Journal of Electronics and Electrical Engineering
https://ojs.wiserpub.com/index.php/JEEE

UNIVERSAL WISER PUBLISHER

Research Article

Design and Implementation of SCADA Architecture Based on MATLAB App Designer for a Hybrid Power System

Azadeh Farhadi Saransari, Farshad Dehghanisanij, M. Tariq Iqbal*

Department of Electrical and Computer Engineering, Memorial University of Newfoundland, St. John's, NL, Canada
E-mail: tariq@mun.ca

Received: 6 July 2024; **Revised:** 28 October 2024; **Accepted:** 28 October 2024

Abstract: Hybrid renewable power systems (HRPS) are now considered reliable solutions for power generation under various conditions. A critical challenge for deploying HRPS involves the design and implementation of an effective supervisory control and data acquisition (SCADA) system, which is essential for real-time monitoring and control. The SCADA system plays a vital role in remote monitoring and control by enabling real-time data management, early fault detection, and timely troubleshooting. This study presents the design and implementation of a SCADA architecture based on a MATLAB App Designer for an HRPS consisting of wind turbines, photovoltaic panels, diesel generators, and batteries. The system utilizes Arduino Mega 2560 as remote terminal units (RTUs) to interface with actuators and measure critical system parameters such as voltage and current. Arduino boards can be controlled directly from MATLAB® via the MATLAB® Support Package for Arduino® Hardware. A laptop is the main terminal unit (MTU), communicating with the Arduino via the Firmata protocol. MATLAB App Designer is the central monitoring interface, providing real-time data acquisition, processing, and visualization. In addition, a relay module is used to control the operation of the diesel generator. The relay module acts as an intermediary between the control system and the diesel generator, allowing for automated control of the generator's operation. It receives commands from the SCADA system to either start or stop the generator based on the energy demands and the availability of renewable resources. By managing the diesel generator's activity, the relay module helps ensure that the HRPS maintains a balanced and efficient power supply, minimizing reliance on non-renewable sources when renewable energy is sufficient. Based on experimental results, the proposed SCADA system effectively monitors and controls HRPS under different conditions.

***Keywords*:** photovoltaic (PV), renewable energy, instrumentation and control, SCADA system, hybrid power systems

1. Introduction

Renewable energy resources have been increasingly significant in electricity generation in the last decade. Clean energy sources like wind, hydro, and solar are integrated with conventional generators to develop highly efficient power systems. Even so, battery storage is essential for hybrid and sustainable energy systems due to their intermittent nature and significant dependence on climate-related factors [1]. Several studies have focused on optimizing battery storage, which is a key component of hybrid power systems [2, 3]. This hybrid approach is more beneficial in rural and remote areas where stable electricity grid access is limited or unreliable. A hybrid power system consists of various components, including

DOI: https://doi.org/10.37256/jeee.3220245261

field instrumentation devices (FDI), microcontrollers, and actuators to record power, voltage, and current for real-time data monitoring and remote and synchronized control. SCADA system is an appropriate solution for these tasks [4].

A SCADA system ensures real-time management, early fault detection, and prompt corrective actions, making hybrid power more efficient and longer-lasting. Accordingly, it has become a core part of industrial automation in the last few decades [5]. A SCADA system is composed of both hardware and software elements. The software comprises a human-machine interface (HMI), a central database (Historian), and various user applications. Hardware includes remote terminal units (RTUs), master terminal units (MTUs), actuators, and sensors. These software components facilitate communication between hardware and software. The RTU gathers sensor data and sends it to the MTU, enabling the SCADA system to monitor and manage the environment effectively [6]. SCADA systems are used in various industries, including industry [7, 8], smart buildings [9], electric power distribution [10], oil production [11], etc. Early versions of SCADA were introduced in the 1950s for minicomputers and telephone relay systems [12]. The evolution of SCADA systems is divided into four generations, each reflecting advancements in communication and computing technologies [5, 13].

- Monolithic SCADA Systems: A first-generation system using large minicomputers such as the PDP-11 series of digital equipment corporations. These systems had no connectivity, and MTUs and RTUs communicated exclusively over WAN protocols. As a result of this isolation, they could not interact with systems from different vendors, underscoring the importance of open standards.

- Distributed SCADA Systems: In the second generation of LANs, computation tasks were distributed across multiple systems. Some systems' roles are to process communications, while others manage interfaces or databases for operators. Despite offering improved redundancy and reliability, these systems used proprietary LAN protocols, which restricted connectivity beyond the local area.

- Networked SCADA Systems: The third generation introduced standard protocols such as IP that enabled WAN connectivity. The integration of third-party devices into SCADA networks was facilitated by open standards, which improved disaster resilience. The development of networked SCADA systems allowed geographically dispersed deployments and greatly improved interoperability.

- IoT-based SCADA Systems: Fourth-generation SCADA systems integrating IoT technologies and cloud computing are revolutionizing industrial automation. IoT-based SCADA systems on the market today use cloud storage to collect and analyze data, offering scalability, interoperability, and advanced analytics.

Through these advancements, SCADA systems have evolved from isolated, proprietary architectures to interconnected, data-intensive platforms capable of supporting complex industrial processes. Standards like OPC UA have standardized communication protocols, enabling seamless integration across diverse industrial automation devices and systems. The progression towards IoT-based SCADA systems underscores a shift towards more efficient, adaptable, and interconnected manufacturing operations in the digital age.

This study presents a SCADA architecture that utilizes MATLAB App Designer to monitor a hybrid renewable power system (HRPS) with solar panels, a diesel generator, and batteries. System sizing and dynamic modeling are described in [14, 15], respectively. Diesel generators are used as backup devices and controlled by relay modules. Whenever wind and solar power are insufficient, a diesel generator is required. The diesel generator is switched off when either source becomes available. Each component's analog current and voltage data is collected by five current sensors and one voltage sensor. These modules are connected to an RTU specified as the Arduino Mega 2560. The data gathered by the RTU is transmitted to the MTU using the Firmata protocol. This is done via a serial port enabled by the MATLAB support package for Arduino. The MATLAB® Support Package for Arduino® Hardware provides a seamless interface for communicating with Arduino boards directly from MATLAB. This integration allows users to read sensor data and send commands to Arduino without code compilation. A personal computer functions as the MTU to facilitate interaction between the RTU and the monitoring and analyzing unit. MATLAB App Designer is utilized as a monitoring and analysis tool to receive real-time data and process, analyze, and display it.

This paper's remainder is structured as follows: Section 2 provides an overview of the related literature, examining previous research and identifying gaps. Sections 3 and 4 describe the system structure and components, giving a comprehensive overview of the SCADA system architecture and essential elements. Section 5 examines developing and integrating system components, including the specific procedures and techniques used for implementation. Section 6 describes the experimental setup and Section 7 examines testing results, demonstrating the performance and capabilities of the developed SCADA system. Section 8 summarizes the critical contributions and outcomes of the study.

2. Literature review

Research communities worldwide have developed and enhanced various SCADA applications.

Asgher et al. [16] described a cost-effective and open-source SCADA system that utilized IoT. The hybrid power system of an offshore aquaculture site was monitored remotely from 2 km from the coast, where there were no traditional electrical or communications infrastructures. Six field sensors, one Arduino Leonardo as an RTU, and a LoRA gateway were used in the hardware. In addition, Arduino IDE, AWS, InfluxDB, and Grafana provided software support. Thus, solar power, battery status, inverter and generator currents, battery voltage, and temperature data were measured and transmitted through these software and hardware components. The data were sent to the RTU and transmitted to The Things Network's (TTN) cloud. Data were stored in InfluxDB, visualized using AWS services, and analyzed by Grafana. The graphs helped monitor each sensor's historical and live data and allowed alarms to be set for user-defined conditions. This low-cost SCADA system demonstrated the potential for providing an efficient remote monitoring solution for offshore aquaculture sites, offering an affordable alternative to proprietary SCADA systems for small and large businesses in the Canadian aquaculture industry.

Qays et al. [17] improved renewable energy asset management through IoT-enabled SCADA systems. The system monitored battery storage, wind, and photovoltaic hybrid renewable energy systems using real-time monitoring. The ThingSpeak website tracked key electrical parameters, including power, current, and voltage, enabling remote control of system components. The system was effective and economical because SCADA was integrated with MATLAB/SIMULINK via the KEP Server EX client. It also used low-cost components like the Arduino ATMega2560. In addition to highlighting promising improvements, the researchers discussed ways to improve network security for IoT communications, making the system more reliable and scalable.

He et al. [18] provided an open-source SCADA system that was IoT-based and specifically designed for monitoring PV systems. It used an ESP32-E microcontroller for data collection, transmitting it to a Banana Pi M4 Berry that served as the MTU. This configuration used Node-RED dashboards to control and monitor real-time data. Data were collected comprehensively from the proposed component with three current sensors and three voltage sensors. Furthermore, the load was monitored with a relay mechanism and an automatic shutdown mechanism to prevent battery over-discharge. This SCADA system's scalability and cost-effectiveness were enhanced by its exploitation of open-source components and platforms. The performance of the proposed architecture was validated under laboratory conditions, demonstrating its potential as a monitoring and control system for PV systems.

Li et al. [19] focused on identifying equipment faults and cyberattacks in SCADA systems, including replay attacks (RA). To develop a data-driven methodology, they integrated a state-space modeling approach with Kalman filters (KF) and linear-quadratic Gaussian controllers (LQG). The method detected and differentiated RAs from various equipment faults, such as controller, plant, sensor, and plant degradation. This was based on changes in the mean shifts and covariance structures of residuals and sensors. Their approach was validated through extensive simulations and physical experiments on rotating machinery, demonstrating its application to real-world situations.

Abdulsalam et al. [20] created a SCADA system that enabled remote monitoring and control of grid-connected inverters, featuring an IoT-based server to facilitate automated management. This system allowed utility providers to stabilize the grid, improve power quality, and optimize economic performance. It did this by adjusting to energy pricing and renewable generation variability. Emphasizing the need for cost-effective and open-source SCADA solutions, the study highlighted the importance of modern SCADA components like MTUs, RTUs, and HMIs for effective energy management.

The SCADA system successfully addressed PV fluctuation challenges, storage system scalability, and grid integration under various climatic conditions.

Gnana et al. [21] narrated the importance of SCADA and PLC Systems for automating the dynamic processes in power systems, water treatment plants, critical structures, etc. Data was collected, stored, and monitored using SCADA systems, while automatic production processes were controlled by PLCs. It also highlighted the detection of faults in wind turbines and the management of electric power facilities. In particular, SCADA systems enhanced the reliability of management information systems, their monitoring in real-time, and the management of resources, making it possible in areas such as energy, transportation, and water treatment processes.

Farooq Saber et al. [22] described an innovative control system for an oil refinery based on SCADA integrating LabVIEW alongside Arduino. This study investigated SCADA capabilities distinct from traditional distributed control systems (DCS) mainly utilized within the oil and gas industry. The aim was to develop a SCADA system for tracking and regulating oil levels in a tank. A refinery is simulated through this exercise. The system used red switches that were controlled, programmed, and interfaced with an Arduino UNO microcontroller using LabVIEW to monitor and control oil levels within the tank. Emphasis was put on improving effectiveness, reducing the need for human involvement, and enhancing system performance at lower operational costs.

Ferlito et al. [23] proposed an innovative SCADA system based on IoT for solar PV utility plants. This system met the requirements of improved operational and maintenance efficiency regarding reducing data transmission costs and adopting low-weight protocols. Such microservices architecture involved deployments, including edge computing and blockchain, ensuring scalability, fault tolerance, and security for data certification. Ferlito underlined the necessity for such integration as cloud and edge resource management for current-state monitoring, maintenance forecasting, and energy delivery at an optimal value, which intends to balance system speed and expansion abilities.

Toma et al. [24] focused on re-configuring the GUI after installing peripheral equipment such as PLCs and RTUs to increase system functionality and versatility. Their publication described the organization of SCADA systems, including the master-slave two-tier architecture, and further discussed the importance of PLCs in the real-time operation and control of the system. RTUs were portrayed as essential data aggregation equipment and the communication center that connects the substations and the upper-tier SCADA master stations. With the Zenon Energy Edition software tool kit, further impressions were made of how to interconnect and reposition signals on the GUI. This was done to enhance efficiency in a SCADA system within a Romanian substation. The research emphasized the importance of alterations to the GUI, mainly due to Hardware changes, which improve system monitoring and efficiency.

Ergashev et al. [25] presented the design of a modern SCADA system with IoT integration for automation and optimization of greenhouse operations. This was to grow vegetables in cities. The system provided real-time monitoring and control of critical environmental factors such as temperature, humidity, and water levels, using hydroponic methods that eliminated soil. This system utilized IoT sensors and RTUs to collect data accurately. It enabled remote control through cloud-based platforms to further enhance urban agriculture efficiency and sustainability. Limited space and poor natural conditions enabled vegetable production.

Muljanto et al. [26] implemented a system at ITN Malang that allowed real-time electrical parameters acquisition such as voltage, current, and energy consumption. In the monitoring system, measuring devices such as SPM 91 and PZEM-017, which supported Modbus communication protocols, give high accuracy in measuring solar plant performance. This information was presented through a SCADA dashboard, which offers real-time access to system conditions. A monitoring system increases efficiency by allowing supervision from a distance and data logging with precision. This improved the management and performance analysis of that system.

According to this comprehensive literature review and current knowledge, no existing SCADA system is designed for hybrid power systems consisting of wind turbines, PV panels, batteries, and a diesel generator. Furthermore, there is a noticeable absence of SCADA systems utilizing MATLAB App Designer for real-time data monitoring and analysis. Additionally, unlike open-source platforms, MATLAB provides a comprehensive and integrated environment for data analysis, visualization, and algorithm development. This integration simplifies the implementation of complex control strategies and enhances the system's responsiveness to dynamic changes in renewable energy generation and consumption. MATLAB offers an integrated platform where algorithms (developed using MATLAB scripts/functions) and user interfaces (created with App Designer) can be designed concurrently. This streamlined integration accelerates development and debugging processes.

Moreover, a significant feature of the SCADA system is its relay mechanism. This was designed to automatically activate the diesel generator when primary power sources (wind and solar) are insufficient to meet demand. This ensures uninterrupted electricity supply, particularly during low renewable energy generation periods or increased power consumption. Automated response enhances system reliability and minimizes downtime, which is crucial for remote or off-grid installations.

3. System description

The SCADA system designed to monitor and control the HRPS is illustrated in Figure 1. This HRPS system comprises various components, including a wind turbine, a photovoltaic panel, a diesel generator, a battery, and an inverter. The SCADA configuration is equipped with five current sensors and one voltage sensor to monitor the output voltage and current of the HRPS. These sensors monitor parameters such as DC bus voltage, wind turbine current, PV panel current, diesel generator current, battery, and inverter current. The PV is also linked to a maximum power point tracker (MPPT) to optimize power output, and one PV current sensor is connected to the MPPT output. Additionally, the diesel generator functions as a backup power source, activated based on power availability through a relay-generated control signal. The power generated by the wind turbine, PV panel, and diesel generator is stored in a battery bank, and DC to AC voltage conversion is achieved using an inverter. All sensors send their readings to an Arduino Mega 2560. The Arduino collects and transfers these values to the MTU via a Firmata protocol serial port. The main computer used in this investigation ran on Microsoft Windows 11 Enterprise, installed on a Core (TM) i7-8565U CPU. This system interfaces with MATLAB/Simulink software for data processing, supported by MATLAB App Designer for dashboard monitoring. A CSV file is created and stored with all collected values to analyze the data.

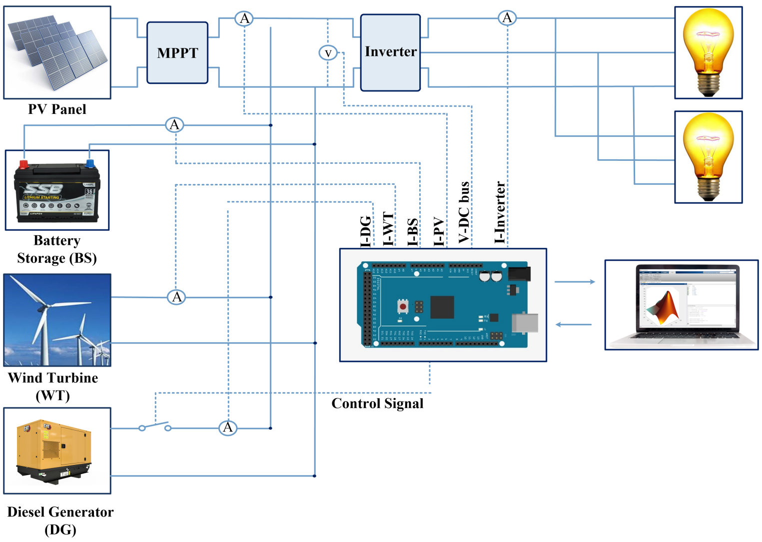

Figure 1. Implemented SCADA system for proposed hybrid renewable energy setup

4. System component

This section provides a comprehensive overview of the hardware details of each component and the software utilized in the development of the SCADA system. The Arduino Mega 2560 microcontroller gathers data from field instruments, including current, voltage, and relay sensors. Subsequently, MATLAB processes the acquired data and transmits it to the MATLAB App Designer for display on the computer screen.

4.1 *Remote terminal units*

In this study, the Arduino Mega 2560, illustrated in Figure 2, serves as a remote terminal unit. It is a powerful microcontroller board featuring the ATmega2560. The board includes 4 UARTs for hardware serial communication,16 analog inputs, and 54 digital I/O pins. This device runs at 16 MHz with a crystal oscillator and features a USB port, power connector, ICSP header, and reset button [27]. It is compatible with most Arduino Uno shields and can be powered via an external power supply or USB. The board is programmed using the Arduino IDE, allowing easy coding and uploading.

The ATmega2560 is equipped with a bootloader, facilitating straightforward code uploads without external programming hardware. It supports serial (TTL), TWI (I2C), and SPI communication protocols, making it suitable for complex applications. The automatic software reset feature simplifies programming and debugging [27]. To achieve a cost-efficient design, affordable electronic components, and the Arduino integrated development environment (IDE) ATMega2560 remote terminal unit were utilized to develop a hardware prototype for experimental analysis.

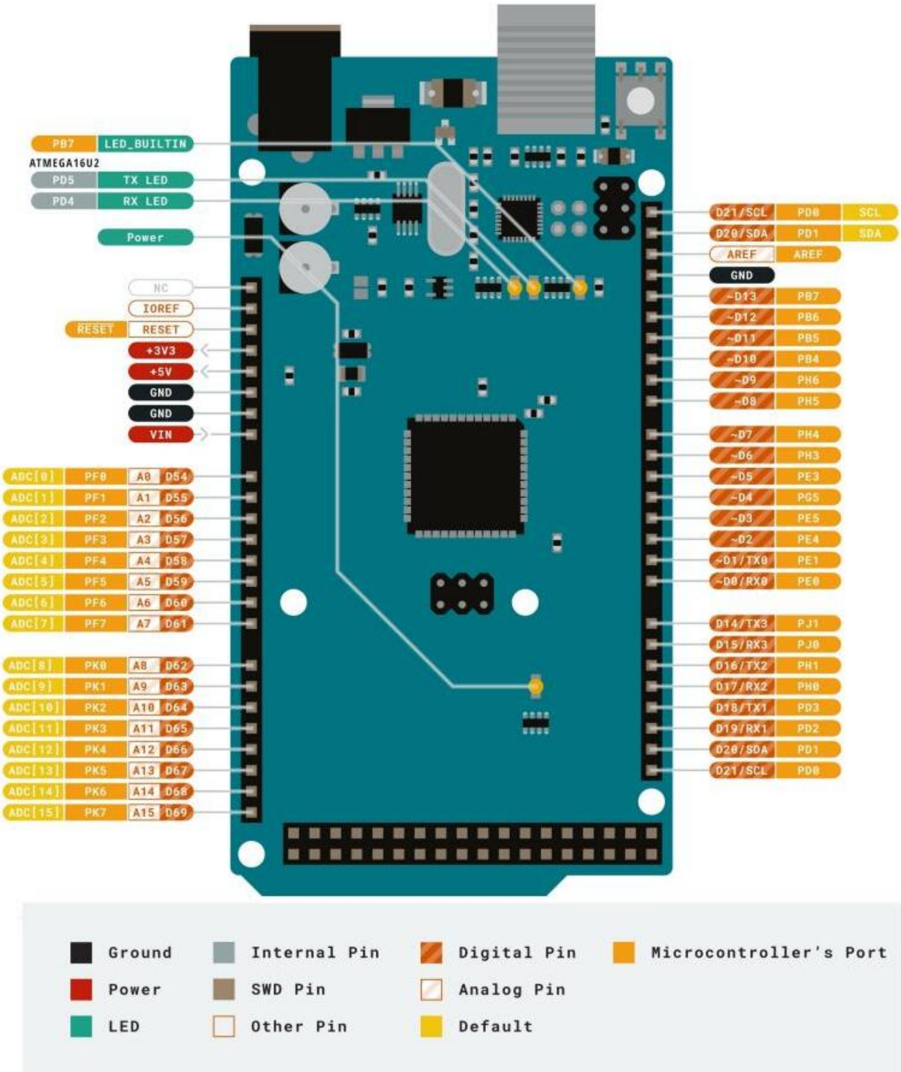

Figure 2. Pinout diagram of mega 2560 [27]

4.2 *Sensors*

4.2.1 *ACS712 current sensor*

The ACS712 is a fully integrated, Hall Effect-based linear current sensor designed for precise current measurement. It operates on the principle of the Hall Effect, detecting the magnetic field generated by current flow through a conductor. This sensor features a linear relationship between the current and output voltage, making it straightforward to interpret the readings. AllegroMicroSystems, LLC makes this ACS712 which can measure AC and DC currents. It functions across three amperage settings: 5 A, 20 A, and 30 A, with a sensitivity ranging between 66 to 185 (mV/A). This study uses the ACS712 at 5 A and 185 (mV/A) of sensitivity to measure both AC and DC currents from a single power source of 5 V.

A voltage divider circuit is employed to adjust the 5 V signal the current sensor needs to the 3.3 V signal level the microcontroller pins can handle for the physical connections. Figure 2 shows the schematic of the ACS712 current sensor. The V_{out} signal changes linearly with the measured unidirectional or bidirectional AC or DC primary current (IP) within the specified range. The capacitor C_f is advised for noise reduction, with its value depending on the specific application [28]. Figure 3 shows the current sensor of the module ACS 712 and its schematic.

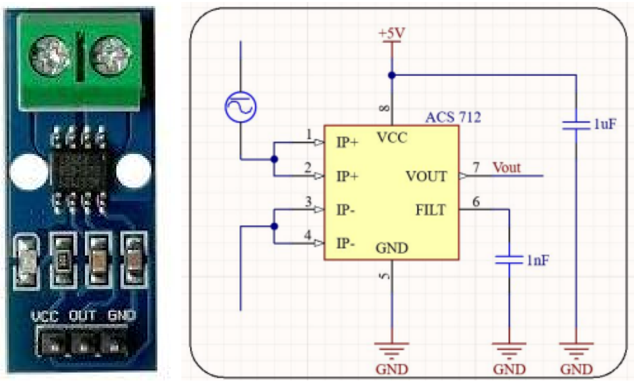

Figure 3. Modul ACS712—current sensor (5 A) [29]

4.2.2 *Voltage sensor*

The F031-06 voltage sensor is an exact and dependable device for recording DC voltages ranging from 0 to 25 V. It uses a resistive voltage divider strategy to ensure accurate readings. The sensor connects to the voltage source via V_{cc} terminals and features two resistors in series with specifications of 30 K Ohms and 7.5 K Ohms. The division ratio between these resistors is 1:5, meaning the actual voltage is one-fifth of the ESP32-E detects. As a result, precision voltage measurements require programming adjustments. Figure 4 shows the F031-06 voltage sensor and its schematic.

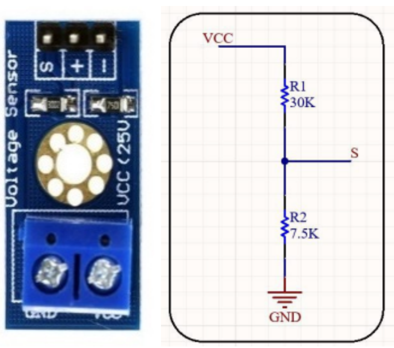

Figure 4. Modul F031-06—voltage sensor (0–25 V) [30]

4.3 *Switching device*

The SRD-05VDC-SL-C Channel Relay Module (also known as Songle) and its schematic illustrated in Figure 5. It is used to regulate the operation of a diesel generator. This relay module is widely employed in Arduino and other microcontroller projects. It is designed to switch high voltage (up to 250 V AC) or high current devices using a low-voltage control signal (typically 5 V DC from Arduino GPIO pins). Within the relay, a switch is linked to an electromagnet. When a HIGH signal is sent to the signal pin, the electromagnet is activated, causing the switch's contacts to open or close [31].

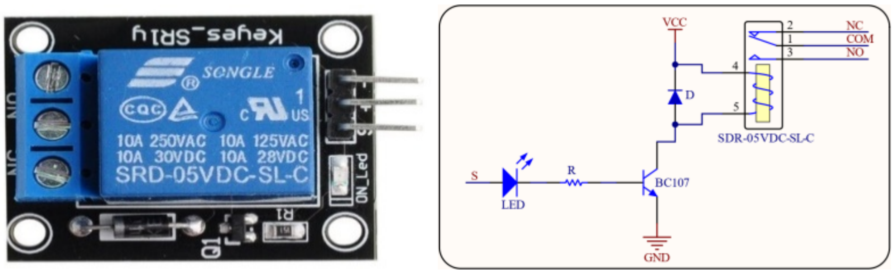

Figure 5. The SRD-05VDC-SL-C relay module [32]

4.4 *Core terminal device*

The main PC used for this research is an Asus Vivobook S13, featuring the following specifications:

- Processor: Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz (boost up to 1.99 GHz), providing robust performance for running computationally intensive applications and simulations.
- System Type: 64-bit Windows 11 Enterprise, Version 22H2, ensuring compatibility with the latest software and security updates.
- RAM: 8.00 GB (7.82 GB usable), sufficient for handling data processing tasks and running multiple applications simultaneously.

This core terminal device plays a crucial role in the research, providing the necessary computational power and resources for the SCADA system's development, testing, and deployment. It ensures efficient handling of data analysis, visualization, and real-time monitoring tasks required for managing the hybrid renewable power system (HRPS).

4.5 *Software tool*

To build the SCADA system, MATLAB, including MATLAB App Designer, is used on Windows 11 Enterprise (Version 22H2). App Designer enables the creation of intuitive user interfaces using MATLAB, while MATLAB provides an integrated environment for analyzing data, visualizing algorithms, and developing algorithms. Combining these approaches simplifies the implementation of complex control strategies and enhances the system's ability to respond to changing renewable energy consumption and generation. Providing a reliable platform for developing and deploying SCADA systems, the robust setup ensures real-time monitoring and control.

5. Implementation methodology

To implement the proposed SCADA system one voltage sensor and five current sensors are interfaced with an Arduino Mega 2560, serving as RTU in the system design. The voltage sensors are connected in parallel with the DC bus, while the current sensors are connected in series with the PV panel, wind turbine, diesel generator, battery, and inverter. Additionally, the relay utilized in this study is connected in series with the diesel generator. All sensors receive power from the Arduino Mega 2560's VCC pin and are grounded through the Arduino ground pin. Table 1 provides details on the pin configuration of the RTU.

Table 1. The Arduino interconnections with sensors

System	Specification	Analog/Digital	PIN #
1	Relay	Digital	24
2	DC bus voltage sensor	Analog	A0
3	Inverter current sensor	Analog	A1
4	PV panel current sensor	Analog	A2
5	Wind turbine current sensor	Analog	A3
6	Battery storage current sensor	Analog	A4
7	Diesel generator current sensor	Analog	A5

The data gathered by the ATMEGA 2560 is sent to the MTU using the Firmata protocol, facilitated by the MATLAB Support Package for Arduino hardware, which is designed for microcontroller interaction. These data are being forwarded to the MATLAB App Designer for monitoring and control purposes. Additionally, all data is stored in a CSV file. The overview of the algorithm implemented on the Atmega 2025 using MATLAB software (Version 2022) is presented in Table 2.

Table 2. Data acquisition and control system algorithm

```
Initialization;
1. Establish the connection between MATLAB and the Arduino board
2. read sensor values (DC bus voltage sensor, inverter current sensor, PV panel current sensor, wind turbine current sensor, battery storage current
sensor, diesel generator current sensor, and relay) from analog and digital pins A0, A1, A2, A3, A4, and A5 and 49 by Arduino
3. Launch App Designer and loading the .mlapp file
  While data is being received from Arduino
        4. Collect measurements and update App Designer UI
        5. Calculate and calibrate the values of data
        6. Display all sensor values on the dashboard and for data visualization;
        7. Write the current data in Excel file row by row
    if PV and wind turbine current decrease by a certain value
        8. Turn the generator on (the green LED);
    else
        9. Turn the generator off (the Red LED);
    end
  end
  if Arduino does not send any data, then
    10. Print 'Data has not been received', reconnect in 5 s;
  else
    11. Go to step 3.
  end
```

6. Experimental configuration

A schematic illustration of the experimental setup for the proposed SCADA system is presented in Figure 6. The setup consists of two primary components: a demo solar/wind energy training system (LabVolt Series) manufactured by Festo [33] and the hardware configuration assembled to implement the SCADA system. The training system uses industrial parts to integrate solar and wind energy with battery storage. This study serves as an experimental setup for the author's previous research on designing a hybrid renewable power system for Postville [14, 15]. The proposed hybrid renewable power system included wind, PV, battery, and a diesel generator as a backup device. However, the current training system lacks a diesel generator, so a DC power supply was used as a substitute and adjusted to match the DC bus voltage.

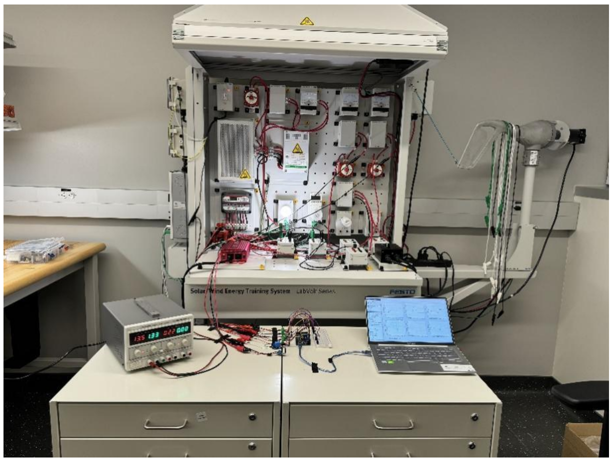

Figure 6. Experimental configuration

Figure 7 depicts the schematic of the wind turbine and the PV panel connected to an MPPT within the test station. The outputs from the solar panel, battery bank, power supply, and wind turbine are all connected to the DC bus, and the DC bus output is connected to a DC/AC inverter. The power inverter converts 12 V DC power into 120 V AC power using a 1 kW DC-to-AC converter. The hardware components previously discussed were assembled to implement the system. Figure 8 shows the hardware configuration, including various sensors such as the DC Bus Voltage Sensor (BVS), Battery Current Sensor (BCS), PV Current Sensor (PCS), Generator Current Sensor (GCS), Wind Turbine Current Sensor (WCS1), and Inverter Current Sensor (ICS2). These sensors measure and send values to the Arduino Mega 2560. The Arduino is connected to a laptop via USB, and MATLAB is run on a Windows machine. Using the MATLAB Support Package for Arduino Hardware, communication between the Arduino and MATLAB is facilitated.

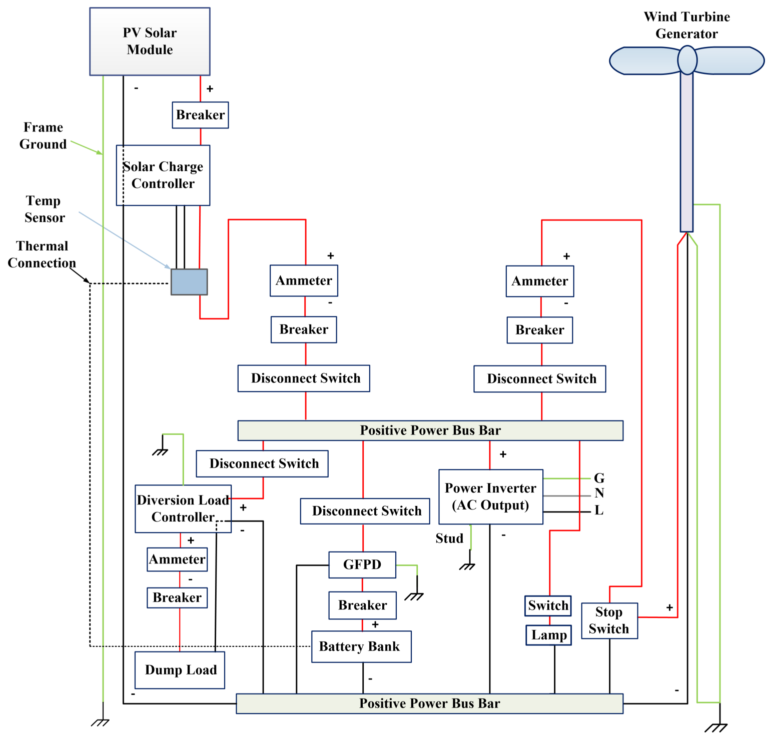

Figure 7. Wind turbine and PV system configuration

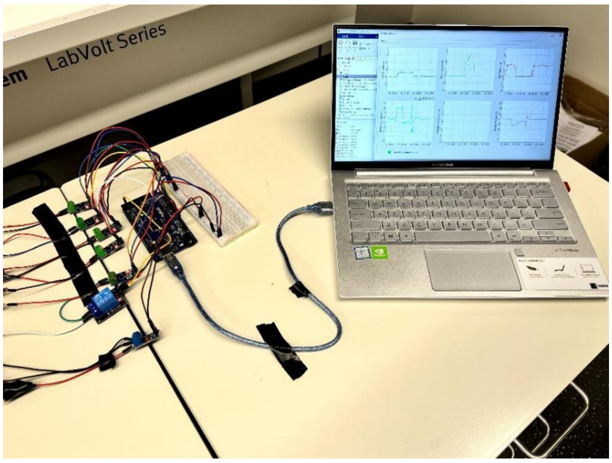

Figure 8. Assembled configuration of the proposed system

The system's core processing unit is the MATLAB App Designer dashboard, which handles all necessary processes including data visualization, control, and fault detection mechanisms. Sensor values were received from designated ports, scaled, and calculated for specific functions. The data were then converted and stored in a CSV file.

7. Results

Figure 9 demonstrates the SCADA architecture results for the proposed system. Each component has a separate display for showing the current of that component, including PV, wind, battery, diesel generator, and inverter, as well as one for showing the voltage on the DC bus. Monitoring each component separately allows for precise performance tracking and early detection of anomalies or faults. This ensures that each part of the system is operating optimally and can help identify areas that may require maintenance or adjustments. Additionally, it facilitates more accurate data collection and analysis, enabling better decision-making for system enhancements and efficiency improvements. In addition, there is one LED in the figure that indicates generator status. When wind and solar power are inadequate, the LED turns green, indicating the diesel generator has been turned on. Conversely, when the LED is red, the diesel generator is switched off. The generator status LED provides a quick and clear visual indication of whether the diesel generator is active or inactive. It is crucial for operators to assess the system's reliance on the generator immediately, as it can impact fuel consumption and overall system efficiency. By having this indicator, maintenance personnel can be alerted to potential issues with the generator's operation, ensuring timely interventions to maintain uninterrupted power supply.

As shown in Figure 9, when both the PV and wind currents are zero, the diesel generator automatically activates, indicated by a green LED, with the generator's current ranging between 1 and 2 A, while the battery voltage stabilizes around 12 V. This sequence occurs due to the hybrid power system's design, where different sources (PV, wind, diesel) interact to balance the energy supply. When renewable sources of power (wind and solar) are not available, the diesel generator takes over, supplying power to maintain system stability. During this time, the battery remains idle or discharges, depending on the load demand.

As the wind and PV systems generate electricity, the diesel generator shuts off, reducing its current to zero. During this phase, the battery starts to charge, leading to an increase in battery current, while the inverter current decreases. However, a temporary voltage drop is observed when the diesel generator is deactivated. The voltage drops due to the system switching between the diesel and renewable sources. When renewable sources again fail to provide sufficient power, the diesel generator resumes operation, and the battery discharges to meet the load demand. This results in an increase in inverter current and battery voltage as the system balances between energy sources. These findings affirm the system's robust capabilities in real-time monitoring and processing and its ability to accurately control signals under various scenarios.

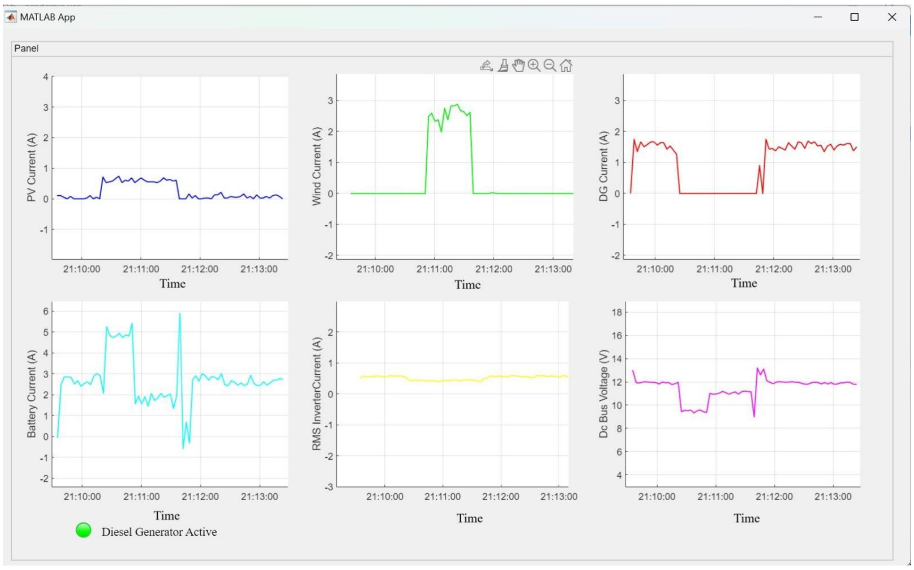

Figure 9. App designer dashboard for visualization

8. Conclusions

In this study, MATLAB-based SCADA architecture is successfully designed and implemented for HRPS, incorporating wind, PV panels, diesel generators, and batteries. This system acquires, processes, and visualizes real-time data using an Arduino Mega 2560 as the RTU, a personal PC as an MTU, and MATLAB App Designer as the monitoring interface. As a result of the experimental results, the system demonstrates its ability to accurately monitor and control the HRPS.

This SCADA system not only provides a cost-effective solution but also leverages MATLAB's robust environment for data analysis and visualization, making it highly suitable for remote areas. Furthermore, the system's design emphasizes the importance of real-time management, early fault detection, and prompt corrective actions, which are critical for enhancing the reliability and efficiency of hybrid renewable energy systems. Overall, this research highlights the potential of MATLAB-based SCADA systems in improving energy management and reliability in industrial automation applications, offering significant contributions towards advancing hybrid power systems.

Conflict of interest

The authors declare that there are no conflict of interest.

References

[1] N. G. Chatzigeorgiou, S. Theocharides, G. Makrides, and G. E. Georghiou, "A review on battery energy storage systems: Applications, developments, and research trends of hybrid installations in the end-user sector," *J. Energy Storage*, vol. 86, p. 111192, 2024.

[2] M. E. Şahin and F. Blaabjerg, "A hybrid PV-battery/supercapacitor system and a basic active power control proposal in MATLAB/simulink," *Electronics*, vol. 9, no. 1, p. 129, 2020.

[3] M. E. Şahin and F. Blaabjerg, "PV powered hybrid energy storage system control using bidirectional and boost converters," *Electric Power Components Syst.*, vol. 49, no. 15, pp. 1260–1277, 2022.

[4] C. Ndukwe, T. Iqbal, and J. Khan, "An Open Source LoRa Based, Low-Cost IoT Platform for Renewable Energy Generation Unit Monitoring and Supervisory Control," *J. Energy Power Technol.*, vol. 4, no. 1, p. 16231, 2022.

[5] P. de Arquer Fernández, M. Á. Fernández Fernández, J. L. Carús Candás, and P. Arboleya Arboleya, "An IoT open source platform for photovoltaic plants supervision," *Int. J. Electr. Power Energy Syst.*, vol. 125, p. 106540, 2021.
[6] G. Yadav and K. Paul, "Architecture and security of SCADA systems: A review," *Int. J. Critical Infrastruct. Protect.*, vol. 34, p. 100433, 2021.
[7] A. Nechibvute and H. D. Mafukidze, "Integration of SCADA and industrial IoT: Opportunities and challenges," *IETE Tech. Rev.*, vol. 41, no. 3, pp. 312–325, 2024.
[8] M. Šverko and T. Galinac Grbac, "Automated HMI design as a custom feature in industrial SCADA systems," *Procedia Comput. Sci.*, vol. 232, pp. 1789–1798, 2024.
[9] G. S. OV, A. Karthikeyan, K. Karthikeyan, P. Sanjeevikumar, S. K. Thomas, and A. Babu, "Critical review Of SCADA And PLC in smart buildings and energy sector," *Energy Rep.*, vol. 12, pp. 1518–1530, 2024.
[10] F. Fauzan, S. Subhan, H. T. Hasannuddin, and E. Eko, "Load Switching SCADA-Based on Three-Phase 15 MW and 20 MW Synchronous Generators," *J. Adv. Res. Appl. Mechanics*, vol. 122, no. 1, pp. 175–185, 2024.
[11] L. Sun, Y. Li, J. Zhang, J. Yang, B. Mao, Z. Deng, et al., "Log specification and intelligent analysis method based on oil and gas pipeline SCADA system," *Appl. Math. Nonlinear Sci.* vol. 9, 2024. https://doi.org/10.2478/amns.2023.2.01261.
[12] R. Ahmed, K. T. I. Ananta, M. F. Anwar, and S. Afrin, "Automatic control of home appliances based on SCADA and IoT system," Ph.D. dissertation, Brac Univ., Mohakhali, Dhaka, Bangladesh, 2021.
[13] B. Babayigit and M. Abubaker, "Industrial internet of things: A review of improvements over traditional SCADA systems for industrial automation," *IEEE Syst. J.*, vol. 18, no. 1, pp. 120–133, 2023.
[14] A. Farhadi and M. T. Iqbal, "Optimal Sizing and Techno-Economic Analysis of a Hybrid Power System for Postville," *Eur. J. Eng. Technol. Res.*, vol. 9, no. 2, pp. 5–12, 2024.
[15] A. Farhadi and M. T. Iqbal, "Simulation and Dynamic Analysis of a Hybrid Renewable Power System for Postville, Labrador," *Eur. J. Electr. Eng. Comput. Sci.*, vol. 8, no. 3, pp. 38–45, 2024.
[16] M. N. Asgher and M. T. Iqbal, "Development of a Low-Cost, Open-Source LoRA-based SCADA System for Remote Monitoring of a Hybrid Power System for an Offshore Aquaculture Site in Newfoundland," *Eur. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 6, pp. 65–73, 2023.
[17] M. O. Qays, M. M. Ahmed, M. A. P. Mahmud, A. Abu-Siada, S. M. Muyeen, M. L. Hossain, et al., "Monitoring of renewable energy systems by IoT-aided SCADA system," *Energy Sci. Eng.*, vol. 10, no. 6, pp. 1874–1885, 2022.
[18] W. He, M. J. A. Baig, and M. T. Iqbal, "An Open-Source Supervisory Control and Data Acquisition Architecture for Photovoltaic System Monitoring Using ESP32, Banana Pi M4, and Node-RED," *Energies*, vol. 17, no. 10, p. 2295, 2024.
[19] D. Li, N. Gebraeel, and K. Paynabar, "Detection and differentiation of replay attack and equipment faults in SCADA systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1626–1639, 2020.
[20] A. Abdulsalam, A. A. J. Alsaadi, and Z. Hamodat, "Control and management of solar PV grid using SCADA system," in *Proc. 2022 Int. Congr. Hum.-Comput. Interact., Optim. Robot. Appl. (HORA)*, Ankara, Turkey, Jun. 9–11, 2022.
[21] G. S. OV, A. Karthikeyan, K. Karthikeyan, P. Sanjeevikumar, S. K. Thomas, and A. Babu, "Critical review Of SCADA And PLC in smart buildings and energy sector," *Energy Rep.*, vol. 12, pp. 1518–1530, 2024.
[22] M. Saber, A. Shareef, S. Ali, O. Hamad, and M. G. Ahmed, "Innovative SCADA-Based Oil Refinery Control with Arduino Integration Using LabVIEW," *Int. J. Multidiscip. Res. Anal.*, vol. 6, no. 10, pp. 4734–4740, 2023.
[23] S. Ferlito, S. Ippolito, C. Santagata, P. Schiattarella, and G. Di Francia, "A Study on an IoT-Based SCADA System for Photovoltaic Utility Plants," *Electronics*, vol. 13, no. 11, p. 2065, 2024.
[24] C. I. Toma, M. Popescu, M. D. Constatinescu, G. C. Buzatu, "Configuration of SCADA System Graphical User Interface for Electrical Station," *Ann. Univ. Craiova, Electr. Eng. Ser.* vol. 24, 2023.
[25] E. O. Alijon O'G'Li, O. Ergashev, and A. X. Juraev, "Design of Greenhouses in a Modern SCADA System," *Eur. J. Emerg. Technol. Discoveries*, vol. 2, no. 4, pp. 36–43, 2024.
[26] W. P. Muljanto, A. Lomi, R. P. M. D. Labib, and M. W. Solihin, "Monitoring System Design of a 4 kWp Off-grid Solar Power Plant," *KnE Soc. Sci.*, pp. 363–372, 2024. https://doi.org/10.18502/kss.v9i10.15742
[27] Arduino Mega Pinout. Accessed: Nov. 16, 2024. [Online]. Available: https://store.arduino.cc/products/arduino-mega-2560-rev3?queryID=undefined.
[28] Fully Integrated, Hall Effect-Based Linear Current Sensor. Accessed: Nov. 16, 2024. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf.
[29] Current sensor. Accessed: Nov. 16, 2024. [Online]. Available: https://techiesms.com/product/acs712-5a-current-sensor-module/.

[30] Voltage sensor. Accessed: Nov. 16, 2024. [Online]. Available: https://components101.com/sensors/voltage-sensor-module.

[31] 5V Relay terminals and pins. Accessed: Nov. 16, 2024. [Online]. Available: https://mounishkokkula.wordpress.com/how-the-5v-relay-works/.

[32] Relay module. Accessed: Nov. 16, 2024. [Online]. Available: https://digipurk.ee/en/product/keyes-5-v-relay-module-for-arduino/.

[33] The wind/solar training system. Accessed: Nov. 16, 2024. [Online]. Available: https://labvolt.festo.com/downloads/datasheet_580181(1946120)_en_120V_60Hz.pdf.