

Article

Development of a Supervisory System Using Open-Source for a Power Micro-Grid Composed of a Photovoltaic (PV) Plant Connected to a Battery Energy Storage System and Loads †

Fernanda Moura Quintão Silva ^{1,2}, Menaouar Berrehil El Kattel ^{1,3} and Igor Amariz Pires ^{1,4,*} 
and Thales Alexandre Carvalho Maia ^{1,5} 

¹ Graduate Program in Electrical Engineering (PPGEE), Universidade Federal de Minas Gerais—UFMG, Belo Horizonte 31270-901, MG, Brazil

² Department of Mechanical Engineering, University of Texas at Austin—UT, Austin, TX 78712, USA

³ Department of Electrical Engineering (PPGEE), Universidade Federal do Ceará—UFC, Fortaleza 60356-001, CE, Brazil

⁴ Department of Electronic Engineering (DELT), Universidade Federal de Minas Gerais—UFMG, Belo Horizonte 31270-901, MG, Brazil

⁵ Department of Electrical Engineering (DEE), Universidade Federal de Minas Gerais—UFMG, Belo Horizonte 31270-901, MG, Brazil

* Correspondence: iap@ufmg.br

† This paper is an extended version of our paper published in: Silva, F.M.Q.; Cardoso, Filho, B.J.; Pires, I.A.; Maia, T.A.C. Design of a SCADA System Based on Open-Source Tools. In Proceedings of the 2021 14th IEEE International Conference on Industry Applications (INDUSCON), Virtual Conference, 15–18 August 2021.



Citation: Silva, F.M.Q.; El Kattel, M.B.; Pires, I.A.; Maia, T.A.C. Development of a Supervisory System Using Open-Source for a Power Micro-Grid Composed of a Photovoltaic (PV) Plant Connected to a Battery Energy Storage System and Loads. *Energies* **2022**, *15*, 8324. <https://doi.org/10.3390/en15228324>

Academic Editors: Saeed Sepasi and Quynh Thi Tu Tran

Received: 28 September 2022

Accepted: 21 October 2022

Published: 8 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: The importance of renewable energies and energy storage system forming a micro-grid and integrating it to the electrical grid is widely spread. A supervisory system plays a crucial role in controlling, managing, and planning the micro-grid. This paper demonstrates the development of a new custom supervisory system based on Internet of Things (IoT), creating an information sharing environment. The proposed supervisory system is based on open-source tools for a micro-grid, composed of a photovoltaic power plant and a storage system, employing smart devices and making non-smart devices compatible with IoT systems. The new supervisory improves the available system by incorporating new features and devices and increasing the data polling rate when necessary. A comparison between the current supervisory system and the proposed one is performed, showing that the new system is more flexible, easily modified, cost-effective, and more fault-resilient.

Keywords: internet of things (IoT); supervisory system; solar power; Raspberry Pi; node-red; Grafana; energy storage

1. Introduction

The main challenges to be surpassed regarding renewable energy sources are availability and cost. Solar and wind resources, for instance, are seasonally and hourly dependent. These can greatly influence the energy generation [1]. The mitigation alternatives to the aforementioned challenges could be supervision, control, and energy storage.

Traditionally, a supervisory and data acquisition system (SCADA) is used to supervise processes by acquiring data from field devices. Users can access the data via screens that provide a visual aspect of the whole system. The supervisory system provides a link among the field devices and the control room [2,3]. This system interacts with field devices via inputs/outputs, using communication equipment to create a link among them, and is generally installed in a central computer.

In order to supervise a micro-grid, the requirements are not the same from the ones used in regular grids, since multiple generation, storage, and load systems are independently connected. There is a lack of research papers related to supervision in micro-grid

studies. The improvement and renovation of supervision and monitoring features could lead to a more efficient management of power generation and micro-grid operation. Adding new capabilities to the system, such as Internet of Things, plays a vital part in the operation quality of renewable source power plants [4–7].

The Internet of Things (IoT) is an exchange information environment with connected devices in wired or wireless networks, enabling easy remote access. Constant monitoring of environmental and electrical data is an extremely promising application [8–10]. With the aid of IoT, it is possible to improve processes by acquiring and processing large amount of data [11].

This paper presents a customized system capable of monitoring and controlling a micro-grid. The micro-grid used for this study is composed of a 37 kWp photovoltaic power plant, which is connected to a battery energy storage system (BESS) of 15 kW/45 kWh. The motivation of developing a supervisory system comes from the fact that the current one, developed by a third party company, has lower flexibility to incorporate new devices, higher costs, and lower sample rate of data acquisition. The proposed SCADA is designed to integrate IoT devices, allowing better efficiency and flexibility to not only monitor and perform data analysis, but also for the management of the power plant in conjunction with the energy storage system. Thus, information sharing between different grids could be enabled in order to better manage the energy supply for a specific load. The information sharing would lead to an increase in efficiency and better usage of the energy.

Another advantage of the proposed system is the segregation of SCADA services into individual hardware, which are network connected. The proposed change promotes a fault tolerant capability, since eventual failures in a service do not affect the other ones. In addition, if the hardware fails, only one service is compromised and easily replaced.

This paper aims to demonstrate the development of a custom-made remote access control and monitoring system of all available electrical and environmental variables. This supervisory system might be flexible—changes and expansions should be easily incorporated—adaptable to research, more cost-effective, and fault-resilient [12,13].

The contributions of this paper are

- Development of a low-cost open-source flexible supervisory system for the monitoring of the measured values of a photovoltaic power plant;
- Development of a supervisory system that combines the traditional capabilities with IoT concepts;
- Development of a supervisory system that allows remote communication and remote data sharing;
- Development of a supervisory with fault tolerant capability.

This paper is divided in six sections: Section 2 provides a brief bibliographic review, with an overview of traditional supervisory systems and an overview of Internet of Things concepts, such as its structure and smart devices. Section 3 details the methodology implemented in the development of the supervisory system. Section 4 describes the analyzed micro-grid, which comprises of a photovoltaic power plant and the energy storage system. Section 5 shows a comparison between the current supervisory system and the new one. Section 6 concludes this paper.

2. Bibliographic Review

2.1. Traditional Supervisory Systems

The supervisory control and data acquisition (SCADA) system is built using hardware devices and software that enable the control and monitoring of the analyzed process manually or automatically. It is traditionally a central controller that consists of network interfaces, input/output, communication equipment, and software. The system gathers data from sensors and field instruments and, via a communication interface, the data are sent to the central controller to showcase an overview of the whole process. Figure 1a illustrates a generic architecture for the SCADA system.

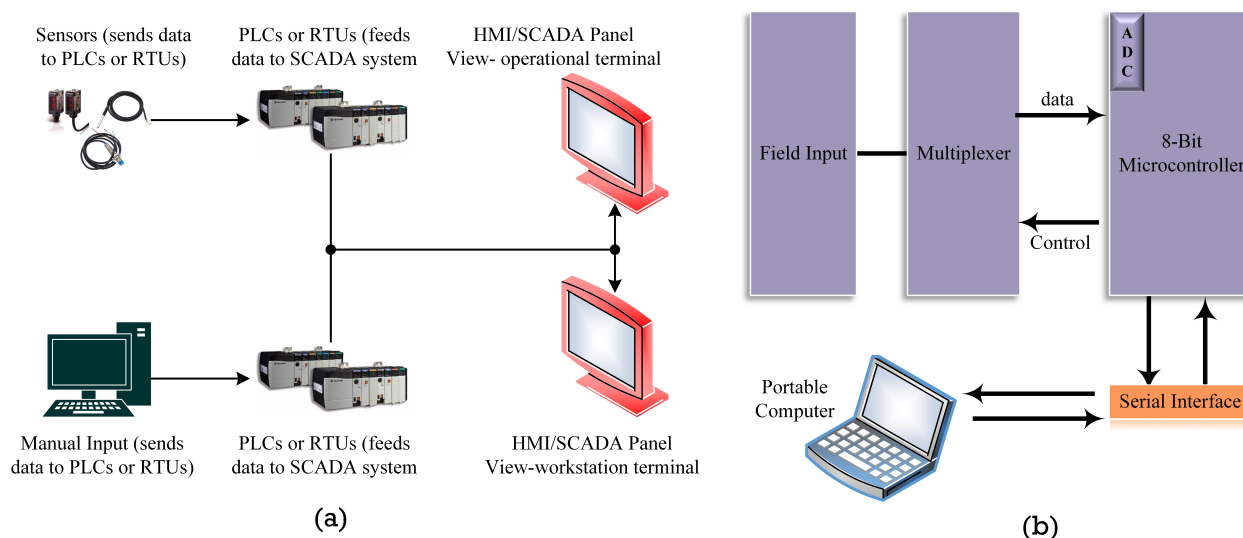


Figure 1. SCADA System. (a) Representation of the hardware architecture of a SCADA system, including field devices such as PLCs (programmable logic controller) or RTUs (remote terminal units) and the computer with the controller. (b) Example of implementation using a Portable Computer and microcontrollers via a serial interface.

The supervisory system acts as a link between the user in the control room and the field equipment, such as intelligent electronic devices, programmable logic controllers (PLC), remote terminal units (RTU), and sensors. These field devices are capable of transmitting data in both directions. They send the field data to the supervisory system and receive commands from the control center [2,3].

Traditionally, data loggers or microcontrollers are used for measurements/data acquisition, which can be transmitted to a user terminal via a serial port (UART/RS232/RS485) [4,5,7,14,15]. A system is developed in order to collect, record, and transmit the measured data to a computer; an example of this implementation is performed in [14], in which the developed structure is summarized in Figure 1b.

The acquisition scheme employs an I/O (input/output) microcontroller to retrieve the measured data. The data are then transmitted to a portable computer via wired connection through a serial port via the communication protocol, and it can be accessed for further analysis. The supervisory and control system are developed on this specific computer using assembly language, and it is only accessible in it. This system allows bidirectional communication, sending field data and receiving commands from the control center [14,15]. There are other literature references that also use this methodology for developing the supervisory system [16–18].

There are limits in the usage of serial communication, such as low sampling rates, and some authors propose the usage of I/O modular devices and data acquisition with high-speed A/D conversion [4]. This project also relies on a local supervisory system with wired-serial connections between the user terminal and the field devices, i.e., the monitoring and access to the system are available only to local users.

More recently, with the technological advancements in wired/wireless networks, the serial communication has been replaced by the establishment of a local network. The use of a local network enables communication between the field and the control room [19,20]. With the aid of this architecture, most operations are automatically executed by the local equipment or by the programmable logic control unit (PLC). This process is illustrated in Figure 2.

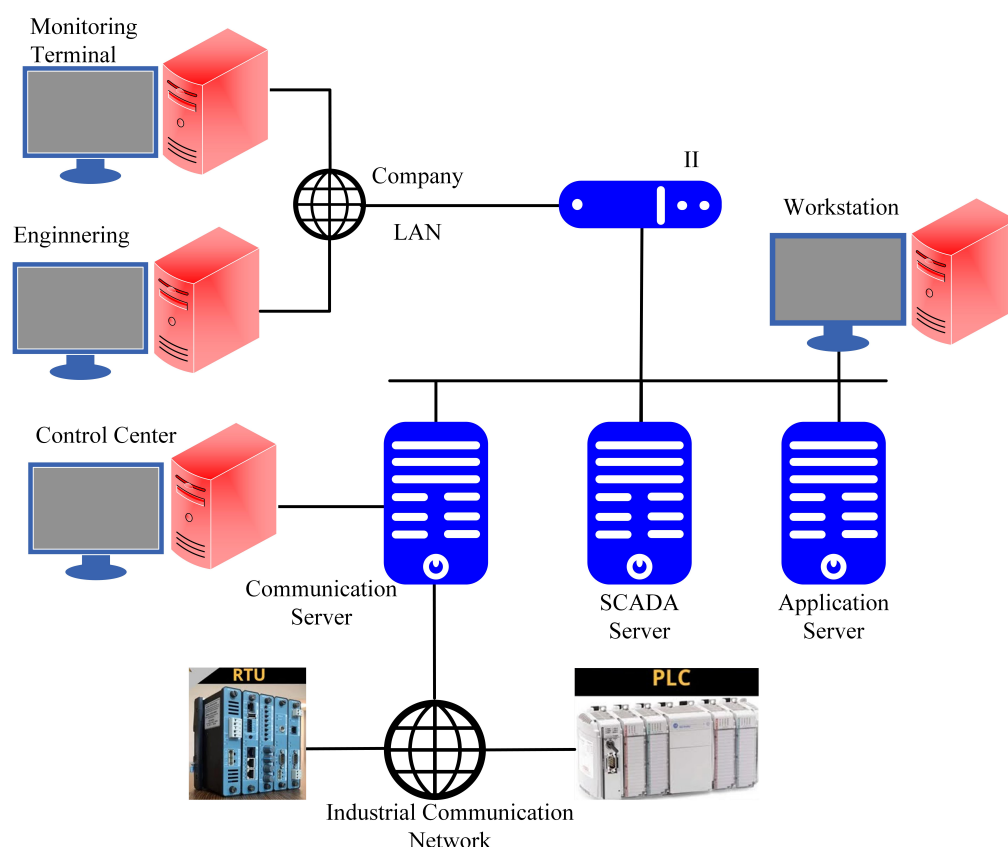


Figure 2. Illustration of a SCADA, data-acquisition devices, servers, workstations, and user terminals, in which data are exchanged over a Local Area Network (LAN) between devices.

Data acquisition begins at the RTU or PLC. The measurements and the state of gauges are communicated to the SCADA application. Dedicated devices are responsible for the monitoring and the control. The supervisory system presents the real-time and the historical data graphically in terminals connected to the local network. This structure is a technological advancement when compared to previously shown systems. It establishes a LAN (local area network—wired or wireless) to promote the communication between field devices and the SCADA serves, the communication server, and the user interfaces.

The authors of [21] develop a supervisory system for a photovoltaic system whose structure is similar to the one seen in Figure 2. The field devices communicate with a local gateway through a serial bus. The gateway provides wireless communication services for the devices using wireless sensor network (WSN) technology with a central terminal, where the data are processed. It is interesting to notice in this research work that the devices are not IoT devices—they are merely wireless. However, after the data are processed in the central terminal, it is published to remote users using publishing services [21]. This work shows a step towards the concept of the IoT.

2.2. Internet of Things—An Overview

IoT has been identified as one of the emerging technologies in IT [22–24]. It is a global Internet-based information architecture, facilitating the exchange of goods and services in global supply chain networks [25]. IoT allows the interaction among devices inside the global network, enabling devices to communicate with each other and interact with inventory systems, customer support systems, and business applications if required. Due to its communication via Internet, this environment also allows remote control [26,27]. A modified model for the IoT architecture can be seen in Figure 3a.

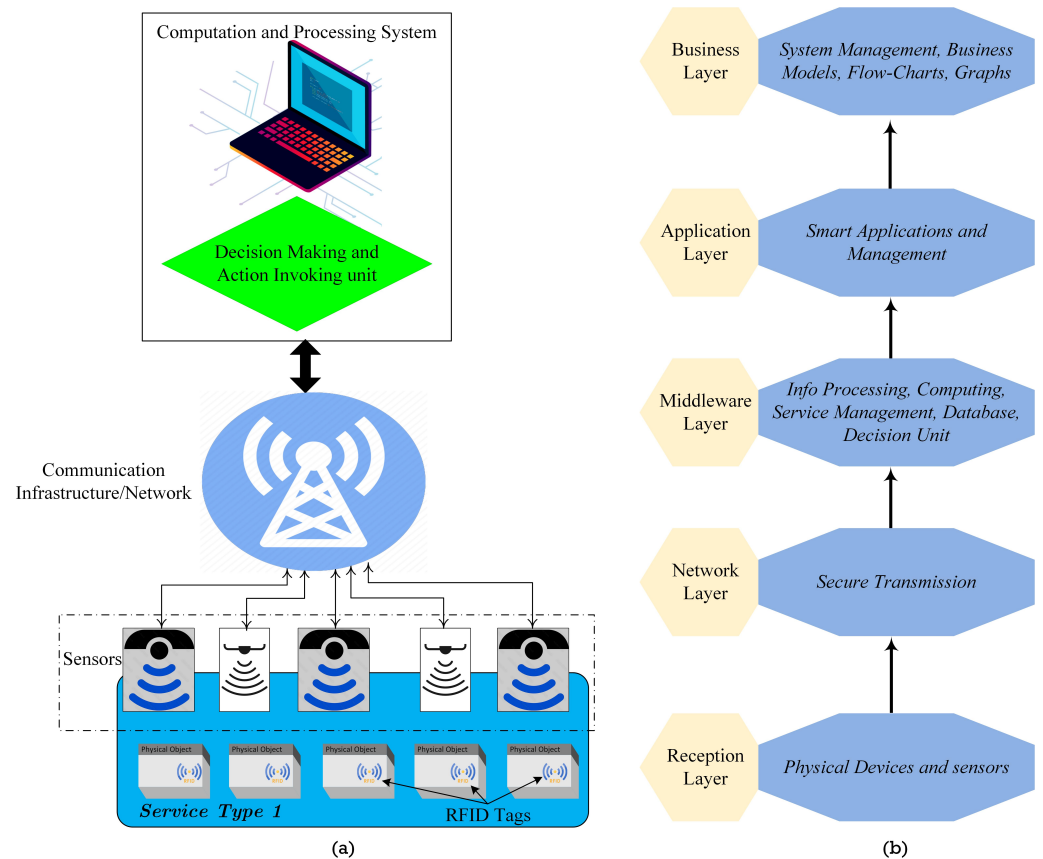


Figure 3. Illustration of IoT structure (a) showcasing each fundamental block, and (b) IoT structural layers, comprised by sensing and data acquisition, up to system management.

The physical objects and sensors block are related to the identification of each individual component when inserted in the context of the network. As it is represented by the arrows, the physical objects can send/receive data to/from the network. The communication infrastructure is related to the communication protocols used to establish the link among devices, devices and services, and devices and users. The decision-making inside the computation and processing unit is related to the ability of extracting information from the devices data in order to activate certain services and to execute local algorithms [28]. The basic structure for IoT is composed of five layers as seen in Figure 3b.

The bottom layer is the perception (or device) layer, and it consists of the physical equipment and the sensing devices. This layer is responsible for the identification of the devices and data acquisition. This information is then passed to the network (or transmission) layer for its secure transmission to the data processing system. The transmission can be wired or wireless. The data are then transmitted to the middleware layer. This layer has links to the database, which is capable of storing the data. The middleware layer processes the data and activates services based on the results of the data processing. The application layer is responsible for global management of the application based on the data processing on the previous layer. Finally, the business layer is responsible for the management of the IoT system, including applications and services [28].

IoT is pushing a whole slew of applications, especially when considering industry and customer-oriented applications, such as automotive and intelligent transportation systems [29,30], remote structural health monitoring [31], smart homes, buildings, and neighborhoods [32–34], smart irrigation in tunnel farming [35], metering monitoring [36], concrete surface, and level of trash monitoring [37]. IoT can also be applied to manage the energy consumption of a building in order to improve costs and efficiency [11].

Smart Devices

The devices and gauges which interact in IoT environment are smart devices. Smart devices are capable of communication and computation. The key features of smart devices are autonomy, connectivity, and context-awareness. Autonomy is the capability of performing tasks without external interference. Connectivity is to establish a link to any type of wired/wireless network to complete a task as, for example, accessing the Internet or sharing information with other devices. Context-awareness means that the device is able to perceive environmental information via sensors or sensing units [38].

Smart devices are able to connect to the Internet via wired/wireless connections, to exchange data with other IoT-gauges. Due to the feature of being smart, it is possible to control and remotely access them; the access can be achieved by users or services. A smart device is a context-aware equipment that can be used as a service provider and can communicate with devices in different networks [38].

Smart devices implement three steps: data acquisition, data process, and communication. In the first step, data acquisition, the device acquires the data measured by its sensing unit. Then, the raw data are processed by the processing unit. In addition, finally, the processed data are transmitted to the network via a communication protocol [39]. Even though this technology has allowed various possibilities to emerge as smart houses and smart grids, there are a number of challenges that must be investigated.

The main challenges in IoT are [28]:

- *Naming and identity management*: Each device connected to the network must have a unique identity over the Internet—an efficient naming and identity system that is capable of managing a large number of devices is required;
- *Interoperability and standardization*: Many manufactures have their own protocols—it is important to standardize them to guarantee interoperability. A network that employs IoT concepts has to deal with heterogeneous elements and different communication protocols [40];
- *Devices safety and security*: It is necessary to prevent security breaches in order to protect the integrity of the devices;
- *Network security*: The data transmission should be secure and capable enough to protect the integrity of the data and to prevent external interference or monitoring.

Even with these challenges, this new technology can provide significant benefits.

3. Methodology for the Development of the Supervisory System

To develop a monitoring and supervision system, one has to focus on the computer service that is inherent to the development of a supervisory system. The step of hardware design could be challenging. There are some demands to fulfill this requirement satisfactorily: implementation of backups services and tools, setup of security features, setup of software updates, and usage of uninterruptible power supply (UPS) systems. One important task to consider is also the adequate management of hardware resources, since it is vital to maintain enough available hardware for expansion when dealing with cloud computing.

To overcome the hardware design hurdle, each specialized subsystem of this supervisory is assigned to a container-like system. Specifically, one resource is allocated for data acquisition from the available field hardware, another one is employed for information processing, another to provide a user interface, and so on. With this structure, a single failure in one of the specialized services does not negatively impact the whole system. The main difference of this architecture over container computing is the usage of a dedicated server for each proposed service. Multiple microcomputers (Raspberry Pis) are used instead of developing services in a single multi-core hardware.

The usage of the proposed modular structure brings several benefits. Employing single separate servers for each service prevents a catastrophic failure from happening when one of the servers presents a failure. If one of the server fails, the error could be easily traced to the failed one. This error could be studied and isolated in order to be fixed,

without impacting the performance of the other functioning servers. In other words, a singular isolated failure in one service would not translate in a failure of the entire system. Another advantage is the possibility of parallel processing. All services could operate concurrently with no impact on each other. The services are allocated to different servers, meaning that, if there is a failure in a process, it does not hinder others. Increases in the efficiency and in the reliability of the system as a whole could also be counted amongst the benefits of this proposed structure since each server is powered by its own dedicated UPS (uninterruptible power supply).

The Raspberry Pi is a series of compact single-board computers. The choice of using Raspberry Pi in this application is due to the fact that its features align with the requirements for the development of the new supervisory system. They are cost effective and a powerful processing unit in a compact board. The Raspberry Pi has built-ins for the usage of different interfaces (HDMI—High Definition Multimedia Interface, USB—Universal Serial Bus, Ethernet, Wi-Fi, Bluetooth), and it supports Linux and Python, enabling easy development of applications [41–44].

The development of a supervisory system requires the implementation of some specialized services in order to function properly. These services would be required to acquire data, interpret, and translate this data, and display it in a user interface via the monitoring system. The establishment of a communication among field devices, supervisory system, data storage, and backup system is necessary. Three servers are established: (1) communication server, (2) database and application server, and (3) the backup server as seen in Figure 4.

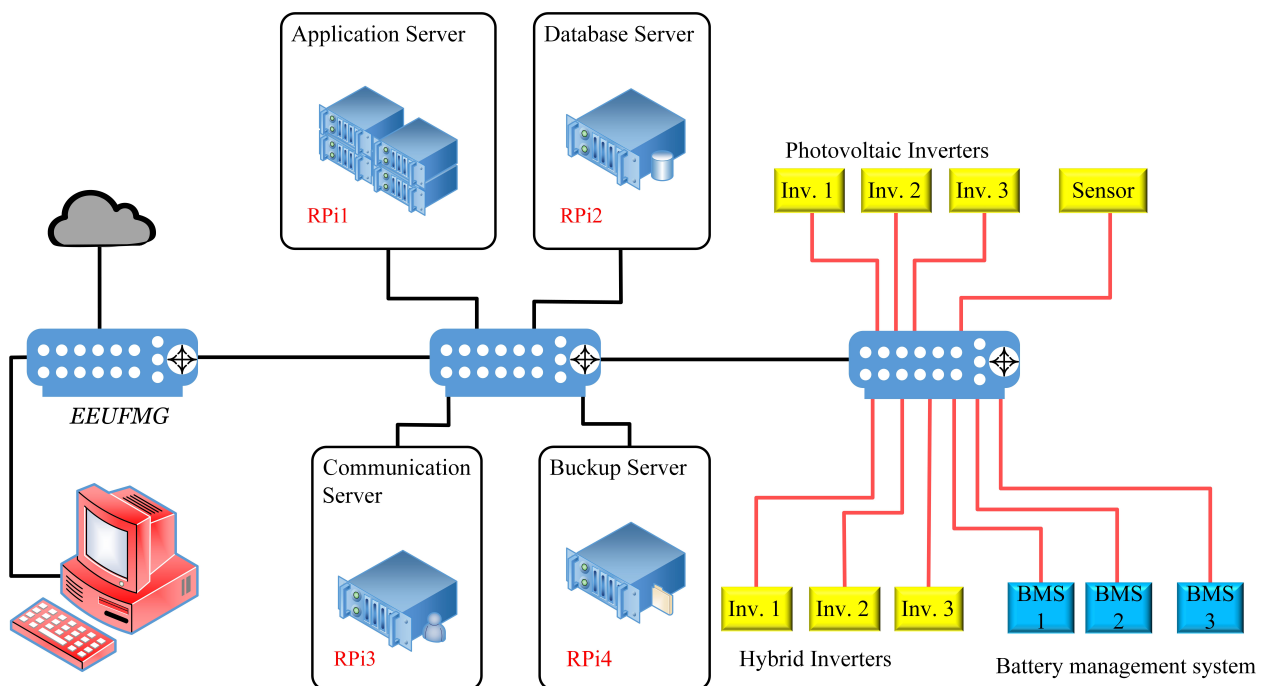


Figure 4. Schematic of the proposed server architecture for the supervisory system presenting the implemented system with the RPi (Raspberry Pi), inverters, and batteries.

Another important feature of the supervisory system is the employment of a set of tools capable of providing an easily expandable and flexible monitoring system. Since it has been established that employing IoT-related concepts is beneficial, the chosen tool set must be integrated into its environment.

There are a lot of open-source software options that could be implemented in the development of the system to meet the requirements. The one that is the most popular and the most versatile is Node-Red [45]. Node-Red is an open source tool developed initially by IBM for flow programming using a local host, meaning it is a browser based flow

editor [46]. The usage of a flow-based programming (FBP) has many advantages. The main reason is that FBP allows parallelism, providing performance benefits in some situations. In FBP, the dataflow is the main driving force of the program. The logical execution flow is expressed by the block diagram created: when a node receives all necessary inputs, the block produces an output that is transmitted to the next node in the path. FBP treats the applications as black-boxes: the important component for this type of programming is the connections between components that are conducted externally to the processes. Due to this style of programming, FBP lends itself to a plug-and-play approach. Node-Red is also able to connect to hardware devices, such as microcontrollers and the Raspberry Pi amongst many others, and to the cloud environment.

The advantages of using Node-Red when comparing it to other open-source tools come especially from the usage of Node.js, a light and agile open-source tool. Due to its flow-based programming and the ability of building custom functions, Node-Red is able to fulfill the requirement of flexibility [47]. Since the developed supervisory system is built for monitoring electrical and environmental variables, the processing rate of messages inside the code does not need to be extremely fast. For this reason, the speed of one message/second of Node-Red is sufficient and suitable for the application.

In order to build the graphical screens for the development of the supervisory system, it is necessary to use another tool that is more user-friendly for the creation of dashboards. The platform used here is Grafana. Grafana is a multi-platform tool that is able to provide graphics, charts, and real-time measurements based on incoming information from any kind of data source [48]. Grafana is commonly used to display data and alarms for further analysis of any type of application. This platform can display real-time data as well historic measurements and events. To achieve the access to the data, Grafana has embedded a database tool that allows the user to configure a database in which the desired data are stored. The database amongst the pre-configured in Grafana that is more aligned with the goals of this paper and with the type of desired monitoring (measurements of a power plant with smart devices) is InfluxDB. InfluxDB is an open-source time series database, and it uses a programming language similar to SQL. By using the InfluxDB nodes in Node-Red and the pre-configured tool in Grafana for InfluxDB databases, it is possible to create a link between the supervisory system and the data requested by Node-Red from the field devices.

3.1. Communication Server and Database Server

The plant where the supervisory system has been developed has smart and non-smart devices. Smart devices are already integrated with the concept of IoT, meaning that they are already able to join an established network and communicate with other smart devices and/or servers directly. The photovoltaic inverters, the hybrid inverter connected to the energy storage system, and the energy meter are smart devices. The non-smart devices cannot communicate in the established network without an interface with an intermediary equipment, such as a data aggregator or a microcontroller; they are not integrated with IoT concepts. The DC-current meter in the photovoltaic inverter's input and the solar meter station are not smart devices. Thus, both of them need an intermediary device to manage the flow of data. In addition, as already stated, the energy storage system communicates with the supervisory system remotely. This means that its communication is conducted via the Internet, via the cloud. The communication architecture can be seen in Figure 5.

Modbus TCP/IP is used for the communication between two of the photovoltaic inverters (Fronius and SMA) and the communication server and between the energy meter and this server. The inverters are capable of measuring electrical data from the DC-bus (the solar panel side) and from the AC-bus (the main grid side) as the energy meter. These data are available via communications protocol for data processing. The Modbus protocol is a highly used flexible open message structure used for the communication between master and slave. The communication is only initiated by the master. The master is responsible for beginning the process by sending a data request to the slave. The connection is conducted point-to-point. The structure of Modbus guarantees that the final recipient receives the

message: the master receives an acknowledgment that the request has reached the slave and the slave receives a flag that the master received the requested data. The Modbus protocol is illustrated in Figure 6a.

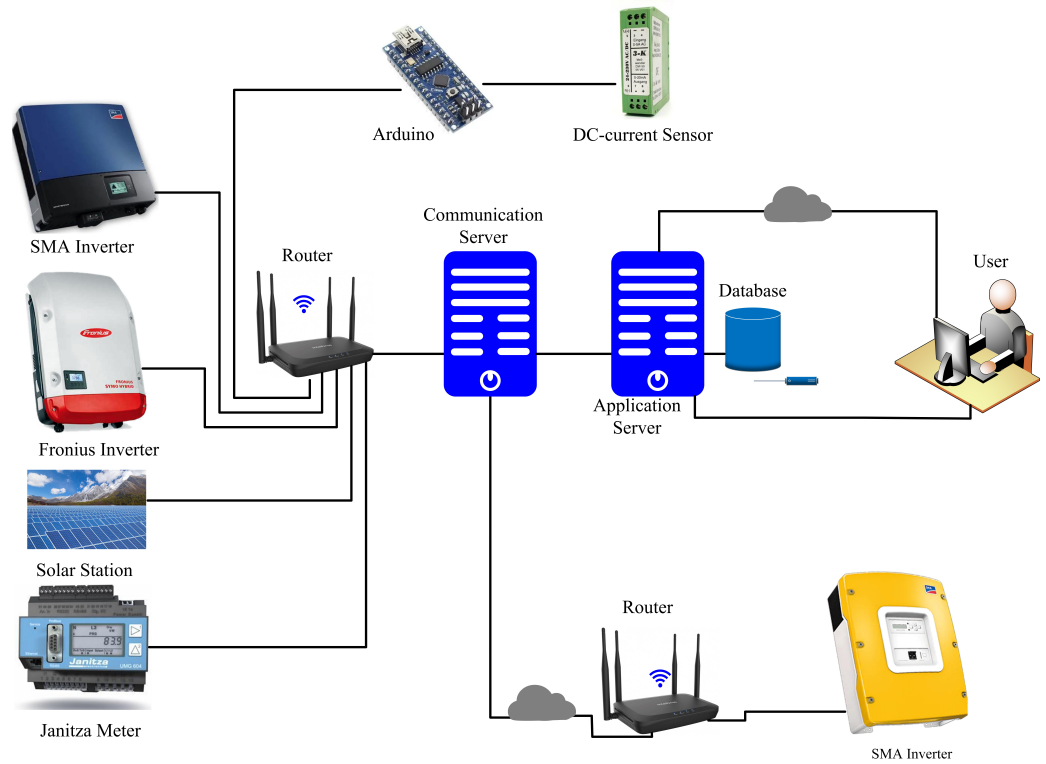


Figure 5. Proposed structure for the data acquisition and supervisory system including the data flow path, field devices, router of the control room, application system, and the database.

The third photovoltaic inverter PHB communicates with the server via Serial Modbus RS-485 since it does not have an Ethernet interface and it does not allow for IP configuration. The Serial Modbus follows the same methodology as the Modbus TCP/IP. The physical connection is conducted with a twisted wire pair plus ground (GND).

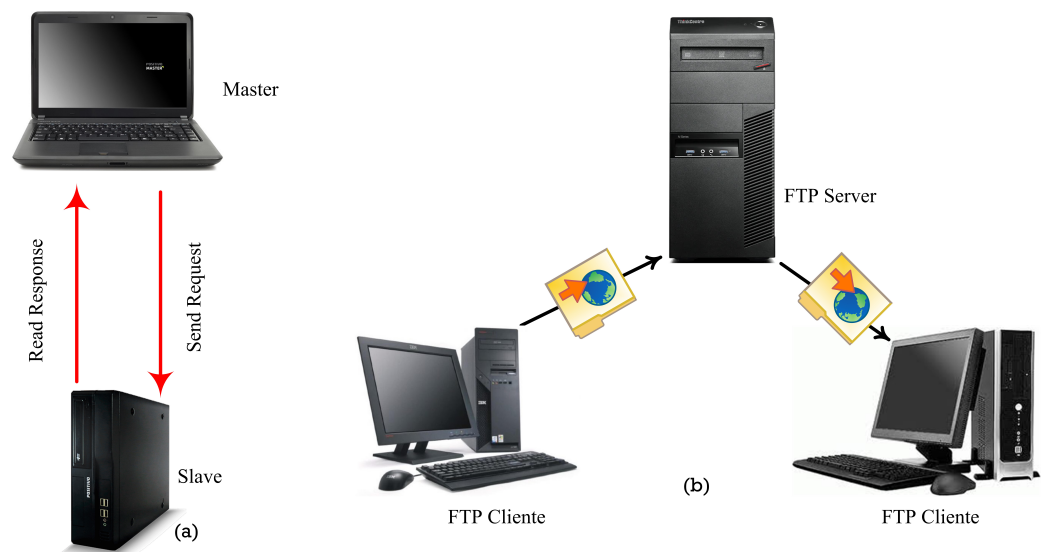


Figure 6. Schematic of the process involved in the Modbus protocols, including the (a) server and client (Master/Slave) communication, and (b) the process to request data using ftp.

Node-Red is equipped with nodes that can be configured to establish a Modbus connection between the communication server (Raspberry installed with Node-Red) and the inverters and the energy meter. That way, the raw data from these devices can be acquired by the request made by Node-Red (acting as a server to modbus clients for certain electrical variables, e.g., AC/DC voltage, current, power, amongst others).

A solar meter station is another gauge incorporated in the supervisory system. It could measure absolute and relative humidity and pressure, ambient temperature, global radiance, and wind direction (a reference cell is monitored separately). This solar station is composed exclusively of non-smart devices, i.e., in order to make their data available, these gauges communicate with an intermediary data-logging device via Modbus RS-485 (Modbus via serial communication). Using the file transfer protocol (FTP), it is possible to transfer the data from these gauges to the communication server. FTP is a secure connection between devices that allow file exchange. FTP also relies on a client–server structure; the client requests access to a certain file and the server grants this access. This protocol is a simple tool that allows high volume of data transfer through a network, and it allows various directories to be transferred at the same instant. The process is shown in Figure 6b.

The idea is to upload the files with the gauges data to the communication server and post it to the application server to incorporate it into the supervisory system. The FTP client needs to act as the FTP link and make requests to the FTP server and receive access to the desired files. In order to supply this link for the current application to retrieve the sensors data from the data-logging device, a Python program is used. This program is uploaded to the communication server as a background service that starts running with the booting of the Raspberry. The upload to the communication server is performed via FTP through the Python service. Figure 7 summarizes this process.

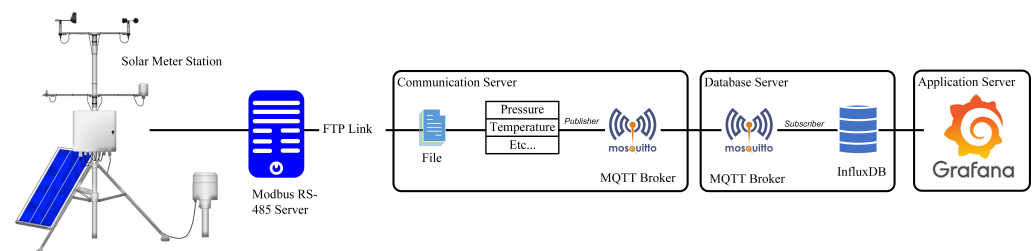


Figure 7. Schematic of the communication established between the solar meter station and the application server, showcasing the interaction between FTP, MQTT, and the database.

There are also the DC-current sensors that are not smart devices. In order to include the data from the sensors to the supervisory system, an intermediary device must be added to the system in order to collect the data and provide it to the application system. To make these measurements available for the network, the output of each current sensor is connected as an analog input to an Arduino Nano with an Ethernet Shield (ENC28J60). Each analog input of the Arduino that is connected to the DC-current sensors has its value read in the Arduino. These values need to be converted to engineering units since they are presented in bits as digital outputs. After they are processed, they are transmitted from the communication server to the application server via MQTT. Figure 8 summarizes this scheme.

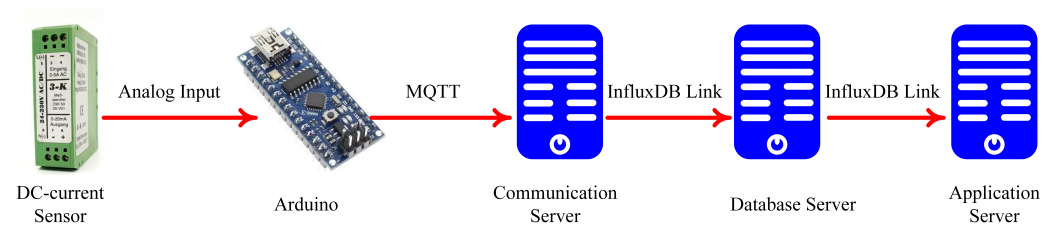


Figure 8. Data acquisition process for the DC-Current sensors deploying smart sensors. The micro-controller uses MQTT to send data to the database.

There are also three hybrid smart inverters, named SMA Sunny Island 8.0h, in this network that are connected to the battery banks that make the energy storage system. These inverters also communicate with another communication server via Modbus TCP/IP.

These inverters not only provide data but also receive data—commands—via Modbus TCP/IP in order to enable the user to control the reactive and active power flow of the grid by managing the charge and discharge of the battery banks. The inverters are capable of providing data regarding the measurements of the grid (AC-side) and data regarding the measurement of whichever battery bank is connected to it (DC-side); it is important to note that each inverter is connected to a phase conductor and they operate separately and do not communicate amongst themselves.

These data must be sent to the main communication server in order to post it to the application server to be incorporated into the supervisory system. The data must be sent via an Internet connection since the energy storage system is placed in a separate grid. It is then necessary to provide a secure communication protocol by applying SSL certificates. This security layer is applied to the lightweight MQTT protocol (MQTTS), together with password authentication, to send the data of the hybrid inverters to the main communication server. This solution cannot be seen as definitive, as research related to cybersecurity is advancing, mainly towards blockchain and the use of a well-structured sensor network [49].

In this type of architecture, there are devices that generate and publish data (publishers) and other devices that consume these data (subscribers). This architecture demands an extra entity that acts as a centralizer in the data exchange, the Message Broker. The devices that generate data publish it to the broker which organizes the data in topics. The devices that consume the data subscribe to the desired topic and are able to retrieve the data assigned to the topic. Every instant new information reaches the topic in the broker, this information is automatically sent to the device that subscribed to the topic. The process is summarized in Figure 9. Node-Red environment was used to implement all protocols due to its steady and reliable communication characteristics.

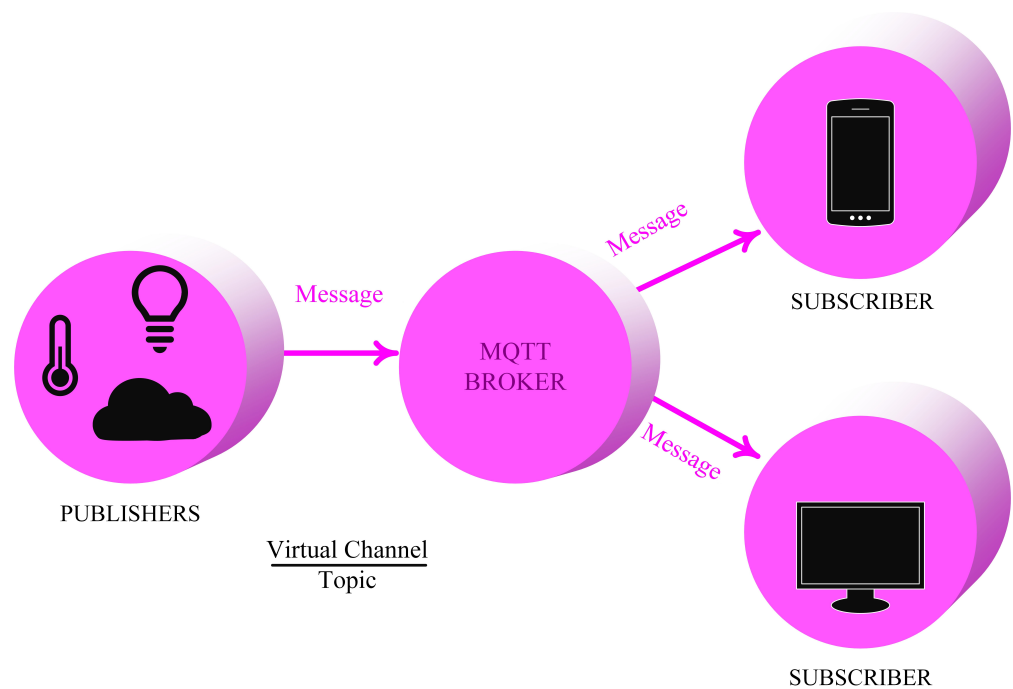


Figure 9. Schematic of the process involved in the MQTT protocol: publishers deliver the data to the broker and then the subscribers make a request to the broker for the desired data.

In order for the application server to access the data from the field devices, it is necessary to transmit the data from the communication server to the database server. For that, the InfluxDB database is used and the Node-Red environment already has built-in nodes that enable an easy configuration of the data transmission. InfluxDB is a password-protected time-series database, allowing the storage of sensor measurements with their timestamps. This database has a built-in time service that ensures the synchronization of time throughout the system. The advantages of using this particular tool are related to the processing and storage speed due to its simplified structure. Once the data are stored in the database server, it is possible to configure the link between it and the application server that houses Grafana, using its own databases configuration tool. Since these servers (communication, database, and application) are all on the same secure network, it is not necessary to implement another layer of security for the established links.

In order to summarize the architecture for the communication links among devices and servers, Figure 10 is shown. A flowchart with the steps of the work and processes along with the communication protocols is presented in Figure 11.

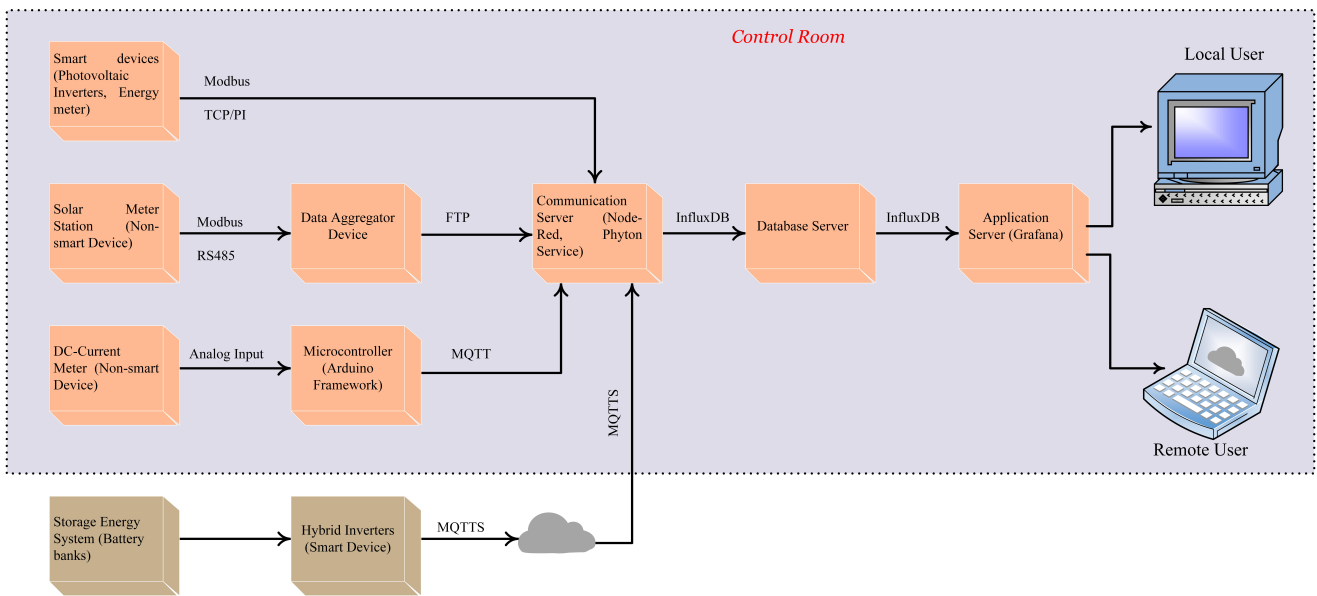


Figure 10. Schematic of the architecture of the data acquisition and data monitoring system, showcasing the communication protocols.

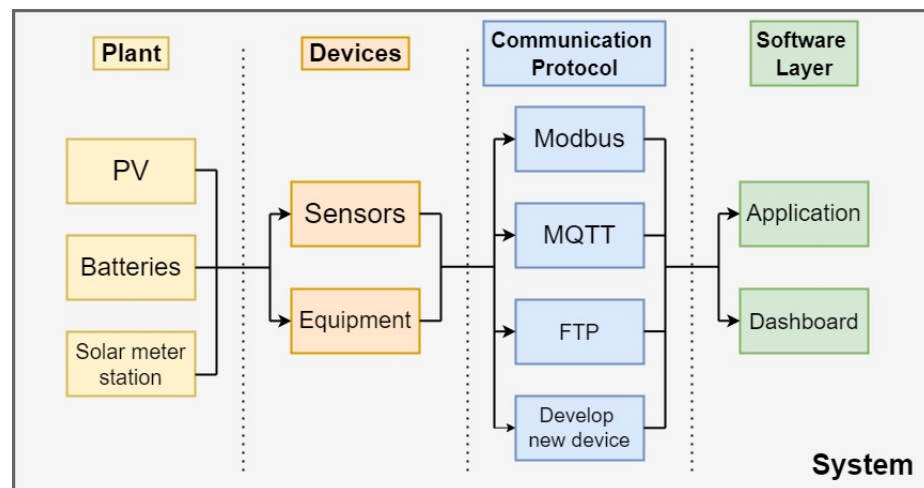


Figure 11. Flowchart of the development of the supervisory system including the steps necessary for the configuration of the communication protocols.

3.2. Application Server

In order to develop the screens of the supervisory system, the data are transmitted from the database server to the Grafana environment. Due to the features of this platform, it is possible to build any custom dashboards, meaning that this tool is capable of fulfilling the flexibility requirement to generate any dashboard necessary.

In order to retrieve the data, the application allows the configuration of the InfluxDB databases created in the database server. For this application, there are six databases and all of them must be configured in Grafana.

The users need an overview of the measurements of the micro-grid in real-time. Thus, it is interesting to provide the single-line diagram of the system with live measurements and the status of the photovoltaic power generation. The new supervisory system should also allow for the plotting of real-time and historical electrical and environmental data for further analysis.

The development of the screens of the supervisory system is conducted solely in the Grafana environment. This platform provides a user-friendly interface to build a custom-made supervisory system, presenting features and gadgets that allow customization. In

addition, due to its built-in InfluxDB tool for the configuration on data transmission, the platform is easily expandable.

Grafana also allows the creation of multiple users. The user can be set up to have different access level rights. One user can be configured as the administrator; another can be a guest and only have viewing access.

The versatility of Grafana is one of the main advantages when comparing it to the provided current supervisory system. For the latter, new features and/or variables are cumbersome to be added; these inclusions have to be performed by the third party company. In addition, the structure of the current supervisory system is more rigid, not allowing customization.

4. Case Study

The case study is a photovoltaic (PV) plant (Tesla Laboratory Experimental Photovoltaic Power Plant) connected to a battery energy storage system and to the loads. The solar plant is 37 kWp, located in Belo Horizonte, Brazil. Figure 12 shows the single-line diagram, showcasing some of the PV panels, the control room, and the battery banks.

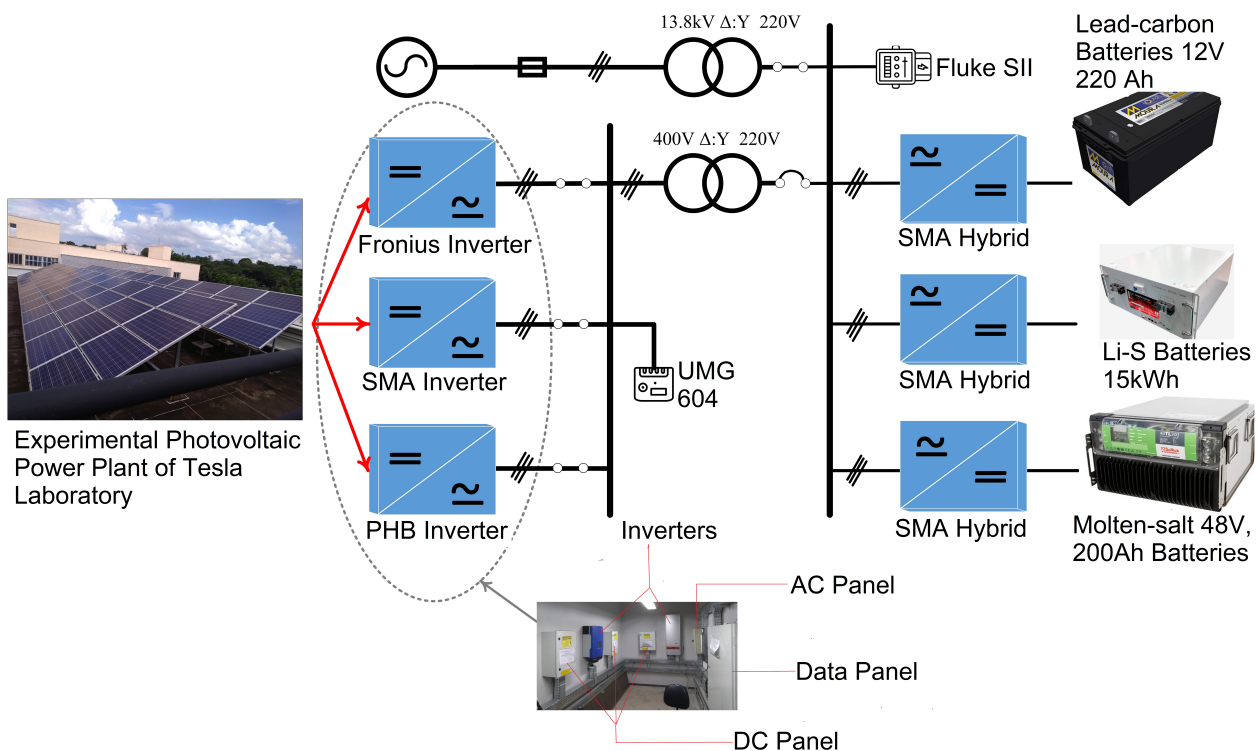


Figure 12. Single-line diagram of the solar power plant with the energy storage system.

The power plant has three power inverters: Fronius IG Plus 150V-3 (10 kW), SMA Sunny Tripower 12000TL (12 kW), and PHB20K-DT (20 kW), as seen in Figure 12—the last one was installed in May/2021. Since this project is a prototype, inverters from different manufacturers were installed in the power plant in order to investigate their differences. The photovoltaic plant is formed by 152 PV panels (Yingli 245P-32b, 245W), connected via inverters through a 45 kVA 400/220 V transformer in the university electrical grid.

The energy storage system is made of three independent battery banks and three independent single-phase hybrid inverters (SMA Sunny Island 8.0h). The latter are connected to the first, enabling the flow of data from/to the battery management system to/from the control room via local or remote commands/monitoring. This connection makes it possible to monitor the battery banks and to control the power flow between the solar power plant and the battery banks. The inverters are able to send commands to them to charge/discharge, adjusting the power flow.

Each battery banks presents a unique electrochemical technology, comprised by: 24 lead-carbon batteries pack, 6p4s of 12 V and 220 Ah, a molten-salt battery bank, 2p1s of 48 V and 200 Ah, and lithium-iron 48V battery bank, 5p1s of 48 V and 100 Ah. All the batteries are shown in Figure 12.

In order to obtain the measurements from the AC-side, a Janitza UMG 604 is connected to the grid. It is an energy meter that is able to provide various electric information: power, energy, frequency, current, and voltage. This device measures the output electric variables of the power plant, before the transformer. A Fluke SII (energy meter) is also installed. This one is connected in the 220 V busbar before the 13.8 k/220 V transformer.

5. Results and Analysis

The commercial supervisory system main screen could be viewed in Figure 13, demonstrating real-time main electrical, environmental data of the PV plant, and monetary value of the generated energy. The structure of the current supervisory system is stiff, not allowing expansions and alterations in the data format. There is also an evaluation page, seen in Figure 14, in which it is possible to plot any available variable along a specified period of time.

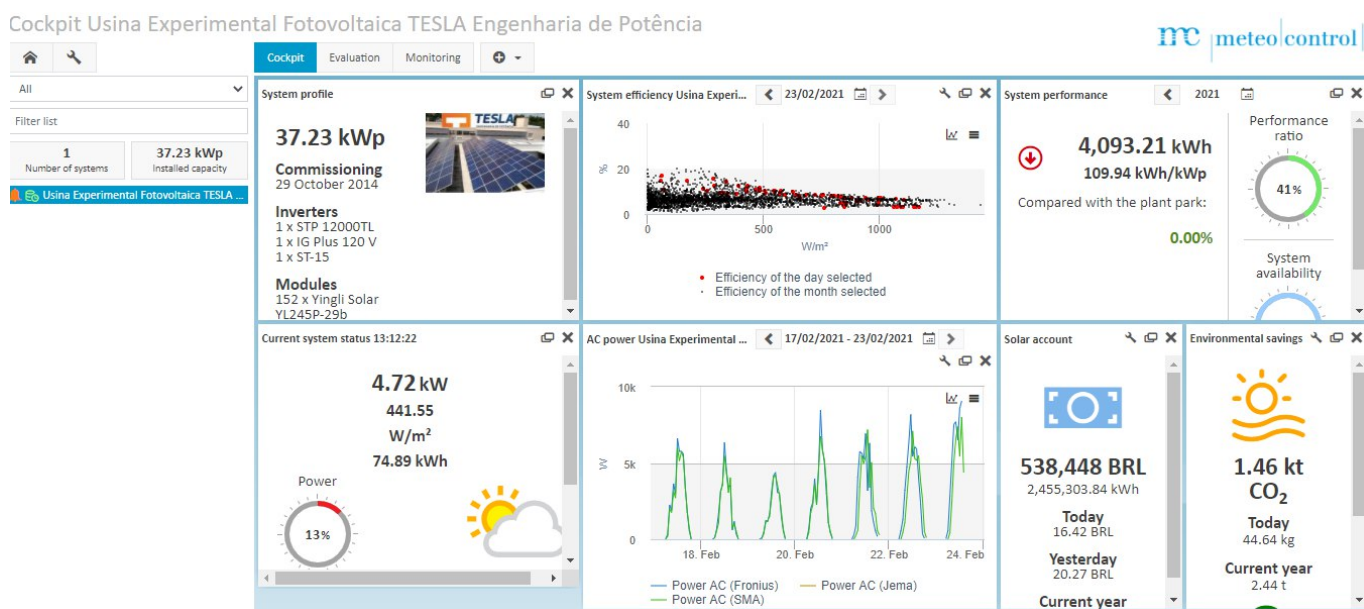


Figure 13. Main screen of the commercial supervisory system.

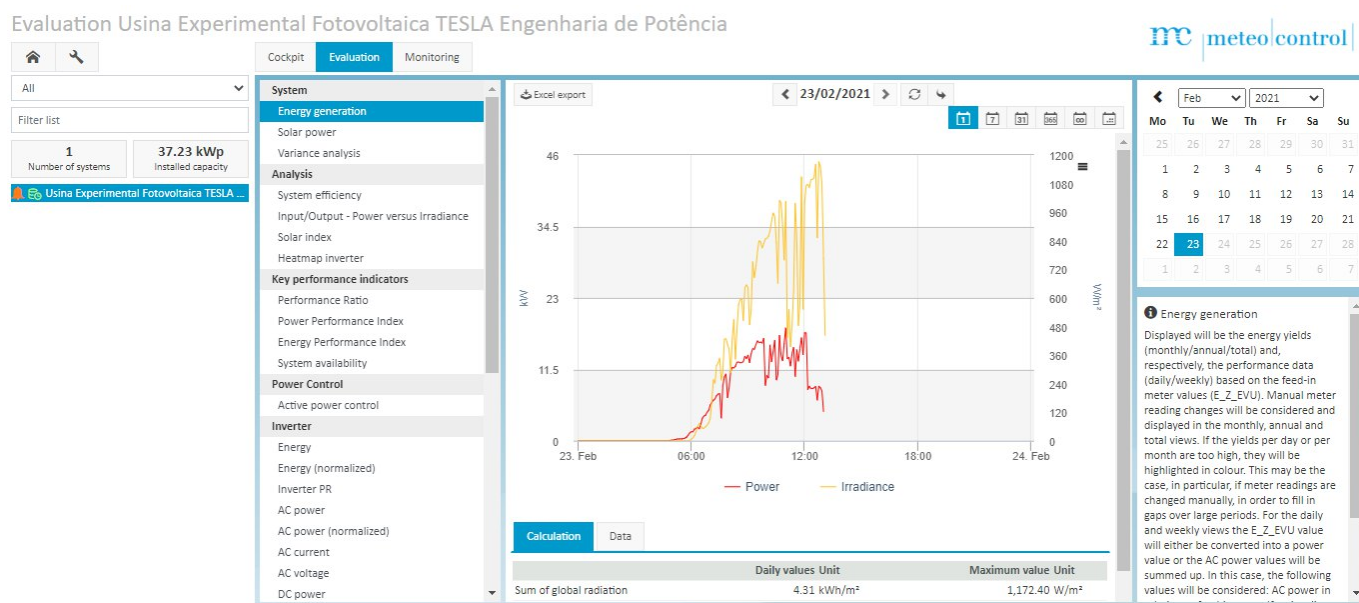


Figure 14. Graphs page of the current supervisory system.

The new supervisory system has been operating since Jan/2021. The box that contains the servers, the router, and the communication links can be seen in Figure 15.



Figure 15. Supervisory system hardware box.

The new supervisory system shows every electric measurement from the AC (from the energy meter and inverters) and DC buses (from the inverters) and the environmental variables made available by the solar meter station. Figures 16–18 are the new supervisory system. The new dashboards have a flexible and expandable structure in contrast with the old one. Any data and data format can be shown on the supervisory system for any user (all users and administrators). The developer is allowed to build any screen without restriction. The screens can be updated at a user-defined rate, allowing it to be more accurate with regard to the sampling rate of each measurement.

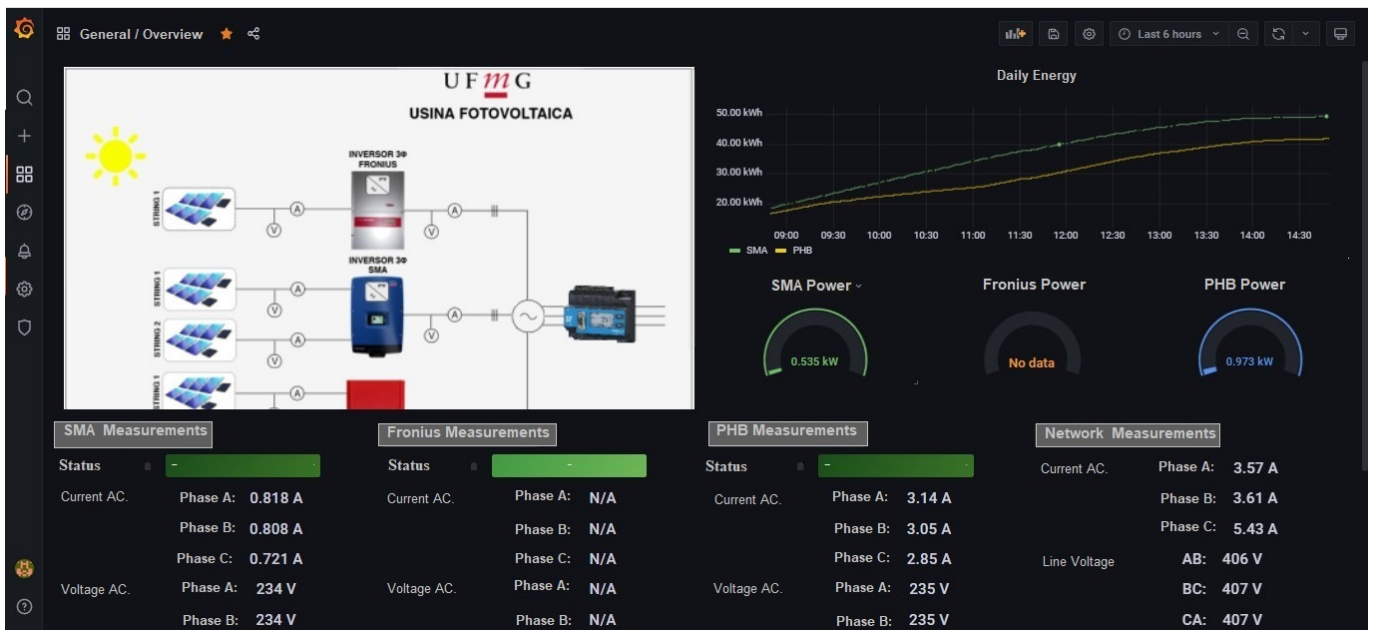


Figure 16. New supervisory system's main screen: PV plant overview.

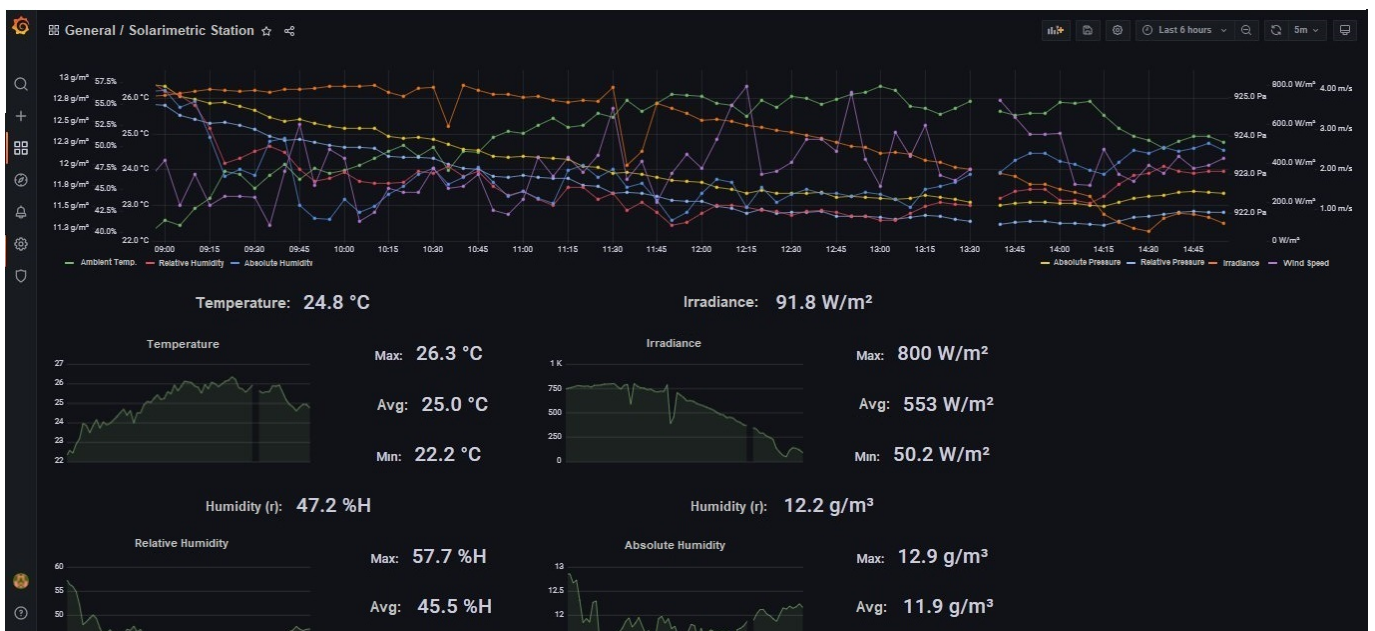


Figure 17. Screen for the measurements of the solar meter station.



Figure 18. AC-Grid measurements for the solar power plant.

The system can also be accessed in a Read-Only capacity for all users, including the ones that do not have administrator capabilities. In addition, all the users are able to download any data in a CSV file for further analysis. All users can choose the sampling rate and the period of time for the file. The administrator user is the only one that can modify the dashboards and the communication with the databases.

Is it possible to see that the new supervisory system is able to display real-time current measurements, also with the ability to display peaks, lows, and averages as seen in the latter. This supervisory system is also capable of presenting historical data and the user (any access level) can select any period of time that is needed.

The new presented system is not only more flexible and more adaptable than the current supervisory system, but the developer and the user can also select a finer resolution for the display of measurements (historical and real-time). It is possible to establish a sampling rate of 5 s if desired; the current supervisory system has a fixed sampling rate of 5 min for every variable. The longer polling time could mean data loss especially on days in which there is a lot of weather variation due to the low inertia of the PV panels (cloudy days). Figure 19 gives a more detailed view in order to show the behavior of the active power during the morning of a cloudy day to highlight the data loss of the original system due to the polling time. The new system detects higher power peaks and lower power valleys since its polling time is lower: 10 s versus 5 min.

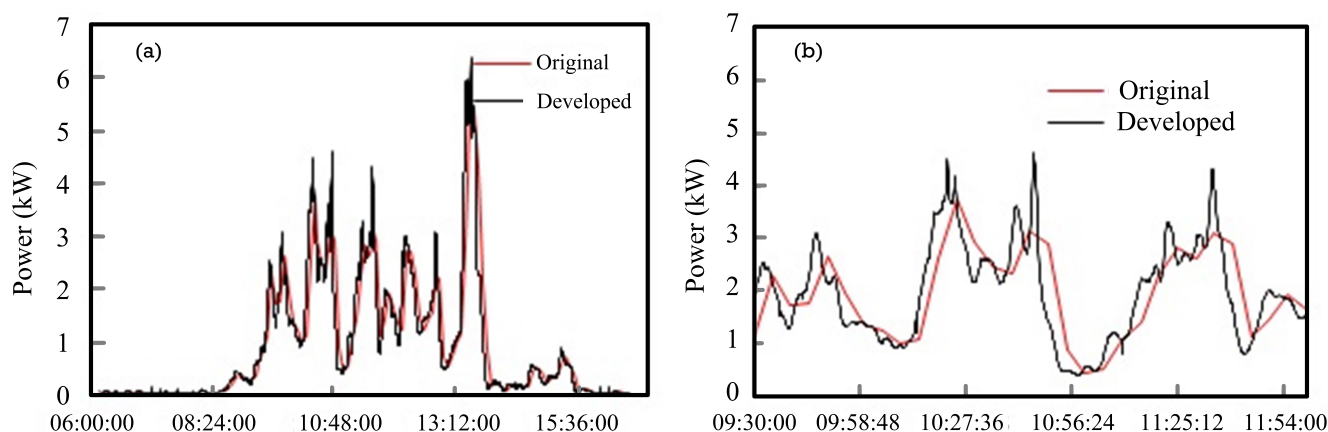


Figure 19. Active AC power measured at the SMA Inverter. (a) Example with a long sampling; (b) Example with a short sampling enabling detailed measurements.

Besides the technical advantages, the comparison between equipment costs can also be used to illustrate other advantages. While the developed system had a total cost of around USD 550, the commercial one was USD 4500, plus USD 25 for monthly subscription. The main reason for this difference is that the commercial equipment can be used for other variety of applications, which has no use for the solar system. In other words, the commercial solution provides ADC inputs and communication interfaces that are not compatible with the solar system.

To sum up the main comparisons between the two supervisory systems, Table 1 is shown.

Table 1. Highlights of the comparison between the previous and the current supervisory systems.

| Feature | Previous Supervisory | Current Supervisory |
|---------------|---------------------------|---------------------------------|
| Flexibility | Low | High |
| Price | USD 4500 + USD 25 monthly | USD 550 |
| Sampling Rate | Fixed at 5 min | Customizable for every variable |
| Compatibility | “Pseudo” Plug & Play | Requires configuration |

As it can be seen, the current supervisory system has advantages in three of the four considered features. The new system is more cost-effective, and the flexibility is considerably higher than the previous one, being more adaptable, and it is possible to expand more easily. The current supervisory system has a customizable sampling rate, which enables a better analysis of the behavior of each variable, making it more suitable for photovoltaic (PV) power plants since PV panels have low inertia. The feature that the previous supervisory system has over the current one is the compatibility since it is practically a system that only has to be installed to initiate the monitoring. The current one needed some configuration and adaptation.

6. Conclusions

The goal was to demonstrate the development steps of a supervisory system of a power plant and the energy storage system using IoT concepts, employing smart devices (network ready) and adapting the non-smart devices. The knowledge and techniques implemented for the case study can be transferred for the development of any supervisory system for any industrial/residential plant providing the means to establish a network.

Using smart devices and adapting the non-smart devices, every piece of information available from the micro-grid could be collected in the new supervisory system. The

communication, database, and application servers could be built in order to develop the supervisory system.

The new supervisory system is a more flexible solution. This system could be modified to integrate any modifications that may be necessary due to field changes in the power plant or in the energy storage system or due to new monitoring requirements. The developer/user of the new system could configure polling times more appropriate to each electrical and environmental variable according to one's goal.

When comparing this study to other papers in this field, it is possible to see that the other research works did not compare traditional supervisory systems with supervisory systems that implemented IoT devices. This paper did stress the advantages technically, analytically, and financially of the current system when confronted with traditional systems and with the previous supervisory.

To further improve the new system, it would be interesting to implement a prediction feature of the daily energy generation. This would allow for the scheduling of the operation of the energy storage system. Adding this feature, the power output of the power plant would be predicted based on the weather forecast for the coming day and the monitoring system could command the field devices to control the charge/discharge of the battery banks based on this forecast in order to supplement the generation when necessary, fine-tuning it in real time.

An optimization tool for self-management of the plant would also be interesting. The goal of the tool could be to maximize the efficiency of the process and lower costs and losses.

Author Contributions: Conceptualization, I.A.P. and T.A.C.M.; Writing—original draft, F.M.Q.S.; Writing—review & editing, M.B.E.K., I.A.P. and T.A.C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the help and the support of CAPES, CNPq, FAPEMIG, Pró-Reitoria de Pesquisa (PRPq)—UFMG, and Pró-Reitoria de Pós-Graduação (PRPG)—UFMG.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Camacho, E.F.; Samad, T.; Garcia-Sanz, M.; Hiskens, I. Control for renewable energy and smart grids. *Impact Control. Technol. Control. Syst. Soc.* **2011**, *4*, 69–88.
2. Montaña, D.A.M.; Rodriguez, D.F.C.; Rey, D.I.C.; Ramos, G. Hardware and Software Integration as a Realist SCADA Environment to Test Protective Relaying Control. *IEEE Trans. Ind. Appl.* **2018**, *54*, 1208–1217. [[CrossRef](#)]
3. Sičanica, Z.; Sučić, S.; Milašinović, B. Architecture of an Artificial Intelligence Model Manager for Event-Driven Component-Based SCADA Systems. *IEEE Access* **2022**, *10*, 30414–30426. [[CrossRef](#)]
4. Forero, N.; Hernández, J.; Gordillo, G. Development of a monitoring system for a PV solar plant. *Energy Convers. Manag.* **2006**, *47*, 2329–2336. [[CrossRef](#)]
5. Correia, D.; Tomé, P.; Costa, P.M.; Marques, L. Monitoring system for small sized photovoltaic power plants. In Proceedings of the 11th Iberian Conference on Information Systems and Technologies (CISTI), Gran Canaria, Spain, 15–18 June 2016; pp. 1–6. [[CrossRef](#)]
6. Lorenzo, G.D.; Araneo, R.; Mitolo, M.; Nicolai, A.; Grimaccia, F. Review of O&M Practices in PV Plants: Failures, Solutions, Remote Control, and Monitoring Tools. *IEEE J. Photovoltaics* **2020**, *10*, 914–926. [[CrossRef](#)]
7. Pereira, R.I.S.; Jucá, S.C.S.; Carvalho, P.C.M.; Souza, C.P. IoT Network and Sensor Signal Conditioning for Meteorological Data and Photovoltaic Module Temperature Monitoring. *IEEE Lat. Am. Trans.* **2019**, *17*, 937–944. [[CrossRef](#)]
8. Adhya, S.; Saha, D.; Das, A.; Jana, J.; Saha, H. An IoT based smart solar photovoltaic remote monitoring and control unit. In Proceedings of the 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC), Kolkata, India, 28–30 January 2016; pp. 432–436. [[CrossRef](#)]

9. Samosir, A.S.; Rozie, A.F.; Purwiyanti, S.; Gusmedi, H.; Susanto, M. Development of an IoT Based Monitoring System for Solar PV Power Plant Application. In Proceedings of the International Conference on Converging Technology in Electrical and Information Engineering (ICCTEIE), Bandar Lampung, Indonesia, 27–28 October 2021; pp. 82–86. [\[CrossRef\]](#)
10. Manoharan, H.; Teekaraman, Y.; Kirpichnikova, I.; Kuppusamy, R.; Nikolovski, S.; Baghaee, H.R. Smart Grid Monitoring by Wireless Sensors Using Binary Logistic Regression. *Energies* **2020**, *13*, 3974. [\[CrossRef\]](#)
11. Motlagh, N.H.; Mohammadrezaei, M.; Hunt, J.; Zakeri, B. Internet of Things (IoT) and the Energy Sector. *Energies* **2020**, *13*, 494. [\[CrossRef\]](#)
12. Silva, F.M.Q.; Filho, B.J.C.; Pires, I.A.; Maia, T.A.C. Design of a SCADA System Based on Open-Source Tools. In Proceedings of the 14th IEEE International Conference on Industry Applications (INDUSCON), São Paulo, Brazil, 15–18 August 2021; pp. 1323–1328. [\[CrossRef\]](#)
13. 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 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 4–7 November 2020; pp. 571–575. [\[CrossRef\]](#)
14. Mukaro, R.; Carelse, X.F. A microcontroller-based data acquisition system for solar radiation and environmental monitoring. *IEEE Trans. Instrum. Meas.* **1999**, *48*, 1232–1238. [\[CrossRef\]](#)
15. Şahin, S.; İşler, Y. Microcontroller-Based Robotics and SCADA Experiments. *IEEE Trans. Educ.* **2013**, *56*, 424–429. [\[CrossRef\]](#)
16. Benganem, M.; Maafi, A. Data acquisition system for photovoltaic systems performance monitoring. *IEEE Trans. Instrum. Meas.* **1998**, *47*, 30–33. [\[CrossRef\]](#)
17. Zedak, C.; Lekbich, A.; Belfqih, A.; Boukherouaa, J.; Haidi, T.; el Mariami, F. A proposed secure remote data acquisition architecture of photovoltaic systems based on the Internet of Things. In Proceedings of the 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, Morocco, 10–12 May 2018; pp. 1–5. [\[CrossRef\]](#)
18. Liu, Y. Research of Automatic Monitoring and Control Strategy of Photovoltaic Power Generation System. In Proceedings of the International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Hunan, China, 10–11 August 2018; pp. 343–347. [\[CrossRef\]](#)
19. Dumitru, C.-D.; Gligor, A. SCADA Based Software for Renewable Energy Management System. *Procedia Econ. Financ.* **2012**, *3*, 262–267. [\[CrossRef\]](#)
20. Leahy, K.; Gallagher, C.; O'Donovan, P.; O'Sullivan, D.T. Issues with data quality for wind turbine condition monitoring and reliability analyses. *Energies* **2019**, *12*, 201. [\[CrossRef\]](#)
21. Papageorgas, P.; Piromalis, D.; Antonakoglou, K.; Vokas, G.; Tseles, D.; Arvanitis, K.G. Smart Solar Panels: In-situ Monitoring of Photovoltaic Panels based on Wired and Wireless Sensor Networks. *Energy Procedia* **2013**, *36*, 535–545. [\[CrossRef\]](#)
22. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [\[CrossRef\]](#)
23. Vashi, S.; Ram, J.; Modi, J.; Verma, S.; Prakash, C. Internet of Things (IoT): A vision, architectural elements, and security issues. In Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 10–11 February 2017; pp. 492–496. [\[CrossRef\]](#)
24. Khan, L.U.; Yaqoob, I.; Imran, M.; Han, Z.; Hong, C.S. 6G Wireless Systems: A Vision, Architectural Elements, and Future Directions. *IEEE Access* **2020**, *8*, 147029–147044. [\[CrossRef\]](#)
25. Weber, R. Internet of Things—New security and privacy challenges. *Comput. Law Secur. Rev.* **2010**, *26*, 23–30. [\[CrossRef\]](#)
26. Lee, I.; Lee, K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Bus. Horizons* **2015**, *58*, 431–440. [\[CrossRef\]](#)
27. Swamy, S.N.; Kota, S.R. An Empirical Study on System Level Aspects of Internet of Things (IoT). *IEEE Access* **2020**, *8*, 188082–188134. [\[CrossRef\]](#)
28. Khan, R.; Khan, S.U.; Zaheer, R.; Khan, S. Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges. In Proceedings of the 10th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2012; pp. 257–260. [\[CrossRef\]](#)
29. Bertoldo, S.; Carosso, L.; Marchetta, E.; Paredes, M.; Allegretti, M. Feasibility analysis of a LoRa-based WSN using public transport. *Appl. Syst. Innov.* **2018**, *1*, 49. [\[CrossRef\]](#)
30. Petajajarvi, J.; Mikhaylov, K.; Roivainen, A.; Hanninen, T.; Pettissalo, M. On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. In Proceedings of the 14th International Conference on ITS Telecommunications (ITST), Copenhagen, Denmark, 2–4 December 2015; pp. 55–59. [\[CrossRef\]](#)
31. Sidorov, M.; Nhut, P.V.; Matsumoto, Y.; Ohmura, R. LoRa-Based Precision Wireless Structural Health Monitoring System for Bolted Joints in a Smart City Environment. *IEEE Access* **2019**, *7*, 179235–179251. [\[CrossRef\]](#)
32. Pasolini, G.; Buratti, C.; Feltrin, L.; Zabini, F.; de Castro, C.; Verdone, R.; Andrisano, O. Smart city pilot projects using LoRa and IEEE802.15.4 technologies. *Sensors* **2018**, *18*, 1118. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Xu, W.; Kim, J.Y.; Huang, W.; Kanhere, S.S.; Jha, S.K.; Hu, W. Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings. *IEEE Internet Things J.* **2020**, *7*, 298–310. [\[CrossRef\]](#)
34. Heller, A. The sensing internet—A discussion on its impact on rural areas. *Future Internet* **2015**, *7*, 363–371. [\[CrossRef\]](#)
35. Munir, M.S.; Bajwa, I.S.; Naeem, M.A.; Ramzan, B. Design and Implementation of an IoT System for Smart Energy Consumption and Smart Irrigation in Tunnel Farming. *Energies* **2018**, *11*, 3427. [\[CrossRef\]](#)

36. Alsayyari, A.; Kostanic, I.; Otero, C.E. An empirical path loss model for Wireless Sensor Network deployment in a concrete surface environment. In Proceedings of the IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON), Cocoa Beach, FL, USA, 13–15 April 2015; pp. 1–6. [[CrossRef](#)]
37. Bezerra, N.S.; Åhlund, C.; Saguna, S.; de Sousa, V.A. Temperature impact in LoRaWAN—A case study in Northern Sweden. *Sensors* **2019**, *19*, 4414. [[CrossRef](#)]
38. Silverio-Fernández, M.; Renukappa, S.; Suresh, S. What is a smart device?-a conceptualisation within the paradigm of the internet of things. *Vis. Eng.* **2018**, *6*, 3. [[CrossRef](#)]
39. Jiménez-García, J.-L.; Baselga-Masia, D.; Poza-Luján, J.-L.; Munera, E.; Posadas-Yagüe, J.-L.; Simó-Ten, J.-E. Smart device definition and application on embedded system: performance and optimization on a RGBD sensor. *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.* **2014**, *3*, 46–55. [[CrossRef](#)]
40. Tightiz, L.; Yang, H. A Comprehensive Review on IoT Protocols' Features in Smart Grid Communication. *Energies* **2020**, *13*, 2762. [[CrossRef](#)]
41. Vaca, N.; Garcia-Loro, F.; Martin, S.; Rodriguez-Artacho, M. Raspberry Pi Applications in Electronics and Control Laboratories. In Proceedings of the IEEE Global Engineering Education Conference (EDUCON), Tunis, Tunisia, 28–31 March 2022; pp. 1709–1713. [[CrossRef](#)]
42. Martin, S.; Fernandez-Pacheco, A.; Ruipérez-Valiente, J.A.; Carro, G.; Castro, M. Remote Experimentation Through Arduino-Based Remote Laboratories. *IEEE Rev. Iberoam. Tecnol. Del Aprendiz.* **2021**, *16*, 180–186. [[CrossRef](#)]
43. Ariza, J.Á.; Gil, S.G. RaspyLab: A Low-Cost Remote Laboratory to Learn Programming and Physical Computing Through Python and Raspberry Pi. *IEEE Rev. Iberoam. Tecnol. Del Aprendiz.* **2022**, *17*, 140–149. [[CrossRef](#)]
44. Mandanici, A.; Sarà, S.A.; Fiumara, G.; Mandaglio, G. Studying Physics, Getting to Know Python: RC Circuit, Simple Experiments, Coding, and Data Analysis With Raspberry Pi. *Comput. Sci. Eng.* **2021**, *23*, 93–96. [[CrossRef](#)]
45. Programming Tool-Node-Red. Available online: <https://nodered.org/> (accessed on 5 July 2022).
46. Sumangali, K.; Kiran, P. Comparative study of open source tools for internet of things. *Int. J. Pharm. Technol.* **2016**, *8*, 26235–26241.
47. Sicari, S.; Rizzardi, A.; Coen-Porisini, A. How to evaluate an Internet of Things system: Models, case studies, and real developments. *Softw. Pract. Exp.* **2019**, *49*, 1663–1685. [[CrossRef](#)]
48. Grafana Labs. Available online: <https://grafana.com/> (accessed on 5 July 2022).
49. Raimundo, R.J.; Rosário, A.T. Cybersecurity in the Internet of Things in Industrial Management. *Appl. Sci.* **2022**, *12*, 1598. [[CrossRef](#)]