UNIVERSIDADE D
COIMBRA

Miguel António Gonçalves Neves

Control of a Smart Resilient microgrid

Dissertação no âmbito do Mestrado em Engenharia Eletrotécnica e de Computadores, ramo de Robótica, Controlo e Inteligência Artificial orientada pelo Professor Doutor António Paulo Mendes Breda Dias Coimbra e pelo Professor Doutor Aníbal Traça de Carvalho Almeida e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra.

July 2024

1 2 9 0
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Control of a Smart Resilient Microgrid

Miguel António Gonçalves Neves

July 2024

Control of a Smart Resilient Microgrid

Supervisor:

Professor Doutor António Paulo Mendes Breda Dias Coimbra e Professor Doutor Aníbal Traça de Carvalho Almeida

Jury:

Professor Doutor Álvaro Filipe Peixoto Cardoso de Oliveira Gomes

Professor Doutor António Paulo Mendes Breda Dias Coimbra

Professor Doutor Rui Alexandre de Matos Araújo

July 2024

This dissertation was developed in collaboration with:

University of Coimbra

UNIVERSIDADE D
COIMBRA

Department of Electrical and Computer Engineering

deer.uc

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
E DE COMPUTADORES

Institute of Systems and Robotics

INSTITUTE OF SYSTEMS AND ROBOTICS
UNIVERSITY OF COIMBRA

Acknowledgements

Agradeço aos professores António Paulo Coimbra e Aníbal Traça de Almeida pela orientação neste trabalho. Obrigado pela disponibilidade e partilha de conhecimentos que tornaram este trabalho o mais completo possível. De outra forma não teria sido possível. Extendo o meu agradecimento aos professores Tony Almeida e Manuel Crisóstomo pelos aconselhamentos dados durante as reuniões do projeto.

Ao Instituto de Sistemas e Robótica da Universidade de Coimbra por me terem dado a oportunidade de fazer parte do projeto ResiMicrogrid, e garantirem que eu tivesse todas as condições para que este trabalho fosse bem-sucedido. Agradeço a todas as pessoas que trabalharam comigo no projeto ResiMicrogrid, em especial ao Miguel Cavaleiro pelas inúmeras horas de trabalho partilhadas até à apresentação final do projeto. Agradeço ainda ao Alexandre Matias pela disponibilidade e aconselhamentos dados não só no projeto, mas também no desenvolvimento deste trabalho.

À Fundação para a Ciência e Tecnologia pelo apoio financeiro nos projetos EXPL/EEI-EEE/1611/2021 e UIDB/00048/2020 (DOI 10.54499/UIDB/00048/2020).

À Universidade de Coimbra e Faculdade de Ciências e Tecnologia por me proporcionarem uma experiência académica única e repleta de memórias que ficarão para sempre no meu coração!

A toda a minha família, principalmente aos meus pais que sempre me apoiaram e deram todas as condições para conseguir alcançar os meus objetivos. Amo-vos muito! À minha irmã Mariana pela sua contagiante boa-disposição. A minha vida não seria a mesma sem as gargalhadas que me fazes dar desde pequeninos.

À Júlia, por ser a pessoa que está sempre comigo nos bons e nos maus momentos. Obrigado por todos os conselhos que me dás a pensar no melhor para mim. Tenho adorado todas as nossas experiências e espero viver muitas mais contigo!

Ao IMP3RIO, meu querido IMP3RIO! Não há palavras para descrever este grupo de sete reis. Agradeço-vos a todos do fundo do coração pelos incontáveis momentos icónicos, de lazer e descontração. Este percurso teria sido muito mais difícil sem vocês. Já são vários anos unidos, a apoiar-nos mutuamente e tenho a certeza que continuará assim. Viva o IMP3RIO!

À Pollux! Júnior Iniciativa da qual faço parte há 2 anos e meio e que foi praticamente a minha segunda casa. Agradeço a todas as pessoas que conheci ao longo destes anos, por tudo aquilo que aprendi convosco e por tornarem a minha jornada académica ainda mais incrível. Levo memórias inesquecíveis como o lançamento do Thestias, passagens de pasta e o projeto com a EFACEC. Desejo-vos o melhor daqui para a frente e que continuem a levar esta JI cada vez mais longe! E vai um grito de guerra malta!

Aos amigos que fiz no curso pela companhia em todas as horas passadas no DEEC, quer a estudar quer a fazer projetos. Obrigado.

Resumo

O aumento da frequência de desastres naturais e a crescente descentralização da rede elétrica, juntamente com a crescente procura por recursos energéticos sustentáveis, têm coletivamente impulsionado o interesse na adoção de novas tecnologias para modernizar a rede elétrica. As micro-redes surgem como resultado dessas tentativas de revolucionar a operação da rede, pois possibilitam a operação de forma independente da mesma e fornecem um método direto para integrar recursos de energia renovável. Esta tese investiga várias arquiteturas e dinâmicas operacionais de micro-redes, com o objetivo de desenvolver e implementar um algoritmo de controlo de cargas destinado a aumentar a resiliência em cenários críticos, como falhas de energia causadas por desastres naturais. Através de uma revisão abrangente da literatura, este estudo explora a viabilidade e os benefícios da implementação de micro-redes. O algoritmo desenvolvido é testado numa micro-rede piloto instalada no DEEC, e operado por um autómato integrado na mesma. O estudo apresenta um algoritmo eficiente e adaptável que gere eficazmente as cargas da micro-rede em função da energia armazenada nas baterias, assegurando resiliência na alimentação de cargas críticas durante interrupções de energia de curta e longa duração. Além disso, o desempenho do algoritmo é avaliado sob várias condições para garantir a sua robustez e escalabilidade. Assim, esta dissertação contribui significativamente para o avanço da tecnologia de micro-redes e otimização das estratégias de controlo de carga como soluções viáveis para o futuro dos sistemas energéticos.

Palavras-chave: micro-rede inteligente, resiliência, demand-response, fontes de energia renováveis, falha de energia, algortimo de controlo de cargas, load-shedding, cargas críticas.

Abstract

The increasing frequency of natural disasters, and the growing decentralization of the power grid, together with the rising demand for sustainable energy resources, have collectively fueled interest in adopting new technologies to modernize the power grid. Microgrids emerge as a result of these attempts to revolutionize the grid operation, as they can operate independently from the utility grid and provide a straightforward method for integrating renewable energy resources. This thesis investigates various architectures and operational dynamics of microgrids, with the goal of developing and implementing a load control algorithm aimed at enhancing resiliency in critical scenarios, such as power failures caused by natural disasters. Through a comprehensive literature review, the research explores the feasibility and benefits of microgrid deployment. The developed algorithm is tested in a pilot microgrid installed in DEEC, operating with an integrated automaton system. The study presents an efficient and adaptable algorithm that effectively manages microgrid loads taking into account the energy stored in the batteries, ensuring resilience by supplying critical loads during short or prolonged power outages. Additionally, the algorithm's performance is evaluated under various conditions to ensure its robustness and scalability. As a result, this research contributes significantly to advancing microgrid technology and optimizing load control strategies as viable solutions for future energy systems.

Keywords: smart microgrid, resilience, demand-response, renewable energy resources, power outage, load controlling algorithm, load-shedding, critical loads.

Contents

List of Acronyms

AC Alternate Current

DC Direct Current

DEEC Departamento de Engenharia Eletrotécnica e de Computadores

DOE Department of Energy

IEC International Electrotechnical Commission

DG Distributed Generation

ESS Energy Storage System

DER Distributed Energy Resources

GVL Global Variable List

MG Microgrid

PCC Point of common coupling

PLC Programmable Logic Controller

EV Electric Vehicles

PO Power Outage

PV Photovoltaic

SOC State Of Charge

POU Program Organization Unit

ST Structured Text

SAIDI System Average Interruption Duration Index

SAIFI System Average Interruption Frequency Index

List of Figures

List of Tables

Listings

1 Introduction

It is clear that the energy sector development has accelerated in the past few years, as the world's dependency on energy has never been higher and technological advancements have never been greater. Microgrids come as a result of those developments, as well as growing concerns about power resilience and integration of renewable sources, as they have the potential to revolutionize the utility grid by introducing a new entity with high controllability and autonomous operating capability.

One of the main characteristics of microgrids is that they can work autonomously from the utility grid using local generation and energy storage resources. The increasing frequency of natural disasters is one of the reasons for the rapid development of microgrids as, in many situations, these hazards consequently cause a large number of power delivery failures to the affected communities.

According to the Institute of Economics and Peace [1], the global number of natural disasters has increased tenfold since 1960 (Figure 1.1), and are expected to become more and more frequent. These natural hazards include droughts, earthquakes, volcanic activity, storms, floods and wildfires. Europe has been impacted by the second largest number of natural disasters with 1324 incidents from 1990 to 2019. Portugal is a good example of a country with a history of natural disasters, especially fires that affect remote locations in the country's interior region every year, but also heavy storms and floods that usually cause extensive power outages [2][3].

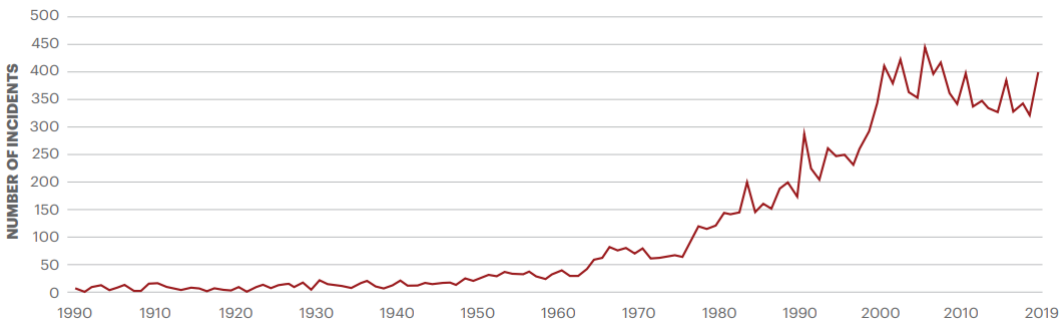


Figure 1.1: *Trend in number of natural disasters, 1900 to 2019. EM-DAT.*

As per the EU Adaptation Strategy for Climate Change [4], strengthening the resilience and capacity of the electric grid to withstand climate changes and extreme events is essential for minimizing economic costs and societal damage. The implementation of local microgrids that can operate and provide energy when power failures occur, is a big step forward in the enhancement of the electrical grid's resilience.

As the power grid continues to decentralize and the demand for reliable and sustainable energy sources increases, there has aslo been a notable surge in the development of microgrids. They are instrumental in optimizing the integration of renewable energy sources, by efficiently balancing supply and demand at the local level, enabling efficient generation, distribution and storage of electricity, as well as offering backup power during grid outages or emergencies. This ensures a continuous and uninterrupted electricity supply to critical facilities.

Additionally, a 2015 report from the U.S. Department of Energy analysed the grid and concluded that 70% of power transformers are 25 years of age or older, 60% of circuit breakers are 30 years or older and 70% of transmission lines are 25 years or older [5]. The aging infrastructure itself not only presents a threat to the power grid, but also opens up for other critical problems like cyber and physical attacks, as well as extreme weather events [6]. Microgrids present an opportunity to modernize and enhance local energy infrastructures, providing a cost-effective alternative to the substantial investments needed to upgrade the entire primary grid. Moreover, the decentralized nature of microgrids minimizes susceptibility to cyber threats that might otherwise target centralized grid infrastructure.

The goal of this thesis is to implement a functional microgrid in DEEC's garage, capable of ensuring a resilient response to any power outage scenario. This is extremely useful, especially when the main grid is under maintenance, or during an emergency such as a catastrophic event that may lead to a power failure. Having a photovoltaic generation system, an energy storage system and a load management algorithm, the microgrid is engineered to sustain power delivery to critical loads for an extended time period. Employing a load management algorithm rooted in load shedding, the microgrid automatically responds to power outage events, maintaining power supply and optimizing energy reserves.

1.1 E-REDES Open Data Portal

The Open Data Portal [7] developed by E-REDES provides a range of datasets with information of the electrical grid in Portugal, regarding renewable energy production, grid availability, along with others. In the context of this dissertation, the data displayed provides a good understanding of the electrical grid behaviour, in the way that it gives an idea of the number of occasions where having a resilient microgrid implemented would be beneficial.

The 'Scheduled Energy Interruptions' dataset presents the scheduled energy interruptions by postal-code. Figure 1.2 displays a chart that sums up the information available in that dataset.

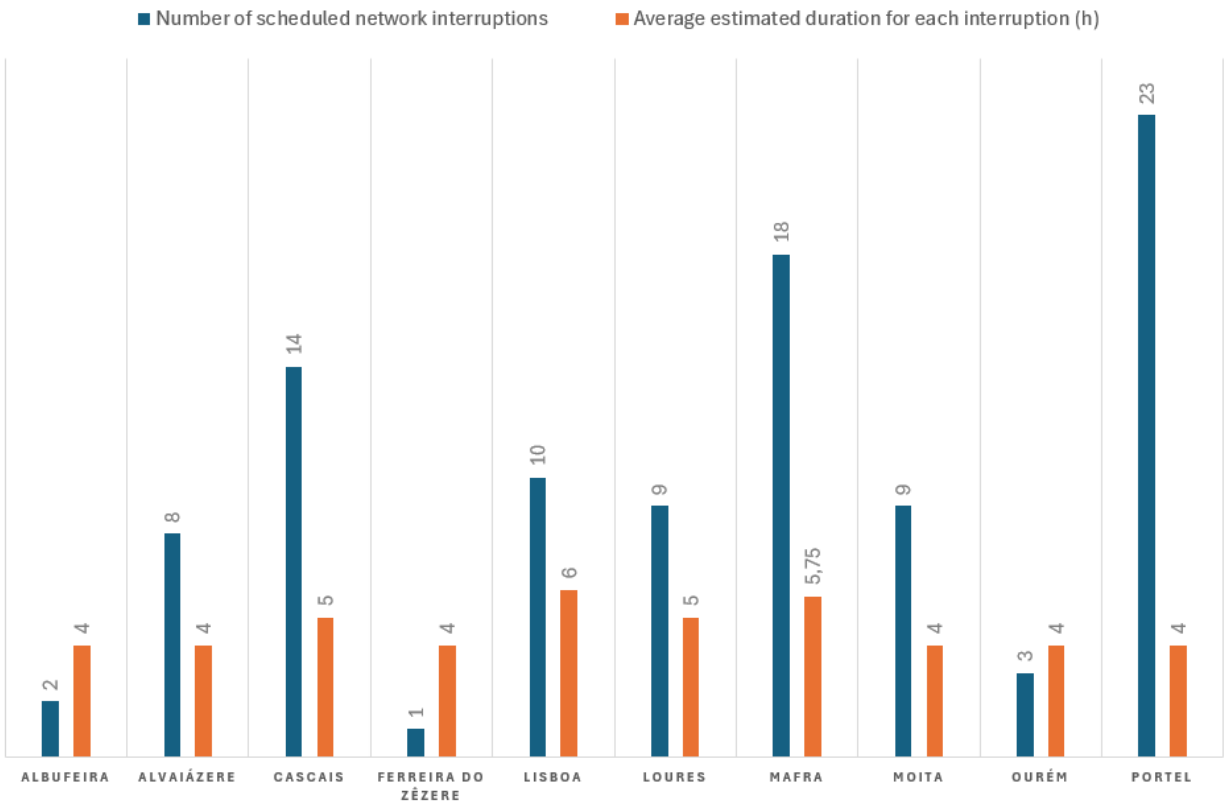
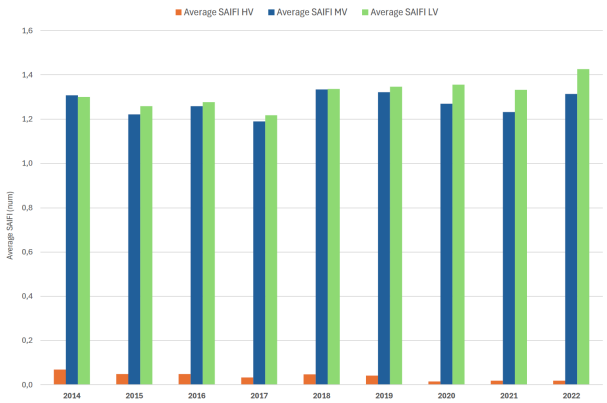


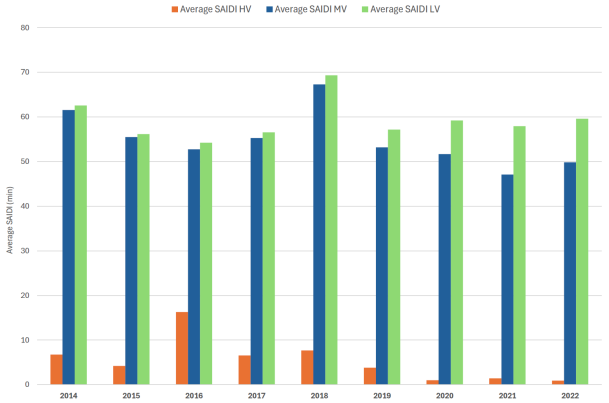
Figure 1.2: *Number of scheduled network interruptions by municipality and respective average estimated duration of each interruption (time period between 16/05 and 23/06).*

Analysing the chart, in just over a month, there are a total of 97 occurrences of planned energy interruptions, mounting up to 462.5 hours in total. Each scheduled interruption lasts between 4 and 6 hours, many times cutting energy supply until the middle of the morning. Installing a microgrid that can keep a power source during these blackout hours, would be crucial to companies or people who work at home that may be affected throughout the day.

To further complement this idea, the 'Service continuity indicators dataset' provides details of energy interruptions for high voltage (HV), medium voltage (MV) and low voltage (LV). To evaluate the reliability of the distribution network, the SAIDI (System Average Interruption Duration Index) and SAIFI (System Average Interruption Frequency Index) indices are used. SAIDI represents the average duration of interruption in the power supply indicated in minutes per customer, while SAIFI represents the average frequency of interruptions in power supply per costumer. Figure 1.3a shows that the SAIFI index on MV and LV has been slowly increasing, with each costumer having, on average, 1.2 to 1.4 interruptions per year. Microgrids that are able to operate independently from the main power grid, maintaining the energy supply to the end consumer, could be a solution for this issue. Figure 1.3b displays the average duration of these interruptions, varying from 50 to 70 minutes. On HV, the number of occurrences is significantly lower compared to MV and LV.



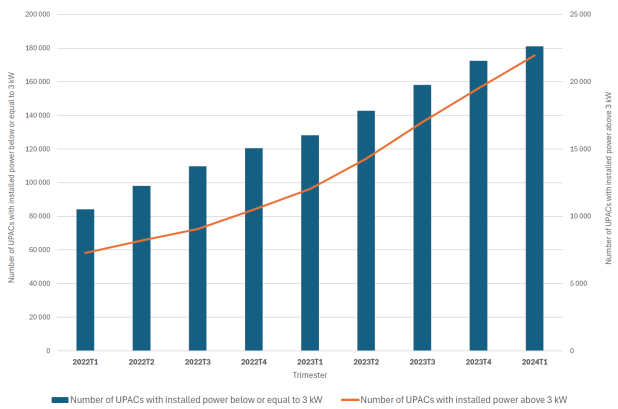
(a) Average value of SAIFI for HV, MV and LV since 2014.



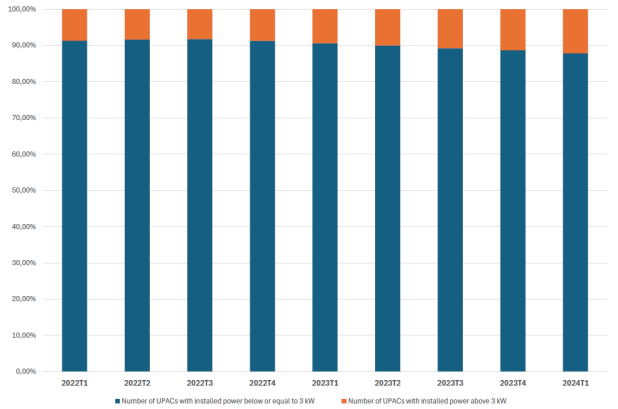
(b) Average value of SAIDI for HV, MV and LV since 2014.

Figure 1.3: *Service continuity indicators dataset analysis.*

On a side note, it is also worth to inspect the 'Total production units for self-consumption (UPAC)' dataset. Even though the microgrid pilot tested in this project can not be considered an UPAC, its electrical installation could be used for that purpose. Nonetheless, it is particularly interesting to analyse how many UPACs with an installed power similar to this project, which has 3 kW, could be resilient.



(a) Number of UPACs with installed power below or equal to 3 kW compared to the number of UPACs with installed power above 3 kW.



(b) Percentage of UPACs with installed power below or equal to 3 kW compared to the number of UPACs with installed power above 3 kW.

Figure 1.4: *Total production units for self-consumption dataset analysis (data available on 19/05/2024).*

The chart in Figure 1.4a shows a clear growth trend of the number of UPACs in the electrical grid since 2022. The number of UPACs with installed power below or equal to 3kW is almost tenfold bigger than the ones with installed power above 3kW. The comparison in percentage is clear on the chart in Figure 1.4b. In the first trimester of 2024, 87.9% of the total UPACs had an installed power below 3 kW. In total, that rounds up to 181 090 UPACs, with an installed power equal or lower than this dissertation's microgrid pilot, which demonstrates that this project is representative of most UPACs in the PT electrical grid.

In summary, the data analyzed, particularly concerning energy service interruptions, allows the conclusion that the microgrid pilot developed in this project could offer a viable solution for maintaining energy supply during the power failure scenarios observed in the datasets.

In the following sections of this chapter, the previous and developed work within the project are presented, as well as the dissertation structure.

1.2 Previous Work

This dissertation is a continuation of previously developed work on the ResiMicrogrid project which began in 2021, specifically papers [8][9][10] as well as master's dissertations in 2022 [11] and 2023 [12]. A concept of a load control algorithm was developed but it was implemented a simpler one, with an automaton that didn't allow a management of the different microgrid loads and had no communication ports available.

1.3 Developed Work

During the development of this dissertation there have been updates concerning the required electric devices, grid's circuit diagram and operation. The load control algorithm was rebuilt and implemented using a PLC that provides several communication ports, enabling configuration of master-slave communication protocols used for reading the microgrid devices data, as well as remote access to the microgrid operation. Furthermore, the state machine operation of the microgrid was also restructured due to renovations of the microgrid pilot architecture. Regarding the microgrid architecture, new devices were installed: an UPS to supply the local network switch and the PLC, a relay to detect power outages, relays connected to some of the PLC's outputs to control the contactors and a manual/automatic button to change the device controlling the microgrid. These changes were made in order to, not only implement features not thought of before, but also to meet the increasingly challenging goals associated with the gradual growth of the project overtime.

Additionally, a paper about the most recent developments was published in May 2024 [13].

1.4 Thesis Structure

This dissertation is structured in the following way:

1. **Introduction**: The introduction explains the motivation, purpose and goals of this thesis project;

2. **Current Research Review**: Explains the concept of microgrid, touching on principles such as microgrid configurations, modes of operation, as well as control strategies;

3. **Microgrid Design and Operation**: This chapter's objective is to present the microgrid pilot structure. It details its architecture, hardware and software used, overall electrical installation and operation;

4. **Microgrid Control Implementation**: Every detail concerning the implementation of the load controlling algorithm and state machine is explained in this chapter, with the intent of providing an overview of the microgrid's operating modes. The first section describes the control variables used for managing the state machine, which is then explained in section two. Finally, section three presents the simulation environment developed for monitoring the microgrid's behaviour;

5. **Results and Discussion**: The operation of the microgrid is validated with on-site testing scenarios that are presented and discussed;

6. **Conclusions and Further Developments**: Brief summary of the work developed. Possible future work that can be developed to improve the system operation.

This microgrid is a demonstration grid pilot of the project ResiMicrogrid developed for FCT (Fundação para a Ciência e Tecnologia).

2 Current Research Review

This chapter presents an overview of the state of the art of microgrids, introducing the definition of microgrid, why they are important and what is their purpose.

2.1 Introduction to Microgrids

The evolution of the state of the art of microgrids has been dynamic, with a wide range of ongoing advancements in new technologies and strategies. Microgrids are defined by the International Electrotechnical Commission (IEC) as a "group of interconnected loads and distributed energy resources with defined electrical boundaries forming a local electric power system at voltage levels of distribution of electricity, that acts as a single controllable entity and is able to operate in island mode" [14].

Power generation in the traditional power grid is highly centralized, with power and energy flowing unidirectionally from large synchronous generators through a distribution network to endusers. However, microgrids offer a flexible and decentralized approach to energy generation, distribution, and management. In urban settings, microgrids contribute to enhanced energy resilience by providing backup power during grid outages, ensuring uninterrupted services for critical infrastructure such as hospitals, data centers, and emergency response facilities. In remote areas, microgrids play a pivotal role in electrification efforts, bringing reliable power to communities that are traditionally underserved by centralized grids. Furthermore, microgrids are instrumental in optimizing the integration of renewable energy sources, by efficiently balancing supply and demand at the local level. At industry level, they are also implemented in order to improve energy efficiency and meet sustainability goals [15]. Overall, the adaptability of microgrids positions them as crucial contributors to the contemporary energy scene, effectively tackling obstacles and propelling initiatives for a power infrastructure that is both more robust, sustainable, and decentralized.

2.2 Exploring the Pros and Cons of Microgrids

By providing versatile solutions, microgrids offer a pathway to a more sustainable future by facilitating the integration of distributed energy resources (DERs). This, therefore makes it further necessary to monitor, manage and optimize the energy flows in a more intelligent way to ensure a balance between energy generation and load, as the latter is heavily dependent on the weather conditions at different times [16]. Intelligent control systems have been a crucial development in the sector of microgrids, as they use sophisticated algorithms and communication technologies to manage the distribution of energy within the microgrid. These smart systems enable microgrids to work efficiently, thus enhancing resilience and diminishing costs.

On the other hand, the development of microgrids has several intrinsic challenges, from high initial costs and lack of standardization, to regulatory barriers and technical integration challenges [17], including:

- **Operation challenges:** Frequency and voltage control discrepancies caused by the switch from grid to island mode operations.
- **Compatibility challenges:** The different parameters of the wide range of components included in a microgrid generate compatibility issues.
- **Integration of DERs:** The integration of DERs into the power grid faces notable obstacles such as uncertainty, unreliability, and dependence on climate conditions. Consequently, the power output from these sources may experience frequent and significant fluctuations, leading to instability in microgrids.
- **Regulation:** Effectively regulating microgrids is a crucial challenge as it guides and streamlines the penetration and integration of DERs into the utility grid. Nevertheless, the regulations governing the deployment of microgrids are limited, posing difficulties in ensuring their proper utilization.

There have been several efforts to overcome these difficulties from organizations like the U.S. Department of Energy (DOE) and IEC, and companies such as Schneider Electric and Siemens, that have been establishing common guidelines for microgrid systems, updating regulations as well as funding research to advance microgrid technologies [18][19][20].

In sum, despite their advantages, microgrids do present significant challenges and more research needs to be conducted to not only to come up with solutions, but also enhance the knowledge of these power systems.

2.3 Microgrid Configurations

This chapter provides an overview of the different microgrid configurations, describing each structures' benefits, drawbacks and actual applications worldwide.

2.3.1 AC Microgrids

An AC microgrid is a local and interconnected energy system that operates on alternating current (AC), designed to generate, distribute, and manage electricity for specific communities, facilities, or user groups. Figure 2.1 presents an AC microgrid structure example [21]. According to Hossain in [22], which resumes a wide range of microgrid applications around the world, AC microgrids are widely favored for their capacity to seamlessly incorporate existing renewable energy sources linked to current distribution networks, requiring minimal modifications to the electrical infrastructure. This architecture is characterized for facilitating the modification of voltage levels with low frequency transformers and for having high fault management capability with a wide range of protection devices. Typically, these microgrids adopt a radial topology, chosen for its minimal impact on grid operation and compatibility with standard protection schemes in distribution networks. In this setup, DERs are commonly linked to a shared busbar connected to a MV distribution network through circuit breakers, contactors, or static switches. Compliant with connection standards, these devices feature protective relays, enabling them to isolate the microgrid in the event of main power outages. Depending on the microgrid's application and control structure, these connection devices may integrate with additional components. Apart from the radial structure, AC microgrids can also adopt looped, meshed, or mixed topologies. However, these alternatives are rarely implemented due to their potential impact on grid operation and the established protection strategies commonly in use.

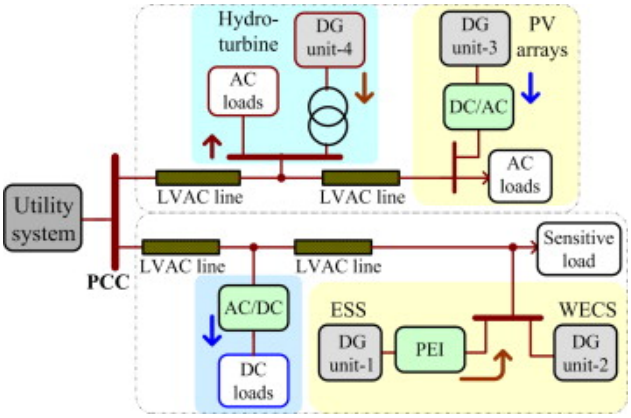


Figure 2.1: *AC microgrid structure with DG units and mixed types of loads [21].*

Even though AC microgrids can be seamlessly integrated into the existing AC power grid, this integration requires sophisticated control strategies for the synchronization process to maintain overall system stability [23], such as the need to synchronize distributed generation (DG) units in order to prevent the circulation of reactive power, which contributes to increased power losses in the transmission system. The extensive use of AC/DC converters presents challenges in terms of protection, communication, and overall operation for such microgrids. As a result, DC microgrids are gaining prominence as a viable solution, particularly in cases where only a few isolated DC devices need to be connected to newly established networks [24].

2.3.2 DC Microgrids

The increasing integration of DERs and electronically controlled loads into the system has heightened interest in DC microgrids over traditional AC microgrids, thanks to numerous advantages. These include no issues related to harmonics or frequencies, elimination of synchronization requirements in islanded mode, and the absence of issues with reactive power control, which increases overall efficiency [25]. Nevertheless, those require a high modification of the current distribution network and consequently the cost increases drastically. Figure 2.2 displays a DC microgrid structure example [21].

DC microgrid configurations hinge on harnessing DERs to curtail carbon emissions. Nevertheless, the introduction of DERs within DC microgrids results in reliability challenges owing to the intermittent nature of renewable sources. To counterbalance these concerns, some microgrids incorporate fossil fuels such as diesel or gas in conjunction with DERs [26][27]. While this bolsters reliability to a certain extent, achieving substantial efficiency implies the formulation of effective power management schemes. Furthermore, it presents consequences

such as emission of greenhouse gases, increasing costs, and the need of easy accessibility to fuel. The inclusion of diverse energy storage devices can significantly enhance the reliability of microgrids. However, it complicates control dynamics, particularly regarding DC link voltage control and power distribution among varied sources and loads [28]. The traditional control mechanisms in DC microgrids, despite their simplicity, may result in fluctuations in DC bus voltage and disparities in current sharing [29].

All in all, many technical issues like complex control, difficult planning and operation, power and SOC imbalances, high fault current resulting from the short circuits and greater uncertainties due to the varying nature of wind speed and irradiance, are observed in DC microgrids. As a result, careful planning and operation are crucial for system planners and operators to uphold the flexibility, reliability, and stability of DC microgrids. Continuous advancements in techniques and cutting-edge technologies are being developed to address emerging challenges in the implementation of DC microgrids. Noteworthy examples include advanced planning methods, efficient operations, enhanced power and current sharing techniques, and optimization methods aimed at improving DC bus voltage restoration. Regardless of the increasing interest in DC microgrids for their potential advantages, those examples remain a challenging task, which arises from the need to address large-scale integration of DER, manage power balance, and regulate DC link voltage — all within the framework of comprehensive planning, operation, and control considerations [25].

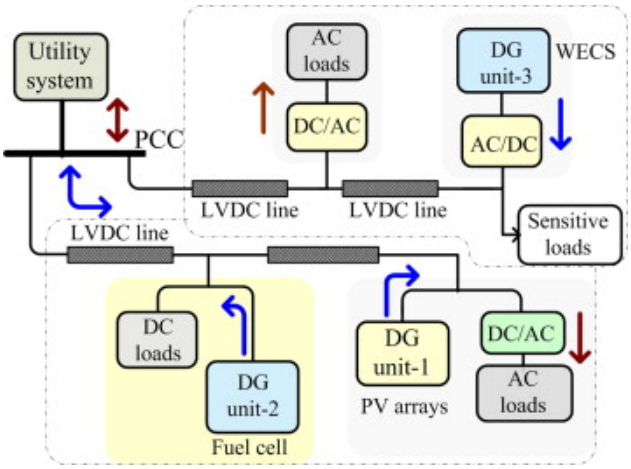


Figure 2.2: *DC microgrid structure with DG units and mixed types of loads [21].*

2.3.3 Hybrid Microgrids

Hybrid AC/DC microgrid configurations have sparked considerable interest by combining the benefits of both AC and DC architectures [30][31]. A key feature of these configurations is the integration of AC and DC networks within the same distribution grid, allowing for the seamless incorporation of AC and DC-based DG, energy storage systems (ESS) and loads. This characteristic offers an efficient means for integrating emerging DER or EV units with minimal modifications to the existing distribution grid, thereby reducing overall costs. Figure 2.3 represents a typical structure for a hybrid microgrid [32].

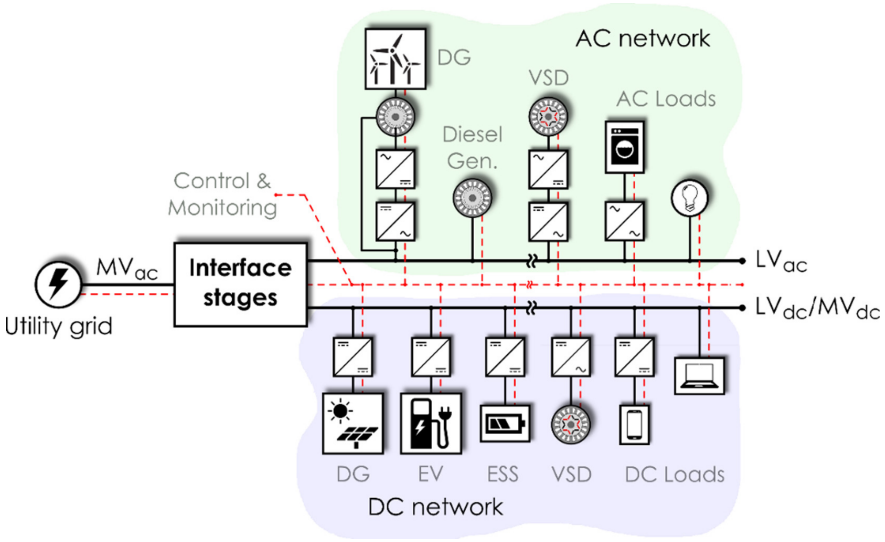


Figure 2.3: *Example of a hybrid microgrid configuration [32].*

The most important advantages of these microgrids are their integration, synchronization, voltage transformation, and economic feasibility [32]. These microgrids allow direct connection of AC or DC-based devices to the network with minimal interface elements, which reduces conversion stages, minimizing energy losses. This feature makes hybrid microgrids well-suited for integrating an increasing number of DC-based units, while still accommodating AC-based devices connected to the AC network. Furthermore, there is no requirement for synchronization of generation and storage units, as they are directly connected to either the AC or DC network. Consequently, the control strategy for these devices is simplified. Moreover, voltage level modification can be easily performed on the AC side using transformers. On the DC side, conversion is achieved through the use of DC-DC converters. Economically, hybrid microgrids can be developed by adding a power converter to the existing distribution grid and communication network for connected devices. While this introduces a higher overall cost compared to AC microgrids due to the main power converter, the investment

can be recovered faster with an increasing number of attached devices, as the total number of interface converters is reduced [32].

However, this architecture also presents several drawbacks that need further research [32]:

- **Protection:** While a wide variety of protection devices are available for AC-based networks, DC protection devices have not been as extensively researched. Fault isolation is simpler in AC networks due to zero-crossings of the current, a method not directly applicable to DC grids.

- **Reliability:** The reliability of hybrid microgrids is lower than that of AC ones due to the introduction of an interface power converter in the distribution network to generate the DC-link. Nevertheless, the reliability of connected devices improves as the number of converter stages decreases.

- **Control Complexity:** Managing hybrid microgrids is more complex than their counterparts, involving control of devices attached to both AC and DC networks and the interface power converter between them. Ensuring stable and reliable power supply for both networks becomes challenging, particularly in autonomous or islanded modes.

In conclusion, hybrid microgrids represent a pivotal advancement in the realm of energy distribution, seamlessly integrating both AC and DC components to create a dynamic and resilient power infrastructure. Their adaptability and versatility make them a compelling solution for diverse applications, providing efficient and sustainable energy solutions.

2.4 Microgrid Modes of Operation

There are two possible modes of operation of a microgrid: grid-connected and islanded mode.

2.4.1 Grid-connected mode

During this operational mode, the microgrid is linked to the utility grid via a static transfer switch, with the connection point referred to as the point of common coupling (PCC). The microgrid controller consistently oversees both generation and demand within the microgrid, facilitating the export of excess power or the import of deficient power through the inverter based on load and source conditions. Upon grid connection, the microgrid looses control over system frequency and voltage, transitioning to P-Q control to regulate active and reactive power.

2.4.2 Islanded-mode

Microgrids transition to islanded mode due to grid faults, maintenance, or economic considerations. In this mode, the microgrid operates independently without support from the main grid, thus increasing the difficulty of the control operation. When connecting a load to the power grid, its impact on grid stability is minor due to the significant difference in size between the load and the total grid load. However, in microgrids that are in islanded mode, even small loads can impact power stability and operation. This is because the microgrid's generation capacity is limited, and any changes in load can directly affect the balance between generation and consumption. During this phase, the microgrid becomes highly susceptible to fluctuations in generation and load variations, and has to be able to quickly regulate sudden voltage and frequency changes at the PCC, assuring active and reactive power sharing [33].

To ensure reliability in islanded conditions, a consistent power source is essential, often met through the use energy storage devices. Efficient control of these devices allows for maintaining constant voltage and frequency in islanded conditions. Autonomous microgrid operation involves diverse control strategies, including master-slave control methods, control area network communication, and voltage and frequency droop strategies based on local measurements.

2.5 Different Control Methods in Microgrids

Within a microgrid, diverse control methods are employed to guarantee dependable operation in both grid-connected and islanded modes. The three primary types of control methods, namely PQ control, V/f control, and droop control, are selected based on the characteristics of the DG and prevailing operating conditions.

2.5.1 V/f Control

V/f control's key objective is to keep the system frequency and voltage magnitude constant, independent of the microsource's actual active and reactive power outputs. In this control method, an active power controller adjusts the active power output to maintain the frequency at a specified reference value. Meanwhile, a voltage controller manages the reactive power output to sustain the voltage at the designated reference value. V/f control finds widespread application when the microgrid operates independently in islanded mode [34].

2.5.2 PQ Control

The primary goal of PQ control is to ensure a constant level of active power and reactive power from the microsource, provided the frequency and voltage deviation remains within specified limits [34].

In PQ control, the initial step involves decoupling active and reactive power to enable independent control. The active power controller is designed to uphold a consistent active power *(P)* output at a designated reference value within the permissible frequency range. Simultaneously, the reactive power controller strives to maintain a steady reactive power *(Q)* output at the specified reference value within the acceptable voltage range. It's worth noting that while this PQ control method effectively manages active and reactive power, it does not inherently sustain constant frequency and voltage. Hence, an additional distributed generator is required to regulate the microgrid's voltage and frequency within acceptable parameters, which is why this method is usually used in grid-connected mode, as the primary power grid assumes responsibility for maintaining the microgrid's voltage and frequency [34].

2.5.3 Droop Control

Droop control is influenced by traditional generators' drooping characteristics, emphasizing the correlation between active power and frequency, as well as reactive power and voltage. It involves setting voltage and frequency references and consists of two main types: the active power control mode, akin to traditional synchronous generator regulation, and the voltage control mode [34].

2.5.3.1 Active Power Control

Within a microgrid, the load undergoes constant fluctuations, leading the generators to adjust their power output in response to frequency deviations. Active power control is accomplished through the manipulation of the operating point of P/f droops, aligning them with the frequency deviation resulting from load fluctuations. Equation 2.1 establishes the droop correlation between real power output and frequency [35]:

$$\Delta P = P_2 - P_1 = \frac{f_1 - f_2}{S_p} \tag{2.1}$$

where ΔP indicates the change in active power output of the DGs and S_p indicates the slope of the V/f droop curve, as displayed in Figure 2.4 [35].

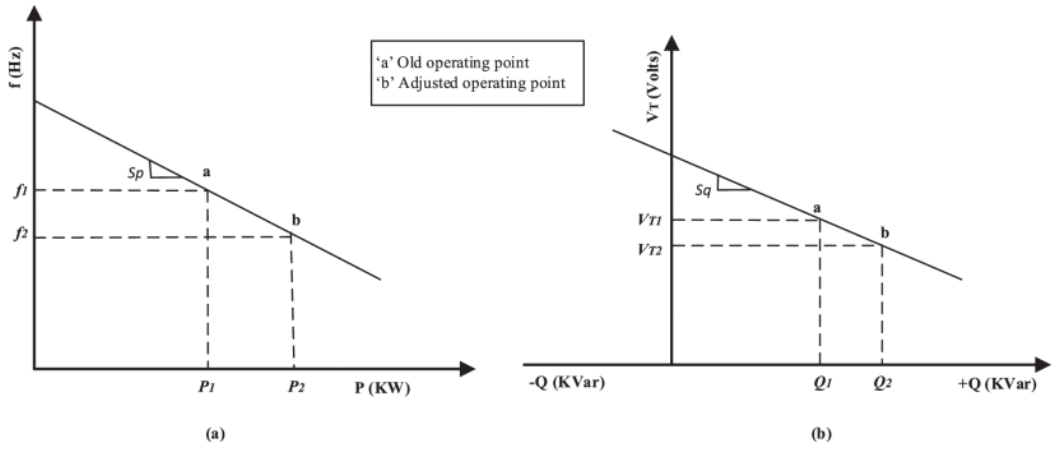


Figure 2.4: *Displays (a) V/f droop (b) Q/V droop characteristics of DGs [35].*

2.5.3.2 Voltage Control

Voltage control is achievable by adjusting the reactive power output of the DGs in response to load fluctuations. The mathematical relationship between reactive power and terminal voltage (V_T) is as follows [35]:

$$\Delta Q = Q_2 - Q_1 = \frac{V_{T1} - V_{T2}}{S_q} \tag{2.2}$$

where ΔQ and S_q indicates the change in the reactive power and the slope of the Q/V curve, respectively, as presented in Figure 2.4 (b) [35].

2.6 Control Strategies in Microgrids

2.6.1 Master-Slave Control

The Master-Slave control strategy comprises a master controller within the microgrid and slave controllers that synchronize with the directives of the communication-linked master controller. The master converter operating in voltage source mode manages the DC bus voltage, while slave converters in current source mode distribute current proportionally based on the overall load current [36]. This control strategy is based on High-Bandwidth Communication (HBC) and one of its drawbacks is its vulnerability in the event of the master unit's failure, as the microgrid lacks a fallback option, therefore being dependent on the master converter. However, it provides a higher level of coordination in multi-layer control by allowing the whole microgrid to operate simultaneously [37][38].

Figure 2.5 presents the master-slave control strategy [36]. Droop control and V/f control are two voltage control strategies utilized by the master converter. However, when compared to droop control, the V/f control method exhibits a slower dynamic response. The V/f controller is applied when the microgrid is in islanded operation mode, and the P/Q controller is for grid-connected mode.

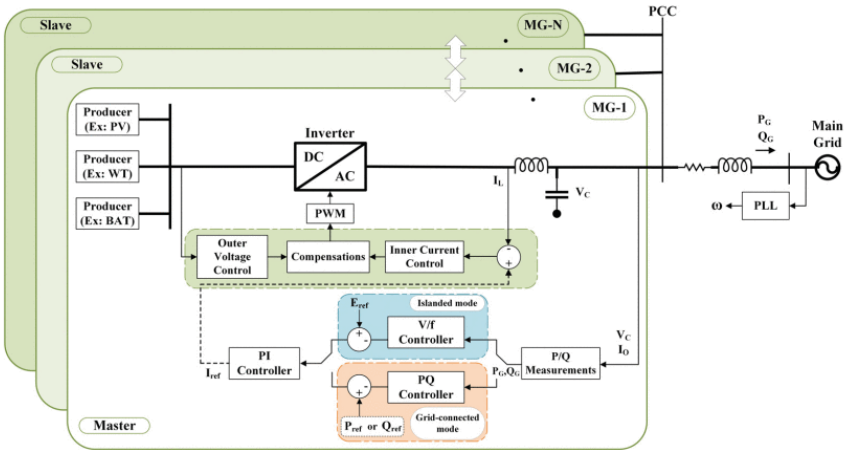


Figure 2.5: *Master-Slave control structure [36].*

2.6.2 Peer-to-peer Control

The peer-to-peer control strategy operates without relying on a hierarchical structure or central controller. Instead, it leverages a computer network comprising a defined set of agents. Figure 2.6 presents the control structure controlled by the peer-to-peer strategy [36].

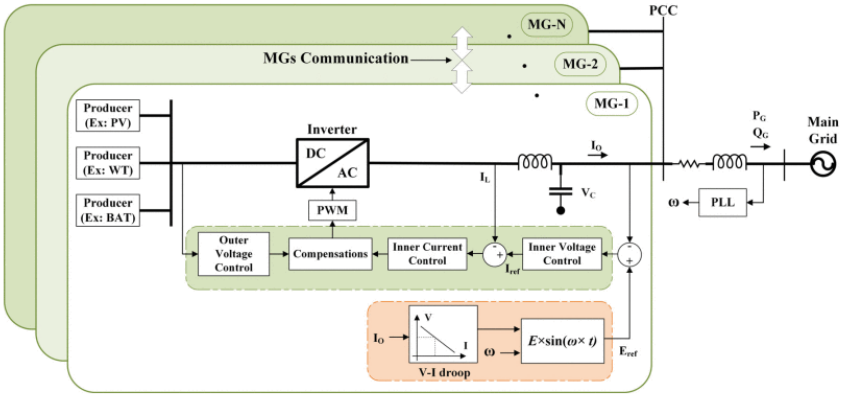


Figure 2.6: *Peer-to-peer control structure [36].*

Droop control is implemented within the voltage control scheme when microgrids are predominantly controlled by the peer-to-peer paradigm [39].

As it can be seen from Figure 2.5, due to the existence of an integrator in the PI controller, the seamless transfer between grid-connected and islanding operation mode is not sufficient. Consequently, master-slave control is commonly employed during islanded states, while the peer-to-peer control scheme is primarily utilized during grid-connected operation [36].

2.6.3 Hierarchy Control

The hierarchical control strategy is widely favored for its ability to ensure smooth transitions between islanded and grid-connected modes during transient operations. This structure comprises primary, secondary, and tertiary control levels, effectively managing the static and dynamic stability of microgrids. Figure 2.7 shows an overview of the incorporation of hierarchical control in a grid-connected individual MG [36].

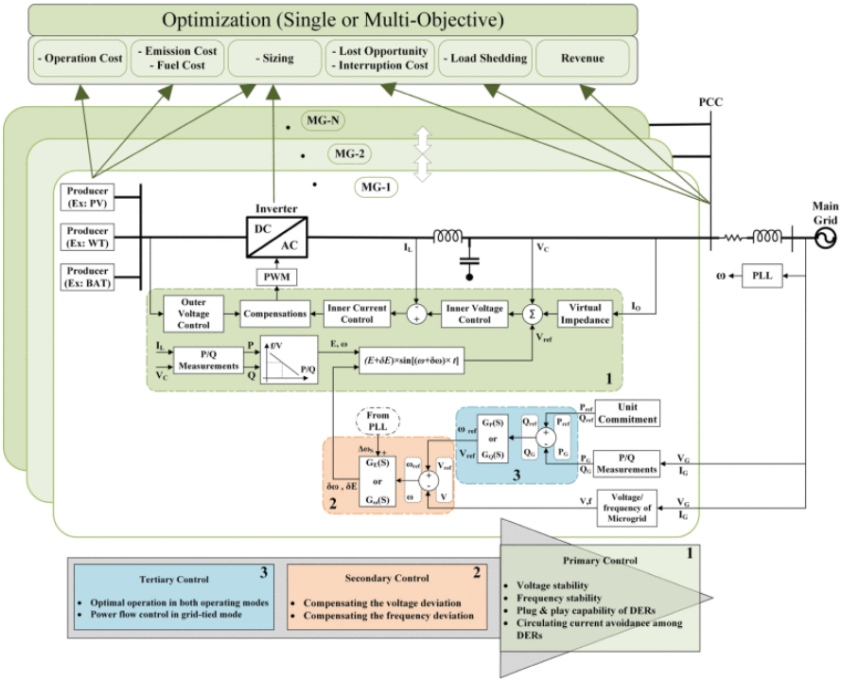


Figure 2.7: *Hierarchical control structure [36].*

Primary control oversees voltage and frequency stability through the regulation of active and reactive power. Any deviations in output voltage and frequency are compensated for in the secondary control stage. Ultimately, the tertiary control level manages the optimal power flow between microgrids and the utility grid.

This chapter has provided a comprehensive review of the state-of-the-art in microgrid technology. The primary objective was to identify the current advancements in the literature related to operation modes and control strategies, and how they depend on the microgrid configuration. These findings underscore the importance of advancements in DERs, ESS, and smart grid technologies, which are instrumental in enhancing the efficiency and reliability of microgrids. Despite these improvements, there are still some gaps that require further investigation, such as the compatibility challenges of integration of different components in microgrids, and limited microgrid deployment regulations. Additionally, there is also a small gap between the development of advanced control algorithms and their real-world implementation.

This review has established a foundation for the following chapters of this thesis. By addressing the identified gaps, my research aims to contribute to the development of more efficient microgrid solutions regarding the implementation of control algorithms. The next chapter will delve into the design of the microgrid pilot developed in this dissertation.

3 Microgrid Design and Operation

In this chapter, the implemented MG's architecture, components and operation are presented and explained.

3.1 Architecture

The electrical block diagram of the used pilot MG installation is shown below (Figure 3.1). The architecture of the MG was developed considering solar photovoltaic generation, energy storage system, and load management. Therefore, an island operation of the MG is vital as the MG is supposed to deliver power during critical and disaster situations, i.e when there is a power outage. This is achieved using contactors controlled by a PLC (K1 and K10 in Figure 3.1) to disconnect the MG from the utility grid.

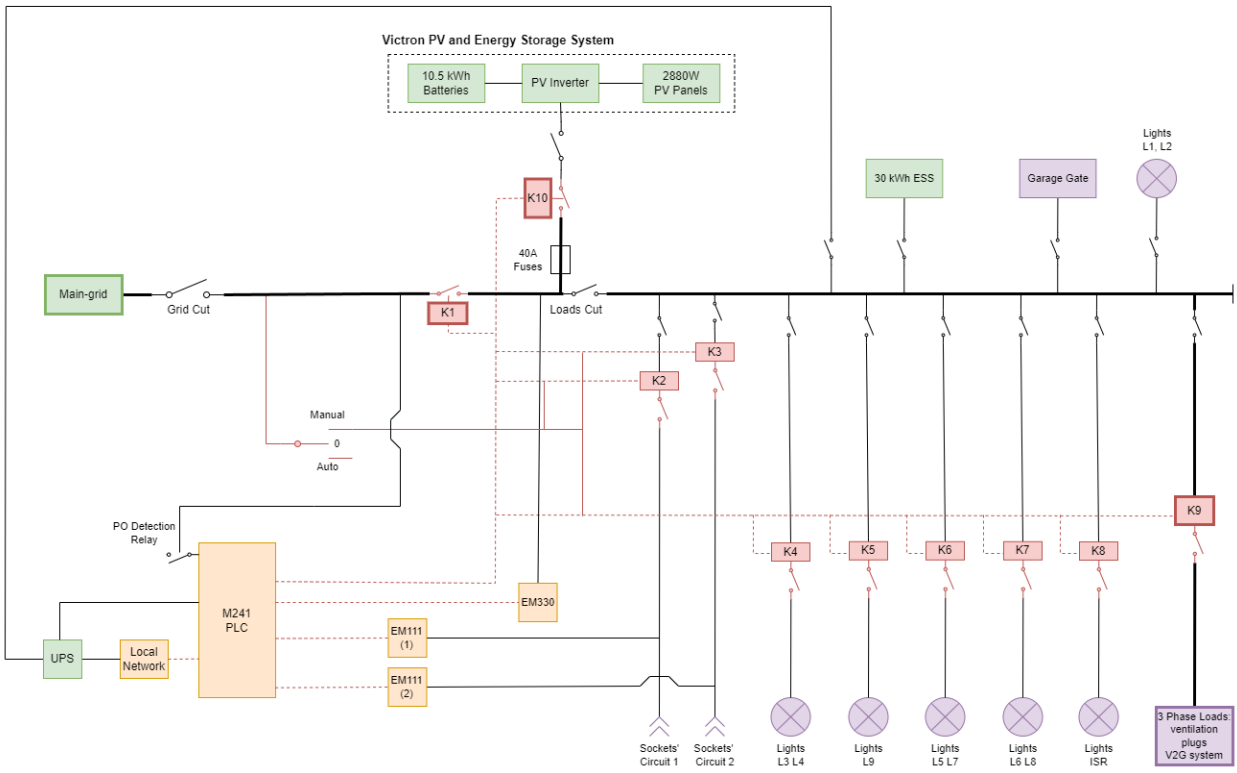


Figure 3.1: *Electrical block diagram of the microgrid.*

3.2 Main Components

This section presents the main components that are essential for the microgrid operation.

3.2.1 Energy Monitoring Devices

- **Carlo Gavazzi EM111DINAV81XS1PFA:** single-phase energy analyser designed for active energy measurement, communicating via the RS485 serial line port 2 of the PLC, utilizing the Modbus RTU protocol in slave function mode. Each socket circuit is monitored by one of these devices;
- **Carlo Gavazzi EM330DINAV53HS1X:** three-phase energy analyser designed for active energy measurement, communicating via the RS485 serial line port 2 of the PLC, utilizing the Modbus RTU protocol in slave function mode. This device can be used for global load metering.

3.2.2 Contactors

Contactors enable the load shedding operation as they either open or close circuits of the electric switching board, controlling the MG operation mode and the loads that are supplied by the PV system. Contactors 1 and 10 control the main grid cut and the PV system respectively, therefore controlling if the MG operates in grid-connected or island mode. The two Carlo Gavazzi EM111 check the power being consumed at socket circuits 1 and 2, which are controlled by contactors 2 and 3 respectively. Contactors 4 to 8 all control different light circuits, while contactor 9 controls the 3-phase loads that need to be cut off when transitioning to island mode. These contactors are connected to the PLC outputs.

3.2.3 Programmable Logic Controller

The PLC used is the Modicon TM241CE24R (datasheet in appendix .1.5). This device offers 10 discrete outputs (6 of them relay outputs, 4 fast transistor outputs) and 14 discrete inputs, having an Ethernet port with a RJ45 connection to the local ethernet network, enabling wireless access through the programming software. The software used to program the PLC is the EcoStruxure Machine Expert software from Schneider Electric.

3.2.4 PLC Software - EcoStruxure Machine Expert

This software is used to program the state machine operation of the microgrid, implement the load control algorithm and configure the master-slave communication protocols needed for reading the data from the microgrid devices. Additionally, a monitoring panel to monitor the microgrid operation is also developed using the software.

The algorithm is developed in Structured Text, a language that enables the use of expressions and instructions, offering a wide range of programming loop constructs. Its versatility makes it well-suited for developing intricate algorithms.

3.2.5 PO Detector Relay

The MG uses a power outage (PO) detection relay connected to a digital input of the PLC to detect a power failure from the main-grid.

3.2.6 Manual/Automatic Switch

A manual/automatic switch is installed so that when in "automatic" the garage's electrical circuits are controlled by the microgrid's PLC and when in "manual" they are not. This is intended to enable changes in the load control algorithm without removing the automaton from the panel board.

3.2.7 UPS

A UPS unit is integrated in the microgrid architecture, supplying both the PLC and the local computer network switch when the utility grid is down. Additionally, the UPS itself is always being supplied by the busbar of the garage's main panel board (see Figure 3.1). This UPS was installed because the PLC and server swicth are indispensable for the operation of the MG since, without the PLC, no loads would be controllable and without the network switch it would not be possible to monitor the MG operation.

3.2.8 Victron PV and Energy Storage System

The Photovoltaic and Energy Storage System has the following main components:

- array of 6 polycrystalline panels, 2880 W total power;
- 1x Victron SmartSolar MPPT 150/35 Charge Regulator;
- 1x Victron MultiPlus II 3000VA Inverter/Charger;
- 1x Victron Cerbo GX;
- 3x Pylontech US3000C Battery, 10.5 kWh total capacity.

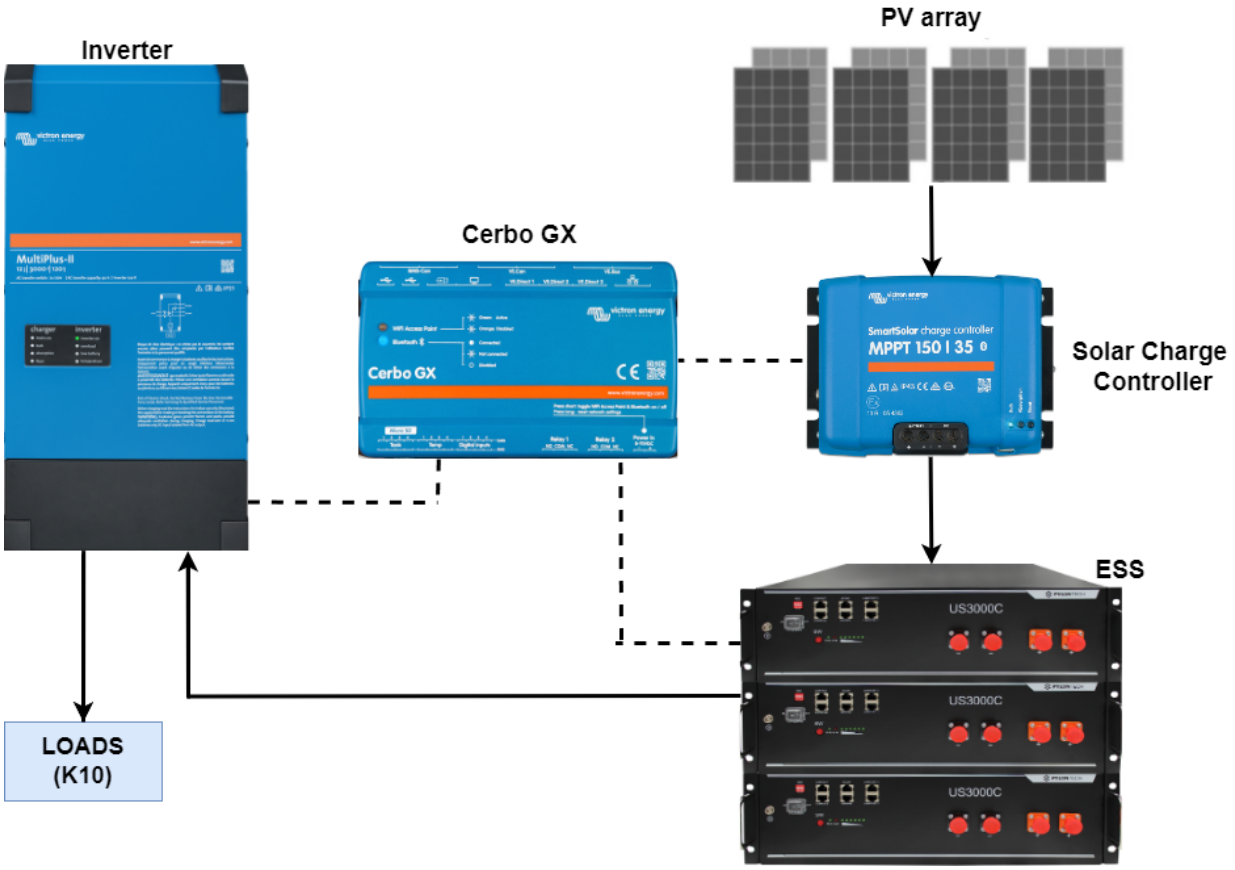


Figure 3.2: *PV and ESS diagram.*

The array of PV panels is the input to the MPPT solar charge controller, which gathers the energy generated from the panels and stores it into the ESS. The SmartSolar MPPT controller, which stands for Maximum Power Point Tracker, maximises the energy-harvest by balancing voltage and current to generate the most power from the PV panels. It then drives it efficiently to achieve full charge in the shortest possible time, maintaining battery health.

The 3000 VA MultiPlus-II PV inverter converts the batteries DC power to AC power, supplying the microgrid's loads. Even though the device is capable of providing 3 kVA, the implemented load control algorithm limits it to 80% of its capacity (2400 VA) to prevent it from overloading. Additionally, the inverter is responsible for providing a good energy quality to the MG, maintaining its frequency at 50 Hz, as well as its voltage at 230 V. Figure 3.2 is representative of the PV system implemented.

The Cerbo GX is the center of the PV installation as all the other components are connected to it, therefore monitoring and controlling the whole system. The device provides a RJ45 port for Ethernet connection to the local network, and various ports for specific Victron communication protocols: VE.Bus for communication with the inverter, VE.Direct for the charge controller and BMS.Can for the ESS. As a result, all the relevant information pertaining to the photovoltaic system, including batteries' SOC and inverter output power, is monitored within the Cerbo GX and accessible through its interface (Figure 3.3) by typing the module's IP address in any browser, while being connected to the ISR network.

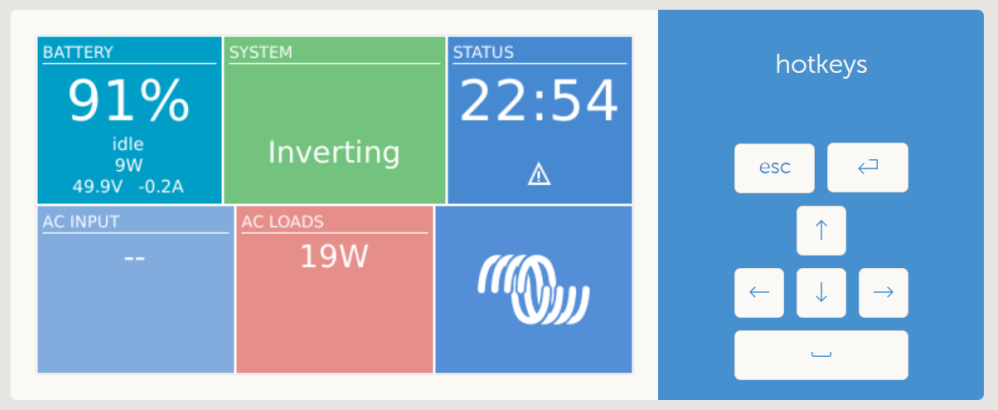


Figure 3.3: *Cerbo GX interface - Main menu.*

In Figure 3.4 the Cerbo GX device list is presented. In this case, the devices that show up are the inverter (MultiPlus-II) and ESS (Pylontech battery).

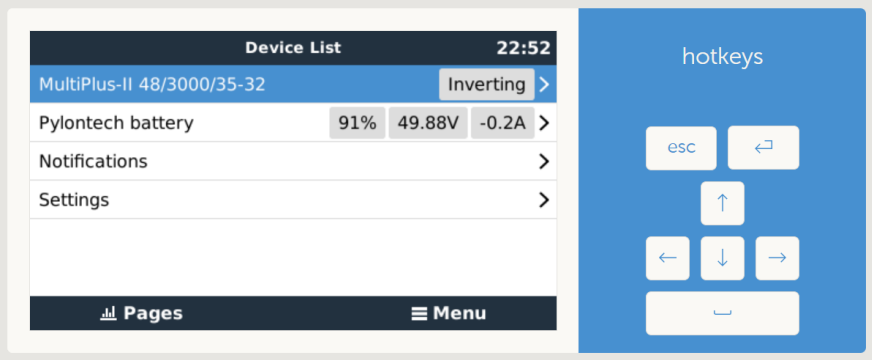


Figure 3.4: *Cerbo GX interface - Device List.*

Figure 3.5 presents the same devices from above with their respective Unit ID, which is an ID used in the Modbus TCP protocol to identify each slave's address. These ID's are used in the EcoStruxure Machine Expert programming software to set up the communication between the PLC (master) and its slaves.

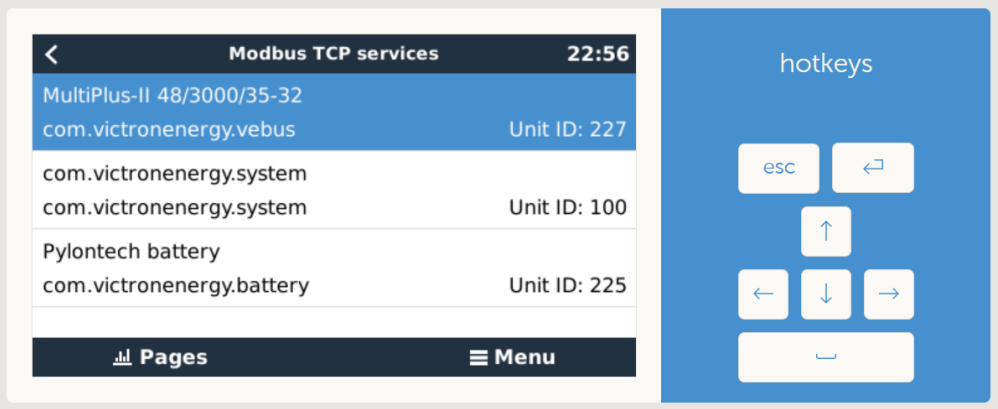


Figure 3.5: *Cerbo GX interface - Modbus TCP services.*

3.2.9 Local Network Cabinet

A local ISR computer network switch is installed in a cabinet in DEEC's garage, where the automaton and the Cerbo GX communication module are connected to. This configuration enables a wireless connection to both devices through the ISR computer network.

3.2.10 PLC Output Relays

Output relays were connected to the PLC's 24V fast transistor outputs. These relays serve to control the contactors operating at 230V. The output relays act as intermediaries, allowing the low voltage signals from the PLC to safely and effectively actuate the high voltage contactors. This setup ensures that the PLC can manage high voltage components without direct exposure to high voltage levels, thereby enhancing both safety and reliability of the control system.

3.3 Electrical Installation

Figure 3.6 shows the on-site installation of the implemented microgrid. Each device/component is identified below.

Figure 3.6: *Photo of the On-Site Installation.*

1. Victron MultiPlus II 3000VA Inverter/Charger;
2. Victron SmartSolar MPPT 150/35 Charge Regulator;
3. Victron Cerbo GX;
4. Network Cabinet;
5. New board panel with PLC, Contactors, Energy Monitors;
6. Pylontech US3000C Battery, 10.5kWh;
7. Fuse Boxes;
8. UPS;
9. Updated original board panel of the garage.

Figure 3.7 presents the electrical board panels. The updated original board panel of the garage contains contactors 1 and 9, as well as the light circuits breakers, the grid and load manual switches. The new panel has contactors 2 to 8 and 10 along with the PLC. Additionally, it also includes the Carlo Gavazzi energy analysers, fuses and finally the needed relays. These devices are all displayed in the block diagram of the microgrid (Figure 3.1).

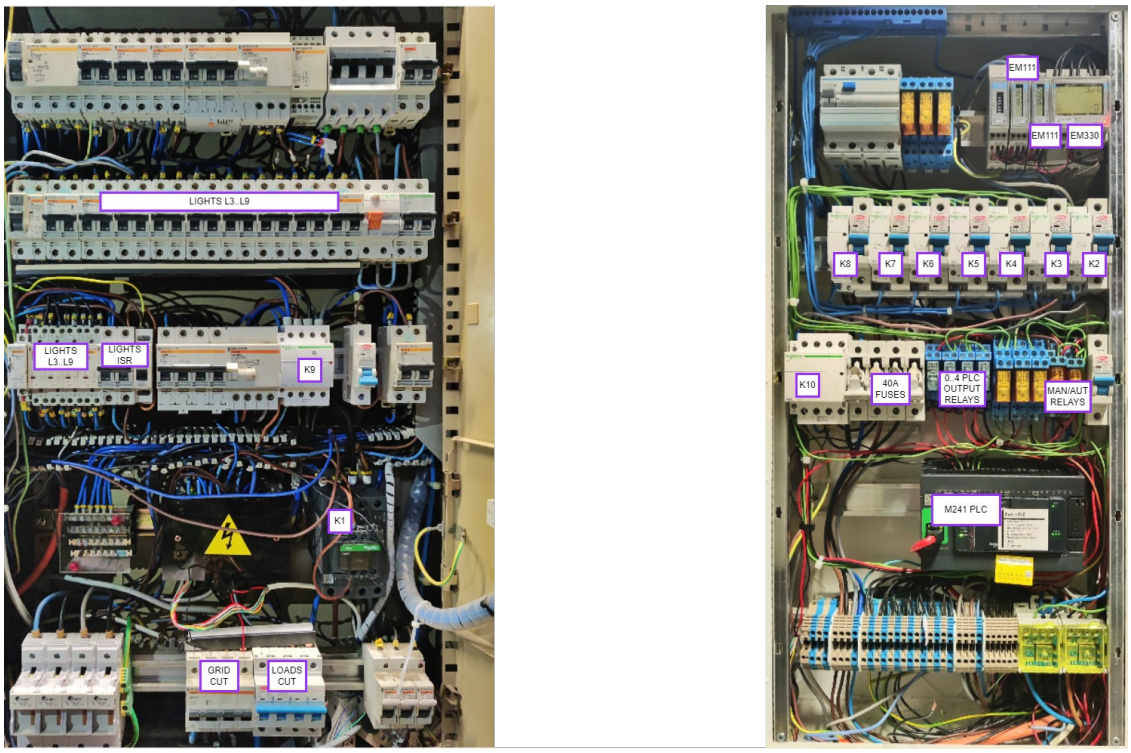


Figure 3.7: *Electrical panels:* ***(a)*** *Updated original panel of the garage.* ***(b)*** *New electrical panel.*

3.4 Load Profiles and Priority Levels

Table 3.1 presents the priority level of the different circuits. This classification is used to decide whether and when to shed the different loads. The priority level assigned to each contactor, from 1 (higher priority) to 7 (lower priority), depends on the power drawn from the load of each contactor and the relevance of that load to the users. When there is a need to shed loads, the power is firstly cut to the ones with a lower priority. The table also identifies the different circuits, their respective contactor, absorbed power and priority level. Contactors 1 and 10 are not on the table because they are only used to manage the transition between modes of the MG. Contactor 9 is also not present as it is just used to cut off the three-phase loads for the MG to operate in a single-phase in island mode.

Contactor	2	6	5	3	8	4	7
Description	Socket Circuit 1	Lighting Circuit L5, L7	Lighting Circuit L9	Socket Circuit 2	Lighting Circuit ISR	Lighting Circuit L3, L4	Lightning Circuit L6, L8
Power (W)	0-2400	1150	230	0-2400	230	1150	920
Priority Level	1	2	3	4	5	6	7

Table 3.1: Load Priority Level.

However, there are a couple loads that are not controlled by the algorithm because they are considered critical, and will always be supplied as long as there is enough energy in the batteries. Those are the garage gate, minimum safety lighting (L1 and L2), the PLC, the network cabinet and the UPS's. These circuits have a higher priority than all the others and can consume up to 600 W in total. Figure 3.8 displays the garage's circuits layout.

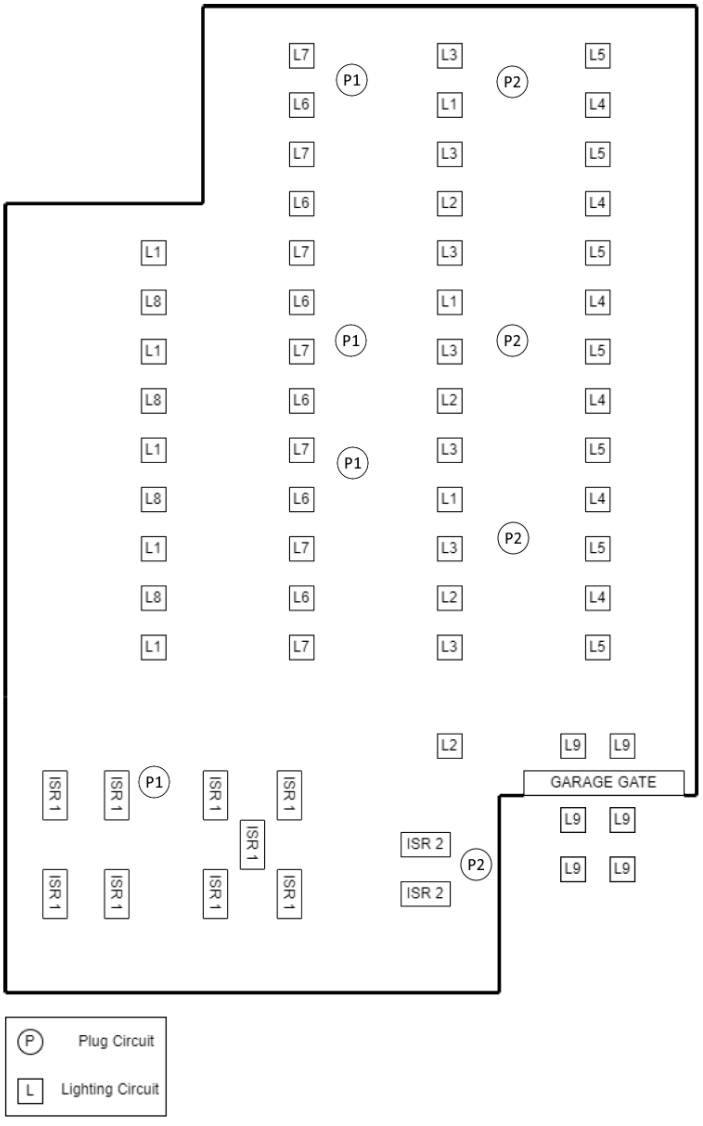


Figure 3.8: *Garage electrical circuit layout.*

3.5 Microgrid Operation

The microgrid typically operates in a grid-connected mode. Upon detecting a power outage from the main grid, it switches to islanded mode. It remains in islanded mode until the main grid power is restored, at which point it transitions back to grid-connected mode.

3.5.1 PO detection and transition to islanded mode

When a PO is detected, the MG transitions to islanded mode opening contactors 1 and 9 and then closing contactor 10 (Figure 3.1), connecting the PV inverter to the three phases of the busbar and making the MG operate as a single-phase system in island mode. It is, therefore, essential that contactor 1 and 10 are never closed at the same time. To ensure that happens, the algorithm always applies a 200 *ms* time delay between the switching of these contactors. This applies to both ways of the transition, from grid to island and vice-versa.

3.5.2 Islanded mode operation

When in island mode, the following variables are monitored: power being supplied by the PV inverter to the MG, batteries' SOC and time elapsed since the beginning of the outage. The maximum power that the inverter can supply at any instance is controlled and calculated constantly in islanded operation, based on the batteries' SOC. Based on that data, the load control algorithm decides which loads need to be shed and which can keep being supplied. Phase one of the islanded mode operation starts right after the transition to islanded mode: all power is supplied to the circuits for one minute without any battery energy management, limited only by 80% of the inverter's nominal power value. After one minute, phase two begins: the load control algorithm starts to manage the batteries' energy, establishing a maximum consumption limit within the MG and shedding the necessary loads to stay within this limit. The goal of this management is to keep the MG critical loads supplied for an hour, shedding the necessary non-critical loads according to the priority level of table 3.1, and reactivating them if there is enough power available. After an hour has passed, the algorithm begins its third and last stage, where the goal is to supply critical loads for ten hours continuously. The batteries' energy continues to be managed by capping the allowed power consumption and shedding the necessary loads.

3.5.3 Batteries critical SOC and transition to grid-connected mode

If at any moment the batteries' SOC falls below 20% and there still is a power outage, the loads are all shed, besides critical ones. Even so, if there is enough energy produced from the PV panels and SOC rises above 20%, the loads are gradually supplied again in order of their priority. When power returns from the utility grid, contactor 10 opens, then contactors 1 and 9 to complete the transition from islanded to grid-connected mode.

4 Microgrid Control Implementation

In this chapter, the load control algorithm and the MG's state machine are technically explained. The microgrid's monitoring panel is also presented.

4.1 Control Variables

In order to control the state of the MG, several control variables were created:

- **PO:** Boolean variable that changes depending on the output of the PO detector relay:

$$\mathbf{PO}: \begin{cases} 0, \text{if PO detector relay output is false (no power outage)} \\ 1, \text{if PO detector relay output is true (power outage)} \end{cases}$$

- **SOC_ESS:** Energy left in the ESS (%). If at any given time this variable drops below 20%, the algorithm goes to a state where only the critical loads are supplied.

- **T:** Time elapsed since the beginning of a power outage:

$$\mathbf{T}: \begin{cases} 0, \text{if } T < 1 \text{ min} \\ 1, \text{if } 1 \leq T < 60 \text{ min} \\ 2, \text{if } T \geq 60 \text{ min} \end{cases}$$

- **PMAX_VALUE:**

In islanded mode, the maximum power that can be supplied by the inverter to the MG is determined by the batteries' SOC and the value of 'T_VAR', as follows:

  - When T=0, 'PMAX_VALUE' is set to 2400 W.
  - When T=1:

    The first priority is to guarantee the needed energy to supply the critical loads until the end of the first hour. Taking into account that they can consume up to 600 W, the needed energy to supply them for an hour is 600 Wh. Regarding battery SOC percentage, that translates to:

$$Remain\_SOC\_CLoads\_1H = \frac{En\_Critical\_Loads\_1H[Wh]}{Max\_Bat\_Energy[Wh]} * 100 + \\ + Min\_SOC\_Bat[\%] \approx 26\%$$

    where $En\_Critical\_Loads\_1H$ is 600 Wh, Max_Bat_Energy is 10.5 kWh, and Min_SOC_Bat is 20%.

    Taking that value into account, there are only two option:

    * if the batteries' SOC is lower than 26%, then $PMAX\_VALUE = 600W$;
    * if the batteries' SOC is greater 26%, then the MG consumption is only limited by 80% the inverter's nominal power, so $PMAX\_VALUE = 2400W$.

  - When T=2:

    The same principle applies when T=1, but the needed energy to supply the critical loads is calculated for 10 hours, which represents 6000 Wh. Following the same formula used in the previous point, that translates to approximately 78% of the SOC. Therefore:

    * if the batteries' SOC is lower than 78%, then $PMAX\_VALUE = 600W$;
    * if the batteries' SOC is greater than 78%, then $PMAX\_VALUE = 2400W$.

'PMAX_VALUE' is the reference to the load shedding operation, meaning that if the power being consumed in the MG is greater than 'PMAX_VALUE', the contactors open according to the loads priority (from lower to higher) until the overall consumption decreases below the reference value ('PMAX_VALUE').

4.2 Control Algorithm and State Machine

In this section is explained how the load control algorithm and state machine were implemented in the PLC software.

4.2.1 Algorithm Operation

4.2.1.1 Global Variables

The developed algorithm uses a large list of global variables (Listing 4.1) that can be used in any of the functions and programs developed, without the need of being declared inside these. Global variables are declared in a specific file named 'GVL'. Some of the most important global variables are 'PO', which tracks if there is a PO from the main grid, 'DELTA_PO' that tracks the time elapsed since a PO as occurred, as well as 'PMAX_VALUE' which is the maximum power the MG can consume at any instance during islanded mode.

```
{attribute 'qualified_only'}
VAR_GLOBAL

  TRACE: BOOL := FALSE;

  PO: INT; // variavel POWER OUTAGE utilizada para controlo na maquina de estados (0, 1)
  T: INT; // variavel TIME SINCE POWER OUTAGE utilizada para controlo na maquina de estados
    (0, 1 ou 2)
  DELTA_PO: ULINT; // tempo passado desde a power outage
  PMAX_VALUE: INT; // potencia maxima disponivel as cargas
  PMAX_VALUE_WO_CLOADS: INT; // potencia maxima disponivel as cargas, exceto criticas

  ACTUAL_CONSUME: INT; // consumo atual das cargas
  ACTUAL_CONSUME_W_CLOADS: INT; // consumo atual das cargas nao criticas
  EM1_1: INT; // tensao FN 1 EM1 (energy monitor 1)
  EM1_2: INT; // tensao FN 2 EM1
  EM1_3: INT; // tensao FN 3 EM1
  VMIN: INT := 190; // tensao minima nas fases da rede

  EM2_POWER: INT; // potencia em watts EM2
  EM3_POWER: INT; // potencia em watts EM3

  INV_POWER: INT; // potencia real aparente a ser consumida pela microrrede (VA)
  MAX_BATTERY_CAPACITY: INT := 10500; // energia maxima das baterias do PV (Wh)
  SOC_ESS: INT := 100; // % atual das baterias do PV
  MIN_SOC_ESS: INT := 20; // % minima permitida para funcionamento normal
  MAX_INVERTER_POWER: INT := 2400;  // potencia maxima permitida de output do inversor

  C4_VALUE: INT := 1150; // consumo dos light circuits do contactor 4
  C5_VALUE: INT := 230; // consumo dos light circuits do contactor 5
```

```
  C6_VALUE: INT := 1150; // consumo dos light circuits do contactor 6
  C7_VALUE: INT := 920; // consumo dos light circuits do contactor 7
  C8_VALUE: INT := 230; // consumo dos light circuits do contactor 8
  CLOADS: INT := 600; // consumo das cargas criticas, nao controlaveis

  PO_TIME: ULINT; //guarda timestamp da PO
  TIMER_2MIN: ULINT;
  TRANSITION_SECURITY: ULINT;

  MIN_1_MS: ULINT := 60000;
  HORA_1_MS: ULINT := 3600000;
END_VAR
```

Listing 4.1: *Global Variables declaration block.*

4.2.1.2 Tasks

A task is used to call a program. The algorithm has two tasks running simultaneously: Master and Read_Values.

The Master task calls the *Main* program where the state machine is coded, which will be explained in section 4.2.3, and runs in freewheel. This means that it does not have a predefined time to run. It is always running and has priority over the other task.

The Read_Values task is a cyclic task that runs with a rate of 50 *ms* and calls the *Read_Values* program. This program iteratively calls functions to update the control variables with the values that the corresponding functions return. It has the following code:

```
PROGRAM Read_Values //chama funcoes iterativamente para atribuir as variaveis globais os
    valores de cada funcao correspondente
VAR
END_VAR
```

Listing 4.2: *Read_Values program declaration block.*

```
GVL.PO := PO_VAR();
GVL.T := T_VAR();
GVL.PMAX_VALUE := CALC_PMAX();
GVL.INV_POWER := LER_INV_POWER(); //inverter output power
GVL.SOC_ESS := LER_SOC_ESS(); //batteries SOC
GVL.EM2_POWER := LER_CG2(); //socket circuit 1 power consumption
GVL.EM3_POWER := LER_CG3(); //socket circuit 2 power consumption
```

Listing 4.3: *Read_Values program code.*

In Structured Text language each program or function has two code blocks, one for declaring local variables used in that same function/program, and another to write the code itself.

4.2.1.3 Modbus RTU Configuration to Communicate with Energy Monitors

Modbus RTU protocol is used for the communication between the PLC and the Carlo Gavazzi energy monitors, following the energy monitors manual. These monitors are connected to the automaton through the RS485 serial line port. This procedure is explained in the Carlo Gavazzi energy analyser manual available in appendix .1.4.

The first step is to configure the serial line port correctly. On the 'Devices Tree' tab on the left-hand side of the software environment, the 'Serial_Line_2' port is available and presents some parameters to configure (Figure 4.1). The energy monitors also have these variables to configure and they need to be set to the same value as the serial line port 2 for the communication to be successful. The values of the parameters can be changed in the monitors themselves. For instructions on this, please see appendix .1.4.

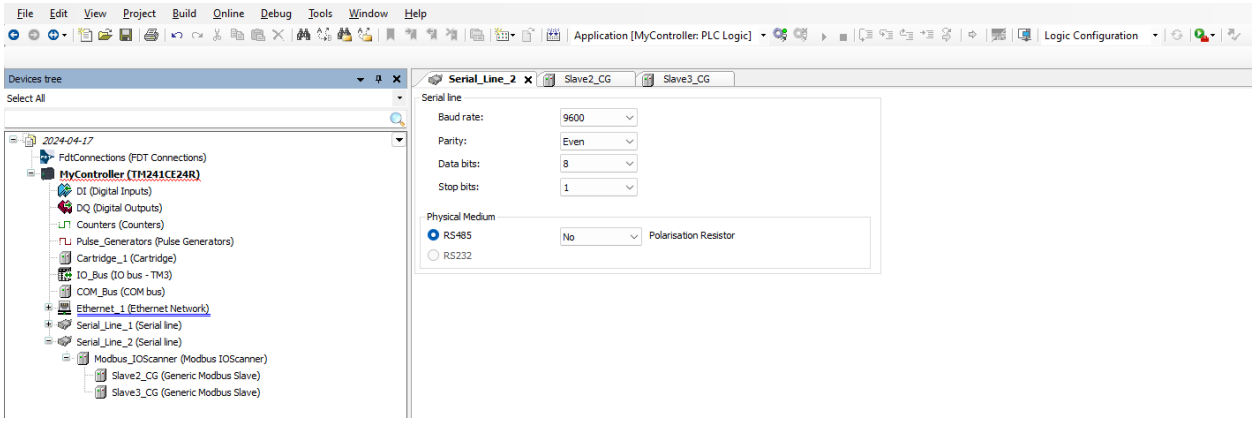


Figure 4.1: *Serial Line 2 configuration ('Devices Tree' tab).*

In the software, a 'Modbus_IOScanner' device is added to the serial line port 2 to set up the type of Modbus protocol (Figure 4.2). In this MG is used Modbus RTU.

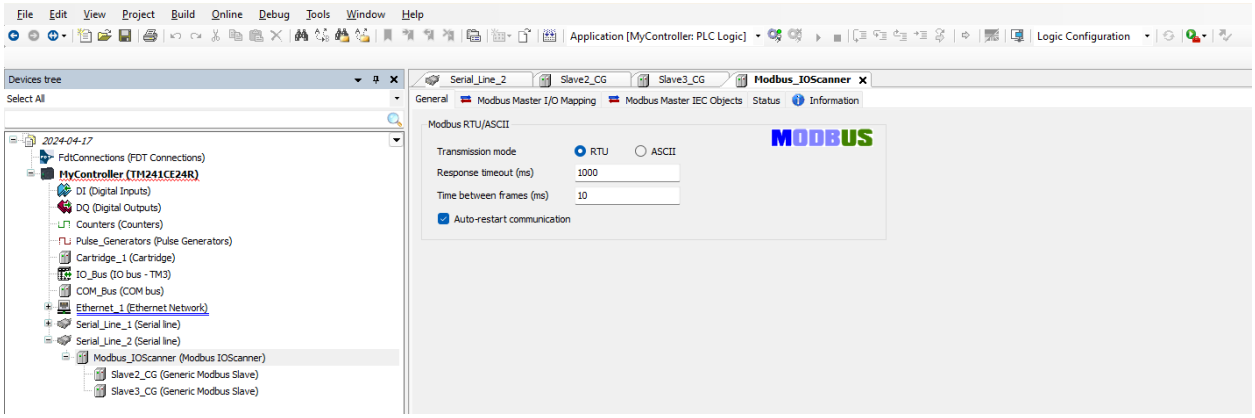


Figure 4.2: *Modbus I/O Scanner configuration.*

The Modbus slaves, the energy monitors, are added to the Modbus I/O Scanner in the Devices Tree. 'Slave2_CG' monitors sockets' circuit 1 and was given address 2 while 'Slave3_CG' monitors sockets' circuit 2 and was given address 1. This address number is configured both in the 'General' tab of the slave devices in the software as seen in Figure 4.3, and in the energy analysers menu.

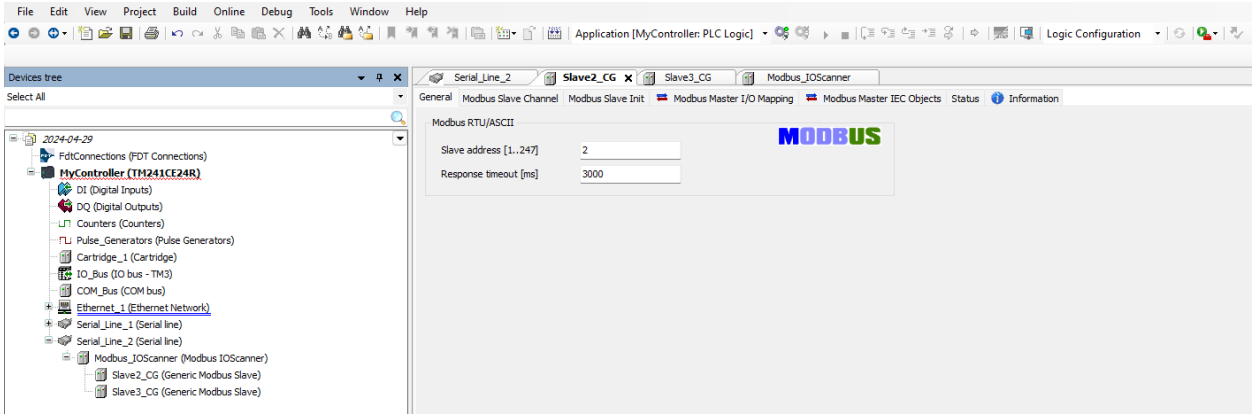


Figure 4.3: *Slave2_CG address configuration.*

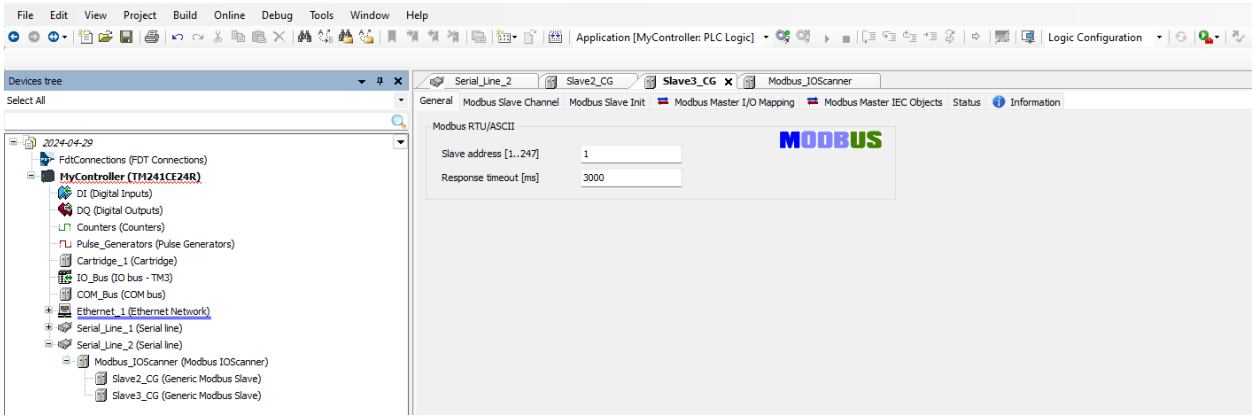


Figure 4.4: *Slave3_CG address configuration.*

In the 'Modbus Slave Channel' tab of the slave devices, the communication channels are configured (Figure 4.5). These are used to establish a communication between the slave and the master to enable the master to send requests to a slave device, that replies with the data requested.

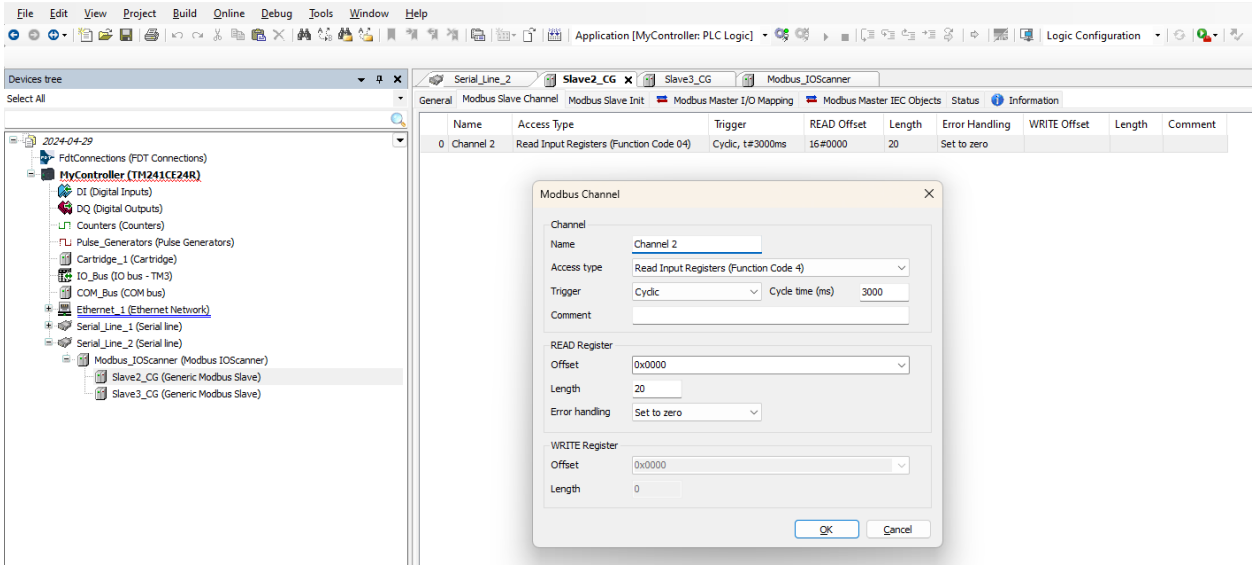


Figure 4.5: *Modbus RTU slave channel configuration.*

The access type defines the function of the channel, either read registers, write registers or both. Registers store the data exchanged between devices. In this situation the goal is to read registers, so code 4 is selected. In the 'READ Register' section the offset is standard and the length is set to 20, which is the number of registers the channel transmits. The same set up is done to Slave 3.

After setting up the communication, the values of the registers can be read in the 'Modbus Master I/O Mapping' presented in Figure 4.6 as well as their respective addresses (%IW32...%IW51). These are automatically attributed by the software and are then allocated to variables in functions to call the value intended to be used. An example is provided in Figure 4.6 that displays the different values read by the energy monitor of sockets' circuit 1. %IW13 corresponds to voltage (V), %IW15 to current (A) and %IW17 to power (W). It is important to note that these values have different scale factors. The voltage of the circuit (register %IW13) is 2294 which represents 229.4 V because of the scale factor of 10. The same applies to power which is 50, 5.0 W in reality. However, the current has a scale factor of 1000, so 2060 is in fact 2.060 A. These scale factors are taken into account in the functions that call these values.

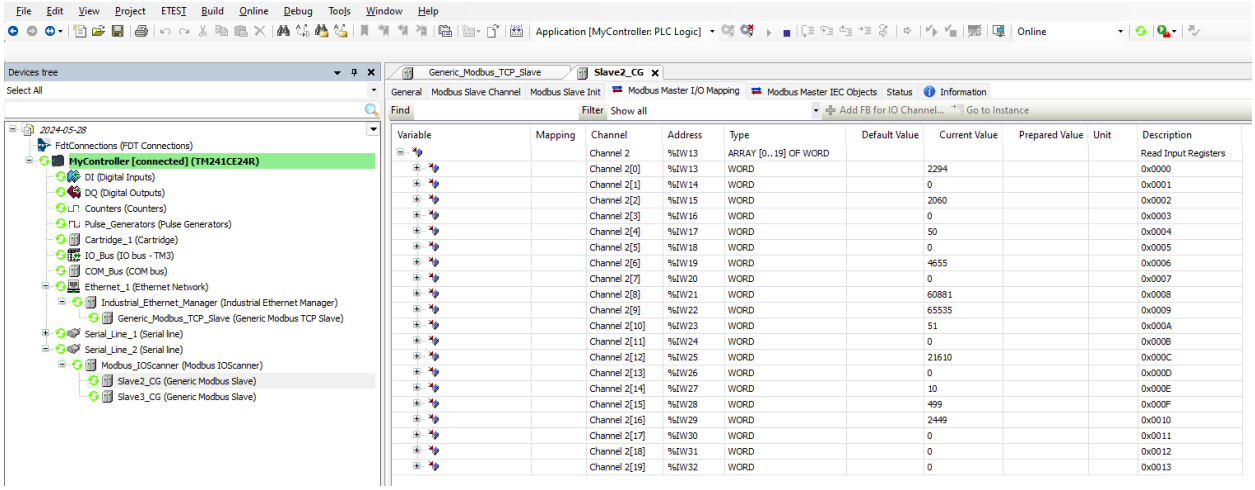


Figure 4.6: *Modbus RTU registers values.*

4.2.1.4 Modbus TCP Configuration to Communicate with Cerbo GX

Modbus TCP is used for the communication between the Cerbo GX and the PLC. This protocol does not imply both the devices to be physically directly connected, but they have to be in the same network.

The communication is firstly set up in the 'Devices Tree' tab on the left-hand side of the software environment, under 'Ethernet_1', adding an 'Industrial_Ethernet_Manager' device, to which is added a 'Generic_Modbus_TCP_Slave' device. Inside the latter's configuration file, in the first tab the slave's IP address is defined (Figure 4.7).

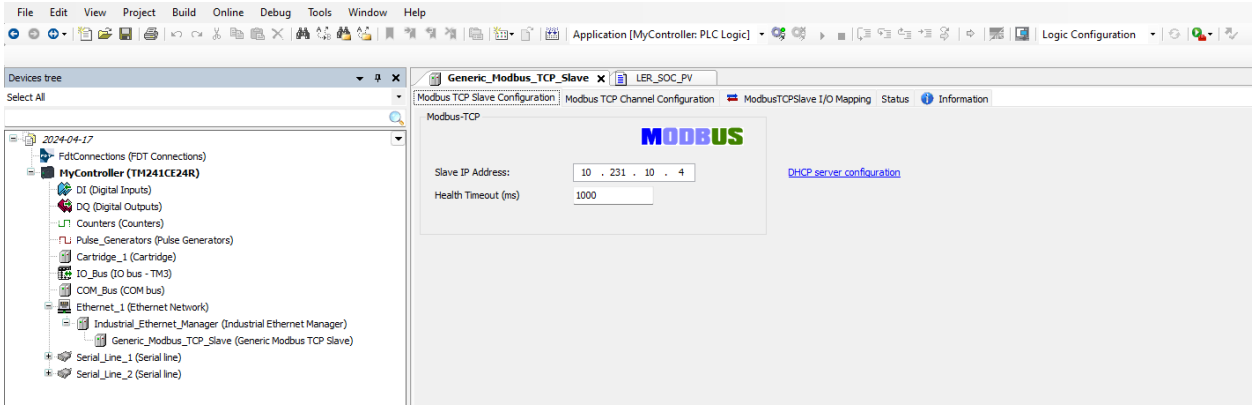


Figure 4.7: *Modbus TCP Slave configuration.*

In the second tab are configured the communication channels. These are used to establish a communication between the slave and the master to enable the master to send requests to a slave device, that replies with the data requested. There are two channels configured: 'PV Inverter Output Current' sends out the value of the inverter output current, and 'SOC ESS' sends the batteries' SOC. The latter is configured in figure 4.8.

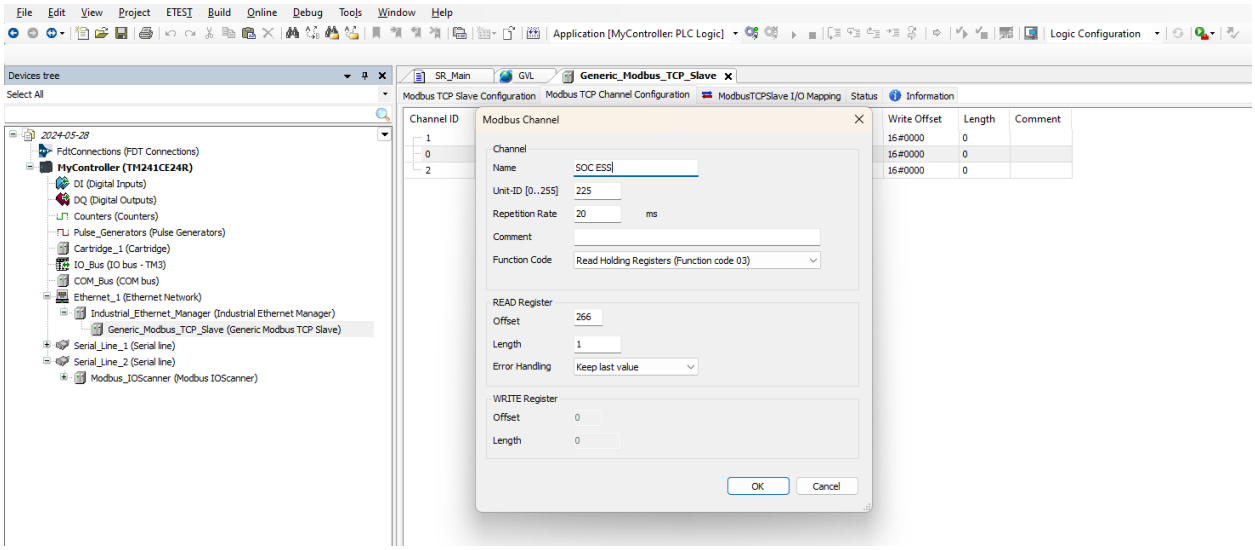


Figure 4.8: *Modbus TCP 'SOC ESS' channel configuration.*

The Unit ID 225 corresponds to the Pylontech battery ID displayed in figure 3.5 and the information is sent through the channel each 20 ms. The function code defines the purpose of the channel, either read registers, write registers or both. Registers store the data exchanged between devices. In this situation the goal is to read registers, so code 03 is selected. Concerning the 'READ Register' parameters, 'Offset' is where the address of the register to be read is defined. Victron Energy provides a Modbus TCP register Excel list on their website that indicates the address of the register with the data to be read. For the URL please see appendix .1.3. The length defines how many registers are requested in a channel.

Figure 4.9 presents the 'PV Inverter Output Current' channel configuration. The Unit ID 227 corresponds to the PV system ID displayed in Figure 3.5 and the registers are requested each 20 ms. The function code is the same used in the 'SOC ESS' channel and the offset 266 is defined following the Victron Modbus TCP register list at the link indicated in appendix .1.3.

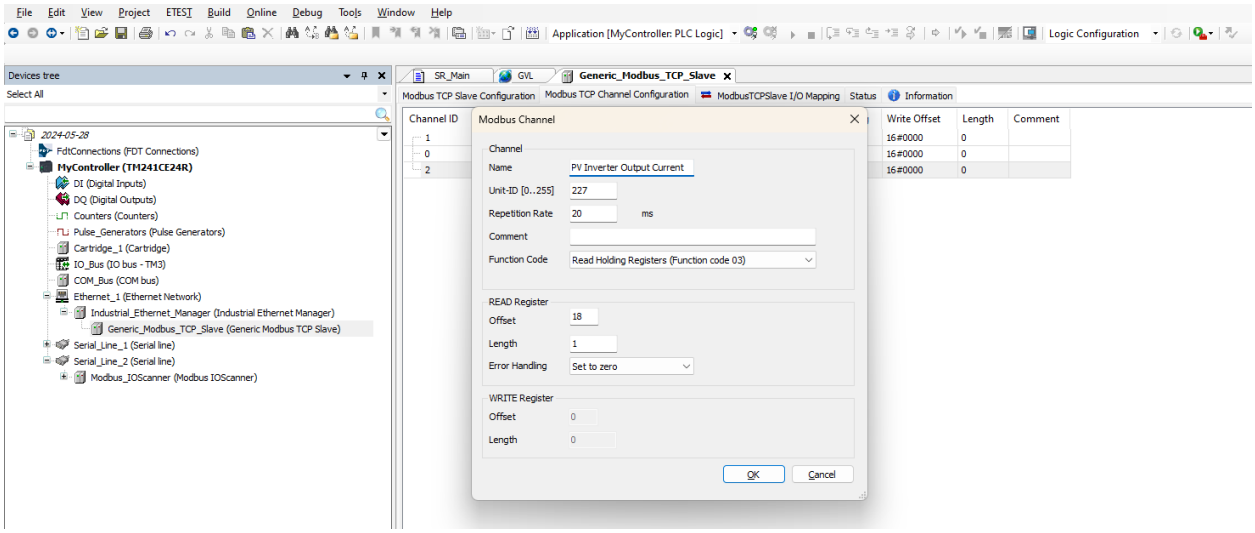


Figure 4.9: *Modbus TCP 'PV Inverter Output Current' channel configuration.*

In the 'Modbus TCP Slave I/O Mapping' tab of the device's configuration, the different registers are presented with their corresponding values and addresses (Figure 4.10). It is important to understand that these addresses are automatically attributed to the registers by the software so that they can be called within a function. These are not the same as the registers addresses. Address %IW12 corresponds to the output current (A) of the inverter and %IW11 to the batteries SOC in percentage. Both registers have a scale factor of 10 so, in reality, the SOC value 990 is 99% and the inverter's output current value of 2 is 0.2 A.

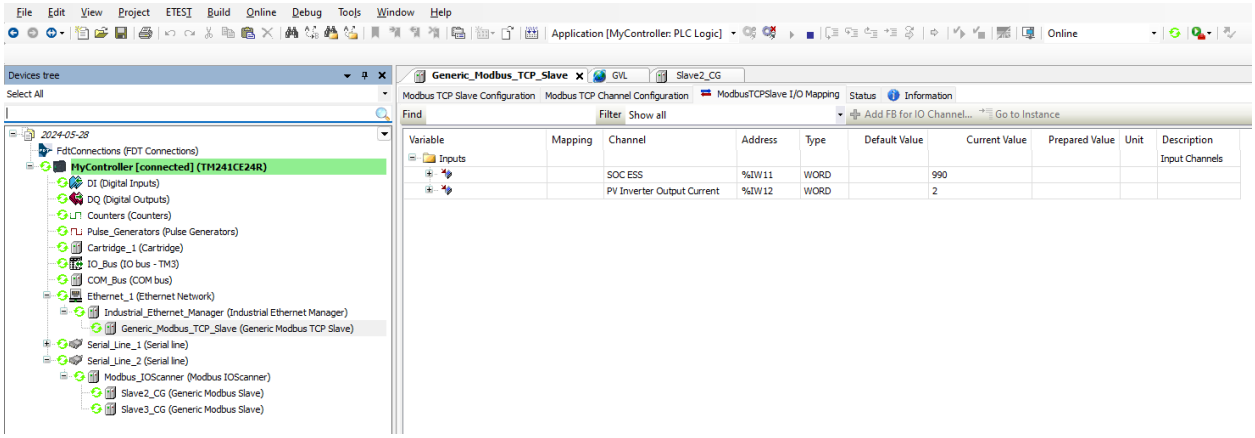


Figure 4.10: *Modbus TCP channel registers.*

4.2.1.5 Digital Inputs/Outputs

The automaton provides 14 digital inputs. The only input connected to the PLC is the PO detector relay defined as *RELE_REDE*. This is configured in the 'Devices Tree' tab on the left, in the 'Digital Inputs' file. The relay is mapped in the first input of the PLC presented on the mapping table in figure 4.11.

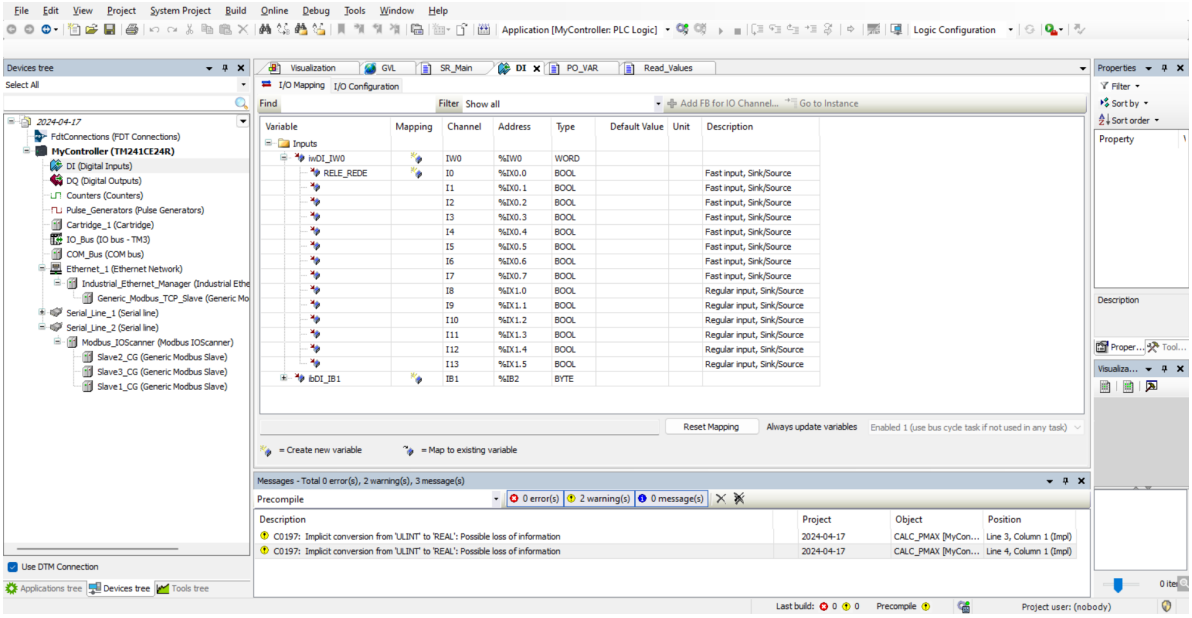


Figure 4.11: *Digital inputs mapping of the PLC.*

The outputs are 10 in total, 4 of them being fast transistor outputs. The outputs correspond to contactors 1 to 10 and are defined in the 'Devices Tree' tab on the left, in the 'Digital Outputs' file (Figure 4.12).

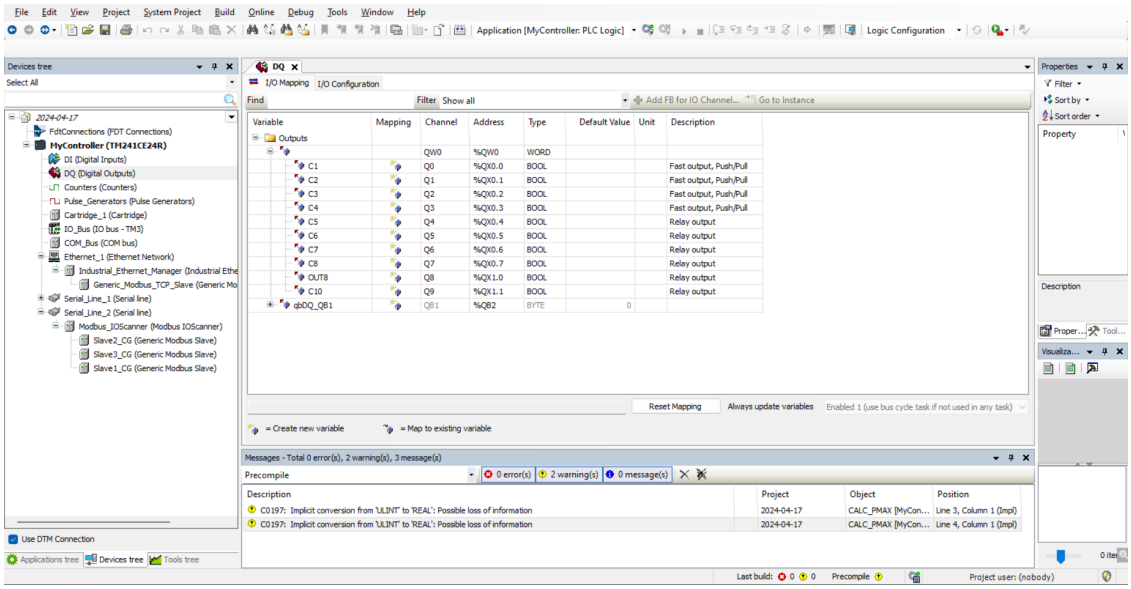


Figure 4.12: *Digital outputs mapping.*

4.2.2 Functions of the algorithm

In appendix .1.1 all 7 functions developed are explained in the order as they appear on the *Read_Values* program (Listing 4.3). The *Read_Values* program calls these functions each 50 ms to update global control variables.

4.2.3 State Machine

Figure 4.13 presents the state machine diagram. The state machine is composed of 6 states: state 0, 10, 20, 30, 40 and 45. Inside the software, the state machine code is developed in the *Main* program. For the explanation and presentation of the developed code for each state, please see appendix .1.2. The state machine operates as follows:

- **State 10 - Grid-connected Mode:** This is the usual state of the algorithm, when there is power coming from the utility grid, which means contactor 1 (main grid) is closed and 10 (inverter) is open. In this state, every garage circuit is supplied by the main grid and the algorithm is just waiting for a power outage to occur;

- **State 45 - Transition and Control:** Central state of the algorithm for 2 reasons: handles the 200 ms transition from grid to island mode and vice-versa. Both contactors 1 and 10 are open. It decides to which state the algorithm has to go next depending on the control variables. When a PO is detected, the state machine goes from state 10 to 45. After the 200 ms transition, the state machine goes to state 20;

- **State 20 - Maximum charges supplied for one minute:** First island mode state and has the goal of supplying every load possible for the first minute since the PO. In state 20, the batteries' energy is not managed, therefore the maximum power available from the inverter is only limited to its nominal power. During a PO the next state is 30. If the main grid power is restored, the next state is 45;

- **State 30 - Maximum supply for the MG to last 1 hour:** The batteries' energy starts to get managed by capping the power the inverter can supply to the MG. The maximum power available (PMAX_VALUE) depends on the energy left in the batteries. The objective of this state is to guarantee minimal battery SOC to supply the critical loads for an hour. Whenever MG consumption is greater than 'PMAX_VALUE', the controllable loads are shed until consumption is lower than the maximum allowed. Afterwards, if there is enough energy in the batteries, the shed

loads are resupplied only if their consumption added to the current MG consumption is lower than 'PMAX_VALUE'. During a PO the nxt state is 40 (one hour after PO start). If the main grid power is restored, the next state is 45;

- **State 40 - Maximum supply for the MG to last 10 hours:** Operates in the same way as state 30, but manages the energy left in the batteries in order for the critical loads to be supplied for the next 10 hours continuously. If the main grid power is restored, the next state is 45. If not, the state holds until SOC lowers from 20%;

- **State 0 - Battery SOC below 20%:** There is less then 20% energy capacity in the batteries in a PO condition. In this state only the critical non-controllable loads are supplied, thus every contactor is open except for contactor 10 that is closed If the main grid power is restored or the SOC increases above 20%, the next state is 45. This is the initial state of the state machine (in a reset of the PLC).

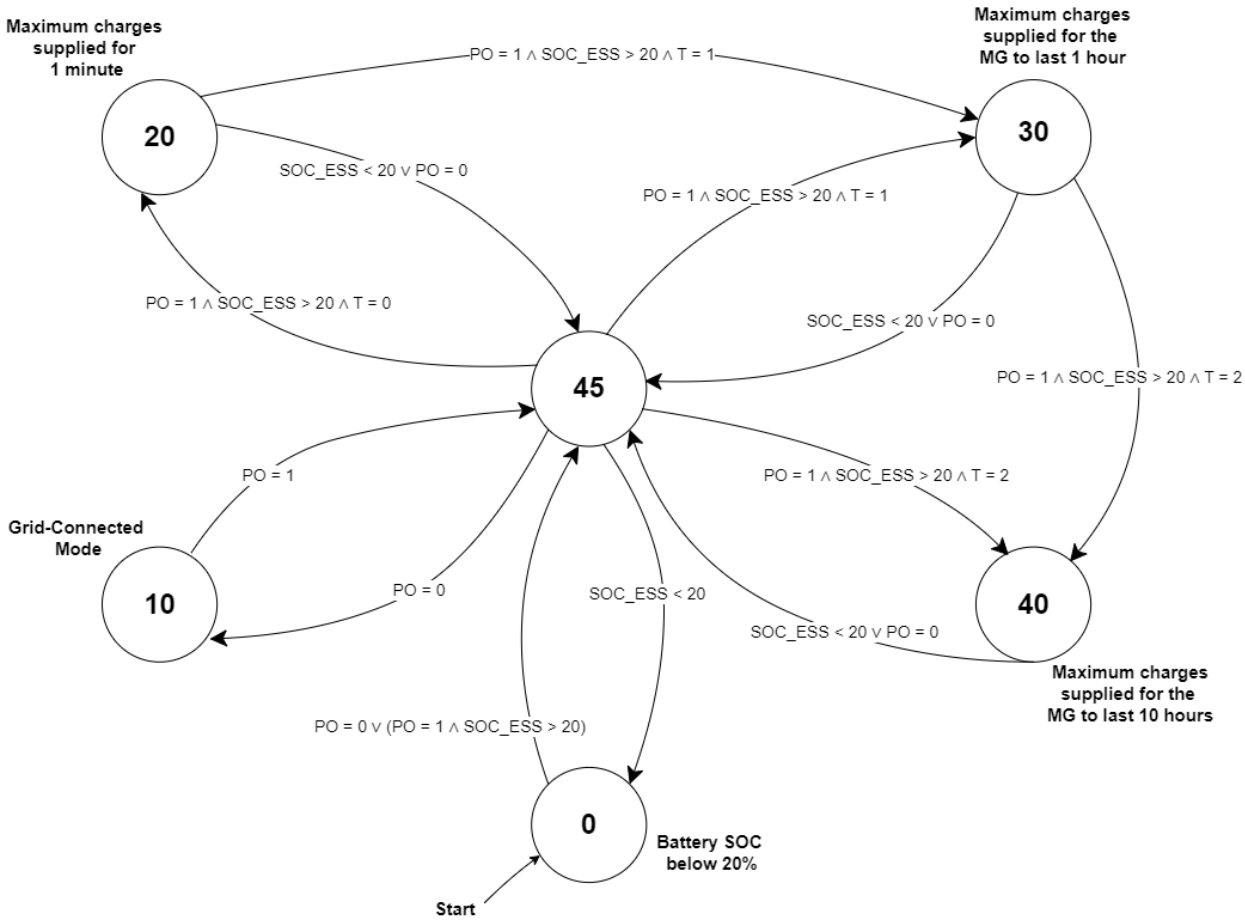


Figure 4.13: *State Machine Diagram.*

4.2.4 Old State Machine

During the work development, a different state machine was deployed which had a different operation of states 30 and 40, specifically the calculation of 'PMAX_VALUE' and the reactivation of contactors. Previously, the control variable 'PMAX_VALUE' was determined by the batteries' SOC and the time left to the end of each state, and was calculated in the following way:

Firstly, the remaining battery percentage until 20% was calculated:

$$REMAIN\_SOC\_ESS = (SOC\_ESS[\%] - 20[\%]) \tag{4.1}$$

Converting the percentage to energy in $W.h$,

$$REMAIN\_EN\_PV = 10500[Wh] * (REMAIN\_SOC\_ESS/100) \tag{4.2}$$

the value of $PMAX$ was:

$$PMAX = \frac{REMAIN\_EN\_PV}{n - REMAIN\_T\_TOnH} \tag{4.3}$$

$REMAIN\_T\_TOnH$ being the time (in hours) remaining for the PO to last n hours.

This mode of operation was scrapped because it didn't allow the connection of high power consuming loads in cases where the energy left in the batteries was enough, not only to supply that load but also the critical ones until the end of the state. Therefore, the state machine was updated to what is described in section 4.2.3, and the calculation of 'PMAX_VALUE' was changed according to section 4.1.

The contactors reactivation procedure was also changed. Previously, a contactor would be reactivated if the MG consumption was lower than 'PMAX_VALUE', even if reactivating that contactor meant surpassing the maximum power available. Consequently, the contactor would be immediately opened. This procedure resulted in constant fluctuations of the MG consumption and, as result, permanent oscillation in MG voltage and frequency.

4.3 Microgrid Monitoring Panel

To follow the MG's behavior throughout its different states, a monitoring panel was developed (Figure 4.14). This panel displays the values of inputs, including energy monitors' readings, inverter power and battery SOC, on the bottom left. On the bottom right the values of the control variables are shown. At the top, every contactor is represented with a lamp that switches on and off if the contactors are closed or open, respectively.

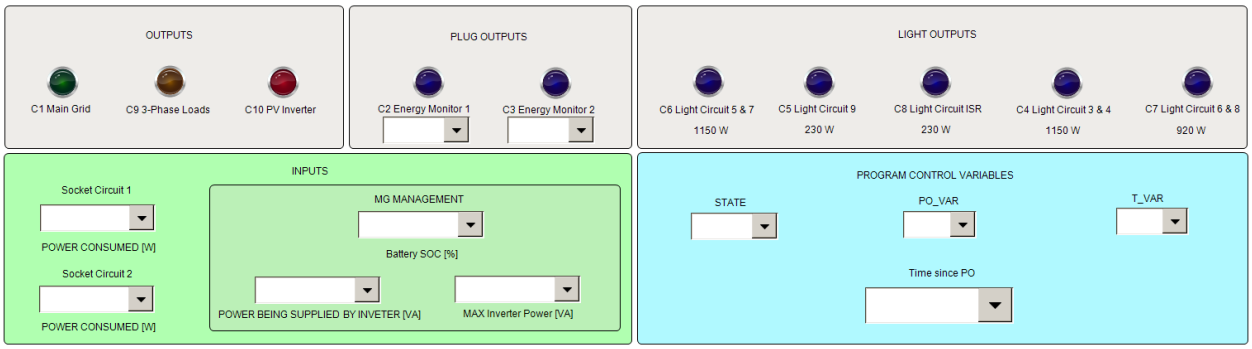


Figure 4.14: *MG Monitoring Panel.*

The panel was developed inside the EcoStruxure Machine Expert software. To visualise this panel, the computer must be connected to the PLC through the ISR network.

5 Results and Discussion

In this chapter the system testing results are presented and analysed.

5.1 Testing Results

The MG was tested in several different power outage scenarios and all the results translate the on-site real MG behaviour. The testing results were recorded using the Machine Expert software trace functionality, and a Fluke 1738 Three-Phase Power Quality Logger connected to 'the loads cut' circuit breaker on the updated original panel shown in Figure 3.7. This device captures voltage, current, power, harmonics and other associated power quality values. To simulate a PO, the 'grid cut' circuit breaker is used (see Figure 3.7).

5.1.1 Transition between grid-connected and islanded mode

The goal of this test is to demonstrate the MG's ability to transition between grid-connected and islanded mode, as well as the consequences of the transition.

Figure 5.1 displays the voltage and frequency charts of the MG. A sudden drop of the frequency is noticeable around the 11 seconds mark at the transition from grid-connected to islanded mode. The MG is able to detect the power failure and transition successfully. Before the transition, it is possible to observe the local electrical grid's fair power quality, mainly different voltage values on each phase and mildly variable frequency. After transitioning to islanded mode, there is a clear improvement of the electrical quality, as the inverter is able to maintain the voltage and frequency at their standard values. Minor disturbances can be observed at the 13 and 40 seconds mark, which are a result of load increases. At the 52 seconds mark, the MG transitions back to on-grid mode.

At the points of transition there is a drastic change in voltage and frequency, but the MG handles these variations successfully. This test highlights the V/f control capabilities of the inverter when the MG is in islanded mode, and in-between mode transitions.

Figure 5.1: *Voltage and frequency variation during transition between grid and island mode.*

5.1.2 Reaction to increased load

When a MG is operating independently from the power grid, even small loads can impact power stability and operation. This is because the MG's generation capacity is limited, and any changes in load can directly affect the balance between generation and consumption.

Figure 5.2 shows the voltage and frequency plots of the MG, along with the inverter's output power and GVL.PMAX_VALUE plots. This experiment consisted of simulating a power outage while increasing the load, with the aim of analysing how the voltage and frequency change according to a load increase in islanded mode. At around the 20 seconds mark a PO occured and all the contactors are closed, but the consumption surpasses the maximum allowed so contactor 4 is opened, followed by 7. At the sime time, the garage gate is opening and stops at around the 50 seconds mark. After 1 minute, a 1000 W load is connected to sockets' circuit 2 and disconnected after 10 seconds. A few seconds later, a second load of 1500 W is connected to the same circuit, which surpasses the maximum consumption and the algorithm opens contactor 8, followed by contactor 3. The load is then disconnected. At 2.9 minutes the garage gate starts closing and once it finishes, contactors 3, 8 and 7 are closed in this order. The fluctuations at 3.5 minutes are misreadings from the PLC of the inverter's power supply, that disappear after a few seconds. At 4.5 minutes the power from the main grid returns and the MG transitions to grid-connected mode.

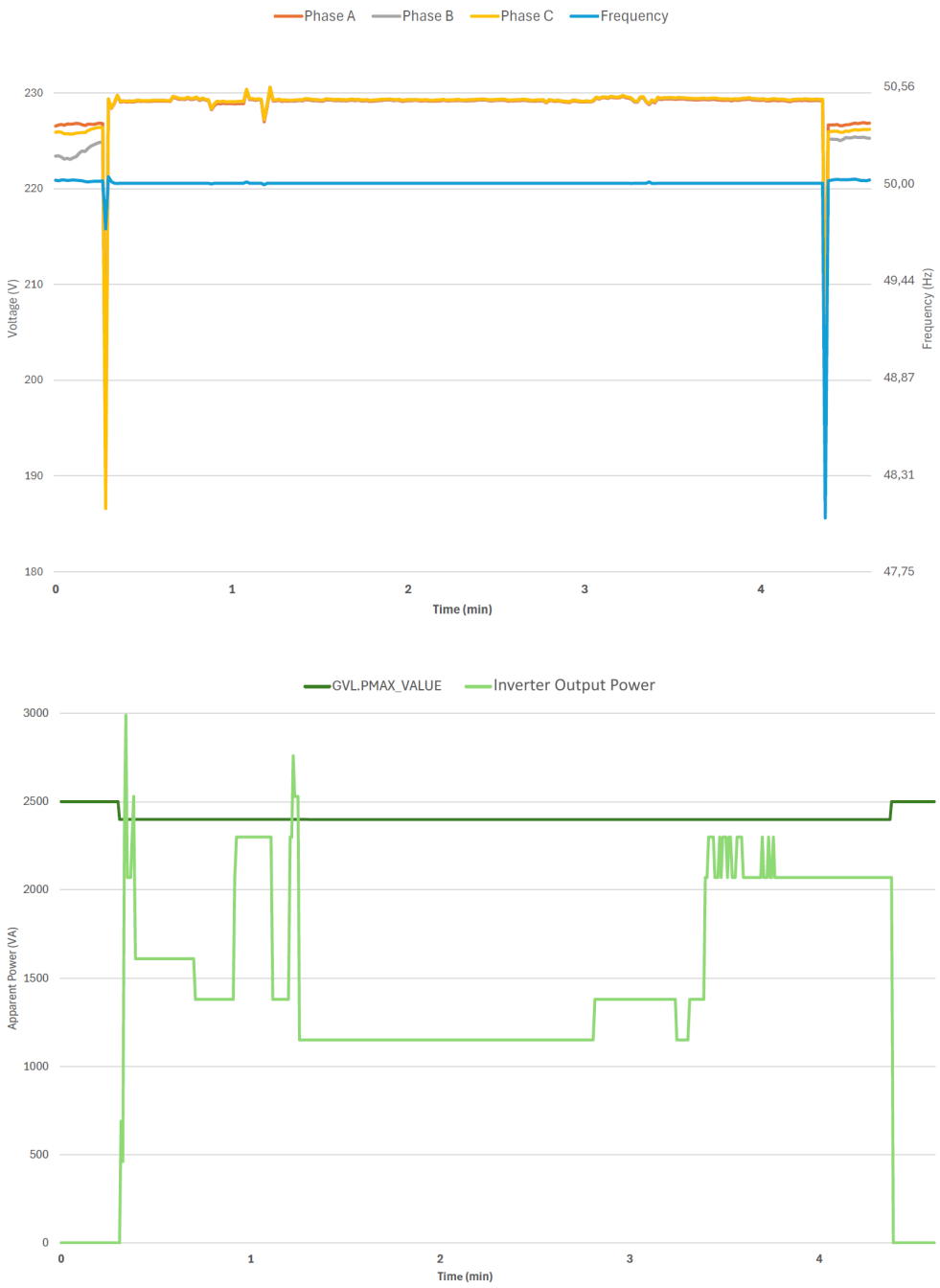


Figure 5.2: *Voltage, frequency, inverter output power and maximum power variation during islanded mode operation.*

Analysing the plots, voltage and frequency disturbances are registered when the MG load consumption increases. On-site, a minor flickering of the lights can also be observed. Nevertheless, the inverter is able to maintain a stable operation throughout the main grid power outage.

5.1.3 MG's reaction time to an overload of the inverter

The aim of this experiment is to increase the MG's power consumption above the inverter's nominal power when in islanded mode, and conclude if the MG can shed its loads quickly enough before the inverter shuts down due to overload.

The first experiment is registered in Figure 5.3, that displays a chart with the inverter's output power and GVL.PMAX_VALUE plots. At 2.4 minutes, only contactor 4 is open and there is no power consumption on the sockets' circuits, bringing the total MG consumption to 2070 VA. At instant I1, a 1800 W load is then manually connected to sockets' circuit 2 (contactor 3), increasing the MG consumption to around 3800 VA, and consequently contactor 7 is opened (by the PLC), followed by 8 and 3, bringing the MG consumption down to 1150 VA. At this point, the 1800 W load was manually disconnected from sockets' circuit 2. At 4.5 minutes (2 minutes later), the opened contactors are gradually closed again (by the PLC) with only contactor 4 left open because even though contactor 4 is prioritized over 7, the algorithm knows the MG can´t handle closing contactor 4 without exceeding GVL.PMAX_VALUE, therefore contactor 7 is closed and 4 remains open. At this point, the MG is consuming 2300 VA and then, at instant I2, another load is connected to sockets' circuit 2, but this time it has a power of 2460 W, which increases the MG power consumption to 4200 VA. Contactor 7 is immediately opened, followed by 8 and 3.

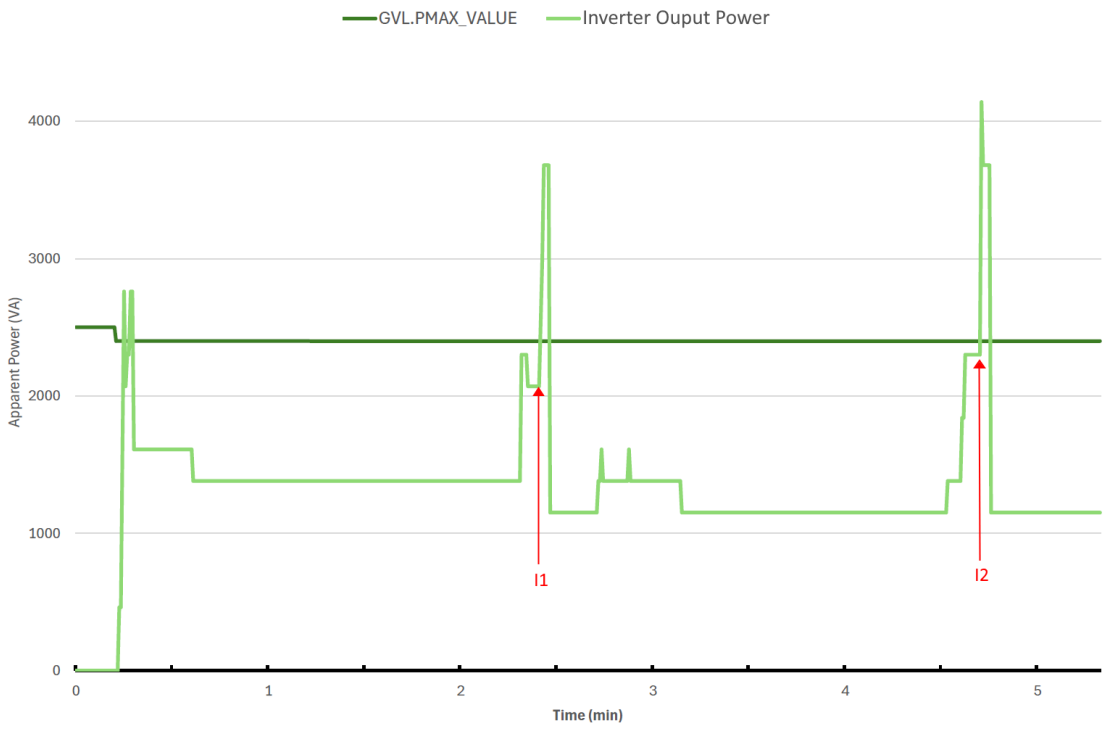


Figure 5.3: *Inverter output power and maximum power variation in islanded mode.*

The second experiment is registered in Figure 5.4 that displays a chart with the inverter's output power and GVL.PMAX_VALUE plots. At 0.4 minutes, only contactors 4 and 7 are open and there is no load on the sockets' circuits. The MG is consuming around 1300 VA and, at instant I1, a 2460 W load is connected to sockets' circuit 2, increasing consumption to 3700 VA. Hence, the PLC opens contactor 8, followed by 3, which decreases MG consumption to 1150 VA. At 2.2 minutes the garage gate opens and at 2.5 minutes, contactors 3, 8 and 7 are closed by the PLC in this order, increasing consumption to 2070 VA. A few seconds afterwards, at instant I2, a 2460 W load is connected to sockets' circuit 1, increasing MG consumption to 3700 VA. As a result, every contactor is opened in order of their priority, from lower to higher, bringing MG consumption to zero. However, this is a misreading of the inverter's output power from the PLC, because power is still being supplied to critical loads as it can be seen with the consumption increase at 3.2 minutes, that represents the garage gate closing.

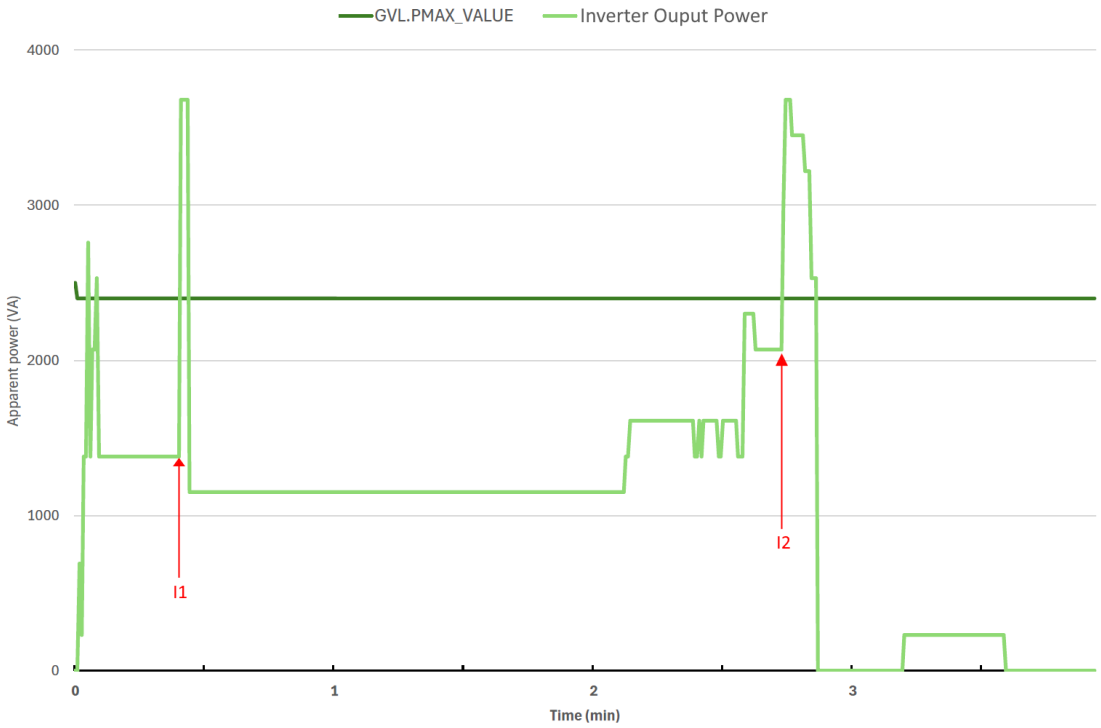


Figure 5.4: *Inverter output power and maximum power variation in islanded mode.*

In summary, this test demonstrates that the MG reacts rapidly enough to an overload of the inverter, preventing it from shutting down. When connecting a load to sockets' circuit 2, the reaction time is around 3 seconds and when connecting to sockets' circuit 1 is around 7-8 seconds. This difference is due to contactor 2 having the most priority over the rest, hence the PLC takes a couple more seconds to open it, compared to contactor 3 which is fourth in the priority table 3.1.

5.1.4 Power factor and reactive power

This test is the same presented in section 5.1.2, but the aim is to determine what is the power factor (PF) of the MG while operating in islanded mode and to quantify the reactive power circulating within the MG.

Figure 5.5 displays a chart with the real, apparent and reactive power plots, together with the PF plot of the MG during islanded operation. The PF is quite low throughout the experience, being always around 0.55. However, at the 1 minute mark it increases to 0.8, drops for a brief moment, and few seconds later reaches 0.9. These peaks happen because a high power consuming resistive load is connected to the MG, the load being of 1000 W in the first instance and 1500 W in the second. At approximately 3.5 minutes, the PF drops to 0.5 due to the activation of contactors 3, 8, and 7, which switch on the corresponding light circuits that are highly inductive loads. Simultaneously, the apparent and reactive power increase significantly, while the active power only shows a slight increase.

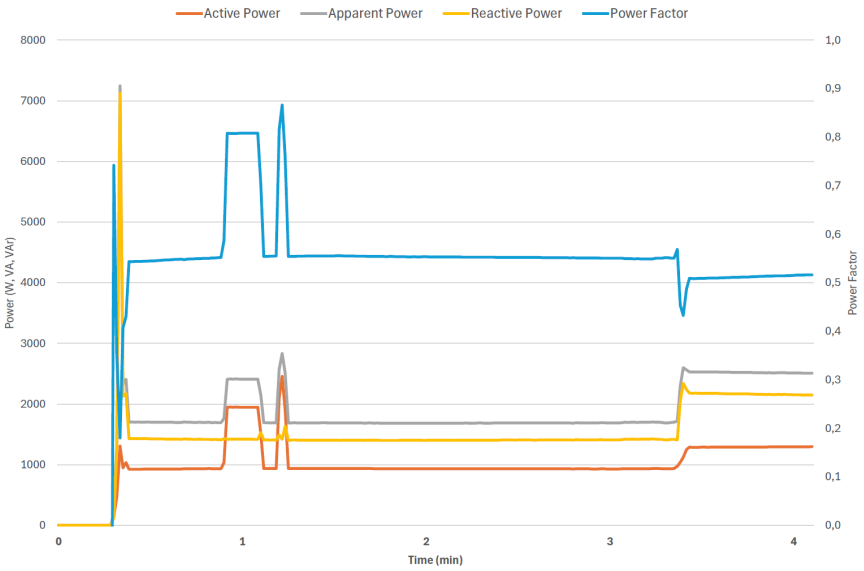


Figure 5.5: *Real power, apparent power, reactive power and PF variation during islanded mode operation.*

In conclusion, the garage light circuits, which are highly inductive loads, contribute to a significant amount of reactive power circulating within the microgrid, resulting in a very low power factor. This condition is acceptable only when an equivalent highly resistive load is connected. Consequently, this experiment demonstrates that the energy generated by the PV system is not utilised efficiently within the microgrid. One potential solution to this issue would be the installation of a capacitor bank in the electrical board to correct the power factor.

5.1.5 Algorithm operation

This experiment aims to showcase the MG operation using the previous and the current state machine, as well as compare both performances.

Figure 5.6 is referent to the old state machine operation and presents a chart with the inverter's output power, maximum power and batteries SOC plots during a 90 minute power failure. A PO is registered and before its first hour, the maximum power available is 2400 W. Constant variances in the MG consumption are registered due to contactors 4 and 7 being consistently closed and opened. After an hour, GVL.PMAX_VALUE is instantly set to around 760 VA using equation 4.3 in section 4.2.4. As a result, MG consumption drops to 460 VA and, consequently, batteries SOC stabilizes.

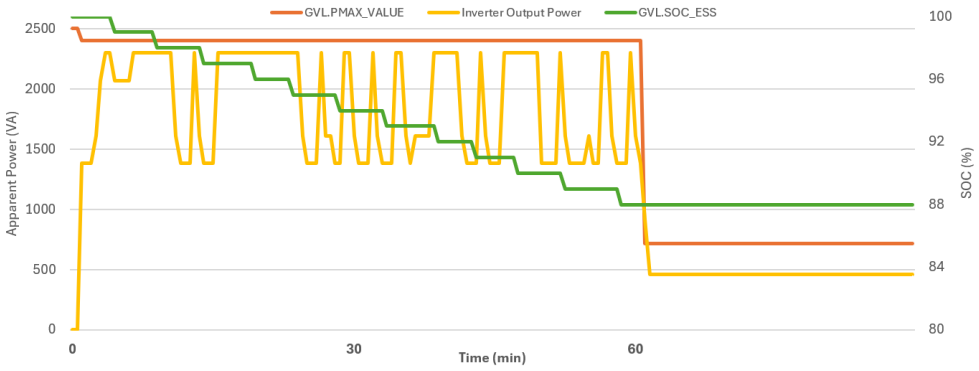


Figure 5.6: *Inverter output power and maximum power variation, along with batteries' SOC.*

Figure 5.7 corresponds to the current state machine operation, and displays a graph of the inverter's output power, maximum power and SOC plots during a 90 minute power failure. A more stable MG power consumption is noticeable, and GVL.PMAX_VALUE is maintained at 2400 W even after the first hour, as SOC is still above 78%.

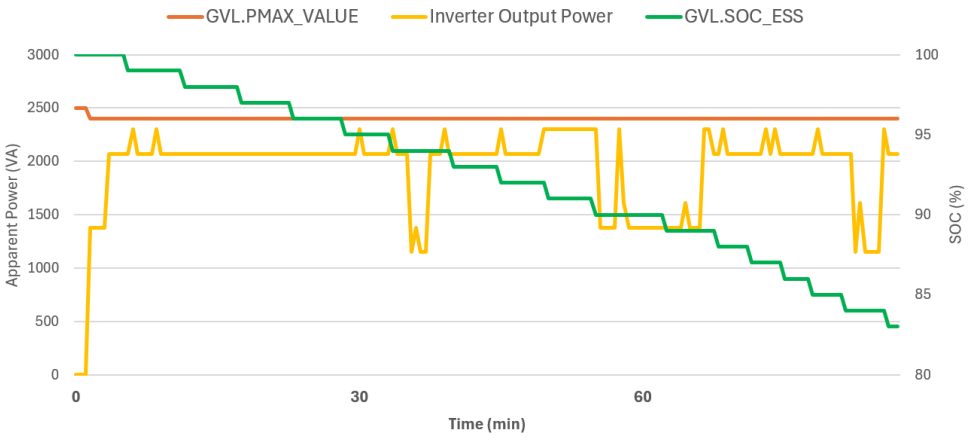


Figure 5.7: *Inverter output power and maximum power variation, along with batteries' SOC.*

The current state machine provides a wiser control of the contactors compared to the previous one, stabilizing the MG power consumption. Additionally, it allows the connection of high power consuming loads for larger time periods, since the energy to supply critical loads is guaranteed.

5.1.6 Long period power failure

The objective of this test is to evaluate the reliability of the microgrid during long period power outages.

Figure 5.8 shows a chart with the GVL.PMAX_VALUE, inverter's output power and GVL.SOC_ESS plots during a PO. The PO starts at 7 PM with the batteries' SOC at 95%. There was no more PV energy production during that day, hence SOC starts decreasing gradually over time. The non-critical loads contactors were manually opened during the first night by switching off the contactors circuit breakers (K2..K8 in Figure 3.7 (b)), therefore only critical loads were being supplied. The inverter's output power readings appear as zero almost all the time when only critical loads are supplied due to the software's low reading resolution. This causes it to misread and round down small power values to zero, even though the actual output power is not zero.

At around 7 AM, 'GVL.PMAX_VALUE' is limited to 600 VA as SOC was lower than 78%. At 10 AM, the PV panels start producing energy and the batteries' SOC starts to increase because the energy produced is greater than the energy supplied to critical loads. At the same hour, a small peak of the inverter's output power is registered due to the garage gate opening/closing. After 10 AM, the non-critical loads contactors circuit breakers that were switched off in the first night were manually switched back on. Once SOC starts increasing and it is greater than 78%, 'GVL.PMAX_VALUE' is set to 2400 VA, and after SOC reaches 84%, the contactors are closed automatically by the PLC. As a result, MG consumption increases and SOC lowers until 77%, setting 'GVL.PMAX_VALUE' to 600 VA which makes the PLC open the non-critical loads contactors. The batteries' SOC returns to rise and when once reaches 84% the PLC re-closes the non-critical loads contactors. Consequently, SOC begins to decrease and, once lower than 78%, 'GVL.PMAX_VALUE' is set to 600 VA and contactors are opened by the PLC. After around 5 PM, PV energy production fell below the energy demand of the critical loads, causing the SOC to continue decreasing throughout the second night, getting to 58% at 7 AM. At 8 AM, PV energy production surpasses the MG's energy demand and SOC starts to increase. Compared to the battery recharge on

the first day, the SOC increased more rapidly on the second day due to more sunlight and clear skies, resulting in higher PV energy production. The peaks registered in the inverter's output power between 8 AM and 13 AM are a result of the garage gate opening/closing, which increases the critical loads power consumption. At around 2 PM, SOC reaches 84%, prompting the PLC to close the non-critical load contactors. This action increases the MG's energy demand, causing the SOC to decrease.

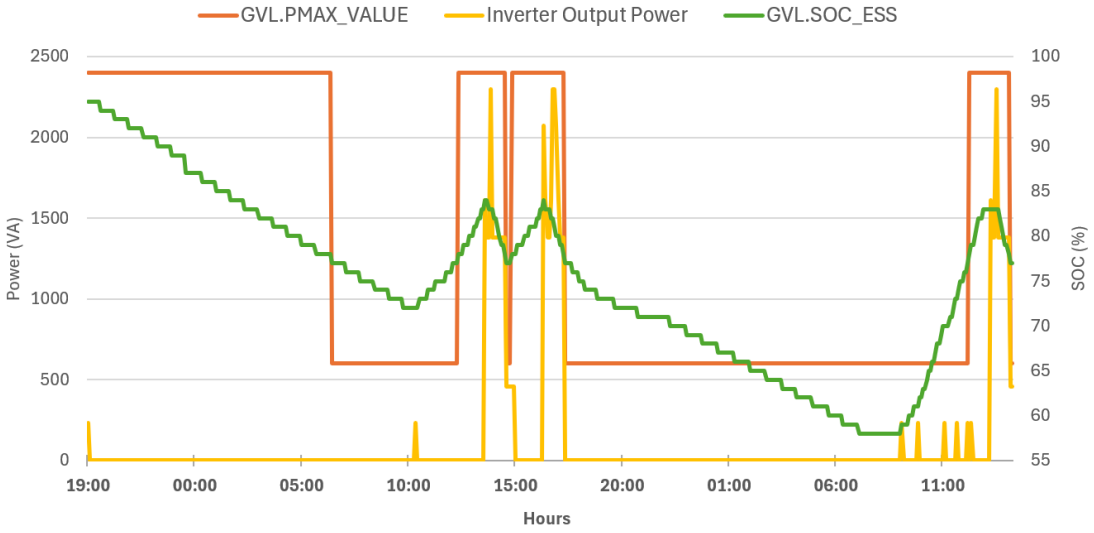


Figure 5.8: *Inverter output power and maximum power variation, along with batteries' SOC, during a 44h PO.*

The experiment demonstrates that the load control algorithm effectively manages the batteries' SOC, consistently supplying critical loads and, when there is sufficient energy stored in the batteries, also supplying non-critical loads. This leads to the conclusion that the load control algorithm is reliable and capable of maintaining the microgrid's power supply during extended power outages.

6 Conclusions and Further Developments

The primary objective of this dissertation was to develop and successfully implement a load control algorithm aimed at enhancing resiliency in critical scenarios, specifically within a microgrid pilot installed in DEEC's garage.

To achieve this goal, the load control algorithm and state machine concepts developed in previous projects needed to be restructured and adjusted to meet the practical requirements of implementation in a microgrid pilot. This process uncovered flaws that were not anticipated during the initial conceptual development of the microgrid operation, which led up to changes having to be made in the pre-existing microgrid architecture. Specifically, these changes involved the installation of new devices required to develop a microgrid operation capable of performing the intended functionalities.

The rebuilding of the microgrid operation involved developing a new state machine that featured fewer states and simpler operation, achieving more efficient transitions between on-grid and islanded modes, as well as intermediate states. A new load control algorithm was developed, setting up communication protocols between the PLC and microgrid devices to incorporate the inverter's output power and batteries' SOC into the algorithm. Additionally, this development included successfully implementing procedures for power outage and power restoration detection and developing a methodology to reactivate shed loads when possible to optimize energy production through PV generation. After developing the load control algorithm, it was tested in the new microgrid architecture. Over time, new versions were uploaded with updates to enhance efficiency, ultimately resulting in the proposed solution in this dissertation.

The results presented in this thesis show that the proposed load control algorithm can maintain energy supply during a main-grid power outage and sustain critical loads for extended periods, demonstrating its resiliency and reliability in various scenarios. Looking at

the analysed data in section 1.1 provided by E-Redes, it shows that this microgrid could maintain critical and some non-critical loads supplied during the energy service interruptions registered in the Open Data portal. Prior analysis of the loads to be supplied and the necessary power generation and energy storage systems is essential before implementation. This is a possible solution to be implemented in facilities such as hospitals, data centers, banks or military facilities, where continuous power supply of critical loads is essential for their successful operation.

6.1 Further Developments

With the successful implementation of the proposed load control algorithm, the next steps involve further enhancing the algorithm's resiliency, efficiency, and versatility.

Regarding efficiency, installing a capacitor bank in the garage's electrical board to correct the power factor, would be an immediate improvement. Furthermore, integrating PV generation data into the algorithm is a crucial short-term development. This data would enable a more efficient decision-making process for supplying non-critical loads, rather than relying solely on the batteries' SOC. Additionally, access to weekly or monthly PV energy production data could pave the way for developing a machine learning solution to predict PV generation based on recent historical data, which could then be integrated into the proposed algorithm in this thesis.

In terms of versatility, it would be advantageous for the load control algorithm to incorporate various operational modes. Currently, the system switches to islanded mode only when a power failure is detected from the main grid. However, expanding the microgrid's capability to operate in islanded mode for economic and optimization purposes could be beneficial on periods of very high demand and during power outages. A three-position switch is installed on the new electrical board of the garage, with each position connected to a digital input of the PLC. The algorithm could be programmed to adapt its operation based on the switch's position. The different operational modes could be: 1- current operation; 2- islanded mode with main grid assistance for backup power as needed; 3- full islanded mode. In addition, four more ON/OFF switches are connected to the PLC inputs. These could be used to adjust the loads priority, providing more flexibility to the end user. Also, one of those switches could maintain or eliminate the 20% battery reserve safety in islanded mode.

7 Bibliography

[1] *Institute for Economics & Peace, Ecological Threat Register 2020: Understanding Ecological Threats, Resilience and Peace.* Profanity Institute, Sydney, 2020. https://www.visionofhumanity.org/wp-content/uploads/2020/10/ETR_2020_web-1.pdf.

[2] Vento forte provocou falhas de energia em Bragança e Vila Real. *MadreMedia/Lusa, SAPO*, Março 2024. https://24.sapo.pt/atualidade/artigos/vento-forte-provocou-falhas-de-energia-em-braganca-e-vila-real.

[3] Incêndios provocam falhas nas comunicações e eletricidade em algumas zonas. *RTP Notícias*, Julho 2022. https://www.rtp.pt/noticias/economia/incendios-provocam-falhas-nas-comunicacoes-e-eletricidade-em-algumas-zonas_v1419351.

[4] Forging a climate-resilient Europe - the new EU Strategy on Adaptation to Climate Change. *EUR*, 2021. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2021:82:FIN.

[5] Robert Walton. Aging grids drive $51B in annual utility distribution spending. *Utility Dive*, Jul 2018. https://www.utilitydive.com/news/aging-grids-drive-51b-in-annual-utility-distribution-spending/528531/.

[6] Chuck Brooks. Three alarming threats to the U.S. Energy Grid – Cyber, physical, and existential events. *Forbes*, Feb 2023. https://www.forbes.com/sites/chuckbrooks/2023/02/15/3-alarming-threats-to-the-us-energy-grid–cyber-physical-and-existential-events/?sh=50a1ac08101a.

[7] Portal Open Data da E-REDES. https://e-redes.opendatasoft.com/pages/homepage/.

[8] Pedro Moura, Alexandre Correia, Joaquim Delgado, Paula Fonseca, and Aníbal de Almeida. University Campus Microgrid for Supporting Sustainable Energy Systems

Operation. In *2020 IEEE/IAS 56th Industrial and Commercial Power Systems Technical Conference (ICPS)*, pages 1–7, 2020. https://ieeexplore.ieee.org/document/9176755.

[9] Alexandre F. M. Correia, Pedro Moura, and Aníbal T. de Almeida. Technical and economic assessment of battery storage and vehicle-to-grid systems in building microgrids. *Energies*, 15(23), 2022. https://www.mdpi.com/1996-1073/15/23/8905.

[10] Alexandre Correia, José Lopes, A. Coimbra, Pedro Moura, and Aníbal de Almeida. Architecture and Control Simulation for Improved System Resiliency of a Building Microgrid. pages 1–8, 05 2023. https://ieeexplore.ieee.org/document/10142074.

[11] José Miguel Brito Lopes. Control and Automation of a Resilient Microgrid. Master's thesis, 2022.

[12] Miguel Santos Cavaleiro. Automatic Control of a Resilient Microgrid. Master's thesis, 2023.

[13] Alexandre F. M. Correia, Miguel Cavaleiro, Miguel Neves, A. Paulo Coimbra, Tony R. O. Almeida, Pedro Moura, and Aníbal T. de Almeida. Architecture and Operational Control for Resilient Microgrids. In *2024 IEEE/IAS 60th Industrial and Commercial Power Systems Technical Conference (ICPS)*, pages 1–12, 2024. https://ieeexplore.ieee.org/document/10563523.

[14] Microgrid Definition Electropedia, London, U.K., 2021. . https://www.electropedia.org/iev/iev.nsf/display?openformievref=617-04-22.

[15] SAAD AHMAD, MD SHAFIULLAH, CHOKRI BELHAJ AHMED, and MAAD ALOWAIFEER. A review of microgrid energy management and control strategies. page 3, 2013. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10050868.

[16] Heri Suyanto and Rina Irawati. Study Trends and Challenges of the Development of Microgrids. In *2017 6th IEEE International Conference on Advanced Logistics and Transport (ICALT)*, pages 160–164, 2017. https://ieeexplore.ieee.org/abstract/document/8547028.

[17] Muhammad Hammad Saeed, Wang Fangzong, Basheer Ahmed Kalwar, and Sajid Iqbal. A Review on Microgrids' Challenges Perspectives. *IEEE Access*, 9:166508–166511, 2021. https://ieeexplore.ieee.org/abstract/document/9648165.

[18] Wei Feng, Ming Jin, Xu Liu, Yi Bao, Chris Marnay, Cheng Yao, and Jiancheng Yu. A review of microgrid development in the United States – A decade of progress on policies, demonstrations, controls, and software tools. *Applied Energy*, 228:1656–1668, 2018. https://www.sciencedirect.com/science/article/pii/S0306261918309644.

[19] David Rebollal, Miguel Carpintero-Rentería, David Santos-Martín, and Mónica Chinchilla. Microgrid and Distributed Energy Resources Standards and Guidelines Review: Grid Connection and Operation Technical Requirements. *Energies*, 14(3), 2021. https://www.mdpi.com/1996-1073/14/3/523.

[20] Kathy Hitchens. Schneider Electric, Siemens and GE Vernova named leading microgrid integrators, Jun 2023. https://www.microgridknowledge.com/industry/article/33007321/schneider-electric-siemens-and-ge-vernova-named-leading-microgrid-integrators.

[21] Jackson John Justo, Francis Mwasilu, Ju Lee, and Jin-Woo Jung. AC-microgrids versus DC-microgrids with distributed energy resources: A review. *Renewable and Sustainable Energy Reviews*, 24:387–405, 2013. https://www.sciencedirect.com/science/article/pii/S1364032113002268.

[22] Eklas Hossain, Ersan Kabalci, Ramazan Bayindir, and Ronald Perez. Microgrid testbeds around the world: State of art. *Energy Conversion and Management*, 86:132–153, 2014. https://www.sciencedirect.com/science/article/pii/S0196890414004233.

[23] David Wenzhong Gao. *Energy storage for sustainable microgrid*. Waltham: Elsevier, 2015.

[24] A. Cagnano, E. De Tuglie, and P. Mancarella. Microgrids: Overview and guidelines for practical implementations and operation. *Applied Energy*, 258:114039, 2020. https://www.sciencedirect.com/science/article/pii/S030626191931726X.

[25] Fahad Saleh Al-Ismail. DC Microgrid Planning, Operation, and Control: A Comprehensive Review. *IEEE Access*, 9:36154–36172, 2021. https://ieeexplore.ieee.org/abstract/document/9366476.

[26] M. Hijjo, F. Felgner, and G. Frey. PV-Battery-Diesel microgrid layout design based on stochastic optimization. In *2017 6th International Conference on Clean Electrical Power (ICCEP)*, pages 30–35, 2017. https://ieeexplore.ieee.org/abstract/document/8004787.

[27] Changjie Yin, Hongwei Wu, Fabrice Locment, and Manuela Sechilariu. Energy management of DC microgrid based on photovoltaic combined with diesel generator and supercapacitor. *Energy Conversion and Management*, 132:14–27, 2017. https://www.sciencedirect.com/science/article/pii/S0196890416310135.

[28] Yong Zhang and Wei Wei. Decentralized coordination control of PV generators, storage battery, hydrogen production unit and fuel cell in islanded DC microgrid. *International Journal of Hydrogen Energy*, 45(15):8243–8256, 2020. https://www.sciencedirect.com/science/article/pii/S0360319920301488.

[29] Tomislav Dragičević, Josep M. Guerrero, Juan C. Vasquez, and Davor Škrlec. Supervisory Control of an Adaptive-Droop Regulated DC Microgrid With Battery Management Capability. *IEEE Transactions on Power Electronics*, 29(2):695–706, 2014. https://ieeexplore.ieee.org/abstract/document/6497633.

[30] Peng Wang, Lalit Goel, Xiong Liu, and Fook Hoong Choo. Harmonizing ac and dc: A hybrid ac/dc future grid solution. *IEEE Power and Energy Magazine*, 11(3):76–83, 2013. https://ieeexplore.ieee.org/abstract/document/6506881.

[31] Zhenhua Jiang and Xunwei Yu. Hybrid dc- and ac-linked microgrids: Towards integration of distributed energy resources. In *2008 IEEE Energy 2030 Conference*, pages 1–8, 2008. https://ieeexplore.ieee.org/abstract/document/4781029.

[32] Eneko Unamuno and Jon Andoni Barrena. Hybrid AC/DC Microgrids—Part I: Review and classification of topologies. *Renewable and Sustainable Energy Reviews*, 52:1251–1259, 2015. https://www.sciencedirect.com/science/article/pii/S1364032115008412.

[33] Hicham Lhachimi, Yassine Sayouti, and Youssef El Kouari. Control of a flexible microgrid during both modes of operations with presence of non-linear loads. *Journal of the Franklin Institute*, 357(11):6498–6538, 2020. https://www.sciencedirect.com/science/article/pii/S0016003220302271.

[34] Tingting Wu, Gang Bao, Yuanyuan Chen, and Jinping Shang. A Review for Control Strategies in Microgrid. In *2018 37th Chinese Control Conference (CCC)*, pages 30–35, 2018. https://ieeexplore.ieee.org/document/8482549.

[35] Shamsher Ansari, Aseem Chandel, and Mohd Tariq. A Comprehensive Review on Power Converters Control and Control Strategies of AC/DC Microgrid. *IEEE Access*, 9:17998–18015, 2021. https://ieeexplore.ieee.org/abstract/document/9178771.

[36] Navid Salehi, Herminio Martínez-García, Guillermo Velasco-Quesada, and Josep M. Guerrero. A Comprehensive Review of Control Strategies and Optimization Methods for Individual and Community Microgrids. *IEEE Access*, 10:15935–15955, 2022. https://ieeexplore.ieee.org/abstract/document/9681082.

[37] Necmi Altin and Süleyman Emre Eyimaya. A Review of Microgrid Control Strategies. In *2021 10th International Conference on Renewable Energy Research and Application (ICRERA)*, pages 412–417, 2021. https://ieeexplore.ieee.org/abstract/document/9598699.

[38] Tommaso Caldognetto and Paolo Tenti. Microgrids Operation Based on Master–Slave Cooperative Control. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2(4):1081–1088, 2014. https://ieeexplore.ieee.org/abstract/document/6868993.

[39] Can Wang, Ping Yang, Chao Ye, Yuewu Wang, and Zhirong Xu. Improved V/f control strategy for microgrids based on master–slave control mode. *IET Renewable Power Generation*, 10:1356–1365, 2016. https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-rpg.2016.0063.

.1 Appendix

.1.1 Algorithm Functions

.1.1.1 PO_VAR

This function sets the value of PO control variable according to the explanation presented in section 4.1 and is used to detect the power failure from the utility grid. The *RELE_REDE* (input 0 of the PLC) shown in Figure 4.11, is the input of the function. The function returns the boolean value of the variable PO. *Read_Values* attributes it to *GVL.PO* (Listing 4.3), as explained in section 4.2.1.2.

```
FUNCTION PO_VAR : INT
VAR_INPUT
END_VAR
VAR
END_VAR
```

Listing 1: *PO_VAR function declaration block.*

```
IF RELE_REDE = TRUE THEN
  PO_VAR := 0; //no power outage
  ELSE
    PO_VAR := 1; //power outage
END_IF
```

Listing 2: *PO_VAR function code.*

.1.1.2 T_VAR

This function tracks the time elapsed since a PO and returns the value 0 in the first minute of a PO, returns 1 between the first minute and the end of the first hour since a PO, an returns 2 after the first hour has passed.

```
FUNCTION T_VAR : INT
VAR_INPUT

END_VAR
VAR
  ACTUAL_TIME: ULINT;
  DELTA: ULINT;
END_VAR
```

Listing 3: *T_VAR function declaration block.*

```
ACTUAL_TIME := GetDateTime(); //returns current time
GVL.DELTA_PO := ACTUAL_TIME - GVL.PO_TIME;  /current time - start time of the PO
IF GVL.PO = 0 THEN
  T_VAR := 0;
END_IF
IF GVL.PO = 1 THEN
  IF  GVL.DELTA_PO < GVL.MIN_1_MS THEN
    T_VAR := 0;
  END_IF
  IF GVL.DELTA_PO >= GVL.MIN_1_MS AND GVL.DELTA_PO < GVL.HORA_1_MS THEN
    T_VAR := 1;
  END_IF
  IF GVL.DELTA_PO >= GVL.HORA_1_MS THEN
    T_VAR := 2;
  END_IF
END_IF
```

Listing 4: *T_VAR function code.*

ACTUAL_TIME registers the present time using the pre-defined function *GetDateTime()*, and is used to calculate the difference of time since the start of the PO, using *GVL.PO_TIME* (start time of the PO).

.1.1.3 CALC_PMAX

CALC_PMAX returns the maximum power the inverter can supply taking into account the batteries' SOC and the value of 'GVL.T' (global variable T, described in section 4.1. The value returned by this function is attributed to the variable 'PMAX_VALUE' by the *Read_Values* program.

```
FUNCTION CALC_PMAX : INT //retorna a potencia maxima que o inversor pode disponibilizar
    tendo em conta o SOC, garantindo que duram ate ao final da primeira hora, com pelo menos
     20%, depois passa para 10 horas
VAR_INPUT
END_VAR
VAR
  PMAX: REAL; // (W)
  REMAIN_T_TO1H: REAL; // (ms)
  REMAIN_T_TOnH: REAL; // (ms)
  EN_CLOADS_T_TOnH: REAL; // (VA)
  REMAIN_SOC_CLOADS_nH: REAL; // (%)
  REMAIN_SOC_CLOADS_1H: REAL; // (%)
  aux: REAL;
  aux2: REAL;
END_VAR
```

Listing 5: *CALC_PMAX function declaration block.*

```
1  IF GVL.T = 0 AND GVL.PO = 0 THEN // ainda nao houve power outage
2    PMAX := 2500; // potencia maxima do quadro eletrico, alimentado pela rede (W)
3    ELSIF GVL.T = 0 AND GVL.PO = 1 THEN // ha PO, esta ainda no primeiro minuto
4      PMAX := GVL.MAX_INVERTER_POWER;
5      ELSIF GVL.T = 1 THEN // ha PO, ja passou o primeiro minuto, ainda nao passou a primeira
       hora
6        REMAIN_SOC_CLOADS_1H := (GVL.CLOADS/GVL.MAX_BATTERY_CAPACITY)*100 + GVL.MIN_SOC_ESS;
7        IF GVL.SOC_ESS >= REMAIN_SOC_CLOADS_1H THEN
8          PMAX := GVL.MAX_INVERTER_POWER;
9          ELSE
10           aux := GVL.CLOADS/(REMAIN_T_TO1H/3600000);
11           PMAX := aux;
12       END_IF
13       ELSIF GVL.T = 2 THEN // ha PO, passou mais de 1 hora
14         EN_CLOADS_T_TOnH := GVL.CLOADS*10; //energia reservada para as cargas criticas para
       as prox n=10 horas (W.h)
15         REMAIN_SOC_CLOADS_nH := (EN_CLOADS_T_TOnH/GVL.MAX_BATTERY_CAPACITY)*100 + GVL.
       MIN_SOC_ESS; //SOC reservado cargas criticas para 10h
16         IF GVL.SOC_ESS >= REMAIN_SOC_CLOADS_nH THEN // ha PO, passou mais de 1 hora e soc >
       minimo
17           PMAX := GVL.MAX_INVERTER_POWER;
18           ELSE
19             aux2 := EN_CLOADS_T_TOnH/(REMAIN_T_TOnH/3600000);
20             PMAX := aux2;
21         END_IF
22 END_IF
23 CALC_PMAX := REAL_TO_INT(PMAX); // (W)
```

Listing 6: *CALC_PMAX function code.*

On line 1, if the condition is satisfied, there is yet to be a PO and therefore PMAX is set to the power provided by the electrical board panel, supplied by the main grid. This value ends up not having any particular impact since, at this point, it is not the inverter supplying the MG.

On line 3 is defined the condition in case there is a PO and it is still in the first minute. At this stage the inverter provides all its available power which is 2400 VA, so PMAX is set to that value on line 4 by equalling it to GVL.MAX_INVERTER_POPWER. This variable is declared in the GVL (Listing 4.1).

From lines 5 to 12 is defined the PMAX behavior when the PO has been elapsed for over a minute and under an hour. At this point, the power supplied by the inverter starts to get managed with the priority of guaranteeing supply to critical loads. As a result and on line 6, the needed battery percentage to supply critical loads is calculated using the equation in section 4.1. From lines 7 to 11 is defined the value of 'PMAX' depending on the defined conditions.

From lines 13 to 21, when the PO has been elapsed for more than an hour, PMAX is managed with the same logic described in the previous paragraph. The difference is the calculated battery percentage for the critical loads on line 15, which takes into account the 10 continuous hours these need to be supplied.

Finally, on line 23 is instructed for the function to return the value of PMAX.

.1.1.4 LER_SOC_ESS

This function returns the batteries' SOC. As displayed on the *Read_Values* program code on Listing 4.3, this variable is updated by the *LER_SOC_ESS* function which has the following code:

```
1 FUNCTION LER_SOC_ESS : INT
2 VAR_INPUT
3 END_VAR
4 VAR
5   socess AT %IW11: INT; //SOC register address
6 END_VAR
```

Listing 7: *LER_SOC_ESS function declaration block.*

```
1 LER_SOC_ESS := socess/10; // retorna o estado de carga das baterias em %
```

Listing 8: *LER_SOC_ESS function code.*

On line 5 of the function declaration block, the address %IW11 is allocated to *socess*. As explained in the Modbus TCP configuration section in 4.2.1.4, this is the batteries SOC value with a scale factor of 10, so on Listing 8 the value is divided by its scale factor to get the real value. *LER_SOC_ESS* then returns the value as a percentage and finally attributes it to *GVL.SOC_ESS* in the *Read_Values* program.

.1.1.5 LER_INV_POWER

This function returns the apparent power (VA) being supplied by the inverter to the MG. As explained in section 4.2, in the Modbus TCP subsection, Figure 4.10 presents the protocol's channel registers with their corresponding value and address. The inverter's output current is registered in address %IW12. On line 5 of the declaration block (Listing 9) the address is allocated to *'corrente'*.

Finally, in the function code is instructed for the function to return the value of the apparent power (*invpower*) after multiplying the current, divided by its scale factor, with the voltage (230 V).

```
FUNCTION LER_PV_POWER : INT
VAR_INPUT
END_VAR
VAR
  corrente AT %IW12 : INT; //inverter output current register address
  invpower : INT;
END_VAR
```

Listing 9: *LER_PV_POWER function declaration block.*

```
inpower := (corrente/10)*230;
LER_PV_POWER := invpower; //retorna potencia em VA fornecida pelo inversor
```

Listing 10: *LER_PV_POWER function code.*

.1.1.6 LER_CG2 and LER_CG3

LER_CG2 and *LER_CG3* return the instant power consumed (W) in socket circuit 1 and 2, respectively. In subsection 4.2.1.3, is explained how the protocol operates and that the power consumed by circuit 1 is registered in address %IW17. Because of that, on line 5 of Listing 11, the value of the address is allocated to *pcg2*.

In Listing 12 is instructed for the function to return the value of *pcg2*, divided by its scale factor.

```
FUNCTION LER_CG2 : INT
VAR_INPUT
END_VAR
VAR
  pcg2 AT %IW17: INT; //socket circuit 1 power register address
END_VAR
```

Listing 11: *LER_CG2 function declaration block.*

```
LER_CG2 := pcg2/10; //retorna potencia em watts lida no EM2
```

Listing 12: *LER_CG2 function code.*

LER_CG3 returns the power being consumed in socket circuit 2 using the same method as *LER_CG2*. The difference is the address that registers the power from socket circuit 2, which is %IW37. As a result, it is allocated to *pcg3* on line 5 of the declaration block and then the function returns that value divided by its scale factor.

```
FUNCTION LER_CG3 : INT
VAR_INPUT
END_VAR
VAR
  pcg3 AT %IW37: INT; //socket circuit 2 power register address
```

```
END_VAR
```

Listing 13: *LER_CG3 function declaration block.*

```
LER_CG3 := pcg3/10;  //retorna potencia em watts lida no EM3
```

Listing 14: *LER_CG3 function code.*

.1.2 State Machine Developed Code

.1.2.1 Main Program Declaration Block

```
PROGRAM SR_Main
VAR
  STATE: INT := 0;
  P_1H: INT;  //potencia max que pode ser consumida para T=1
  P_10H: INT; //potencia max que pode ser consumida para T=2
  // P_CONS: INT;
  P_DISP_1M: INT; //potencia max que pode ser consumida para T=0
  CHECK_TIMER: ULINT; //timer para ajudar a garantir seguranca da transicao
  DELTA_T: ULINT; //diferenca de tempo dos timers
  TIMER_CORTE_CONTACTOR: ULINT := 0;  //timer para as variaveis terem tempo de atualizar
    antes do proximo contactor desligar
  TIMER_FECHAR_CONTACTOR: ULINT := 0; //timer para as variaveis terem tempo de atualizar
    antes do proximo contactor ligar
  REATIVACAO_TIMER_CORTE_CONTACTOR: ULINT := 0;
  REATIVAR: ULINT :=0;              //usados para controlar a reativacao das cargas
END_VAR
```

Listing 15: *Main program - Declaration block.*

.1.2.2 State 10 - Grid-connected Mode

As soon as a PO is detected, the time of the PO is registered in a variable using the pre-defined function *GetDateTime()*, which registers the current date and time from the PLC and returns a string with that information. Contactor 1 is opened and a security timer is launched to ensure contactor 10 is not closed while 1 is still switching. The program then moves on to state 45.

```
  10: // GRID CONNECTED
    C1 := TRUE;      //contactor 1 fechado
    C10:= FALSE;     //contactor 10 aberto
    C2 := TRUE;      //contactor 2 fechado
    C3 := TRUE;      //contactor 3 fechado
    C4 := TRUE;      //contactor 4 fechado
    C5 := TRUE;      //contactor 5 fechado
    C6 := TRUE;      //contactor 6 fechado
```

```
9     C7 := TRUE;      //contactor 7 fechado
10    C8 := TRUE;      //contactor 8 fechado
11
12    IF GVL.PO = 1 THEN
13      GVL.PO_TIME := GetDateTime();
14      C1 := FALSE; // aqui abrimos da rede (contactor 1)
15      C10:= FALSE; // REDUNDANTE SEGURANCA
16      GVL.TRANSITION_SECURITY := GetDateTime();    (*set timer*)
17      STATE := 45;
18    END_IF
```

Listing 16: *Main program - State 10 code.*

.1.2.3 State 45 - Transition and Control

Firstly, the program makes sure the delta time (DELTA_T) is above 200 *ms* to guarantee a secure transition. Following that, the *IF* on line 6 handles the transition back to grid-connected mode, in case PO = 0. Then, on line 13, the algorithm checks the control variables and only now closes contactor 10, completing the transition to island mode. After checking all variables it decides the next state which, following this timeline and assuming the batteries have enough energy, is state 20.

```
1     45: // CONTROL
2     CHECK_TIMER := GetDateTime();
3     DELTA_T := CHECK_TIMER - GVL.TRANSITION_SECURITY;
4     IF DELTA_T > 200 THEN // check timer
5
6       IF GVL.PO = 0 THEN //TRANSITION FROM ISLANDED MODE TO ON-GRID
7         GVL.TRANSITION_SECURITY := 0;
8         DELTA_T := 0; // reset timer
9         C1 := TRUE; // fechamos a rede (CONTACTOR 1)
10        STATE := 10;
11      END_IF
12
13      IF GVL.PO = 1 THEN
14        GVL.TRANSITION_SECURITY := 0;
15        DELTA_T := 0; // reset timer
16        C10 := TRUE; // so aqui se fecha o inversor (CONTACTOR 10)
17        IF GVL.SOC_ESS > GVL.MIN_SOC_ESS THEN
18          IF GVL.T = 0 THEN
19            STATE := 20;
20          END_IF
21          IF GVL.T = 1 THEN
22            STATE := 30;
23          END_IF
24          IF GVL.T = 2 THEN
25            STATE := 40;
26          END_IF
```

```
27          ELSE
28          STATE := 0;
29          END_IF
30        END_IF
31      END_IF
```

Listing 17: *Main program - State 45 code.*

.1.2.4 State 20 - Maximum charges supplied for one minute

As explained in subsection .1.1.3, the *CALC_PMAX()* function returns the maximum power the inverter can supply. Looking into its code on Listing 6 on lines 3 and 4, when T=0 and PO=1, PMAX is set to 'GVL.MAX_INVERTER_POWER', which is defined in the GVL (Listing 4.1) on line 26. Finally, in the *Read_Values* program (Listing 4.3), on line 3, the returned value of the function is attributed to 'GVL.PMAX_VALUE'. This variable is equalled to P_DISP_1M on line 5 of the state 20 code block (Listing 18), and is used for comparison with the real MG power consumption for the load shedding operation.

From lines 7 to 66 is defined a set of conditions to operate the load shedding. The blocks of conditions are in order of the contactor's priority defined in table 3.1, from lowest to highest. Therefore, contactor 7 is operated in the first block from lines 7 to 12 in the following way: if the contactor is closed ('TRUE' in code) and the MG power consumption (GVL.INV_POWER) is greater than the maximum power available from the inverter, the contactor is opened ('FALSE' on code). GVL.INV_POWER returns the value of the *LER_INV_POWER* function, as explained in section .1.1.5. On line 10, after opening the contactor, the time when it was opened is saved in 'TIMER_CORTE_CONTACT0R' using *GetDateTime()*. This variable is then used in the beginning of the next block of conditions on line 14, where a condition is set that only enables the operation of contactor 4 after 1.5 seconds (1500 ms) since opening the previous contactor. This is done to give enough time for the 'GVL.INV_POWER' variable to update the MG power consumption value. The following blocks of conditions use the same operation philosophy in order of the contactor's priority.

The last set of conditions that begins on line 68, is always present at the end of each island mode state to check the control variables and change state when needed.

```
1  20: // ISLANDED MODE, MAXIMUM LOADS SUPPLIED FOR 1 MINUTE SINCE POWER OUTAGE
2    C1 := FALSE;  // REDUNDANTE SEGURANCA
3    C10:= TRUE;   // REDUNDANTE SEGURANCA
4
5    P_DISP_1M := GVL.PMAX_VALUE; //2400 VA
```

```
6
7     IF C7 = TRUE THEN
8       IF GVL.INV_POWER > P_DISP_1M THEN
9         C7 := FALSE;
10        TIMER_CORTE_CONTACTOR := GetDateTime();
11      END_IF
12    END_IF
13
14    IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
15      IF C4 = TRUE THEN
16        IF GVL.INV_POWER > P_DISP_1M THEN
17          C4 := FALSE;
18          TIMER_CORTE_CONTACTOR := GetDateTime();
19        END_IF
20      END_IF
21    END_IF
22
23    IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
24      IF C8 = TRUE THEN
25        IF GVL.INV_POWER > P_DISP_1M THEN
26          C8 := FALSE;
27          TIMER_CORTE_CONTACTOR := GetDateTime();
28        END_IF
29      END_IF
30    END_IF
31
32    IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
33      IF C3 = TRUE THEN
34        IF GVL.INV_POWER > P_DISP_1M THEN
35          C3 := FALSE;
36          TIMER_CORTE_CONTACTOR := GetDateTime();
37        END_IF
38      END_IF
39    END_IF
40
41    IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
42      IF C5 = TRUE THEN
43        IF GVL.INV_POWER > P_DISP_1M THEN
44          C5 := FALSE;
45          TIMER_CORTE_CONTACTOR := GetDateTime();
46        END_IF
47      END_IF
48    END_IF
49
50    IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
51      IF C6 = TRUE THEN
52        IF GVL.INV_POWER > P_DISP_1M THEN
53          C6 := FALSE;
54          TIMER_CORTE_CONTACTOR := GetDateTime();
55        END_IF
56      END_IF
```

```
57    END_IF
58
59    IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
60      IF C2 = TRUE THEN
61        IF GVL.INV_POWER > P_DISP_1M THEN
62          C2 := FALSE;
63          TIMER_CORTE_CONTACTOR := GetDateTime();
64        END_IF
65      END_IF
66    END_IF
67
68    // -------------------------- mudanca de estado -----------------------------
69    IF GVL.T = 1 THEN
70      STATE := 30;
71    END_IF
72    IF GVL.SOC_ESS < GVL.MIN_SOC_ESS THEN
73      STATE := 45;
74    END_IF
75    IF GVL.PO = 0 THEN
76      C1 := FALSE; // REDUNDANTE SEGURANCA
77      C10 := FALSE; // abrimos o inversor (CONTACTOR 10)
78      GVL.TRANSITION_SECURITY := GetDateTime(); // set timer
79      STATE := 45;
80    END_IF
```

Listing 18: *Main program - State 20 code.*

Once again, assuming there is still a PO and the batteries are above 20% capacity, the algorithm moves on to state 30.

.1.2.5 State 30 - Maximum supply for the MG to last 1 hour

Similarly to state 20, on line 5 the maximum power the inverter can supply is attributed to 'P_1H', and the variable is then used as reference for the load shedding operation. From lines 10 to 69, the load shedding *modus operandis* is the same as in state 20. From lines 73 to 160 is defined the set of conditions for reactivating the loads that were shed before. The loads are reactivated in order of priority, from highest to lowest, according to table 3.1. On line 73, the current time is saved in 'REATIVACAO_TIMER_CORTE_CONTACTOR', and on the next line is used to calculate the time since the contactor was opened, saving it in 'REATIVAR'. From lines 75 to 82, contactor 2 is managed in the following way: if the contactor has been opened ('FALSE' in code) for more than 2 minutes (120 000 ms), and the batteries SOC (GVL.SOC_ESS) is higher than 30%, then the contactor is closed ('TRUE' on code) only if the sum of the MG and the contactor's circuit consumption is lower than the maximum power available from the inverter. The reason for the 30% of the SOC is because

the guaranteed SOC to supply critical loads, when T=1, is around 25% as explained in section 4.1, so a 5% gap is placed to allow a small delay in the contactors reactivation, making the transition smoother. On line 79, 'TIMER_FECHAR_CONTACTOR' saves the time when the contactor was closed and is used in the beginning of the next block of conditions on line 84, to only enable the operation of the next contactor 2.5 seconds (2500 ms) after closing the previous contactor. The reason for this is to give enough time for the circuits to fully reactivate, in order for 'GVL.INV_POWER' to update the MG consumption value correctly. Note that for contactors 4 and 7 this threshold is 5 seconds because these light circuits take more time to turn on compared to others. The following blocks of conditions use the same operation philosophy in order of the contactor's priority (highest to lowest).

```
30: // ISLANDED MODE, MAXIMUM LOADS FOR MICROGRID TO LAST 1 HOUR SINCE POWER OUTAGE
  C1 := FALSE;
  C10:= TRUE;

  P_1H := GVL.PMAX_VALUE; //potencia


  // -------------- corte de cargas por nivel de prioridade ---------------

  IF C7 = TRUE THEN
    IF GVL.INV_POWER > P_1H THEN
      C7 := FALSE;
      TIMER_CORTE_CONTACTOR := GetDateTime();
    END_IF
  END_IF

  IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
    IF C4 = TRUE THEN
      IF GVL.INV_POWER > P_1H THEN
        C4 := FALSE;
        TIMER_CORTE_CONTACTOR := GetDateTime(); //
      END_IF
    END_IF
  END_IF

  IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
    IF C8 = TRUE THEN
      IF GVL.INV_POWER > P_1H THEN
        C8 := FALSE;
        TIMER_CORTE_CONTACTOR := GetDateTime(); //
      END_IF
    END_IF
  END_IF

  IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
    IF C3 = TRUE THEN
```

```
37          IF GVL.INV_POWER > P_1H THEN
38            C3 := FALSE;
39            TIMER_CORTE_CONTACTOR := GetDateTime();
40          END_IF
41        END_IF
42      END_IF
43
44      IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
45        IF C5 = TRUE THEN
46          IF GVL.INV_POWER > P_1H THEN
47            C5 := FALSE;
48            TIMER_CORTE_CONTACTOR := GetDateTime();
49          END_IF
50        END_IF
51      END_IF
52
53      IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
54        IF C6 = TRUE THEN
55          IF GVL.INV_POWER > P_1H THEN
56            C6 := FALSE;
57            TIMER_CORTE_CONTACTOR := GetDateTime();
58          END_IF
59        END_IF
60      END_IF
61
62      IF GetDateTime() > TIMER_CORTE_CONTACTOR + 1500 THEN //tempo resposta da variavel
63        IF C2 = TRUE THEN
64          IF GVL.INV_POWER > P_1H THEN
65            C2 := FALSE;
66            TIMER_CORTE_CONTACTOR := GetDateTime();
67          END_IF
68        END_IF
69      END_IF
70
71      // ------------ realimenta  o de cargas por nivel de prioridade ---------------
72
73      REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
74      REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
75      IF C2 = FALSE THEN
76        IF REATIVAR > 120000 AND GVL.SOC_ESS > 30 THEN
77          IF GVL.INV_POWER + GVL.EM2_POWER < P_1H THEN
78            C2 := TRUE;
79            TIMER_FECHAR_CONTACTOR := GetDateTime();
80          END_IF
81        END_IF
82      END_IF
83
84      IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 2500 THEN
85        IF C6 = FALSE THEN
86          REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
87          REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
```

```
88          IF REATIVAR > 120000 AND GVL.SOC_ESS > 30 THEN
89            IF GVL.INV_POWER + GVL.C6_VALUE < P_1H THEN
90              C6 := TRUE;
91              TIMER_FECHAR_CONTACTOR := GetDateTime();
92            END_IF
93          END_IF
94        END_IF
95      END_IF
96
97      IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 2500 THEN
98        IF C5 = FALSE THEN
99          REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
100         REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
101         IF REATIVAR > 120000 AND GVL.SOC_ESS > 30 THEN
102           IF GVL.INV_POWER + GVL.C5_VALUE < P_1H THEN
103             C5 := TRUE;
104             TIMER_FECHAR_CONTACTOR := GetDateTime();
105           END_IF
106         END_IF
107       END_IF
108     END_IF
109
110     IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 2500 THEN
111       IF C3 = FALSE THEN
112         REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
113         REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
114         IF REATIVAR > 120000 AND GVL.SOC_ESS > 30 THEN
115           IF GVL.INV_POWER + GVL.EM3_POWER < P_1H THEN
116             C3 := TRUE;
117             TIMER_FECHAR_CONTACTOR := GetDateTime();
118           END_IF
119         END_IF
120       END_IF
121     END_IF
122
123     IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 2500 THEN
124       IF C8 = FALSE THEN
125         REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
126         REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
127         IF REATIVAR > 120000 AND GVL.SOC_ESS > 30 THEN
128           IF GVL.INV_POWER + GVL.C8_VALUE < P_1H THEN
129             C8 := TRUE;
130             TIMER_FECHAR_CONTACTOR := GetDateTime();
131           END_IF
132         END_IF
133       END_IF
134     END_IF
135
136     IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 5000 THEN
137       IF C4 = FALSE THEN
138         REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
```

```
139         REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
140         IF REATIVAR > 120000 AND GVL.SOC_ESS > 30 THEN
141           IF GVL.INV_POWER + GVL.C4_VALUE < P_1H THEN
142             C4 := TRUE;
143             TIMER_FECHAR_CONTACTOR := GetDateTime();
144           END_IF
145         END_IF
146       END_IF
147     END_IF
148
149     IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 5000 THEN
150       IF C7 = FALSE THEN
151         REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
152         REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
153         IF REATIVAR > 120000 AND GVL.SOC_ESS > 30 THEN
154           IF GVL.INV_POWER + GVL.C7_VALUE < P_1H THEN
155             C7 := TRUE;
156             TIMER_FECHAR_CONTACTOR := GetDateTime();
157           END_IF
158         END_IF
159       END_IF
160     END_IF
161
162
163     // --------------------------- mudanca de estado ------------------------------
164     IF GVL.T = 2 THEN
165       STATE := 40;
166     END_IF
167     IF GVL.SOC_ESS < GVL.MIN_SOC_ESS THEN
168       STATE := 45;
169     END_IF
170     IF GVL.PO = 0 THEN
171       C1 := FALSE; // REDUNDANTE SEGURANCA
172       C10 := FALSE; // abrimos o inversor (CONTACTOR 10)
173       GVL.TRANSITION_SECURITY := GetDateTime(); // set timer
174       STATE := 45;
175     END_IF
```

Listing 19: *Main program - State 30 code.*

From line 164 to the end there are the same conditions from state 20 related to the state transition, with the exception of the time elapsed since the PO being more than an hour, therefore changing $GVL.T$ to '2'. In that case, assuming there is still a PO and battery energy is above 20%, the next state is 40.

.1.2.6 State 40 - Maximum supply for the MG to last 10 hours

State 40 has the same logic as the previous one and the code itself is very similar. The only aspect that changes is the variable to which is attributed the calculated maximum power the

inverter can supply (P_10H in line 5), and the SOC to reactivate the opened contactors. As explained in section 4.1, the needed SOC to supply critical loads for 10 continuous hours is around 78% when T=2. As a result, similarly to state 30, a 5% interval is added to smooth the transition to reactivate loads.

```
40: // APOS 1 HORA, ALIMENTA CARGAS DE MODO QUE AS BATERIAS TENHAM CARGA PARA FORNECER
  ENERGIA DURANTE 10H
  C1 := FALSE;
  C10:= TRUE;

  P_10H := GVL.PMAX_VALUE;

  // ----------- corte de cargas por nivel de prioridade ---------------

  IF C7 = TRUE THEN
    IF GVL.INV_POWER > P_10H THEN
      C7 := FALSE;
      TIMER_CORTE_CONTACTOR := GetDateTime(); //
    END_IF
  END_IF

  IF GetDateTime() > TIMER_CORTE_CONTACTOR + 2500 THEN //tempo resposta da variavel
    IF C4 = TRUE THEN
      IF GVL.INV_POWER > P_10H THEN
        C4 := FALSE;
        TIMER_CORTE_CONTACTOR := GetDateTime(); //
      END_IF
    END_IF
  END_IF

  IF GetDateTime() > TIMER_CORTE_CONTACTOR + 2500 THEN //tempo resposta da variavel
    IF C8 = TRUE THEN
      IF GVL.INV_POWER > P_10H THEN
        C8 := FALSE;
        TIMER_CORTE_CONTACTOR := GetDateTime(); //
      END_IF
    END_IF
  END_IF

  IF GetDateTime() > TIMER_CORTE_CONTACTOR + 2500 THEN //tempo resposta da variavel
    IF C3 = TRUE THEN
      IF GVL.INV_POWER > P_10H THEN
        C3 := FALSE;
        TIMER_CORTE_CONTACTOR := GetDateTime(); //
      END_IF
    END_IF
  END_IF

  IF GetDateTime() > TIMER_CORTE_CONTACTOR + 2500 THEN //tempo resposta da variavel
    IF C5 = TRUE THEN
```

```
45          IF GVL.INV_POWER > P_10H THEN
46            C5 := FALSE;
47            TIMER_CORTE_CONTACTOR := GetDateTime(); //
48          END_IF
49        END_IF
50      END_IF
51
52      IF GetDateTime() > TIMER_CORTE_CONTACTOR + 2500 THEN //tempo resposta da variavel
53        IF C6 = TRUE THEN
54          IF GVL.INV_POWER > P_10H THEN
55            C6 := FALSE;
56            TIMER_CORTE_CONTACTOR := GetDateTime(); //
57          END_IF
58        END_IF
59      END_IF
60
61      IF GetDateTime() > TIMER_CORTE_CONTACTOR + 2500 THEN //tempo resposta da variavel
62        IF C2 = TRUE THEN
63          IF GVL.INV_POWER > P_10H THEN
64            C2 := FALSE;
65            TIMER_CORTE_CONTACTOR := GetDateTime(); //
66          END_IF
67        END_IF
68      END_IF
69
70
71      // ------------ realimenta  o de cargas por nivel de prioridade -------------
72
73      REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
74      REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
75      IF C2 = FALSE THEN
76        IF REATIVAR > 120000 AND GVL.SOC_ESS > 83 THEN
77          IF GVL.INV_POWER + GVL.EM2_POWER < P_10H THEN
78            C2 := TRUE;
79            TIMER_FECHAR_CONTACTOR := GetDateTime();
80          END_IF
81        END_IF
82      END_IF
83
84      IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 2500 THEN
85        IF C6 = FALSE THEN
86          REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
87          REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
88          IF REATIVAR > 120000 AND GVL.SOC_ESS > 83 THEN
89            IF GVL.INV_POWER + GVL.C6_VALUE < P_10H THEN
90              C6 := TRUE;
91              TIMER_FECHAR_CONTACTOR := GetDateTime();
92            END_IF
93          END_IF
94        END_IF
95      END_IF
```

```
96
97    IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 2500 THEN
98      IF C5 = FALSE THEN
99        REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
100       REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
101       IF REATIVAR > 120000 AND GVL.SOC_ESS > 83 THEN
102         IF GVL.INV_POWER + GVL.C5_VALUE < P_10H THEN
103           C5 := TRUE;
104           TIMER_FECHAR_CONTACTOR := GetDateTime();
105         END_IF
106       END_IF
107     END_IF
108   END_IF
109
110   IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 2500 THEN
111     IF C3 = FALSE THEN
112       REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
113       REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
114       IF REATIVAR > 120000 AND GVL.SOC_ESS > 83 THEN
115         IF GVL.INV_POWER + GVL.EM3_POWER < P_10H THEN
116           C3 := TRUE;
117           TIMER_FECHAR_CONTACTOR := GetDateTime();
118         END_IF
119       END_IF
120     END_IF
121   END_IF
122
123   IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 2500 THEN
124     IF C8 = FALSE THEN
125       REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
126       REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
127       IF REATIVAR > 120000 AND GVL.SOC_ESS > 83 THEN
128         IF GVL.INV_POWER + GVL.C8_VALUE < P_10H THEN
129           C8 := TRUE;
130           TIMER_FECHAR_CONTACTOR := GetDateTime();
131         END_IF
132       END_IF
133     END_IF
134   END_IF
135
136   IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 5000 THEN
137     IF C4 = FALSE THEN
138       REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
139       REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
140       IF REATIVAR > 120000 AND GVL.SOC_ESS > 83 THEN
141         IF GVL.INV_POWER + GVL.C4_VALUE < P_10H THEN
142           C4 := TRUE;
143           TIMER_FECHAR_CONTACTOR := GetDateTime();
144         END_IF
145       END_IF
146     END_IF
```

```
147     END_IF
148
149     IF GetDateTime() > TIMER_FECHAR_CONTACTOR + 5000 THEN
150       IF C7 = FALSE THEN
151         REATIVACAO_TIMER_CORTE_CONTACTOR := GetDateTime();
152         REATIVAR := REATIVACAO_TIMER_CORTE_CONTACTOR - TIMER_CORTE_CONTACTOR;
153         IF REATIVAR > 120000 AND GVL.SOC_ESS > 83 THEN
154           IF GVL.INV_POWER + GVL.C7_VALUE < P_10H THEN
155             C7 := TRUE;
156             TIMER_FECHAR_CONTACTOR := GetDateTime();
157           END_IF
158         END_IF
159       END_IF
160     END_IF
161
162     // --------------------------- mudanca de estado ------------------------------
163     IF GVL.SOC_ESS < GVL.MIN_SOC_ESS THEN
164       STATE := 45;
165     END_IF
166     IF GVL.PO = 0 THEN
167       C1 := FALSE; // REDUNDANTE SEGURANCA
168       C10 := FALSE; // abrimos o inversor (CONTACTOR 10)
169       GVL.TRANSITION_SECURITY := GetDateTime(); // set timer
170       STATE := 45;
171     END_IF
```

Listing 20: *Main program - State 40 code.*

.1.2.7 State 0 - Battery SOC below 20%

There is only one condition to verify the control variables and transition back to state 45 when verified.

```
1   0:
2     C1 := FALSE;
3     C10:= TRUE;
4     C2 := FALSE;
5     C3 := FALSE;
6     C4 := FALSE;
7     C5 := FALSE;
8     C6 := FALSE;
9     C7 := FALSE;
10    C8 := FALSE;
11
12    IF GVL.PO = 0 OR (GVL.PO = 1 AND GVL.SOC_ESS > GVL.MIN_SOC_ESS) THEN
13      C1 := FALSE; // REDUNDANTE SEGURANCA
14      C10 := FALSE; // abrimos o inversor (CONTACTOR 10)
15      GVL.TRANSITION_SECURITY := GetDateTime(); // set timer
16      STATE := 45;
```

```
17      END_IF
```

Listing 21: *Main program - State 0 code.*

.1.3 Victron Modbus TCP Register List

https://www.victronenergy.com/upload/documents/CCGX-Modbus-TCP-register-list-3.30.xlsx

.1.4 Energy Management Energy Analyzer Type EM111 Manual

https://www.mouser.com/datasheet/2/1032/em111ds-1805365.pdf

.1.5 TM241CE24R PLC Datasheet

https://iportal2.schneider-electric.com/Contents/docs/TM241CE24R_DATA%20SHEET.PDF