

Article

Innovative Multi-Layered Architecture for Heterogeneous Automation and Monitoring Systems: Application Case of a Photovoltaic Smart Microgrid

Isaías González , Antonio José Calderón  and José María Portalo

Department of Electrical Engineering, Electronics and Automation, University of Extremadura, 06006 Badajoz, Spain; ajcalde@unex.es (A.J.C.); jportalo@alumnos.unex.es (J.M.P.)

* Correspondence: igonzp@unex.es; Tel.: +34-924-289-600

Abstract: Intelligent energy facilities, e.g., smart grids and microgrids are the evolution of traditional energy grids through digital transformation. These modern paradigms are expected to foster the utilization of renewable energies, sustainable development, and resilience of the power grid. A barrier found when deploying experimental smart grids and microgrids consists of handling the heterogeneity of the required hardware and software components as well as the available commercial equipment. Despite the fact that there is various architecture proposed in previous literature, it commonly lacks experimental validation, specification of involved equipment concerning industrial/proprietary or open-source nature, and concretization of communication protocols. To overcome such drawbacks, this paper proposes an innovative multi-layered architecture to deploy heterogeneous automation and monitoring systems for microgrids. The architecture is structured into six functional layers to organize the hardware and software equipment in an integrated manner. The open protocol Modbus TCP is chosen to harmonize communications, enabling the interconnection of equipment from industrial and energy scopes, indeed of open-source nature. An experimental photovoltaic-based smart microgrid is reported as the application case to demonstrate the suitability and validity of the proposal.

Keywords: microgrid; photovoltaic; multi-layered architecture; Modbus TCP; communication network; monitoring; open-source; automation



Citation: González, I.; Calderón, A.J.; Portalo, J.M. Innovative Multi-Layered Architecture for Heterogeneous Automation and Monitoring Systems: Application Case of a Photovoltaic Smart Microgrid. *Sustainability* **2021**, *13*, 2234. <https://doi.org/10.3390/su13042234>

Academic Editor: Mehdi Savaghebi

Received: 18 January 2021

Accepted: 10 February 2021

Published: 19 February 2021

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



Copyright: © 2021 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/>).

1. Introduction

The digital transformation of the energy industry is leading to the intelligent power grids, i.e., smart grids [1]. Microgrids also belong to this paradigm, comprising a set of distributed energy resources (renewable and/or nonrenewable), loads, energy storage means [2] as well as advanced automation and monitoring systems interconnected around a communication network [3]. Among the advantages that microgrids bring are the integration of renewable energies, environmental-friendly operation, stability, and improved performance [2]. In fact, smart microgrids are expected to significantly contribute to a more sustainable electricity delivery in future, decentralized paradigm of power systems [4]. Related modern concepts are the so-called Energy Internet [5], Energy Digitalization [5], the Internet of Energy (IoE) [6], and the Energy 4.0 [7], pivoting all of them around the interconnection of a variety of hardware and software nodes through communication networks for energy monitoring and management.

The path towards such frameworks requires the establishment of data communications among all actors, being able to work together seamlessly in an interoperable manner [8]. Regarding microgrids, reliable and secure communication networks are considered one of their main features [9] and an essential component to ensure a stable operation [10]. Therefore, the design of the communication network is a crucial topic for their development,

aimed at establishing communication among several components to monitor and automate the overall facility in real-time [11].

Examples of components are sensors, actuators, energy meters, data acquisition cards (DAQ), data loggers, controllers, human–machine interfaces (HMI), data servers, and computers with software for monitoring and analytics, from different vendors, with diverse communication interfaces, topologies, operative systems, and characteristics. In this situation, data acquisition, exchange, and processing are achieved in a distributed way between heterogeneous data sources and consumers; systems integration requires an enormous effort [12]. This heterogeneous ecosystem must be integrated through communication networks in order to exchange data for managing smart grids and microgrids [13].

In this sense, interoperability has been recognized as a crucial component to foster grid modernization [8]. According to the Institute of Electrical and Electronics Engineers (IEEE), interoperability is “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” [14].

Moreover, one of the most important areas in microgrid communication is interoperability between components [15]. Interoperability and the reliability of communication networks are among the challenges that monitoring systems are currently facing [16], often hard to solve by means of traditional solutions. In other words, the wide variety of protocols, interfaces, and proprietary systems to accomplish the communication and automation infrastructure represents the major barrier to the effective deployment of smart grids [17]. In fact, interoperability was identified from the very beginning as the main challenge for the deployment of smart grids [8]. The selection of the communication interface to achieve an effective interoperability constitutes a decision that seriously affects aspects such as economic costs, maintainability, expandability, security, and resilience [12]. Indeed, an expectation of future IoE scenario consists of “plug-and-play” interoperability, as proposed by Wu et al. [18].

Consequently, it is necessary to build a harmonization system between different communication standards to support their interoperability and improve the operation of the microgrids [9]. This lack of standardization implies that microgrids receive research efforts not only from the energy scope but also from automation, monitoring and communications disciplines [1]. In this regard, as pointed by Gomes et al. [19], currently, there is a significant gap between academic concepts and the physical implementations of microgrids. In a similar sense, experimental tests and demonstrative projects stand out as fundamental means to derive new methods and tools to address the issues of future energy grid [20].

Aiming at facilitating the orchestration and experimental deployment of this advanced infrastructure, innovative, multi-layered architecture is proposed in this paper. The architecture is structured into functional layers to design and implement energy automation and monitoring systems with seamless data sharing. Modbus TCP [21], a widespread protocol profusely applied in industry, enables the interconnection of equipment from industrial and energy scopes, including open-source technology.

As a proof of concept, to validate the proposed architecture, an application case devoted to the monitoring of a photovoltaic (PV) system and lithium-ion battery in an experimental smart microgrid is reported. Such a facility is framed within an R&D project for the deployment and digital replication of microgrids involving renewable energies and hydrogen.

The structure of the rest of the paper is as follows. Section 2 is devoted to carry out a brief literature review of layered architectures and communication protocols focused on the energy scope. Section 3 deals with the description of the involved equipment, both hardware and software, as well as the microgrid used as the experimental setup. Section 4 expounds the developed multi-layered architecture based on Modbus TCP and its main features. Section 5 reports the experimental validation of such architecture in a photovoltaic-based microgrid. The successful achieved results are shown in the sixth section. To conclude, the final remarks are provided.

2. Literature Review

Conceptual architecture or frameworks are critical in distributed systems and networks to facilitate the understanding of roles, locations, and levels of abstraction of different hardware, software, and networking components [22]. As indicated in [23], a growth of research publications on automation architectures discussing the involved actors, functionalities, and their interconnections can be witnessed. In the energy context, layered architecture is needed to evolve from the traditional grid to the intelligent grid and to handle interoperability [18].

A starting point is the automation pyramid, defined by the International Society of Automation (ISA) in the ISA-95 standard [24], which establishes five levels for the vertical and hierarchical integration of enterprise and control systems. Another proposal oriented towards automation systems is found in ref 13, where four-layered architecture based on the standard Open Platform Communications (OPC) is validated through facilities in energy, industry, and education scopes. Regarding the Internet of Things (IoT), layered architecture has been proposed over the last years, but there is no consensus on a single IoT architecture [16]. A common approach consists of three-layered architecture comprising perception, network, and application layers [25]. A different scheme is found in the work of ref [22], which developed the so-called IoTecture, conceiving five layers to orchestrate software, hardware, and communication technologies. Likewise, there are a multitude of conceptual frameworks focused on the IoT applied in the industrial environment, such as Industrial IoT [26]. In the Big Data scope, there is also a five-layered architecture proposal for microgrids reported in [27]. For the concept of block-chains, there are also proposals of layered architecture, for instance decomposed into six layers [28].

For energy automation and monitoring, a number of layered frameworks are found. In fact, a lot of research has been conducted in the area of microgrids [23]. A four-tier architecture for microgrids control is reported in [29]. Srinivasan et al. [30] consider eight layers to divide the functionalities for smart grids, covering the path from local control (lowest level) to global optimization (highest level). For renewable energy-based microgrids, three layers are considered in the communication architecture proposed in [10]. Organized in three levels, layered architecture to promote interoperability in the smart grid context is proposed in [18]. A conceptual organizational scheme for microgrids is defined in [4], considering four functional layers. To deploy advanced metering infrastructure in smart grids, three-layered architecture is reported in [31]. For PV plant monitoring, architecture based on three layers is described in [32]. A layer-based architecture comprising four interrelated levels is proposed in [33] for energy management in smart grids, including edge and fog computing IoT devices. The work in [34] establishes a two-layered scheme for communications in multi-microgrid systems. Considering four layers, an IoT-based architecture for monitoring distributed solar farms is presented in [35]. Based on IoT concept, researchers have designed a four-layer hierarchical control scheme to automate microgrid-enabled buildings [36]. Marzal et al. [6] implemented a four-layered scheme for microgrids management framed in the IoE paradigm. In [37], agent-based architecture for demand side management in the context of microgrids and smart grids is proposed, considering six layers to implement the required functionalities. Focusing on microgrids, hierarchical control architecture is proposed in [38], consisting of four control layers. Cagnano et al. [11] implemented a control system for microgrids with a hierarchical structure arranged in five layers. With the same number of layers, an IoT-enabled framework for energy automation is proposed in [5].

The Smart Grid Architecture Model (SGAM) [39] was created in 2012 by the CEN (Comité Européen de Normalisation), CENELEC (European Committee for Electrotechnical Standardization), and ETSI (European Telecommunications Standards Institute) as a conceptual framework for smart grid design and deployment. It comprises five layers for the so-called Interoperability Dimension within a three-dimensional representation. An example of architecture in terms of SGAM is reported in [23], a methodology for interoperability testing in smart grids according to the SGAM approach is provided in [8], and

an in-depth review about the SGAM application and implications can be found in [40]. It must be noted that the Reference Architecture Model for Industry 4.0 (RAMI 4.0) is derived from the SGAM, focusing on industrial automation towards the Industry 4.0 concept [40].

Given the abovementioned importance of the communication network, when designing conceptual architecture, a proper selection of the communication protocols is required. In this sense, there are a plethora of communication protocols for smart grids and microgrids. Coming from the industrial scope illustrative examples are Distributed Network Protocol 3 (DNP3), Open Platform Communications Unified Architecture (OPC UA), Modbus, and PROFIBUS. Framed in the IoT paradigm, there are various protocols such as Message Queue Telemetry Transport (MQTT), Common Object Request Broker Architecture (CORBA), Data Distribution Services (DDS), Constrained Application Protocol (CoAP), and Advanced Message Queuing Protocol (AMQP), just to name a few. In addition, the standard IEC 61850 was developed by the International Electrotechnical Commission (IEC) for substation automation and is oriented towards modern energy facilities. However, its practical application is still hindered by the lack of commercially available microcontrollers [11]. Most of the existing microgrids adopt protocols widely used for the industrial sector [11]. Among the existing options, Modbus is the most used due to its simplicity [11], and its wide availability in energy-related commercial equipment.

The architecture proposed in this paper is based on the industrial communication protocol Modbus TCP. Modbus was created by Modicon (currently Schneider Electric) in 1979, for data communication between the Programmable Logic Controllers (PLC) of the manufacturer. It rapidly became a de facto standard due to its simplicity and open features. There are two main versions of Modbus, the so-called RTU (Remote Terminal Unit) and TCP (Transmission Control Protocol). Known also as Modbus TCP/IP, it basically consists of the Modbus RTU protocol with a TCP interface that runs on Ethernet. This variant is the consequence of evolution and adaptation to new decentralized frameworks. This way, Modbus TCP combines the benefits of Ethernet networks, of the prevalent standard TCP/IP, and of the information representation independent of the manufacturer. In this sense, the modern version is considered as ideal protocol for situations which involve a multiplicity of devices and software linked together over TCP/IP [41]. There are hundreds of Modbus TCP-enabled devices and software suites in the market. Devoted to energy facilities, manufacturers such as Victron Energy [42] and SMA [43] include this communication in equipment for solar facilities. Indeed, a version of Modbus TCP named SunSpec Modbus has been developed and included in energy-related devices, e.g., inverters [44].

Within energy automation and monitoring, Modbus TCP has been applied in a number of scenarios. Kenner et al. [45] studied different schemes for gathering data from distributed energy analyzers, highlighting the scalability that Modbus TCP provides. In [17], infrastructure for the integration of legacy equipment into smart grids is developed, based on a middleware to handle different protocols such as Modbus. An IoT platform is proposed in [34] for energy management in multi-microgrid systems, with Modbus TCP used as the communication channel. Kim et al. [9] propose and simulate a platform to provide interoperability between microgrid components, including various communication standards, among which, Modbus is considered. In [46], an automated testbed for batteries is deployed, integrating a monitoring system and commercial solar inverters using Modbus TCP. In [19,37], a system for microgrid management based on multi-agents is presented, performing communication integration through Modbus TCP. An energy management system devoted to buildings is presented in [47], including Modbus TCP linkage. The work in [11] implements a control system for an experimental microgrid using Modbus TCP as the communication channel. An IoT-based supervisory system is presented in [48], where the master controller exchanges data with energy meters through Modbus TCP. In [11], a review of communication networks in existing microgrids is presented; Modbus TCP is used as protocol in many of them, which highlights its applicability within this field. Additionally, in [49], Modbus TCP implements data communication in the control system

of a microgrid studied as an alternative source to strengthen the station blackout power supply in nuclear power plants.

On the view of previous literature, the relevance and research interest of layered architecture for smart grids and microgrids has been clearly demonstrated. Nevertheless, research is still needed to bridge the gap between reference and use-case specific architectures [50]. Abstract reference architecture is technology-neutral, and there are a lack of concrete realization strategies and details [50].

In this respect, previous architecture is interesting and makes valuable contributions to advancements for energy digitalization, but presents certain limitations. Namely, three main boundaries have been detected: lack of experimental validation, no specification of the involved equipment concerning industrial/proprietary or open-source nature, and no concretization of communication protocols. Regarding this latter aspect, some frameworks rely on protocols that are not massively supported by industrial and commercial equipment, mainly MQTT. In other cases, two different protocols are considered, which hinder the handling of heterogeneity. Table 1 summarizes these features in the previously surveyed works. As can be observed, the architecture developed in this work is a novelty in the literature and overcomes the identified limitations. In other words, the present proposal contributes to addressing the gap between conceptual approaches and experimental facilities as well as to provide concrete strategies and details to deploy heterogeneous automation and monitoring systems for microgrids.

Table 1. Summary of the main features of related publications.

Work	Industrial/Proprietary Equipment	Open-Source Equipment	Experimental Validation	Concretization for Communications
4	Unspecified	Unspecified	No (only description)	No
5	Unspecified	Yes	No	Yes (NGSI)
6	Yes	No	No (emulation)	Yes (TCP/IP)
10	Unspecified	Unspecified	No (simulation)	No
11	Unspecified	Unspecified	No (only description)	Yes (Modbus TCP)
18	Unspecified	Unspecified	No (only description)	No
23	Unspecified	Unspecified	No (only description)	Yes (IEC 61850)
29	Unspecified	Unspecified	No (only description)	No
30	Unspecified	Unspecified	No (only description)	No
31	Unspecified	Unspecified	No (emulation)	Yes (MQTT)
32	No	Yes	No (only description)	No
33	No	Yes	Yes	Yes (MQTT)
34	Yes	Yes	No (simulation)	Yes (Modbus TCP and MQTT)
35	No	Yes	Yes	Yes (MQTT)
36	Unspecified	Yes	Yes	Yes (NGSI)
37	No	Yes	Yes	Yes (Modbus TCP and RESTful)
38	Unspecified	Unspecified	No (simulation)	No
39	Unspecified	Unspecified	No	No
Present Proposal	Yes	Yes	Yes	Yes (Modbus TCP)

IEC: International Electrotechnical Commission; MQTT: Message Queue Telemetry Transport; NGSI: Next Generation Service Interface; REST: Representational State Transfer; TCP/IP: Transmission Control Protocol/Internet Protocol.

3. Materials and Methods

In this section, the main equipment concerning both hardware and software is described. Furthermore, the photovoltaic-based smart microgrid used as the application case is also expounded.

3.1. Hardware and Software

- Raspberry Pi (RPi) is an open-source and single-board microcomputer developed by the Raspberry Pi Foundation of the University of Cambridge in 2012. This device provides huge computation resources and wide connectivity options at very low-cost.

Model 4 version B has been selected for the application case; it is the latest version and, hence, the most powerful. Namely, a processor ARM Cortex A72 at 1.5 GHz and a random-access memory (RAM) of up to 4 GB are the key features of this board. Additionally, USB 3.0 and Gigabit Ethernet ports are available.

- PLCs are electronic units hardened for factory-floor use with a long trajectory in industrial automation. The high-performance series S7-1500 manufactured by Siemens (Munich, Germany, Europe) has been chosen for the facility. Namely, the central processing unit (CPU) model S7-1516 is adequate for applications with high requirements concerning networking and processing speed. This device provides 5 MB of memory for data, 1 MB of memory for programs, a slot for a memory card, a port for PROFIBUS communication, and three ports for Ethernet connectivity. In addition, modules for digital and analog inputs/outputs (I/Os) provide the ability to connect a variety of sensors and actuators.
- Color Control GX (CCGX) is a controller manufactured by Victron Energy (Almere Haven, The Netherlands, Europe) devoted to energy-related equipment. This device exchanges data and commands with solar chargers, inverters, batteries, and other measurement appliances. Additionally, a screen provides a human-machine interface for in situ monitoring of the connected devices. Additionally, the exchanged data with energy equipment are accessible through this appliance, acting as a gateway of communication protocols such as Modbus or MQTT.
- The Decentralized Periphery Station (DPS) model ET200, manufactured by Siemens, acquires data of remote I/Os. A set of sensors are connected to this module which exchange data with a PLC through a PROFINET network.
- Arduino is an open-source hardware platform devoted to electronic prototyping. The model Mega 2560 is used to collect temperature measurements of PV modules by means of a network of digital temperature sensors.
- Battery management units (BMUs) are dedicated to handle the status and charge of lithium-ion batteries. They consist of an electronic board manufactured by BYD (Shenzhen, China).
- A charge controller regulates the power extraction from PV modules. The model BlueSolar 150/35, manufactured by Victron Energy, provides maximum power point tracking (MPPT) to extract the optimal power of the PV generator.
- An Ethernet switch device is required to interconnect equipment around an Ethernet network. Particularly, the unmanaged Gigabit Ethernet Switch GS308 manufactured by Netgear (Santa Clara, CA, USA) is used. This switch integrates 8 ports and provides a transfer speed of up to 1 Gbit/s.
- A number of sensors are used to measure the magnitudes required for a proper monitoring and management of the microgrid. Irradiance, temperature, voltage, current, etc. are sensed and acquired.
- Raspbian is the operating system of the RPi. It is based on the Linux distribution Debian, so it is open-source and freely available on the Internet.
- Python is an open-source programming language used in the present research to run scripts in the RPi. In addition, the Pymodbus library [51] is applied to establish Modbus TCP communication, and is expounded in Section 5.2.
- Grafana is applied to design highly customizable web-based graphical user interfaces GUIs composed of graphs displaying time-series. It is open-source software that runs as a web application which the user accesses online from devices such as personal computers (PCs), tablets, or smartphones.
- MariaDB is an open-source database management system (DBMS) for relational databases. It is used for data storage due to its lightweight, stable, and open features.
- Totally Integrated Automation (TIA) Portal is the software engineering framework of Siemens for industrial automation systems. In this research, it is used for programming the algorithm, managing I/Os, and configuring communications of PLC and DPS.

3.2. Photovoltaic Smart Microgrid

A smart microgrid hybridizing renewable energy and the energy carrier hydrogen acts as the application case of the proposal of the paper. This facility is a Direct Current (DC) microgrid for off-grid and environmentally-sustainable operation by means of PV-generated hydrogen, i.e., green hydrogen. Namely, a PV generator composed of 6 monocrystalline modules of 185 W and 24 V of nominal voltage (LDK Solar, Jiangxi, China) constitutes the renewable energy source. For hydrogen production, a Polymer Electrolyte Membrane (PEM) electrolyser, manufactured by Hydrogen Works (Madrid, Spain), provides up to 750 Ncm³/min of hydrogen flow. Hydrogen is accumulated by using metal-hydride canisters. For the contrary process, a PEM fuel cell uses the hydrogen to generate electricity by means of an air-cooled stack of 500 W manufactured by Horizon Fuel Cell Technologies (Singapore, Singapore). Short-term electricity storage is achieved through a lithium-ion modular battery of 5.12 kWh model B-Box, manufactured by BYD (Shenzhen, China). This device materializes the low voltage DC bus around which the rest of components are interconnected. Additionally, an electronic programmable load model 32,612 manufactured by Prodigit Electronics (New Taipei City, Taiwan) is supplied by the DC bus according to custom-designed load profiles. Figure 1 depicts the block diagram of the described components.

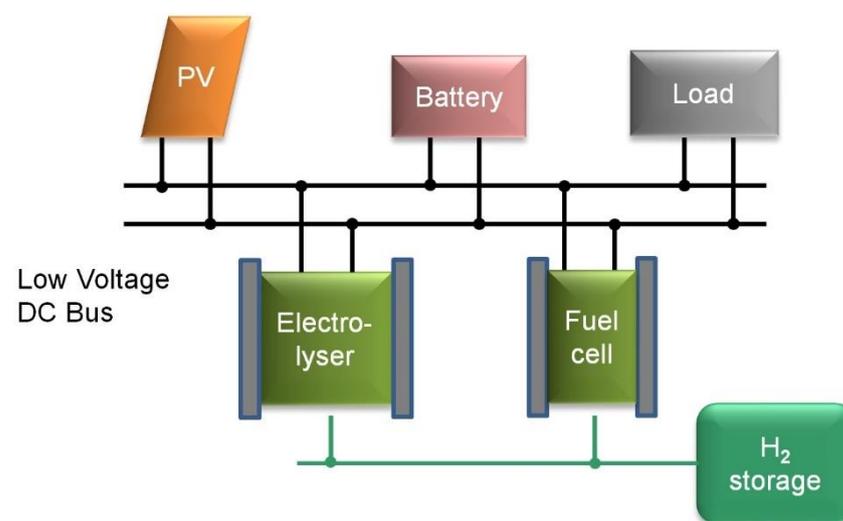


Figure 1. Block diagram of the photovoltaic (PV) smart microgrid.

Concerning automatic operation of the microgrid, an industrial PLC is responsible for managing the energy flows between the aforementioned components. For this purpose, a number of sensors measure the main magnitudes of the microgrid, and an energy dispatching algorithm is implemented by the PLC in order to power the connected load using the PV energy. The flowchart depicted in Figure 2 illustrates the decision sequence. If the battery is fully charged and there is a PV energy excess, the electrolyser (EL) is switched on to produce hydrogen. On the opposite situation, if the battery has a very low state of charge, e.g., it is discharged, the fuel cell (FC) is activated and converts hydrogen into electricity, supplying the demand of the load. This way, the generation of hydrogen from a PV source and its consumption enables autonomous behavior.

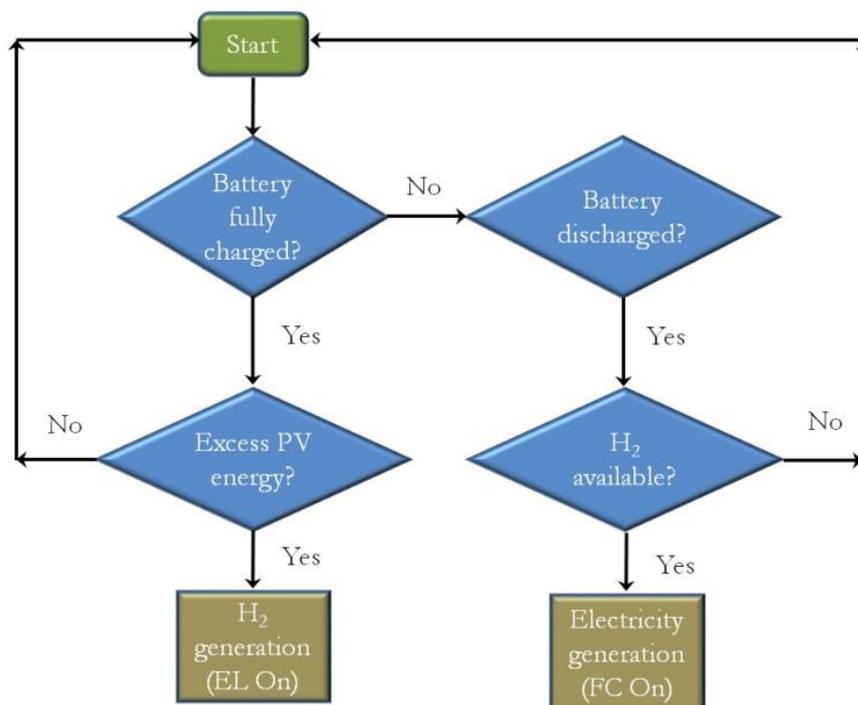


Figure 2. Flowchart of the energy dispatching algorithm of the microgrid.

4. Proposed Multi-Layered Architecture

An innovative multi-layered architecture for energy automation and the monitoring of microgrids is presented. A functional layout is defined to organize the software, hardware, and communication technologies in a coordinated, versatile, and interoperable manner. The proposal is structured in six layers, as shown in Figure 3.

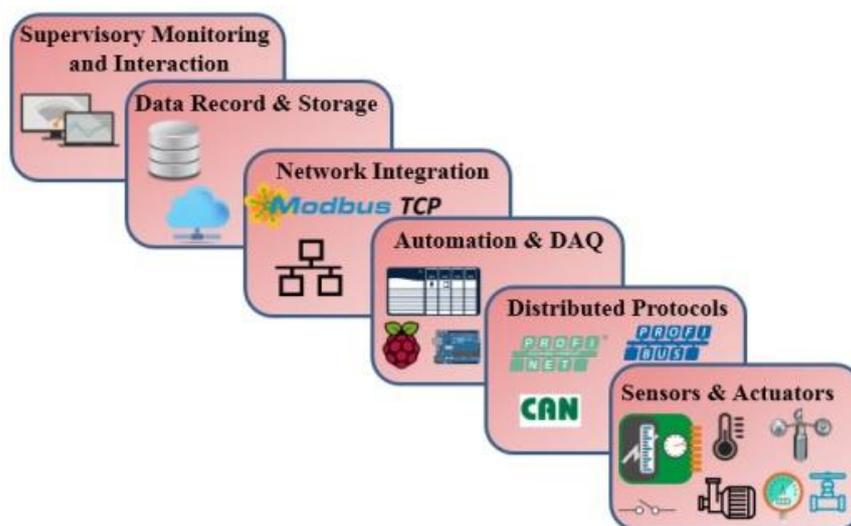


Figure 3. Multi-layered architecture for energy automation and monitoring of microgrids.

The so-called Supervisory Monitoring and Interaction layer provides visualization functionalities through graphical data representation, alarm management, as well as direct interaction with the user. They must be informed in real-time of the current status of the microgrid, as well as of previous operation, by means of numerical and trend graphs, displayed on a graphical user interface (GUI). The main data for this layer are information about the status of sensors and actuators physically placed in the microgrid; the

corresponding I/O signals are exchanged through the conceived layers. The monitoring function can be accessed in local and/or remote manner, depending on the capabilities of the chosen software. For example, Siemens WinCC allows local connection, whereas some middleware would be required to enable remote desktop connections for online access. Other commonly used software for plants involving renewable energies is LabVIEW, which is generally applied for local monitoring [52]. In the case of open-source suites, they commonly include remote connection through web interfaces. ScadaBR and Grafana are examples of this type of options. In fact, given the widespread connectivity through the network, remote monitoring is a requirement for modern digitized energy facilities [3].

The Data Record & Storage layer is responsible for accumulating the exchanged information. This function can be implemented by means of in-house or cloud-hosted databases, or even a combination in order to avoid data losses. In traditional monitoring systems, such data storage was commonly performed by the supervisory/monitoring system. However, in modern infrastructure, the volume of sensed magnitudes and exchanged information to be stored imposes an accessible and structured accumulation system. Databases allow data logging and storage in time-series or relational forms. In fact, traditional monitoring and supervisory software are moving towards Structured Query Language (SQL) databases [15], and open-source packages include in-built connectivity with such types of databases. The recorded information can be used for real-time monitoring, but also for further treatment towards forecasting or performance analyses using sophisticated algorithms.

The fourth level is called the Network Integration layer and harmonizes communications around the Modbus TCP protocol. As indicated in the Introduction, Modbus is the open protocol most commonly applied in the industrial environment. Particularly, Modbus TCP is the backbone of the communication network in order to establish an open and reliable transmission of information among the whole infrastructure system. Modbus TCP is a client/server protocol and uses the standard port 502. The client entity initiates the communication by sending a request to the server to transfer data. Once a connection is established, the server provides the client with the queried information until the client finishes the connection. Each involved device can play the role of client or server, depending on the information flow. Using this protocol, devices, either client or server, talk to each other via the IP addresses [34]. Moreover, the same device can develop both roles facing different equipment. Client and server do not know the physical/hardware or software nature of the counterpart; they are abstract from the manufacturer, model, or features. Only the protocol is used to enable the transmission of data messages.

An Ethernet network materializes the Network Integration layer, whereas the Modbus TCP protocol corresponds to the Application level of the well-known Open Systems Interconnection (OSI) model for networked communications. Using an Ethernet network facilitates the replacement or addition of devices [53], especially given the fact that this kind of network is the prevalent trend for Information Technology (IT), IoT-enabled equipment, and modern automation systems [54]. Both wired and wireless communications are considered in the proposal. Nonetheless, wired networks are preferred because of reliability and security reasons [4].

Indeed, a relevant characteristic of Modbus TCP is that it is supported by both proprietary and open-source hardware/software, so different equipment can seamlessly share information enabled by this protocol. For example, Modbus is natively supported by most industrial monitoring/supervisory software such as NI LabVIEW, Siemens WinCC, or Wonderware InTouch. On the other hand, there are libraries for Modbus communication for different programming languages such as Java, C or Python. Furthermore, IoT open-source hardware (Arduino, RPi, etc.) and software (Node-RED, ScadaBR, etc.) also implement Modbus communication through freely available libraries or in-built modules.

The Automation & DAQ layer groups equipment for automation and control such as PLC and PC-based controllers, as well as peripheral devices for remote I/O data acquisition purposes such as DAQ, DPS, HMI panels, etc. Communication between devices inside this layer can take place through the Network Integration layer or by means of the Distributed

Protocols level. In this sense, data flows can take place in vertical and/or in horizontal directions. Horizontal data flow takes place in the same layer, concretely; it is enabled between devices of the Automation & DAQ layer by means of the communication provided by the Network Integration level. For example, information allocated in a PLC can be shared with another device placed in the same layer as a DAQ. In the vertical direction, a monitoring interface is able to gather data from a PLC through the Modbus TCP linkage.

The layer named Distributed Protocols is envisioned to host digital communications materialized by industrial fieldbuses different from Modbus TCP (PROFINET, PROFIBUS, EtherCAT, etc.) and by modern protocols such as MQTT or OPC UA. This way, master nodes can retrieve data from slave nodes, commonly featured by minor processing capabilities. For example, a PROFINET linkage between a PLC and a DPS belongs to this layer.

The bottom level, the Sensors & Actuators layer, groups all those sensors, actuators and other instruments needed for automation and monitoring of the physical facility. Examples of devices within this layer are temperature probes, pyrometers, current sensors, relays, contactors, electro-valves, and so on.

As indicated in the Section 2, there are a lack of concrete realization strategies and details concerning layered architecture [50]. To overcome this limitation, the present proposal relies on a specific communication protocol (Modbus TCP) for the Network Integration layer. Interoperability issues are solved, facilitating the systematic design and deployment of the communication network. Indeed, with a practical view, in the market it is easy to find power meters, power electronic devices, data loggers and gateways with embedded Modbus TCP interface. Therefore, the selection and acquisition of equipment is facilitated due to the widespread integration of Modbus TCP in available commercial and open-source equipment.

The proposal is appropriate for modern devices, as well as for legacy automation equipment. The ability to interconnect older-type equipment is an important benefit because it reduces investments by reutilizing already existent devices, for instance PLC a with remaining lifespan. As asserted in [47], the adoption of Modbus protocol provides a wide range of implementation options without replacing the existing sensor and actuator devices.

As has been highlighted, both proprietary and open-source technologies can co-exist, interconnected under the proposed architecture. In this sense, the integration of proprietary industrial and open-source equipment brings diverse benefits and empowers the development of innovative scenarios such as microgrids [1].

Furthermore, despite the fact that the presented framework is oriented towards microgrids, it is also applicable for facilities in industrial environments involving heterogeneous equipment for automation and monitoring. In this regard, in order to take into account the decentralization trend of modern systems (IoT, Industry 4.0), this proposal is not a rigid hierarchical layout. It is suitable for both centralized and decentralized control schemes.

5. Experimental Validation

The validation of the proposed architecture was carried out over an experimental PV microgrid which involved equipment of both proprietary and open-source nature. Figure 4 depicts the application of the multi-layered architecture to the microgrid described in Section 3.2. Particularly, equipment devoted to automating and monitoring the PV–battery subsystem is orchestrated and deployed according to the layers defined in such architecture.

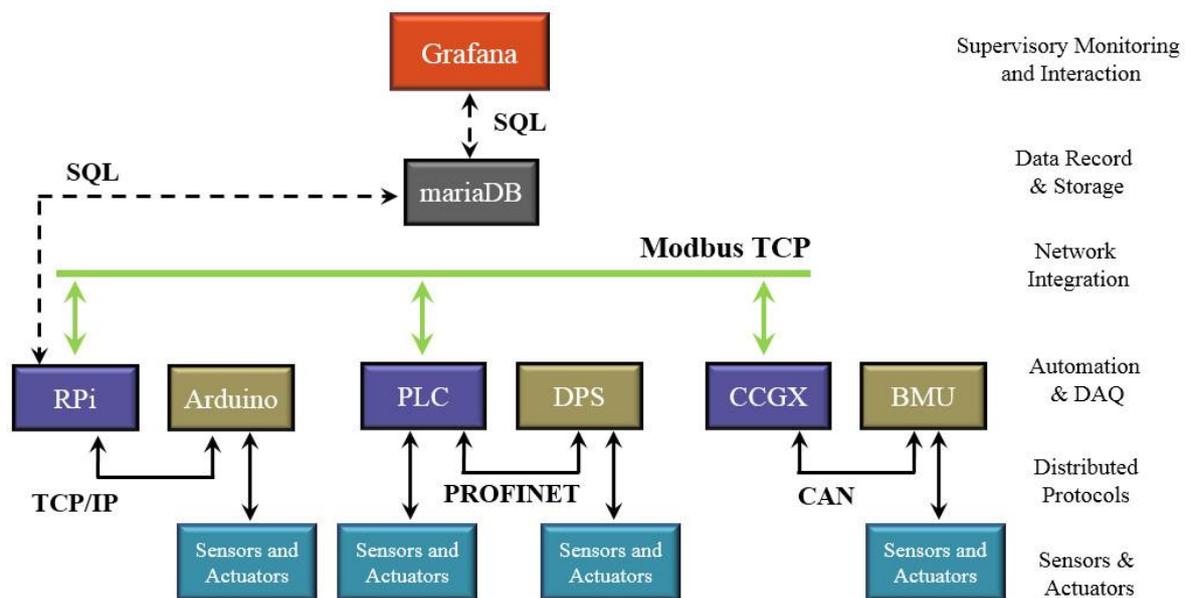


Figure 4. Multi-layered architecture based on Modbus TCP applied to an experimental smart microgrid.

A number of sensors and actuators were accommodated in the Sensors & Actuators layer. For instance, this level included devices to measure solar irradiance in the inclined plane of the PV modules, voltage and current sensors, Pt-100 probes, as well as actuators such as relays. It must be noted that weather measurements were carried out by a Pt-100 probe devoted to sense ambient temperature and an anemometer. Both magnitudes affect the temperature of the PV modules and contribute to analyzing their situation and performance. Table 2 lists the magnitudes and the corresponding sensors.

Table 2. List of magnitudes and sensors in the experimental setup.

Magnitude	Sensors
Irradiance	1 pyrometer
Wind speed	1 anemometer
Ambient temperature	1 Pt-100
Voltage	2 potentiometric voltage dividers
Current	8 Hall-effect sensors
Temperature of PV modules	6 Pt-100 and 12 digital sensors

The Automation & DAQ layer was composed of devices devoted to gathering data and operating the energy equipment. Concretely, a PLC is the main controller of the microgrid, making control decisions to manage the energy flows, generation, and consumption. A DPS was placed in the roof close to the PV generator to collect data from the modules. Other appliances within this layer were a MPPT solar charger and a BMU for lithium-ion batteries. Both were connected to exchange data with a controller, the so-called CCGX, for their harmonized operation. Open-source hardware was also incorporated in this layer, namely the microcomputer RPi and the microcontroller Arduino. The latter retrieved information from a set of digital temperature sensors, feeding it to the RPi, which stored it jointly with other relevant data shared through the Network Integration layer.

In this sense, Modbus TCP over an Ethernet network served as a physical means to implement the Network Integration layer. Above all, information was shared through this layer between three devices which belonged to the Automation & DAQ layer. Namely, the RPi acted as client of the PLC and of the CCGX; consequently, these latter devices played the role of Modbus servers.

As can be observed in Figure 4, the Distributed Protocols layer hosted four different communication protocols. RPi and Arduino shared information by means of a TCP/IP linkage. The well-known industrial bus PROFINET was used for the exchange of I/O signals between the DPS and the PLC. A Controller Area Network (CAN) bus was used to interconnect the BMU with the CCGX. The MPPT charger was linked to the CCGX through a proprietary bus, the so-called VE.Direct. This device has not been included in the previous figure for a clearer depiction.

Concerning the Data Record & Storage layer, MariaDB was the DBMS responsible for storing operational data. This entity was hosted by the RPi, thus implementing an in-house local cloud. This configuration provided a reduction in delays in recording measurements received from the data collecting equipment. Another advantage consisted of eliminating dependencies on external web hosting services.

The open-source Grafana suite implemented the Supervisory Monitoring and Interaction layer. Grafana played the role of a web-based monitoring interface, feeding the operator with the gathered information in the form of time-series charts in a user-friendly manner. The database was accessed by the Grafana dashboards to query the information of the graphical and numerical elements displayed. The linkage between the RPi, the database, and the Grafana GUI is represented in Figure 4, with a dotted line representing a logical connection within the RPi by means of SQL.

Some papers mention the utilization of Modbus TCP, but with scarce details about the experimental implementation and application. Therefore, in order to facilitate the deployment of the proposed architecture, insights on the configuration of the main actors are provided in next Sections, namely the CCGX, the RPi, and the PLC.

5.1. Configuration of Modbus Communication in CCGX

As described above in Section 3.1, this proprietary device acts as gateway enabling the exchange of information from the solar charger and BMU with equipment of different manufacturers. This device includes an Ethernet port, so it is easily integrated in the network where both the RPi and PLC are connected. A fixed IP address is also configured to establish a reliable and stable communication.

In order to share data, the CCGX can be configured to use MQTT or Modbus TCP protocols, with the latter selected to adopt the proposed architecture. To this purpose, it is required to activate this communication service within the Services option. Figure 5 shows this detail of the screen displayed by the CCGX.

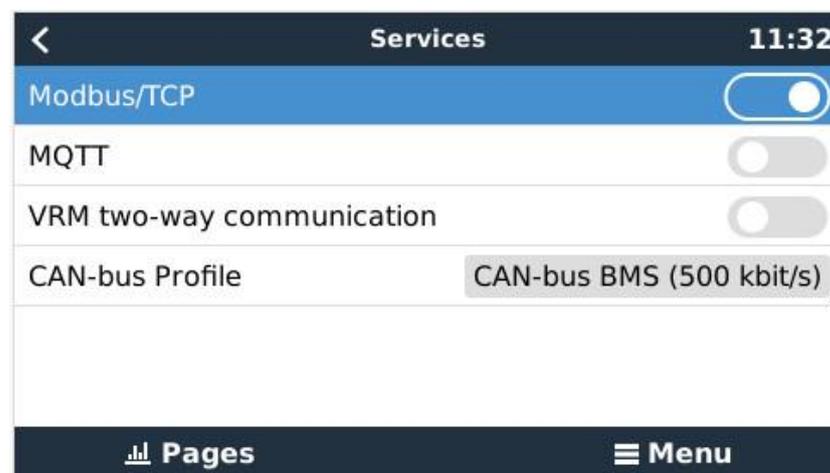


Figure 5. Modbus TCP service activation in the CCGX configuration menu.

5.2. Configuration of Modbus Communication in RPi through Python

The RPi acts as the Modbus client of both the PLC and the CCGX. Namely, on the one hand, the microcomputer shares data with the PLC concerning the sensors directly connected to it or through the DPS. On the other hand, measurements related to the battery and to the PV generator are also logged from the CCGX.

To exchange data via Modbus, the library Pymodbus is applied to share data over Modbus TCP using Python language. Firstly, a connection to the destination host through the default port is established. After that, the values of a certain number of registries from an address of the unit are read. Each host can include different units with information, being identified by a number; for instance, the CCGX uses the number 247 for the data from the solar charger and 225 for the BMU. The next step consists of sending the information to the database. For this aim, a connection is established with the device where the database is hosted, after which an SQL command inserts the data in the corresponding table. Figure 6 contains a flowchart to illustrate the described steps.

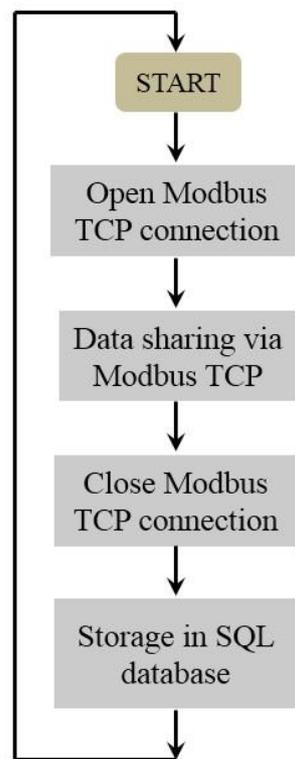


Figure 6. Flowchart of steps for sharing data over Modbus TCP in the RPi.

To exemplify these sequential stages, the main code of Python is described in Algorithm 1. To begin with, the libraries for Modbus communication (*ModbusTcpClient*) and for connection to the database (*mysql.connector*) are imported. Later, an instance *ModbusClient* is created for the CCGX, indicating its IP address and the default port for Modbus. After that, the connection is established; the magnitudes of interest within the memory of the Modbus server are read by means of the instruction *client.read_holding_registers*. For example, one register (16 bits) starting at address 266 of the unit 255 (BMU) is read and stored in the variable *batstategx_r* in order to retrieve the state of charge (SoC) of the battery. After reading all data, the connection with the destination host, e.g., the CCGX, is closed.

Algorithm 1. Modbus communication through Python.

```

from pymodbus.client.sync import ModbusTcpClient as ModbusClient
import mysql.connector as mariadb
client = ModbusClient(host = '192.168.99.191', port = 502)
client.connect()
batstategx_r = client.read_holding_registers(266, 1, unit = 225)
...
client.close()
client = ModbusClient(host = '192.168.99.143', port = 502)
client.connect()
inclirradiance_r = client.read_holding_registers(0, 1, unit = 1)
...
client.close()

```

The same procedure is performed to gather data from the PLC. A *ModbusClient* instance is created indicating the IP address and the port, after which the communication is established. Magnitudes of interest are shared in the same way.

5.3. Configuration of Modbus Communication in PLC

TIA Portal includes configuration modules to establish Modbus TCP connections, both for client and server roles: MB_Client and MB_Server, respectively. The PLC acts as server so the RPi can read the memory position shared via Modbus, i.e., the RPi plays the role of Modbus client.

A flowchart of the PLC sequential operations is shown in Figure 7. Firstly, local and remote I/O signals are collected, after which the corresponding data processing is carried out, e.g., scaling and normalizing for analogue measurements. Later, the accumulation of these data is performed in a data block of the local memory. Once the data block is completed, the information is shared through Modbus by calling the MB_Server block. This way, the client (RPi in this case) is able to read the stored signals, establishing a Modbus TCP network communication. After this step, the PLC executes the rest of operations required to manage the microgrid behavior.

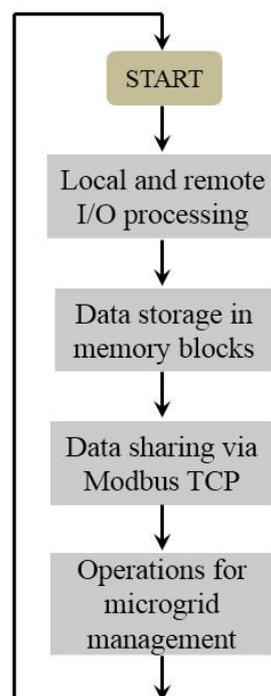
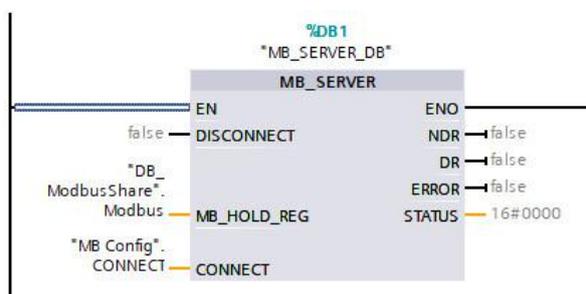


Figure 7. Flowchart of the PLC operations.

For the Modbus server role, the memory positions to be shared must be assigned to the input MB_HOLD_REG of the block MB_Server (Figure 8a). Thus, those positions are made accessible to the Modbus client. Additionally, in the input named as CONNECT, it is required to specify a data block named as MB Config, where a set of parameters are defined to establish the communication (Figure 8b). The most representative parameters are now briefly commented. To begin with, a value of 11 is assigned to ConnectionType for TCP linkage. The IP address of the device which acts as the Modbus client is specified by means of an array of four bytes, ADDR. As can be observed, for illustrative purposes, “0” has been introduced for each byte of the address, so any client is allowed to start this communication. Finally, the default port for Modbus, 502, is kept. Further details about the involved communication parameters can be found in the manufacturer documentation.



(a)

	Name	Data type	Start value
1	Static		
2	CONNECT	TCON_IP_v4	
3	InterfacelId	HW_ANY	64
4	ID	CONN_OUC	1
5	ConnectionType	Byte	11
6	ActiveEstablished	Bool	false
7	RemoteAddress	IP_V4	
8	ADDR	Array[1..4] of Byte	
9	ADDR[1]	Byte	0
10	ADDR[2]	Byte	0
11	ADDR[3]	Byte	0
12	ADDR[4]	Byte	0
13	RemotePort	UInt	0
14	LocalPort	UInt	502

(b)

Figure 8. Configuration details for the PLC communication: (a) block MB_Server; (b) parameters for Modbus TCP connection.

6. Results and Discussion

In this section, the most illustrative experimental results are reported as a proof of the validity of the proposed architecture. To begin with, the physical implementation of the equipment in laboratory and roof are expounded. Later, the data sharing through the proposed multi-layered architecture is proved by means of the monitoring system Grafana. Finally, the achieved results are discussed in Section 6.2.

6.1. Results

Figure 9 shows the experimental setup in the laboratory. As can be observed, the Ethernet switch, the MPPT charger, the RPi, the PLC and the CCGX were placed on the same metal sheet panel.

The sensors placed in the roof to measure the PV generator main magnitudes are seen in Figures 10 and 11. The mounting angle of the irradiance sensor was the same as that of the photovoltaic modules, in such a way that the direct irradiance actually received by these modules was measured. Concerning temperatures, sensors are adhered to the back plane of the PV modules. Two DS18B20 digital temperature sensors were placed in the diagonal of the modules, whereas a Pt-100 temperature probe was located in the center area.

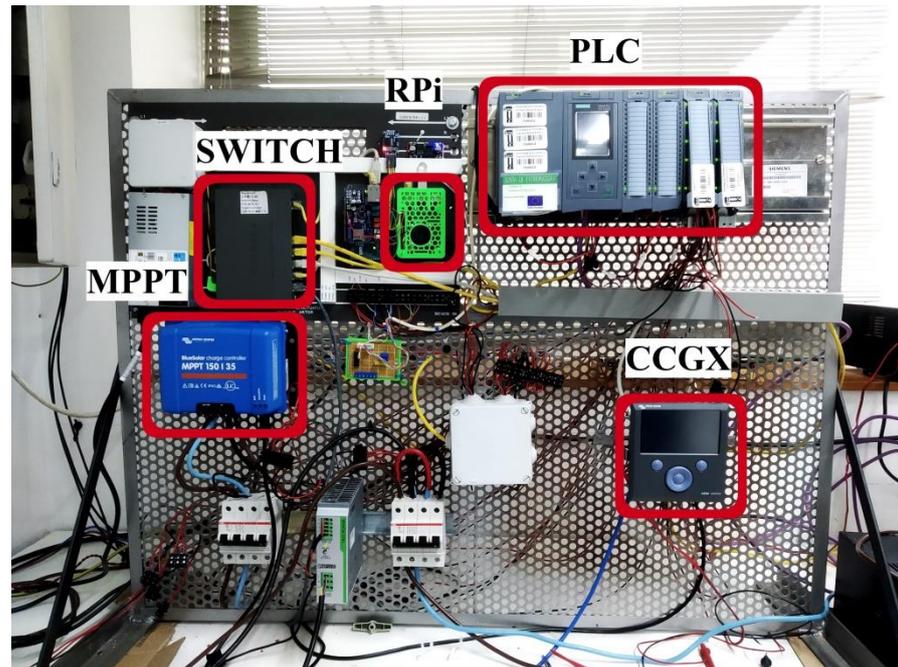


Figure 9. Experimental setup in laboratory.

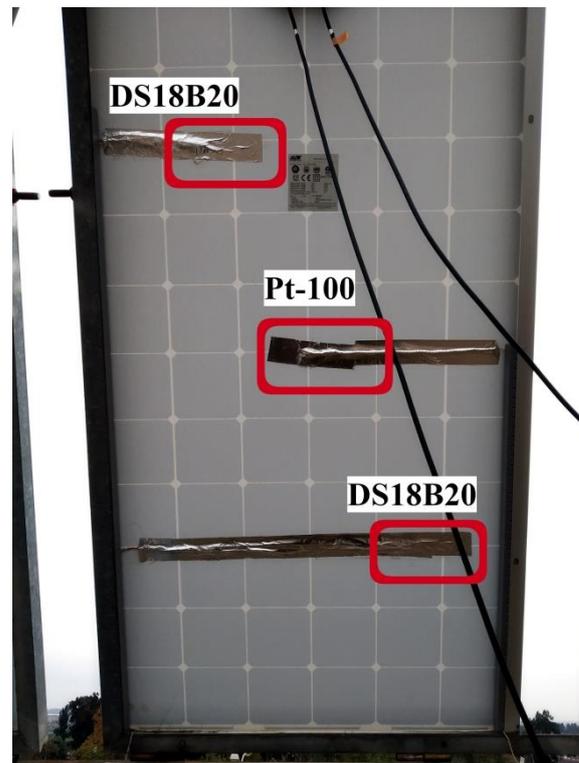


Figure 10. Experimental assembly of temperature sensors in PV modules.

Diverse dashboards have been built using Grafana to provide an intuitive interaction with the user through the GUI. Among them, one was devoted to take advantage of the synergy of process signals provided by the multi-layered architecture. In other words, complementary magnitudes are shown together for a proper tracking of the smart microgrid behavior. The user/operator can see at a glance the magnitudes shared via Modbus in real-time; indeed, the data source is indicated between parentheses in the legend of each

magnitude. They are also able to select previous time intervals because the corresponding data are stored in the DB; thus, it is rapidly displayed. Figure 12 shows the GUI composed of numerical data and time series so the user/operator can see at a glance the situation of the microgrid.



Figure 11. Irradiance sensor in the inclined plane of the PV modules.

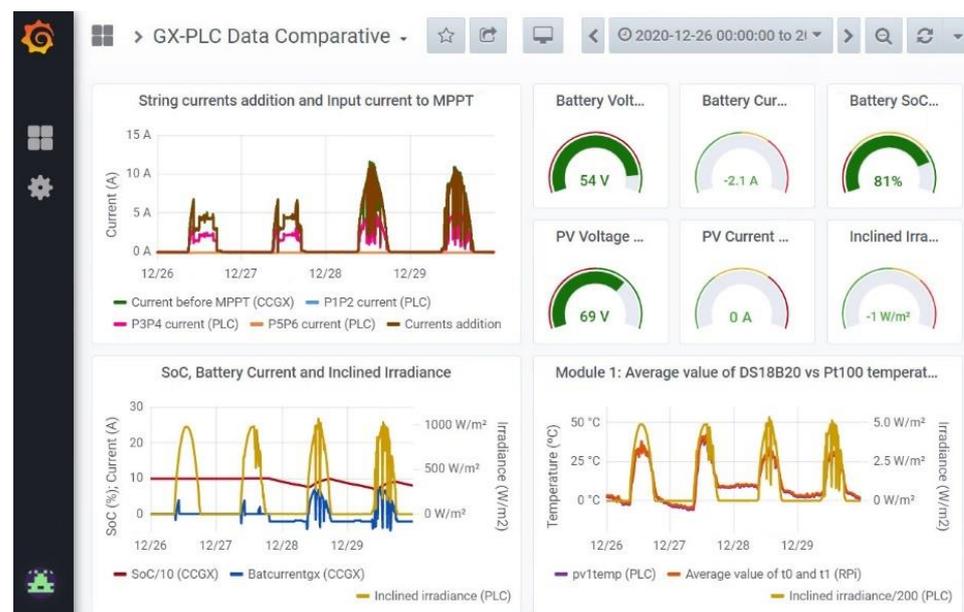
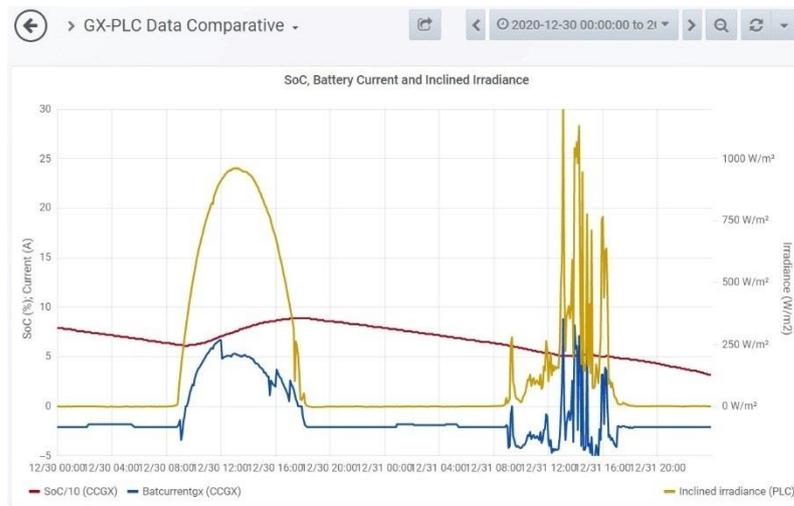
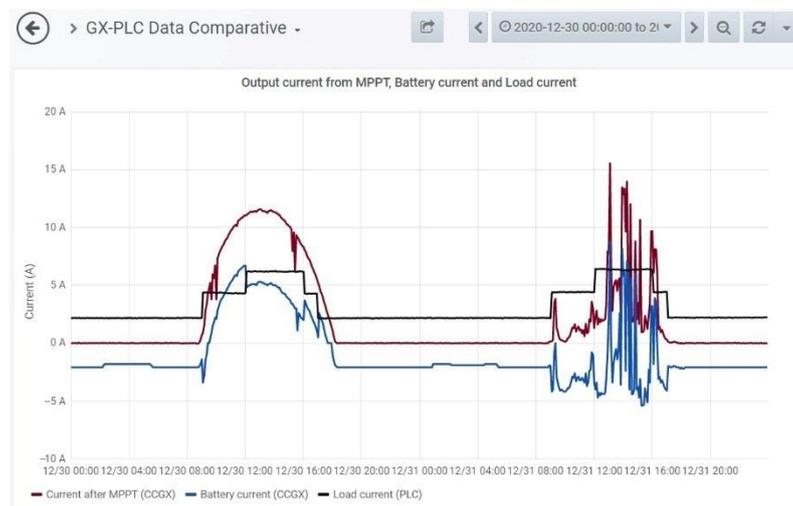


Figure 12. Overview of the Grafana graphical user interface (GUI) developed for Modbus—shared magnitudes.

The days 30 and 31 December 2020, have been chosen to illustrate the behavior of the PV generator and the battery during sunny and cloudy days, respectively. Incident irradiance (yellow curve) and the current of the battery (blue color), together with the SoC (red color) evolution are displayed in Figure 13a. On the sunny day, the energy stored in the battery increased, as shown by its SoC, which grew from 61% up to 89% due to the PV generation. Note that SoC/10 is the SoC divided by 10, in order for it to be represented within the same graphic.



(a)



(b)



(c)

Figure 13. Evolution of magnitudes in Grafana GUI: (a) battery current, state of charge (SoC) and irradiance; (b) currents of load, battery, and output of the MPPT charger; (c) power of PV generator, battery, and load.

The SoC of the battery increased at the same time that the load consumed the certain current. Figure 13b depicts the currents of the load, of the battery and the output of the MPPT charger during both days. As can be visualized, the current generated by the PV modules (burgundy curve), once adapted by the solar charger, was devoted to fulfill the demand of the load and to charge the battery. In fact, the current demanded by the load changed according to a custom-designed profile (black color in Figure 13b). Moreover, it can be observed that the battery delivers the current required during a transient short interval, until the MPPT charger adapts the power of the PV modules to the new demand, or when the existing irradiance is not enough to cover the load demand solely with the PV-generated current.

As can be seen, on the sunny day at 12:00, the current load increased, so the current received by the battery decreased. This was due to the fact that the PV modules were providing the maximum possible current, but it was not enough to fully satisfy the demand of both the battery and the load. Thus, the charge of battery was slowed. At 16:00, the situation was the opposite; the load current was reduced and, consequently, the current delivered to the battery increased.

In order to visualize power flows, Figure 13c shows the evolution of the power generated by the PV modules (green curve), demanded by the load (black color) and corresponding to the battery (blue color). The trend and interactions were the same as those observed for the currents depicted in Figure 13b. The CCGX already provided measurements of the power generated by the PV modules, whereas the power of the battery and the power delivered to the load were calculated from the measurements gathered by the PLC.

The evolution of the temperatures of a single PV module can be observed together with the incident irradiance in another chart (Figure 14). The purple color corresponds to analogue measurement of the Pt100 linked to the PLC, whereas the signal provided by digital sensors are represented in orange. There is a difference between both curves of around 1 °C, which is coherent with the different position of the sensors. Both temperature values vary with the same trend, according to the irradiance curve for the sunny and the cloudy days. Note that in the figure, Inclined Irradiance/200 is the solar irradiance in the inclined plane, divided by 200, in order to be represented in the same chart.

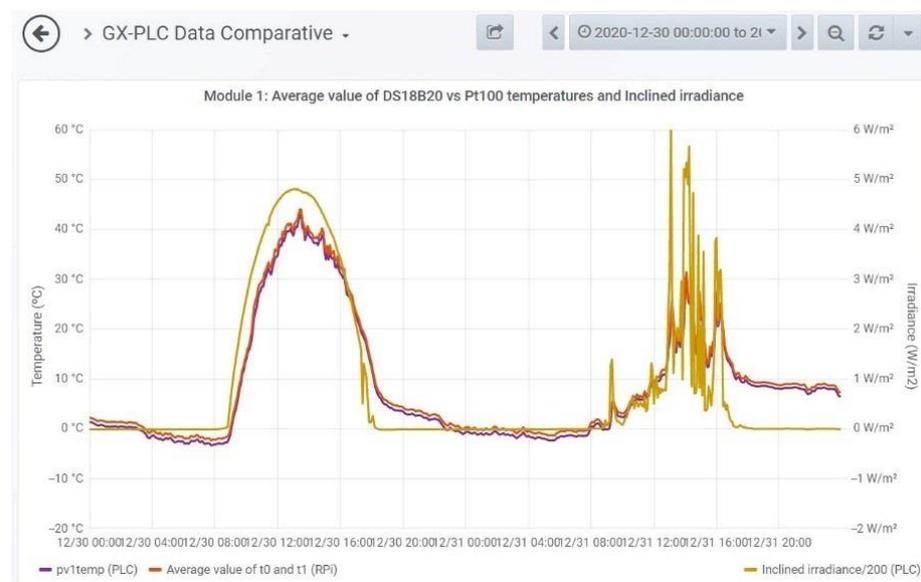


Figure 14. Evolution of temperature measurements for one PV module and irradiance in Grafana GUI.

As can be seen in Figure 15, the sum of the currents corresponding to each string of PV modules (brown color) is equal to the current provided by the CCGX (green), with the latter one named as Current before MPPT (CCGX) in the chart. Additionally, currents delivered by each pair of PV modules is observed (clear blue, pink and orange colors). It must be noted that the CCGX gathers the current produced by the whole PV generator from the MPPT charger, and does not consider the electrical interconnection of the modules. In the viewed days, data of current generated by a string of modules were zero (orange color) due to the fact that it had been disconnected. This way, differences or malfunctions of the modules could be detected through analyses and comparison of the Modbus-shared measurements.

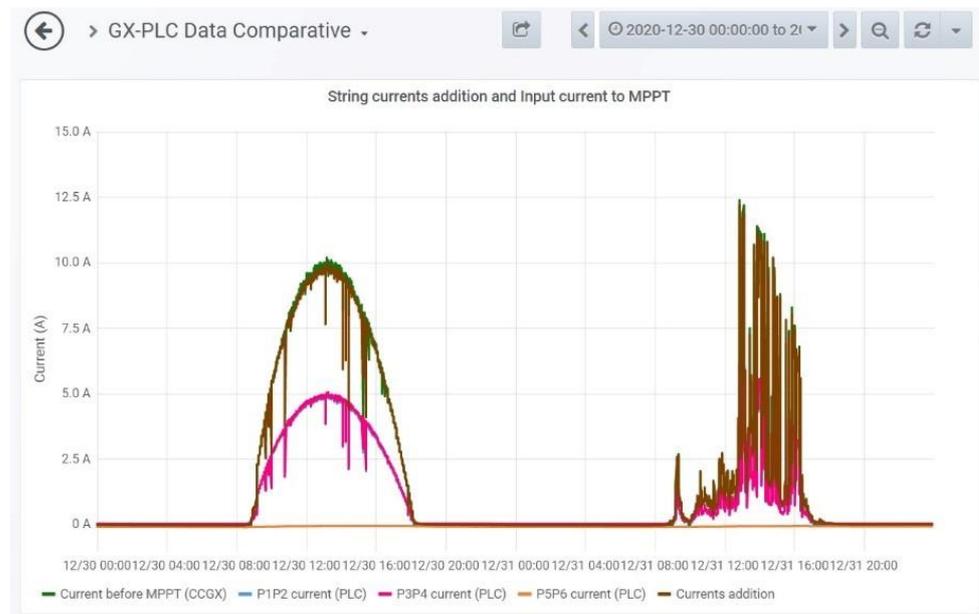


Figure 15. Evolution of PV-generated currents in Grafana GUI.

6.2. Discussion

The presented architecture is general and scalable; it does not rely on particular manufacturers, operative systems, hardware or software, with the exception of the open communication protocol Modbus TCP. Specifically, for a high level of concretion and applicability, Modbus TCP implements the integration of communications, allowing an integrated and effective data exchange between the involved equipment.

The reported experimental results validate the proposal. Namely, the monitoring and automation equipment of a PV-based microgrid are successfully orchestrated and integrated following the architecture.

A notable remark when developing layered frameworks is that, depending on the designers' viewpoint, the number of layers should be between two and eight, as has been commented in the literature review. In this sense, from the humble perspective of the authors, a compromise between scarce layers and an excessive number of them must be found. A low number of layers could imply scarce separation of functionalities, so the division makes no sense. On the contrary, a large number of layers could lead to an excessive atomization of tasks. The presented multi-layered architecture was conceived, refined, and deployed, evolving from a simpler initial version of four levels up to the final configuration expounded in the paper. Namely, six layers fully satisfied the functions that must be performed within energy automation and monitoring.

The present work reports a fully functional application of the architecture using hardware and software equipment which correspond to real facilities. Industrial equipment has been used to implement the microgrid, both for energy generation/consumption and automation, apart from the open-source equipment. In addition, improvements in open-

source equipment, as well as new proprietary devices, can be included in automation and monitoring systems deployed following this framework.

Among the benefits of the proposal, the capability to seamlessly interconnect heterogeneous proprietary and open-source equipment is highlighted. Indeed, the monitoring system Grafana enables visualization of the gathered information in an user-friendly manner. This suite has been chosen for its easy configuration, versatility and increasing application in research works [16].

From an economical point of view, the presented architecture allows investment reduction thanks to the convergence of proprietary and open-source technologies. In fact, in the experimental microgrid, a severe cost decrease was achieved by using the free software Grafana and MariaDB instead of commercial supervisory suites with licensing expenses.

Despite the reported results, the present work has some limitations, which are now expounded. If required, protocol converters can be adopted in order to integrate devices with different communication protocols into the Network Integration Layer. This option must be studied carefully to avoid potential bottlenecks when a large amount of data are to be transferred [11]. Still related to data transmission, obtaining performance and error metrics of the deployed network would be interesting. Cyber-security measures have not been implemented in the proposal so far; nonetheless, measures in this regard for vulnerabilities of the Modbus TCP network should be taken into account [55]. The developed system has been operating continuously for a whole year; however, performance over long time periods should be evaluated, especially concerning open-source equipment [1].

7. Final Remarks

The digital transformation of energy infrastructure is mainly promoted by the deployment of smart grids and microgrids. Their effective implementation overcomes challenges, among which research efforts towards the standardization of communication protocols and networks are found.

To this aim, in this paper, innovative multi-layered architecture has been proposed and validated. Establishing a functional division into six layers, such architecture is devoted to organizing the heterogeneous elements for energy automation and monitoring in microgrids. Around the Modbus TCP network, interoperability is appropriately handled, and data transmission successfully performed. Signals from sensors, controllers, and software environments are properly shared by means of the multi-layered architecture.

It must be remarked that this proposal is a novelty in the existing literature, and overcomes limitations identified in previous works such as a lack of concretization about communication technologies, proprietary/industrial or open-source equipment, and experimental validation. In other words, this work contributes to addressing the gap between conceptual approaches and experimental facilities, providing concrete strategies and details. The developed architecture is envisioned to empower the real deployment of microgrids and, consequently, foster the digital transformation of the power grids.

The target groups are scientists, engineers, and practitioners in the fields of energy automation and monitoring. The selection of devices, software suites and communication protocols will be assisted by the reported theory and experimentation. Moreover, maintenance, modifications, and expansions will be facilitated, given the advantages that this architecture provides. Therefore, the paper is expected to act as a useful resource in the design and deployment of this type of systems, both for R&D activities and implementation in real facilities.

Further works around the implemented architecture include the addition of controllers and sensors for the rest of the microgrid components. Analyses about energy efficiency of the involved components will be carried out, taking advantage of the accumulated measurements. The creation of digital replicas of the components using information provided by the Data Record & Storage layer is also being approached.

Author Contributions: Conceptualization, I.G., J.M.P. and A.J.C.; methodology, I.G. and A.J.C.; software J.M.P. and I.G.; validation, J.M.P. and A.J.C.; investigation, I.G., A.J.C.; data curation, J.M.P. and A.J.C.; writing—original draft preparation, I.G. and J.M.P.; writing—review and editing, I.G. and A.J.C.; supervision, A.J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This project was co-financed by European Regional Development Funds FEDER and by the Junta de Extremadura (IB18041).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

AMQP	Advanced Message Queuing Protocol
BMU	Battery Management Unit
CAN	Controller Area Network
CoAP	Constrained Application Protocol
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
DAQ	Data Acquisition Card
DBMS	Database Management System
DC	Direct Current
DDS	Data Distribution Services
DNP3	Distributed Network Protocol 3
DPS	Decentralized Periphery Station
GUI	Graphical User Interface
HMI	Human–Machine Interface
I/O	Input/Output
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IoE	Internet of Energy
IoT	Internet of Things
ISA	International Society of Automation
IT	Information Technology
MPPT	Maximum Power Point Tracking
MQTT	Message Queue Telemetry Transport
NGSI	Next Generation Service Interface
OPC UA	Open Platform Communications Unified Architecture
OSI	Open Systems Interconnection
PC	Personal Computer
PEM	Polymer Electrolyte Membrane
PLC	Programmable Logic Controller
PV	Photovoltaic
R&D	Research and Development
RAM	Random-Access Memory
RAMI 4.0	Reference Architecture Model for Industry 4.0
RPi	Raspberry Pi
RTU	Remote Terminal Unit
SGAM	Smart Grid Architecture Model
TIA	Totally Integrated Automation
SoC	State of Charge
SQL	Search Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol

References

1. González, I.; Calderón, A.J. Integration of open source hardware Arduino platform in automation systems applied to smart grids/micro-grids. *Sustain. Energy Technol. Assess.* **2019**, *36*, 100557. [CrossRef]
2. Mariam, L.; Basu, M.; Conlon, M.F. Microgrid: Architecture, policy and future trends. *Renew. Sustain. Energy Rev.* **2016**, *64*, 477–489. [CrossRef]
3. González, I.; Calderón, A.; Andújar, J.; Andújar-Márquez, J.M. Novel remote monitoring platform for RES-hydrogen based smart microgrid. *Energy Convers. Manag.* **2017**, *148*, 489–505. [CrossRef]
4. Martín-Martínez, F.; Sánchez-Miralles, A.; Rivier, M. A literature review of microgrids: A functional layer based classification. *Renew. Sustain. Energy Rev.* **2016**, *62*, 1133–1153. [CrossRef]
5. Wu, Y.; Wu, Y.; Guerrero, J.M.; Vasquez, J.C. Digitalization and decentralization driving transactive energy internet: Key technologies and infrastructures. *Int. J. Electr. Power Energy Syst.* **2021**, *126*, 106593. [CrossRef]
6. Marzal, S.; Gonzalez-Medina, R.; Salas-Puente, R.; Garcera, G.; Figueres, E. An embedded internet of energy communication platform for the future smart microgrids management. *IEEE Internet Things J.* **2019**, *6*, 7241–7252. [CrossRef]
7. Satuyeva, B.; Sauranbayev, C.; Ukaegbu, I.A.; Nunna, H.K. Energy 4.0: Towards IoT applications in Kazakhstan. *Procedia Comput. Sci.* **2019**, *151*, 909–915. [CrossRef]
8. Ginocchi, M.; Ahmadifar, A.; Ponci, F.; Monti, A. Application of a smart grid interoperability testing methodology in a real-time hardware-in-the-loop testing environment. *Energies* **2020**, *13*, 1648. [CrossRef]
9. Kim, J.-S.; So, S.M.; Kim, J.-T.; Cho, J.-W.; Park, H.-J.; Jufri, F.H.; Jung, J. Microgrids platform: A design and implementation of common platform for seamless microgrids operation. *Electr. Power Syst. Res.* **2019**, *167*, 21–38. [CrossRef]
10. Ahmed, M.A.; Kang, Y.C.; Kim, Y.-C. Communication network architectures for smart-house with renewable energy resources. *Energies* **2015**, *8*, 8716–8735. [CrossRef]
11. Cagnano, A.; De Tuglie, E.; Mancarella, P. Microgrids: Overview and guidelines for practical implementations and operation. *Appl. Energy* **2020**, *258*, 114039. [CrossRef]
12. González, I.; Calderón, A.J.; Figueiredo, J.; Sousa, J.M.C. A literature survey on Open Platform Communications (OPC) applied to advanced industrial environments. *Electronics* **2019**, *8*, 510. [CrossRef]
13. González, I.; Calderón, A.J.; Barragán, A.J.; Andújar, J.M. Integration of sensors, controllers and instruments using a novel OPC architecture. *Sensors* **2017**, *17*, 1512. [CrossRef]
14. Institute of Electrical and Electronics Engineers. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*; IEEE Press: New York, NY, USA, 1990.
15. Nițulescu, I.-V.; Korodi, A. Supervisory control and data acquisition approach in node-RED: Application and discussions. *IoT* **2020**, *1*, 76–91. [CrossRef]
16. Fernández, P.D.A.; Fernández, M.; Ángel, F.; Candás, J.L.C.; Arbolea, P.A. An IoT open source platform for photovoltaic plants supervision. *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106540. [CrossRef]
17. De Araújo, P.R.C.; Filho, R.H.; Rodrigues, J.J.P.C.; Oliveira, J.P.C.M.; Braga, S.A. Infrastructure for integration of legacy electrical equipment into a smart-grid using wireless sensor networks. *Sensors* **2018**, *18*, 1312. [CrossRef]
18. Wu, F.F.; Varaiya, P.P.; Hui, R.S. Smart grids with intelligent periphery: An architecture for the energy internet. *Engineering* **2015**, *1*, 436–446. [CrossRef]
19. Gomes, L.; Vale, Z.; Corchado, J.M. Microgrid management system based on a multi-agent approach: An office building pilot. *Measurement* **2020**, *154*, 107427. [CrossRef]
20. Bracco, S.; Delfino, F.; Pampararo, F.; Robba, M.; Rossi, M. A dynamic optimization-based architecture for polygeneration microgrids with tri-generation, renewables, storage systems and electrical vehicles. *Energy Convers. Manag.* **2015**, *96*, 511–520. [CrossRef]
21. Modbus Homepage. Available online: <http://www.modbus.org/> (accessed on 21 December 2020).
22. Zyrianoff, I.; Heideker, A.; Silva, D.; Kleinschmidt, J.; Soininen, J.-P.; Cinotti, T.S.; Kamiński, C. Architecting and deploying IoT smart applications: A performance—Oriented approach. *Sensors* **2019**, *20*, 84. [CrossRef]
23. Angioni, A.; Lu, S.; Hooshyar, H.; Cairo, I.; Repo, S.; Ponci, F.; Della Giustina, D.; Kulmala, A.; DeDe, A.; Monti, A.; et al. A distributed automation architecture for distribution networks, from design to implementation. *Sustain. Energy Grids Networks* **2018**, *15*, 3–13. [CrossRef]
24. ISA 95 Standard Homepage. Available online: <https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa95> (accessed on 21 December 2020).
25. Paredes-Parra, J.M.; Mateo-Aroca, A.; Silvente-Niñirola, G.; Bueso, M.C.; Molina-García, Á. PV module monitoring system based on low-cost solutions: Wireless raspberry application and assessment. *Energies* **2018**, *11*, 3051. [CrossRef]
26. Bader, S.R.; Maleshkova, M.; Lohmann, S. Structuring reference architectures for the industrial internet of things. *Futur. Internet* **2019**, *11*, 151. [CrossRef]
27. Moharm, K. State of the art in big data applications in microgrid: A review. *Adv. Eng. Inform.* **2019**, *42*, 100945. [CrossRef]
28. Teng, S.Y.; Touš, M.; Leong, W.D.; How, B.S.; Lam, H.L.; Máša, V. Recent advances on industrial data-driven energy savings: Digital twins and infrastructures. *Renew. Sustain. Energy Rev.* **2021**, *135*, 110208. [CrossRef]
29. Stadler, M.; Mashayekh, S.; Narayanan, S.; DeForest, N.; Brandt, T. *Supervisory Controller for PV and Storage Microgrids*; Technical Report; Lawrence Berkeley National Laboratory: Berkeley, CA, USA, 2015.

30. Srinivasan, S.; Kotta, U.; Ramaswamy, S. A layered architecture for control functionality implementation in smart grids. In Proceedings of the 2013 10th IEEE International Conference on Networking, Sensing AND Control (ICNSC), Evry, France, 10–12 April 2013; pp. 100–105.
31. Pau, M.; Patti, E.; Barbierato, L.; Estebansari, A.; Pons, E.; Ponci, F.; Monti, A. A cloud-based smart metering infrastructure for distribution grid services and automation. *Sustain. Energy Grids Netw.* **2018**, *15*, 14–25. [[CrossRef](#)]
32. Victor, J.L.F.; Jucá, S.C.S.; Pereira, R.I.S.; Carvalho, P.C.M.; Fernández-Ramírez, L.M.; Federal Institute of Technology of Ceará (IFCE). IoT monitoring systems applied to photovoltaic generation: The relevance for increasing decentralized plants. *Renew. Energy Power Qual. J.* **2019**, *17*, 536–545. [[CrossRef](#)]
33. Ferrández-Pastor, F.J.; García-Chamizo, J.M.; Gomez-Trillo, S.; Valdivieso-Sarabia, R.; Nieto-Hidalgo, M. Smart management consumption in renewable energy fed ecosystems. *Sensors* **2019**, *19*, 2967. [[CrossRef](#)] [[PubMed](#)]
34. Moghimi, M.; Liu, J.; Jamborsalamati, P.; Rafi, F.H.M.; Rahman, S.; Hossain, J.; Stegen, S.; Lu, J. Internet of things platform for energy management in multi-microgrid system to improve neutral current compensation. *Energies* **2018**, *11*, 3102. [[CrossRef](#)]
35. Shapsough, S.; Takroui, M.; Dhaouadi, R.; Zualkernan, I.A. Using IoT and smart monitoring devices to optimize the efficiency of large-scale distributed solar farms. *Wirel. Netw.* **2018**, 1–17. [[CrossRef](#)]
36. Wu, Y.; Wu, Y.; Guerrero, J.M.; Vasquez, J.C.; Palacios-García, E.J.; Guan, Y. IoT-enabled microgrid for intelligent energy-aware buildings: A novel hierarchical self-consumption scheme with renewables. *Electronics* **2020**, *9*, 550. [[CrossRef](#)]
37. Gomes, L.; Spínola, J.; Vale, Z.; Corchado, J.M. Agent-based architecture for demand side management using real-time resources' priorities and a deterministic optimization algorithm. *J. Clean. Prod.* **2019**, *241*, 118154. [[CrossRef](#)]
38. Sirviö, K.; Kauhaniemi, K.; Memon, A.A.; Laaksonen, H.; Kumpulainen, L. Functional analysis of the microgrid concept applied to case studies of the sundom smart grid. *Energies* **2020**, *13*, 4223. [[CrossRef](#)]
39. CEN-CENELEC-ETSI. *CEN-CENELEC-ETSI Smart Grid Reference Architecture*; Technical Report; CEN-CENELEC-ETSI: Brussels, Belgium, 2012. Available online: https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf (accessed on 6 December 2020).
40. Uslar, M.; Rohjans, S.; Neureiter, C.; Andrés, F.P.; Velasquez, J.; Steinbrink, C.; Efthymiou, V.; Migliavacca, G.; Horsmanheimo, S.; Brunner, H.; et al. Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: A european perspective. *Energies* **2019**, *12*, 258. [[CrossRef](#)]
41. Sanchez-Herrera, R.; Marquez, M.A.; Andujar, J.M. Easy and secure handling of sensors and actuators as cloud-based service. *IEEE Access* **2020**, *8*, 10433–10442. [[CrossRef](#)]
42. Victron Energy Homepage. Available online: https://www.victronenergy.com/live/ccgx:modbustcp_fa (accessed on 21 October 2020).
43. SMA Homepage. Available online: <https://www.sma-america.com/products/monitoring-control/modbus-protocol-interface.html> (accessed on 15 November 2020).
44. SunSpec Homepage. Available online: <https://sunspec.org/sunspec-modbus-specifications/> (accessed on 10 December 2020).
45. Kenner, S.; Thaler, R.; Kucera, M.; Volbert, K.; Waas, T. Comparison of smart grid architectures for monitoring and analyzing power grid data via modbus and rest. *EURASIP J. Embed. Syst.* **2016**, *2017*, 5. [[CrossRef](#)]
46. Lezynski, P.; Szczesniak, P.; Waskowicz, B.; Smolenski, R.; Drozd, W. Design and implementation of a fully controllable cyber-physical system for testing energy storage systems. *IEEE Access* **2019**, *7*, 47259–47272. [[CrossRef](#)]
47. Soetedjo, A.; Nakhoda, Y.I.; Saleh, C. An embedded platform for testbed implementation of multi-agent system in building energy management system. *Energies* **2019**, *12*, 3655. [[CrossRef](#)]
48. Samanta, H.; Bhattacharjee, A.; Pramanik, M.; Das, A.; Das Bhattacharya, K.; Saha, H. Internet of things based smart energy management in a vanadium redox flow battery storage integrated bio-solar microgrid. *J. Energy Storage* **2020**, *32*, 101967. [[CrossRef](#)]
49. Khatua, S.; Mukherjee, V. Application of PLC based smart microgrid controller for sequential load restoration during station blackout of nuclear power plants. *Ann. Nucl. Energy* **2021**, *151*, 107899. [[CrossRef](#)]
50. Trunzer, E.; Calà, A.; Leitão, P.; Gepp, M.; Kinghorst, J.; Lüder, A.; Schauerte, H.; Reifferscheid, M.; Vogel-Heuser, B. System architectures for industrie 4.0 applications. *Prod. Eng.* **2019**, *13*, 247–257. [[CrossRef](#)]
51. Pymodbus Library on GitHub. Available online: <https://github.com/riptideio/pymodbus> (accessed on 10 December 2020).
52. Pereira, R.I.; Dupont, I.M.; Carvalho, P.C.; Jucá, S.C. IoT embedded linux system based on raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant. *Measurement* **2018**, *114*, 286–297. [[CrossRef](#)]
53. Godoy, A.J.C.; Pérez, I.G. Integration of sensor and actuator networks and the SCADA system to promote the migration of the legacy flexible manufacturing system towards the industry 4.0 concept. *J. Sens. Actuator Netw.* **2018**, *7*, 23. [[CrossRef](#)]
54. Wollschlaeger, M.; Sauter, T.; Jasperneite, J. The future of industrial communication: Automation networks in the Era of the internet of things and industry 4.0. *IEEE Ind. Electron. Mag.* **2017**, *11*, 17–27. [[CrossRef](#)]
55. Figueroa-Lorenzo, S.; Añorga, J.; Arrizabalaga, S. A role-based access control model in modbus SCADA systems. A centralized model approach. *Sensors* **2019**, *19*, 4455. [[CrossRef](#)] [[PubMed](#)]