*applied sciences*

MDPI

*Article*

# Conversion of a Network Section with Loads, Storage Systems and Renewable Generation Sources into a Smart Microgrid

**Oscar Izquierdo-Monge [1], Paula Peña-Carro [1], Roberto Villafafila-Robles [2], Oscar Duque-Perez [3,*], Angel Zorita-Lamadrid [3] and Luis Hernandez-Callejo [4,*]**

[1] CEDER-CIEMAT, Autovía de Navarra A15 Salida 56, 422290 Lubia (Soria), Spain; oscar.izquierdo@ciemat.es (O.I.-M.); paula.pena@ciemat.es (P.P.-C.)
[2] Centre d'Innovació Tecnològica en Convertidors Estàtics i Accionaments (CITCEA-UPC), Departament d'Enginyeria Elèctrica, Universitat Politècnica de Catalunya, ETS d'Enginyeria Industrial de Barcelona, Av. Diagonal 647, 2nd Floor, 08028 Barcelona, Spain; roberto.villafafila@citcea.upc.edu
[3] ADIRE-ITAP, University of Valladolid, Paseo del Cauce 59, 47011 Valladolid, Spain; zorita@eii.uva.es
[4] ADIRE-ITAP, University of Valladolid, Campus Universitario Duques de Soria, 42004 Soria, Spain
[*] Correspondence: oscar.duque@eii.uva.es (O.D.-P.); luis.hernandez.callejo@uva.es (L.H.-C.); Tel.: +34-975129418 (L.H.-C.)

check for updates

**Citation:** Izquierdo-Monge, O.; Peña-Carro, P.; Villafafila-Robles, R.; Duque-Perez, O.; Zorita-Lamadrid, A.; Hernandez-Callejo, L. Conversion of a Network Section with Loads, Storage Systems and Renewable Generation Sources into a Smart Microgrid. *Appl. Sci.* **2021**, *11*, 5012. https://doi.org/10.3390/app11115012

Academic Editors: Víctor Alonso Gómez, Sergio Nesmachnow, Vicente Leite, Javier Prieto, Ângela Ferreira and Hannu Laaksonen

Received: 1 May 2021
Accepted: 26 May 2021
Published: 28 May 2021

**Abstract:** This paper shows an experimental application case to convert a part of the grid formed by renewable generation sources, storage systems, and loads into a smart microgrid. This transformation will achieve greater efficiency and autonomy in its management. If we add to this the analysis of all the data that has been recorded and the correct management of the energy produced and stored, we can achieve a reduction in the electricity consumption of the distribution grid and, with this, a reduction in the associated bill. To achieve this transformation in the grid, we must provide it with intelligence. To achieve this, a four steps procedure are proposed: identification and description of the elements, integration of the elements in the same data network, establishing communication between the elements and the control system, creating an interface that allows control of the entire network. The microgrid of CEDER-CIEMAT (Renewable Energy Centre in Soria, Spain) is presented as a real case study. This centre is made up of various sources of generation, storage, and consumption. All the elements that make up the microgrid are incorporated into free software, Home Assistant, allowing real-time control and monitoring of all of them thanks to the intelligence that has been provided to the grid. The novelty of this paper is that it describes a procedure that is not reported in the current literature and that, being developed with Home Assistant, is free and allows the control and management of a microgrid from any device (mobile, PC) and from any place, even though not on the same data network as the microgrid.

**Keywords:** smart microgrid; Home Assistant; monitoring and control system

## 1. Introduction

Energy has become a key factor in the development of our lives, leading to an increase in energy needs at a global level, presenting a challenge for the supply of the traditional electricity system. For years, different solutions have been implemented to guarantee supply to all citizens, trying to ensure that the electricity supply comes from different renewable energy sources, making the system as distributed as possible. With the implementation of this diversification, the aim is to improve efficiency, reduce transport losses, and make better use of renewable energy sources, giving rise to microgenerators and with them options such as microgrids and smartgrids.

Microgrid is a term that can be defined as the U.S. Department of Energy [1] proposes: "a group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that act as a single controllable entity respect to the grid. A microgrid can be connected and disconnected from the distribution grid to allow it to operate in grid or island mode. A remote microgrid is a variation of a microgrid operating in island conditions".

Another option is the smartgrid, a newer concept than microgrid but increasingly implemented due to the advantages it offers. It can be defined as a microgrid that has been equipped with an intelligent communications infrastructure, allowing it to operate and act semi-autonomously and/or autonomously on all the elements that make up the microgrid. This management method improves reliability, safety, and efficiency. It optimizes energy flows and improves the detection of grid supply shortages, making quick decisions to provide the necessary supply to the microgrid.

In either case, whether it is a microgrid, or a smartgrid, the installation of storage systems (such as batteries, super-capacitors, or flywheels) is highly recommended. They can counteract the energy imbalances produced [2] and curb energy fluctuations, managing to adjust the generation and overall load curve of the microgrid [3,4], which is affected by the uncontrollable generation associated with such systems [5–8].

The current trend is either the creation of smartgrid or the conversion of an already created microgrid into a smartgrid due to the operational improvements that can be obtained from the whole.

Nowadays, there are many installations or buildings that have micro-generators and storage systems but do not have them integrated in an intelligent way, which means that these installations are not efficient. The problem that this work aims to solve is to show how these networks can be made intelligent in a straightforward way and using free software.

The procedure described in this paper has already been implemented in practice, being the main objective of this work to present the procedure followed at CEDER to convert a grid section with renewable generation sources and different storage systems and loads into a smart microgrid. To achieve this, the Home Assistant tool has been used to create a control software (CMEMS-CEDER Microgrid Energy Management System).

All the elements that make up the microgrid are integrated in CMEMS, which allows real-time monitoring and management to achieve greater efficiency and autonomy.

This procedure consists of four steps: identification and description of the elements, integration of the elements in the same data network, establishment of communication between the elements and the control system, creation of an interface that allows control of the entire network.

Although the work describes a procedure for an "experimental application case", it could be applied generically to other similar network sections with generation, storage, and consumption elements, since the steps described are valid for any grid.

The development of the article continues in Section 2, which presents a revision of literature. Section 3 present the four steps procedure to transform a grid section into a microgrid. Section 4 presents the experimental application case. Finally, conclusions are drawn.

## 2. Revision of Literature

To manage the entire microgrid more efficiently, it is necessary to install a metering, communication, and data processing system to allow semi-autonomous or autonomous operation. In this way, possible supply problems can be solved in a shorter period [9,10]. Within this field, we can find various communication technologies, such as Zigbee, WLAN/Wi-Fi, serial communication (Ethernet/RS-232), WiMAX, power line communication, GSM/GPRS, and DASH 7 smart [11]. Among them, the most used are the first mentioned, due to the security, reliability, and scope they offer [12–15]. The implementation of one or another technology is influenced by the communication distance and the budget available for this component [16].

Independently of the communication technology chosen, every microgrid must have the following communication elements: a defined local communication structure, a hierarchical system of supervision, control and management of the different elements, and intelligent controllers for loads, consumption, and storage systems.

As discussed above, a microgrid can operate either connected to the main distribution grid via the point of common coupling (PCC) or disconnected from it, in island mode. A stable and economically efficient mode of operation is required [16].

The microgrid is governed by a central controller (MGCC). It provides set point signals to the equipment controllers, such as generation systems, storage systems, and loads. The control system is responsible for regulating the operating voltage and frequency. It is also responsible for redistributing the load between the different distributed generation (DG) and storage elements, managing the flow with the main grid, and optimising operating costs.

In order to control all the generation systems and consumption systems that make up the microgrid and to ensure stable operation and economic efficiency [17], it is necessary to have the installation of smart meters in each piece of the equipment (in case the device does not have an integrated measurement and communication system). The smart meters will monitor in real time different parameters such as voltage, current, power, consumption, etc. to know instantly the operating status, and the existence of any problem that may arise if communication with the generation and/or storage device that records all these parameters is impossible, or to monitor more parameters than those offered by the device. To transmit the data collected by the smart meters to an HMI (Human Machine Interface) in which users can read these data and send operating instructions, it is necessary to install another device.

Arduino, which consists of a board with a microcontroller and open software, can be used for this purpose. This device can read the data collected by the Smart Meters, perform calculations if necessary, and send the results where the user wishes. Arduino can communicate with a Raspberry Pi [14,18], a computer with a reduced board with which greater versatility and calculation power is achieved. Raspberry has Wi-Fi and Ethernet connectivity integrated in the board. These devices are responsible for collecting the data registered by Arduino and communicating all these values with other devices, platforms the user has established, and applications and/or local or online databases to store and visualize all those values. Users who want to collect historical data of all the parameters that the installed devices record, have two different options: the manual download when the device memory is full, or the automated sending to a database (normally in the cloud) [13,19–21].

For everything to work, communication protocols (mentioned above) must be established between the different installed devices. The protocol selected differs for each case, device, and location.

In order to manage all the equipment, it is necessary to display all the registers in an interface and allow the user to interact with each one of them. For this purpose, the software is used to act semi-autonomously and/or autonomously on the different equipment that integrates the microgrid. This operation mode will give the user the possibility to optimize the generation and demand curve and therefore reduce the cost of the electricity bill, one of the final targets of this proposal.

We can find different tools, like Matlab/Simulink, Python or another programming language (R, C++, etc.), Labview, etc. that enable the development of software to monitor and control the elements (generation sources, storage systems, and load) that make up a microgrid and turn it into a smart microgrid.

Among all the tools proposed [8,22–24], there are characteristics common to all of them that can present themselves as an advantage: custom code development. Thanks to the ability to adapt the code to each of the particular cases, they are widely used software for the configuration of all the elements of the microgrid. For the specific tool, Home Assistant, or programming languages (R, C++, Python, etc.), we can add another advantage, related to the costs derived from the control and monitoring of the microgrid [25,26]. To use the full license, it is not necessary to pay for it; it is a free program available to any user.

Concerning the disadvantages of its use in any of the cited software, it is the high amount of time dedicated to its preparation, due to the complexity of writing the code.

Home Assistant stands out from this aspect, as it does not require long programming codes for its configuration, nor high programming knowledge to implement all the elements of the microgrid in the software.

For correct management of the microgrid, as mentioned in the previous point, and to optimize, both globally and individually, all the elements that make it up, it is necessary to have software that allows the user to interact with each of them within the same interface.

Different software allows these actions. Examples of some of them are Advance EMS-Platform, ETAP Energy Management System, Monarch TM-Open Systems International, Wattics, etc. In our case, we have developed our own software (CMEMS) using Home Assistant due to its advantages and the total coverage of the required needs. Table 1 shows the advantages and disadvantages of each of the applications mentioned above.

**Table 1.** Advantages and disadvantages of different software. Source: prepared by the authors from various documents [22–26]. Own elaboration.

| Software | Advantages | Disadvantages |
|---|---|---|
| Advance EMS–Platform [27] | Alarm notification<br>Real-time readings and management<br>Access control<br>Historical data<br>Desktop version | Paid license<br>Only for Wind, PV, and Storage<br>No smartphone version<br>Development and configuration by the company |
| ETAP Energy Management System [28] | Alarm notification<br>Real-time readings and management<br>Access control<br>Historical data<br>Desktop version | Paid license<br>No smartphone version<br>Development and configuration by the company |
| Monarch TM–Open Systems International [29] | Alarm notification<br>Real-time readings and management<br>Access control<br>Historical data<br>Desktop and smartphone version | Paid license<br>No readings and management for storage systems<br>Development and configuration by the company |
| Wattics [30] | Read data from meters and sensors<br>Alarm notification<br>Real-time readings and management<br>Access control<br>Historical data<br>Desktop and smartphone version | Paid license<br>Development and configuration by the company |
| CMEMS (developed with Home Assistant [31] | Open-source<br>No limitation on the type and number of equipment to be monitored and managed<br>Alarm notification<br>Real-time readings and management<br>Access control<br>Historical data<br>Desktop and smartphone version<br>Development and configuration by yourself | |

The software mentioned above have several features in common, such as real-time monitoring, alarm notification, secure access, or access from a computer. Home Assistant has been the software of choice mainly for three reasons that make it stand out from the other options: it does not require a license fee to use any of the levels offered by the program, which reduces the overall cost of installation and control of the microgrid [25,26]; it can be configured and edited at any time, without dependence on the company that developed it; and any generation, storage, or load system can be registered without any limitation, making it very suitable for this case study.

## 3. Procedure for the Conversion of a Network Section into a Smart Microgrid

As mentioned in Section 1, this document aims at presenting an experimental application case to provide intelligence to an electricity grid with different generation sources, storage systems, and loads and transform it into a smart microgrid that can operate with high efficiency.

Although the described procedure is applied to an "experimental application case", it could be applied in a generic way in similar grids, since the described steps are valid for any grid of these characteristics.

The procedure to convert an electrical grid with renewable generation sources, storage systems, and consumption that do not communicate with each other, into a smart microgrid is shown in Figure 1.

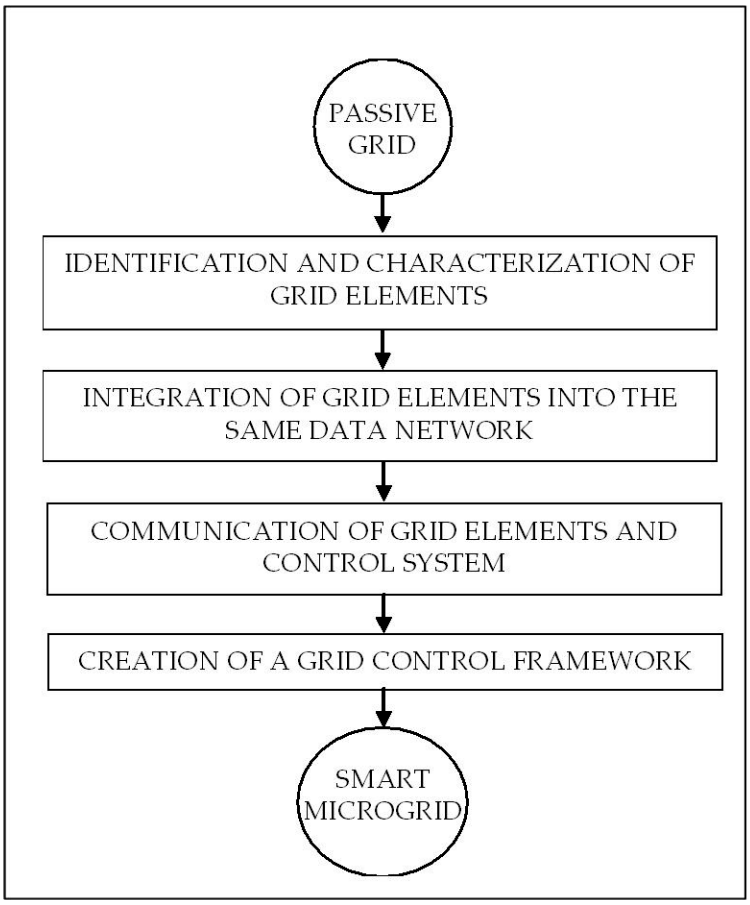

PASSIVE GRID
IDENTIFICATION AND CHARACTERIZATION OF GRID ELEMENTS
INTEGRATION OF GRID ELEMENTS INTO THE SAME DATA NETWORK
COMMUNICATION OF GRID ELEMENTS AND CONTROL SYSTEM
CREATION OF A GRID CONTROL FRAMEWORK
SMART MICROGRID

**Figure 1.** Stages of the procedure to transform passive grid into a smart microgrid. Source: prepared by authors.

**Step 1**. Identification and characterization of grid elements: the first step to provide intelligence to an electricity grid is to identify and characterize all the elements connected: generation sources, storage systems, and loads.

There are different generation sources (renewable and non-renewable) that can be integrated in an electrical microgrid. The most common are photovoltaic, small wind turbines, generator sets, and mini/micro-hydraulic turbines. For each one, it is necessary to know the nominal installed power and the range of power in which it can operate. It is also necessary to know if it is a dispatchable or non-dispatchable generation source. As shown later, if they are non-dispatchable sources, it will be enough to monitor the parameters to be integrated into the control system that is developed (mainly instantaneous power). In the case of dispatchable generation sources, it will not only be necessary to monitor the desired parameters, but it will also be necessary to develop a software (SCADA-

Supervisory Control and Data Acquisition) to operate them and integrate it into the control system of the microgrid.

It is also convenient to know the type of generation source, renewable or non-renewable, to monitor (if possible) the resource in case of renewable energies (wind speed for wind energy, solar radiation in case of photovoltaic, water level for hydraulic turbines, etc.) or fuel in case of non-renewable energies (diesel for generator sets, biomass, etc.).

In case of storage systems, the most used are batteries, or pumping associated to a mini-hydraulic turbine. This procedure can be applied to them or to any storage system such as flywheels, compressed air, etc. All of them are dispatchable, so in addition to monitoring their capacity, having energy that can store, power, performance, etc., it will be necessary, as in the case of dispatchable energies, to develop a software to operate them that must also be integrated into the microgrid control system.

In addition to the generation and storage systems, it is necessary to have a deep knowledge of grid consumptions. It is important to know the power contracted with the energy distribution company, the energy consumed monthly (if possible, at least twelve months to avoid seasonal effects), as well as the most significant loads. As with non-dispatchable generation elements, the microgrid control system cannot control the loads—they will either start or stop depending on the microgrid users. Therefore, it will only be necessary to monitor the instantaneous power and some other parameter that is considered appropriate. It will also be interesting to establish a ranking of loads by priority and influence of the load profile.

Along with all these electrical issues, it will be necessary to know and detail aspects related to communications as shown in step 3.

**Step 2**. Integrate grid elements into the same data network: once the elements that will be part of the microgrid are well known, they must be connected (including the control system) to the same data network.

As explained in Section 1, there are different communication technologies to integrate each element of the microgrid into the data network of the control system, such as: ZigBee, Ethernet, Wi-Fi, GRPS, GSM, WLAN.

The control system will be developed with Home Assistant, and here lies the main novelty of this work and its main advantage. Typically, it is used for home automation applications, but it is a robust solution, economically affordable, and with great potential for monitoring and managing microgrids in real time.

Home Assistant, developed in Python, is open-source software which reduces costs. It allows the user to connect with almost any device regardless of the communication technology used, through a wide range of communication protocols. In addition, it allows developing a SCADA for each dispatchable element and its integration in an interface.

Once Home Assistant is installed in a Raspberry Pi, a computer or even a server, and all the elements of the microgrid defined, it can be managed from every device and everywhere due to its web server, including a computer connected or not to the same data network or from a mobile phone due to the Home Assistant app and the web server.

To integrate each element of the grid to the data network, it is necessary to know some issues related to communications. First, it is necessary to know if each element has a communication card, which allows access to the data network, either locally or remotely (for many years, all devices have had it).

If the element does not allow communication, which is not very common except in loads and in non-dispatchable outdated generation sources, it will be necessary to install a grid analyzer with a communication card (TCP/IP, Wi-Fi, etc.) to measure parameters seen in step 1 (instantaneous power, etc.). In the case of loads, it may be interesting to monitor the most significant, but it would be enough to monitor the total consumption of the microgrid.

If the element only allows connection in local mode (usually through RS232 or RS485), it will be necessary to install some intermediate device (Arduino, data acquisition system, Raspberry Pi, etc.) to connect it to the data network or a converter to switch local mode to remote mode (for instance RS232 to TCP/IP).

If the element allows remote connection through a network card, it can be connected directly to the data network.

**Step 3**. Establish communication between the elements of the grid and the control system: once the components of the grid have been identified and described, and connected to the microgrid control system data network, communication must be established between all of them, so that they do not work independently, but as a whole, and there can be an interaction between them.

To connect each device with the Home Assistant's control system, it is also necessary to know their communication protocols. There are multiple communication protocols that allow the transmission of information, provided that all the devices are connected to the same data network or to equipment in local mode connected to the network. All communication protocols are developed under the framework known as OSI Model (Open System Interconnection).

Among all the protocols, Modbus (RTU, TCP, or RTU over TCP) stands out, which is widely extended and is the one used by most equipment (Programmable Logic Controllers-PLC, photovoltaic inverters, wind inverters, battery chargers, network analyzers for measuring loads, etc.) since it is robust, easy, open-source, and therefore free and above all reliable.

Finally, once all the equipment is connected to the same data network and the protocol used by each one is known (mainly Modbus), communication must be established between them and Home Assistant in the following way:

1. The element (generation source, storage system, or load) must be in Home Assistant configuration file, in a different way depending on the protocol used, as follows:

- Modbus TCP/IP (Transmission Control Protocol/Internet Protocol): it is based on a client/server architecture and allows communication over an Ethernet grid, no CRC required. It is the most common protocol. A name must be assigned to the element, and the IP address defined, the type of communication, and the communication port (502), as follows:
  ```
  - name: Photovoltaic1      # name assigned to the element.
  type: tcp                  # protocol type
  host: 192.168.15.75        # IP address
  port: 502                  # communication port
  ```
- Modbus RTU: this is a master/slave architecture for linking a control system to a Remote Terminal Unit (RTU) via a serial port. Commonly, the master is an HMI or a SCADA system that sends a request and the slave is a sensor or PLC that returns a response. A CRC (Cyclic Redundancy Checksum) is used as an error-checking mechanism and as a procedure to ensure data reliability. A name must be assigned to the element, the type defined, and method of communication, the communication port, baudrate, stopbits, bytesize and parity, all given, as follows:
  ```
  - name: Battery1       # name assigned to the element.
  type: serial           # type of communication.
  method: rtu            # method of communication.
  port:/dev/ttyUSB0      # port (USB in this case).
  baudrate: 19200
  stopbits: 1
  bytesize: 8
  parity: E
  ```
- Modbus RTU over TCP: it is a combination of Modbus TCP and Modbus RTU. It is based on client/server architecture for communications over Ethernet as Modbus TCP/IP but uses a CRC as Modbus RTU. The same issue as in TCP/IP must be defined.
  ```
  - name: SmallWindTurbine1
  type: rtuovertcp
  ```

```
host: 192.168.15.87
port: 7128
```

- HTTP protocol: The element must be defined, a name assigned, the http address for the information required, and the scan interval provided: sensor:

```
- platform: command_line
name: "Pump1"
command: "curl -s 'http://admin:password@192.168.15.107/io.cgi?' (accessed on
1 January 2021) | awk '/^relays/{print substr($2, 1, 1)}'"
scan_interval: 1
```

Despite that there are other many communication protocols, most of the elements that can be connected to a microgrid use one of those described above.

Once all elements are defined, the registers with the required information must be read from the addresses of each one. This procedure is also carried out in the configuration file of Home Assistant.

In Modbus protocol, it is necessary to have the Modbus frame (it should be in the manufacturer's manual, but many times it is not and it is necessary to contact the manufacturer to provide it) to know the addresses where the variables to be read are.

For instance, to read instantaneous power in the photovoltaic defined above, the following definition must be included in the configuration:

```
- platform: Modbus                     #Modbus
scan_interval: 1                       #scan interval
registers:
- name: Photovoltaic1_Instant_Power    #monitored parameter
hub: Photovoltaic 1                    #element defined previously
register_type: input                   #type of register (input, holding, etc.)
unit_of_measurement: W                 #unit of the parameter
slave: 1                               #Modbus Id
register: 44                           #Modbus address
count: 1                               #number of address affected
scale: 1                               #scale (x1, x10, etc.)
```

In http protocol, we do not need to include all this information; in the http address defined before is included the position of the parameter to monitor (and password if required):

http://admin:password@192.168.15.107/io.cgi?' | awk '/^relays/{print substr($2, 1, 1)}' (accessed on 1 January 2021)

This will have to be done for each parameter to monitor.

In dispatchable generation elements, such as the photovoltaic, wind turbines, and loads, it is just necessary to read power (instantaneous values) since action orders cannot be sent to them. By contrast, in dispatchable generation elements and storage systems, it is necessary to read all the registers. With them, a SCADA is developed that allows for control of their operation through the control panel, integrated with all the elements of the grid.

The steps to integrate grid elements into the same data network and establish communication with the Home Assistant control system are shown in Figure 2.

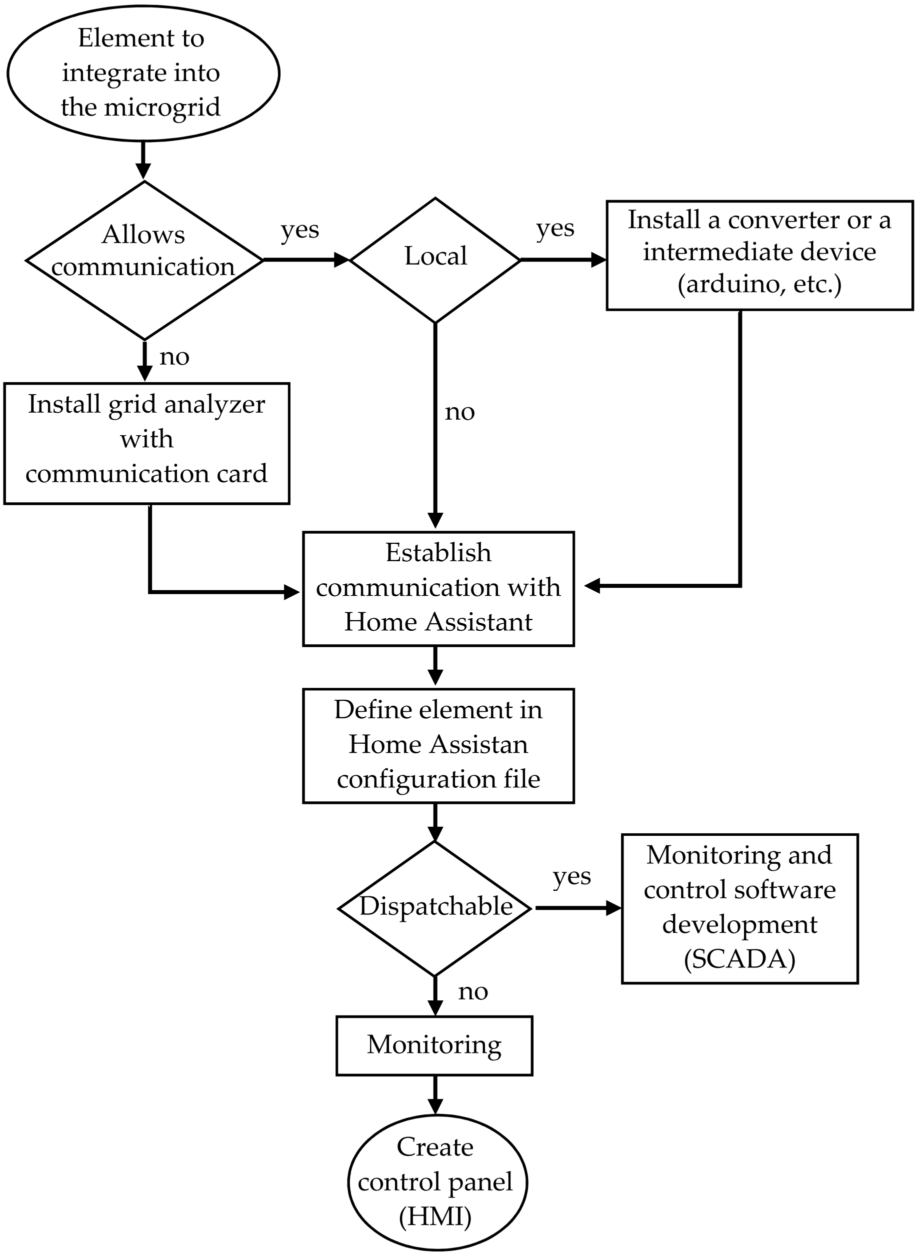

Element to integrate into the microgrid
Allows communication
yes
Local
yes
Install a converter or a intermediate device (arduino, etc.)
no
Install grid analyzer with communication card
no
Establish communication with Home Assistant
Define element in Home Assistan configuration file
Dispatchable
yes
Monitoring and control software development (SCADA)
no
Monitoring
Create control panel (HMI)

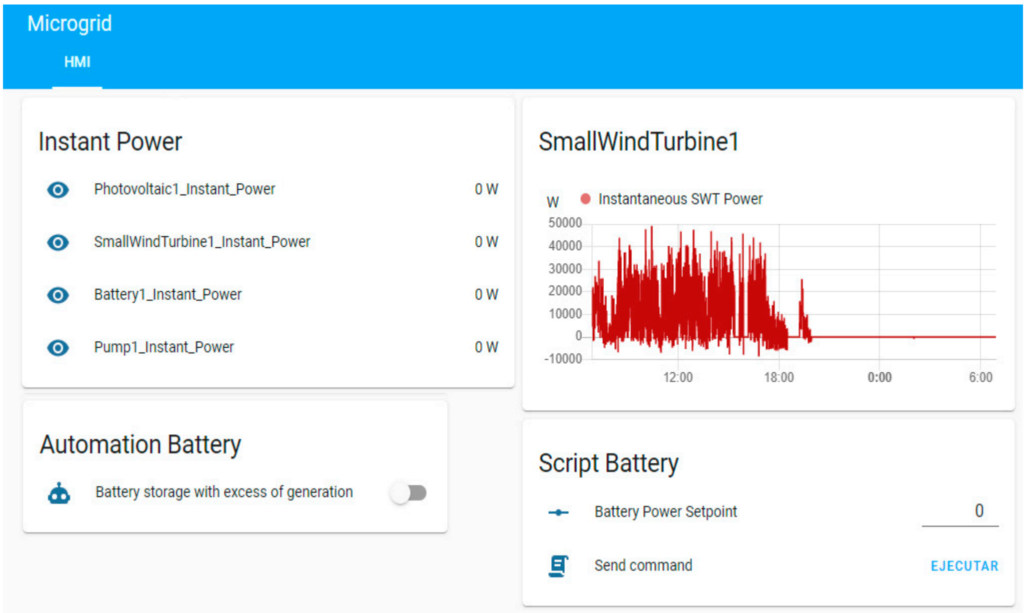
**Figure 2.** Stages of the procedure to set up communication with each element of the grid. Source: prepared by authors.

**Step 4.** Create a control framework: once the communication has been established between all the elements of the grid, a control interface (HMI) must be created to visualise, simply and intuitively, the working mode of each element of the grid, and to send operating instructions to achieve optimisation in the performance of the microgrid, maximising its efficiency.

With Home Assistant, it is easy to create an interface to control the grid after establishing communication with all the grid elements and defining them in the configuration file. The starting point is a graphical interface that gives the possibility to insert cards with different functionalities. It is possible to add cards with maps or weather forecasts that facilitate the estimation of renewable generation systems' power production. There is also the option to include cards with all the registered values (in the configuration file, as seen above) numerically (entities), or to graphically represent these values (historical graph), or to jointly represent (graphically and numerically) a value using the sensor card.

Home Assistant also allows including scripts to send orders to each dispatchable element of the microgrid, start/stop generation elements, charging/discharging storage systems, etc.

In addition, automations can be defined (we can turn them on or off with a switch) to schedule the performance of each dispatchable element of the microgrid based on the values of the monitored parameters, as seen in Figure 3.



Microgrid

HMI

Instant Power

| | |
|---|---|
| Photovoltaic1_Instant_Power | 0 W |
| SmallWindTurbine1_Instant_Power | 0 W |
| Battery1_Instant_Power | 0 W |
| Pump1_Instant_Power | 0 W |

SmallWindTurbine1

Automation Battery

Battery storage with excess of generation ○

Script Battery

Battery Power Setpoint 0

Send command EJECUTAR

**Figure 3.** Home Assistant control panel. Source: prepared by authors.

Scripts and automation can be programmed via code in Home Assistant configuration files, or directly in a frame special for that purpose which makes it easier since there is no need to program them in Python.

Home Assistant has a Telegram integration (also free software) that allows sending text messages to the cell phone of the microgrid supervisor to inform him of relevant events to take the appropriate actions or perform the necessary maintenance tasks.

It is very convenient for later study, although it is not necessary for the operation of the microgrid, to store all the data monitored by the system. Home Assistant stores the data, but for a more efficient management, it also allows to store them (thus establishing them in its configuration file) in one of the multiple existing databases, such as MySQL, which makes easier their later treatment.

Once the control software is fully developed in Home Assistant, ensuring its functioning and analyzing the data registered, microgrid management strategies must be defined to improve its efficiency. In this case, the main strategy for the batteries is to charge them when there is surplus of generation, that is, when the generation systems produce more energy than the demand for the consumption elements of the microgrid.

To discharge the battery, the best strategy based on the analysis of the data recorded for months is to provide power to the microgrid for two hours from 7 am to 9 am (if possible), because is when the start-up of the main loads takes place, and therefore the moment of greatest consumption of CEDER.

Once these strategies are implemented, new data must be analyzed to validate their effectiveness.

## 4. Case Study

CEDER (Centro para el Desarrollo de las Energías Renovables) is the experimental application case. This centre belongs to CIEMAT (Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas), which is a Spanish public research body, assigned to the Ministry of Science and Innovation, focusing on energy and environment and the technologies related to them. It is located in Lubia, in the province of Soria (Spain), with a built-up area of 13,000 m$^2$ in three distinct areas, out of a total of 640 ha (see Figure 4).

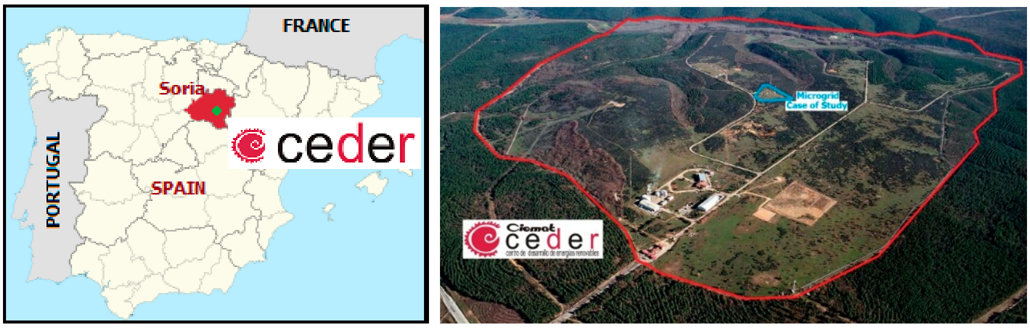


**Figure 4.** CEDER location and distribution. Source: prepared by authors.

CEDER grid is connected to a 45 kV distribution network and transforms at its input to 15 kV. Eight transformation centres make up the grid, reducing the voltage to 400 V. The centre has various non-controllable renewable (photovoltaic and wind), controllable renewable (hydraulic turbine), and non-renewable (diesel generator) generation systems, several storage systems, mechanical (pumping system with tanks at different levels) and electrochemical (lithium-ion and Pb-acid batteries), as well as various consumption elements connected to each of the transformation centres monitored with grid analyzers (PQube) installed in the low voltage part of each of the transformation centres.

To simplify the control system, the case study focuses on a single transformer substation, so there will be no redundancies as much of the installed equipment is similar in the different cases.

### *4.1. Identification and Specification of the Grid Components*

The first step to develop as indicated in the previous section is to identify and describe all the elements that are connected to the network (see Figure 5). In this case, these are detailed below:

- Photovoltaic: four series of six photovoltaic panels (210 W each one) for a total of 5 kW. Panels are polycrystalline silicon and are assembled on a floor structure with a variable inclination angle and are connected to a 5 kW inverter (Ingeteam-Ingecon Sun Lite). It is necessary to connect with the inverter to be able to read instantaneous power values. This model of inverter has a network card included which allows to connect it directly to the CEDER data network (Ethernet) and enables communication via Modbus TCP/IP.
- Wind power: there is a three-blade horizontal axis small wind turbine. Diameter 15 m and 50 kW power. It is operating leeward. A National Instruments FieldPoint is installed to measure variables such as wind speed, power, etc. and can be connected to the CEDER data network via Modbus TCP/IP protocol. With the installation of this element, it is possible to have a data register as this wind turbine does not have a previous connection.
- Battery energy storage system (BESS): formed by 120 Pb-acid cells. Total voltage: 240 V, capacity: 1080 Ah at 120 h ($C_{120}$). BESS is connected to a 50 kW charger/regulator/inverter. The inverter is connected with a SCADA for battery control via RS485 to the serial port of a computer. The control system of the microgrid to be developed will communicate locally with the charger using Modbus RTU protocol.
- Consumption: loads of the microgrid come from the buildings located inside the part of the CEDER where the laboratories and workshops are located. Consumption is measured by means of power quality/grid analyzers (PQube) that allow connection to the CEDER's data network via Ethernet and Modbus TCP/IP communication. These

are installed in the low voltage part of the transformation centre. For this case study, there are no critical loads that need to be supplied in the event of failure or serious disturbance of the microgrid.

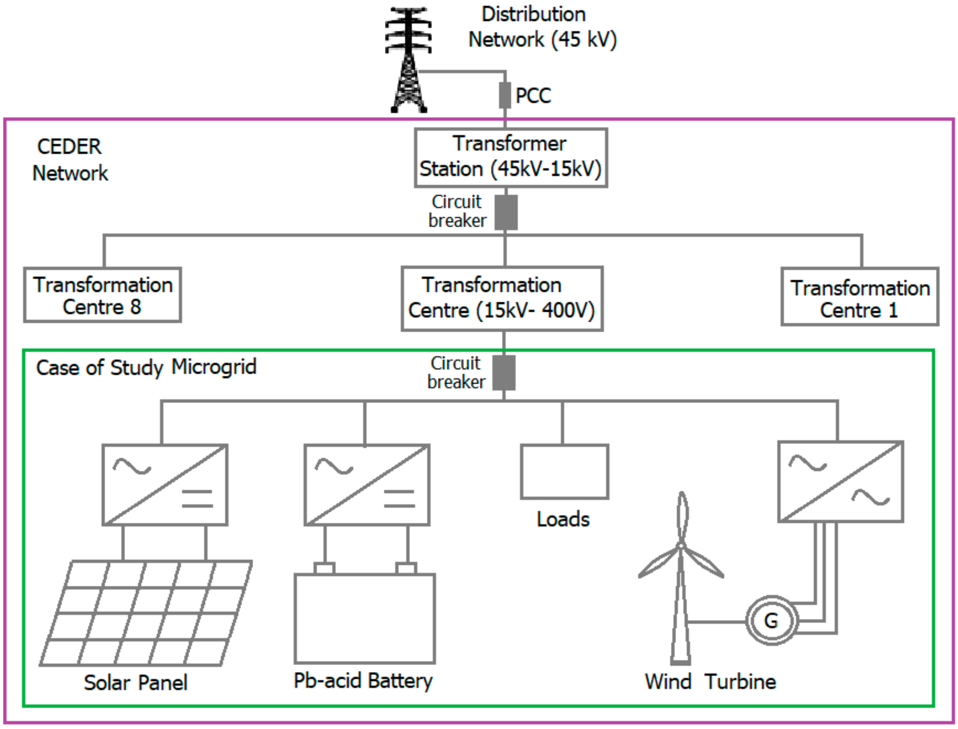


**Figure 5.** Case of study microgrid. Source: prepared by authors.

### *4.2. Integration of Grid Elements into the Same Data Network*

After identifying, characterizing, and establishing the form of communication of all the elements of the microgrid, the next step is the incorporation of each of the elements into the same data network.

In these cases, we have to connect the following elements to the data network: photovoltaic, wind turbine, and battery inverters, and a grid analyzer (PQube) to measure the loads.

Photovoltaic inverter and grid analyzer (PQube) have a TCP/IP Ethernet communication card. Battery control system allows local communication (RS232), so, as seen previously, we have to install a converter RS232 to TCP/IP with Modbus. Finally, the wind turbine inverter is quite old and does not allow any communication, so we have installed a data acquisition system (National Instrument Compact FieldPoint, which allows TCP/IP communication with Modbus) to measure different parameters and communicate with Home Assistant control system.

### *4.3. Establish Communication between Grid Elements*

Subsequently, the communication configuration of each of the elements of the microgrid must be carried out with Home Assistant. There are different communication protocols that allow the transmission of information when all the devices are connected to the same data grid, or to other devices in local mode, and these in turn are connected to the same grid.

Among the most common communication protocols, Modbus stands out from the rest. With it, we can control a network of devices (in our case, generation, storage, and consumption systems) and communicate between them with a control system (Raspberry Pi 4 with Home Assistant). It is the standard communication protocol in the sector as it is robust, easy to use, free and, above all, reliable.

The microgrid scheme communications are shown in Figure 6.

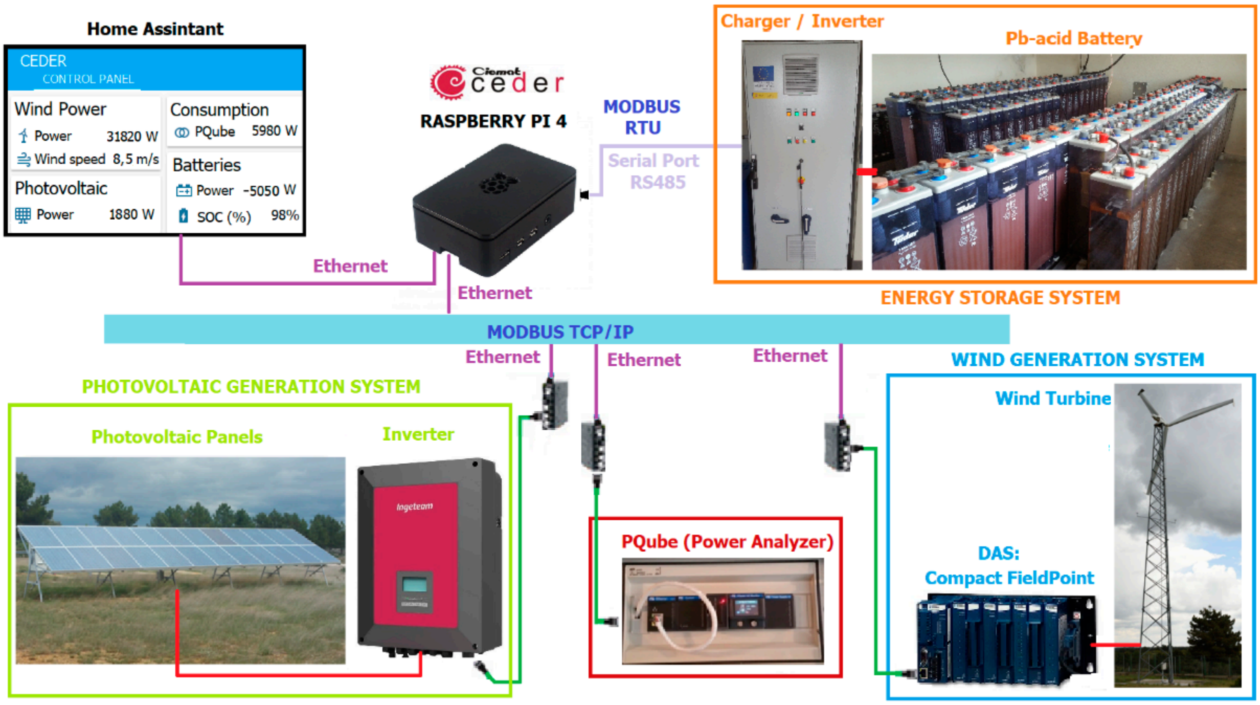


**Figure 6.** Elements of the CEDER's microgrid. Source: prepared by authors.

To communicate between each element (photovoltaic panels, wind turbine, Pb-acid batteries, and loads) with Home Assistant, and monitoring and recording required information, we have to proceed as seen in step 3, Section 2.

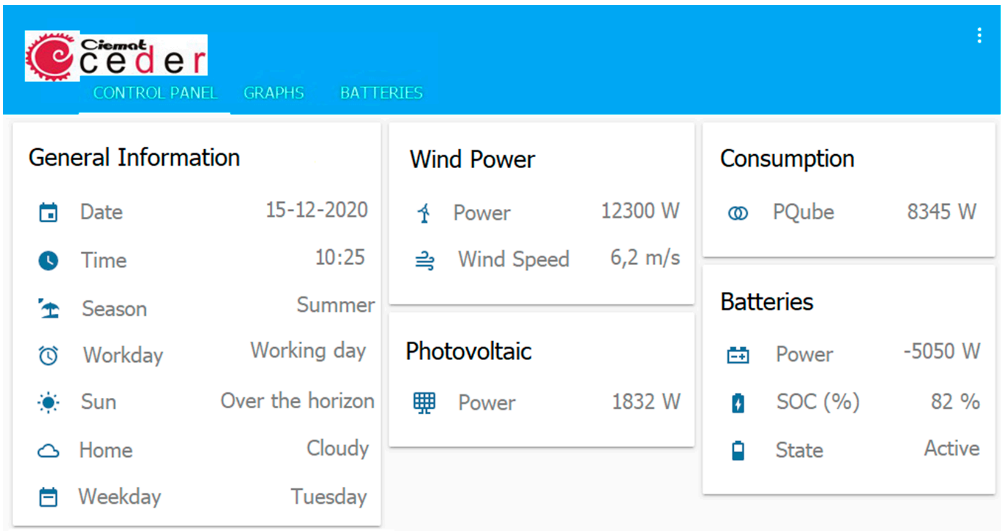
For the non-dispatchable generation elements, such as the wind turbine, photovoltaic inverter, and loads (PQube grid analyzer), only the instantaneous power will be read, as it is not possible to send them any setpoint values. On the other hand, for the controllable generation elements and storage systems (Pb-acid batteries), all the records will be read, with which a SCADA will be developed to control their operation via the control panel in which they will be integrated with all the elements of the microgrid.

### *4.4. Control Grid Framework Creation*

After configuring the communications with all the equipment, we have to design an interface that provides the user with the ability to see all the registers in real time and to have the possibility to execute commands on all of them, such as start/stop actions of generation equipment, load/unload of storage systems, etc.

Figure 7 shows instantaneous values monitored in real-time for the microgrid. As shown, there are 1320 W of photovoltaic generation, 8020 W of wind generation, with a wind speed of 5.1 m/s, compared to a demand of 9880 W. In this case, the batteries are at a standstill as there is no high demand and no surplus energy production to charge them. Otherwise, the storage system is almost fully charged, with a State of Charge (SOC) of 98%.

The SCADA for the control of the batteries also has to be created, as can be seen in Figure 8. It shows how the batteries are in a charging process. At that moment, the microgrid is injecting 1780 W to the batteries, which have a SOC of 98%, and 266 V.



General Information

| | |
|---|---|
| Date | 15-12-2020 |
| Time | 10:25 |
| Season | Summer |
| Workday | Working day |
| Sun | Over the horizon |
| Home | Cloudy |
| Weekday | Tuesday |

Wind Power

| | |
|---|---|
| Power | 12300 W |
| Wind Speed | 6,2 m/s |

Photovoltaic

| | |
|---|---|
| Power | 1832 W |

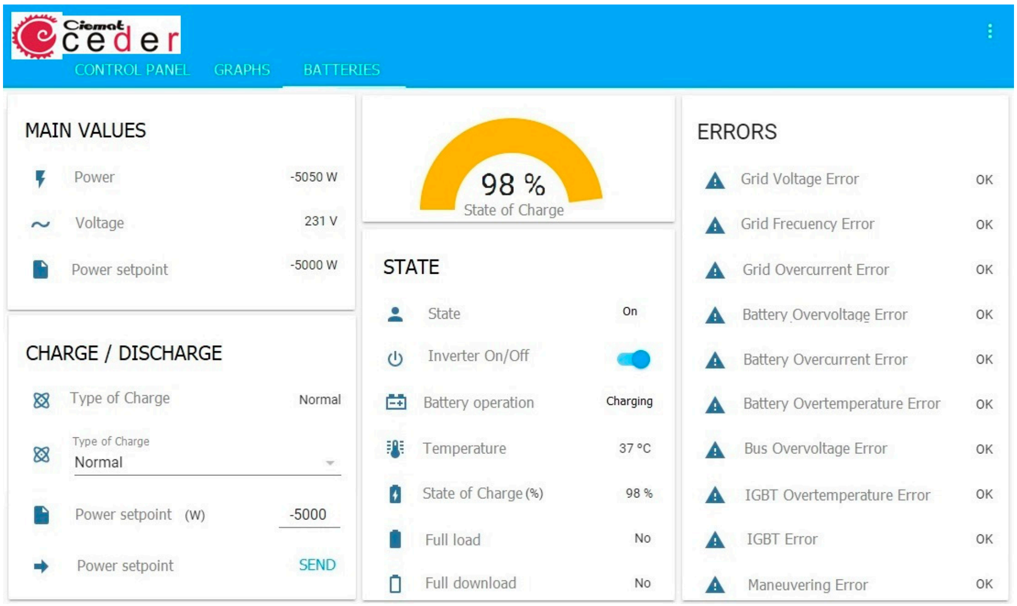
Consumption

| | |
|---|---|
| PQube | 8345 W |

Batteries

| | |
|---|---|
| Power | -5050 W |
| SOC (%) | 82 % |
| State | Active |

**Figure 7.** CEDER's microgrid control framework developed in Home Assistant. Source: prepared by authors.



MAIN VALUES

| | |
|---|---|
| Power | -5050 W |
| Voltage | 231 V |
| Power setpoint | -5000 W |

CHARGE / DISCHARGE

| | |
|---|---|
| Type of Charge | Normal |
| Type of Charge | Normal |
| Power setpoint (W) | -5000 |
| Power setpoint | SEND |

STATE

| | |
|---|---|
| State | On |
| Inverter On/Off | |
| Battery operation | Charging |
| Temperature | 37 °C |
| State of Charge (%) | 98 % |
| Full load | No |
| Full download | No |

ERRORS

| | |
|---|---|
| Grid Voltage Error | OK |
| Grid Frecuency Error | OK |
| Grid Overcurrent Error | OK |
| Battery Overvoltage Error | OK |
| Battery Overcurrent Error | OK |
| Battery Overtemperature Error | OK |
| Bus Overvoltage Error | OK |
| IGBT Overtemperature Error | OK |
| IGBT Error | OK |
| Maneuvering Error | OK |

**Figure 8.** CEDER's microgrid battery SCADA developed in Home Assistant. Source: prepared by authors.

Home Assistant allows visualising in real time the value of the variables collected from each of the elements, as shown in Figure 9.

The initial situation was that of an electrical grid with two renewable generation sources (wind turbine and photovoltaic panels), a Pb-acid batteries storage system, and loads. All these elements worked independently. After applying the procedure proposed, all the elements are integrated into a single control system, and they can work in a coordinated way, which allows establishing management strategies. For example, we can set a strategy to charge the batteries when there is excess generation (See Figures 10–12).

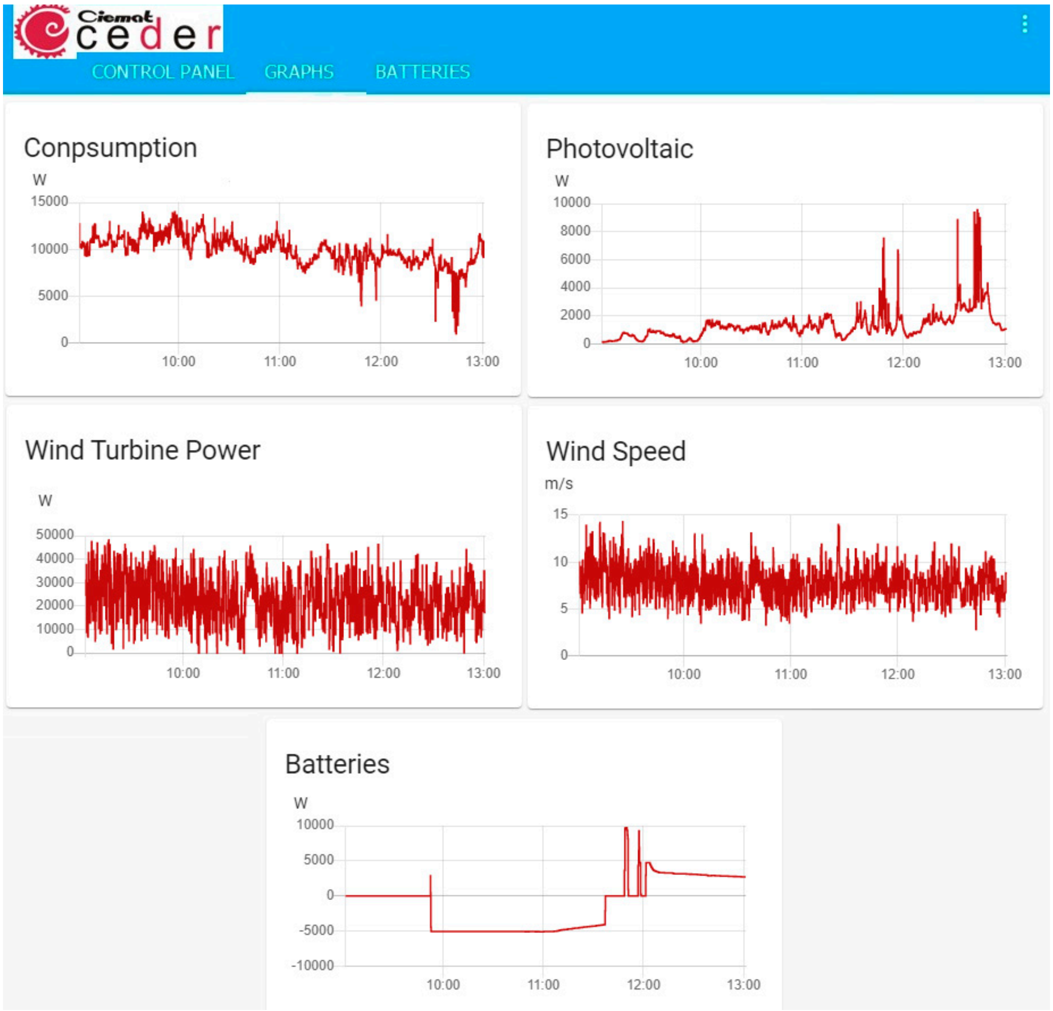


**Figure 9.** CEDER's microgrid elements' instantaneous values graphically represented. Source: prepared by authors.

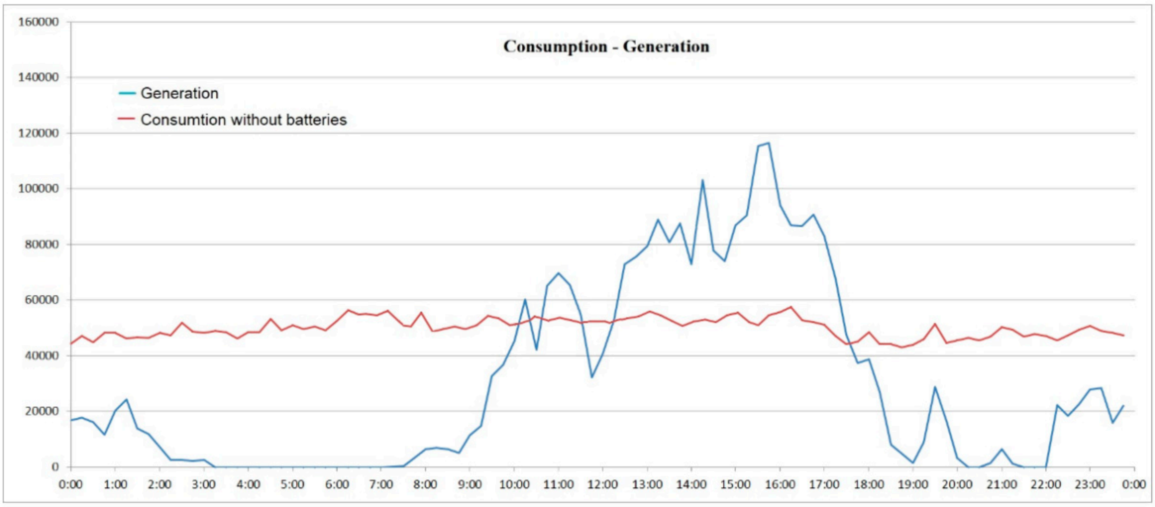


**Figure 10.** Daily grid generation–consumption graphic (before implementation). Source: prepared by authors based on the results of this study.

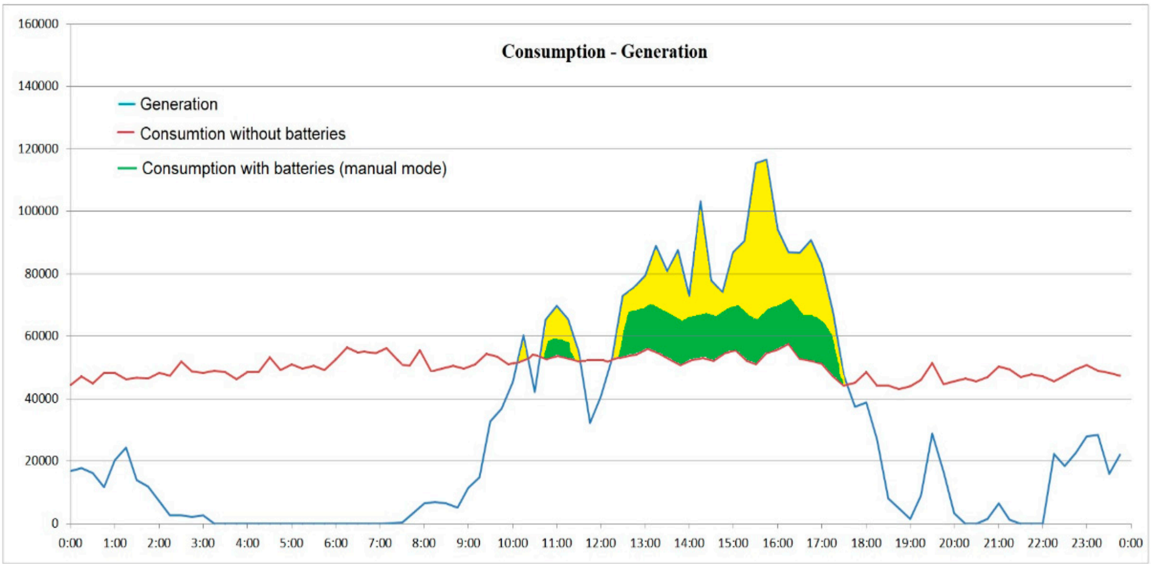


**Figure 11.** Daily grid generation–consumption graphic with batteries in manual mode (before implementation). Source: prepared by authors based on the results of this study.

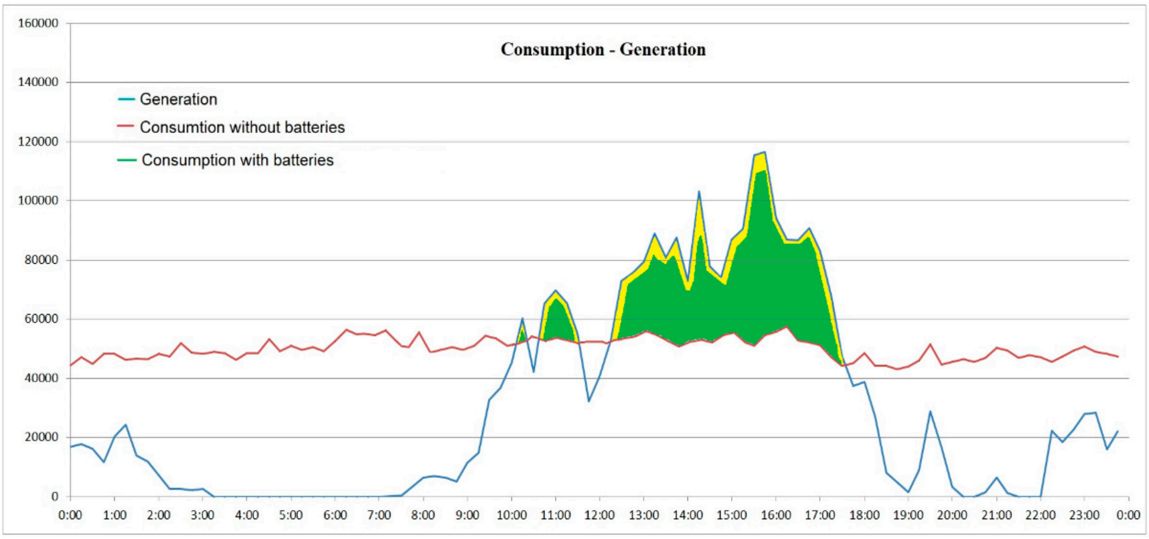


**Figure 12.** Daily microgrid generation–consumption after implementation. Source: prepared by authors based on the results of this study.

Figure 10 shows grid one-day generation and consumption curves before the conversion into a smart microgrid. The yellow area represents the power surplus for that day. As there is no communication between the elements of the grid, batteries cannot see there is a power surplus and they do not do anything.

Figure 11 shows grid one-day generation and consumption curves (before the conversion into a smart microgrid), including batteries' manual charge when there is a power surplus, that is, an operator sees the power surplus and active batteries charge with a fix power value. The green area represents energy stored in batteries. The yellow area represents the power surplus after the batteries' manual charge.

Figure 12 shows microgrid one-day generation and consumption curves after the implementation of the proposed procedure to convert the network section into a smart microgrid, including automatic batteries' charges when there is a power surplus. All the elements of the grid are integrated in the same control system developed with Home Assistant; therefore, the control system can detect the power surplus and send to the batteries the best charging instructions at all times. The green area represents energy stored in batteries (optimized). The yellow area represents the power surplus after the batteries' automatic charge.

## 5. Conclusions

This work presents an experimental application case for the conversion of a grid section with the generation, storage, and load systems into a smart microgrid that can be replicated in any grid section. It is based on a four-stage procedure: (1) identification and characterization of grid elements, (2) integration of grid elements into the same data network, (3) establishment of communication between grid elements, and (4) creation of a control grid framework that allows the management of the microgrid.

The most significant advantage is the use of the free tool Home Assistant to develop a software (CMEMS) to manage the microgrid. Although Home Assistant is designed for home automation, it offers all the necessary features to monitor, manage, and integrate in real time, and in a single HMI (CMEMS), all the renewable generation sources, storage systems, and consumption elements are connected to a grid to turn it into a smart microgrid. It has to be taken into account that Home Assistant is an open software, and therefore free; behind it, there is a large community of users who continually develop new capabilities, solve problems from previous versions, and answer questions in forums for free and without the need to hire any technical service. In addition, it can be managed from every device and everywhere due to its web server and its app for mobile phones.

The software developed is open, so new elements (generation sources, storage systems, or loads) can be included easily.

All this is an important advantage compared to other software shown in Section 2. All of them require the payment of a license fee, and have some limitations to connect some kind of devices. Moreover, the development and configuration are done by the company, so there is no option to modify it to include new systems on our own.

Finally, the developed software allows to implement strategies, after the analysis of the recorded data, to optimize the performance of the microgrid by maximising the use of renewable generation sources and reducing the consumption of the microgrids as much as possible with the help of storage systems. This will allow a reduction in the cost of the electricity bill.

**Author Contributions:** Conceptualization, O.I.-M.; methodology, O.I.-M., P.P.-C. and L.H.-C.; software, O.I.-M.; writing—original draft preparation O.I.-M. and P.P.-C.; writing—review and editing, L.H.-C., A.Z.-L. and O.D.-P.; supervision, R.V.-R. 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:** Confidential data.

**Acknowledgments:** The authors are grateful and would like to acknowledge the support granted by Carlos Barrera and Gonzalo Martín for valuable discussions.

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

## References

1. Ton, D.T.; Smith, M.A. The U.S. Department of Energy's Microgrid Initiative. *Electr. J.* **2012**, *25*, 84–94. [CrossRef]
2. Divya, K.C.; Østergaard, J. Battery energy storage technology for power systems-An overview. *Electr. Power Syst. Res.* **2009**, *79*, 511–520. [CrossRef]
3. Sachs, T.; Gründler, A.; Rusic, M.; Fridgen, G. Framing Microgrid Design from a Business and Information Systems Engineering Perspective. *Bus. Inf. Syst. Eng.* **2019**, *61*, 729–744. [CrossRef]
4. Jirdehi, M.A.; Tabar, V.S.; Ghassemzadeh, S.; Tohidi, S. Different aspects of microgrid management: A comprehensive review. *J. Energy Storage* **2020**, *30*, 101457. [CrossRef]

5. Li, Q.; Xu, Z.; Yang, L. Recent advancements on the development of microgrids. *J. Mod. Power Syst. Clean Energy* **2014**, *2*, 206–211. [CrossRef]
6. Jia, Y.; Lyu, X.; Lai, C.S.; Xu, Z.; Chen, M. A retroactive approach to microgrid real-time scheduling in quest of perfect dispatch solution. *J. Mod. Power Syst. Clean Energy* **2019**, *7*, 1608–1618. [CrossRef]
7. Kaur, A.; Kaushal, J.; Basak, P. A review on microgrid central controller. *Renew. Sustain. Energy Rev.* **2016**, *55*, 338–345. [CrossRef]
8. Ahmad Khan, A.; Naeem, M.; Iqbal, M.; Qaisar, S.; Anpalagan, A. A compendium of optimization objectives, constraints, tools and algorithms for energy management in microgrids. *Renew. Sustain. Energy Rev.* **2016**, *58*, 1664–1683. [CrossRef]
9. Gharavi, H.; Ghafurian, R. *Smart Grid: The Electric Energy System of the Future*; Gharavi, H., Ghafurian, R., Eds.; IEEE: New York, NY, USA, 2011; Volume 99, pp. 917–921.
10. Teufel, S.; Teufel, B. The Crowd Energy Concept. *J. Electron. Sci. Technol.* **2014**, *12*, 263–269. [CrossRef]
11. Aslam, W.; Soban, M.; Akhtar, F.; Zaffar, N.A. Smart meters for industrial energy conservation and efficiency optimization in Pakistan: Scope, technology and applications. *Renew. Sustain. Energy Rev.* **2015**, *44*, 933–943. [CrossRef]
12. Khan, K.R.; Siddiqui, M.S.; Saawy, Y.A.; Islam, N.; Rahman, A. Condition Monitoring of a Campus Microgrid Elements using Smart Sensors. *Procedia Comput. Sci.* **2019**, *163*, 109–116. [CrossRef]
13. Vargas-Salgado, C.; Aguila-Leon, J.; Chiñas-Palacios, C.; Hurtado-Perez, E. Low-cost web-based Supervisory Control and Data Acquisition system for a microgrid testbed: A case study in design and implementation for academic and research applications. *Heliyon* **2019**, *5*, e02474. [CrossRef]
14. Kunicki, M.; Borucki, S.; Zmarzły, D.; Frymus, J. Data acquisition system for on-line temperature monitoring in power transformers. *Measurement* **2020**, *161*, 107909. [CrossRef]
15. Nikolic, D.; Negnevitsky, M. Smart Grid in Isolated Power Systems–Practical Operational Experiences. *Energy Procedia* **2019**, *159*, 466–471. [CrossRef]
16. Eissa, M.M.; Awadalla, M.H.A. Centralized protection scheme for smart grid integrated with multiple renewable resources using Internet of Energy. *Glob. Transit.* **2019**, *1*, 50–60. [CrossRef]
17. Shuai, Z.; Sun, Y.; Shen, Z.J.; Tian, W.; Tu, C.; Li, Y.; Yin, X. Microgrid stability: Classification and a review. *Renew. Sustain. Energy Rev.* **2016**, *58*, 167–179. [CrossRef]
18. Hanah, A.; Farook, R.; Elias, S.J.; Rejab, M.R.A.; Fadzil, M.F.M.; Husin, Z. IoT Room Control And Monitoring System Using Rasberry Pi. In Proceedings of the 2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), Kedah, Malaysia, 27–29 November 2019; pp. 1–4. [CrossRef]
19. Singh, S.; Weeber, M.; Birke, K.P.; Sauer, A. Development and Utilization of a Framework for Data-Driven Life Cycle Management of Battery Cells. *Procedia Manuf.* **2020**, *43*, 431–438. [CrossRef]
20. Li, W.; Rentemeister, M.; Badeda, J.; Jöst, D.; Schulte, D.; Sauer, D.U. Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation. *J. Energy Storage* **2020**, *30*, 101557. [CrossRef]
21. Kim, T.; Makwana, D.; Adhikaree, A.; Vagdoda, J.; Lee, Y. Cloud-Based Battery Condition Monitoring and Fault Diagnosis Platform for Large-Scale Lithium-Ion Battery Energy Storage Systems. *Energies* **2018**, *11*, 125. [CrossRef]
22. Anandan, N.; Sivanesan, S.; Rama, S.; Bhuvaneswari, T. Wide area monitoring system for an electrical grid. *Energy Procedia* **2019**, *160*, 381–388. [CrossRef]
23. Petrollese, M.; Valverde, L.; Cocco, D.; Cau, G.; Guerra, J. Real-time integration of optimal generation scheduling with MPC for the energy management of a renewable hydrogen-based microgrid. *Appl. Energy* **2016**, *166*, 96–106. [CrossRef]
24. Kennedy, J.; Ciufo, P.; Agalgaonkar, A. Intelligent load management in microgrids. *IEEE Power Energy Soc. Gen. Meet.* **2012**, 1–8. [CrossRef]
25. Kornas, T.; Wittmann, D.; Daub, R.; Meyer, O.; Weihs, C.; Thiede, S.; Herrmann, C. Multi-Criteria Optimization in the Production of Lithium-Ion Batteries. *Procedia Manuf.* **2020**, *43*, 720–727. [CrossRef]
26. Augustins, E.; Jaunzems, D.; Rochas, C.; Kamenders, A. Managing energy efficiency of buildings: Analysis of ESCO experience in Latvia. *Energy Procedia* **2018**, *147*, 614–623. [CrossRef]
27. GE Digital. Advanced Energy Management System (AEMS). Available online: https://www.ge.com/digital/applications/transmission/advanced-energy-management-system-aems (accessed on 1 March 2021).
28. Operation Technology, Inc. Etap Powering Success. Available online: https://etap.com/es/home (accessed on 3 March 2021).
29. Open Systems International (OSI). Available online: https://www.osii.com/index.asp (accessed on 2 January 2021).
30. Wattics. Available online: https://www.wattics.com/dashboard/ (accessed on 1 January 2021).
31. Home Assistant. Available online: https://www.home-assistant.io/ (accessed on 3 March 2021).