DEVELOPMENT OF A SUPERVISORY SYSTEM FOR A POWER PLANT

FERNANDA MOURA QUINTÃO SILVA

MASTER'S THESIS DEFENSE DATE: 06/04/21

UNIVERSIDADE FEDERAL DE MINAS GERAIS – UFMG
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGEE

DEVELOPMENT OF A SUPERVISORY SYSTEM FOR A SMART POWER PLANT

FERNANDA MOURA QUINTÃO SILVA

Master's Thesis submitted to the Examination Board assigned by the Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, as a requirement for the acquisition of the Title of Master in Electrical Engineering.

Concentration Field: Power Engineering
Line of Research: Power Electronics

Advisor: Prof. Dr. Thales Alexandre Carvalho Maia
Co-Advisor: Prof. Dr. Igor Amariz Pires

BELO HORIZONTE
APRIL – 2021

S586d

Silva, Fernanda Moura Quintão.
Development of a supervisory system for a power plant [recurso eletrônico] / Fernanda Moura Quintão Silva. - 2021.
1 recurso online (vi, 67 f. : il., color.) : pdf.

Orientador: Thales Alexandre Carvalho Maia.
Coorientador: Igor Amariz Pires.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 57-60.

Bibliografia: f. 61-67.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Internet das coisas - Teses. 3. Energia renovável - Teses. I. Maia, Thales Alexandre Carvalho. II. Pires, Igor Amariz. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pela bibliotecária Roseli Alves de Oliveira CRB/6 2121
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
Programa de Pós-Graduação em Engenharia Elétrica

ATA DA 1229ª DEFESA DE DISSERTAÇÃO DE MESTRADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO da aluna **Fernanda Moura Quintão Silva** - registro de matrícula de número 2019716547. Às 13:00 horas do dia 06 do mês de abril de 2021, reuniu-se na Escola de Engenharia da UFMG a Comissão Examinadora da DISSERTAÇÃO DE MESTRADO para julgar, em exame final, o trabalho intitulado **"Development Of A Supervisory System For A Power Plant"** da Área de Concentração em Engenharia de Potência, Linha de Pesquisa Eletrônica de Potência. O Prof. Thales Alexandre Carvalho Maia, orientador da aluna, abriu a sessão apresentando os membros da Comissão e, dando continuidade aos trabalhos, informou aos presentes que, de acordo com o Regulamento do Programa no seu Art. 8.16, será considerado APROVADO na defesa da Dissertação de Mestrado o candidato que obtiver a aprovação unânime dos membros da Comissão Examinadora. Em seguida deu início à apresentação do trabalho pela Candidata. Ao final da apresentação seguiu-se a arguição da candidata pelos examinadores. Logo após o término da arguição a Comissão Examinadora se reuniu, sem a presença da Candidata e do público, e elegeu o Prof. **Thales Alexandre Carvalho Maia** para presidir a fase de avaliação do trabalho, constituída de deliberação individual de APROVAÇÃO ou de REPROVAÇÃO e expedição do resultado final. As deliberações individuais de cada membro da Comissão Examinadora foram as seguintes:

Membro da Comissão Examinadora	Instituição de Origem	Deliberação	Assinatura
Prof. Dr. Thales Alexandre Carvalho Maia - Orientador	(UFMG)	Aprovada	Thales Maia
Prof. Dr. Igor Amariz Pires - Co-orientador	(UFMG)	Aprovada	
Prof. Dr. Thiago Ribeiro de Oliveira	(UFMG)	Aprovada	Thiago Ribeiro de Oliveira
Prof. Dr. Marcos Augusto Menezes Vieira	DCC (UFMG)	Aprovada	Marcos Augusto Menezes Vieira
Prof. Dr. Thiago Augusto Araujo Moreira	DEM (CEFET-MG)	Aprovada	Thiago AA Moreira

Tendo como base as deliberações dos membros da Comissão Examinadora a Dissertação de Mestrado foi **aprovada**. O resultado final de **aprovada** foi comunicado publicamente à Candidata pelo Presidente da Comissão, ressaltando que a obtenção do Grau de Mestre em ENGENHARIA ELÉTRICA fica condicionada à entrega do TEXTO FINAL da Dissertação de Mestrado.A Candidata terá um prazo máximo de 30 (trinta) dias, a partir desta data, para fazer as CORREÇÕES DE FORMA e entregar o texto final da Dissertação de Mestrado na secretaria do PPGEE/UFMG. As correções de forma exigidas pelos membros da Comissão Examinadora deverão ser registradas em um exemplar do texto da Dissertação de Mestrado, cuja verificação ficará sob a responsabilidade do Presidente da Banca Examinadora. Nada mais havendo a tratar o Presidente encerrou a reunião e lavrou a presente ATA, que será assinada pelo Presidente da Comissão Examinadora. Belo Horizonte, 06 de abril de 2021.

Thales Alexandre C. Maia

ASSINATURA DO PRESIDENTE DA COMISSÃO EXAMINADORA

"Development Of A Supervisory System For A Power Plant"

Fernanda Moura Quintão Silva

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 06 de abril de 2021.

Por:

Thales Alexandre C. Maia

Prof. Dr. Thales Alexandre Carvalho Maia
(UFMG) - Orientador

Prof. Dr. Igor Amariz Pires
(UFMG) - Co-orientador

Thiago Ribeiro de Oliveira

Prof. Dr. Thiago Ribeiro de Oliveira
(UFMG)

Marcos Augusto Menezes Vieira

Prof. Dr. Marcos Augusto Menezes Vieira
DCC (UFMG)

Thiago Augusto Araujo Moreira

Prof. Dr. Thiago Augusto Araujo Moreira
DEM (CEFET-MG)

Acknowledgments

I would first like to thank my thesis advisor Dr. Thales Alexandre Carvalho Maia and my thesis co-advisor Dr. Igor Amariz Pires. They always steered me in the right direction throughout the development of this thesis and offered the guidance needed whilst allowing this work to be my own.

I would also like to thank Petrobras for the opportunity to join the project that allowed me to deepen my knowledge in promising areas. I would also like to thank Petrobras for the financial support.

I would also like to acknowledge the team involved in the project at Universidade Federal de Minas Gerais for without their efforts I would not be able to make progress in my own work.

Finally, I would like to express my gratitude to my parents, my sister, and to André for always providing me with support and encouragement throughout my studies. This accomplishment would not be possible without them. Thank you.

Resumo

Ao longo dos anos, ocorreu gradativamente um avanço tecnológico no campo da automação industrial, que impulsionou uma evolução constante dos sistemas de supervisão e controle empregados no monitoramento e gerenciamento de plantas. Hoje em dia, com a chegada da quarta revolução industrial que trouxe o conceito de Internet das Coisas, é possível criar um ambiente de compartilhamento de informações que simplifica e aumenta a eficácia dos sistemas supervisório e de controle.

Este trabalho aplica esses conhecimentos por meio da implementação de um novo sistema de supervisão usando conceitos da Internet das Coisas em uma usina solar real. O sistema supervisório desenvolvido neste trabalho é capaz de fornecer ao usuário um ambiente amigável para monitorar qualquer medição desejada e/ou estados dos dispositivos.

Esta dissertação apresenta o desenvolvimento de um sistema supervisório que utiliza dispositivos integrados à Internet das Coisas. As técnicas usadas podem ser aplicadas para qualquer tipo de planta. Este trabalho detalha as ferramentas e serviços que estão disponíveis ao usar esses conceitos novos e como pode ser benéfico para a planta ter um sistema de supervisão associado à tecnologia de Internet das Coisas. Os benefícios são ainda mais evidentes quando é apresentada uma comparação entre o sistema supervisório tradicional e o sistema aqui desenvolvido.

Palavras-chave: sistema supervisório, fontes renováveis, Internet das Coisas

Abstract

Throughout the years, there has been a steady technological advance in the industrial automation field, that has propelled a constant evolution of the supervisory and control systems employed to monitor and manage plants, including power plants. Nowadays, with the coming of the fourth industrial revolution which has brought the concept of the Internet of Things, it’s possible to apply this concept to create an information sharing environment that simplifies and increases the effectiveness of supervisory and control systems.

This work applies this research field by implementing a new supervisory system using the ideas of the concept of Internet of Things in a real solar power plant. The supervisory system developed in this work is capable of providing to a user a friendly environment to monitor any desired measurement and/or status of the devices that compose this power plant.

This dissertation presents the development of this supervisory system that uses devices integrated with the Internet of Things that can be implemented in any type of power plant. This work goes in detail about the tools and services that are available when using these novel concepts and how it can be beneficial for the plant having a supervisory system developed employing this technology and strategy. The benefits are even more highlighted when a comparison between traditional supervisory systems and the system developed here is presented.

Keywords: supervisory system, renewable sources, Internet of Things

Summary

1 Introduction

The main problems with renewable energy sources are cost and availability, for example, wind and solar power are not always available when and where needed – the energy is not “dispatchable”. Daily and seasonal changes affect greatly the generation of energy. Oscillations in wind speed and solar radiation would translate directly to variations in energy production, reducing its predictability and resulting in an intermittent generation (CAMACHO, SAMAD, GARCIA-SANZ, & HISKENS, 2010). Control and energy storage have been used to mitigate this type of problem and significant progress has been made throughout the years, making it possible to employ this form of energy as a reliable and viable source for a cleaner generation. Therefore, there has been an effort in order to better understand the possibilities of this energy source besides the possible enhancements which can be done due to the advancement of technology.

As an interested party in developing clean energy, the department of Electrical Engineering of UFMG (Universidade Federal de Minas Gerais) concluded the construction of the Usina Experimental Fotovoltaica TESLA Engenharia de Potência with capacity of 37 kWp (FOUREAUX, MARRA, & ANTHONY, 2014). The production of generated energy by this power plant is connected to the grid. The solar power plant is described in greater details in a future section of this dissertation.

Alongside the rise in interest in renewable energy sources, it’s also noticeable a technological advance in the industrial automation field. This field plays a vital part to the monitoring and control of photovoltaic power plants (data processing and commands sending to the plant). A number of acquisition and processing data systems have already been developed to the measurement, processing and acquisition of environmental data, evaluation of the performance of the power plant, amongst others (FORERO, HERNÁNDEZ, & GORDILLO, 2006).

With the advancement of technology in wired and wireless networks, there is a tendency nowadays to renew the industrial automation concepts and to associate them to the Internet of Things concept. The Internet of Things (IoT) is an information sharing environment, in which the devices are connected to wired or wireless networks, making easy remote access viable. Considering the growing interest in clean energy and the increasingly recurrent construction of renewable energy sites and the necessity of constant monitoring of variables related to the application (environmental and electrical data), applying the power of the Internet of Things to micro-grids seems extremely promising (ADHYA, SAHA, DAS, JANA, & SAHA, 2016).

1.1 Objectives

This dissertation work is part of the multidisciplinary project under development by the department of Electrical Engineering of UFMG. This project is currently funded by the Petrobras P&D Project "Desenvolvimento de Sistema Smart Battery e Planta Piloto de Armazenamento de Energia Associado à Geração Distribuída de Energia Elétrica" (development of a smart battery and a pilot plant of energy storage associated to distributed generation of electric energy in free translation), ANEEL 21/2016, PD-00553-0046/2016. The goal of this dissertation is to provide the control and monitoring of the photovoltaic system, applying the concept of the Internet of Things to the plant automation for the acquisition and processing of data as well as plant control, i.e., this concept is used for the supervisory system development.

Currently, the analyzed solar power plant is monitored by a supervisory system developed by a third party company. The contract for this service must be renewed every three years; if not, the user doesn't have access to the real-time nor to the historical data. This supervisory system is capable of presenting an overview of environmental and electrical variables of the system. It's possible to monitor graphically these field variables along any specified period of time. However, since this system is made by a third party company, it's not easily customizable to add or delete variables; meaning that further expansions - such as the inclusion of a hybrid inverter and the battery banks - would require further developments by this company. Also, this system doesn't provide any information in regards to the current state of the devices nor does it show a schematic of the circuitry. And finally the current supervisory system doesn't allow the sending of commands to the field devices, preventing the control of power flow.

The main idea is to provide a custom-made monitoring and control system of easy remote access of all environmental and electrical variables available by the photovoltaic panels, inverters and battery banks. This supervisory system should be flexible – easy to incorporate expansions and/or changes -, adaptable to research, and cheaper. That means, the main idea is to provide a free supervisory system that is able to surpass the limitations cited above presented in the supervisory system currently in use in the power plant.

The specific goals are:

- Develop a low-cost open-source flexible supervisory system for the monitoring of the measured values of the analyzed photovoltaic power plant, using IoT devices and adapting non-IoT devices to the new generation of current automation (IoT);
- Develop a system that allows communication between IoT devices in different networks in order to incorporate data from different networks;
- Develop a communication structure for a battery management system and for the power flow control;
- Develop a graph-plotting feature that allows sampling rate configuration in order to provide graphics able to track all the variations in the variables throughout the day.

1.2 Methodology

Due to the limitations of the current supervisory system - its structure's rigidity, non-adaptability, and difficulty in further expansions - and due to its incapacity of fulfilling the specific goals, another set of tools has to be used in order to provide the power plant with a flexible supervisory system that is easily expandable. Also, another motivation for developing a new supervisory system is being the design authority of the system, removing the necessity of further development by a third-party company. Since it's interesting to mesh the concept of the Internet of Things and renewable energy generation, the idea is to find tools that are already aligned with IoT ideas. Also, it's important to find software that is easily programmable in order to create a supervisory system that fulfills any requirement (monitoring and controlling) in order to make any necessary expansions that may occur.

There is an array of open-source software options that would fulfill the requirements. The one that is the most popular and the most versatile is Node-Red. Another tool that would be a good complement to Node-Red to provide the power plant with a complete supervisory system is Emoncms, that is frequently used to monitor generated and consumed energy and to develop dashboards. Both are free Web-Service platforms already integrated with the IoT concept. Node-Red is a tool based on visual programming flow to connect hardware devices, APIs and online services. Emoncms is an application to process, register and visualize energy and other environmental and electrical data. In later sections of this master's thesis, the advantages of employing Node-Red and Emoncms in this project when comparing to other tools is further detailed.

With the platforms to program the software already chosen, it's necessary to select the hardware that acts as a server for necessary services for the development of the monitoring and control system of the power plant. The chosen hardware is the microcomputer Raspberry Pi 3 B+. The Raspberry Pi is a series of compact single-board computers. To establish the supervisory system, it is necessary to implement some services in order to gather the data, process it and display it in the screens of the supervisory system. To this end, it's necessary to establish communication between the field equipment and the application (supervisory system), data storage, and a backup system. Three servers are established using the Raspberry: communication server, database and application server and the backup server as seen in Figure 1-1.

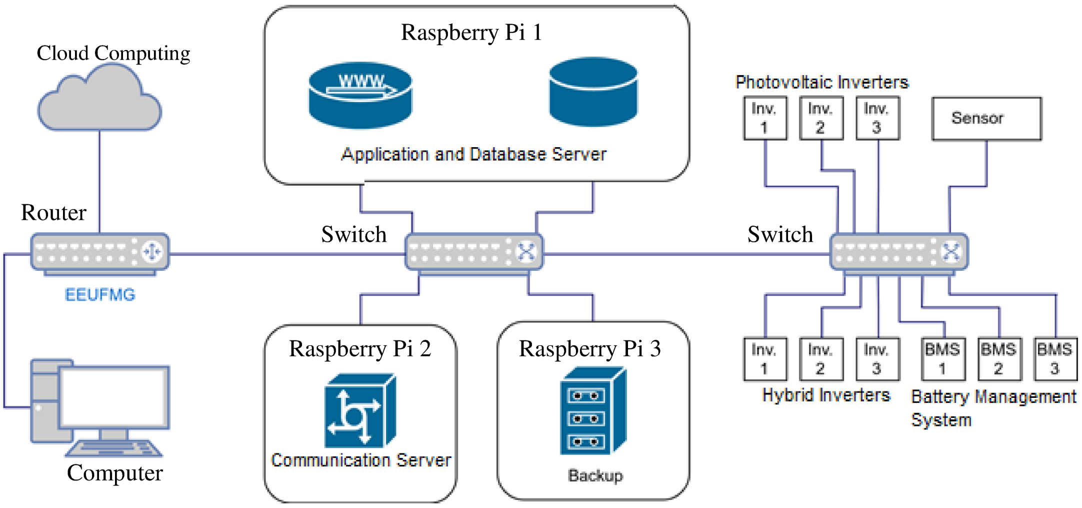


Figure 1-1 – Schematic of the proposed servers architecture for the supervisory system showcasing the modularized architecture

In this schematic, it’s possible to see that the services are separated in different servers; the system is modularized. If one of the servers is not working properly or if a server becomes unavailable, the remaining servers are not affected by this failure; the supervisory system would not be completely affected and the maintenance would be simpler since the defect would be more traceable to the defective server and it would be easier to isolate it and correct the error.

Some devices in the power plant can be easily incorporated in the network established in the power plant, such as the inverters and the energy meter that are already equipped to communicate with a wired/wireless network. But there are some gauges that are not; for these, it`s necessary to adapt them to communicate with the power plant network established for the project. For this purpose, another device has to be connected to the gauges so that their data is also available for the development of the supervisory system.

The proposed scheme can be seen in Figure 1-2. The figure is not on scale – the elements are depicted in a way to show the proposed scheme.

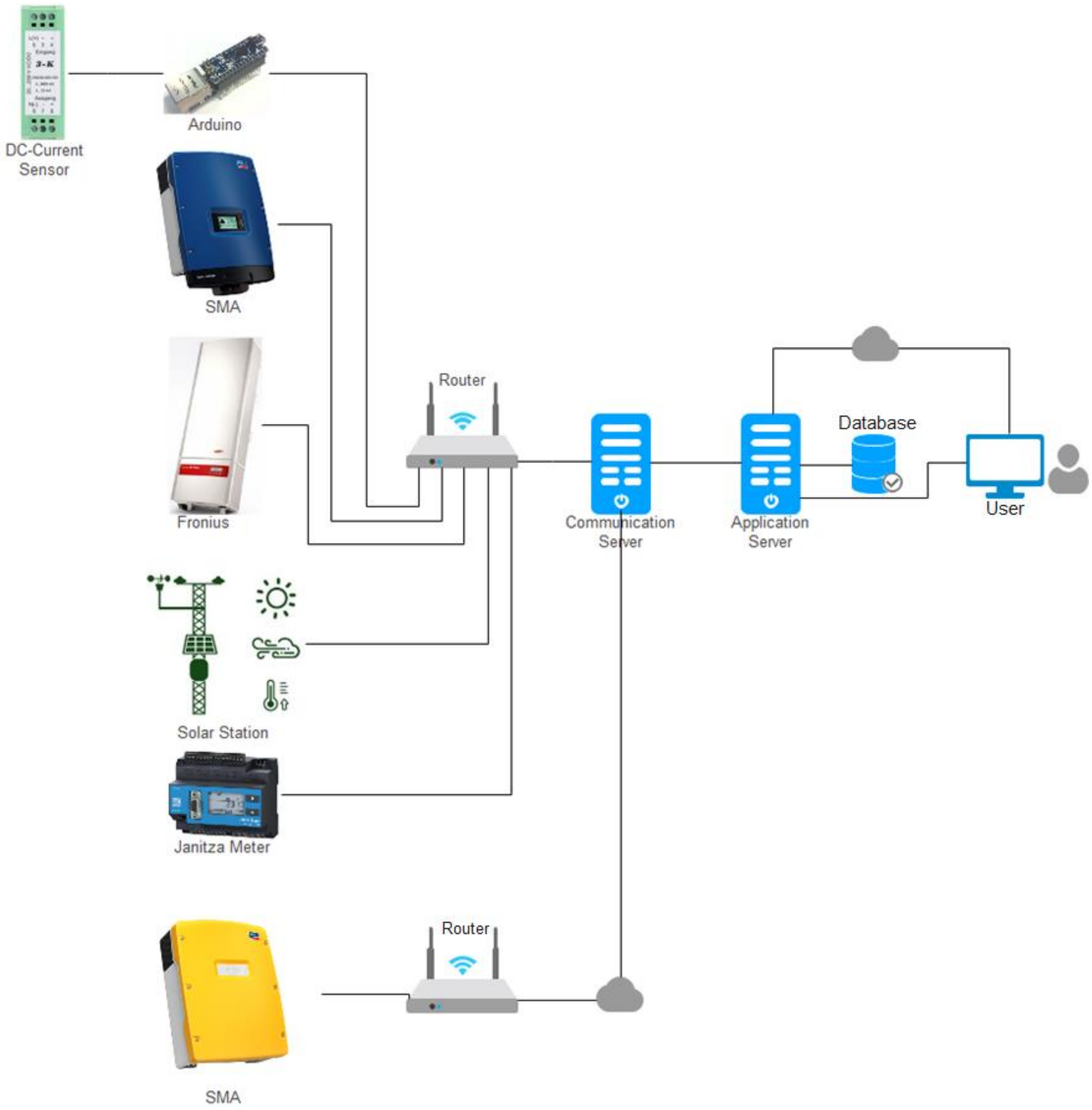


Figure 1-2 – Proposed structure for the data acquisition and supervisory system including the data flow path

Figure 1-2 shows the proposed schematic for the development of the supervisory system since the data acquisition from the field devices going through the communication and application servers, ending in the user terminals, local and remote via the Internet. Some field devices are already configured to establish communication with a network whilst others aren’t; the latters have to be connected to an intermediary device in order to provide their data to the network. There are devices that are not in the same network as the supervisory system servers. These devices communicate with the supervisory system via the Internet; this communication is shown in Figure 1-2 with the cloud symbol.

It’s important to note all the devices and servers are wired connected to the routers and switches. The wireless connection is negatively impacted when the distance between the router and the device is long. Besides, wireless connections can be obstructed by items and structures, such as walls, ceilings, and furniture. Finally, the network security is a greater issue in wireless connections, meaning that the information is less secure and the data can be more easily compromised.

1.3 Scientific Contributions

Throughout the development of this dissertation, some scientific contributions were made. These contributions were manifested as scientific papers, cited bellow:

- “A Long Term Evaluation of Photovoltaic Systems under Power Quality Problems” in collaboration with Prof. Dr. Thales Maia and Prof. Dr. Igor Pires – this paper analyzed the impact of power quality problems in solar power devices, motivated by the failure of one photovoltaic inverter. The paper was published in the XXIII Congresso Brasileiro de Automática;
- “Design of a SCADA System Based on Open-Source Tools” in collaboration with Prof. Dr. Braz Cardoso Filho, Prof. Dr. Thales Maia, and Prof. Dr. Igor Pires – this paper presented the development of a supervisory system employing techniques related to the Internet of Things. This paper is submitted for approval in the IEEE/IAS International Conference on Industry Applications 2021.

1.4 Text Organization

The text was organized in the following manner: the current chapter presents the motivation, the specific goals and the used methodology to reach them.

In **Chapter 2**, there is a bibliographic review on micro-grids and the systems that have been used to monitor and control plants traditionally, in addition to detailing the specific plant for this work. Also, the Internet of Things (IoT) concept is presented in more detail and the two programs used for data acquisition and control (Node-RED) and for supervising variables (Emoncms) are explained in depth.

In **Chapter 3**, the schemes for the data acquisition and control developed in Node-Red and for the supervisory system developed in Emoncms are proposed. Furthermore, the proposed logic proposed for the integration of the Arduino to the power plant is outlined.

In **Chapter 4**, the results are shown. There is a comparison between the current supervisory system provided by a third party company and the supervisory system developed along this dissertation.

The main conclusions and proposed continuation are presented in **Chapter 5**. The bibliographic references cited throughout the text are listed at the end.

2 Bibliographic Review

In this section, the subjects related to the research are introduced in order to establish important definitions that are used in subsequent chapters to develop the supervisory system for the power plant.

The techniques used to develop the supervisory system can be integrated to any plant. However, in order to deepen the amassed knowledge, it's integrated to a specific power plant that is built as a micro-grid. So, the first topic of interest to be introduced is micro-grids. It's also interesting to present the techniques traditionally used when developing a supervisory system in order to compare them with the novel techniques employed here in order to build a new type of supervisory system.

One of the goals of this project when developing the supervisory system is to follow the current trend in the automation field, which means employing the ideas of the new wave of industrial revolution: the Internet of Things (IoT). In this chapter, a short overview of this new concept, the devices related to it and its key challenges are discussed in order to prepare the groundwork for the development of the supervisory system.

After this bibliographic review of concepts and traditional methods, the tools selected for the building of the supervisory system in this project are introduced as is the power plant of interest.

2.1 Micro-grids

For the context of the project analyzed in this dissertation, the concept of micro-grids implies a group of loads and micro-sources that operate as a single unity that provides energy to the area. The components can be assemblies of micro-turbines, fuel cells, photovoltaic panels and other small generators, storage systems and controllable loads. To the final consumer, the micro-grid can be seen as a means to satisfy specific requirements, such as improvement of the local reliability, reduction of losses, amongst others (MARINHO, 2011).

Micro-grids combine modular energy sources, such as wind, photovoltaic, with energy storage systems and controllable loads, forming a low-voltage distribution system. A micro-grid is a small-scale, self-supporting network by on-site generation energy, able to disconnect from the main grid due to sustainability and/or energy security (VENAYAGAMOORTHY G. K., SHARMA, GAUTAM, & AHMADI, 2016). An example of a generic micro-grid can be seen in Figure 2-1.

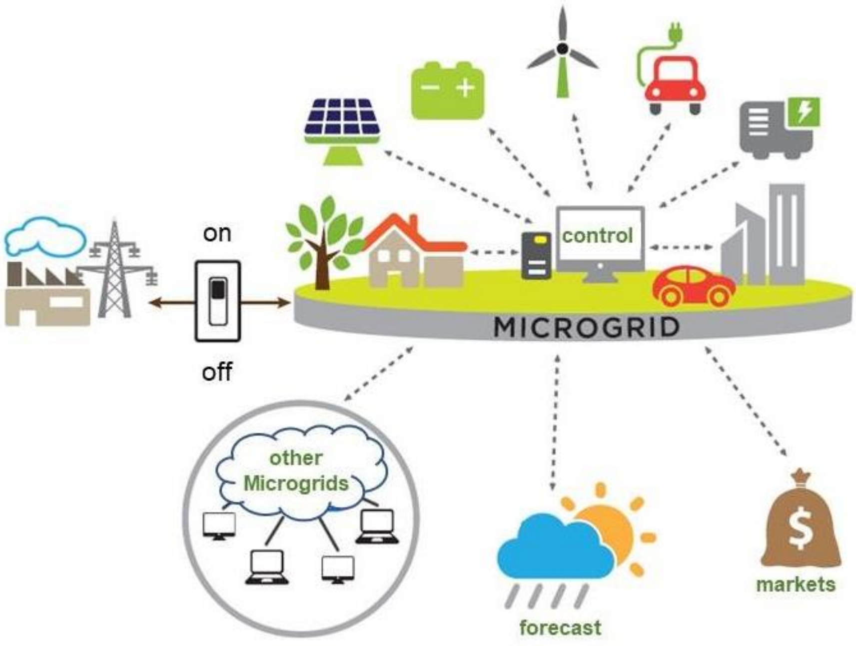


Figure 2-1 – Illustration of a generic micro-grid, highlighting the type of data that can be processed by its control and supervisory systems (STADLER & NASLÉ, 2019)

Figure 2-1 depicts a generic micro-grid that is able to connect and disconnect to a main power plant. It also depicts the type of information that can be monitored when interacting with the supervisory and control system of a micro-grid, such as: data from renewable power plants, energy storage systems, weather forecast, markets, and other micro-grids.

Distributed energy is the main generation system in micro-grids architectures. According to the International Energy Agency (IEA), distributed generation is the energy production from wind and/or solar, amongst others. The energy generation technology is close to the consumer unity. Therefore, this type of generation doesn’t require high voltage transmission lines for the energy transportation to the consumer (NARUTO, 2017). Micro-grids can operate in two distinct modes: connected mode, the micro-grid is connected to the main grid, and island mode, the micro-grid is not connected to the main grid (ÁLVAREZ, CAMPOS, GARCÍA, GONZÁLEZ, & DÍEZ, 2010).

Micro-grids are an effective tool to employ in the use of local sources of renewable energy (non-renewable as well). There are some advantages when using micro-grids: improved energy efficiency, reduced energy consumption and environmental impact, higher reliability and resilience and more cost efficient maintenance (ÁLVAREZ, CAMPOS, GARCÍA, GONZÁLEZ, & DÍEZ, 2010). Micro-grids also reduce long-distance electricity losses as electricity transmission accounts for approximately 4% of total power consumption (HU, CHEN, CHEN, & CHANG, 2013).

Due to the fact that micro-grids are small-scale and they are usually provided with a robust communication system, the control and monitoring are decentralized, explaining the reason they have higher reliability and resilience (NARUTO, 2017).

The output of renewable generation may vary according to the weather condition of the day (wind speed, solar radiation to name a few factors) and time of day. This means that the majority of the renewable resources can't guarantee a continuous and steady power generation on their own. Since this may have a destabilizing effect, it's important to consider it when there is a desire to integrate renewable energy sources into traditional grids; one of the tools to mitigate this concern is to incorporate energy storage systems into the micro-grid. (VENAYAGAMOORTHY G. K., SHARMA, GAUTAM, & AHMADI, 2016).

Since control and supervision are vital to energy generation and distribution especially for micro-grids that can be disconnected from the main grid and are related to energy storage, it's important to establish an information flow between the micro-grid and the user in order to enable tracking and monitoring of the processes and electrical and environmental variables. For this reason, it's interesting to review the most common techniques employed in the development of supervisory systems (not only for micro-grids). The traditional supervisory system is presented next in order to familiarize oneself with the commonly used tools so that the innovations in the supervisory system developed in this dissertation become clearer.

2.2 Traditional Supervisory Systems

SCADA (supervisory control and data acquisition) is a system of hardware and software components that enable process control. Automatic data acquisition system has been used for monitoring system performance and control. SCADA is traditionally a central control that consists of controller's network interfaces, input/output, communication equipment, and software. It's used to monitor and control the equipment in processes, gathering data from sensors and instruments to showcase an overview of the whole process.

Generally, the SCADA includes local processors, operating (field) equipment, PLCs (programmable logical controller), instruments, sensors, remote terminal unit (RTU), intelligent electronic devices (IEDs), and a computer with a human interface (RAGHVENDRA, 2019). Figure 2-2 shows an overview of the structure.

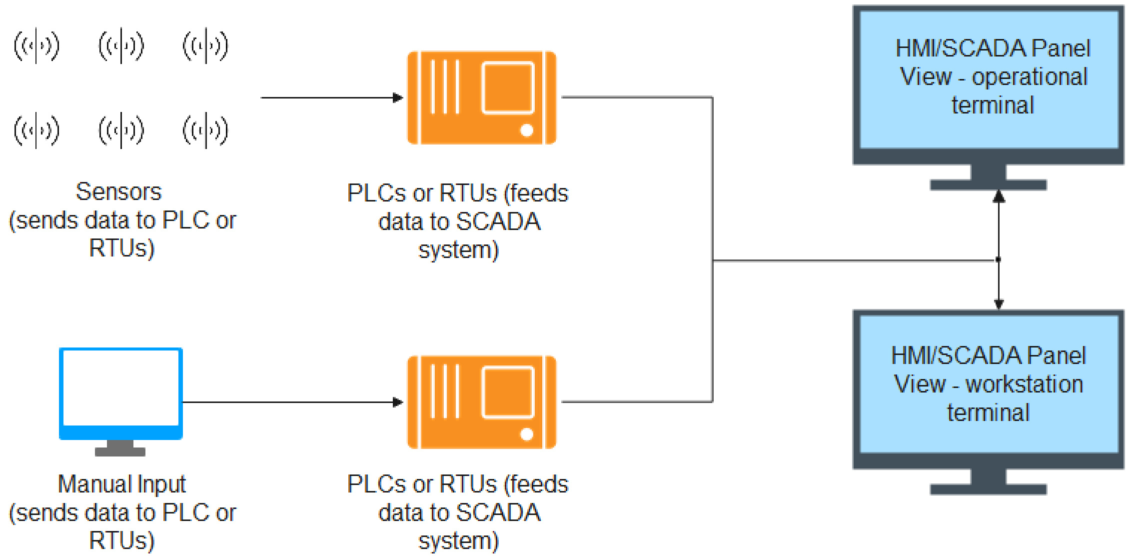


Figure 2-2 - Representation of the hardware architecture of a SCADA system. Modified from (Inductive Automation, 2020)

The supervisory system acts as a link between the user in the control room and the equipment like PLCs, RTUs, IEDS, and sensors. The RTUs are connected to field devices; they are controlled by microprocessors and are used for transmitting recorded data to the supervisory system. They also receive commands in order to control the field devices. The PLCs are connected to sensors in order to convert the sensor output signal into digital data (RAGHVENDRA, 2019).

Traditionally, a common characteristic of such structure is the use of data loggers or microcontrollers for measuring and acquiring data and transmitting them to a user terminal (personal computer, microcomputer) via a serial port (RS232) (FORERO, HERNÁNDEZ, & GORDILLO, 2006), (MUKARO & CARELSE, 1999). A system is developed in order to collect, record, and transmit the measured data to a computer. The system developed in (MUKARO & CARELSE, 1999) is seen in Figure 2-3 in a simplified representation.

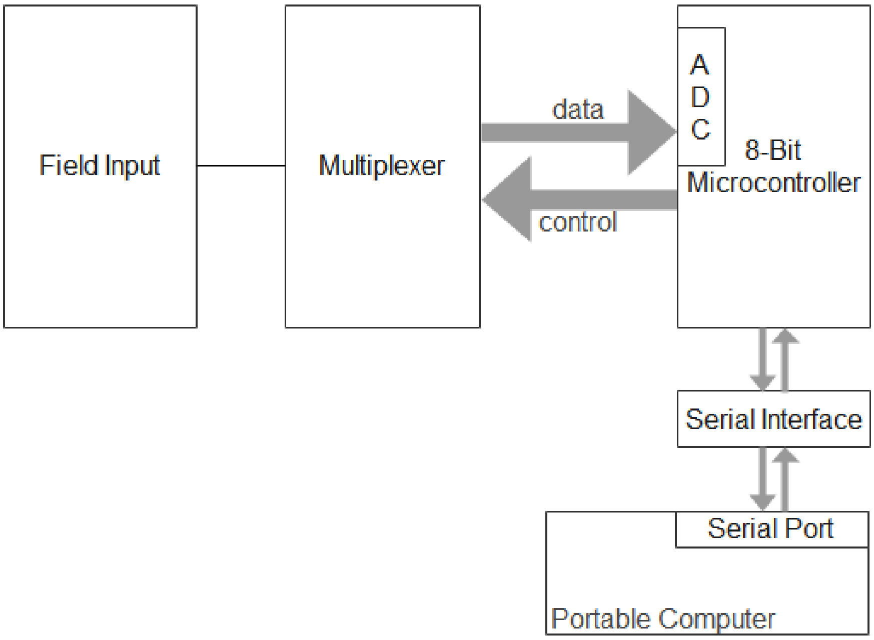


Figure 2-3 - Block diagram of a traditional data acquisition system using serial communication interface

As it can be seen, this acquisition scheme relies on an I/O (input/output) microcontroller in order to retrieve the measured data and it transmits this data to a portable computer via wired connection through a serial port. The data is then downloaded to the computer via the communication protocol and then it can be accessed for further analysis. The supervisory and control system are developed on this specific portable computer using assembly language and it's only accessible in it (MUKARO & CARELSE, 1999).

In (BENGHANEM & MAAFI, 1998), this methodology is also employed even though another microcontroller is used. This system uses bidirectional communication; the system receives data from the sensor and sends commands through this serial protocol. This data is treated and converted to physical values and used by the central computer to study the performance of the photovoltaic system (BENGHANEM &

MAAFI, 1998); this study can only be done in this computer where the supervisory system was programmed and the data is only accessible inside the local network.

Due to some limitations when using serial port, such as limited data transmission when data acquisition at high sampling rates is required, some research has been done in order to minimize the impacts of this limitation. In (FORERO, HERNÁNDEZ, & GORDILLO, 2006), to avoid this, it`s proposed to use I/O modular devices (with field point analog modules) and data acquisition cards with high-speed analog to digital conversion. This project also relies on a local supervisory system with wired-serial connections between the user terminal and the field devices, enabling only local users the monitoring of the power plant. In (KOUTROULIS & KALAITZAKIS, 2003), the authors use the same methodology utilizing high-speed cards and I/O modules.

More recently, with the technological advancements in wired/wireless networks, the serial communication has been replaced by the establishment of a local network. In (DUMITRU & GLIGOR, 2012), the authors present the more popular approach nowadays for the development of a SCADA (supervisory control and data acquisition) system for any plant. The scheme discussed in this study, implements a modernization in comparison to the previous ones discussed in this dissertation up to this point by establishing a local network. Most operations are automatically executed by the local equipment of data acquisition and control (RTU) or by the programmable logic control unit (PLC). A generic SCADA system can be seen in Figure 2-4.

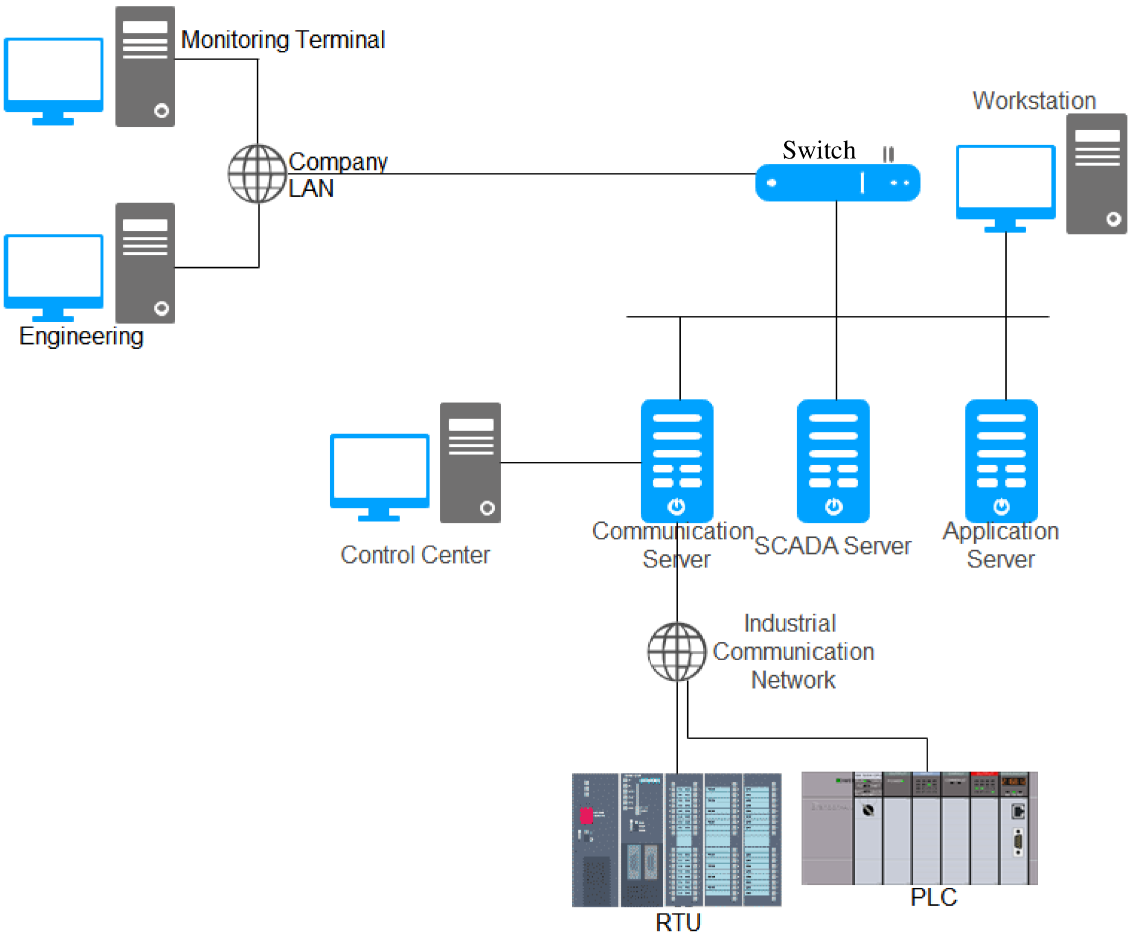


Figure 2-4 – Illustration of a SCADA, showcasing the data-acquisition devices, servers, workstations and user terminals

Data acquisition begins at the RTU or PLC and involves reading the measurements and the state of gauges that are transmitted to SCADA application. The monitoring and control systems are made by dedicated devices. The values of the measured parameters are presented graphically in real time or historically in the users terminals connected to the local network; in Figure 2-4, by serial communication, router of the network or by the communication server. The control means sending commands towards the system from the user interface (DUMITRU & GLIGOR, 2012). This structure shows progress in regards to the other discussed here since it considers a LAN (local area network), which can be wired or wireless, for the building of a communication path between the field devices (RTU, Local Terminal, PLC) and the SCADA serves, the communication server and the user interfaces.

In (PAPAGEORGAS, PIROMALIS, ANTONAKOGLOU, VOKAS, & TSELES, 2013), the authors propose a supervisory system that uses a similar structure presented in Figure 2-4 for a photovoltaic solar plant. The field devices communicate with a local gateway through a serial bus. The gateway provides wireless communication services for the devices using wireless sensor network (WSN) technology with a central terminal, where the data is processed. It's interesting to notice in this research work that the devices are not IoT devices – they are merely wireless; they can't access the Internet. However, after the data is processed in the central terminal, it's published on the Internet to remote users using publishing services such as Cosm, that is used in (PAPAGEORGAS, PIROMALIS, ANTONAKOGLOU, VOKAS, & TSELES, 2013). This work shows a step towards the concept of the Internet of Things.

2.3 Internet of Things

Due to the advancements in wired/wireless networks and in sensing, the Internet of Things has been identified as one of the emerging technologies in IT (GUBBIA, BUYYAB, MARUSIC, & PALANISWAMI, 2013). It is a global Internet-based information architecture, facilitating the exchange of goods and services in global supply chain networks (WEBER, 2010). The Internet of Things is a new technology paradigm that enables the setup of a global network of machines and devices that are capable of interacting with each other. The main differential of this new technology is the possibility to enable devices to communicate with each other and interact with inventory systems, customer support systems, and business applications. Also, the Internet of Thing allows the remote control of devices since the communication is performed via the Internet. Due to IoT, a whole slew of applications emerge, especially when considering industry and customer oriented applications, such as smart houses, smart neighborhoods, and smart cars (LEE & LEE, 2015). However, embedded in this new technology, there are a lot of new challenges: data management (due to the high volume of data), privacy and security, and standardization (WORTMANN & FLÜCTHER, 2015).

The devices and gauges capable of interacting with an Internet of Things environment are smart devices. Smart devices are equipment capable of communication and computation; they range from simple sensing gauges to home appliances and smartphones. The key features of smart devices are autonomy, connectivity, context-awareness, user-interaction, mobility, and data storage (SILVERIO-FERNÁNDEZ, RENUKAPPA, & SURESH, 2018).

The Internet of Things connects a vast amount of objects, creating an extremely large data flow and a necessity for large data storages. Besides these two factors, the Internet of Things has to deal with new challenges, such as data security and privacy. The architecture for IoT needs to incorporate features that address the new challenges like scalability, interoperability, and reliability, amongst others. The basic architecture for the Internet of Things is seen in Figure 2-5 (KHAN, KHAN, ZAHEER, & KHAN, 2012).

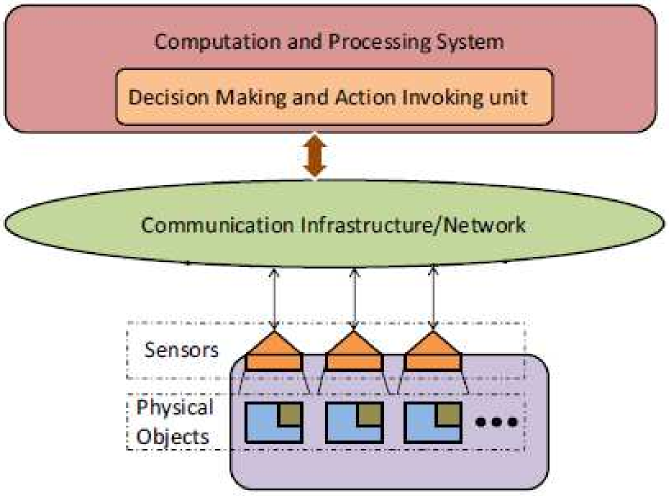


Figure 2-5 – Architecture of the IoT structure, showcasing each fundamental block. Modified from (KHAN, KHAN, ZAHEER, & KHAN, 2012)

The initial block depicting the physical objects and sensors is related to the identification of each individual component when inserted in the context of the network; each device has its own IP address for example. As it’s represented by the arrows, the physical objects can send/receive data to/from the network. The communication infrastructure is related to the communication protocols used to establish the link between devices, devices and services, devices and users. The decision making inside the computation and processing unit is related to the ability of extracting information from the devices data in order to activate certain services and to execute local algorithms (KHAN, KHAN, ZAHEER, & KHAN, 2012). The basic structure for IoT is composed of 5 layers as seen in Figure 2-6.

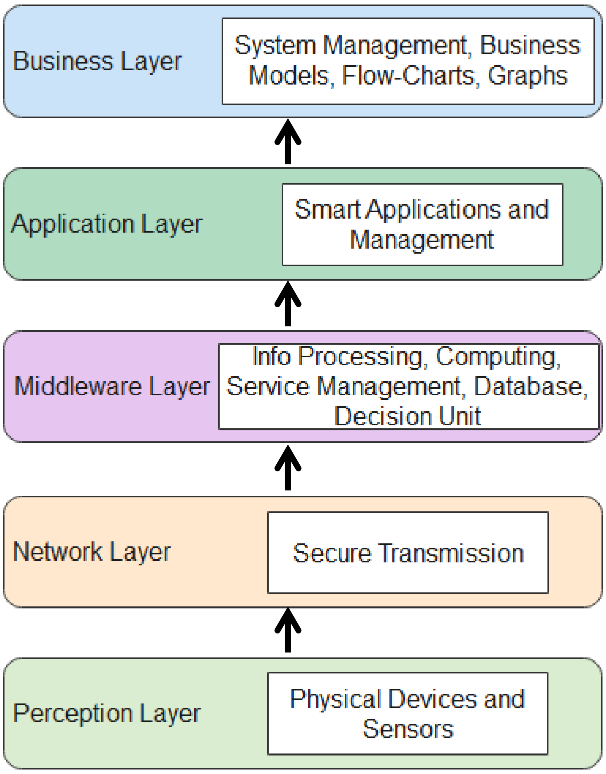


Figure 2-6 – IoT 5 structural layers – layers applied in the IoT environment from sensing and data acquisition until system management. Modified from (KHAN, KHAN, ZAHEER, & KHAN, 2012)

The bottom layer is the perception (or device) layer; it consists of the physical equipment and the sensing devices. This layer is responsible for the identification of the devices and data acquisition. This information is then passed to the network (or transmission) layer for its secure transmission to the data processing system. The transmission can be wired or wireless. The data is then transmitted to the middleware layer. This layer has links to the database; it's capable of storing the data. The middleware layer processes the data and activates services based on the results of the data processing. The application layer is responsible for global management of the application based on the data processing on the previous layer. And lastly, the business layer is responsible for the management of the IoT system, including applications and services (KHAN, KHAN, ZAHEER, & KHAN, 2012).

2.3.1 Smart Devices

Smart devices are able to connect to the Internet via wired/wireless connections, being capable of establishing data exchange with other smart devices and allowing control and access to remote users and services. A smart device is a context-aware equipment that can be used as a service provider and can communicate with devices in different networks (SILVERIO-FERNÁNDEZ, RENUKAPPA, & SURESH, 2018).

Smart devices implement three steps: data acquisition, data process, and communication. In the first step, data acquisition, the device acquires the data measured by its sensing unit. Then, the raw data is processed by the processing unit. And, finally, the processed data is transmitted to the network via a communication protocol (JIMÉNEZ-GARCÍA, BASELGA-MASIA, POZA-LUJÁN, MUNERA, POSADAS-YAGÜE, & SIMÓ-TEN, 2014).

The key features considered by (SILVERIO-FERNÁNDEZ, RENUKAPPA, & SURESH, 2018) for a device to be considered smart are: autonomy, context-awareness and connectivity. Autonomy would relate to the feature of smart devices of being capable of performing tasks without external interference – no direct user command necessary to perform the task. Context-awareness means the device is able to perceive environmental information via sensors or sensing units. The connectivity is the feature of establishing a connection to a wired/wireless network of any size in order to achieve a goal: access the Internet, share information with other devices, amongst others.

With technological advancements, the devices processing power and storage abilities increased whilst their sizes were reduced. Due to its features, the smart device technology can lead to various possibilities: smart homes, smart cities, industry applications, smart metering, and smart security to name a few (KHAN, KHAN, ZAHEER, & KHAN, 2012). However, there are still some challenges that must be investigated regarding smart devices specially when considering the popularity of this technology in different fields nowadays.

2.3.2 Key Challenges in IoT

There are some challenges when it comes to IoT; some of them are (KHAN, KHAN, ZAHEER, & KHAN, 2012):

- Naming and identity management: Each device connected to the network must have a unique identity over the Internet – an efficient naming and identity system that is capable of managing a large number of devices is required;

- Interoperability and standardization: Many manufactures have their own protocols – it's important to standardize them to guarantee interoperability;
- Devices safety and security: It's necessary to prevent security breach in order to protect the integrity of the devices;
- Network security: The data transmission should be secure and capable enough to protect the integrity of the data and to prevent external interference or monitoring.

Even with these challenges, it's important to note that this new technology can provide significant benefits (KHAN, KHAN, ZAHEER, & KHAN, 2012). As discussed previously, the smart devices are capable of connecting to the Internet. Therefore, it's interesting to develop a supervisory system for smart plants with Web-Services that are already built with the IoT concept embedded.

2.4 Open-Source IoT Tools for Supervisory System

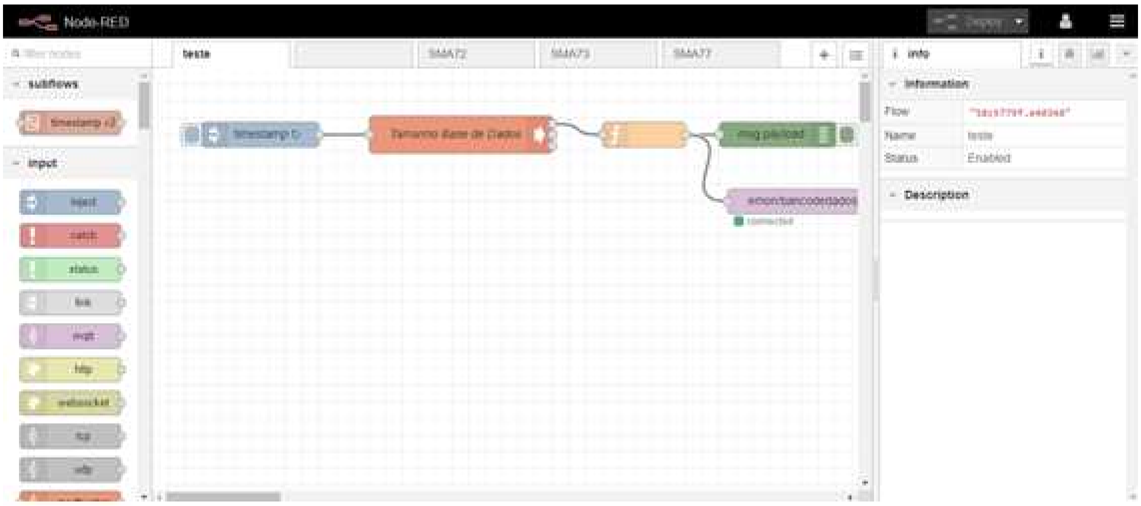
There are a variety of Web-Service tools available for the development of supervisory systems of processes equipped with smart devices. As discussed previously, the tools chosen for this project are the free Web-Services Node-Red and Emoncms due to their flexibility, popularity - since they have a mature, dependable, and stable structure at their core -, reliability, and to their friendly and robust user interface based on flow programming. Also, since they are both Web-Services, they can be used in any web browser (Google Chrome, Safari, Firefox, Chromium, Opera, etc.), providing the desired versatility and portability to the supervisory system developed in this work. Since their structure supports a lightweight runtime environment, they can easily fit on devices like the Raspberry PI, which is the chosen microcomputer for this work.

2.4.1 Node-Red

Node-Red is an open source tool developed initially by IBM for flow programming using a local host, meaning it's a browser based flow editor (SUMANGALI & KIRAN, 2016).

It's interesting to use flow-based programming (FBP) when connecting to hardware due to a number of reasons. The main reason is that FBP allows parallelism, providing performance benefits in some situations. In FBP, the dataflow is the main driving force of the program. The logical execution flow is expressed by the block diagram created: when a node receives all necessary inputs, the block produces an output that is transmitted to the next node in the path. The transmission of data through the nodes establishes the execution order of the functions. Each dataflow path runs parallel to each other, meaning that if there is a bottleneck in the code, this “hotspot” will not interfere in other processes. Also, FBP treats the applications as black-boxes: the important component for this type of programming is the connections between components that are done externally to the processes. Due to this style of programming, FBP lends itself to a plug-and-play approach, connecting all the hardware present in the plant. This type of programming allows the programmer to coordinate components into entire systems.

Node-Red consists of nodes that can be connected by wires (SICARI, RIZZARD, & COEN-PORISIN, 2018). This application is able to connect to hardware devices, such as microcontrollers and the raspberry pi amongst many others, and to the cloud environment. Node-Red runs on Java Script and it uses the Node.js framework (SUMANGALI & KIRAN, 2016). Figure 2-7 shows an example of a code in the environment.



Name teste
Status Enabled

Figure 2-7 – Example of the structure of Node-Red and its interface showcasing an example of a Node-Red code – showcasing its flow-based programming structure

The visual representation provides the users a better understanding of the interactions happening in the IoT network architecture as well as the messages being transmitted. Different kinds of hardware, e.g. sensors, and software, e.g. services, can be modeled (SICARI, RIZZARD, & COEN-PORISIN, 2018). Also, the user can program customized functions and can develop new nodes in order to fulfill the requirements of the designed application. Node-Red natively provides nodes for developing a well-structured dashboard with charts, gauges, amongst others, such as the one seen in Figure 2-8.

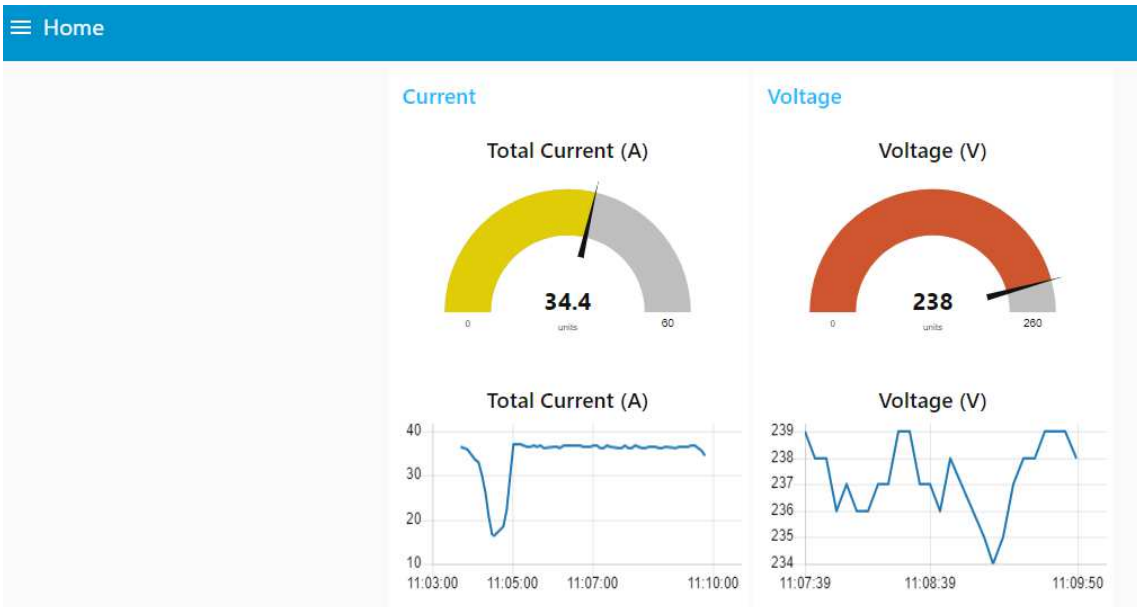


Figure 2-8 - Example of Node-Red Dashboard developed in the Node-Red platform using available nodes for creation of dashboards

Node-Red makes available simultaneously many flows acting as clouds-nodes or Raspberry PIs, Arduino platforms and so on. It enables the real connection of hardware devices and APIs (SICARI, RIZZARD, & COEN-PORISIN, 2018).

Due to the lightweight nature of Node.js and the simplicity of the Node-Red execution engine, Node-Red can be deployed with good performance on smaller computers and microcontrollers, such as the Raspberry Pi or Arduino (SICARI, RIZZARD, & COEN-PORISIN, 2018).

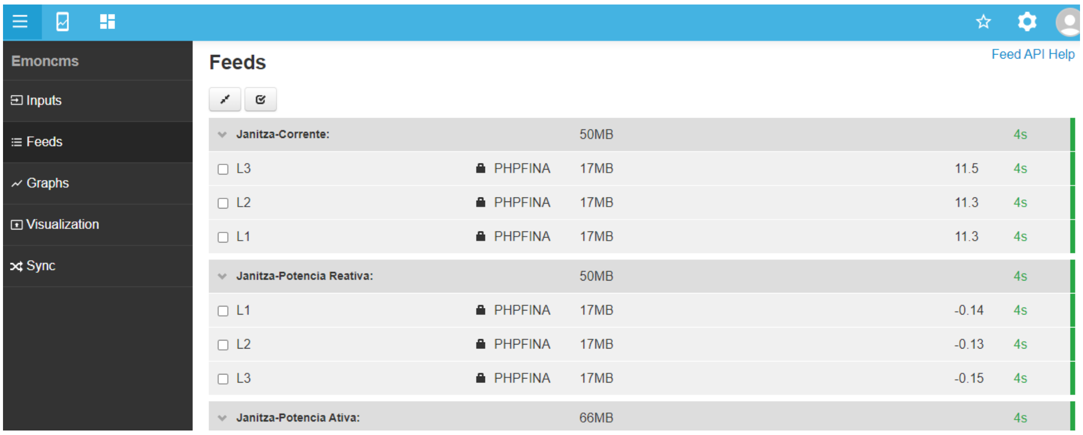
The advantages of using Node-Red when comparing it to other open-source tools, such as Eclipse Kura and Flogo, come especially from the usage of Node.js. Node-js is a lighter and agile open source tool. Node-Red with its mature framework also presents a friendlier user interface for programming. Due to the employment of JSON for its metadata, Node-Red has a more protected metadata structure by not relying on XML. Besides, due to its flow-based programming and the ability of building custom functions, Node-Red is able to fulfill the requirement of flexibility. Due to the nature of the project, a supervisory system for monitoring electrical and environmental variables, the processing rate of messages inside the code doesn't need to be extremely fast. For this reason, the speed of 1 message/second of Node-Red is sufficient and more suitable than a tool as Crosser (10 messages/second), whose performance would surpass the requirement.

2.4.2 *Emoncms*

Emoncms is a powerful open-source web-app developed initially by Open Energy Monitor for processing, logging and visualizing energy-related data. The web application provides an API key in order to allow reading and writing requests to its data. It enables the user to do the setup of the inputs, feeds, graphs, and of the dashboard (BENAMMAR, ABDAOUI, AHMAD, TOUATI, & KADRI, 2018).

Emoncms uses the Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS) and Javascript to build its user interface. The user communicates with the Application Programming Interface (API) of the server through Asynchronous Javascript and XML (AJAX), exchanging JSON messages. The API HTTP of the server activates the internal models to perform functions as data registration, processing and validation (MARIANO, 2016).

This application is divided in modules, enabling individual function development – the main modules are Feeds and Inputs, which can be seen in the Emoncms environment in Figure 2-9.



Janitza-Corrente:		50MB		4s
☐ L3	PHPFINA	17MB	11.5	4s
☐ L2	PHPFINA	17MB	11.3	4s
☐ L1	PHPFINA	17MB	11.3	4s
Janitza-Potencia Reativa:		50MB		4s
☐ L1	PHPFINA	17MB	-0.14	4s
☐ L2	PHPFINA	17MB	-0.13	4s
☐ L3	PHPFINA	17MB	-0.15	4s
Janitza-Potencia Ativa:		66MB		4s

Figure 2-9 - Example of Emoncms environment showcasing its user interface and its tabs: Inputs, Feeds, Graphs, and Visualization

The raw data that is received by the application forms the Inputs module. They need to be treated before being stored in the database in order to optimize its processing for the eventual visualization. The tool provides various functions to treat the input and convert it (if necessary) to the desired value to be stored in the database. The input can be attached to different functions that produce an output that is linked to a Feed; the Feeds module then is composed of processed raw data that is suitable for storage in the database (MARIANO, 2016). This configuration is made manually by the user in the Emoncms user interface.

The tool also allows the user to do the setup of the database when publishing new inputs and configuring them as feeds. This database is not based on MySQL; the data is stored in folders in the local server based on a NoSQL approach to store an arbitrary number of time points without an index since it's more lightweight and it's more efficient when it comes to the process involved in the application. Figure 2-10 shows an example of a dashboard developed in the Emoncms environment.

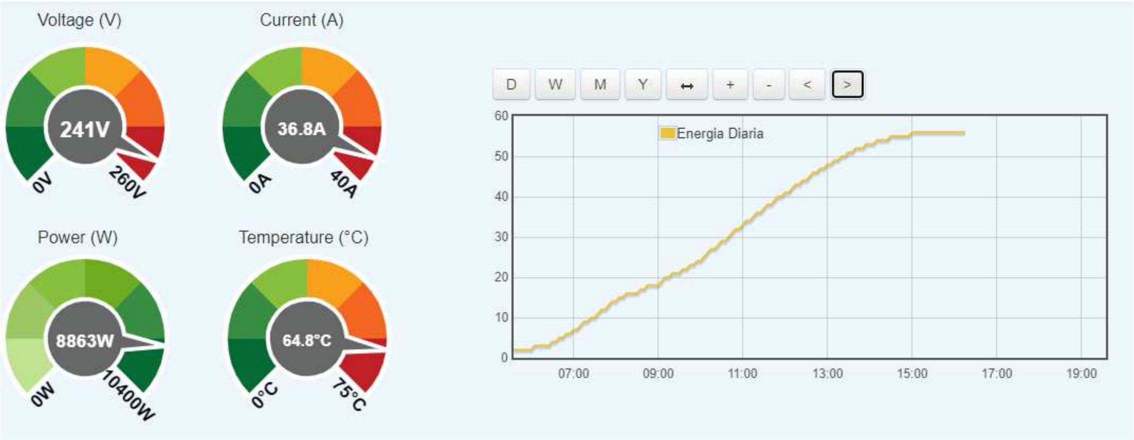


Figure 2-10 - Example of Emoncms dashboard developed in the Emoncms platform using the gadgets for creation of dashboards

2.5 Micro-grid Tesla

The micro-grid analyzed in this project is composed by a battery energy storage system (battery banks and three hybrid inverters capable of disconnecting the micro-grid from the main grid), a photovoltaic power plant (Usina Solar Fotovoltaica Tesla – USF Tesla), loads (programmable loads and feeders). The schematic for the whole system can be seen in Figure 2-11. This project focuses on the power plant USF Tesla and in the energy storage management system.

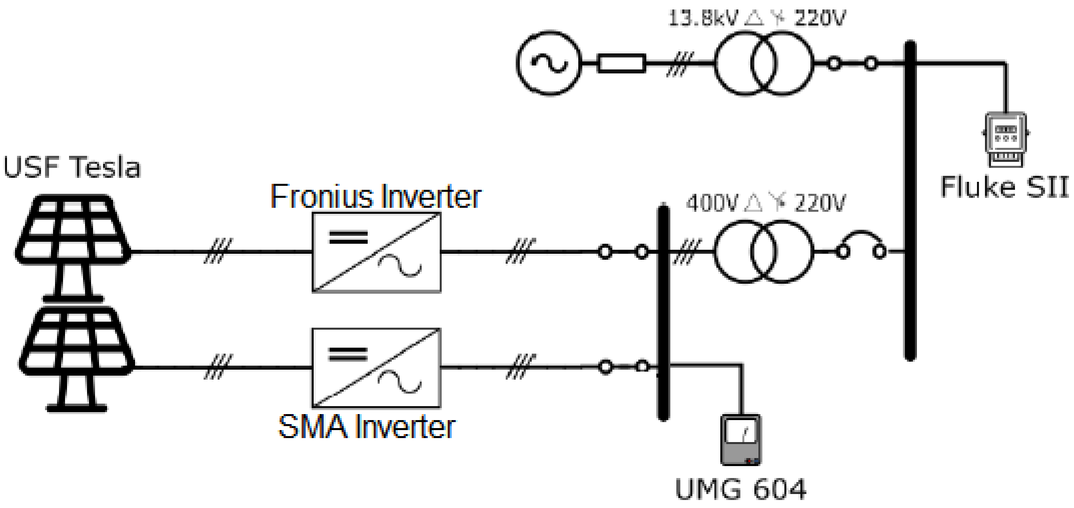


Figure 2-11 - Micro-grid Tesla single-line diagram

2.5.1 *Photovoltaic Solar Power Plant Tesla*

The power plant has been operational since July, 2016; its capacity is 37 kWp. The construction project was funded by the P&D 0047-0060/2011, 13/2011 ANEEL, project. The photovoltaic panels can be seen in Figure 2-12; there are 152 silicon polycrystalline cells, each 245Wp. And the control room can be seen in Figure 2-13.

Figure 2-12 - Photovoltaic panels installed in the photovoltaic solar power plant Tesla

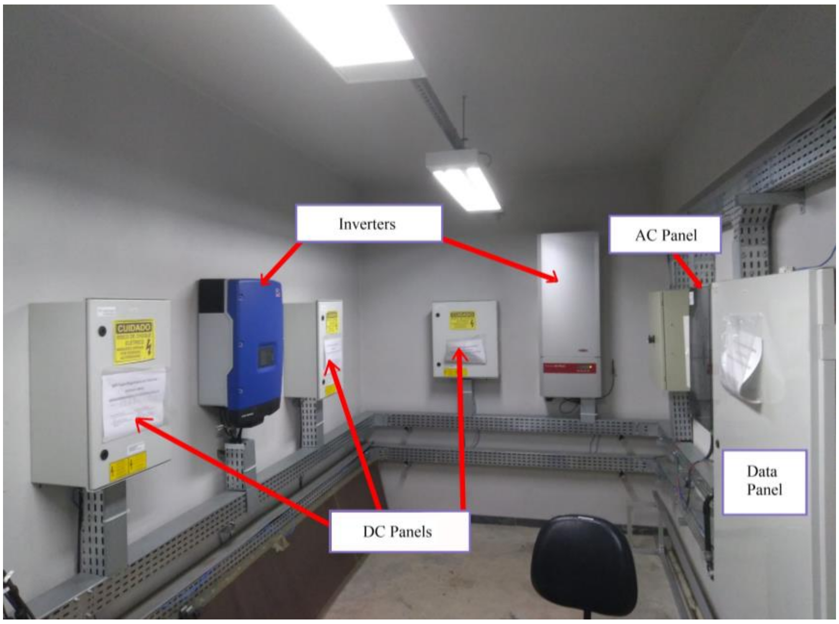


Figure 2-13 - Control room of the solar photovoltaic power plant, highlighting the inverters, DC and AC panels, and data panel

As it can be seen, there are two power inverters of two distinct manufacturers: SMA and Fronius. The main characteristics of each one are described in Table 2.1. Both of them have a WLAN and an Ethernet interface which enables the connection to a wireless/wired network; both of them are IoT ready, capable of delivering data to the network, data about its own variables or the grid's. The inverters SMA and Fronius can be seen respectively in Figure 2-14a and Figure 2-14b.

Table 2.1 - Power inverters main electrical characteristics.

Inverter	SMA Sunny Tripower 120000TL	Fronius IG Plus 150V-3
Peak Power (kWp)	12	12
MPP Voltage Range (V)	440-800	230-500
Max. Intput Voltage (Vdc)	1000	600
Max. Input Current (A)	55.6	36
Number of MPP	2	1
Output Voltage (Vac)	400	400

Number of Modules/String	15	10
Number of Strings in Parallel	3	5
Installed Peak Power (kWp)	11.025	12.25

a) b)

Figure 2-14 – a) SMA Sunny Tripower 12000TL b) Fronius IG Plus 150V-3

There are sensors throughout the power plant: energy meter Janitza UMG604 seen in Figure 2-15, current sensors, pyranometer, thermometer, barometer, humidity sensor and anemometer. The sensors make up the solar meter station and the measurement details can be seen in Table 2.2.

Figure 2-15 - Energy Meter Janitza UMG604

Table 2.2 – Characteristics of the measurements of the solar meter station

Measurement	Measuring Range	Accuracy	Sampling Rate
Air temperature	-50°C ... +60°C	+/-0.2°C	1 minute
Humidity	0 ... 100% RH	+/-2% RH	1 minute
Air Pressure	300 ... 1200hPa	+/-1.5hPa	1 minute
Wind Speed	0 … 60m/s	+/-0.3m/s	10 seconds
Wind Direction	0 … 359.9°	< 3°	10 seconds
Global Radiation	0.0 … 1400.0 W/m²	< 1 W/m²	1 minute

2.5.2 *Energy Storage System*

This system is not connected to the solar power plant yet; it’s still in its testing stages. This system will enable power dispatch in the connection point between the micro-grid and the main grid. The batteries are also vital for the operation in island mode, when disconnecting the micro-grid from the main grid. The system is composed by three independent single-phase hybrid inverters and three battery banks – each one a different technology.

The first bank is made by 20 lead-carbon batteries seen in Figure 2-16a distributed in 5 blocks in parallel, each block containing 4 batteries in series. The second one is the molten-salt battery bank seen in Figure 2-16b, with 2 batteries in parallel. And, lastly, the lithium-ion bank battery seen in Figure 2-16c has 6 batteries in parallel. The main characteristics of each technology are presented in Table 2.3.

Table 2.3 - Batteries in the Energy Storage System.

Battery	Lead-Carbon: Narada	Molten-Salt: FZSoNick	Lithium-Ion: Unipower

	REXC70	48TL200	UPLFP48-100
Nominal Voltage (V)	12	48	48
Capacity (Ah)	70	200	100

Figure 2-16 – a) Battery Narada REXC70, b) Battery FZSoNick 48TL200, c) Battery Unipower UPLFP48-100

Each battery bank is connected to the correspondent hybrid inverter, enabling the monitoring and control of each electrical variable through the battery management system.

The hybrid inverters are 3 single-phase independent SMA Sunny Island 8.0h inverters seen in Figure 2-17. They have a WLAN and an Ethernet interface which enable them to connect to a wireless/wired network. The hybrid inverters are connected to the battery banks and are capable of enabling dataflow from/to the battery management system to/from the control room via a local network or via the Internet. They make any variable available for the user in this network: their own, the ones of the main grid and the variables related to the connected battery bank. The hybrid inverters enable the possibility of data monitoring and power flow control. The variables measured by the DC-side related to the batteries and to the AC-side related to the grid are important to the development of a supervisory system. These inverters also allow the sending of commands to control the charge and discharge of the battery banks in order to adjust the power flow.

Figure 2-17 - SMA Sunny Island 8.0h

3 Supervisory System

In this chapter, the methodology for developing the supervisory system is detailed. The supervisory system is developed through the usage of Raspberry Pi 3b+ microcomputers seen in Figure 3-1. There are 3 servers that constitute the supervisory system: communication server, application and database server and the backup server. The communication server consists of data acquisition and processing. The data acquisition and its processing, e.g. converting the raw data to engineering unit, are performed in the Node-Red environment. Then, the processed data can either be sent to the Emoncms environment in the application server in order to build the supervisory dashboard (a schematic of the power plant that shows the real-time measurements), or be used in the Node-Red dashboard environment. The Node-Red dashboard can be built to monitor variables in graphs or it can be used to send commands to the field devices, such as sending commands to control the active and reactive power flow. The data is stored in the application and database server through the configuration of the input data received by Emoncms.

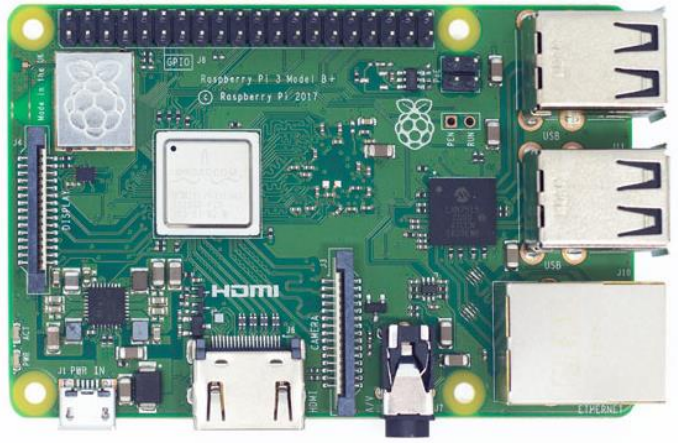


Figure 3-1 – Hardware: Microcomputer Raspberry Pi 3 b+

There are two distinct networks that can communicate with each other: the solar power plant with the power inverters and the environment measurements, and the energy storage system with the hybrid inverters and the battery banks.

3.1 Data Acquisition and Processing

3.1.1 Solar Power Plant

The solar power plant has two power inverters and an energy meter that are smart devices: the SMA Sunny Tripower 12000TL, the Fronius IG Plus 150V-3 and the Janitza UMG604 energy meter; they can connect to the TCP/IP. The inverters are capable of measuring electrical data from the DC-bus (the solar panel side) and from the AC-bus (the main grid side) as the energy meter. This data is available via communications protocol for data processing.

The communication protocol used in this dissertation for the communication between the communication server and the smart devices (the inverters and the meter) is the Modbus TCP/IP since they are already configured for this protocol; the meter and each inverter can be configured for a specific fixed IP determined by the user and has its own Modbus map.

The Modbus protocol is a highly used flexible open message structure used for the communication between master (client) and slave (server). The communication is only initiated by the master; the master is responsible for initiating the communication between devices by sending a request to the slave for it to send its data as seen in Figure 3-2. The master may request/receive data to/from the slave at any moment since the communication is not initiated by the slave. The slave receives the request and responds by sending the required data; the sent data may be binary or numeric. Figure 3-3 shows how the message is built in the master and in the slave and how the Modbus message exchange happens; it's possible to see that this protocol requires a polling structure (FREITAS, 2014).

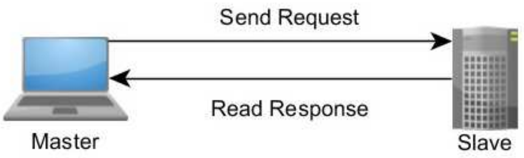


Figure 3-2 – Illustration of the communication between master and slave in the Modbus protocol (NI, 2019a)

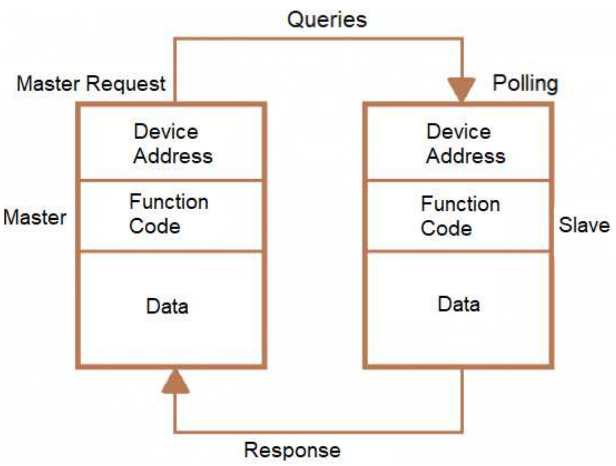


Figure 3-3 – Schematic of the Modbus Message Exchange. Modified from (FREITAS, 2014)

The device address corresponds to the IP address of the device that can be configured by the user. The function code refers to the type of request the master is performing: read/write analog/digital registers. The TCP port 502 is the standard port for the connection with Modbus TCP servers (NI, 2019a). The Ethernet standard (TCP/IP) in the Modbus protocol may have some variations when it comes to the available speed: it can reach 100Mbps or even 10Gbps (NI, 2019b).

Modbus can also use the serial RS-485 standard to establish communication between devices. This standard allows communication rates that may reach 12 Mbps or 50 Mbps (NI, 2019b). Ethernet is faster and easier to troubleshoot; however, sensors, transmitters don’t need to report data very fast and Modbus RS-485 may be less expensive since it doesn’t need a switch when having various Modbus servers in the network (RINALDI, 2013). A generic Modbus RS-485 connection can be seen in Figure 3-4. The physical connection is done with a twisted wire pair plus ground (GND).

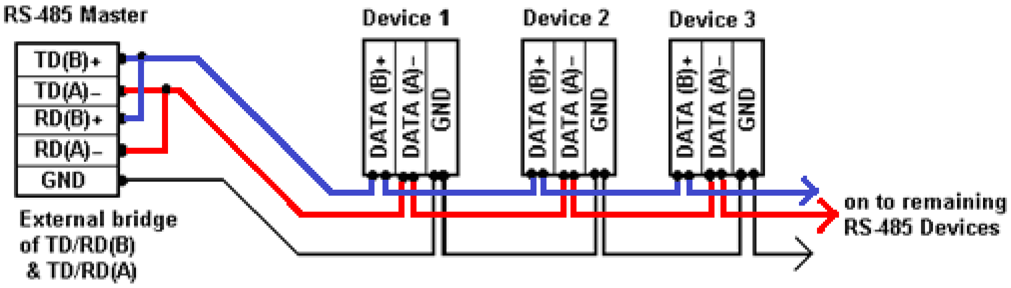


Figure 3-4 - An example of a wired-connection between devices using the RS-485 Modbus protocol. Modified from (BB Smart Worx, 2020)

The Modbus protocol doesn't need an intermediary (such as a broker) since the connection is done point-to-point. The way this communication is performed guarantees that the final recipient receives the message: the master knows the request has reached the slave and the slave knows the master has received the response. However, since the protocol requires a request-response structure, the network may be sub utilized or the application code may become extremely complex and the consumption of network resources is high. One important disadvantage of Modbus is that it doesn't incorporate security resources (SILVA, 2019).

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

The raw data obtained for each device has to be processed in order to obtain the data in engineering units, meaning that each data has to be converted to a float value that represents the real value of the electrical variable. For this reason, some conversion functions are developed in Java Script using the Node-Red environment as seen in Figure 3-5. In Figure 3-5, it's also possible to see that the processed data is then sent to the Node-Red Dashboard as a graph and to the Emoncms environment to be a part of the supervisory dashboard.

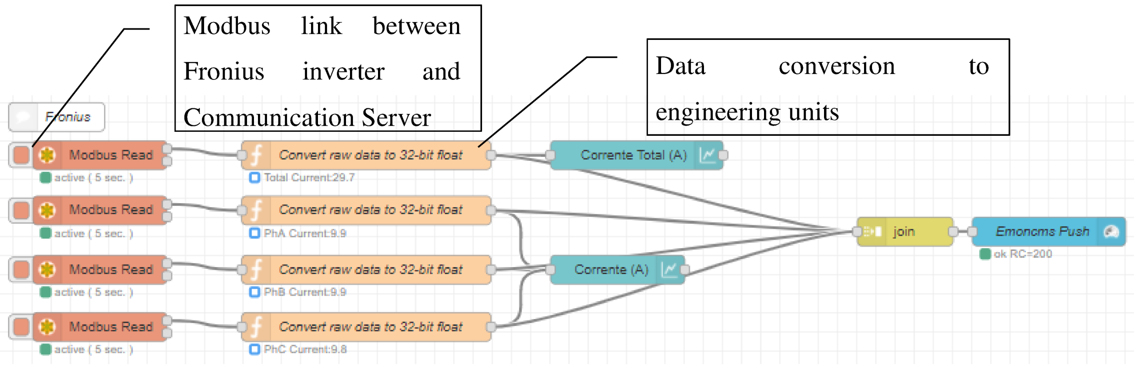


Figure 3-5 – An example of the acquisition and processing of data in the Node-Red environment

Besides the meter and the power inverters, there are gauges that are responsible for measuring the relative and absolute (calculated from relative humidity and air temperature) humidity, the relative (referenced to sea level, calculated using the barometric formula with the aid of local altitude) and absolute pressure, the ambient temperature, the wind direction and the global irradiance – the solar meter station. There are also gauges to measure the temperature and the radiance of a reference cell.

These gauges are not smart devices; however, they are currently being monitored by a data-logging device provided by the third party company that is currently supplying the supervisory system. This device communicates with the gauges via Modbus RS485. Therefore, it's possible to retrieve this data from the third party company device and upload it to the application server through the file transfer protocol (FTP).

FTP is a secure connection between devices that allows file exchange as seen in Figure 3-6. The file transfer protocol serves as a bridge between devices connected to the same network or connected to the Internet. FTP also relies in a client-server structure; the client requests access to a certain file and the server grants this access. After the file transfer is completed, the communication is automatically terminated. This protocol is a simple tool that allows high volume of data transfer through a network and it allows various directories to be transferred at the same instant (L, 2020).

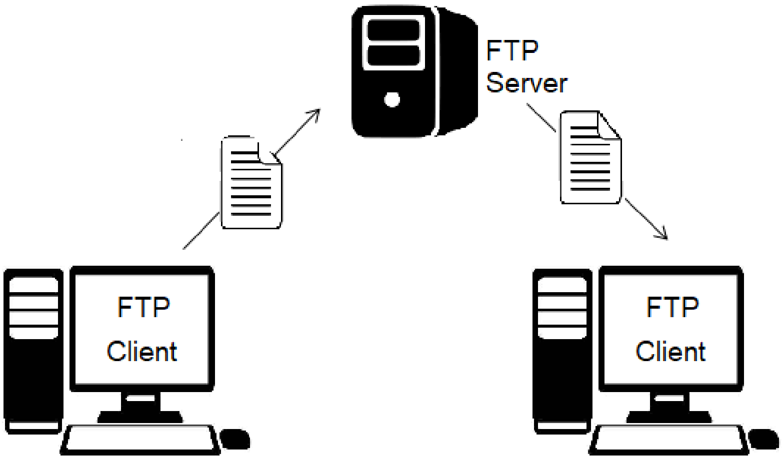


Figure 3-6 – Illustration of the process related to the File Transfer Protocol. Modified from (L, 2020)

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

In order to establish a link between the communication server and the application server to post the data from the solar meter station to the application, it's necessary to integrate another communication protocol in the Python service. The protocol selected is the MQ Telemetry Transport (MQTT) protocol. MQTT is a communication protocol based on a publish-subscribe architecture. In this type of architecture, there are devices that generate and publish data (publishers) and other devices that consume this data (subscribers). This architecture demands an extra entity that acts as a centralizer in the data exchange, the Message Broker. The devices that generate data publish it to the broker which organizes the data in topics. The devices that consume the data subscribe to the desired topic and are able to retrieve the data assigned to the topic. Every instant new information reaches the topic in the broker, this information is automatically sent to the device that subscribed to the topic. It's important to note that the publishers and subscribers never interact directly; the link is done via the broker (SILVA, 2019). The process is summarized in the scheme seen in Figure 3-7.

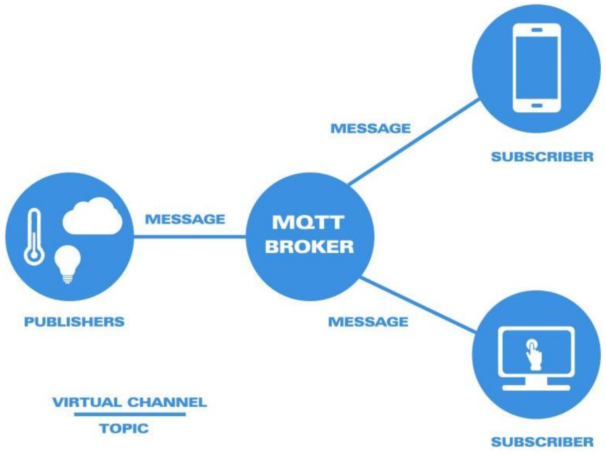


Figure 3-7 – Illustration of the communication performed in the MQTT protocol (SILVA, 2019)

There is no need for the subscribers to know the address of the publishers; they only have to connect to the broker and subscribe to the desired topics to receive the data. The publisher doesn't have to wait for a request from any device to publish its data; the publisher determines the best moment to send its data to the broker, enabling a more efficient energy management. This protocol has implemented security layers; it requires login and password for broker access and for message exchange. And, even though it can implement complex security measures, it’s a lightweight protocol (SILVA, 2019).

In addition to the aforementioned advantages, the installation of Emoncms also installs a lightweight MQTT broker (the open source message broker Mosquitto). Therefore, the MQTT protocol is a well-suited option for this project.

By using the MQTT protocol in the Python service, it’s possible to publish the data retrieved from the files obtained by FTP to topics to which Emoncms (and, therefore, the application server) is already subscribed to. The environmental data is sent directly as an input to the Emoncms environment in the application server; it doesn’t need to be processed since it’s already in engineering units, so it isn’t sent to Node-Red.

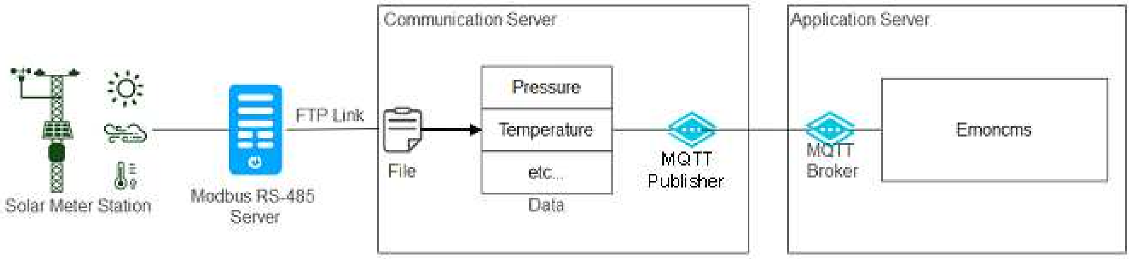


Figure 3-8 - Schematic of the communication established between the solar meter station and the application server, showcasing the FTP and MQTT links

In order to send the measurements of the inverters from the communication server to the application server, the protocol MQTT is also used. However, to send the data from the energy meter to the application server, another communication protocol is used: HTTP. The HTTP (Hypertext Transfer Protocol) is a protocol that enables hypermedia documents transfer (MDN Web Docs, 2020). It was developed in order to enable communication between Web browsers and Web servers. It is based on a client-server structure seen in Figure 3-9; the client initiates the link by making a request and waits until the server provides a response. After the request is responded by the server, the connection between the client and the server is severed. A new connection must be done for every new request (DE SOUZA, 2019).

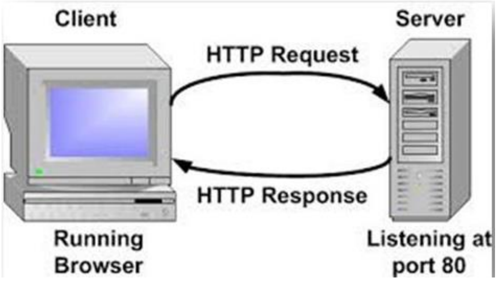


Figure 3-9 – Illustration of the process related to the HTTP Client-Server Communication. Modified from (DE SOUZA, 2019)

Node-Red has an Emoncms node that uses this protocol to establish a path between Node-Red and Emoncms for the exchange of data. This node provides resources to post data to Emoncms via PUSH transaction. This means that it’s possible to initiate the request to publish data to the client with no initial request from the receiver – in this case, Emoncms (MDN Web Docs, 2020). This is possible by using the API key available for reading/writing found in the Emoncms administrator page.

There are also current sensors that measure the DC input current seen in Figure 3-10. These current sensors aren’t smart devices as well. In order to make these measurements available for the network, the output of each current sensor (there are three) is connected as an analog input to an Arduino Nano with an Ethernet Shield (ENC28J60) seen in Figure 3-11.

Figure 3-10 - DC-Current Sensor used in the power plant for the measurement of the DC input current in the power inverters (3-K, 2020)

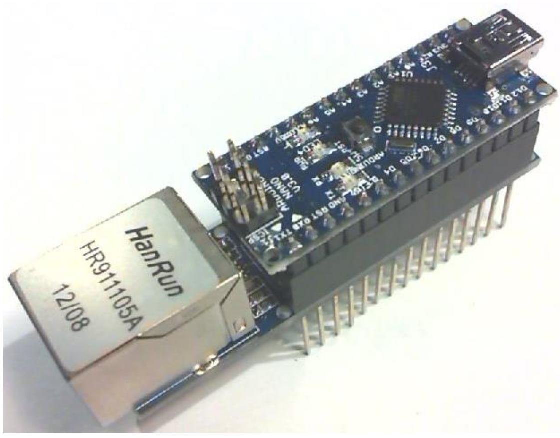


Figure 3-11 – Microcontroller with the Arduino Nano framework with Ethernet Shield

Arduino is an embedded open source platform in which various controllers may be used with a microcontroller Atmel AVR that is capable of interacting with the environment via hardware and software (AQEEL, 2018). The device is built with input and outputs and the programming language is based on C/C++. Arduino boards are used in robotics, embedded systems, and electronic projects in which automation is necessary to the process. The Arduino Nano is a small, compatible, flexible microcontroller and ideal to smaller projects due to its reduced size (SoldaFria, 2020). Besides this, the microcontroller selected has more than one analog input – there are three current sensors – unlike another viable option that would be the ESP8266. Other advantages of the selected microcontroller are its price, protection, and resilience that are considerably more suitable for the application than another option such as the ESP32 that has a higher cost, a slew of protection problems, and it's more fragile. However, the selected microcontroller has a limited RAM memory (32kb) and it doesn't have embedded resources as WiFi or Bluetooth; therefore, it's necessary to connect a shield to it in order to incorporate it in IoT projects (AQEEL, 2018). So, the Ethernet Shield is used to provide a network connection to the Arduino Nano. A fixed IP for the Arduino Nano is determined via an Ethernet library – possible due to the Ethernet Shield.

Each analog input of the Arduino that is connected to the DC-current sensors has its value read in the Arduino. These values need to be converted to engineering units since they are presented in bits as digital outputs. After they are processed, they're transmitted from the communication server to the application server via MQTT. Then, they are stored in the database and are available for the development of the monitoring screens of the supervisory system. The schematic shown in Figure 3-12 summarizes this process.

Figure 3-12 - Data acquisition for the DC-Current sensors using a microcontroller – developing a smart sensor

3.1.2 *Energy Storage System*

There are three hybrid smart inverters, SMA Sunny Island 8.0h, in this network that are connected to the battery banks detailed in Chapter 2 of this dissertation. These inverters also communicate with a communication server via Modbus TCP/IP.

These inverters not only provide data but also receive data – commands – via Modbus TCP/IP in order to enable the user to control the reactive and active power flow of the grid by managing the charge and discharge of the battery banks. The inverters are capable of providing data regarding the measurements of the grid (AC-side) and data regarding the measurement of whichever battery bank is connected to it (DC-side); it’s important to note that each inverter is connected to a phase conductor and they operate separately and don’t communicate amongst themselves. As with the power inverters in the solar power plant, the data sent by these inverters need to be further processed in order to convert the measurements to engineering units. After this processing, the data is sent to the application server, to be transmitted and/or to the Emoncms dashboard or the Node-Red dashboard as seen in Figure 3-5.

3.1.2.1 *Secure Protocol*

In order to send the data to the Emoncms in the application server, the MQTT protocol is also used. However, since the networks are different, it’s necessary to establish this link via the Internet. Therefore, it’s preferable to use the secure protocol: MQTTS. The MQTTS is the implementation of the MQTT protocol over an additional security layer that uses the SSL/TLS protocol, using encryption to protect the transfer of data and information. The SSL/TLS protocol secures communications between a server and a client regardless of the chosen communication protocol, meaning it can be applied to HTTP or to MQTT or to FTP, amongst others.

The secure communication starts with a TLS handshake: the server and the client exchange messages to acknowledge and verify each other, they establish the used encryption algorithms and agree on session keys that are used to encrypt and decrypt all communications in this session after the TLS handshake (for each new session, a new set of random keys is generated). Two keys are used during the session: a public key (client side) and a private key (server side). An example of the exchange between client and server can be seen in Figure 3-13.

Figure 3-13 - Example of a TLS link with 2 encryption keys in the SSL/TLS secure protocol communication (CLOUDFARE, 2020)

The identity verification happens via the authentication of the SSL certificates that have to be stored in the server and can be stored in the client (if the communication requires client certificates), ensuring each party of the link is who they claim to be. The SSL certificate is a file containing the public key and other information; without the certificate, the communication cannot be encrypted by TLS. TLS also ensures that the data from the server has not been altered, since a message authentication code accompanies the data during the exchange, preventing on-path attacks.

To use this protocol, it’s necessary to generate certificates for the server (in this case, the communication server that sends the data to the Emoncms) and for the client (in this case, the application server that houses the Emoncms). These two sets of certificates must be applied to the configuration in the server side and also in the client side so that the link between them via MQTTS works properly.

Since the Emoncms environment doesn’t allow communication via the MQTTS protocol, the link between the server in the energy storage system and the solar power plant system is done using the Node-Red environment at both ends. The data from the energy storage system is transmitted to the Node-Red in the communication server of the solar power plant and then the data is transmitted to Emoncms via MQTT.

3.2 Data Monitoring, Storage and Control

3.2.1 Node-Red

As detailed in the previous section, the processed data can be sent to the Node-Red Dashboard to be monitored in a graphic format. The processed data from the communication server is sent to a dashboard node that presents the real-time information in a graphic format.

Node-Red Dashboard is also used to build a supervisory system to send commands to the energy storage system by allowing the user to configure the active and reactive power flow in the grid. The dashboard is assembled in a way that allows the user to enter with the desired value for 4 parameters for the active and reactive power flow control. These values are then transmitted to the hybrid inverter that is responsible for managing this information by charging or discharging the battery bank connected to it.

3.2.2 Emoncms Dashboard

As detailed in the previous section, the processed data can be sent to the Emoncms Dashboard to be monitored. As detailed in Chapter 2, the Emoncms environment gives the possibility to create custom dashboards in order to provide flexibility regarding the type of dashboard to be generated.

For this research, the idea is to generate a dashboard in Emoncms that represents the single-line diagram of the system, including the photovoltaic solar power plant and the energy storage system. Also, this dashboard has to be able to provide further information about the electrical variables in a graphic format if the user wants to analyze real-time and historical data.

The communication server is responsible for sending all the necessary data to the application server with Emoncms. The data is received by the Emoncms environment as inputs. In order for these inputs to be available for usage for the supervisory system, the user has to configure them as feeds to the database. The assembly of the database and its manipulation is performed in the application server via Emoncms, since it has a database tool embedded into it that doesn’t rely on MySQL. The problem with MySQL in this application would be that, with the increase of stored data, the process for retrieving historical data would be significantly slowed. So, the solution implemented by Emoncms developers was to create database engines that don’t rely on the creation or manipulation of tables on MySQL, but generate files that can be easily perused by the software when a historical request is made.

The data is recorded at a fixed interval determined by the user when configuring the feeds. Emoncms provides different processes that can be applied to the input in order to produce the feed, such as: logging, converting power to energy, performing mathematical or logical operations with one or more inputs.

In Figure 3-14, some inputs can be seen; in this case, all the inputs are only logged to the database. In Table 3.1, a sample of the feeds can be seen; the whole list of configured feeds can be seen in Appendix A.

Fronius - Frequencia: 2s
☐ 1 log 2s 60

Fronius - Potencia Reativa: 3s
☐ 1 log 3s 0.82

Fronius-CC: 2s
☐ 1 log 2s 4.47
☐ 2 log 2s 17.2
☐ 3 log 2s 259

Fronius-Corrente: 4s
☐ 1 log 4s 6.2
☐ 2 log 4s 18.1

Figure 3-14 – Sample of inputs in the Emoncms environment – showcasing electrical measured values from the Fronius inverter

Table 3.1 – Sample of feeds configured in the Emoncms environment

Variable	Feed	Database	Sample Time

Total	Fronius-Corrente	PHPFINA	10s
FaseA			
FaseB			
FaseC			
FaseA	Fronius-Tensao		
FaseB			
FaseC			
Corrente	Fronius-CC		
Tensao			
Potencia			
Frequencia	Fronius-Medicoes		
Potencia Reativa			
Potencia Ativa			
Energia Ativa			
Geracao_Fronius	-		
Geracao_CA_Fronius			

Note: Highlighted variables indicate virtual feeds – not measured values.

With all the necessary feeds already configured, it’s possible to start the assembly of the supervisory system in the Emoncms dashboard configuration environment. Each new dashboard is a new screen; the environment for the building of the screen is seen in Figure 3-15.

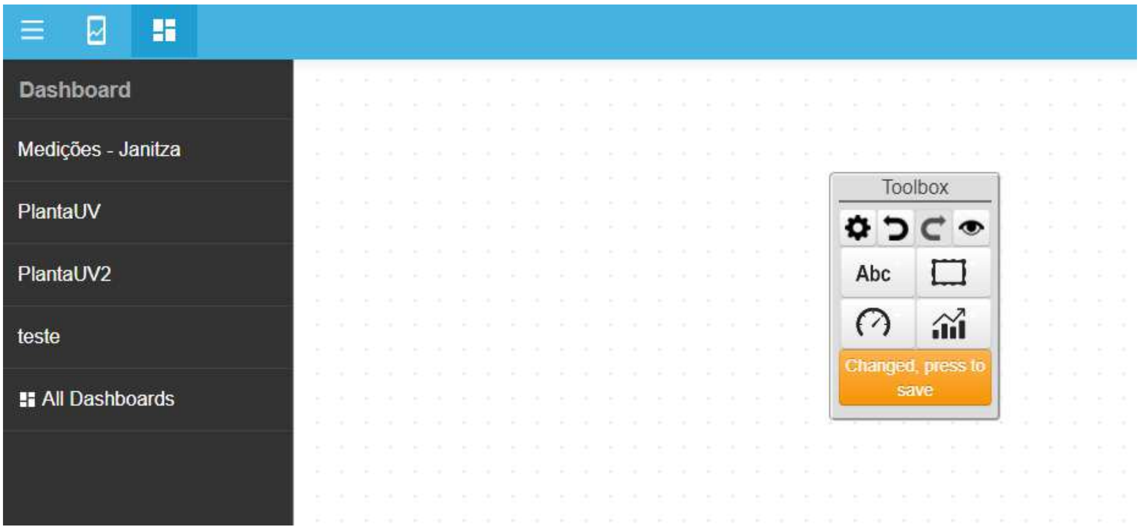


Figure 3-15 – Emoncms Dashboard assembly screen, showcasing the toolbox with the gadgets to develop the desired screens

The Toolbox provides a vast variety of gadgets that can be used in order to create the desired screen. Each gadget is used in conjunction with one or more of the configured feeds. In this work, the gadgets mostly used are graphs – real time and historical – and LEDs that represent the status of the energy transmission and status of charge of the battery banks. There is also the possibility to program HTML functions in order to further customize the dashboard. In this research, the single-line diagram is built using this feature in order to upload to the Emoncms Dashboard an image that represents the diagram.

3.3 Final Architecture

In order to present the whole structure - from the data acquisition until the data display via the developed screens -, Figure 3-16 is shown.

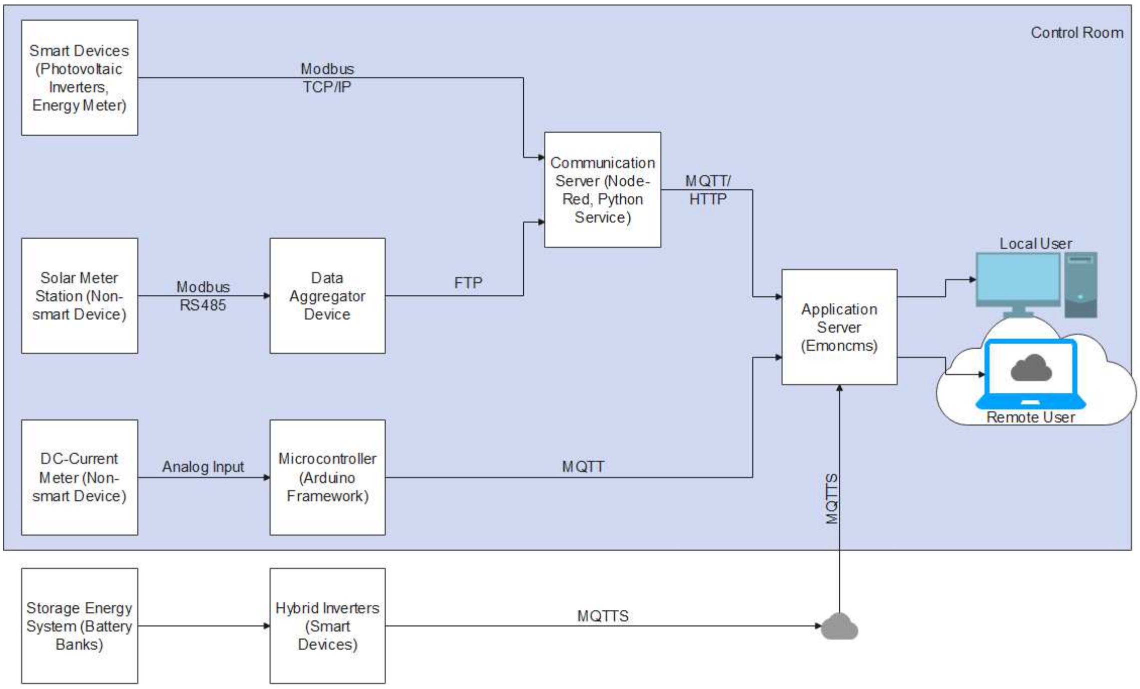


Figure 3-16 - Final architecture of the data acquisition and data monitoring of the developed supervisory system, showcasing the communication protocols

The schematic shown in Figure 3-16 summarizes the connections made beginning with the field devices and ending in user terminals in order to acquire, store, and monitor the data from the solar power plant and from the energy storage system. All the communication protocols implemented and detailed in the present chapter are also seen in Figure 3-16.

4 Results

In this chapter, the complete supervisory system with its screens in Emoncms and in Node-Red are presented. The screens were developed as detailed in the previous chapter and the interactions between the supervisory system and the user are also described in this chapter. The screens developed are compared to the current supervisory system in order to better understand the benefits of the new developed system.

4.1 Solar Power Plant

The dashboard is primarily developed in the Emoncms platform since it provides a user-friendly interface to build a custom-made supervisory system, providing some built-in gadgets and allowing customization. Besides, due to the features that Emoncms offers to its users related to its inputs and database storage, the platform is extremely adaptable to fulfill any requirements and it’s easily expandable – to insert new variables, devices and/or features.

This versatility of Emoncms is an important benefit of the platform specially when comparing it to the current supervisory system available in the solar power plant. For the latter, it’s cumbersome to add any new features and/or variables to it, since the system is developed by a third party company and the source is not available. Besides, the available structure of the current supervisory system is more rigid while Emoncms provides an environment that allows for any type of required dashboard.

The current supervisory system can be seen in Figure 4-1 and Figure 4-2. As it can be seen, this supervisory system has two tabs for the user. The main page seen in Figure 4-1, Cockpit, depicts an overview of the current main electrical and environmental measurements of the solar power plant. The Evaluation page seen in Figure 4-2 provides the user with a graphical visualization of the available variables for a customizable period of time.

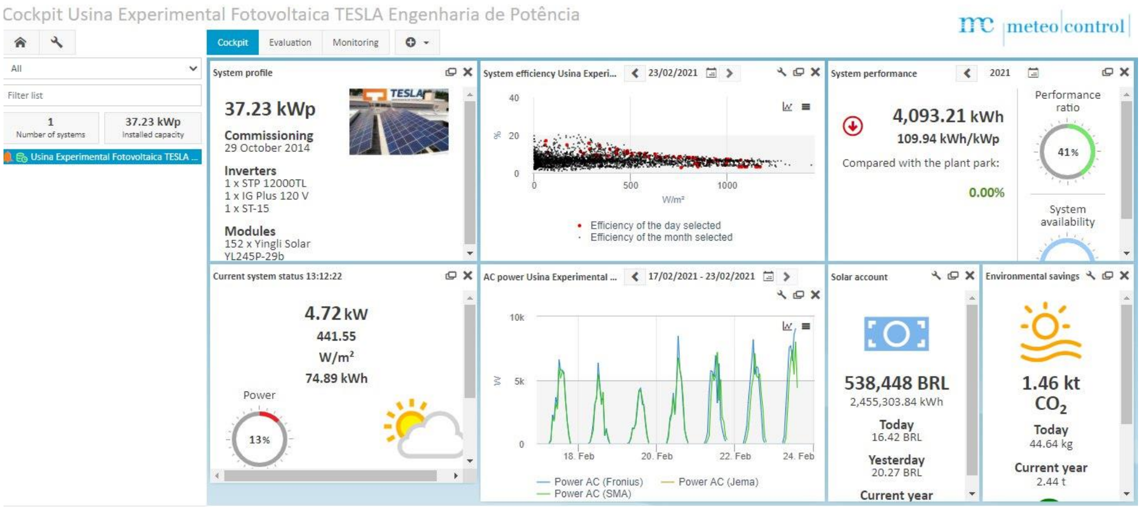


Figure 4-1 - Main page of the current supervisory system

	Daily values Unit	Maximum value Unit
Sum of global radiation	4.31 kWh/m²	1,172.40 W/m²

Energy generation

Displayed will be the energy yields (monthly/annual/total) and, respectively, the performance data (daily/weekly) based on the feed-in meter values (E_Z_EVU). Manual meter reading changes will be considered and displayed in the monthly, annual and total views. If the yields per day or per month are too high, they will be highlighted in colour. This may be the case, in particular, if meter readings are changed manually, in order to fill in gaps over large periods. For the daily and weekly views the E_Z_EVU value will either be converted into a power value or the AC power values will be summed up. In this case, the following values will be considered: AC power in

Figure 4-2 - Evaluation page of the current supervisory system

The current supervisory system doesn't allow for the creation of an overview of the schematic of the plant, which would be important for the user specially when tracking the availability of the plant throughout the day. This overview would also be interesting to understand the structure of the power plant and to easily detect any potential problems and/or failures.

The new supervisory system developed in Emoncms allows the inclusion of this overview besides allowing the creation of any necessary graphics and visual gadgets required to fully inform the user of the current status and operation of the power plant. The screens created in the dashboard can be seen in Figure 4-3, Figure 4-4 and Figure 4-5.

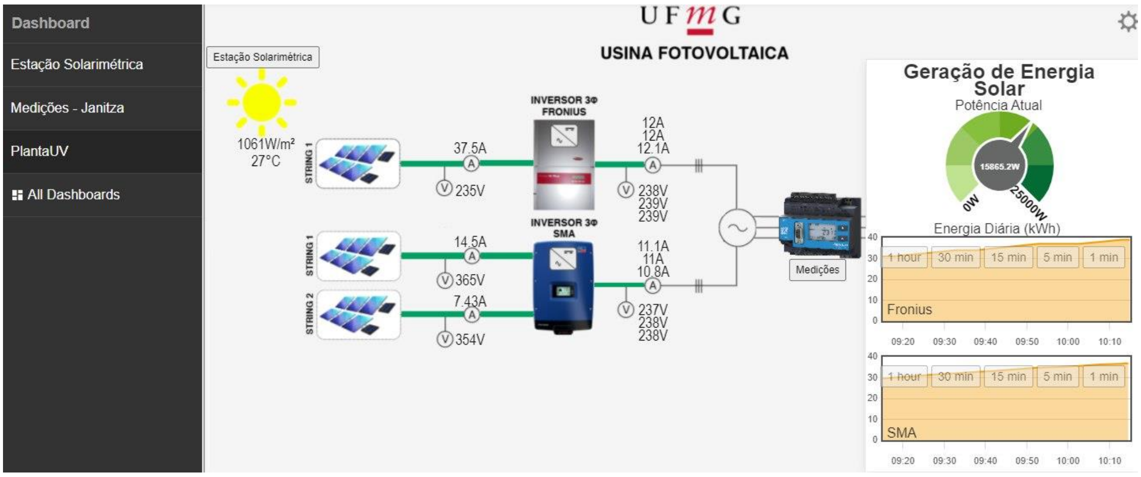
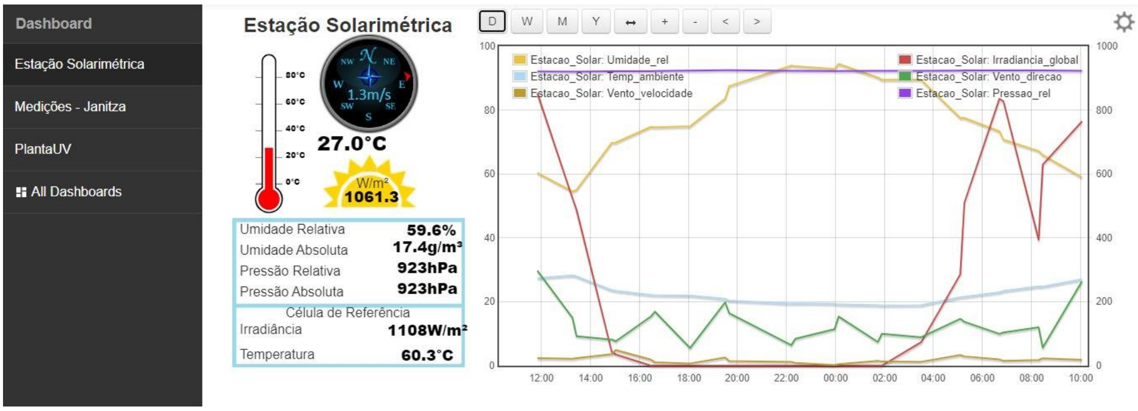


Figure 4-3 - Main page of the new supervisory system developed in Emoncms



Umidade Relativa	59.6%
Umidade Absoluta	17.4g/m³
Pressão Relativa	923hPa
Pressão Absoluta	923hPa
Célula de Referência	
Irradiância	1108W/m²
Temperatura	60.3°C

Figure 4-4 - Solar Meter Station page of the new supervisory system developed in Emoncms

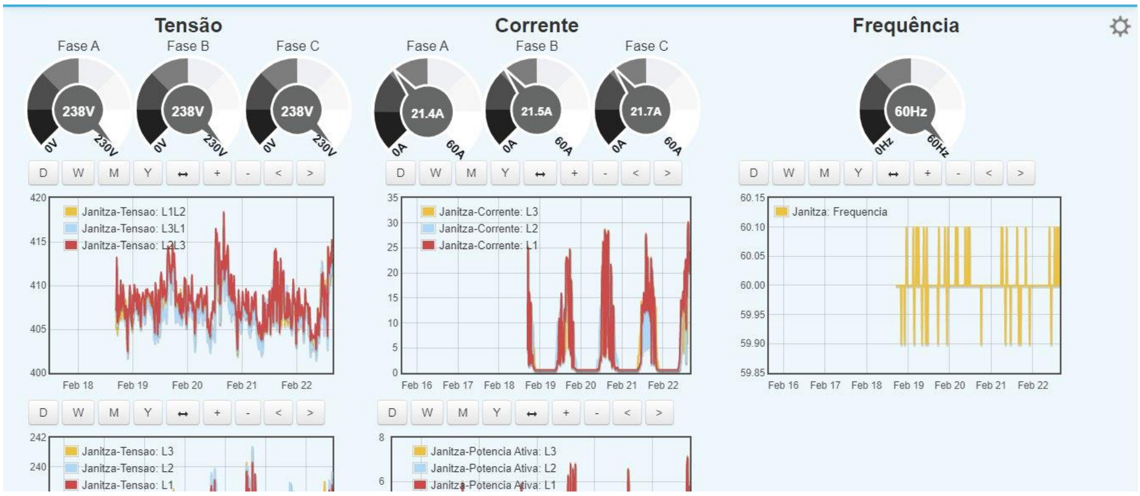
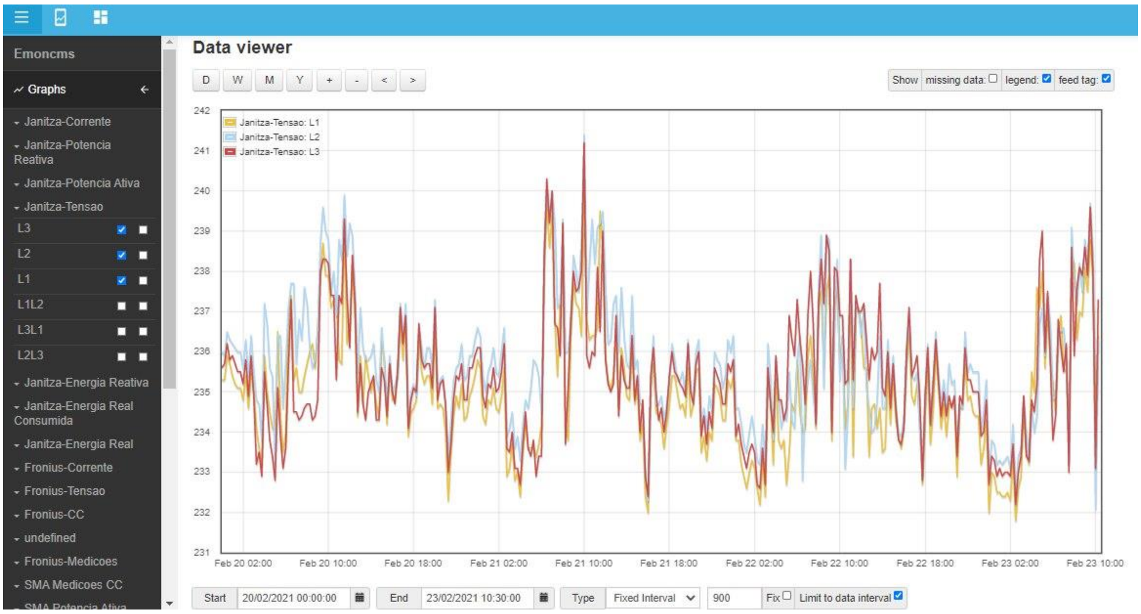


Figure 4-5 - Grid measurements page of the new supervisory system developed in Emoncms

The main screen seen in Figure 4-3 shows the some electrical variables: DC/AC voltage, current of each inverter and the AC voltage/current in the AC side being measured by each inverter. In this overview, it’s also interesting to note that it’s possible to detect whether the solar power plant is generating power by following the connections to and from the inverters. If the connection is active, it's shown as green. If it's not, it's shown as red. Also, the solar radiance and the ambient temperature can be seen. In this screen, there are also buttons that redirect the user to detailed data of the energy meter seen in Figure 4-5 and of the solar meter station seen in Figure 4-4. The user can monitor all the UMG604 variables and all the data from the solar meter station. The detailed screens are built with graphs. These graphs are also capable of displaying historical information – if the user wants to see historical information, it's only needed to set the desired dates and the graphic will be updated to show the desired time frame.

Any variable that is available in the database accessed by the Emoncms environment can be added to the dashboard easily graphically or via a gauge. So, if there's a requirement to include any variable, the environment is readily able to fulfill this requirement.

Besides the dashboard, the Emoncms offers a Graphs tab that allows the user to plot any variable available in the database in a customizable period of time. This tab can be seen in Figure 4-6. In comparison to the Evaluation tab available in the current supervisory system seen in Figure 4-2, the Emoncms Graphs tab integrates new data more easily and effectively since it doesn't require any other configuration as long as the data is stored in the database. If the data is stored in the database, the data is automatically available in the Graphs tab and ready to be plotted.

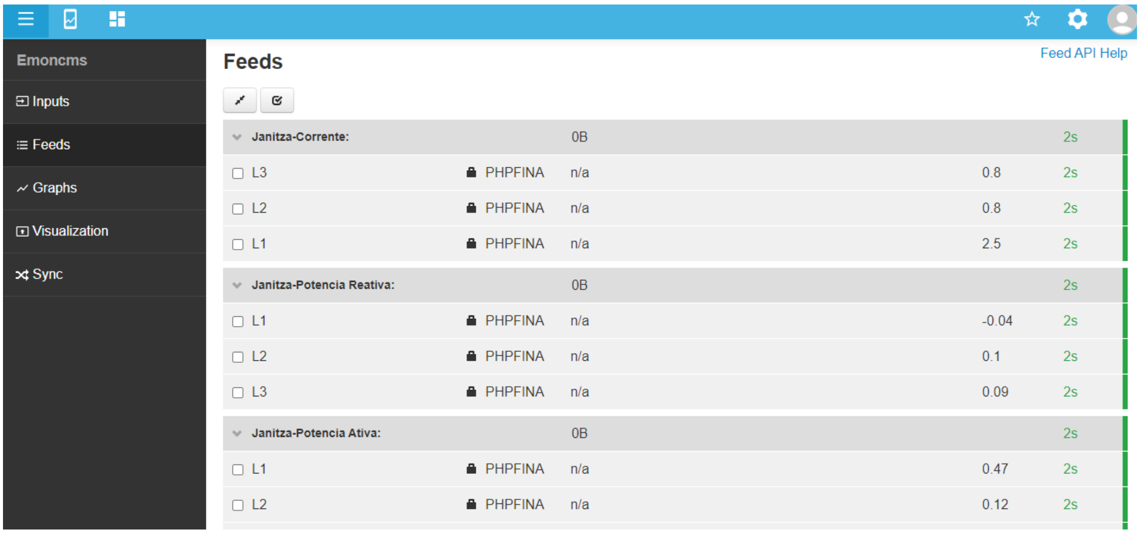


Show missing data: ☐ legend: ☑ feed tag: ☑

Start 20/02/2021 00:00:00 End 23/02/2021 10:30:00 Type Fixed Interval 900 Fix ☐ Limit to data interval ☑

Figure 4-6 - Graph tab available in the Emoncms environment built-in the new supervisory system

Both supervisory systems provide the option of downloading in a CSV format any available data in its environment for a customizable period of time. The current supervisory system allows this feature in the Evaluation tab and the new supervisory system allows this feature in the Feeds tab seen in Figure 4-7.



Janitza-Corrente:		0B		2s
☐ L3	PHPFINA	n/a	0.8	2s
☐ L2	PHPFINA	n/a	0.8	2s
☐ L1	PHPFINA	n/a	2.5	2s
Janitza-Potencia Reativa:		0B		2s
☐ L1	PHPFINA	n/a	-0.04	2s
☐ L2	PHPFINA	n/a	0.1	2s
☐ L3	PHPFINA	n/a	0.09	2s
Janitza-Potencia Ativa:		0B		2s
☐ L1	PHPFINA	n/a	0.47	2s
☐ L2	PHPFINA	n/a	0.12	2s

Figure 4-7 - Feeds tab available in the Emoncms environment built-in the new supervisory system

It’s interesting to compare the graphs built in both systems in order to understand the difference between the smaller polling time (10 seconds) in the developed supervisory system and the bigger polling time (5 minutes) in the original supervisory system. Figure 4-8 shows the graph comparing the acquisition of the measurement of active power of the SMA inverter by both systems. Figure 4-9 highlights the difference between the two supervisory systems by zooming in the data shown in Figure 4-9.

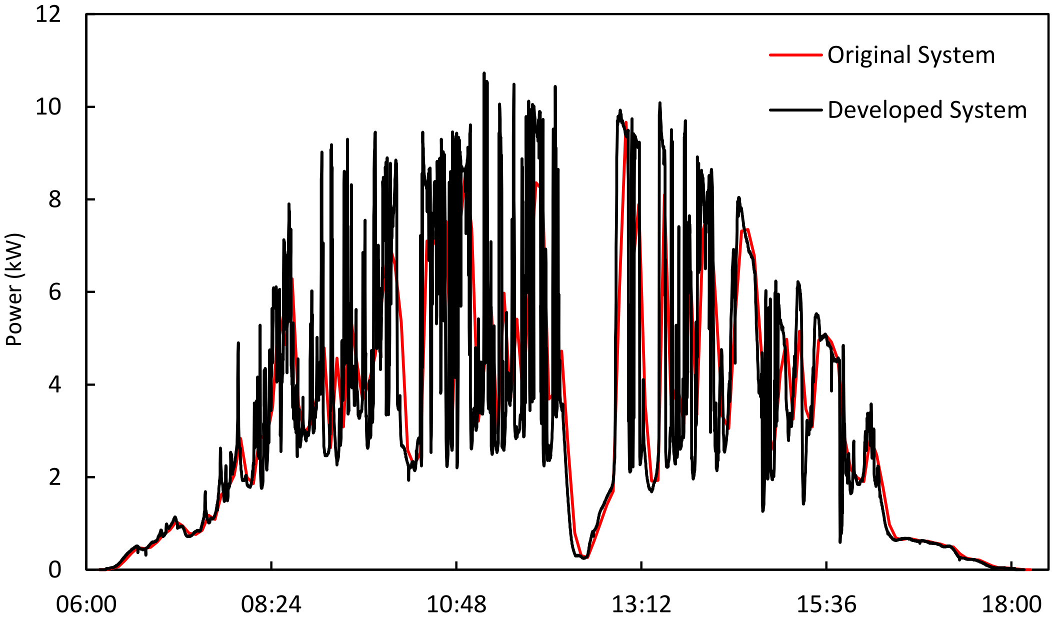


Figure 4-8 - Active AC power of SMA Inverter - Data acquired by the original and the developed system

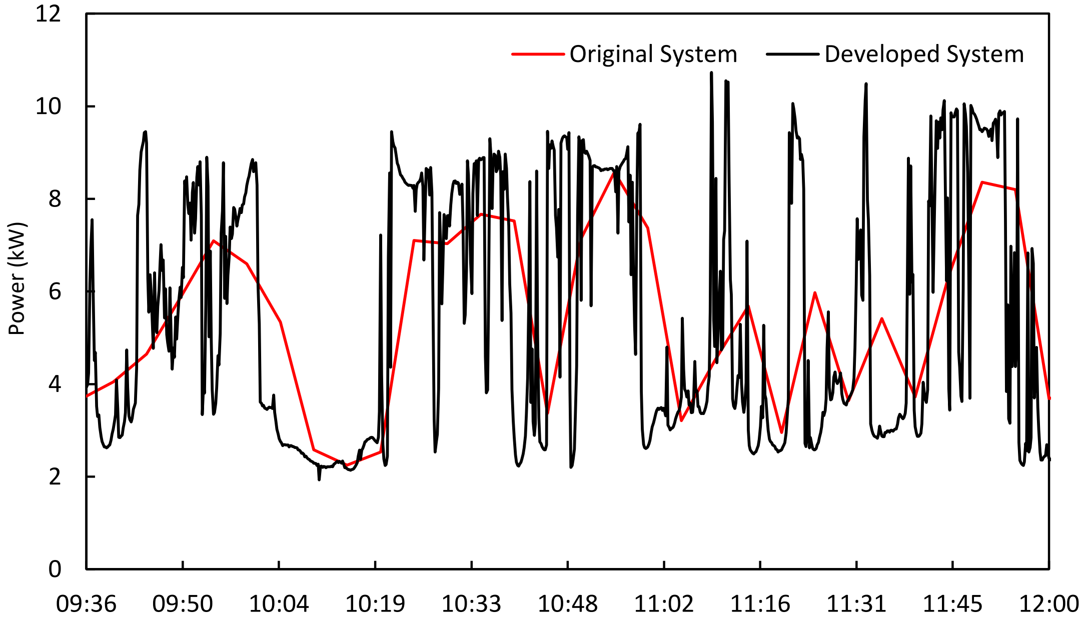


Figure 4-9 - Zoom of active AC power of SMA Inverter - Comparison analysis of original and developed systems

It’s possible to see that due to the bigger polling time, the original system loses a considerable amount of data that relates to variation in the power generation measured by the SMA inverter. By having a smaller polling time, the developed system is able to track all the variations that occur throughout the day, including higher peak values and lower valley values. It’s important to note that due to the polling time, the original system may lose important data for analysis especially cloudy days or when there are a lot of weather changes during the day. As the user can configure the system for lower polling times, the system can overcome this particular challenge. Also, this developed system provides the choice of determining the polling time for each feed. That means that different variables can each have a polling time that is more suitable for the type of data. In the original system, all the variables have the same polling time of 5 minutes.

The new supervisory system also includes graphs in the user interface available in Node-Red in order to present the user in the communication server with measurements to facilitate an eventual debugging of any potential communication failure. The data that is displayed in the Node-Red environment is shown via graphics to provide the user with a view of the behavior of certain variables during the last hour. The electrical variables that are on display in the Node-Red dashboard are the DC/AC voltage, DC/AC current, DC power and AC reactive and active power of each inverter. Measurements from Janitza UMG604 are also on display in this interface. These screens can be seen in Figure 4-10, Figure 4-11 and Figure 4-12.

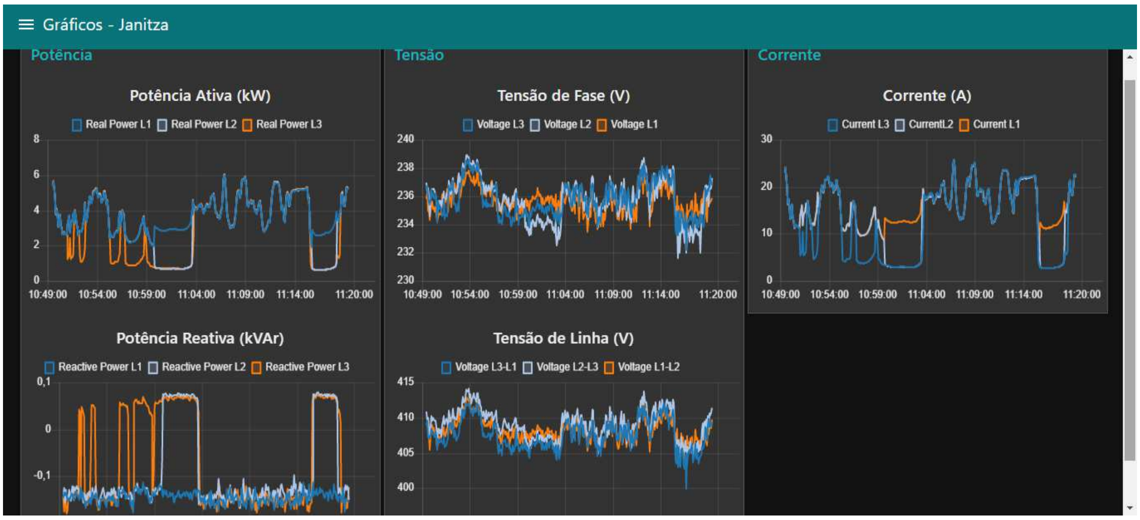


Figure 4-10 - Node-Red User Interface depicting data from the Janitza UMG604 meter

Figure 4-11 - Node-Red User Interface depicting some of the data on display of the Fronius inverter

Figure 4-12 - Node-Red User Interface depicting some of the data on display of the SMA inverter

As seen in Figure 4-10, Figure 4-11, Figure 4-12, the user can track the variable of interest for the last hour by selecting the desired device and searching for the correspondent graphic. With this feature, it's possible to trace a potential communication failure by checking if the communication server is connected correctly to the field devices.

5 Conclusions

The main goal of this work was to develop a supervisory system for a power plant using the concepts related to the Internet of Things, employing devices that were network ready and adapting the ones that weren't. In order to exemplify the usage of this supervisory system, it was applied to an existing solar power plant installed in the Escola de Engenharia at the Universidade Federal de Minas Gerais. However, it's important to notice that all the knowledge and the techniques used during the implementation of the supervisory system could be easily transferred not only for any power plant (with any type of source) but also for any industrial/residential plant as long as it's possible to establish a network in the plant.

Using smart devices (such as the power inverters used in the power plant and the hybrid inverters connected to the energy storage system) and adapting the devices that are not smart (the solar meter station connected to an intermediary device for example), it was possible to collect all the available data of the solar power plant and of the energy storage system. With this data accessible in the network, making it available for the selected microcomputer, Raspberry Pi, it was possible to develop different servers: communication and database servers that could help assembly the application server that houses the supervisory system per se. That way, the supervisory system could be populated and the screens depicting the entire system could be developed, providing the user with the means to monitor the operation of the power plant and of the energy storage system, and also the control of the latter.

The supervisory system was developed with Internet of Things concepts since this new technology provides a number of advantages. One of these advantages is potential increased autonomy for the plant since the devices are connected to a network that provides an information sharing environment (i.e. the devices can exchange data amongst them in order to operate more efficiently and accurately). Other advantages include how this technology facilitates the establishment of remote monitoring since it uses web platforms, such as the ones used here: Node-Red and Emoncms, and increased reliability, productivity, and efficiency.

5.1 Proposed Continuation

One idea for a continuation of this project would be to integrate a prediction of the energy generation in order to be able to schedule the operation of the energy storage system. This added capability would be able to predict the power output of the system based on the weather forecast for the next day and schedule when to charge/discharge the battery bank based on the forecast, adjusting in real time during the actual operation. This type of application is in research currently and the preliminary results are promising for renewable power plants (solar, wind).

In this vein, it would also be interesting to develop an optimization system for the automatic control of the plant in order to maximize the efficiency of the energy generation and minimize costs and losses. Currently, it's possible to see that many research works are developing optimization controls concurrently with the development of supervisory system for renewable power plants that are integrated with the Internet of Things.

Another proposed continuation would be to incorporate the status of any switches and circuit breakers that are installed in the plant in order to get an overview of the whole operation of the plant, not only of the measurements and status of the inverters. It would also be interesting to provide for an authorized user the possibility to control these devices in order to manage the plant.

Appendix A – Emoncms Feeds

The following tables, Table A.1, Table A.2, Table A.3, Table A.4 , show the complete feeds configured in the Emoncms environment detailing their name, number identification and sampling rate. The feeds are categorized by each individual device. The feeds that are highlighted indicate virtual feeds.

Table A.1 - Fronius Feeds

Variable	Feed	Database	Sample Time	Feed ID
Total	Fronius-Corrente	PHPFINA	10s	89
FaseA				90
FaseB				91
FaseC				92
FaseA	Fronius-Tensao			93
FaseB				94
FaseC				95
Corrente	Fronius-CC			96
Tensao				97
Potencia				98
Frequencia	Fronius-Medicoes			119
Potencia Reativa				120
Potencia Ativa				121
Energia Ativa				123
Geracao_Fronius	-			109
Geracao_CA_Fronius				114

Table A.2 - Janitza Feeds

Variable	Feed	Database	Sample Time	Feed ID
L3	Janitza-Corrente	PHPFINA	10s	45
L2				46
L1				47
L3	Janitza-Tensao			56
L2				57
L1				58
L1L2				59
L3L1				60
L2L3				61
L1	Janitza-Potencia Reativa			48
L2				49
L3				50
L1	Janitza-Potencia Ativa			51
L2				52
L3				53
Total				54
L1	Janitza-Energia Reativa			62
L2				63
L3				64
L1	Janitza-Energia Real Consumida			65
L2				66
L3				67
Total				68
L1	Janitza-Energia Real			69
L2				70
L3				71
Total				72

Table A.3 - SMA Feeds

Variable	Feed	Database	Sample Time	Feed ID
Total	SMA Potencia Ativa	PHPFINA	10s	128
L1				129
L2				130
L3				131
L1	SMA Tensao CA			132
L2				133
L3				134
Corrente	SMA Medicoes CC			125
Tensao				126
Potencia				127
Corrente	SMA Medicoes CC 2			142
Tensao				143
Potencia				144
Frequencia	SMA Frequencia			135
L1	SMA Corrente CA			136
L2				137
L3				138
Geracao_CC	SMA			140
Geracao_CA				141
Geracao_CC 2				145

Table A.4 - Solar Station Feeds

Variable	Feed	Database	Sample Time	Feed ID
Umidade_rel	Estacao_Solar	PHPFINA	10s	148
Pressao_rel				149
Vento_direcao				150
Irradiancia_global				151
Temp_ambiente				152

Vento_velocidade				153
Umidade_absol				154
Pressao_absol				155
Irradiancia	Celula_referencia			156
Temperatura				157

Bibliographic References

3-K Electric. Datasheet: **DMU51 + 60-V02 Spannungsmessumformer.** 2020. Available: https://www.3-k-elektrik.de/index.php?page=shop_product&id=13&lng=de. Access: 12 January 2021.

ADHYA, S.; SAHA, D.; DAS, A.; JANA, J.; SAHA, H. **An IoT Based Smart Solar Photovoltaic Remote Monitoring and Control Unit,** *In:* International Conference on Control, Instrumentation, Energy & Communication (CIEC), 2, 2016, Kolkata. p. 432-436. Available: https://ieeexplore.ieee.org/document/7513793. Access: 12 January 2021.

ÁLVAREZ, E.; CAMPOS, A. M.; GARCÍA, R.; GONZÁLEZ, S.; DÍEZ, C.; **Scalable and Usable Web Based Supervisory and Control System for Micro-grid Management,** *In:* International Conference on Renewable Energies and Power Quality, 1, 2010, Granada. p. 763-768.. Available: http://www.icrepq.com/icrepq'10/467-Alvarez.pdf. Access: 12 January 2021.

AQEEL, A. **Introduction to Arduino Nano,** *In:* The Engineering Projects**.** 2018. Electronic Archive. Available: https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html, Access: 12 January 2021.

BENAMMAR, M.; ABDAOUI, A.; AHMAD, S.; TOUATI, F.; KADRI, A. A Modular IoT Platform for Real-Time Indoor Air Quality Monitoring, **Sensors,** v. 18, n. 2, p. 1-18, Feb. 2018. Available: https://www.researchgate.net/publication/323181052_A_Modular_IoT_Platform_for_Real-Time_Indoor_Air_Quality_Monitoring. Access: 12 January 2021.

BB Smart Worx. **Frequent Asked Questions: RS-485 Connections FAQ.** 2020. Electronic Archive. Available: https://www.bb-elec.com/Learning-Center/All-White-Papers/Serial/RS-485-Connections-FAQ.aspx**.** Access**:** 12 January 2021.

BENGHANEM, M.; MAAFI, A. Data Acquisition System for Photovoltaic Systems Performance Monitoring, **IEEE Transactions on Instrumentation and Measurement,**

v. 47, n. 1, p. 30-33, Feb. 1998. Available: https://ieeexplore.ieee.org/document/728784. Access: 12 January 2021.

CAMACHO, E.; SAMAD, T.; GARCIA-SANZ, M.; HISKENS, I. Control for Renewable Energy and Smart Grids, **The Impact of Control Technology,** v. 6, p. 19-25, 2011. Available: https://www.researchgate.net/publication/284025192_Control_for_renewable_energy_and_smart_grids. Access: 12 January 2021.

CARNEIRO, A. V. **Projeto, Desenvolvimento e Implementação de Microrrede em Campus Universitário com Tecnologia Solar Fotovoltaica e de Armazenamento,** 2017, 177 f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia, Universidade Federal do Ceará, Fortaleza, 2017.

Cloudfare. Blog Post: **How Does SSL Work? SSL Certificates and TLS.** 2020. Electronic Archive. Available: https://www.cloudflare.com/pt-br/learning/ssl/how-does-ssl-work/. Access: 12 January 2021.

DE SOUZA, I. Blog Post: **Enteda o que é HTTP e o quão importante esse protocolo é para o seu site,** *In:* Rockcontent**.** 2019. Electronic Archive Available: https://rockcontent.com/br/blog/http/#:~:text=HTTP%20%C3%A9%20um%20protocolo%20de,do%20ingl%C3%AAs%20Hypertext%20Transfer%20Protocol. Access: 12 January 2021.

DUMITRU, C.; GLIGOR, A. SCADA Based Software for Renewable Energy Management System, **Procedia Economics and Finance,** v. 3, p. 262-267, Oct. 2012. Available: https://www.sciencedirect.com/science/article/pii/S2212567112001505. Access: 12 January 2021.

FORERO, N.; HERNÁNDEZ, J.; GORDILLO, G. Development of a Monitoring System for a PV Solar Plant, **Energy Conversion and Management,** v. 47, n. 15-16, p. 2329-2336, Jan. 2006. Available: http://www.sciencedirect.com/science/article/pii/S0196890405003134. Access: 12 January 2021.

FOUREAUX, N.; MARRA, F.; ANTHONY, A. **Usina Experimental Fotovoltaica TESLA Engenharia de Potência:** technical report, p. 1-35. Belo Horizonte, May, 2014.

FREITAS, C. M. Blog Post: **Protocolo Modbus: Fundamentos e Aplicações,** *In:* Embarcados**.** April. 2014. Electronic Archive. Available: https://www.embarcados.com.br/protocolo-modbus/. Access: 12 January 2021.

GUBBIA, J.; BUYYAB, R.; MARUSIC, S.; PALANISWAMI, M. Internet of Things (IoT): A vision, architectural elements, and future directions, **Future Generation Computer Systems,** v. 29, n. 7, p. 1645-1660, Sept. 2013. Available: https://www.sciencedirect.com/science/article/pii/S0167739X13000241. Access: 12 January 2021.

HILL, C.; SUCH, M. C.; CHEN, D.; GONZALEZ, J.; GRADY, W. M. Battery Energy Storage for Enabling Integration of Distributed Solar Power Generation, **IEEE Transactions on Smart Grid,** v. 3, n. 2, p. 850-857, Jun. 2012. Available: https://ieeexplore.ieee.org/abstract/document/6198748. Access: 12 January 2021.

HU, M.; CHEN, Y.; CHEN, Y.; CHANG, Y. Optimal Operating Strategies and Management for Smart Micro-grid Systems, **Journal of Energy Engineering,** v. 140, n. 1, p. 04013011-1 - 04013011-9, Mar. 2014. Available: https://www.researchgate.net/publication/264365782_Optimal_Operating_Strategies_and_Management_for_Smart_Microgrid_Systems. Access: 12 January 2021.

Inductive Automation. Blog Post: **What is SCADA?** *In:* Inductive Automation, 2020. Electronic Archive. Available: https://www.inductiveautomation.com/resources/article/what-is-scada. Access: 12 January 2021.

JIMÉNEZ-GARCÍA, J. L.; BASELGA-MASIA, D.; POZA-LUJÁN, J. L.; MUNERA, E.; POSADAS-YAGÜE, J. L.; SIMÓ-TEN, J. E, Smart device definition and application on embedded system: performance and optimi-zation on a RGBD sensor, **ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal**, vol. 3, n. 1, p. 46-55, Oct. 2014. Available: https://doi.org/10.14201/ADCAIJ2014384655. Access: 13 April 2021.

KHAN, R.; KHAN, S.; ZAHEER, R.; KHAN, S. **Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges,** *In:*International Conference on Frontiers of Information Technology, 10, 2012, Islamabad: p. 257-260. Available: https://ieeexplore.ieee.org/document/6424332. Access: 12 January 2021.

KOUTROULIS, E.; KALAITZAKIS, K. Development of an Integrated Data-Acquisition System for Renewable Energy Sources Systems Monitoring, **Renewable Energy,** v. 28, n. 1, p. 139-152, Jan. 2003. Available: https://www.sciencedirect.com/science/article/pii/S0960148101001975. Access: 12 January 2021.

L., A. Tutorial: **FTP: o que é, como funciona e qual o tipo para gerenciar arquivos na internet**, *In:* Hostinger Tutoriais**.** Nov. 2020. Electronic Archive. Available: https://www.hostinger.com.br/tutoriais/ftp-o-que-e-como-funciona#:~:text=FTP%20%C3%A9%20a%20sigla%20para,dois%20computadores%20conectados%20%C3%A0%20internet**.** Access: 12 January 2021.

LEE, I; LEE, K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises, **Business Horizons**, v. 58, n. 4, p. 431-440, Jul. 2015. Available: https://www-sciencedirect.ez27.periodicos.capes.gov.br/science/article/pii/S0007681315000373?via%3Dihub. Access: 13 April 2021.

MARIANO, F. A. **Integração de Tecnologias da Internet das Coisas a um Supervisor de Energia Através da Programação Orientada à[sic] Fluxos,** 2016, 71 f. Monografia (Curso Técnico em Sistemas de Telecomunicações) – Tecnologia em Sistemas de Comunicação, Instituto Federal de Santa Catarina, São José, 2016.

MARINHO, G. S. P. **Apresentação de uma Indústria Sucroalcooleira sob os Conceitos de Microrrede de Energia Elétrica**, 2011, 195 f. Dissertação (Mestrado em Engenharia Elétrica) – Engenharia Elétrica, Universidade Estadual Paulista, Ilha Solteira, 2011.

MDN Web Docs. Tutorial: **HTTP,** *In:* Developer Mozilla**.** 2020. Electronic Archive. Available: https://developer.mozilla.org/pt-BR/docs/Web/HTTP. Access: 12 January 2021.

MUKARO, R.; CARELSE, X. A Microcontroller-Based Data Acquisition System for Solar Radiation and Environmental Monitoring, **IEEE Transactions on Instrumentation and Measurement,** v. 48, n. 6, p. 1232-1238, Dec. 1999. Available: https://ieeexplore.ieee.org/document/816142. Access: 12 January 2021.

NARUTO, D. T. **Vantagens e Desvantagens da Geração Distribuída e Estudo de Caso de um Sistema Solar Fotovoltaico Conectado à Rede Elétrica,** 2017, 97 f. Monografia (Engenharia Elétrica) – Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017.

NI, Blog Post: **Desenvolvimento de aplicações com Modbus,** *In:* National Instruments**.** 2019a. Electronic Archive. Available: https://www.ni.com/pt-br/innovations/white-papers/14/application-development-with-modbus.html. Access: 12 January 2021.

NI, Blog Post: **O protocolo Modbus em detalhes,** *In:* National Instruments**.** 2019b. Electronic Archive. Available: https://www.ni.com/pt-br/innovations/white-papers/14/the-modbus-protocol-in-depth.html. Access: 12 January 2021.

OLIVEIRA, G.; RIBEIRO, R.; ROCHA, T.; COSTA, F.; NETO, J., **Controle Cooperativo do Fluxo de Potência com uso de Bateria,** *In:* Seminar on Power Electronics and Control – SEPOC 2019, 12, 2019, Natal. Available: http://sepoc2019.ct.ufrn.br/sepoc2019/images/arquivos/papers/track2/17.-Controle-Cooperativo-do-Fluxo-de-Potncia-com-uso-de-Bateria.pdf. Access: 12 January 2021.

PAPAGEORGAS, P.; PIROMALIS, D.; ANTONAKOGLOU, K.; VOKAS, G.; TSELES, D.; ARVANITIS, K. G. Smart Solar Panels: In-situ Monitoring of Photovoltaic Panels Based on Wired and Wireless Sensor Networks, **Energy Procedia,** v. 36, p. 535-545, 2013. Available: https://www.sciencedirect.com/science/article/pii/S187661021301148X. Access: 12 January 2021.

RAGHVENDRA, N. Blog Post: **SCADA System – Components, Hardware, & Software Architecture, Types,** *In: Electricalfundablog***.** 2019. Electronic Archive. Available: https://electricalfundablog.com/scada-system-components-architecture/. Access: 12 January 2021.

RINALDI, J. S. Blog Post: **Modbus TCP vs. Modbus RTU,** *In:* Real Time Automation**.** 2013. Electronic Archive. Available: https://www.rtautomation.com/rtas-blog/modbus-tcp-vs-modbus-rtu/. Access: 12 January 2021.

SICARI, S.; RIZZARD, A.; COEN-PORISIN, A. How to Evaluate an Internet of Things System: Models, Case Studies, and Real Developments, **Software Practice and Experience,** v. 49, n. 7, p. 1663-1685, Jul. 2019. Available: https://www.researchgate.net/publication/334593703_How_to_evaluate_an_Internet_of_Things_system_Models_case_studies_and_real_developments. Access: 12 January 2021.

SILVA, F. C. de S. Blog Post: **Artigo: Modelos Cliente-Servidor do Modbus TCP e Publish –Subscribe do MQTT: quando utilizar cada um deles e quais suas vantagens e desvantagens?** *In:* Novus. 2019. Electronic Archive. Available: https://www.novus.com.br/site/default.asp?Idioma=55&TroncoID=053663&SecaoID=0&SubsecaoID=0&Template=../artigosnoticias/user_exibir.asp&ID=618088#. Access: 12 January 2021.

SILVERIO-FERNÁNDEZ, M.; RENUKAPPA, S; SURESH, S. What is a smart device? - a conceptualisation within the paradigm of the internet of things, **Visualization in Eng**, vol. 6, n. 3, p. 1-10, May 2018. Available: https://doi.org/10.1186/s40327-018-0063-8. Access: 13 April 2021.

SoldaFria. Blog Post: **O que é um Arduino, para que serve, como funciona, onde comprar?** *In:* Blog do Soldafria. 2020. Electronic Archive. Available: https://www.soldafria.com.br/blog/o-que-e-um-arduino-para-que-serve-como-funciona-onde-comprar. Access: 12 January 2021.

STADLER, M.; NASLÉ, A. Planning and Implementation of Bankable Microgrids, **The Electricity Journal,** v. 32, n. 5, p. 24-29, Jun. 2019. Available: https://www.sciencedirect.com/science/article/pii/S1040619019301216. Access: 12 January 2021.

SUMANGALI, K.; KIRAN, P. Comparative Study of Open Source Tools for Internet of Things, **International Journal of Pharmacy and Technology,** v. 8, n. 4, p. 26235-26241, Dec. 2016. Available:

https://www.researchgate.net/publication/316886994_Comparative_study_of_open_source_tools_for_internet_of_things. Access: 12 January 2021.

TIGGEMANN, H. **Sistema de Monitoramento Remoto e Controle de Microrrede Híbrida Isolada,** 2015, 84 f. Dissertação (Mestrado em Engenharia Mecänica) – Faculdade de Engenharia, Universidade Federal de Minas Gerais, Belo Horizonte, 2015.

VENAYAGAMOORTHY, G. K.; SHARMA, R. K.; GAUTAM, P. K.; AHMADI, A. Dynamic Energy Management System for a Smart Micro-grid, **IEEE Transactions on Neural Networks and Learning Systems,** v. 27, n. 8, p. 1643-1656**,** Aug. 2016. Available: https://ieeexplore.ieee.org/document/7387757. Access: 12 January 2021.

WEBER, R. H. Internet of Things – New security and privacy challenges, **Computer Law & Security Review,** v. 26, n. 1, p. 23-30, Jan. 2010. Available: https://www.sciencedirect.com/science/article/pii/S0267364909001939. Access: 12 January 2021.

WORTMANN, F.; FLÜCTHER, K. Internet of Things, **Bus Inf Syst Eng**, v. 57, p.221-224, Mar. 2015. Available: https://doi-org.ez27.periodicos.capes.gov.br/10.1007/s12599-015-0383-3. Access: 13 April 2021.