System (HPS) at a remote natural gas pipeline control station, addressing the limitations of existing proprietary and non-configurable SCADA architectures. The proposed system comprises voltage and current sensors acting as Field Instrumentation Devices for required data collection, an ESP32-WROOM-32E microcontroller that functions as the Remote Terminal Unit (RTU) for processing sensor data, a Blynk IoT-based cloud server functioning as the Master Terminal Unit (MTU) for historical data storage and human–machine interactions (HMI), and a GSM SIM800L module and a local WiFi router for data communication between the RTU and MTU. Considering the remote locations of such control stations and the potential lack of 3G, 4G, or Wi-Fi networks, two configurations that use the GSM SIM800L and a local Wi-Fi router are proposed for hardware integration. The proposed system exhibited a low power consumption of 3.9 W and incurred an overall cost of 40.1 CAD, making it an extremely cost-effective solution for remote natural gas pipeline control stations.

**Keywords:** Blynk; control station; hybrid power system; IoT; open source; pipeline; SCADA

check for updates

**Citation:** Waqas, M.; Jamil, M. Smart IoT SCADA System for Hybrid Power Monitoring in Remote Natural Gas Pipeline Control Stations. *Electronics* **2024**, *13*, 3235. https://doi.org/10.3390/electronics13163235

Academic Editor: Ahmed Abu-Siada

Received: 23 July 2024
Revised: 10 August 2024
Accepted: 11 August 2024
Published: 15 August 2024

## 1. Introduction

Global energy demand and consumption have increased in the current era, resulting in a widespread reliance on fossil fuels that are detrimental to humanity and the environment. This situation, combined with the growing and impending shortage of fossil fuels, has prompted the scientific community to seek alternative electricity generation methods through renewable energy sources [1]. Recently, there has been a substantial increase in the development and adoption of renewable energy sources (RESs). As energy specialists endeavor to utilize environmentally friendly and sustainable energy for society's benefit, renewable energy sources are progressively being incorporated into existing power networks. Notably, solar energy and wind energy have received significant attention [2,3].

Over time, Pakistan has increasingly depended on natural gas as its main energy source. Natural gas fulfills a substantial portion of Pakistan's energy needs, making it a major contributor to electric power generation. Because of the difficulties associated with storing natural gas, it must be quickly transmitted to its ultimate destination after being extracted from the gas wells. High-pressure natural gas pipelines are operated and maintained by several control stations. However, their dispersed strategic locations, particularly in remote areas, present operational hurdles due to challenges such as unreliable electricity supply and logistical constraints. Additionally, the efficient operation of these control stations relies on various electrically powered devices installed within them [4]. Therefore, an uninterrupted and reliable power supply and a monitoring system for such control stations have become necessary.

As renewable energy sources such as solar and wind are subject to intermittent conditions dictated by the environment, traditional energy generation methods are combined with renewable power sources to form a Hybrid Power System (HPS). Hybrid power systems typically require energy storage systems to maintain stability in system operations and power supply. These storage systems play a crucial role in mitigating fluctuations in energy output, ensuring frequency regulation and facilitating load balancing, among other essential roles. Hybrid power systems typically combine conventional energy sources such as fossil fuels with one or more renewable sources such as wind and solar, alongside energy storage systems, power electronic converters such as inverters, and additional power system devices such as communication systems. These systems are often deployed across expansive geographical regions, including different situations, such as offshore locations and wetlands. Given their distributed nature, interconnecting these systems for power generation and supply poses several challenges. These include addressing power quality, controlling frequency, managing voltage tolerances, synchronizing the grid, facilitating data exchange and communication between components, and ensuring safety and security protocols [5]. Hence, modern systems must incorporate advanced digital and high-tech equipment that enable bidirectional communication between the supply and demand sides. They offer features such as real-time monitoring, communication, and control between generation and demand, which lowers energy consumption, improves efficiency, and enhances system resilience. This results in secure, adaptable, and intelligent operation [6].

To address these challenges and maintain smooth power system operations, a range of sensors, microprocessors, microcontrollers, pumps, actuators, valves, and other devices are typically interconnected throughout the entire HPS. They gather crucial data such as voltage, current, temperature, pressure, etc., enabling live monitoring and remote coordinated control. Supervisory Control and Data Acquisition (SCADA) serves as the ideal solution for these tasks. SCADA systems are the foundation of industrial automation. The automation pyramid, or automation hierarchy, is a conceptual model that organizes industrial automation systems into different levels, from the physical devices on the plant floor to higher-level management systems. At the lower levels of the automation pyramid, SCADA is responsible for monitoring, controlling, and coordinating operations across all industrial devices. At the higher levels, SCADA serves as an interface for human supervision and Manufacturing Execution Systems (MES), facilitating decision making and the management of industrial processes [7].

SCADA is a widely recognized and proven technology used in various systems for remote monitoring, offering control and oversight capabilities for distant devices [8]. A SCADA system comprises of two parts: a hardware part for data collection, communication, operation, and control, and a software part for data storage, visualization, data processing, optimization, and alert management. A master terminal unit (MTU), remote terminal units (RTUs), and field instrumentation devices (FIDs) make up a SCADA system's hardware. An RTU is a microcontroller or programmable logic controller (PLC) that connects to FIDs such as sensors, transmitters, pumps, etc. RTUs and MTU exchange data over a communication network using a designated protocol. The MTU acts as the central unit of the SCADA system, processing data and displaying it on the human–machine interface (HMI) for the operator to easily monitor and operate. The HMI, which is a software component of SCADA, is a valuable feature that provides users or operators with an informative display for plant monitoring. Developers often utilize SCADA software that includes HMI design tools to build the SCADA user interface because of its convenience and rapid development capabilities [9].

Over the last three decades, SCADA technology has advanced to monitor and control distributed processes. Before SCADA, plant personnel used push buttons and selector switches for monitoring and controlling industrial processes, requiring many on-site workers. As industrial operations expanded and sites got increasingly remote, timers and relays were introduced to aid in supervision and control, reducing the need for on-site personnel. Although timers and relays offered a reasonable level of automation, significant resources were still needed for management. To address the demand for greater process automation and the limitations of earlier monitoring and control systems, the first generation of SCADA systems, known as Monolithic SCADA, was developed in the 1970s as standalone units. The second generation, called Distributed SCADA, emerged in the 1980s and 1990s due to the development of Local Area Networks (LAN) and HMI software, along with advancements and miniaturization in computer systems. However, most communications then were proprietary, restricting access to third-party suppliers of a particular SCADA system [10]. Wide-area network (WAN) connectivity increased as SCADA systems adopted open-source architectures and non-proprietary communication protocols in the 1990s and 2000s. This resulted in the third generation, known as Networked SCADA. However, rapid advancements in computer technologies soon made these systems outdated. SCADA manufacturers responded by developing the fourth generation, known as IoT SCADA, which integrates classic SCADA with cloud services to improve monitoring and control. This enables real-time access to plant information from anywhere, across multiple platforms. The IoT concept involves connecting physical objects with sensors, embedded electronics, software, and networking to allow devices and operators to exchange data over a shared platform or the internet [11,12].

There are two methods for transitioning to an IoT-based SCADA system. The first method includes designing a new SCADA system from scratch using IoT or cloud technology. The second approach focuses on transitioning the current SCADA system to the cloud or integrating it with IoT-compatible components [13]. Despite the IoT technology having been available for some time, experts are still unable to agree on a common IoT architecture. Various scholars have suggested distinct architectural designs, with the three-layer and five-layer models being the most common, as discussed extensively in [14]. Regardless of which architecture is implemented, reliable communication is crucial in IoT-based systems due to the widespread distribution of IoT devices. IoT technologies typically utilize the four layers of the basic TCP/IP model [15].

Research indicates that key design factors for IoT-based real-time applications such as SCADA include message encoding formats, communication protocols, and the chosen web-based or IoT platform. The TCP/IP model provides a framework for communication over the internet, with each layer performing specific tasks to ensure data is transmitted efficiently and accurately from one device to another. It comprises four layers, namely the Network Interface Layer, the Internet Layer, the Transport Layer, and the Application Layer. Consequently, selecting the appropriate protocol at each layer of the IoT protocol stack is crucial for a specific application. For instance, in the Application Layer, choices include MQTT, HTTP, and CoAP, while suitable protocols must also be chosen for the Internet, Transport, and Network Access Layer [16].

## 2. Literature Review

The site selection process is integral to designing a hybrid power system, playing a pivotal role in determining its performance and effectiveness. Key considerations include assessing the readiness of renewable resources, for instance, solar irradiation, wind speed, and hydro potential. Understanding the site-specific load profile is essential for appropriately sizing and configuring system components to meet energy demand. Environmental factors, including terrain, climate conditions, and regulatory requirements, also influence system design. Economic viability hinges on factors such as installation costs, potential energy savings, and payback periods, all of which are influenced by site selection. Ultimately, a well-chosen site maximizes energy generation potential while minimizing environmental impact and operational costs, laying a strong foundation for the hybrid power system's success [17].

Research communities worldwide have significantly developed and enhanced SCADA systems for various applications. Sufficient research has been done to tackle the high costs and compatibility challenges of commercial SCADA systems by creating a range of open-source SCADA solutions that offer varying functionalities and price points.

A comprehensive literature survey reveals that SCADA systems are widely employed across various applications and sectors. In the industrial sector, SCADA systems are extensively used for monitoring and controlling manufacturing processes, ensuring efficiency and safety. The adoption of SCADA in these diverse applications highlights its versatility and critical importance in ensuring operational excellence and safety. Table 1 provides a brief overview of how SCADA systems are utilized across different applications and sectors.

**Table 1.** Overview of SCADA system applications.

| Reference | Design Parameter |
|---|---|
| [5] | Design of low-cost SCADA system for distributed assets |
| [9] | Monitoring of plant leaf temperature and air and soil parameters |
| [18] | Monitoring and control of solar-powered water pumping system |
| [19] | Lightning damage detection on wind turbine blade |
| [20] | Remote control and monitoring of grid-connected inverters |
| [21] | Monitoring and control of reverse osmosis desalination system |
| [22] | Remote monitoring of PV system |
| [23] | Monitoring of HPS for a renewable energy laboratory |
| [24] | Secure cloud monitoring and control of high-power machines |
| [25] | A low-cost remote monitoring system for a PV system |
| [26] | Local data logging and monitoring of the PV system for an industrial plant |
| [27] | Monitoring and control of hybrid power systems |
| [28] | Remote control of motors and sensors deployed for oil and gas facilities |
| [29] | Monitoring of hybrid renewable systems in offshore and remote areas |
| [30] | Configuration and reconfiguration of an autonomous assembly system |
| [31] | Monitoring of power consumption using fog computing |
| [32] | Vulnerability analysis of SCADA systems and security measures for a standard wastewater treatment plant |
| [33] | A traffic control system |
| [34] | Detection of antenna and microphone problems in the telecommunication sector |
| [35] | Monitoring and control of renewable energy systems |

A substantial amount of literature has been reviewed throughout this study, with pertinent findings summarized in the preceding section of this article. It has been noted that IoT-based remote monitoring and control systems have targeted a range of sectors and applications. These include electric power plants [5,12], plant monitoring [9,29], industrial applications [30], electric power distribution [23,31], education [24], wastewater treatment [32], traffic control [33], telecommunications [34], oil production [28], and so on.

As natural gas pipeline control stations are strategically placed in remote areas that often lack a reliable and consistent power supply, resulting in power generation through conventional fossil fuels or the internal use of natural gas. To conserve the depleting natural gas resource, minimizing or eliminating its internal use for power generation is essential, which can be achieved through using renewable energy sources. Therefore, a hybrid power system along with a cost-effective and efficient SCADA system is essential to ensure efficiency and safety. The SCADA system will collect data and remotely monitor and coordinate control from various field instrument devices spread across different locations, ensuring the optimal operation of these remote stations with a reliable power supply. To the best of the author's knowledge and based on the reviewed literature, there are currently no SCADA systems specifically designed for hybrid power monitoring at remote natural gas pipeline control stations that minimize internal natural gas consumption for enhanced efficiency, use both GSM and Wi-Fi communication, are low-cost and power-efficient, and utilize IoT-based open-source platforms instead of proprietary vendor-specific network protocols. These facts inspired us to create an IoT-based SCADA system for such control stations. It is important to note that the research presented in this paper builds upon our previous work published in [4]. Our primary goals were to design a cost-effective solution for remote natural gas pipeline control stations that facilitates fast and easy development while being capable of measuring various parameters. Table 2 compares the proposed SCADA system with some prior research, emphasizing its distinct features and capabilities that are not present in other systems.

**Table 2.** Comparison of SCADA systems.

| **Ref.** | **SCADA System Features** | | | | | | |
|---|---|---|---|---|---|---|---|
| | **SCADA Platform** | **Open Source** | **GSM** | **Wi-Fi** | **Micro-Controller** | **Protocol** | **Total Cost ($)** |
| [9] | Haiwell IoT Cloud | X | X | ✓ | Raspberry Pi Zero | TCP | 200 |
| [18] | Node-RED | ✓ | ✓ | X | Arduino Uno | MQTT | Not Mentioned |
| [21] | Node-RED | ✓ | X | ✓ | Arduino | Firmata | Not Mentioned |
| [22] | Node-RED | ✓ | X | ✓ | ESP32-E | MQTT | 94.5 |
| [26] | Emoncms | ✓ | X | Serial port Comm. | Arduino Mega 2586 | MQTT | 761.72 |
| [27] | Node-Red | ✓ | X | ✓ | Arduino Mega2560 | Firmata | 93 |
| [28] | Node-Red | ✓ | X | ✓ | Arduino Uno<br>Arduino Mega | HTTP<br>Firmata | Not Mentioned |
| [29] | Raspberry Pi (Raspbian OS) | ✓ | X | UART + WiFi | Arduino Uno<br>Arduino Mega | TCP/IP | Not Mentioned |
| [35] | Wonderware Intouch + ThingSpeak | ✓ | X | ✓ | Arduino Mega2560 | Not Mentioned | Not Mentioned |
| Proposed SCADA | Blynk | ✓ | ✓ | ✓ | ESP32-E | HTTP<br>TCP/IP | 40.1 |

This research introduces a cost-effective, energy-efficient, open-source SCADA system utilizing the latest IoT-based SCADA architecture. This proposed design integrates web services with traditional SCADA for enhanced supervisory control and monitoring. The proposed SCADA system employs robust, widely accessible components to perform the four fundamental functions of a commercial SCADA system: remote monitoring, data collection and presentation, networked data transmission, and supervisory control [12]. This study aims to design a SCADA system featuring the following elements:

1. The proposed system is a cost-effective SCADA solution with a user-friendly HMI interface that can be accessed remotely, uses minimal power, and incorporates the most recent IoT architecture.
2. The proposed system's scalability is enhanced by using Blynk, an open-source platform, along with commercially available components. This approach allows flexibility in selecting communication channels and the number of sensors.

3. The proposed system's configuration is set up on a local host using the Blynk visual programming language, which is accessible via a web browser for convenient control.
4. Web-based system offers intuitive dashboards and robust data analytics capabilities for live monitoring and control.
5. Considering the remote locations of natural gas pipeline control stations and the potential lack of access to 3G, 4G, or Wi-Fi networks, the proposed SCADA system is designed and tested using both GSM and Wi-Fi modules for communication. This approach enhances the system's adaptability, enabling users to remotely check the system status.

## 3. System Description

The proposed SCADA system, intended to be both cost-effective and open-source, employs the IoT SCADA architecture, recognized as the latest development in SCADA technology. SCADA enables the real-time exchange of data among central control and field devices associated with distributed operations.

Given the remote locations of natural gas pipeline control stations and the potential lack of access to 3G, 4G, or Wi-Fi networks, two configurations are considered for integrating the hardware components. In the first configuration, Field Instrumentation Devices (FIDs), such as voltage and current sensors used to measure various site parameters, interface with a Remote Terminal Unit (RTU). An ESP32 microcontroller serves as the RTU and is responsible for receiving, processing, and transmitting data collected from the FIDs. Considering the unavailability of 3G, 4G, or Wi-Fi networks in this configuration due to the remote location of natural gas pipeline control stations, the data collected by the ESP32 microcontroller in this configuration is transmitted to the Master Terminal Unit (MTU) via a GSM SIM800L module. The MTU plays a pivotal role in data acquisition, programming, visualization, and storage. In this configuration, the MTU is Blynk, a versatile software suite that facilitates the prototyping, deployment, and remote management of connected electronic devices at any scale. Blynk allows users to connect their hardware to the cloud and develop mobile and web applications for remote control. It provides real-time and historical data analysis capabilities, remote device control, alerts, and more. Blynk features an intuitive drag-and-drop interface for constructing custom dashboards and interfaces without necessitating coding skills, supporting a wide range of microcontrollers and single-board computers [36]. A mobile device can be used as an HMI to monitor and analyze real-time data via TCP/IP communication protocol in this configuration.

In the second configuration, assuming the availability of 3G, 4G, or Wi-Fi networks, data collected by the ESP32 are transmitted via its integrated Wi-Fi module to the Master Terminal Unit (MTU). The MTU can be categorized into hardware and software components in this configuration. A DELL laptop (11th Generation Intel(R) Core(TM) i5) serves as the primary hardware for the MTU. As for the software component, the system utilizes Windows 10 Pro as an operating system alongside Blynk, an IoT-based platform. Communication between the RTU and MTU in this configuration is facilitated through the HTTP protocol. Figure 1 illustrates the configuration of the proposed SCADA system.

### *3.1. Components of the Designed SCADA System*

This research article section provides a brief overview of the cost-effective hardware components and software utilized to implement the proposed SCADA system architecture. The hardware components include voltage and current sensors to acquire site parameters; the versatile ESP32 microcontroller, which functions as an RTU for collecting, processing and transmitting FIDs data; an LCD; an SIM800L GSM module, which provides GSM functionality to the ESP32 microcontroller when 3G, 4G, or Wi-Fi networks are unavailable; the LM2596 voltage regulator, which serves as a buck converter to step down voltage; and a 5 V single-channel relay. The software components feature Blynk, an IoT-based platform that functions as an MTU and is employed to develop a user interface that can be accessed locally for the proposed SCADA system.

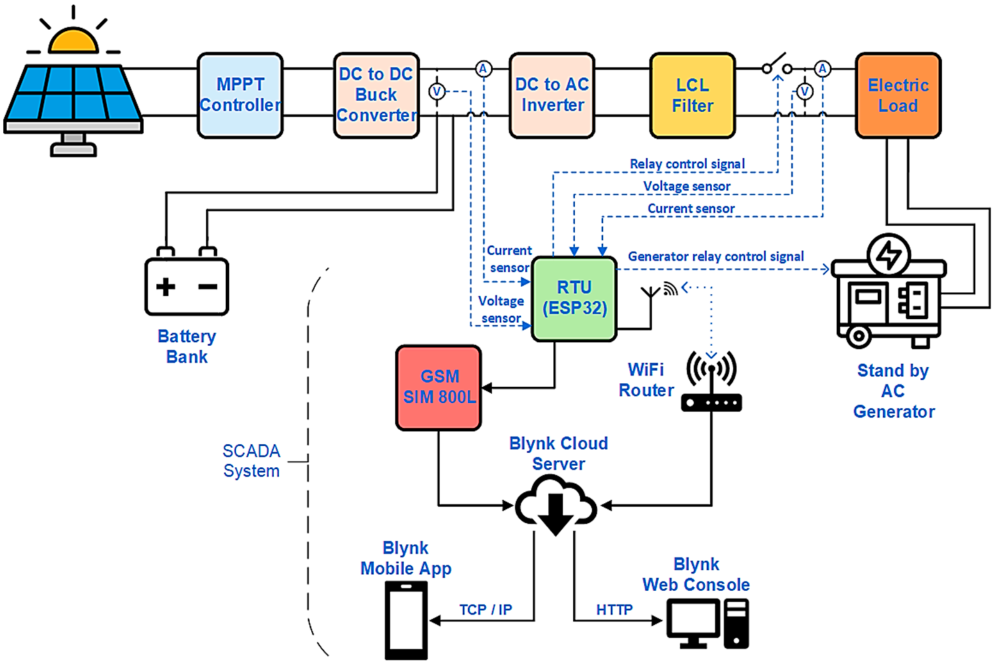


**Figure 1.** Proposed SCADA system schematic diagram.

The suggested SCADA system employs current and voltage sensors as FIDs to collect the necessary site data. The system utilizes ACS712 Hall Effect Current Sensors and ZMPT101B Voltage Sensor as the analog sensors. The analog sensors operate at a signal voltage (VCC) of 5V, while the ESP32-WROOM-32E microcontroller operates between 0 V and 3.3 V. Due to the voltage difference, the current sensor cannot be connected directly to the ESP32-WROOM-32E's analog-to-digital converter (ADC) pins. To ensure compatibility, level shifting is achieved through a step-down resistor configuration, aligning the sensor's output with the 3.3 V input requirement of the ESP32-WROOM-32E, thus preserving measurement accuracy. The characteristics and usage of these sensors in this system are outlined below.

### 3.1.1. Hall Effect Current Sensor (ACS712)

The ACS712 Hall Effect Current Sensor, designed and manufactured by Allegro MicroSystems in Manchester, NH, USA, is a cost-effective, fully integrated linear current sensor IC utilizing Hall Effect technology. It features 2.1 kVRMS isolation and offers a current path with minimal resistance. Operating with a low-noise analog signal pathway and powered by a 5 V single supply, it offers an output sensitivity ranging from 66 to 185 mV/A with minimal magnetic hysteresis. In operation, the sensor detects the magnetic field generated by current flowing through its copper conductor, converting this field into a corresponding output voltage [37].

The ACS712 sensor comes in three different models, optimized for various current measurement ranges ±5 A, ±20 A, and ±30 A. For our research, we utilized the ACS712 current sensors rated at ±30 A to monitor the current in our designed PV system. This module outputs 2.5 V when there is 0 A current and 5 V when there is 30 A current. For the physical connections, a voltage divider circuit is used to adapt the 5 V signal required by the Current Sensor to the 3.3 V signal level supported by the pins of ESP32 microcontroller. Figure 2 illustrates the setup with step-down resistors connecting the sensor to the ESP32, and Equation (1) defines the formula for voltage division.

$$V_{ESP32} = \frac{R_2}{R_1 + R_2} \times VCC_{(ACS712)} \tag{1}$$

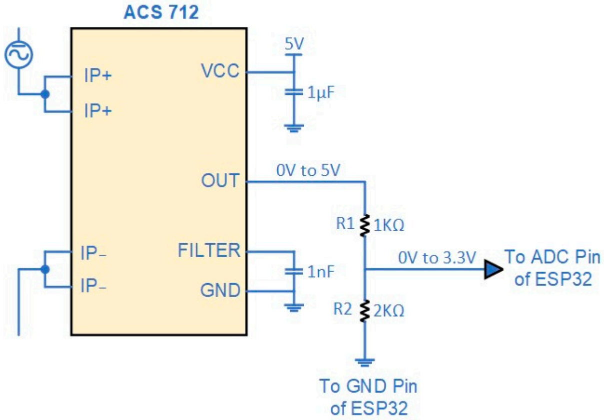


**Figure 2.** ESP32 voltage divider connection.

### 3.1.2. Voltage Sensor (ZMPT101B)

The Voltage Sensor ZMPT101B is a widely used module for measuring AC voltage, ideal for applications in power monitoring and energy management. It features a high-precision voltage transformer with a voltage ratio of 1:1 and an operational amplifier circuit, enabling accurate voltage measurements by amplifying the signal for analog-to-digital conversion [38].

The sensor provides electrical isolation between the high-voltage AC side and the low-voltage DC side, ensuring safety and protecting microcontrollers from voltage spikes. The Voltage Sensor ZMPT101B provides an output signal ranging from 0–5 V analog, operates within a voltage range of DC 5 V–30 V, and is capable of measuring up to 250 V AC. It has a rated input current of 2 mA and dimensions of 49.5 mm in length and 19.4 mm in width. The sensor functions effectively within an operating temperature range of −40 °C to +70 °C [39]. While the module is specifically designed for AC voltage and requires calibration for precise readings, its integration in microcontroller-based systems makes it suitable for smart meters, home automation, and industrial automation applications. Figure 3 shows the internal schematic diagram of ZMPT101B voltage sensor.

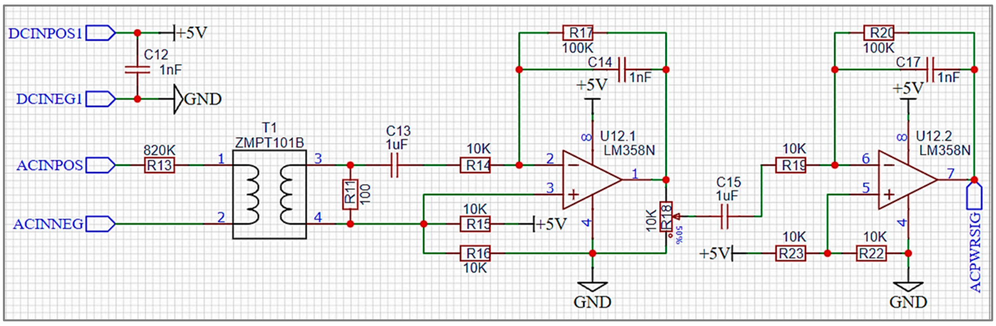


**Figure 3.** Internal schematic diagram of ZMPT101B voltage sensor.

### 3.1.3. ESP32-WROOM-32E Microcontroller (RTU)

The ESP32 series includes a range of microcontroller models such as ESP32-WROOM and ESP32-WROVER, developed by Espressif Systems. ESP32-WROOM-32E is a versatile and powerful microcontroller module from Espressif Systems, ideal for IoT applications featuring integrated Wi-Fi and Bluetooth capabilities [40]. The ESP32-WROOM-32E is a specific module within the ESP32 series, containing the ESP32-D0WD-V3 chip. It is designed for a wide range of applications, providing robust performance with its dual-core processor, integrated Wi-Fi, and Bluetooth. It features a dual-core Tensilica LX6 processor running up to 240 MHz, 520 KB of SRAM, 4 MB of flash memory, and supports both 2.4 GHz and 5 GHz Wi-Fi as well as Bluetooth v4.2.

The ESP32-WROOM-32E features numerous input and output pins, along with ample processing power, allowing it to handle additional sensors and energy equipment. With

34 programmable GPIO pins, 18 ADC channels, 2 DAC channels, and numerous peripheral interfaces (SD card, SPI, I2C, UART, I2S, CAN, SDIO, GPIO, IR), it offers extensive connectivity options. Operating at 3.3 V, it boasts low power consumption, with deep sleep mode drawing only 10 µA. Security features include hardware encryption, secure boot, and flash encryption. The module also includes a real-time clock, touch sensors, and built-in hall and temperature sensors, making it suitable for smart home devices, industrial automation, and wireless sensor networks. It can be programmed using various environments, such as Arduino IDE and ESP-IDF, supporting multiple languages, such as C and Python [41]. Figure 4 illustrates the pin layout of the ESP32 module with the ESP32-WROOM-32 chip.

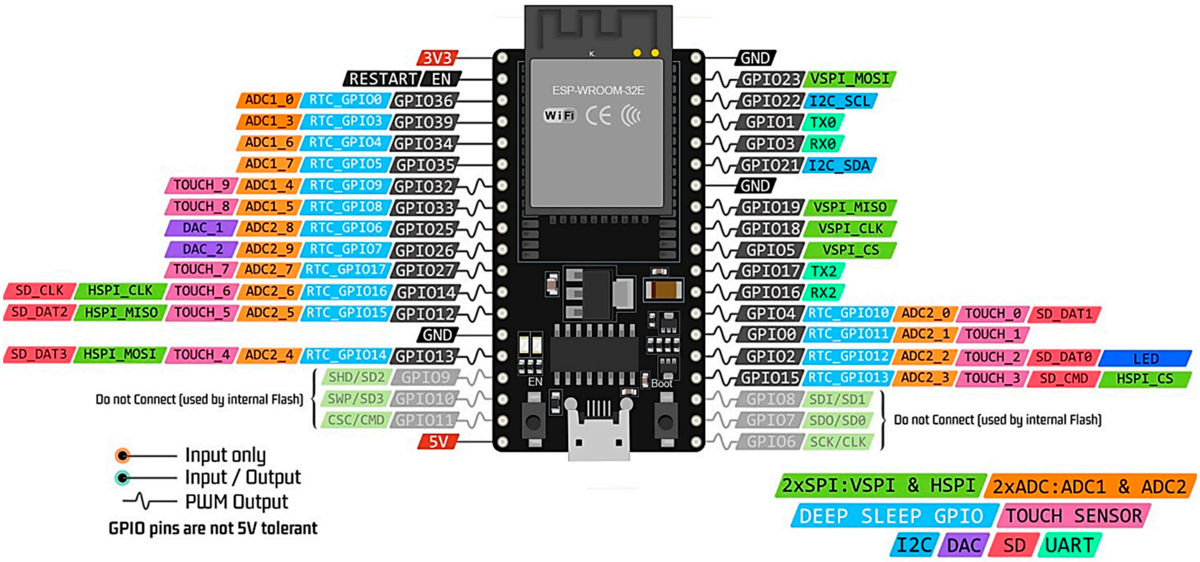


**Figure 4.** Pin layout of ESP32 module with the ESP32-WROOM-32 chip.

### 3.1.4. GSM SIM800L Module

GSM, or Global System for Mobile Communications, is a widely adopted standard for 2G digital cellular networks, developed by the European Telecommunications Standards Institute (ETSI) in 1989. It employs digital transmission for voice and data services, enhancing call quality and enabling features such as SMS and internet access. Operating in various frequency bands (900 MHz and 1800 MHz in Europe, 850 MHz and 1900 MHz in North America), GSM uses SIM cards for user identification and facilitates easy switching between devices.

The SIM800L module is a compact and versatile GSM/GPRS module designed for embedded applications, enabling devices to connect to GSM networks for voice, SMS, and data services. It supports quad-band frequencies (850/900/1800/1900 MHz), ensuring global usability. Measuring about 25 mm × 23 mm, it is easy to integrate into various projects [42]. The module requires a 3.4 V to 4.4 V power supply and features low power consumption, typically drawing 1.1 mA in sleep mode. The SIM800L module functions within a temperature range of −40 to +85 °C and includes a microSIM card socket, UART interface, and IPX antenna connector. When powered, the module automatically searches for and connects to cellular networks, with an onboard LED indicating connection status. With features such as auto baud rate detection, power-saving mode, and an integrated LDO voltage regulator, the SIM800L supports GPRS multi-slot class 12/10, CS-1, CS-2, CS-3, and CS-4 coding schemes, and offers a maximum data rate of 85.6 kbps for uploading and downloading. These specifications make the SIM800L ideal for IoT applications, remote monitoring, and GSM-based communication projects due to its affordability and reliability [43].

### 3.1.5. LM2596 Buck Converter

The LM2596 is a versatile and efficient step-down (buck) voltage regulator designed by Texas Instruments, ideal for converting a higher DC input voltage (4.5 V to 40 V) to a lower DC output voltage, adjustable between 1.23 V and 37 V. It can deliver up to 3 A of continuous output current with high efficiency, typically around 80–90%, minimizing heat generation and power loss. Operating at a fixed switching frequency of 150 kHz, the LM2596 includes built-in thermal shutdown and current limit protection, ensuring reliable performance. With a low standby current, it enhances efficiency during idle periods. Due to its robust design and protective features, this regulator is widely used in power supplies for electronic devices, battery chargers, distributed power systems, and automotive and industrial applications [44].

In our proposed system, the LM2596 effectively lowers the higher DC voltage generated by PV panels to a steady lower voltage, necessary for charging batteries or operating devices. By fine-tuning the output voltage to meet the needs of the charge controller and connected loads, the LM2596 improves the overall efficiency and dependability of the PV system. Its adjustable output voltage and current limits provide versatility in system setup and enhancement. In essence, the LM2596 buck converter is pivotal in ensuring the effectiveness and longevity of PV systems by efficiently managing voltage from solar panels.

### 3.1.6. Liquid Crystal Display (LCD)

Liquid Crystal Display (LCD) is a flat-panel display technology that utilizes liquid crystals sandwiched between polarized glass layers to create images or text. It operates using two main types: Passive Matrix LCDs, which are cost-effective with simpler electrode patterns but have slower response times and limited viewing angles, and Active Matrix LCDs (TFT-LCDs), featuring thin-film transistors (TFTs) per pixel for faster response times, better image quality, and wider viewing angles. Operating typically between 4.7 V to 5.3 V, LCDs have a low current consumption of about 1 mA without backlight. They are commonly used in alphanumeric modules capable of displaying letters and numbers, with typical configurations featuring two rows of 16 characters each. Each character is composed of a 5 × 8 pixel matrix. LCDs can operate in both 8-bit and 4-bit modes, allowing flexible interfacing with microcontrollers. Additionally, they support custom character generation and are available with green and blue backlight options, making them versatile for various applications in consumer electronics and industrial devices [45].

### 3.1.7. 5V Single Channel Relay

A 5 V single-channel relay is an electromechanical switch triggered by a 5 V DC signal, compatible with microcontrollers and digital circuits. It can handle loads up to 10 A at 250 V AC or 30 V DC, making it ideal for controlling high-power devices in home automation, industrial systems, automotive electronics, and DIY projects. The relay features normally open (NO) and normally closed (NC) contacts, providing flexible switching options. It offers electrical isolation between the low-voltage control and high-voltage load circuits, enhancing safety and preventing interference. A 5 V single-channel relay is a crucial component in electronic circuits, enabling the control of high-power devices with a low-power signal. Its ease of integration, safety features, and versatility make it widely used in home automation, industrial control, automotive electronics, and various DIY projects [46].

### 3.1.8. Master Terminal Unit (MTU)

In SCADA systems, the Master Terminal Unit (MTU) serves as the central hub for data collection, processing, monitoring, and controlling operations. It collects data from Remote Terminal Units (RTUs) and Programmable Logic Controllers (PLCs), processes and stores these data, and provides live monitoring and system control. The MTU manages communication between the central control room and remote sites, ensuring reliable and secure data transfer using various communication protocols. Additionally, it often includes or interfaces with human–machine interface (HMI) software, allowing operators to interact with the system via graphical displays and dashboards. The MTU's reliability, scalability, and real-time processing capabilities make it essential for managing infrastructure in utilities, manufacturing, oil and gas, and transportation sectors.

The MTU can be categorized into hardware and software components. Our proposed SCADA system features two configurations. In the first configuration, Blynk, an IoT-based platform, is the MTU. The GSM SIM800L module facilitates communication between the RTU and MTU through the Blynk cloud server. Data transmission can occur either via text messages using 2G technology or through the TCP/IP communication protocol. A mobile device along with the Blynk app is used as HMI. In the case of the second configuration, A DELL laptop (11th Generation Intel(R) Core (TM) i5) serves as the primary hardware for the MTU. As for the software component, the system utilizes Windows 10 Pro as an operating system alongside Blynk, an IoT-based platform. Communication between the RTU and MTU in this configuration is facilitated through the HTTP protocol. In our proposed SCADA system, the MTU is Blynk, a versatile software that makes it easier to prototype, install, and remotely manage linked electronic devices of any size. Blynk enables users to link the hardware to the cloud and develop mobile and web applications for remote control. It offers real-time and historical data analysis, remote device control, alerts, and more. Featuring an intuitive drag-and-drop interface, Blynk allows for the development of custom dashboards and interfaces without requiring coding skills and supports a wide range of microcontrollers and single-board computers.

Designing an open-source SCADA system involves implementing robust security protocols to protect data and ensure system integrity, especially in remote locations [47]. The Blynk platform involves implementing robust security protocols to protect data and ensure system integrity, particularly in remote locations. Data are secured through SSL/TLS encryption during transmission and end-to-end encryption to prevent unauthorized access. Secure authentication mechanisms and role-based access control ensure only authorized users can access the system. Network security is enhanced with firewalls, VPNs, and strong WiFi security measures such as WPA3 encryption. Regular firmware updates, secure boot, and code signing protect the device itself, while physical security measures prevent unauthorized access. Real-time monitoring, logging, and auditing help detect and address security breaches, while checksums, hashing, redundancy, and backups ensure data integrity. Secure remote updates and two-factor authentication for remote access further enhance the system's security, making it resilient against evolving threats [36].

## 4. Implementation Methodology

While deploying the proposed SCADA system, analog sensors act as FIDs and are connected to the previously designed hybrid power system for collecting the required data. The ESP32-WROOM-32E microcontroller acts as an RTU, receiving data from FIDs and transmitting it through the local set-up Wi-Fi network to Blynk, an IoT-based server designed for monitoring remotely and supervisory control. Current sensors for monitoring current levels are connected in series with these components. Voltage sensors for monitoring voltage levels in a hybrid power system are connected in parallel to the load and battery bank. Additionally, the relay is connected in series with the load to control the power supply, whether from the PV system or a standby AC generator operating on natural gas. All system components are properly grounded.

The SCADA system under consideration is deployed using two different approaches:

- An open-source IoT platform is built which features the ESP32-WROOM-32E microcontroller paired with a SIM800L GSM module. It interfaces with the Blynk app and its cloud server through the TCP/IP protocol. This approach addresses scenarios where access to 3G, 4G, or Wi-Fi networks might be unavailable.
- An IoT-based open-source platform is designed using an ESP32-WROOM-32E microcontroller directly interfaced with the Blynk app and its cloud server through a local Wi-Fi router and HTTP protocol, providing a web console-based HMI.

To implement the proposed SCADA system, the process begins with designing a schematic diagram to illustrate the connections of the various hardware components. This design is created using EasyEDA software (V6.5.44). EasyEDA is a web-based software tool for electronic design automation (EDA). It allows users to design schematics, simulate circuits, and create PCB layouts in a user-friendly interface. The software supports importing and exporting various file formats, integrates with popular component libraries, and offers real-time collaboration features [48]. The next step involves soldering the hardware components such as ESP32-WROOM-32E, GSM SIM800L, current sensors, voltage sensors, relay, buck converters, and LCD onto the zero PCB. The Blynk app, an open-source IoT platform, is configured for the software setup. Then, code is developed for the ESP32 to manage all SCADA system parameters. Finally, the ESP32 is programmed using Arduino IDE software (Version 2.3.2), and its functionality is tested and verified with the Arduino IDE serial monitor. Figure 5 shows the schematic diagram of the proposed SCADA system designed using EasyEDA software.

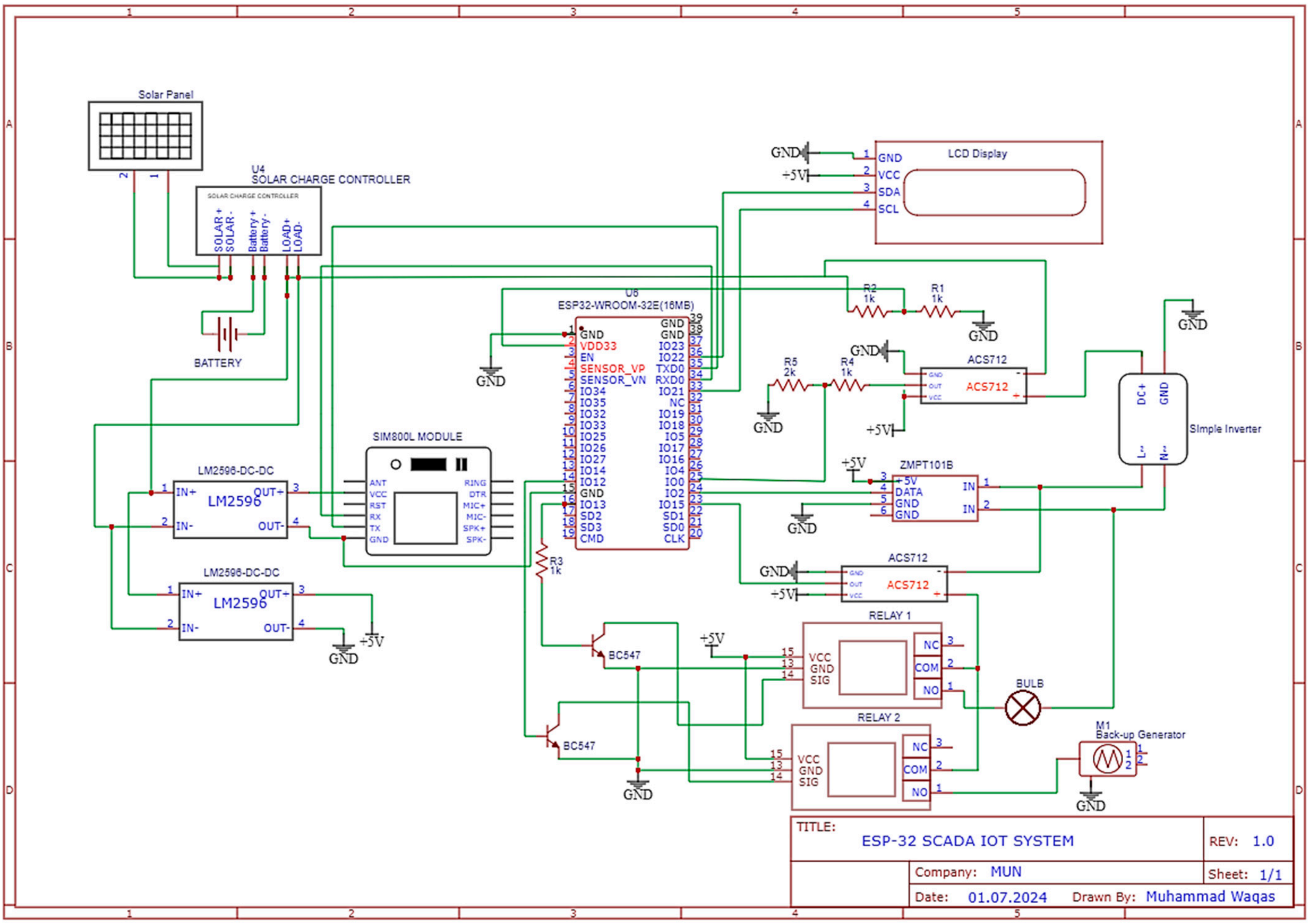


**Figure 5.** Proposed SCADA system schematic diagram.

As depicted in the schematic diagram, the solar panel connects to a solar charge controller, commonly referred to as an MPPT controller, which then connects to a 12-volt, 5-ampere battery. The output from the charge controller feeds into two LM2596 buck converters, which step down the higher voltage from the PV panel to a stable lower voltage (from 12 V to 5 V). Furthermore, a basic inverter is incorporated to convert DC to AC, which is crucial for the selected site, where the standard operating voltage is 220 V AC. Table 3 illustrates the connections of current and voltage sensors to the ESP32 microcontroller.

**Table 3.** ESP32-WROOM-32E interconnections with FIDs and other components.

| Component No. | Description | Analog/Digital | ESP32-E Pin No. |
|---|---|---|---|
| 1 | ACS712 Current Sensors | Analog | 23, 25 |
| 2 | ZMPT101 Voltage Sensor | Analog | 24 |
| 3 | GSM SIM800L | Digital | 34, 35 |
| 4 | 5 V Relays | Digital | 16, 14 |
| 5 | LCD Display | Analog | 21, 22 |

Algorithm 1 presents the pseudocode utilized in the ESP32-WROOM-32E and GSM SIM800L, programmed with Arduino IDE software (Version 2.3.2), to demonstrate the implementation methodology.

**Algorithm 1:** GSM, Voltage, and Current Sensor Data Reading Algorithm.

```
Start
1.    Initialization
    1.1.    Include necessary libraries (ZMPT101B, Arduino JSON, Software Serial, LiquidCrystal_I2C).
    1.2.    Initialize the LCD and define the RX, TX, sensors, and relay pins.
    1.3.    Initialize 'software serial' for communication with ESP-32.
    1.4.    Initialize variables for storing sensor reading and calibration values.
    1.5.    Set the sensitivity for the AC voltage sensor and set the pins for sensor inputs and relay output.
2.    Setup Function
    2.1.    Begin serial communication for debugging and ESP-32 communication.
    2.2.    Initialize the LCD and set pin modes for the sensor and relay.
    2.3.    Set the initial state of the relay to low; Print "initializing" on the LCD.
    2.4.    Initialize the Wi-Fi connection and SIM800L.
3.    Main Loop (Loop Function)
    3.1.    Continuously check for incoming data from ESP-32:
        3.1.2.    If data is received, read the value; turn the relay ON or OFF based on the received value.
    3.2.    Read and process sensor data.
        3.2.1.    'take DC vol ()'-Read and calculate DC voltage; 'take DC cur ()'-Read and calculate DC.
        3.2.2.    'take AC vol ()'-Read and calculate AC voltage; 'take AC cur ()'-Read and calculate DC.
    3.3.    Check if AC voltage is below the threshold:
        3.3.1.    If AC voltage is below the threshold, turn ON the generator by setting the relay HIGH
        3.3.2.    Otherwise, turn OFF the generator by using the relay LOW.
        3.3.3.    Update the LCD with the latest sensor readings.
        3.3.4.    Send the sensor data to ESP-32 in JSON format.
4.    Functions for sensor readings:
    4.1.    'takeDCvol( )'
        4.1.1.    Read analog value from the DC voltage pin; Calculate the DC voltage value of the serial monitor
    4.2.    'takeDCcur( )'
        4.2.1.    Read the analog value from the DC pin; Calculate the DC using the calibration formula.
        4.2.2.    Print the DC value to the serial monitor.
    4.3.    'takeACvol( )'
        4.3.1.    Use the ZMPT101B library to get the RMS voltage for AC.
        4.3.2.    Ensure the AC voltage is within a specific range; Print the AC voltage value to the serial monitor.
    4.4.    'takeACcur( )'
        4.4.1.    Calculate peak-to-peak voltage from the AC sensor.
        4.4.2.    Convert the peak-to-peak voltage to RMS voltage; Calculate the RMS current using the calibration formula;
                  Print the AC value to the serial monitor.
```

**Algorithm 1:** *Cont*.

```
5.      LCD update function
        5.1.    'LCD update( )'
                5.1.1.  Clear the LCD.
                5.1.2.  Set the cursor positions and print DC and AC voltage and current on the LCD.
6.      Data sending function:
        6.1.    'send data( )'
                6.1.1.  Create a JSON document and populate it with sensor data.
                6.1.2.  Serialize the JSON document to a string; Send the JSON string via Wi-Fi.
                6.1.3.  Send the JSON string via GSM using SIM800L.
                6.1.4.  Delay for 2 s before sending the next set of data.
7.      Help function:
        7.1.    'getVPP1( )'
        7.2.    Measure the peak-to-peak voltage from the AC sensor.
        7.3.    Return the calculated peak to peak voltage.
End
```

The designed system can positively impact the environment by reducing reliance on fossil fuels through solar energy, minimizing greenhouse gas emissions, and decreasing the need for frequent site visits, which lowers vehicle emissions. Continuous monitoring enhances resource management and reduces the risk of environmental contamination. Overall, the deployment of this SCADA system can be environmentally sustainable with proper planning and adherence to environmental protection protocols.

## 5. Prototype Design and Setup of the Blynk IoT Platform

The design of hardware is crucial in research dealing with physical systems. It allows for testing, prototyping, and real-world validation, ensuring that the proposed system functions correctly and performs well. Additionally, it enables customization, improves hands-on skills, and provides a tangible proof of concept. This section provides an overview of the hardware setup for the proposed IoT-based open-source SCADA system, which uses a single 10-watt PV panel and a 12 V battery exclusively for testing. Figure 5 illustrates the connection of the analog voltage and current sensors to the ESP32-WROOM-32E microcontroller on a zero PCB, with step-down resistors arranged for the current sensor. The sensor inputs are wired to key points in the PV System to measure and acquire the necessary current and voltage ratings. Section 3 outlines the power supplies for each component. The designed circuit is powered by the DC voltage generated by the PV panel and stored in the 12 V DC battery connected via the MPPT. A basic inverter is used to convert this DC power to AC power. ESP32-WROOM-32E WiFi module and the GSM SIM800L module are used to form the two previously described configurations of the proposed SCADA system. Figure 6 shows the prototype design of the proposed SCADA system.

For the software part, Blynk, a versatile IoT platform that facilitates the development of mobile and web applications for controlling hardware remotely is used. It supports a wide range of microcontrollers and single-board computers, making it highly adaptable for various IoT projects. The platform operates on a client–server model, where the Blynk app acts as the client running on a mobile device or web browser and the Blynk cloud server serves as the backend infrastructure.

For setting up the Blynk IoT platform, users begin by creating an account and project within the Blynk app, specifying the hardware type (such as ESP32) and connection method (Wi-Fi or GSM). Setting up the Blynk app involves creating a user interface (UI) by dragging and dropping widgets, such as buttons, sliders, and graphs, onto a virtual canvas. These widgets are then linked to specific hardware components or sensors connected to the microcontroller via interfaces such as UART or Wi-Fi. The Blynk app communicates with the cloud server using the MQTT or HTTP protocols, allowing for seamless data exchange and control commands between the user interface and the hardware.

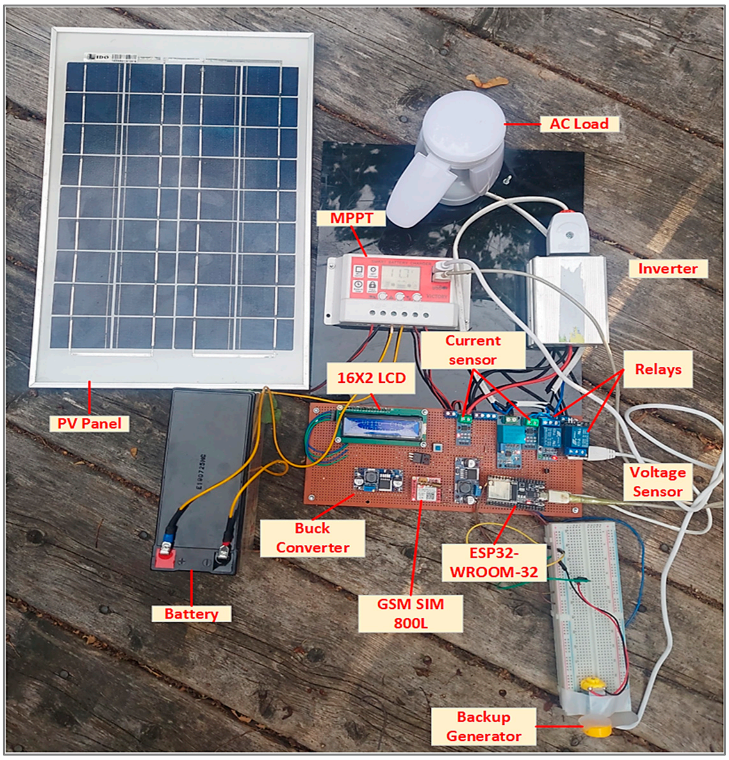


**Figure 6.** Prototype design of the proposed SCADA system.

On the backend, the Blynk cloud server manages authentication, data storage, and communication between multiple Blynk apps and connected devices. It provides secure access to the IoT devices through tokens and APIs, ensuring data privacy and integrity. This setup enables real-time monitoring, control, and data logging from anywhere with an internet connection, making Blynk a powerful tool for IoT applications in research and beyond.

To ensure the system remains operational and effective over time, it is crucial to perform regular physical inspections of hardware components, such as the ESP32 microcontroller, sensors, and solar panels, to promptly identify and address any issues. Routine maintenance tasks include updating firmware to fix security vulnerabilities, conducting real-time monitoring with Blynk to detect anomalies, and periodically reviewing collected data for trends that may signal the need for maintenance. Preventive measures, such as cleaning, calibrating sensors, and keeping a stock of spare parts, are vital for minimizing downtime and maintaining system accuracy. Additionally, setting up a replacement schedule for components with limited lifespans and utilizing technical support from the open-source community and manufacturers will further enhance the system's reliability.

## 6. Experimental Setup and Results

To assess the real-time performance of the designed open-source SCADA system, it has been set up to analyze solar PV parameters such as current and voltage of PV installation on the rooftop of the core science building at Memorial University of Newfoundland. The system description outlined in Section 3 served as the basis for the hardware connections. Figure 7 shows the experimental setup of the proposed SCADA system. The PV installation is shown in Figure 8. It consists of 12 solar panels with a combined surface area of 14 square meters and each solar panel can produce up to 130 W and 7.6 A at maximum. The system includes six maximum power point tracking (MPPT) controllers and a battery bank comprising four lead-acid batteries to maximize and sustain energy efficiency, as shown in Figure 9. For this test, the SCADA system was connected to two panels, providing a maximum output of 260 W and 15.2 A, to focus on the system's data collection and control capabilities.

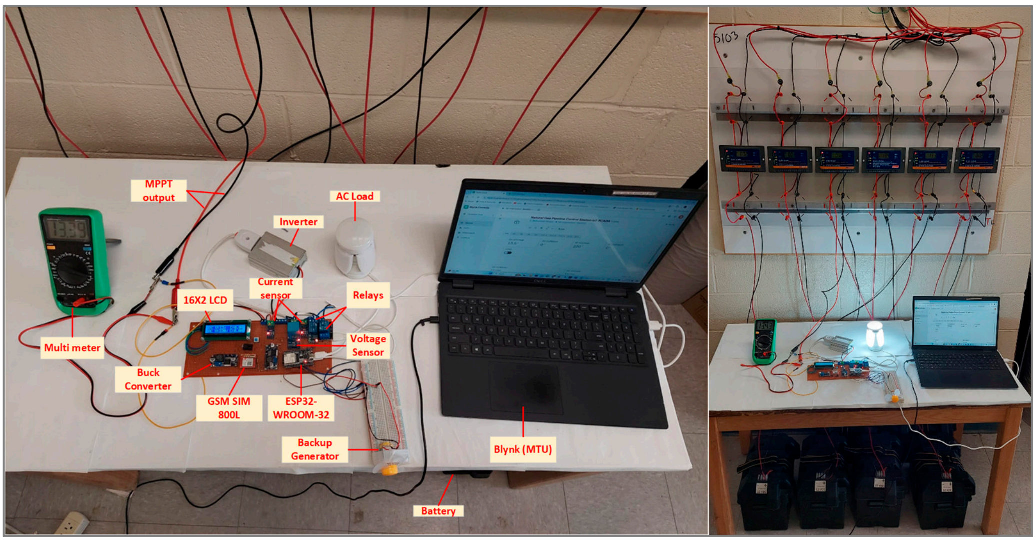


**Figure 7.** Designed SCADA system experimental setup.

**Figure 8.** PV panels setup at the rooftop of the MUN core science building.

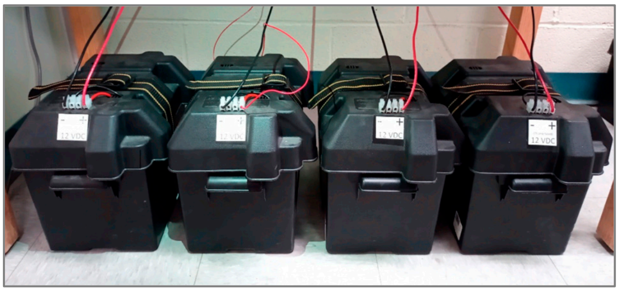


**Figure 9.** Battery bank.

The voltage and current sensors gather data and transmit it to the analog pins of the ESP32 microcontroller, as depicted in the system's schematic diagram shown in Figure 5. The ESP32 then processes these analog signals, applying the necessary conversion factors to determine the actual voltage and current values. Using these data, the ESP32 manages load switching via a 5 V single-channel relay. The processed information is subsequently transmitted to the Blynk cloud server using one of the two proposed configurations of the SCADA system. A digital multimeter was used at different points during the system testing phase to measure the required values. In both setups, the sensor data collected corresponded closely with the locally measured values from the multimeter, showing only slight differences. Moreover, during testing, an electric light bulb was used as a load to the battery, ensuring a substantial current to flow from the PV system through the MPPT for recharging the battery.

In the first configuration of the proposed SCADA system, considering the possibility of no local Wi-Fi access, the GSM SIM800L module is employed to transmit data collected by the ESP32-WROOM-32E microcontroller from FIDs. The GSM SIM800L module connects to the internet using the cellular network, utilizing APN credentials provided by the SIM card carrier. The microcontroller and the SIM800L module are managed through AT commands communicated over a serial interface. Once an internet connection is established, the microcontroller firmware utilizes the Blynk library to transfer data to the Blynk cloud server via the TCP/IP protocol over the GSM network, ensuring reliable communication. The Blynk app then displays real-time data from the PV system, including DC and AC voltage readings and current measurements, offering a detailed overview of the system's performance. Figure 10 shows the DC and AC voltage and current parameters with AC load OFF and ON. Similarly, Figure 11 shows a graphical representation of DC and AC voltage and current parameters recorded at different points and times within the PV system, as displayed on the Blynk mobile app.

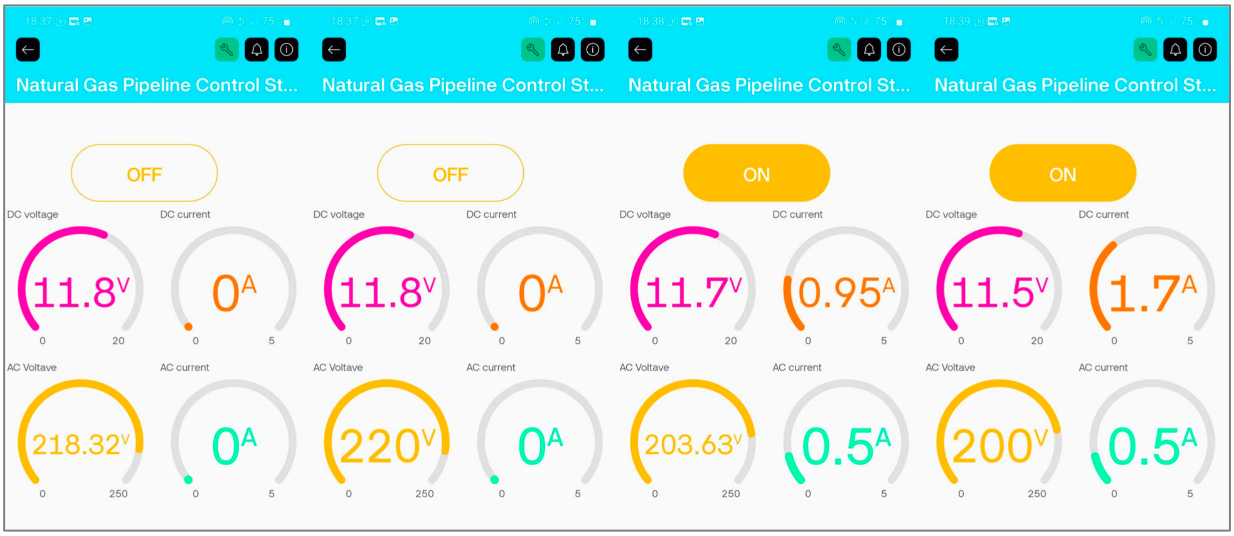


**Figure 10.** Display of FID parameters on Blynk mobile app using the GSM SIM800L module with load OFF and ON.

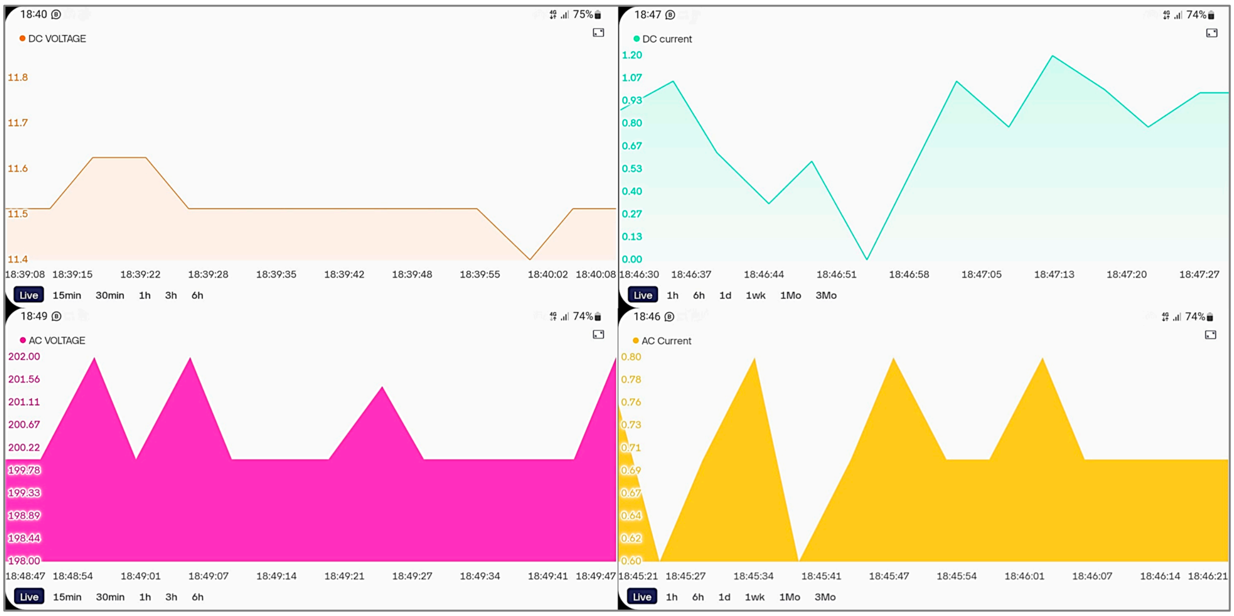


**Figure 11.** Graphical representation of FID parameters on Blynk mobile.

When there is no load connected to the power source, the terminal DC voltage reaches its maximum, known as the open-circuit voltage (Voc), while the current remains zero due to the absence of a closed circuit for current flow. When no load is connected and in the "OFF" state, the system registers DC voltage at 13 V and current at 0 A, while AC voltage stands at 220 V with a minimal current of 0 A, as shown in Figure 12.

**Figure 12.** FID voltage and current display on the LCD when the load is off.

Once a load is connected, the terminal voltage decreases from its open-circuit level due to internal source resistance and voltage drop across the load. Simultaneously, current begins to flow through the circuit, its magnitude determined by the load's resistance and the applied voltage as per Ohm's Law. Upon connecting a load, the AC voltage decreases due to the inverter's internal impedance and wiring resistance. Figure 13 illustrates the display of FIDs parameters on an LCD screen when the load is on.

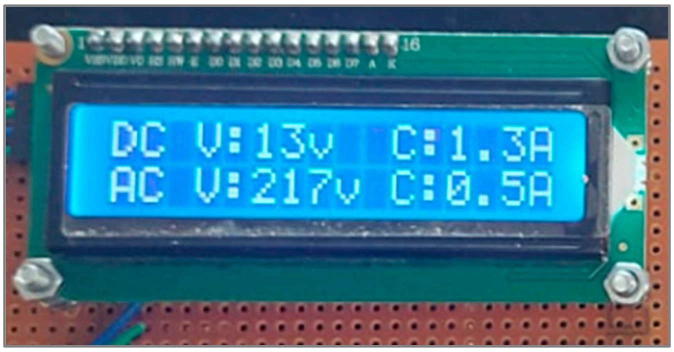
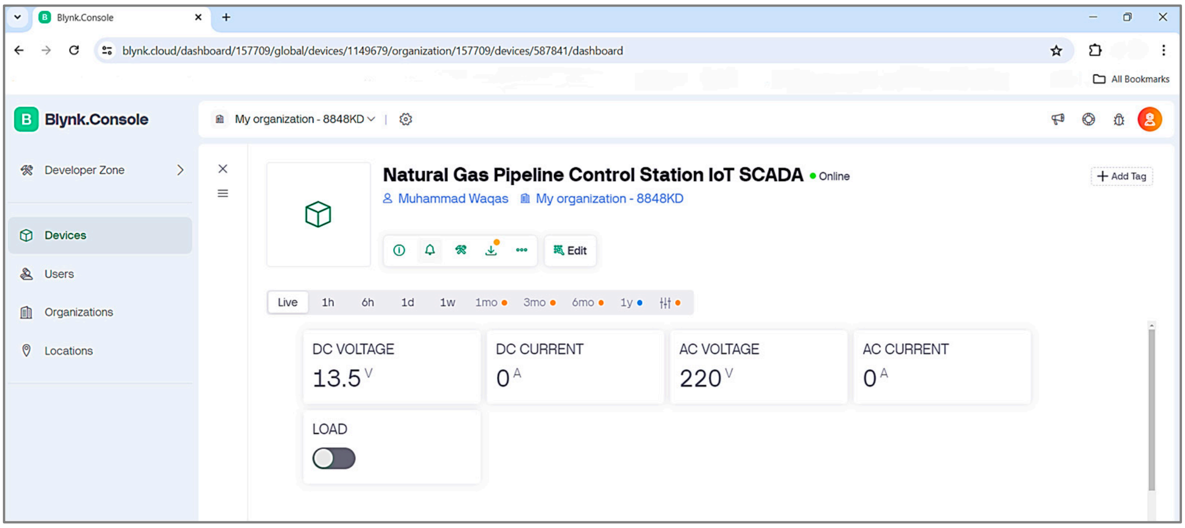


**Figure 13.** FID voltage and current display on the LCD.

Moreover, to validate the supervisory control capabilities of our system, the Blynk app facilitates remote control of the load connected to the PV system. Through the app, users can toggle the connected load between "ON" and "OFF" states using virtual pins that correspond to specific digital pins on the microcontroller controlling relays or switches linked to the load. The microcontroller processes commands using the Serial2.write(val); command within the BLYNK_WRITE(V4) function, which receives input from the Blynk application and transmits it to the relay to switch the load accordingly. This integration facilitates efficient remote monitoring and management of the PV system via the Blynk app, enhancing system usability and functionality.

In the second configuration, the ESP32 collects data and transmits it through its integrated Wi-Fi module to the Master Terminal Unit (MTU) using a local WiFi router. The MTU in this setup comprises both hardware and software components. A DELL laptop equipped with an 11th Generation Intel(R) Core(TM) i5 processor serves as the primary hardware for the MTU. On the software side, the system runs Windows 10 Pro and utilizes Blynk, an IoT-based platform. Communication among the RTU and MTU is facilitated via the HTTP protocol. The system connects to the Blynk cloud server using HTTP over port 80, facilitated by the BlynkSimpleEsp32.h library. Network credentials are configured for WiFi access, and data communication with external hardware is managed through

SoftwareSerial (Serial2). The primary functionality involves parsing JSON data received from Serial2, which includes measurements for DC voltage (DCvol), DC (DCcur), AC voltage (ACvol), and AC (ACcur). These values are processed, displayed on the serial monitor for debugging, and simultaneously updated on the Blynk web console using Blynk.virtualWrite() within the blynkupdate() function. The Blynk web interface presents live voltage and current measurements from various points in the PV system, allowing remote monitoring and control to detect operational irregularities. Figures 14 and 15 illustrate the Blynk console dashboard in both "OFF" and "ON" states, while Figure 16 depicts real-time PV data monitoring on the Blynk console.



DC VOLTAGE 13.5 V

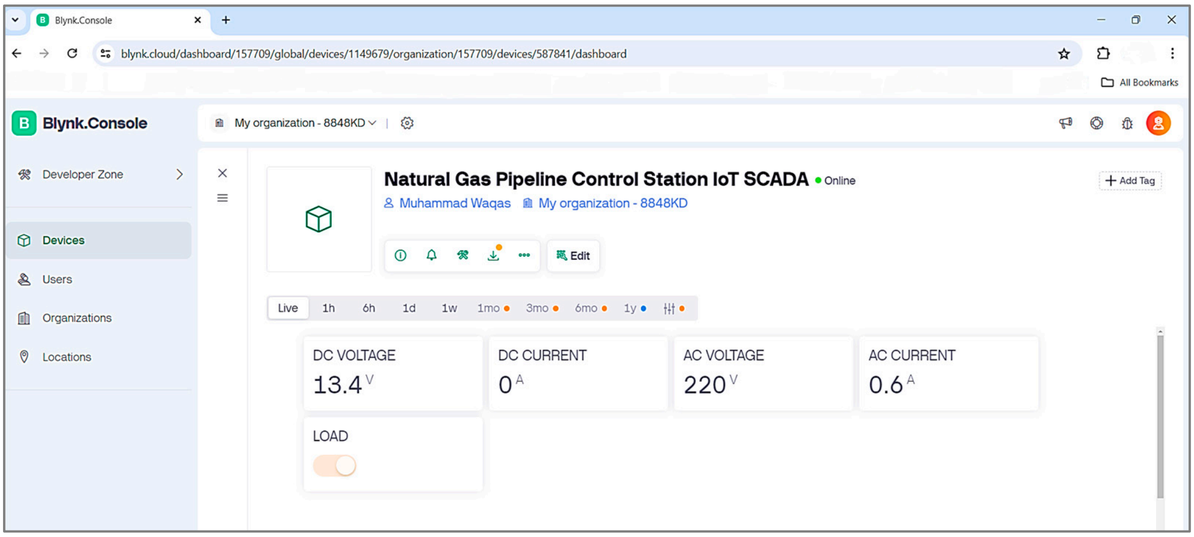
DC CURRENT 0 A

AC VOLTAGE 220 V

AC CURRENT 0 A

LOAD

**Figure 14.** Blynk console dashboard when the load is OFF.



DC VOLTAGE 13.4 V

DC CURRENT 0 A

AC VOLTAGE 220 V

AC CURRENT 0.6 A

LOAD

**Figure 15.** Blynk console dashboard when the load is ON.

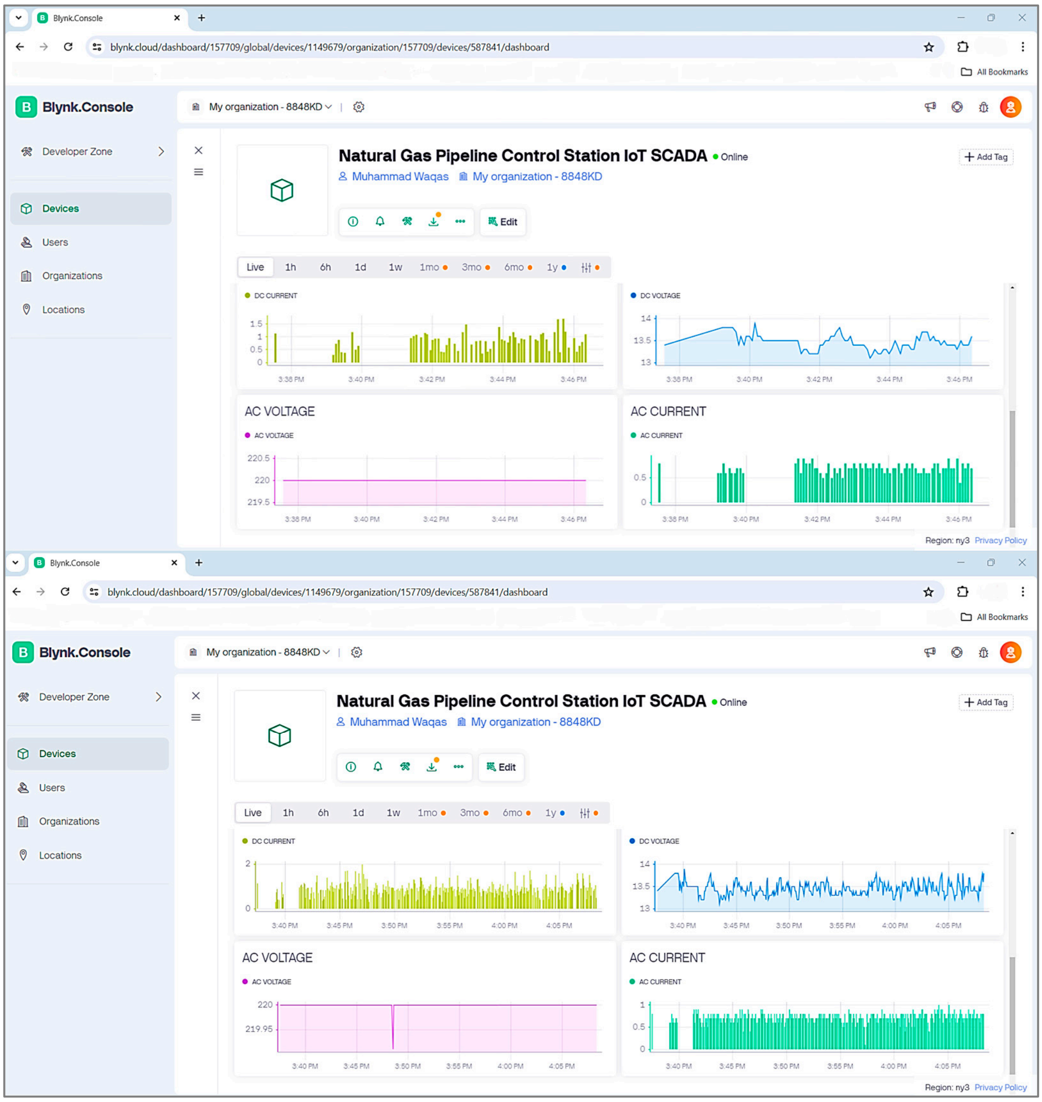


**Figure 16.** Real-time data monitoring on Blynk web console.

## 7. Discussion

This section describes the salient characteristics and advantages of the open-source, low-cost SCADA system designed for a remote natural gas pipeline control station in Pakistan.

**System Architecture:** The proposed system is built on the fourth and latest IoT SCADA architecture, incorporating the four fundamental components of a SCADA system previously discussed. It features Field Instrumentation Devices (FIDs), an ESP32-E-based Remote Terminal Unit (RTU), a GSM SIM800L module, and a Master Terminal Unit (MTU) set up on the Blynk IoT platform. The system utilizes both TCP/IP and HTTP communication protocols for two distinct configurations, enabling efficient data processing and facilitating human–machine interactions.

**User-Friendly Interface:** The primary Blynk IoT server platform handles data processing and user interactions efficiently, requiring minimal customer training for ongoing use. This stands in contrast to typical commercial SCADA systems, where complex MTU platforms necessitate extensive training and experience to operate effectively.

**Remote Monitoring, and Automated Control:** The required data are gathered locally through the RTU using FIDs and can be monitored and managed via a mobile app or web-based interface for efficient operation. Operators of the SCADA system can remotely deactivate the load as needed. Moreover, the load will be automatically deactivated when the voltage of the battery falls below a set value, ensuring battery safety.

**System Versatility:** In both proposed configurations, the RTU wirelessly connects to the MTU using the GSM SIM800L module and the integrated Wi-Fi module, eliminating the restriction on the physical locations of the RTU and MTU.

**System Availability and Reliability:** In the designed system, due to the use of open-source components and a locally installed, self-managed cloud server, system operators or administrators can easily ensure continuous reliability and availability. This stands in contrast to proprietary SCADA systems, where reliance on a single vendor may lead to delays in addressing customer concerns, as continuous on-site operator availability is often impractical.

**Open-Source System:** The system utilizes the open-source Blynk software (v1.2.0), accessible on any operating system without the need for licensing or annual fees, thereby eliminating ongoing operational expenses.

**Energy-Efficient and Cost-Effective System:** All components utilized in the designed system are sourced from various manufacturers, making them easily accessible and cost-effective. Each component's power consumption and cost breakdown are detailed in Table 4, outlining the specifics of price and energy usage for transparency.

**Table 4.** Power rating and price of components used in the proposed SCADA system.

| Sr. No. | Component | QTY | Power (Watt) | Total Price (CAD) |
|---|---|---|---|---|
| 1 | ACS712 Current Sensors | 2 | 0.128 | 4.84 |
| 2 | ZMPT101B Voltage Sensor | 1 | 0.025 | 1.38 |
| 3 | GSM SIM800L | 1 | 1.4 | 5 |
| 4 | 5 V Relay | 2 | 0.9 | 3.38 |
| 5 | LCD Display | 1 | 0.25 | 4.16 |
| 6 | ESP32-WROOM-32E | 1 | 0.528 | 7.76 |
| 7 | LM2596 | 2 | 0.48 | 3.58 |
| 8 | Miscellaneous (Zero PCB, jumper wires, resistors, etc.) | 1 | 0.28 | 10 |

## 8. Conclusions

High-pressure natural gas pipeline control stations are strategically located in remote, challenging terrains and increasingly rely on traditional and renewable energy sources, such as solar PV systems and wind turbines, due to unreliable power supply for smooth operations. Ensuring a reliable and consistent power supply along with a cost-efficient and advanced monitoring and control system is crucial for such control stations.

This research addresses the need for a low-power, cost-effective, and open-source SCADA system to enhance the efficient operation of remotely located natural gas pipeline control stations. The proposed solution utilizes an IoT-based SCADA architecture, the latest evolution in SCADA systems. The design includes the five essential elements of a SCADA system: Field Instrumentation Devices (voltage and current sensors), Remote Terminal Units (ESP32-WROOM-32E microcontroller), Master Terminal Units (Blynk IoT Server Platform), and SCADA Communication Channels (GSM SIM800L module and local Wi-Fi network). Two configurations were devised considering the potential lack of access to 3G, 4G, or Wi-Fi networks in such remote locations. The first configuration uses the GSM SIM800L module to communicate data from the RTU to the MTU (Blynk cloud server) via the TCP/IP protocol, while the second configuration employs the ESP32's integrated Wi-Fi module and a local Wi-Fi network using HTTP protocol.

The designed SCADA system's functionality was validated in a laboratory at the Core Science Building of Memorial University, NL, Canada. It successfully acquired and remotely monitored data from a 260 W, 12 V Solar PV System and performed necessary supervisory control activities. The SCADA system demonstrated capabilities in data acquisition, networked data communication, data presentation, remote monitoring, and supervisory control. The entire system operated with a low power consumption of 3.9 W, and incurred an overall cost of 40.1 CAD, making it an extremely cost-effective solution for such critical applications.

**Author Contributions:** Conceptualization, M.W.; Methodology, M.W.; Software, M.W.; Validation, M.W.; Resources, M.W.; Writing—original draft, M.W.; Writing—review & editing, M.J.; Supervision, M.J.; Project administration, M.J.; Funding acquisition, M.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The study incorporates its original findings within the article, and any additional questions can be directed to the corresponding authors.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript.

| | |
|---|---|
| FIDs | Field Instrument Devices |
| GSM | Global System for Mobile Communications |
| HPS | Hybrid Power System |
| HTTP | Hypertext Transfer Protocol |
| IoT | Internet of Things |
| LCD | Liquid Crystal Display |
| MTU | Master Terminal Units |
| PV | Photo Voltaic |
| PLCs | Programmable Logic Controllers |
| RTU | Remote Terminal Unit |
| RESs | Renewable energy sources |
| SCADA | Supervisory Control and Data Acquisition |
| UI | User Interface |

## References

1. Vargas Salgado, C.; Águila-León, J.; Chiñas-Palacios, C.D.; Hurtado-Pérez, E. Design and Deployment of a Web SCADA for an Experimental Microgrid. In Proceedings of the Proceedings INNODOCT/20. International Conference on Innovation, Documentation and Education, Valencia, Spain, 11–16 November 2020. [CrossRef]
2. Meng, D.; Yang, H.; Yang, S.; Zhang, Y.; Jesus, C.J.; Tiago Fazeres-Ferradosa, M.W.; Branco, R.; Zhu, S.-P. Kriging-assisted hybrid reliability design and optimization of offshore wind turbine support structure based on a portfolio allocation strategy. *Ocean. Eng.* **2024**, *295*, 116842. [CrossRef]
3. Meng, D.; Yang, S.; Yang, H.; Jesus Abílio, M.P.; Correia, J.; Zhu, S.-P. Intelligent-inspired framework for fatigue reliability evaluation of offshore wind turbine support structures under hybrid uncertainty. *Ocean. Eng.* **2024**, *307*, 118213. [CrossRef]
4. Waqas, M.; Jamil, M.; Khan, A.A. Hybrid Power System Design and Dynamic Modeling for Enhanced Reliability in Remote Natural Gas Pipeline Control Stations. *Energies* **2024**, *17*, 1763. [CrossRef]
5. Aghenta, L.O.; Iqbal, M.T. Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard and MQTT protocol. *AIMS Electron. Electr. Eng.* **2020**, *4*, 57–86. [CrossRef]
6. Kermani, M.; Adelmanesh, B.; Shirdare, E.; Sima, C.A.; Carnì, D.L.; Martirano, L. Intelligent energy management based on SCADA system in a real Microgrid for smart building applications. *Renew. Energy* **2021**, *171*, 1115–1127. [CrossRef]
7. De Arquer Fernández, P.; Fernández, M.A.F.; Candás, J.L.C.; Arboleya, P.A. An IoT open source platform for photovoltaic plants supervision. *Int. J. Electr. Power Energy Syst.* **2020**, *125*, 106540. [CrossRef]
8. Baig, M.J.A.; Iqbal, M.T.; Jamil, M.; Khan, J. Blockchain-Based Peer-to-Peer Energy Trading System Using Open-Source Angular Framework and Hypertext Transfer Protocol. *Electronics* **2023**, *12*, 287. [CrossRef]
9. Soetedjo, A.; Hendriarianti, E. Development of an IoT-Based SCADA System for Monitoring of Plant Leaf Temperature and Air and Soil Parameters. *Appl. Sci.* **2023**, *13*, 11294. [CrossRef]
10. What Is SCADA? *(Supervisory Control and Data Acquisition)—RealPars.* Available online: https://www.realpars.com/blog/scada (accessed on 11 June 2024).
11. Medrano, K.; Altuve, D.; Belloso, K.; Bran, C. Development of SCADA using a RTU based on IoT controller. In Proceedings of the 2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA), Concepcion, Chile, 17–19 October 2018; pp. 1–6. [CrossRef]

12. Aghenta, L.O.; Iqbal, M.T. Development of an IoT Based Open Source SCADA System for PV System Monitoring. In Proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 5–8 May 2019; pp. 1–4. [CrossRef]
13. Yi, M.; Mueller, H.; Yu, L.; Chuan, J. Benchmarking Cloud-Based SCADA System. In Proceedings of the 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Hong Kong, China, 11–14 December 2017; pp. 122–129. [CrossRef]
14. Sethi, P.; Sarangi, S.R. Internet of Things: Architectures, Protocols, and Applications. *J. Electr. Comput. Eng.* **2017**, *2017*, 9324035. [CrossRef]
15. Moustafa, N.; Turnbull, B.; Choo, K.K.R. An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. *IEEE Internet Things J.* **2018**, *6*, 4815–4830. [CrossRef]
16. Sultana, T.; Wahid, K.A. Choice of Application Layer Protocols for Next Generation Video Surveillance Using Internet of Video Things. *IEEE Access* **2019**, *7*, 41607–41624. [CrossRef]
17. Deveci, M.; Cali, U.; Pamucar, D. Evaluation of criteria for site selection of solar photovoltaic (PV) projects using fuzzy logarithmic additive estimation of weight coefficients. *Energy Rep.* **2021**, *7*, 8805–8824. [CrossRef]
18. Ahmed, O.; Iqbal, M.T. Remote Monitoring, Control and Data Visualization for a Solar Water Pumping System. *Eur. J. Electr. Eng. Comput. Sci.* **2023**, *7*, 71–77. [CrossRef]
19. Matsui, T.; Yamamoto, K.; Sumi, S.; Triruttanapiruk, N. Detection of Lightning Damage on Wind Turbine Blades Using the SCADA System. *IEEE Trans. Power Deliv.* **2020**, *36*, 777–784. [CrossRef]
20. Abdulsalam, A.B.; Alsaadi, H.A.J.; Hamodat, Z. Control And Management of Solar PV Grid Using Scada System. In Proceedings of the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 9–11 June 2022; pp. 1–5. [CrossRef]
21. Uddin, S.U.; Baig, M.J.A.; Iqbal, M.T. Design and Implementation of an Open-Source SCADA System for a Community Solar-Powered Reverse Osmosis System. *Sensors* **2022**, *22*, 9631. [CrossRef]
22. He, W.; Baig, M.J.A.; Iqbal, M.T. An Open-Source Supervisory Control and Data Acquisition Architecture for Photovoltaic System Monitoring Using ESP32, Banana Pi M4, and Node-RED. *Energies* **2024**, *17*, 2295. [CrossRef]
23. Vargas-Salgado, C.; Aguila-Leon, J.; Chiñas-Palacios, C.; Hurtado-Perez, E. Low-cost web-based Supervisory Control and Data Acquisition system for a microgrid testbed: A case study in design and implementation for academic and research applications. *Heliyon* **2019**, *5*, e02474. [CrossRef]
24. Osman, F.A.; Hashem, M.Y.M.; Eltokhy, M.A.R. Secured cloud SCADA system implementation for industrial applications. *Multimed. Tools Appl.* **2022**, *81*, 9989–10005. [CrossRef]
25. Aghenta, L.O.; Iqbal, M.T. Low-Cost, Open Source IoT-Based SCADA System Design Using Thinger.IO and ESP32 Thing. *Electronics* **2019**, *8*, 822. [CrossRef]
26. Ahsan, L.; Baig, M.J.A.; Iqbal, M.T. Low-Cost, Open-Source, Emoncms-Based SCADA System for a Large Grid-Connected PV System. *Sensors* **2022**, *22*, 6733. [CrossRef]
27. Omidi, S.A.; Baig, M.J.A.; Iqbal, M.T. Design and Implementation of Node-Red Based Open-Source SCADA Architecture for a Hybrid Power System. *Energies* **2023**, *16*, 2092. [CrossRef]
28. Osaretin, C.A.; Zamanlou, M.; Iqbal, M.T.; Butt, S. Open Source IoT-Based SCADA System for Remote Oil Facilities Using Node-RED and Arduino Microcontrollers. In Proceedings of the 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 4–7 November 2020; pp. 0571–0575. [CrossRef]
29. Moshayedi, A.J.; Roy, A.S.; Liao, L.; Li, S. Raspberry Pi SCADA Zonal based System for Agricultural Plant Monitoring. In Proceedings of the 2019 6th International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 20–22 December 2019; pp. 427–433. [CrossRef]
30. Fazlollahtabar, H. Internet of Things-based SCADA system for configuring/reconfiguring an autonomous assembly process. *Robotica* **2022**, *40*, 672–689. [CrossRef]
31. Tom, R.J.; Sankaranarayanan, S. IoT based SCADA integrated with Fog for power distribution automation. In Proceedings of the 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), Lisbon, Portugal, 21–24 June 2017; pp. 1–4. [CrossRef]
32. Stanculescu, M.; Badea, C.A.; Marinescu, I.; Andrei, P.; Drosu, O.; Andrei, H. Vulnerability of SCADA and Security Solutions for a Wastewater Treatment Plant. In Proceedings of the 2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, 28–30 March 2019; pp. 1–6. [CrossRef]
33. Noor, N.I.M.; Nasir, N.N.M. Traffic Light Control System by Using Supervisory Control and Data Acquisition (SCADA), Programmable Logic Controller (PLC) Technology and Image Processing. *J. Electr. Power Electron. Syst.* **2020**, *2*. Available online: https://fazpublishing.com/jepes/index.php/jepes/article/view/35 (accessed on 10 August 2024).
34. Kumar, K.S.; Kumar, R.J.V.A.; Mirza, Y.Y. Automation Control of Antenna and Microwave in Telecom Towers using SCADA. *Int. J. Mod. Trends Sci. Technol.* **2022**, *8*, 27–33. [CrossRef]
35. Qays, M.O.; Ahmed, M.M.; Parvez Mahmud, M.A.; Abu-Siada, A.; Muyeen, S.M.; Hossain, M.L.; Yasmin, F.; Rahman, M.M. Monitoring of renewable energy systems by IoT-aided SCADA system. *Energy Sci. Eng.* **2022**, *10*, 1874–1885. [CrossRef]
36. Blynk: A Low-Code IoT Software Platform for Businesses and Developers. Available online: https://blynk.io (accessed on 15 June 2024).

37. Hall-Effect-Based Linear Current Sensor IC—ACS712 | Allegro MicroSystems. Available online: https://www.allegromicro.com/en/products/sense/current-sensor-ics/zero-to-fifty-amp-integrated-conductor-sensor-ics/acs712 (accessed on 22 June 2024).
38. Laisina, L.H. Leak Current Detector in Low Voltage Network (Jtr) System Using Zmpt101b Sensor Based on the Internet of Things (IoT). *Int. J. Multidiscip. Sci. Arts* **2023**, *2*, 48–54. [CrossRef]
39. ZMPT101B AC Single Phase Voltage Sensor Module, VISHA WORLD. Available online: https://vishaworld.com/products/zmpt101b-ac-single-phase-voltage-sensor-module (accessed on 22 June 2024).
40. Oton, C.N.; Iqbal, M.T. Low-Cost Open Source IoT-Based SCADA System for a BTS Site Using ESP32 and Arduino IoT Cloud. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 1–4 December 2021; pp. 0681–0685. [CrossRef]
41. ESP32-DevKitC V4 Getting Started Guide—ESP32—ESP-IDF Programming Guide v5.2.2 documentation. Available online: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/hw-reference/esp32/get-started-devkitc.html (accessed on 22 June 2024).
42. Kanani, P.; Padole, M. Real-time Location Tracker for Critical Health Patient using Arduino, GPS Neo6m and GSM Sim800L in Health Care. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; pp. 242–249. [CrossRef]
43. SIM800L-Datasheet.pdf. Available online: https://components101.com/wireless/sim800l-gsm-module-pinout-datasheet-equivalent-circuit-specs (accessed on 24 June 2024).
44. Fang, X.; Li, X. Design and Performance Analysis of Photovoltaic Power Generation Light Emitting Diode Device. *J. Nanoelectron. Optoelectron.* **2024**, *19*, 605–612. [CrossRef]
45. 16x2 LCD Module. Components101. Available online: https://components101.com/displays/16x2-lcd-pinout-datasheet (accessed on 24 June 2024).
46. Hameed, A.A.; Sultan, A.J.; Bonneya, M.F. Design and Implementation a New Real Time Overcurrent Relay Based on Arduino. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *881*, 012142. [CrossRef]
47. Krichen, M.; Lahami, M.; Cheikhrouhou, O.; Alroobaea, R.; Maâlej, A.J. Security Testing of Internet of Things for Smart City Applications: A Formal Approach. In *Smart Infrastructure and Applications*; Springer: Cham, Switzerland, 2019; pp. 629–653. [CrossRef]
48. EasyEDA—Online PCB Design & Circuit Simulator. Available online: https://easyeda.com/ (accessed on 25 June 2024).